

特別試読版

逆引きで分かる!!

Redmine ハンドブック

バージョン

5.0

対応



川端 光義 著

わかりやすいチケットを書く
今やるべきチケット表示する
関係しあうチケットを辿りやすくする
ガントチャートで進捗を管理する
プロジェクトメンバーの活動状況を把握する



非売品

特別試読版

REDMINE HANDBOOK

CHAPTER

6

タスク・プロジェクト 管理とは

本紙は近日発売予定の『Redmine ハンドブック バージョン 5.0 対応版』の CHAPTER 6 を試読いただくことを目的に制作されています。発売時の書籍の内容とは異なりますこと、ご理解ください。





プロローグ

夏休みの宿題がギリギリになるのはなぜ？

子供の頃の夏休みの宿題、あなたは休み期間のどれぐらいで終わらせていましたか？前半で終わってしまう人もいれば、最終日ギリギリに終わらせる人もいましたよね。学校からは「宿題を終わらせる」という「タスク」は与えられるものの、それをどのように終わらせていくのかという「方法」については教えてくれません。

ある調査によると、「全体の6割強の人が夏休み前半までに宿題を終わらせたいとっていました。ところが……」。なんと実際には約3人に1人、34.9%が最終日ギリギリまでかかったと答えていました（電通リサーチ調べ）。

勉強は本来子供にとってやりたいことではないでしょう。面倒なこと、義務としてやらなければならないことは、なかなか取り掛かりにくいものです。それは、人間にはつらいことは後回しにしたいという性質があるためです。

同じようなことは、仕事においても言えます。やらなければという義務感でタスクをこなす人も多いでしょう。

きちんとやっているつもりでも、モチベーションが影響し思ったほど進まず、日々やることに追われ、どんなにこなしてもいつも期日が迫っている状態、言わば夏休みの最終日がずっと続いているような感覚になってしまうのです。

コツコツと計画通りに進めていくためには、タスク管理やプロジェクト管理はとても重要です。

この章でタスク管理やプロジェクト管理のノウハウを知っておくことで、Redmineというツールの効果を高めることができます。単なるタスク・プロジェクト管理ツールとして利用するのではなく、ツールをうまく活用して、自分の思うように仕事を進めていけるようになることを目的としています。

タスク管理・プロジェクト管理を学んだ夏休みの宿題プロジェクト

タスク管理・プロジェクト管理を学ぶと、夏休みの宿題プロジェクトはどのように進められるかを考えてみます。

1. 目標を立てる

夏休みが7月20日から8月31日までとします。ギリギリにはしたくないので、余裕を10日ちょっと設け、8月19日までに宿題を終わらせるという目標を立てます。そうすると、宿題ができる期間は30日となります。

2. 計画を立てる

まず1日に宿題ができる時間を考えます。人によってはクラブ活動に励んで、それほど時間が取れない人も居るでしょう。

次に、宿題の全体ボリュームを把握し、1日に進められる宿題の量を考えてみます。1日10ページ、ドリルを進められると思えば、全体で100ページあるなら10日でドリルが終わるという計画を立てられます。

3. やってみて検証する

計画が思い通りに行かないことはよくあります。

実際に宿題をやってみて、計画していた1日10ページのドリルがどのくらい時間がかかるのかを測ります。

1日5ページしか進められなかったとすると、ドリルを終えるのに20日掛かるといことが判明します。ドリル以外にも宿題があり、このままでは計画通りに宿題が終わらないということが分かるのです。

間違いのない計画を立てることが大事ではなく、プロジェクトの初期に間違いに気付くことが大事なのです。

4. 計画を修正する

なぜ想定通りに宿題が進められなかったかをタスク管理の中で振り返ります。分からないことにつまずいていたことが原因であれば、友達に教えてもらうことで解決するかもしれません。

そもそも想定が甘かったという見積もりの間違いに気付くかもしれません。それらを事実として受け止めて、計画を修正する対策を考えます。

1日に宿題をする時間を増やす。家では集中できないから図書館に行き効率を上げる。目標を遅らせる。などです。

そして、改めて実績に基づいた計画を立て直します。

こうすることで、夏休み最後の方になって宿題に追われる日々になるのではなく、毎日、宿題を計画通りに進めることができます。日々終わったという達成感を得られ、翌日も宿題に対して前向きに取り掛かれるほどになるかもしれません。



6-1

タスクとタスク管理を理解する

6-1-1 タスクとは何か

タスクとは、具体的にイメージできる作業のことです。

例えば、

- ・ 日報を書く
- ・ A社向け請求書を発行する
- ・ 山田さんにメールを送る

など、作業を行う人がその内容を理解できていれば、それはタスクと言えるでしょう。上記では、山田さんにメールを送る内容を具体的にイメージできていれば、タスクと言えます。

図 6-1-1 山田さんにメールを送る作業のイメージ

山田さんにメールを送信する

-  連絡内容
山田さんに訪問日時の連絡する
-  メールの宛先
山田さん
-  メール本文に記載する内容
目的：定例ミーティング
訪問日・時間



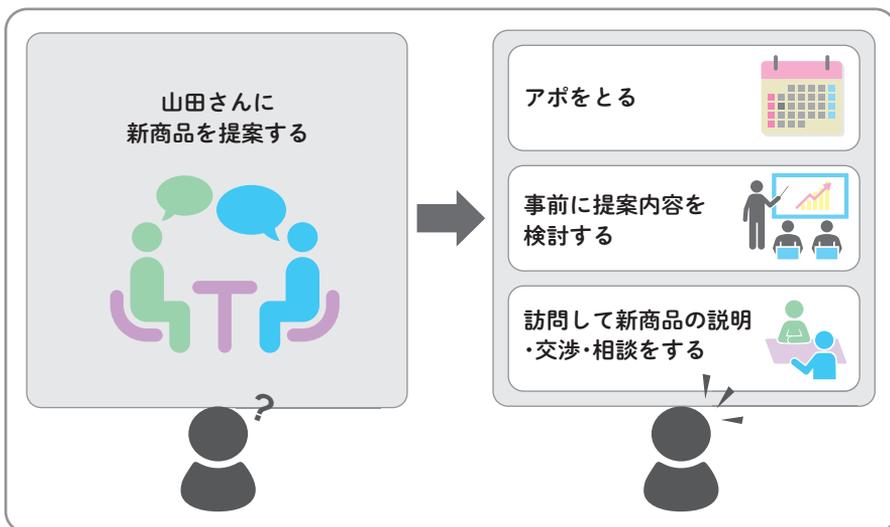
一方で、「山田さんと商談する」というタスクは、今すぐにできることでもありませんし、具体的にイメージすると、やるべきことが複数出てきます。

まずは「山田さんにアポを取る」というタスクから始め、「商談のための資料を作成する」タスクができてようやく、「訪問して商談する」というタスクができます。

このように具体的にイメージできる作業になるまでタスクを分解していくことをタスク分割といいます。

どこまでタスク分割を行うかは、経験や人によっても異なり、「商談のための資料を作成する」が具体的にイメージできなければ、更にタスク分割を行います。

図 6-1-2 タスク分割をしているイメージ



6-1-2 タスク管理とは何か

タスクにはやるべき内容の他に様々な情報があります。最低限必要なこととして、タスクをやったか、やっていないかという**タスクの状態（ステータス）**です。チームで仕事をする場合、そのタスクは誰が担うのかという、**担当者**の情報。そのタスクはいつまでに終わらせないといけないのかという、**期限**。そのタスクはどれくらい時間がかかるのかという、**作業時間（工数）**など。これらの情報を追加したり、更新していくことが**タスク管理**です。

タスク管理は、作業の優先順位をつけたり、忘れずに全ての作業をこなすために必要不可欠なものです。

6-1-3 ワークフローの必要性

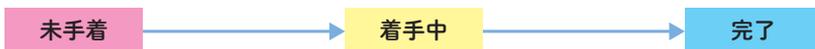
ワークフローとは、仕事（ワーク）の流れ（フロー）を意味します。自分個人のタスク管理であれば、やったかやっていないかだけの管理で問題ないでしょう。タスクを箇条書きにして消し込んでいくだけで良いです。これがチームにおけるタスク管理となると、そうもいかない状況があります。自分の予定していたタスクが早く終わり、チームのタスクを率先してやれることもあるでしょう。その場合に、やろうとしたタスクが実は他のメンバーがやっている最中だったということが起こります。このようなことにならないために、タスクに着手している「着手中」や「Doing」というステータスを追加してメンバーに示せると良いでしょう。

別のケースとして、タスクを依頼した人と担当者が異なる場合、担当者がタスクを終えたら終わりということもなく、依頼した人はタスクの成果が合っているかどうかを確認する必要もあります。この場合、「作業終了」の後に、依頼者による「確認中」や「レビュー中」というステータスを経て、確認が問題なければタスクの「完了」となります。

このようなタスクの流れのことを**ワークフロー**といいます。個人レベルのタスクでは必要ありませんが、チームや他の人が関わるタスクになると、ワークフローが必要になります。

