

# 仕事の基礎ガイド

2025-12-30



# 目次

はじめに . . . . .	4
0.1 Part 1: 日常業務編 . . . . .	4
0.1.1 朝イチでやること . . . . .	4
0.1.1.1 朝のルーティン . . . . .	4
0.1.2 タスク管理の基本ルール . . . . .	5
0.1.2.1 4 象限マトリクス . . . . .	5
0.1.2.2 「やらない」の判断基準 . . . . .	5
0.1.3 報告の仕方 . . . . .	5
0.1.3.1 基本構造：PREP 法 . . . . .	5
0.1.3.2 良い例 vs 悪い例 . . . . .	6
0.1.3.3 場面別テンプレート . . . . .	6
0.1.3.4 30 分ルール . . . . .	6
0.1.4 時間を区切って作業する . . . . .	6
0.1.4.1 タイムボックス . . . . .	6
0.1.4.2 タイムブロック . . . . .	7
0.2 Part 2: プロジェクト遂行編 . . . . .	8
0.2.1 プロジェクト参加時の心構え . . . . .	8
0.2.1.1 最初の 1 週間でやること . . . . .	8
0.2.2 タスク遂行のフロー . . . . .	8
0.2.2.1 ステップ 1: タスクの理解 . . . . .	8
0.2.2.2 ステップ 2: 計画 . . . . .	8
0.2.2.3 ステップ 3: 実行 . . . . .	9
0.2.2.4 ステップ 4: 報告・共有 . . . . .	9
0.2.3 会議の参加の仕方 . . . . .	9
0.2.3.1 参加者として . . . . .	9
0.2.3.2 ファシリテーターとして . . . . .	9
0.2.4 成果物作成のポイント . . . . .	10
0.2.4.1 ドキュメント作成 . . . . .	10
0.2.4.2 成果物レビュー . . . . .	10
0.2.5 小さなチームやタスクのリーダーになったら . . . . .	11
0.2.5.1 リーダーになった時の心構え . . . . .	11
0.2.5.2 メンバーへの指示の出し方 . . . . .	11
0.2.5.3 進捗管理の基本 . . . . .	11
0.2.5.4 問題が起きた時の対応 . . . . .	12
0.2.5.5 リーダー代理や小さなタスクの主導者の場合 . . . . .	12
0.3 Part 3: 困ったときの対処法 . . . . .	13

0.3.1	よくある困りごと	13
0.3.1.1	「何をすればいいかわからない」	13
0.3.1.2	「タスクが終わらない」	13
0.3.1.3	「完璧主義で時間がかかる」	13
0.3.1.4	「質問するのが怖い」	13
0.3.2	上司との 1on1 の活用	14
0.3.2.1	1on1 で話すこと	14
0.3.2.2	効果的な 1on1 の準備	14
0.4	付録: チェックリスト	15
0.4.1	日次チェックリスト	15
0.4.1.1	朝(10～15 分)	15
0.4.1.2	夕方(10～15 分)	15
0.4.2	週次チェックリスト	15
0.4.2.1	週初(月曜、15 分)	15
0.4.2.2	週末(金曜、15 分)	15
0.4.3	月次チェックリスト	15
0.4.3.1	月初	15
0.4.3.2	月末	15
0.4.4	プロジェクト開始時チェックリスト	16
0.4.4.1	初日	16
0.4.4.2	1 週間以内	16

## はじめに

このガイドの狙いは、指示待ちではなく自分で考えて動けるようになること。報告や相談の仕方、タスク管理、時間管理など、仕事の基本を押さえておけば日々の業務が楽になります。

## 0.1 Part 1: 日常業務編

### 0.1.1 朝イチでやること

毎朝、仕事を始める前に以下を確認してください。10～15 分かかりますが、この時間が一日の生産性を変えます。

#### 0.1.1.1 朝のルーティン

1. **メールとチャットの確認**
  - 緊急対応が必要なものがないかチェック
  - 必要なら朝会前に簡単に返信
2. **今日のタスクリスト作成**
  - 昨日の残タスク + 今日の予定タスク
  - 優先順位をつける(後述の 4 象限マトリクス参照)
3. **カレンダー確認**
  - 会議時間を把握
  - 作業時間の確保を確認

