

タスク管理術

著者：関 勝寿 公開日：2024 年 6 月 23 日 ソース：<https://sekika.github.io/2024/06/23/TaskManagement/>

タスク管理、つまり ToDo リストの管理について、私が採用している方法を紹介する。これまでの人生で色々と試行錯誤しながらやり方を変えてきたが、この方法は 10 年以上安定稼働している。重要なことは以下の 3 点である。

- ・タスクの日付を締切日に設定しない。
- ・今日やることが今日のタスクとして表示されるようにする。
- ・タスクをサブタスクに分割し実行する順番に列記する。

1. アプリの要件

「タスク管理アプリ」「ToDo リストアプリ」は**様々なものがある**が、以下の要件を満たすものを使う。この程度の要件であれば、多くのアプリが満たしているであろう。

- ・タスクに名前をつけて登録できる。
- ・タスクに日付を割り当てることができる。今日のタスクを一覧できる。
- ・タスクの詳細を書き込むことができる。詳細に URL を書き込むとリンクが作成される。
- ・タスクを「完了」することができる。
- ・PC、スマホ、タブレットのどこからでも同じデータを閲覧・編集できる。

私の場合は **ToodleDo** を 2010 年から使っていたが、2023 年から **Google ToDo リスト** に乗り換えた（サードパーティアプリ）。

2. タスク管理方法の基本

まず、一番重要なことは**タスクの日付を締切日に設定しない**ことである。多くの人が、タスク管理ツールを導入したときにタスクの「締切日」を設定してしまうため、締切日までそのタスクに取りかかることなく、締切日にそのタスクを実行する余裕がなくて終わってしまうというミスを繰り返している。また、タスクの「重要度」などという機能は使わない。「いつが締め切りか」でタスクを考えるのではなく「いつ次のアクションをするか」でタスクを管理することで、常に**今日やることが今日のタスクとして表示される状態**を作ることが肝である。なぜならば、タスク管理で重要なことは「たくさんのタスクの中で、今やるべきタスクは何か」を把握して、それ以外のタスクは忘れることだからである。

新しいタスクを登録する場合には、基本的に「今日の日付」で登録する。そして、必要な情報を詳細に書き込む。たとえば、メールに返信するというタスクであれば、Gmail 上の当該メールのリンクを貼る。締切日はここに書く。タスク発生時には詳細な情報を書き込む余裕がない場合もあるため、そのときにはとにかくタスクの登録をして「今日の日付」で登録をするところまでをしておく。ある程度詳細を記入して、今日やらないのであればそれから別の日に設定し直す。

なお、タスクの登録をするよりも実行する方が早い場合には、即座に実行する。たとえば「メールの返信を書く」というタスクは、その場ではメールを書く余裕がないためにタスクとして登録をするのであって、その場で返信してしまうことで「タスクの登録をしない」ことが可能であればそちらが優先される。

複雑なタスクの場合には、まずは**サブタスクに分割して実行する順番に列記**するところから始める。たとえば、あるタスクの詳細に

- ・サブタスク 1
- ・サブタスク 2
- ・サブタスク 3

と書いてある場合には、次に「サブタスク 1」をいつ実行するかを考える。それが今日であれば日付は今日のままであり、明日実行するのであれ

ば明日の日付にする。サブタスク 1 が完了したら消去し、次にいつサブタスク 2 を実行するかを考える。すなわち、一連の仕事は 1 つのタスクとして管理する。サブタスクを別々のタスクとして登録すると、タスク全体を整理して考えるにくってしまうためである。

3. 今日のタスクの選別

「今日実行するタスク」が今日のタスクとして入っているため、それを順次実行すれば良い。タスクが多すぎて「必ず今日中にやらなければいけないこと」がすぐにわからない場合には、「今日やらなければいけないこと」だけを残してそれ以外は明日またはそれ以降の日に設定し直す。今日のタスクが完了して余裕がある場合には、明日以降のタスクの中から今日やることを選んで今日に設定する。このようにして「今日のタスク以外のことは考えない」状態を作ることが重要である。

なお、大変な仕事、あまりやりたくない仕事で締め切りが遠い場合、あるいは締め切りが設定されていない場合は、後回しにされがちである。何度か次の日や翌週に日付を設定し直して、ようやく着手する場合もある。このように、日付を設定し直すという作業を繰り返すことで、着手しないまでもその仕事について定期的に思い出すという効果がある。定期的に思い出すのも面倒なのでやってしまうか、ということで仕事にとりかかることもある。