図 6-1-3 ワークフローの説明

単純なタスク処理業務



他の人からの作業依頼





6-2

タスク管理の必要性を知る

タスク管理は、自分やチームのやることを管理することであり、その情報をうまく扱うことです。それによって、複数のタスクを複数人で同時に進めることができたり、仕事の状況を把握しやすくなったり、チーム全体をマネジメントしたりもできます。

ただ、人間は不完全な生き物であり、ときに不合理であるため、うまく情報を扱うこと自体が難しいのも事実です。やらなければならないことがあっても、それができないということは多々あります。その理由として、以下のことが考えられます。

6-2-1 記憶力の問題

人間の記憶力には限界があります。複数のタスクを記憶だけで管理するのは難しいでしょう。たとえば、タスクのメモ書きに「メールを送る」と書いたとして、タスクを作ったときにはその内容を理解していても、数日後にやろうとしたら誰に何の内容のメールを送るか忘れてしまうといったこともあります。

対策 → タスク管理ツールに記録する

→ メモに詳細に記入する

6-2-2 タイミングの問題

やるべきタスクを適切なタイミングで思い出せないと意味がありません。たとえば、会議が10時にあることを10時に思い出しても事前準備が間に合いません。

対策 → リマインダーを利用する

6-2-3 モチベーションの問題

タスクを認識したときと、それを実行する段階とで、モチベーションに違いが起こることがあります。人間はモチベーションを自由にあやつれないため、後回しにしたり意欲が低下したりするのです。締め切り間近にならないとモチベーションが高まらなかつ

たり、やりたくないタスクや苦手な作業より興味のある作業を優先したりしてしまいがちです。

対策 → タスクを細分化し達成感を得やすいようにする

6-2-4 認知バイアスの問題

人間の判断基準は一定していません。緊急性が高いタスクがあっても、直近で発生した作業を重視してしまうことがあり、判断ミスが起こることがあります。たとえば、クライアントに大事なメールを送ろうとしてメーラーを立ち上げたら、興味のあるイベントの案内メールが気になって開いてしまうなどといったことがあげられます。

このようなことは認知バイアスの問題であり、行動経済学として有名です。人の不合理な心理によって優先順位などの判断ミスが起こるのです。

対策 → 時間管理のマトリクスで優先順位を付ける

→ タスクリストで優先順位を確認する

6-2-5 マルチタスクの問題

人間が並行して考える能力には限界があります。一つに集中して行うときと並行して行うときとは、効率に差が生じます。たとえば、資料を作りながらチャットをしようとしても、内容が頭に入ってきません。マルチタスクで同時にこなすより、シングルタスクを一つひとつこなしていくほうが良いでしょう。

対策 → ポモドーロテクニックで集中する

こういった人間の不完全性と不合理性を補うために、タスク管理があります。それは単にタスクのやったかやってないかの管理だけでなく、どのように集中して行えるかなどのメソッドも含まれています。

それでは、この後にタスク管理のツールやメソッドを詳しく紹介していきます。



6-3

タスクの種類を理解する

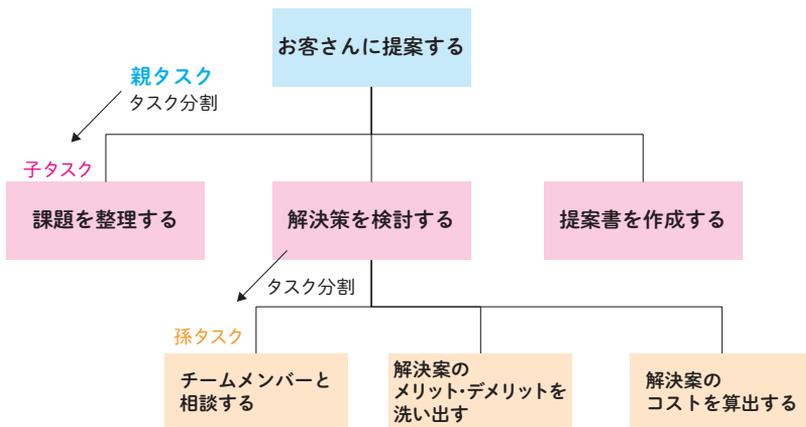
このセクションでは、基本的なタスクの種類について解説します。

6-3-1 サブタスク（親子タスク）

タスクとは具体的にイメージできる作業のことですが、具体的にイメージできない場合、タスク分割を行います。そうして複数に分割されたタスクのことを**サブタスク**または**子タスク**といいます。分割前のタスクは抽象的な位置付けとなり、親タスクと言えます。Redmine の呼び方では「親子チケット」に当たるでしょう。

「お客さんに提案する」というやるべきことが発生した場合、たとえば経験豊富なコンサルタントであれば、それを一つのタスクとして扱えるでしょう。解決策も分かっている、提案書の作り方も頭に入っているからです。しかし、仕事を覚えただけの社会人ならそうはいきません。具体的にイメージできるまでタスクを分割します。分割した子タスクでもイメージできなければさらに分割し、それらは孫タスクとなります。

図 6-3-1 お客さんに提案するをタスク分割したサブタスク



6-3-2 スケジュール

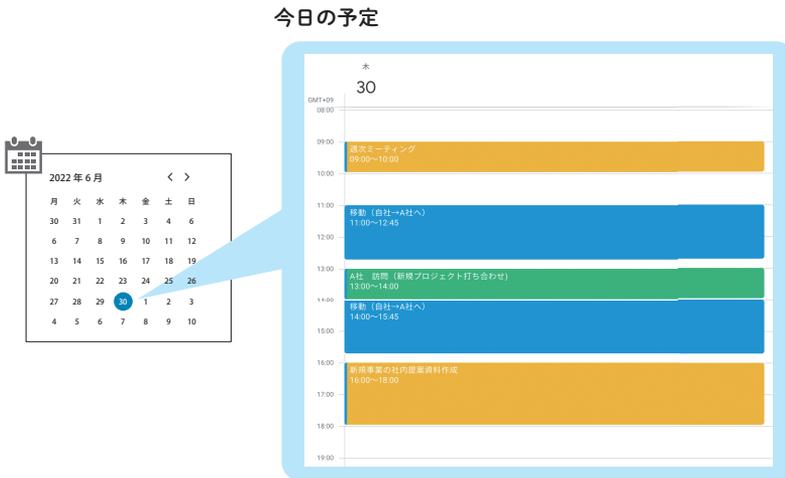
スケジュールは日時が決まっている予定であり、多くの場合、他者との約束です。

タスクは自分の好きなタイミングで開始できますが、スケジュールは開始時間が決まっているもので、主にカレンダーで管理されます。ミーティングや訪問、新幹線の移動もスケジュールです。

他者との約束が多い人は、タスク管理の多くがスケジュール管理になります。スケジュールを先に埋めて、空いた時間にタスクを行っていきます。

また、すべてのタスクをスケジュール化する方法もあります。タスクの開始時間をカレンダーに入れることで、自分との約束にしてしまいます。そうすることで、タスクとスケジュールが一つにまとまります。

図 6-3-2 カレンダーにスケジュールが埋まっている様子



6-3-3 ルーチン

ルーチンは、繰り返し行われるタスクのことです。**リピートタスク**ともいいます。

毎朝メールチェックをする、毎月請求書を送るなど、毎日、週一、月一などの単位で繰り返されるものは、**日次ルーチン**、**週次ルーチン**、**月次ルーチン**と呼ぶことができます。

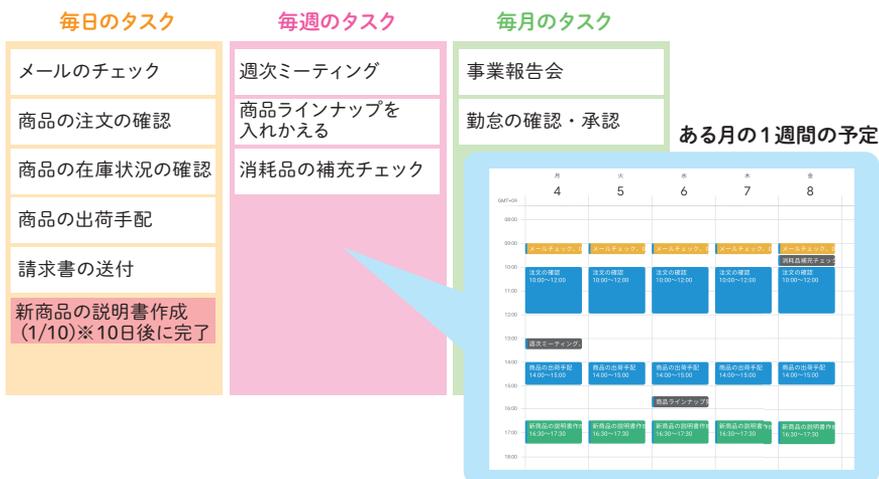
100 ページもの資料を作成するために、一日 10 ページずつ作業をしていく（10 個のタスクに分割する）といったやり方も目的を持ったルーチンです。