朝イチのルーティンは習慣化が重要です。毎日同じ順序で行うことで、考えなくても自然にできるようになります。慣

れてくれば 10 分以内で終わるようになります。

### 0.1.2 タスク管理の基本ルール

タスクが多すぎて何から手をつければいいのか分からない時は、4 象限マトリクスで整理します。

#### 0.1.2.1 4 象限マトリクス

タスクを「緊急度」と「重要度」の 2 軸で分類：

	緊急	緊急でない
重要	<b>第 1 象限:</b> 今すぐやる (例：納期直前の仕事、クリティカルなバグ修正)	<b>第 2 象限:</b> 計画的にやる (例：スキルアップ、改善活動、ドキュメント整備)
重要でない	<b>第 3 象限:</b> 可能な限り断る/委譲 (例：急ぎだが自分でなくてもいい依頼)	<b>第 4 象限:</b> やらない (例：意味のない作業、惰性で続けていること)

#### 第 1 象限と第 2 象限の見分け方

どちらも重要なタスクですが、違いは「期限の切迫度」です：

- **第 1 象限：** 期限が迫っている、または今対応しないと大きな問題になる
  - － 例：明日が締切のレポート、顧客からのクレーム対応
- **第 2 象限：** 重要だが今すぐでなくても大丈夫
  - － 例：来月のプレゼン準備、業務の効率化

迷ったら上司に「このタスク、いつまでにやればいいですか？」と確認してください。実は第 1 象限だと思っていたものが第 2 象限だった、ということもよくあります。

#### 0.1.2.2 「やらない」の判断基準

以下に当てはまるタスクは、思い切って「やらない」選択肢を検討してください：

- 誰も見ない資料作成
- 属人化された非効率な作業 (改善できないか相談)
- 完璧を目指しすぎた作業 (60% で十分なことも多い)

迷ったら上司に相談してください。「このタスク、優先度低いと思うのですが、やらなくても大丈夫ですか？」と聞くだけで OK です。

### 0.1.3 報告の仕方

報告は結論から話すのが基本です。経緯から話すと、聞き手は「で、結局どうなったの？」となります。

結論を先に知ること、聞き手は心構えを持って話を聞けます。たとえ悪い報告でも、最初に結論を知った上で経緯を聞く方が、適切に対応できます。

#### 0.1.3.1 基本構造：PREP 法

##### 1. Point (結論)

2. Reason (理由)
3. Example (具体例)
4. Point (再度結論)

#### 0.1.3.2 良い例 vs 悪い例

悪い例：＞「昨日から〇〇の調査をしまして、まず A を試したんですけど、それがうまくいなくて、次に B を試したら... (以下、経緯が続く)」

良い例：＞「〇〇の調査、完了しました。結論としては B の方法が最適です。理由は...」

#### 0.1.3.3 場面別テンプレート

##### ■0.1.3.3.1 進捗報告

【結論】タスク A は予定通り完了、タスク B は遅延しています

【詳細】

- － タスク A：完了(予定：金曜、実績：金曜)
- － タスク B：50% 完了(予定：金曜、見込み：来週火曜)

【理由】依存する〇〇の仕様が未確定のため

【対応】月曜に〇〇の担当者と打ち合わせ予定

##### ■0.1.3.3.2 相談

【相談内容】〇〇の実装方法について相談させてください

【背景】△△の機能を実装中です

【選択肢】

1. 方法 A：メリット○、デメリット
2. 方法 B：メリット○、デメリット

【私の考え】方法 B がよいと思います(理由：…)

【確認したいこと】この方針で進めてよいでしょうか

#### 0.1.3.4 30 分ルール

30 分詰まったら相談する。一人で考え込んで時間を浪費するより、早めに相談する方が効率的です。「30 分考えても分からなかったので相談させてください」と言えば、誰も怒りません。