4. ファイル管理

「詳細」に書く事項が増えた場合、またタスクに関連するファイルがある場合には、別のシステムで管理してその URL を詳細欄に記入する。そのシステムは **Notion**, **Evernote**, **Slack** 等、自分が使いやすいものであればなんでも良いが、当該情報の URL を取得できて、スマホからでもタスクの内容を確認できることが望ましい。私の場合は、職場で導入されている **ガルーン** を使うか（自分専用のスペースを作って、そこにタスクごとのスレッドを立てる。共同作業の場合にはそもそもそのためのスレッドが立っている。）、あるいは **GitHub** のプライベートリポジトリでファイルを管理している。

GitHub を使う場合は、タスクに対応するフォルダを作成し、**Readme.md** にマークダウンでタスクを整理する。そのマークダウンファイルの **GitHub** における URL をタスクの詳細に書く。たとえば「学会発表する」というタスクであれば、出張申請、学会の登録、交通・宿泊の確保、発表の準備、当日の発表、出張報告など様々なサブタスクが発生するため、このようにマークダウンで整理して実行する。マークダウンとして記録が残るため、後日同様のタスクが発生した時にある程度使い回すことが可能である。そのような場合には、サブタスクが完了した時にそれを消すのではなく「完了」というセクションを作ってそこに移動しておく。

チームで仕事をしている場合には、チームとしてのタスク管理ツールが導入されている場合もあるであろう。そのような場合であっても、個人としてのタスク管理は必要となる。なぜならば、個人のタスクには「チームとしてのタスク」以外の細々としたタスクが発生するためである。そのような場合には、個人のタスクツールには「チームとしてのタスク」の項目を書き、チームのタスクツールへのリンクを貼る。

5. 定期的なタスク

定期的なタスクはタスクが完了したときに「完了」するのではなくて日付を設定し直すことで管理する。毎週水曜日に実行するタスクがあれば、そのタスクは次の水曜日に設定する。そして、水曜日にそのタスクが終わったら、タスクを「完了」するのではなく、次の週の水曜日に設定する。毎月実行するタスクも同様に、タスクが完了したら来月の実行する日に設定する。「免許更新」というタスクが完了したら、次の免許更新日近くに設定して、免許の有効期限を詳細に書き込んでおく。

曜日ごとのルーチンワークを1つのタスクとしてまとめることも可能である。たとえば、私の場合は特に授業に関連するタスクが多いため「授業準備」という名前のタスクを作成し、マークダウンファイルに曜日ごとのルーチンタスクを記述している。たとえば、この曜日にはこの科目の小テストの問題を作成して、この曜日にはその小テストを採点して、といったように曜日ごとにやるべきことを一覧としているため、月曜日のタスクが完了したら火曜日に日付を変える、といったような形で授業に関するルーチンのタスクを1つのタスク項目でまとめている。時々、大量のレポートの採点など大きめのタスクが入る場合には、別途タスクを作成する。

6. 待ちのサブタスク

あるタスクをサブタスクで分割した時に「待ち」のサブタスクが入る場合がある。たとえば、ある事項を確認するメールを送るというサブタスクが終わると、次は「返信を待つ」というサブタスクとなる。この場合は、状況に応じて翌日、あるいは1週間後などにタスクを移動する。その日になっても待ちが続いていればまたさらに後の日にタスクを移動するということを繰り返すこととなるが、催促の必要があれば「催促」などのサブタスクを入れておく。

7. 優先度の低いタスク

実行する義務はないが、時間があればやってみたいことなどは、適宜時間がありそうな日付にタスクを設定しておく。そしてその日になってもやる気が出なければ、あるいは忙しければ、また数ヶ月後などに設定しなおす。実行する見込みがなくなったらタスクを完了する。

8. スケジュール管理は切り離す

スケジュールはGoogleカレンダーなどで別途管理し、タスク管理とは切り離す。すなわち「日時が確定している予定」はスケジュールで「日時が確定していない仕事」はタスクで管理する。ただし、たとえばある日に設定された会議になんらかの事前準備が必要であれば、タスクが発生するためタスクとして登録する。スケジュールとタスクを同じアプリで管理することも可能であるが（たとえば、Googleカレンダーでタスクを管理する、あるいは逆にTodoリストアプリでスケジュールを管理する）、やってみるとこれはかえって混乱する。

なお、スケジュールは実際にはGoogleカレンダーではなくGitHubとDiscordで管理しているが、これは特殊すぎるため詳細は割愛する。