タスク管理においては、ルーチンに対するスタンスが 2 つあります。

一つは、全く無視してしまうスタンスです。習慣化されているルーチンであれば、わざわざ管理しなくていいという考え方です。

もう一つは、ルーチンをすべてタスク管理に組み込むスタンスです。この方がタスク全体の管理はしやすくなり、ルーチンも含めて業務時間内に収まるかを確認できます。

図 6-3-3 毎朝のルーチンのチェックリスト





6-4

タスク管理のツールを理解する

このセクションでは、タスク管理の代表的なツールについて解説します。

6-4-1 タスクリスト

タスク管理の基本となるツールです。そのままタスクのリストですが、表現方法は様々あります。シンプルに箇条書きになっているもの、スプレッドシートとしてタスクに関する情報も管理しているもの、付箋紙にタスクを書いてステータスを進めると移動するカンバン方式のもの、すべてタスクリストです。

タスクリストについては、6-5「タスク管理のリストを理解する」p.15で様々なリストを紹介します。

図 6-4-1 シンプルな箇条書きのタスクリスト

顧客への提案資料を作成する

打ち合わせ日程を決める

打ち合わせ結果を提案書に反映する

提案書（最終版）を顧客に提出する

他社へ類似提案ができないか検討する

6-4-2 リマインダー

タスクを思い出させてくれるものが**リマインダー**です。リマインダーには、アクティブ（能動的）なりマインダーと、パッシブ（受動的）なりマインダーの2種類があります。

アクティブリマインダーは、デジタルツールを介して設定した時刻になったらアラームやメールなどでタスクを知らせてくれる形式です。よく使われるのがスマホアプリにも搭載されている「リマインダー」です。Redmineにも通知機能があります。ツールの方から働きかけてくれて、「山田さんにメールを送る」といった通知が来ることでタスクを思い出すことができます。

パッシブリマインダーは、やるべきことをメモしておいて、そのメモを見ることで思い出す形式です。たとえば、付箋紙やメモ帳にタスクを書いてモニターに貼っておくとそれがリマインダーとなります。これは仕事をするうえで必ず目に入る場所にあって、それを見に行つてタスクを思い出すので、自分からの働きかけが必要なものです。明日必ず持っていくものを玄関に置いておくというのもパッシブリマインダーです。

図 6-4-2 スマホから通知が来たリマインダー



6-4-3 カレンダー

日付や時間と紐づけてスケジュールを管理するのが**カレンダー**です。すべてのカレンダーはスケジュールを俯瞰するためにも使われます。単に「何月何日に何をやる」というのを確認するだけでなく、その前後にどんなスケジュールが入っているかを確認するのも目的です。

1ヶ月単位や週間の表示などデジタルツールであれば、任意で表示期間を変更することもできます。スケジュールは1回限りのものと、週次ミーティングなどのルーチンがあります。デジタルカレンダーであれば、繰り返し設定を行うと、自動的に以降のスケジュールも登録され、忘れるのを予防できます。

カレンダーを毎朝見る習慣がついていれば、それ自体がパッシブなりマインダーにもなります。

図 6-4-3 週表示のカレンダー





6-5

タスク管理のリストを理解する

このセクションでは、タスク管理における様々なリストを解説します。

6-5-1 オープンリストとクローズドリスト

タスクを管理する時、2つの側面があります。「**タスク全体を把握したい**」俯瞰の側面と「**今やるべきことに集中したい**」集中の側面です。

タスク管理のリストにおいて、この2つを表しているのが**オープンリスト**と**クローズドリスト**です。

オープンリストは、いくらでもタスクを追加できる状態を指します。そしてそれはタスク全体を俯瞰できるリストとなります。

クローズドリストは、いったんリストを作れば原則は追加できないものです。追加できないため、リストの意味する時間内に、確実にタスクをこなしていくことができ、集中するためのリストとなります。

扱うリストが、どちらの側面なのかで、オープンなのかクローズドなのかを意識して管理すると良いでしょう。

6-5-2 マスターリストとフィルターリスト

タスク管理のリストは2つの概念に分けられます。1つは**マスターリスト**であり、すべてのタスクを集めたものです。もう1つは**フィルターリスト**で、マスターリストから一部分だけを抽出（フィルタリング）したものです。

マスターリストはオープンリストであり、タスク全体を俯瞰できますが、作業する際に数が多すぎる場合には、フィルタリングします。

6-5-3 タスクリスト

タスクとは具体的にイメージできる作業でしたから、その作業が並んでいるリストが**タスクリスト**です。

- ・山田さんにメールする
- ・提案書を作成する
- ・アプリケーションをアップデートする

すべて「やるべきこと」のタスクとして洗い出しますが、中には具体的にイメージできていないタスクもあります。その場合、タスクリストが階層化され、タスク分割されたサブタスクリストとなります。

- ・山田さんにメールする
- ・提案書を作成する
 - ・課題をヒアリングする
 - ・解決案を検討する
 - ・提案書としてまとめる
- ・アプリケーションをアップデートする

タスクリストは、いつでもタスクを追加でき、全体を俯瞰するために**オープンリスト**として扱います。

タスクリストとカレンダーの併用

タスクリストとカレンダーを併用する場合は注意が必要です。タスクとスケジュールを管理するツールが2つになるので、片方のツールを見忘れて、タスクの実施もれが生じてしまうということもあり得ます。それを避けるための工夫として、常に2つのツールを表示してすぐに気付ける状態にする。メインとなるツールをどちらかにしてそのツールに情報を集約するなど、自分が管理しやすい方法で管理します。

例えば、タスクリストメインで使用するなら、朝一番に当日のカレンダーの情報をタスクリストに転記します。逆に、スケジュールメインで使用するなら、その日のタスクリストをカレンダーに書き込むと良いでしょう。

筆者はタスクリストを付箋紙でデスク上に並べて、すぐ横に iPad でカレンダーを常に表示させています。

6-5-4 デイリータスクリスト・ウィークリータスクリスト

デイリータスクリストは、今日やることに加え、ルーチンのタスクも含めた、実際にやることのすべてを網羅するリストです。「メールの返信をする」「会議資料を確認する」といった小さな粒度のタスクまで入ります。ちょっとした隙間時間にも細かいタスクを入れ込み、1日のすべての作業を表します。

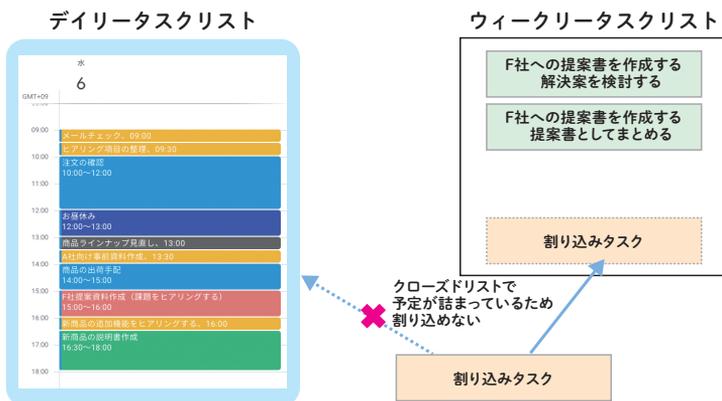
仕事としてはその日だけの計画ではなく、1週間先までのやるべきことを計画しておくことは大事です。

ウィークリータスクリストは1週間分のタスクをリスト化しますが、デイリータスクリストほど細かくはせず、管理しやすい粒度にしておきます。

今日やるべきデイリータスクリストも朝一番に作成し、常に1週間先のウィークリータスクリストを更新していきます。

デイリータスクリスト、ウィークリータスクリストは、今日中や今週中という制限を設け、タスクを実行していくことに集中したいため**クローズドリスト**として扱います。一方、緊急度の高いタスクが入ってくる場合もあります。その際は、今やるべきことの優先順位を確認してリストの中のタスクと入れ替えるか、別のリストに入れて目の前の作業を終わらせるのを優先するかを検討します。

図 6-5-1 デイリータスクリストに割り込みタスクが入ったときのイメージ



6-5-5 いつかやることリスト

これは今後やるべきもののリストです。今すぐ取り掛かるわけではないのですが、後々やるときに忘れないために保存しておきます。すぐに作業する必要のない場合、あるいはそもそも作業できないような場合にこのリストに追加します。業務上の研修を受ける、オフィスの大掃除などがその例です。このリストを見るタイミングを作らないと、いつかやるつもりでいても、いつまでもやらないままになってしまいます。気がつけばやるべきタイミングを過ぎてしまっているということにならないためにも、いつかやることリストをレビューするタイミングを作ります。そのときにこのリストを見返して、遂行可能であれば他のリストに転記して、進めていきます。訪問したい会社、会いたい人、読みたい本などもリストに入れておくとう有効です。