#### 0.1.4 時間を区切って作業する

集中力を維持するには、時間を区切って作業することが重要です。

##### 0.1.4.1 タイムボックス

一つのタスクに対して、事前に時間を決めて取り組む方法です。

- ・「このレビューは 30 分で終わらせる」

- 「この調査は 1 時間で一旦区切る」

時間が来たら、途中でも一旦止めて状況を確認します。完璧を目指さず、60% の完成度で見せることを意識しましょう。

#### 0.1.4.2 タイムブロック

カレンダーに「作業時間」を確保する方法です。

- 午前中の 2 時間を「集中作業タイム」としてブロック
- 会議と会議の間の 1 時間を「タスク処理」として確保

会議が多い人ほど、意識的に「作業時間」を確保する必要があります。カレンダーに予定を入れて、他の人から会議を入れられないようにしましょう。

## 0.2 Part 2: プロジェクト遂行編

### 0.2.1 プロジェクト参加時の心構え

新しいプロジェクトに参加した時、最初の 1 週間でやるべきことがあります。

#### 0.2.1.1 最初の 1 週間でやること

##### 1. プロジェクトの全体像を把握

- プロジェクトの目的は何か
- 期限はいつか
- 誰が何を担当しているか

##### 2. 分からないことリストを作る

- 用語、システム構成、ツール、etc.
- 分かったものからチェックを入れる

##### 3. 自分の役割を明確にする

- 何を期待されているか
- 誰に報告・相談すればいいか
- 判断に迷った時は誰に聞けばいいか

初日に聞くべき 3 つの質問：

1. 「このプロジェクトのゴールは何ですか？」
2. 「私の役割は何ですか？」
3. 「困った時は誰に相談すればいいですか？」

### 0.2.2 タスク遂行のフロー

タスクを受け取ってから完了するまでの基本的な流れです。

#### 0.2.2.1 ステップ 1: タスクの理解

- 何をすればいいのか(What)
- なぜそれが必要なのか(Why)
- いつまでに必要なのか(When)
- 誰のためのものか(Who)

分からなければ必ず確認してください。想像で進めるのは危険です。

#### 0.2.2.2 ステップ 2: 計画

- どうやって進めるか(How)
- どれくらい時間がかかりそうか
- 何が分からないか、何をリスクと考えるか

計画を上司と共有することをおすすめします。方向性のズレを早期に防げます。



### 0.2.2.3 ステップ 3: 実行

- タイムボックスを使って区切りながら進める
- 30 分詰まったら相談
- 60% で一度見せる

### 0.2.2.4 ステップ 4: 報告・共有

- 完了したら必ず報告
- 途中でも定期的に進捗共有
- 問題があれば早めに相談

60% で見せるメリット：

- 早めにフィードバックがもらえる
- 方向性のズレを修正できる
- 完璧を目指して時間を浪費しない
- 「報連相ができる人」と評価される

## 0.2.3 会議の参加の仕方

### 0.2.3.1 参加者として

会議に参加する時の心構え：

#### 1. 事前準備

- アジェンダを確認
- 分からない用語は調べておく
- 自分の意見を考えておく

#### 2. 会議中

- メモを取る（後で議事録を書く）
  - 議事録作成ツールを使ってもよいが、重要な決定事項やアクションアイテムはその場で手元にメモした方がよい
  - 後で録音や自動議事録と突き合わせて確認できる
- 分からないことは質問する
- 自分の意見を言う（遠慮しない）

#### 3. 会議後

- 議事録を書く（または確認）
- アクションアイテムを自分のタスクリストに追加

### 0.2.3.2 ファシリテーターとして

会議を進行する時のポイント：

#### 1. 開始時

- アジェンダを共有
- 終了時刻を明確に

#### 2. 進行中

- 時間を管理

- 全員が発言できるように配慮
- 脱線したら軌道修正

### 3. 終了時

- 決定事項を確認
- アクションアイテムと担当者を明確に
- 次回の予定を決める

## 0.2.4 成果物作成のポイント

### 0.2.4.1 ドキュメント作成

#### 1. 読み手を意識する

- 誰が読むのか
- 何を知りたいのか
- どれくらいの前提知識があるか

#### 2. 構造を意識する

- 見出しで構造を作る
- 箇条書きを活用
- 長い文章は避ける
- 図解を使うのも効果的(ホワイトボードツール、手書きの絵、フローチャート、ロジックツリーなど)