6-5-6 ペンディングリスト

ペンディングリストは、人に作業を依頼し、その結果を待っているもののリストです。依頼日時と締め切りを明記して管理します。締め切りが近いものが出てきたら、作業の進捗を確認します。できるだけスムーズにやり取りするためには、事前に相手にリマインドを送るなどの対策をしておくといいでしょう。

6-5-7 チェックリスト

毎日の作業、繰り返し行われる作業について、やるべきことの手順を確認するためのリストです。繰り返し作業するタスクを標準化・効率化したり、作業が多かったり煩雑だったりして覚えきれないタスクを忘れないようにすることに役立ちます。数の多いタスクの手順をリスト化しておき、それにしたがって一つひとつ進めると作業が抜けることはありません。打ち合わせやミーティングが設定されたときに、そのミーティングのアジェンダは何か、事前に準備する資料はないか、参加者は誰かなどといったリストがチェックリストになります。たとえば事前資料が必要ない場合でも、それが不必要ということを確認できるでしょう。

チェックリストは、基本的にはクローズドリストとして運用されるものです。作業中はそこに追加することはせず、作業が終わって直すべき点が見つかったときにはリストを修正して運用すると良いでしょう。

図 6-5-2 チェックリストのイメージ

会議前のチェックリスト	
<input checked="" type="checkbox"/>	参加者の確認
<input type="checkbox"/>	会議室の空き状況確認
<input type="checkbox"/>	アジェンダの準備
<input type="checkbox"/>	会議資料の準備



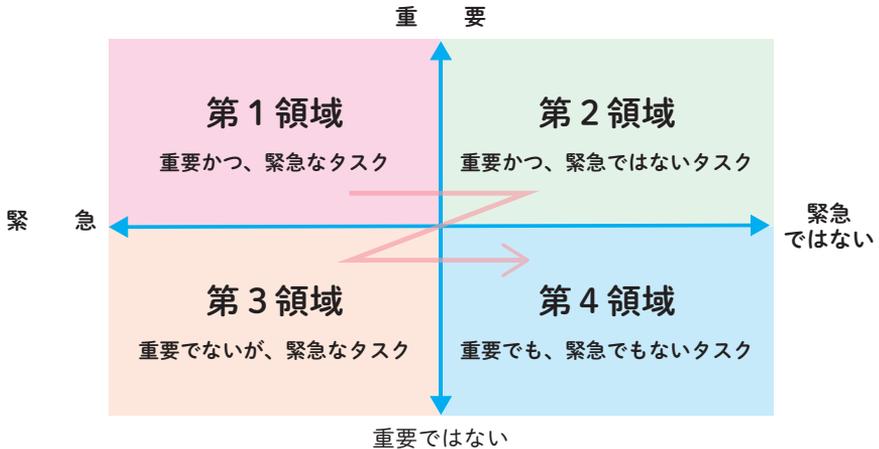
6-6

タスクに優先順位を付ける

タスクが山積みになっている際に、優先順位をつけられずに悩んだ経験はないでしょうか？

タスクが多くてどれから手をつけたら良いかわからない、優先度が高いタスクばかりになってしまう……といったときに役立つのが、優先順位を決めるための手法である「時間管理のマトリクス」です。アイゼンハワーマトリクス、緊急度重要度マトリクスとも呼ばれます。

図 6-6-1 時間管理のマトリクス（米国の作家、コンサルタントのスティーブン・コヴィーが提唱）



6-6-1 時間管理のマトリクスに振り分ける

まずはタスクを第1領域から第4領域に振り分けていきます。

図にあるように、第1領域にある重要かつ緊急のタスクが最も優先順位が高く、次に第2領域の重要かつ緊急ではないもの、続けて第3、4領域の順になります。人は緊急のタスクがあると、重要じゃなくても優先順位を高くしがちです。急に割り込みで入ってきた会議や電話、Slack から通知されるメンションの多くも実際のところ重要ではな

かたりします。第3領域のタスクばかりを先にやってしまい、本当に重要な第2領域のタスクが期日遅れにならないよう、冷静に判断する必要があります。

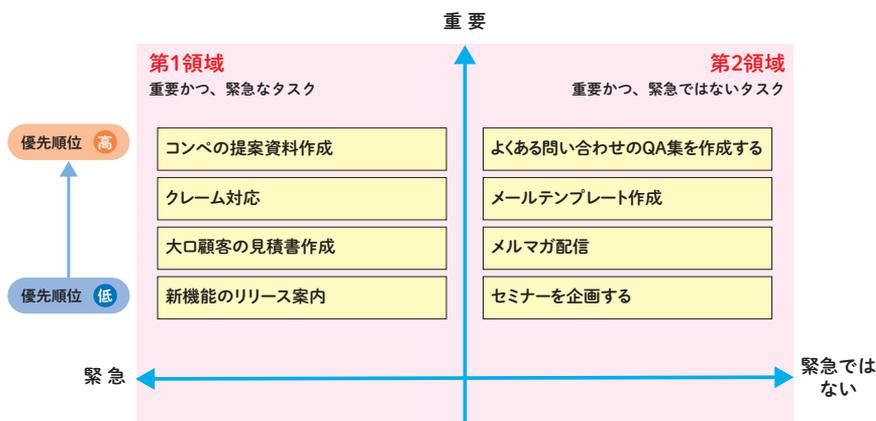
6-6-2 リストで優先順位付けをする

それぞれの領域で複数のタスクが出てきますが、すべて優先度が高いからと、マルチタスクで進めるとは余計に効率が悪くなります。領域の中をリストにして上から一つずつ優先順位を付けていきます。

まずは第1領域と第2領域だけ優先順位付けするだけでも良いでしょう。タスクに求められている期日が順位付けの判断材料にもなります。リスト化した後、俯瞰的に見て優先順位を確認します。

もし緊急の割り込みがあった場合、優先順位付けされていることで、どこに割り込むべきかの判断も付きやすくなります。

図 6-6-2 優先順位付けしたリスト





6-7

タスクの工数を見積もる

6-7-1 工数見積もりの重要性

タスクを完了させるまでにどのくらい時間がかかるか見積もることを、**工数見積もり**と言います。

タスクの工数見積もりを怠り、直感でスケジュールを引くと納期に遅れるリスクが高まるでしょう。

納期間近になり、遅延しそうなことが分かったら、遅れを取り戻すために、残りの日数が残業続きになるかもしれません。そしてスケジュールの遅延報告がギリギリになると、マネージャーは対策を取れず納期遅れを招きます。そうならないためにも、タスクごとに工数見積もりをしておくことで、期日までに終わらない可能性があることが早々にわかり、他の人に手伝ってもらったり期日を延期してもらったりなど事前に調整や対策が取れます。

図 6-7-1 どのくらい時間がかかるか？



6-7-2 1日以内の工数見積もりにする

タスクリストにしている時点で、ある程度具体的にイメージできる作業に落とし込めてはいるはずですが、1タスクにおいて工数が2、3日かかるという場合には、**1日**以

内に収まるまでタスク分割していきます。分割しても1タスクで1日以上かかるボリュームがあるのであれば、数量などで分割します。2日で20ページの成果物なら、10ページずつの2分割です。

そうしておく、チーム内での見積もり精度が上がり、負荷の平準化もしやすくなります。特定の人に負荷が偏るのを防げるでしょう。詳細な見積もりをしていくことで、タスクの洗い出しだけでは分からなかった、タスクの内容や具体的なやり方の不明点が見つかり、想定以上に工数が掛かるということに早めに気付くことができます。

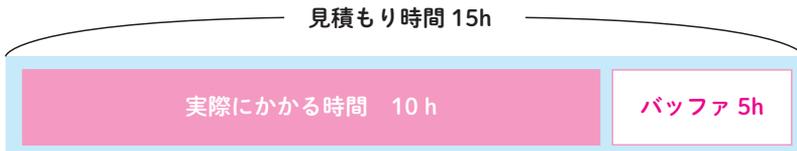
6-7-3 バッファを設ける

バッファは時間的な余裕を設けることです。

想定外の出来事があれば、見積もりをオーバーしてしまいます。見積もり通りに進まない、さらにその次のタスクの期日や予定にも影響していくため、想定外のことにも対応できるようなバッファを設けておくことが重要です。

バッファの設定の仕方は、タスクによって違いはあるものの、一律で見積もり時間の1.2倍、1.5倍などで設定しておくのが良いでしょう。たとえば、10時間かかるタスクであればバッファ込みで15時間としておきます。

図 6-7-2 バッファのある見積もり時間



6-7-4 スケジューリングをする

タスクの見積もりをした後は、直近の日程でスケジューリングします。

2週間のタスクリストやウィークリータスクリストをスケジューリングできると良いでしょう。ウィークリータスクリストの場合、たとえば1タスクあたりバッファ込みで8時間かかるとすると、1週間は40時間として5つタスクをスケジューリングできます。

週の頭にスケジューリングすることで、収まらなかった場合に対策も早めに立てられるでしょう。



タスクの作業時間を記録する

タスクにかかった時間が予定よりもオーバーしていたり、時には予定より早くできていたりすることもあるでしょう。見積もりと違っていたら、後のスケジュールにも影響します。作業時間を記録することで、見積もりが正しかったかどうかを確認でき、予定と違う場合に早めに対策を立てることができます。また、作業効率の改善にもつながります。

図 6-8-1 作業時間を記録する



6-8-1 作業時間を記録する

タスクの着手時に時計を見て、タスクの開始時刻から終了時刻で計算してかかった時間を出すこともできますが、実際にはタスクをやっている最中に休憩したりすると時刻で計算するのは面倒になります。

タスク着手時にストップウォッチで計測して、休憩時間に止めて終了したらその時間を出す方が簡単です。

最近はタイムトラッキング用のアプリがいくつもあります。アプリを利用すると、単なる時間の記録だけでなく作業にかかった時間をグラフで見ることができ、作業の予定時間に対して比較できるものもあります。

予定していた時間よりも実際の作業時間が多い傾向がある場合、割り込みが多いならバッファを設けますし、単純に見積もりが甘いなら見積もりを修正します。



タスク実行のメソッドを理解する

タスクリストの計画ができたら、タスクを実行していきます。ここでは、実行を手助けしてくれるメソッドを紹介します。

6-9-1 ファーストタスク

ファーストタスクは、一番初めのタスク、つまり朝一番に取り掛かるタスクのことです。朝の最初のタスクを決めたら、それだけに集中して取り掛かります。朝は最も体力があり、集中力も高まりやすい時間帯。また、メールや電話、その他の干渉が入りにくいという意味でも集中しやすいです。この時間に一番進めたいタスクを終わらせてしまうという方法です。

6-9-2 タイムボックス

タイムボックスは、時間に区切りを入れて取り組む方法です。意図的に締め切りをつけて自分を追い込みます。たとえば、1日8時間稼働する場合に、1日ではなく2時間ごとの4つのタイムボックスとしてとらえます。そして、それぞれのタイムボックスごとにタスクを決めて取り組みます。このように区切ることで、あと30分で締め切りが来るという状況を定期的作ることができ、メリハリをつけられます。

6-9-3 ポモドーロテクニック

ポモドーロテクニックはタイマーを使い、タスクに集中する時間と短い休憩で構成します。タイマーを25分でセットし、その25分間だけは単一のタスクに集中します。その後5分の休憩を入れ、その30分(25+5)を1ポモドーロとし、4ポモドーロ(2時間)ごとに15~30分の休憩を取ります。これを繰り返します。

ポモドーロテクニックには実行メソッドだけでなく、ワークフローとしてレビューが組み込まれています。作業単位が均一になっているので、タスクが完了するまで何ポモドーロで完了できたか、回数を記録していくことができ、やり方の改善に繋がります。

図 6-9-1 ポモドーロテクニックの流れ



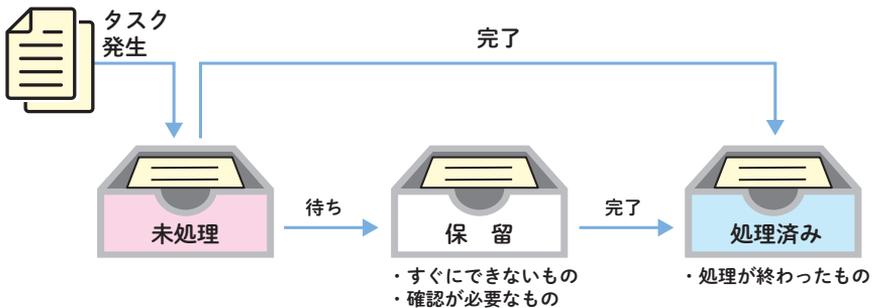


タスク管理のワークフローを理解する

6-10-1 トレーシシステム

トレーシシステムは、3つのトレーを使った書類整理法です。タスクが発生したらまず「未処理トレー」に入れていき、処理が終わったものを「処理済みトレー」に移動させます。すぐできないものや確認が必要なものなどは「保留トレー」へと移動させます。このように、タスクの状態によって置き場所を変え、トレーそのものをリストとして扱うことで、ハッキリと今やるべきタスクはどれなのかがわかります。

図 6-10-1 トレーシシステム



6-10-2 インボックス・ゼロ

インボックスとは、トレーシシステムであげた「未処理トレー」のように、入ってきたばかりでまだ処理されていないタスクを入れる場所のことです。アナログなら書類箱、メールなら受信箱がインボックスとなります。

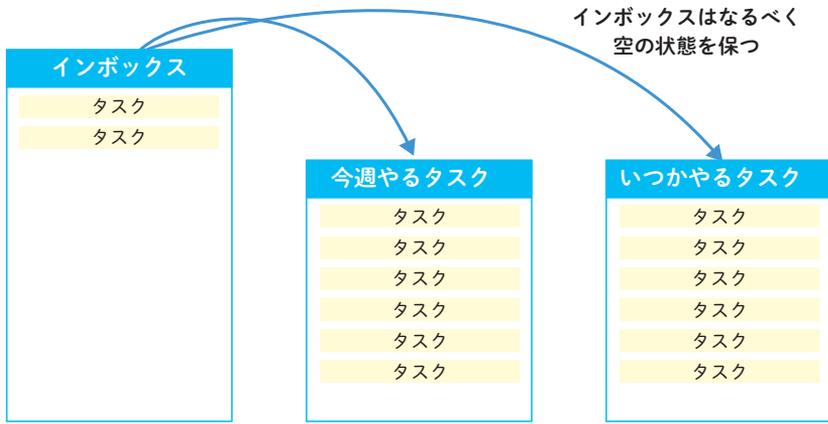
できるだけこのインボックスの中身を空っぽの状態にしておく指針のタスク整理法が**インボックス・ゼロ**です。

メールが溜まってくると、受信箱そのものがタスクリストになってしまいます。次々にメールが入ってくる受信箱はオープンリストであり、常に新しいタスクが追加されて

いる状態。どんなに実行しても終わりがありません。つまり、インボックスは終わりのないタスクリストになってしまうのです。

そのため、受信箱に入ってくるリストをクローズしているタスクリストに振り分け、タスクリストを自分がコントロールしていくことが大事です。

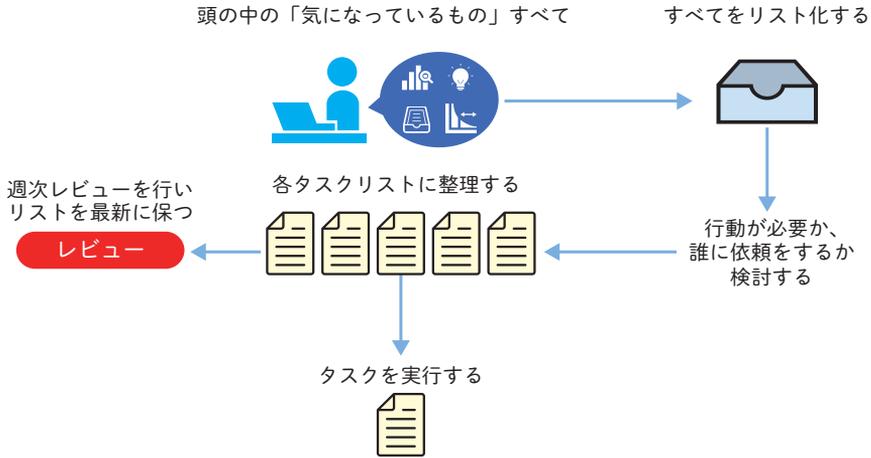
図 6-10-2 インボックス・ゼロ



6-10-3 GTD

GTD は、Getting Things Done を略したタスク管理の手法です。この方法では、タスクを全て書き出し、それらを各リストに振り分けてから、そのリストを参照しながらタスクを実行していきます。GTD のワークフローも、ポモドーロテクニックのように定期的なレビューが組み込まれています。週に1度はレビューを行い、リストを最新の状態にアップデートしていくことが大事です。

図 6-10-3 GTD



6-10-4 マニャーナの法則

マニャーナの法則は、新しく発生したタスクは明日やるというタスク管理です。これは、タスクが発生したら、その日に取り掛かるのをなるべく避けるという方法です。

タスクは新たにどんどんと追加されていくもの。上司からの指示や顧客からの要望などが発生したら、その場で着手したくなるかもしれませんが、それをやってしまうと今日やるべきタスクが終わらず残業になることもあるでしょう。クローズドリストを使い、今日やるべきタスクだけに集中し、新たに発生したタスクは本当に緊急なものを除いていったん翌日以降に先送りします。