#### 3. 完璧を目指さない

- 60% で一度見せる
- フィードバックをもらって改善

### 0.2.4.2 成果物レビュー

コード、ドキュメント、資料、設計書など、何を作る場合でも、レビューしてもらうことで品質が上がります。

#### 1. 小さく分けてレビュー依頼

- 一度に大量の成果物を出さない
- コードなら：機能ごと、ファイルごと
- ドキュメントなら：章ごと、セクションごと
- プレゼン資料なら：前半・後半に分けるなど

#### 2. レビュー依頼時に伝えること

- 何を作ったか(目的、対象読者・利用者)
- どこを重点的に見てほしいか
- 自分で気になっている点

#### 3. フィードバックへの対応

- 防衛的にならない
- 分からないことは質問
- 学びの機会と捉える

60% の完成度で一度見せることで、方向性のズレを早期に修正できます。完璧にしてから見せるより、早めに見せて改善の方が効率的です。

### 0.2.5 小さなチームやタスクのリーダーになったら

入社数年で、小さなプロジェクトのリーダーや作業の主導者を任されることがあります。「自分がまとめる側になる」というのは不安かもしれませんが、基本を押さえれば大丈夫です。

#### 0.2.5.1 リーダーになった時の心構え

##### 1. 完璧なリーダーを目指さない

- 分からないことは素直に聞く
- メンバーの方が詳しいこともある(それで OK)
- 上司やベテランに相談しながら進める

##### 2. 自分一人で抱え込まない

- リーダーの仕事は「全部自分でやる」ことではない
- メンバーに任せる、上司に相談する、これも重要な仕事

##### 3. 決めることと、相談することを分ける

- 自分で決めていいこと、上司に確認すべきことを明確に
- 迷ったら「これは自分で決めていいですか？」と聞く

#### 0.2.5.2 メンバーへの指示の出し方

指示を出す時は、以下を明確に伝えます：

##### 1. 何をやってほしいか(What)

- 具体的なタスク内容
- 完了の基準(どうなったら完了か)

##### 2. なぜやるのか(Why)

- 背景や目的を伝える
- 目的が分かれば、メンバーは自分で考えて動ける

##### 3. いつまでか(When)

- 期限を明確に
- 途中報告のタイミングも伝える

##### 4. 困った時はどうするか

- 相談先を明確に(自分に聞く、別の人に聞く、など)
- 30 分ルールを共有(詰まったら早めに相談)

悪い指示の例：> 「〇〇の調査、お願いします」

良い指示の例：> 「〇〇の調査をお願いします。△△を導入すべきか判断するための情報が欲しいので、メリット・デメリットを整理してください。金曜までに結果を教えてください。途中で分からないことがあれば、いつでも相談してください」

#### 0.2.5.3 進捗管理の基本

メンバーの進捗を把握する方法：

##### 1. 定期的な確認

- 週 1 回程度の短いミーティング(15 分～30 分)
- 「今週やったこと」「来週やること」「困っていること」を聞く

## 2. 困っている様子を見逃さない

- 質問が来ない = 順調とは限らない
- 「何か困っていることない？」と声をかける

## 3. 遅延の兆候を早めにキャッチ

- 「予定通り進んでいますか？」と確認
- 遅れそうなら早めに対策を考える(スコープ調整、人の追加、etc.)

進捗確認は「監視」ではありません。メンバーが困っている時に早めに助けるためのものです。

### 0.2.5.4 問題が起きた時の対応

トラブルが起きた時のリーダーの役割：

#### 1. 事実を把握する

- 何が起きたのか
- なぜ起きたのか
- 影響範囲はどこまでか

#### 2. すぐに上司に報告・相談

- 自分で解決しようとして遅れるより、早めに報告
- 「〇〇の問題が起きました。△△が原因と思われます。□□の対応を考えていますが、相談させてください」

#### 3. 責めない

- メンバーのミスを責めても解決しない
- 「どうすれば防げたか」「次にどうするか」を考える

#### 4. チームで対応する

- 一人に押し付けない
- 必要なら役割分担を見直す

### 0.2.5.5 リーダー代理や小さなタスクの主導者の場合

正式なリーダーではないが、実質的に取りまとめ役になることもあります：

#### • 自分の役割を確認する

- どこまで自分で決めていいのか
- 誰に報告・相談すればいいのか
- メンバーは自分の指示を聞く立場なのか

#### • 困ったら正式なリーダーや上司に相談

- 「〇〇について、自分はこう考えていますが、これで進めてよいですか？」
- 代理の立場なら、遠慮せず上にエスカレーションする

## 0.3 Part 3: 困ったときの対処法

### 0.3.1 よくある困りごと

#### 0.3.1.1 「何をすればいいかわからない」

対処法：

1. まず朝のルーティンを実行
2. タスクリストを作成
3. それでも分からなければ上司に相談

相談の仕方：

「今日やるべきことを整理したいのですが、〇〇と△△、どちらを優先すべきでしょうか？」

#### 0.3.1.2 「タスクが終わらない」

対処法：

1. 4 象限マトリクスで整理
2. 「やらないタスク」を決める
3. 上司に優先順位を相談

相談の仕方：

「現在、A、B、C のタスクを抱えています。すべて期限内に終わらせるのが難しそうです。優先順位をつけたいのですが、相談させてください」

#### 0.3.1.3 「完璧主義で時間がかかる」

対処法：

1. タイムボックスを設定
2. 60% で見せることを意識
3. 「完了の定義」を上司と確認

考え方の転換： - 完璧な成果物を遅く出すより、60% の成果物を早く出す方が価値が高い - フィードバックをもらって改善する方が、一人で完璧を目指すより効率的

#### 0.3.1.4 「質問するのが怖い」

対処法：

1. 30 分ルールを守る
2. 質問は具体的にする
3. 「調べたけど分からなかった」と伝える

良い質問の例：

「〇〇について、公式ドキュメントと△△を確認しましたが、□□の部分が分かりませんでした。教えていただけますか？」

### 0.3.2 上司との 1on1 の活用

定期的な 1on1 は、成長のための重要な機会です。

#### 0.3.2.1 1on1 で話すこと

##### 1. 進捗報告

- 今週やったこと
- 来週やること
- 困っていること

##### 2. 相談

- キャリアの方向性
- スキルアップの方法
- チーム内の課題

##### 3. フィードバック

- 自分の仕事ぶりへの評価
- 改善点
- 期待されていること

#### 0.3.2.2 効果的な 1on1 の準備

事前に準備すること：

- 話したいトピックのリスト
- 質問事項
- 自分の考えや意見

1on1 後にやること：

- 議事録を書く
- アクションアイテムを実行
- 次回までの目標を設定

## 0.4 付録: チェックリスト

### 0.4.1 日次チェックリスト

#### 0.4.1.1 朝(10～15 分)

- ☐ メールとチャットを確認
- ☐ 今日のタスクリストを作成
- ☐ カレンダーで会議時間を確認

#### 0.4.1.2 夕方(10～15 分)

- ☐ 今日完了したタスクをチェック
- ☐ 明日のタスクを整理
- ☐ 報告すべきことを報告
- ☐ 翌日の準備(資料の確認など)

### 0.4.2 週次チェックリスト

#### 0.4.2.1 週初(月曜、15 分)

- ☐ 今週の目標を設定
- ☐ 週次会議の予定を確認
- ☐ 重要なタスクをカレンダーにブロック

#### 0.4.2.2 週末(金曜、15 分)

- ☐ 今週の振り返り(何ができた？何ができなかった？)
- ☐ 来週のタスクを整理
- ☐ 週次報告を作成

### 0.4.3 月次チェックリスト

#### 0.4.3.1 月初

- ☐ 今月の目標を設定
- ☐ 大きなタスクを週に分割
- ☐ スキルアップ計画を立てる

#### 0.4.3.2 月末

- ☐ 今月の振り返り
- ☐ 目標の達成度を確認
- ☐ 来月の計画を立てる
- ☐ 1on1 で上司と共有

## 0.4.4 プロジェクト開始時チェックリスト

### 0.4.4.1 初日

- ☐ プロジェクトの目的を理解
- ☐ 自分の役割を明確化
- ☐ チームメンバーを把握
- ☐ 相談先を確認

### 0.4.4.2 1週間以内

- ☐ 開発環境のセットアップ
- ☐ コードベースの把握
- ☐ 用語・ツールの理解
- ☐ 最初のタスクを完了

---

このガイドは継続的に改善されています。フィードバックや改善案があれば、[GitHub の Issues](#) でお知らせください。