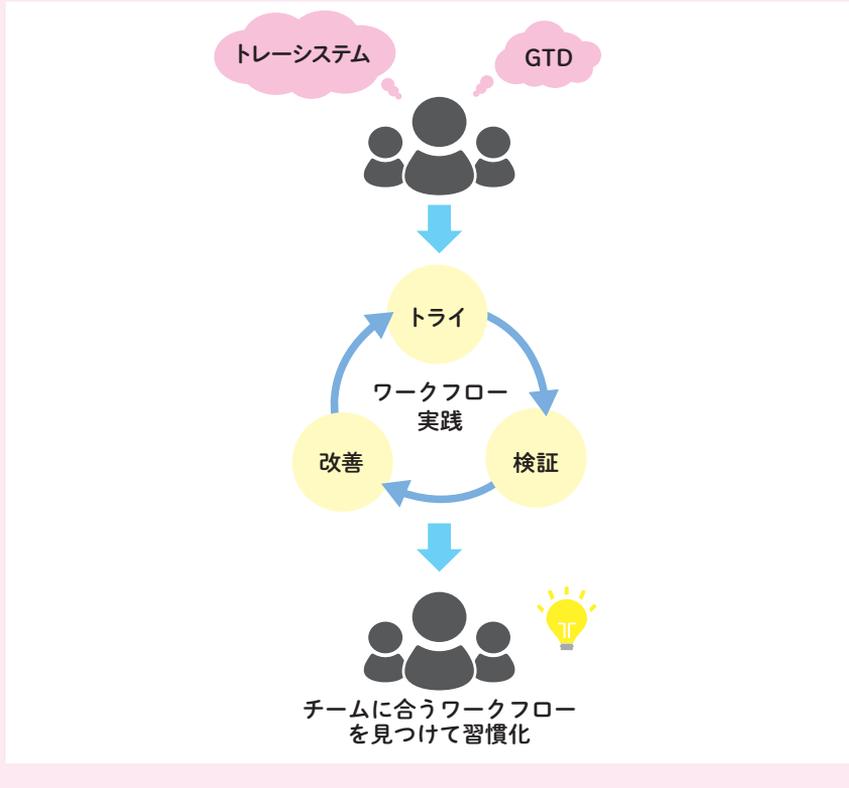
図 6-10-4 マニャーナの法則



タスクを管理することもタスク

タスク管理そのものもタスクです。人間のやることです。タスク管理自体がうまくいかないこともよくあります。自分に合ったメソッドやワークフローを色々試してみて、失敗したら検証、改善、またトライしていく必要があります。そして、タスク管理自体を習慣化させていけるようにすることが大切です。たとえば毎朝カレンダーを見るのが習慣づいていたら、そこにファーストタスクを入れておくなど自分のやりやすい管理方法を取り入れると良いでしょう。そうした習慣化により、より確実により効率良くタスクを実践できるようになっていきます。

図 6-10-5 タスク管理もタスク





6-11

プロジェクトと プロジェクト管理を 理解する

6-11-1 プロジェクトとは

プロジェクトとは、**プロジェクトの目的を達成するための業務**です。多くの場合、期間や期限が設定されています。

たとえば、

- ・今までより便利になる新製品を今年中にリリースする
- ・来年までに家を建てる
- ・ランキング10位に入るゲームを開発する

などといった例もプロジェクトです。

6-11-2 プロジェクトとタスクの違い

たとえば、今晚食べるための「カレーを作る」というのはタスクです。ですが、「今年中に日本一のカレーを作る」となると**プロジェクト**です。

プロジェクトはタスクではなく目的であり、達成するためにチームメンバーを編成します。

「カレーを作る」タスクは、具体的にイメージできる作業なので、すぐに取り掛かれるでしょう。「日本一のカレーを作る」プロジェクトは、目的達成のために「スパイスを仕入れる」「最高のカレーを作る」「最高のお米を仕入れる」といった複数のタスクが発生します。

このように、プロジェクトは、複数のタスクで構成されます。

図 6-11-1 プロジェクトは複数のタスクで構成される



6-11-3 プロジェクト管理とは

プロジェクトを成功に導くために、人員や進捗、スケジュール、コスト、品質、解決すべき課題などを管理することを**プロジェクト管理**と言います。複数人のタスク管理そのものを管理することも含まれています。

プロジェクト管理において、タスクを効率的に運営していくために作業内容を「見える化」し、マネジメントしやすくすることが、プロジェクトの成功に役立ちます。

6-11-4 PMBOKとは

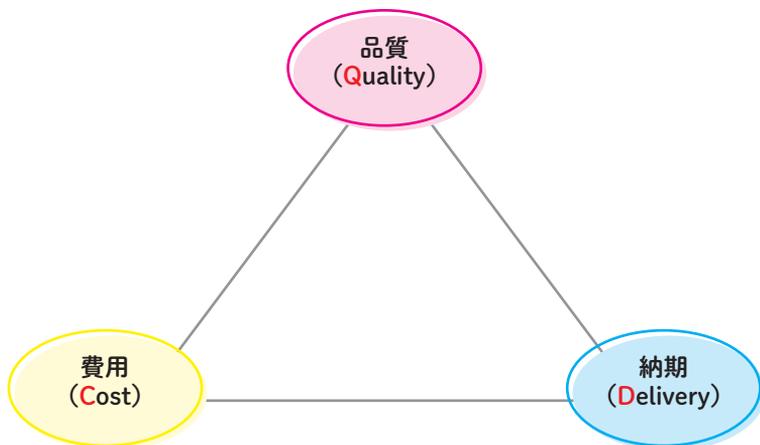
PMBOK (Project Management Body Of Knowledge : ピンボック) は、プロジェクト管理に関する知識を体系的にまとめたものです。「品質」「原価」「スケジュール」など10の知識エリアと「立ち上げ」「計画」「実行」などの5つのプロセスが定義されていて、QCDを適切に管理することもPMBOKの目標となっています(最新の『PMBOK第7版』では大きく内容が変わっています)。

6-11-5 QCDとは

QCDとは、Quality (品質)、Cost (コスト)、Delivery (納期)の頭文字をとったもので、QCDを守ることがプロジェクト管理に求められます。

QCDの3要素は互いに影響し合う関係であり、例えば、品質を上げるためには、費用を上げて、納期を遅らせる必要があるなどです。

図 6-11-2 QCD の関係



QCD の 3 要素に「S」を追加した「QCDS」というものがあり、Scope（スコープ：範囲）を意味します。

IT を導入する場合やシステム開発する場合に、提供する機能の範囲を決めるものです。スコープの調整ができることで、QCD の調整もより柔軟になります。

6-11-6 代表的なプロジェクト管理手法

複数人が関わるプロジェクト管理は個人のタスク管理よりも複雑なため、プロジェクトを成功させるために管理手法は特に重要です。管理手法の中でも、ウォーターフォール型とアジャイル型は IT のシステム開発でもよく用いられるため、この後のセクションで詳しく解説します。

ウォーターフォール型

ウォーターフォール型は、時系列に沿って工程を一つずつ進めていきます。計画通りにプロジェクトを完了させていくのに適していて、要件の追加や変更が少ないもの、大規模なプロジェクトで採用されます。

アジャイル型

アジャイル型は 1 ヶ月以内の期間に分割し、それを繰り返してプロジェクトを進めていきます。小さくリリースし成長させていくもの、要件が変化しやすいもの、スピード重視のプロジェクトに適しています。

ガントチャート

やるべきタスクをツリー構造で縦軸に表示し、各タスクのスケジュールを横軸に表示した表で表現することで、ひと目見ただけで進捗状況を俯瞰的に把握して管理する方法です。日程重視で管理するタスクが多くないプロジェクトであれば、ガントチャートだけでも十分でしょう。

カンバン

カンバンボードに「未着手」「作業中」「作業完了」などのステータス（状態）の仕切りを作り、チームのタスクをボードに配置することで、進捗状況を俯瞰的に把握して管理する方法です。優先順位を重視し一定期間内にチームメンバー全員で完了を目指すプロジェクトに適しています。

クリティカルチェーンプロジェクトマネジメント（CCPM）

CCPMは、プロジェクトの納期をできる範囲で短く設定し、プロジェクト全体でバッファ（余裕）の消費具合を管理する手法です。

アーンドバリューマネジメント（EVM）

EVMは、計画値（PV）や出来高（EV）、実績値（AC）などの要素をすべてコストとして換算し、進捗管理する手法です。



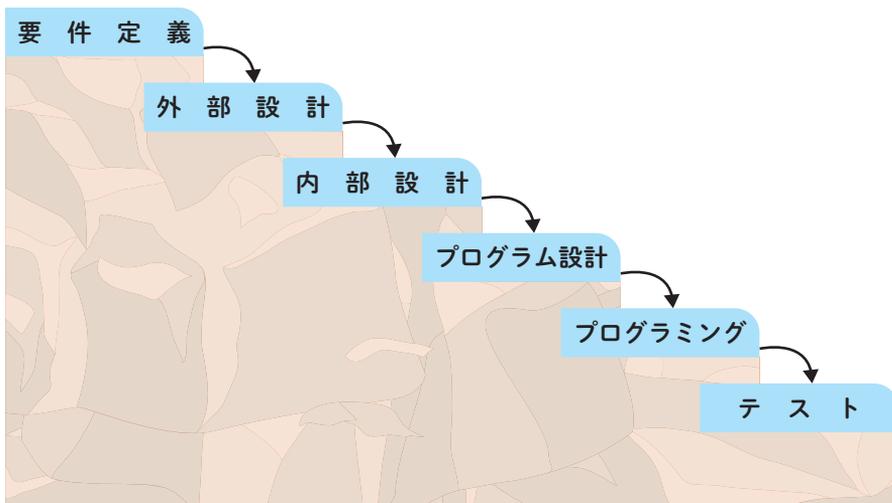
ウォーターフォール型開発の概要を知る

6-12-1 ウォーターフォール型開発とは

ウォーターフォール型開発は、システムやソフトウェア開発において有名なプロジェクト開発手法の一つです。ですが元々、建築や機械製造などの手法を参考にしたもので、主にもものづくりの現場で広く有効です。

システム開発の場合、「要件定義」「外部設計」「内部設計」「プログラム設計」「プログラミング」「テスト」といった工程に分け、一つひとつ順序通りに完了させていきます。

図 6-12-1 ウォーターフォール型開発



各工程が終わってから次に進むという方法で、前の工程には戻りません。それぞれの工程で成果物を作成し、品質を確保していきます。

「ウォーターフォール (waterfall)」とは英語で「滝」という意味です。まさに水が上から下へ落下するかの如く、プロジェクトを進めていく手法です。

6-12-2 ウォーターフォール型開発のメリット

ウォーターフォール型開発のメリットは、「プロジェクト全体の計画を立てやすい」「予算・人員を予測しやすい」「進捗を管理しやすい」といったことが挙げられます。

しっかりと計画を事前に行うことで、プロジェクトの終了までの見通しがわかりやすくなり、進捗管理や時間管理をしやすくなります。やるべきことを機能分割、タスク分割することでWBSを作り、それぞれにプロジェクトメンバーをアサインして、いつからいつまでといったスケジュールを立てることでガントチャートを作成します。プロジェクト開始後も俯瞰的にガントチャートを把握できるため、各工程ごとに進捗管理もしやすくなります。

このように、早い段階でプロジェクト全体を把握できるため、計画通りにプロジェクトを完了させるのに適していて、要件の追加や変更が少ないもの、大規模なプロジェクトで採用されます。

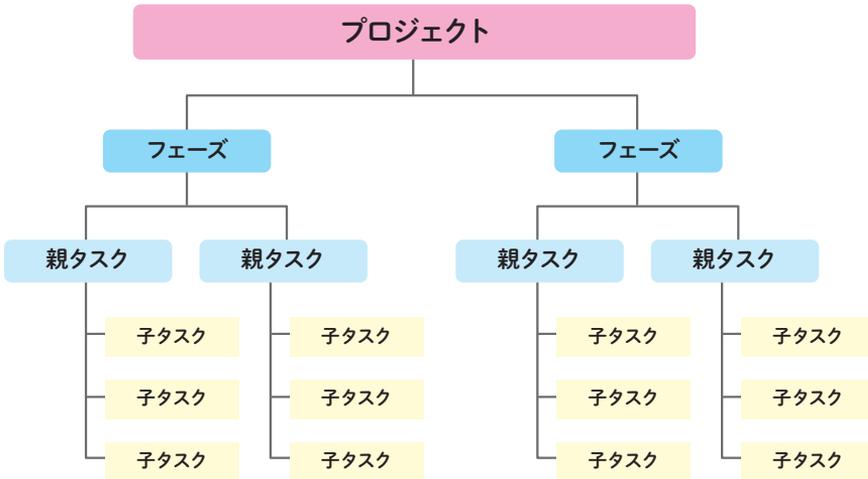


WBSと ガントチャートを理解する

6-13-1 WBSとは

WBS (Work Breakdown Structure) は、作業を細分化して構造図で示す手法です。プロジェクトにおける作業を大きな粒度のタスクから小さなタスクへと分解していきます。大きなタスクから次第にタスクの粒度を小さくしていくというプロセスを辿ることで、より精度の高いタスクの洗い出しができます。

図 6-13-1 WBS (Work Breakdown Structure)

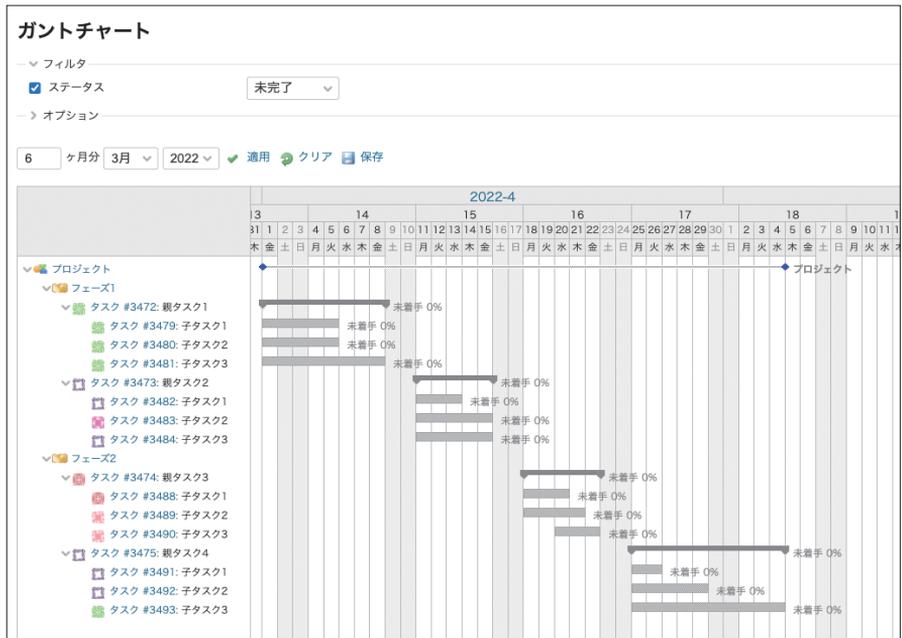


タスクが分解できるとそれぞれのタスクに担当者を割り当てることができ、工数の見積もりも可能となってきます。そうすることで、各タスクのスケジュールリングができるようになります。

6-13-2 ガントチャートとは

ガントチャートとは、WBSの情報を元にプロジェクトの進捗管理をするチャート（図表）のことです。縦軸にタスク、横軸に時間が表され、直感的にプロジェクトの全体を把握できます。全てのタスクや工数の管理、進捗状況、メンバーの状況が見える化されており、効率良くかつ抜けなく作業を進めるのに役立ちます。

図 6-13-2 ガントチャート



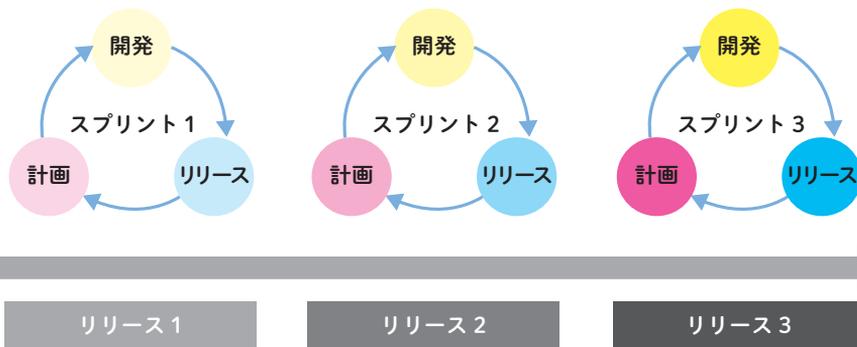


アジャイル開発の概要を知る

6-14-1 アジャイル開発とは

アジャイル開発とは、システムやソフトウェア開発におけるプロジェクト開発手法の一つです。動作する機能単位に開発して、なるべく早く小さいサイクルで繰り返しリリースしていく手法です。

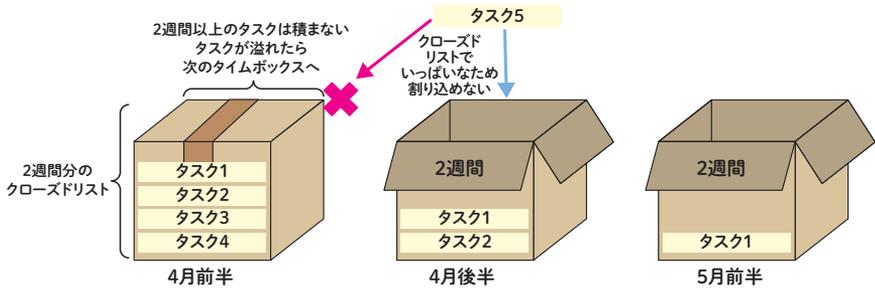
図 6-14-1 アジャイル開発



アジャイル開発の具体的な手法には、**スクラム**、**エクストリーム・プログラミング**などがあります。スクラムでは一連の工程を短期間で繰り返す1～2週間といった開発サイクルの期間の単位を**スプリント**といいます。優先度の高い要件から順番に開発を進めていきます。

ウォーターフォール型開発と違い、プロジェクト全体の計画を綿密に立てるわけではなく、2週間や1ヶ月という期間で時間を区切り、それを繰り返して計画を立てていきます。これを**タイムボックス管理**といいます。

図 6-14-2 タイムボックス管理



6-14-2 アジャイル開発のメリット

アジャイル開発のメリットは、動作する機能ごとに開発・リリースを行うため、より早くユーザーにシステムや新機能を提供できることです。

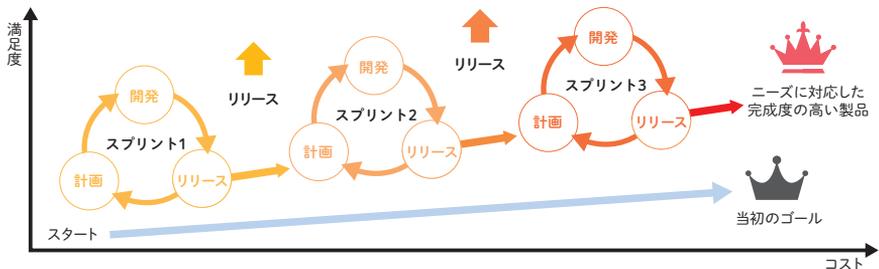
実際にユーザーが使えるようリリースすることで、使い勝手やフィードバックなどを早い段階から確認できるため、仕様変更に対応でき、システムの価値を高めることができます。

アジャイル開発は、プロジェクト全体の進捗管理はわかりにくいですが、スプリントごとに開発にかかった工数の実績がより正確に出るため、次のスプリントの見積り精度が上がります。

同じペースでスプリントごとにリリースできる成果物がこれぐらいの量というのがわかると、次のスプリントでも同じだけの成果物が出てくるという根拠を持てるため、同じリズムで開発できるようになってきます。「半年後にはこれだけできている」という、より現実的な見通しを把握できるようになります。

システム開発以外の分野においても同じリズムで成果を出していきたいプロジェクトには、このようなタイムボックス管理は有効です。具体的な管理方法については、次のセクションで解説していきます。

図 6-14-3 当初より価値の高いものを提供



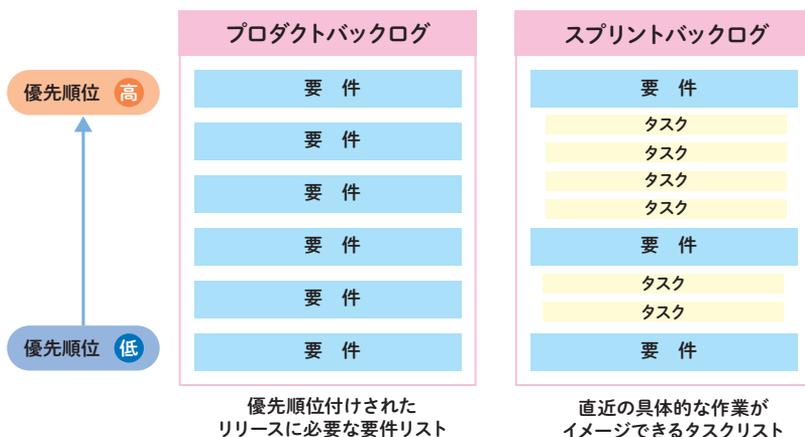


バックログとカンバンを理解する

6-15-1 バックログとは

アジャイル開発手法の「スクラム」では、**プロダクトバックログ**と**スプリントバックログ**という2種類のバックログでプロジェクトを管理していきます。

図 6-15-1 プロダクトバックログとスプリントバックログ



プロダクトバックログは、機能や改善要望などに優先順位をつけてリスト化したものです。優先順位をつけることで、優先順位が高いのに終わっていない状態が続いていると、そこに問題があるなどといった状況に気づきやすくなります。これは、プロダクトについての今後のやることリストだと言えます。オープンリストであり、ユーザーの要望などをいつでも入れることができます。

スプリントバックログは、プロダクトバックログのスプリント期間で行うものを抜き出したもので、具体的にイメージできるレベルにタスク分割したタスクリストのことで、タスクリストを2週間ずつなどのタイムボックスで区切ります。

タスク管理のセクションでも解説したように、このスプリントバックログ、プロダクトバックログの定期的なレビューをすることも重要です。常に最新の状態に保つていくために、スプリントが終わったタイミングでタスクリストに対するレビューを行い、各バックログのリストを見直していきましょう。

6-15-2 カンバンとは

カンバンは、1つのスプリントのタスク状況を俯瞰的に見える化したものです。**カンバンボード**とも呼ばれます。

図 6-15-2 カンバンボード



シンプルに管理するカンバンであれば、トレーシステムのように、タスクを「**未着手**」「**進行中**」「**完了**」という状態のレーンを用意して、そこにタスクリストを優先順位が高い順に並べていきます。一つずつタスクを進めていき、進行中のレーン、完了のレーンへと移動していきます。

カンバンボードで一目で見える化されているので、優先順位の高いものから右にタスクが完了していることが確認できます。

進行中のタスクが多くなると作業がうまく流れなくなってしまうです。進行中のタスクが増えすぎないように制限することで、効率的にタスクを進めることができます。

カンバンボードにすることで未完了のタスクが溜まっている場合に見つけやすくなり、チームワークでスプリントのタスクを終わらせようという意識が高まり、チーム力が高まります。

図 6-16-2 アナログとデジタルツール



タスク管理やプロジェクト管理を行う上で、個人の場合でもうまくいかないことがあります。複数人での管理となると、なおさら複雑でうまくいかないことも多いでしょう。

チームでうまくいかない場合に、1人でやり方を変えようとするにはハードルが高いものです。チームメンバーでふりかえりを行い、KPT（継続すること（Keep）、課題があること（Problem）、新しく取り組むべきこと（Try））について話し合うことで、メンバー内で共通の認識を持ち、やり方を変えようという意識が自然と生まれるようになります。

何より大事ななのは、**タスク・プロジェクト管理をうまくいかせてプロジェクトを成功させること**です。

仕事の負荷を分散させたりプロジェクトの失敗を未然に防いだりするために、有効なタスク・プロジェクト管理方法をチーム内で相談して見出していくことで、よりスムーズにプロジェクトを成功へと導けるようになります。

違う方法を試してみよう、もっとうまくやりたい、そんな気持ちを本書が後押しできれば幸いです。

図 6-16-3 チームのふりかえり



タスク管理についてさらに知りたい方に

タスク管理についてさらに学びたい場合は、以下の文献をご覧ください。

p.15 「オープンリスト」「クローズドリスト」

これらの用語はマーク・フォスター『仕事に追われない仕事術 マニャーナの法則 完全版』（2016/10）で説明されています。

p.24 「ファーストタスク」

ファーストタスクは、マーク・フォスター『仕事に追われない仕事術 マニャーナの法則 完全版』（2016/10）で説明されています。

p.24 「ポモドーロテクニック」

ポモドーロテクニックは、イタリアのフランチェスコ・シリロにより 1987 年に提唱されました。

p.26 「トレーシシステム」

トレーシシステムは、倉下忠憲『「やること地獄」を終わらせるタスク管理「超」入門』（2019/02）で説明されています。

p.26 「インボックス・ゼロ」

インボックス・ゼロは、「43 Folders」の Merlin Mann により 2006 年に提唱されました。

p.27 「GTD」

GTD は、アメリカの経営コンサルタント、デビッド・アレンにより 2002 年に提唱されました。

p.28 「マニャーナの法則」

マニャーナの法則は、マーク・フォスター『仕事に追われない仕事術 マニャーナの法則 完全版』（2016/10）で説明されています。

R Redmine ハンドブック バージョン5.0対応

逆引きで分かる!!

非売品

書籍の内容

CHAPTER 0 Redmine 5.0の新機能

CHAPTER 1 Redmineとは

Redmineとは

CHAPTER 2 チームにおけるタスク管理

チケットの作成

チケット一覧

チケットの更新

親子チケット

関連するチケット

通知

プライベートチケット

作業時間

他システムとの連携

CHAPTER 3 個人設定

個人設定

パーソナライズ

CHAPTER 4 プロジェクト管理

プロジェクト

ウォーターフォール型プロジェクト

アジャイル開発プロジェクト

日々のプロジェクト運用

プロジェクト管理の制御

工数管理

バージョン管理システムとの連携

CHAPTER 5 システム管理

Redmineの導入

Redmineの初期設定

プロジェクト運用に関わる設定

他システムとの連携

CHAPTER 6 タスク・プロジェクト管理とは

タスク管理

プロジェクト管理

ご予約はこちら

