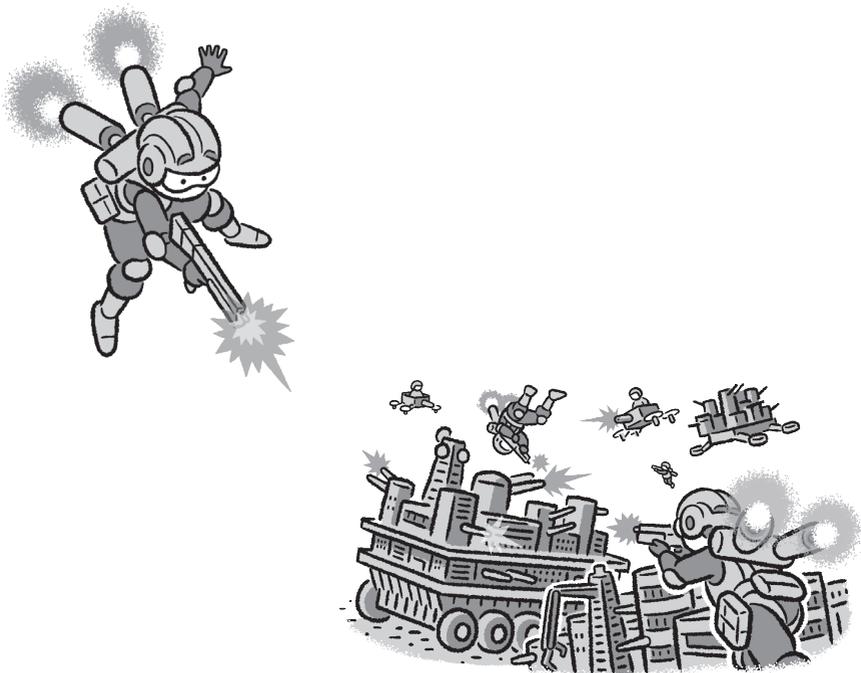


contents

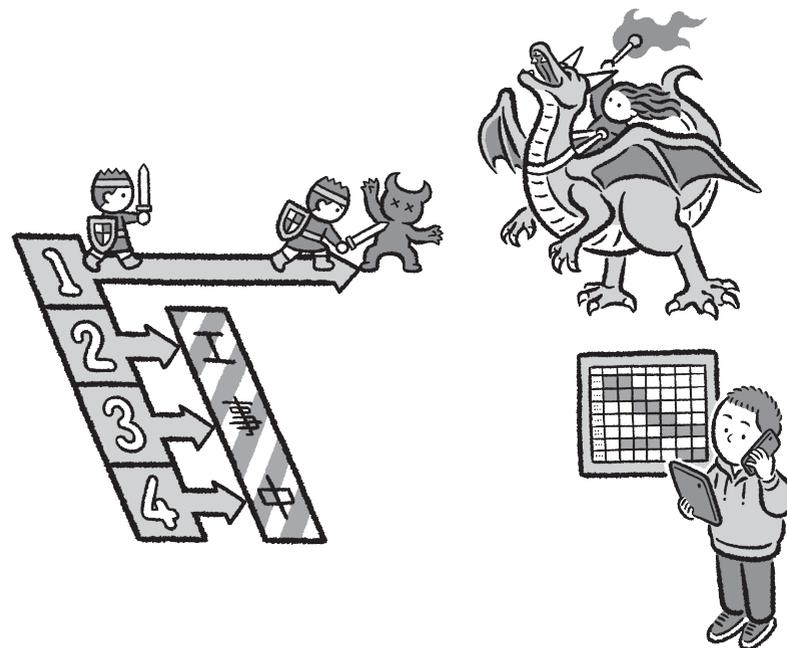
まえがき	9
イントロダクション	10
本書の方針	11

第1部 ゲーム開発プロジェクト管理の概要

▶ I ゲーム開発の概要	15
はじめに	16
製品リリースまでの流れ	17
運営	34



▶ II ゲーム開発プロジェクトの概要	37
定常業務とプロジェクト	38
プロジェクトの構成	41
ゲーム開発プロジェクトの特徴	50
▶ III ゲーム開発プロジェクト管理の例	53
プロジェクト管理の重要性	54
ゲーム開発プロジェクトの計画管理例	55
ゲーム開発プロジェクトの開発管理例	64
ゲーム開発プロジェクトのリリースと運営管理例	67



第2部 計画の管理

▶ IV 計画の方針	73
開発計画を管理する	74
▶ V プロジェクトの計画	81
プロジェクトの要素	82
プロジェクトのための組織(チーム)	86
▶ VI プロジェクトの仕様設計	93
仕様の設計	94
仕様書へのフィードバック	101



第3部 開発の管理

▶ VII タスク管理の進め方	105
タスクでプロジェクトを管理する	106
▶ VIII マイルストーン管理の進め方	121
マイルストーンを活用する	122
仕様の見直し	132



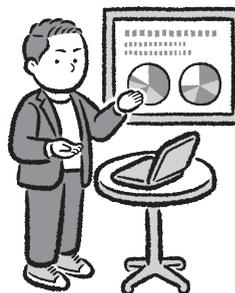
第4部 リリース・運営

IX リリースまでの進め方	137
検証作業	138
マスターアップとリリース	147
X ゲーム運営プロジェクト管理	151
開発後もプロジェクトが続くケース	152
ゲーム運営プロジェクトの分析	157
XI 開発管理スタイル	169
主流のスタイル	170
ゲーム開発現場の独自開発スタイル	183



第5部 まとめと演習

XII プロジェクト管理の演習	185
基本演習	186
Appendix 管理ツールの利用	195
ツールを利用した開発管理	196
あとがき	204



イントロダクション

プロジェクト管理と言われたら、あなたはどんな印象を持ちますか？

人を支配して言うことを聞かせる悪の手口を想像するかもしれません。あるいは人を機械のように働かせる非人間的な体制でしょうか。

本書で語るプロジェクト管理は、人々が生き生きと活躍するために役立つ方法です。

コンピューターを使ったゲームは半世紀ほど前に生まれ、ハードウェアとプログラム技術を猛スピードで発展させてきました。かつては白黒の画面上を小さな四角や文字が動いていただけだったのに、今では実写と見まがうような映像で世界が描かれ、世界中のプレイヤーたちとオンラインでつながって驚異的な冒険を共にすることもできます。Unity や Unreal Engine などの商用ゲームエンジンを使うことで、ゲームを開発するハードルもぐっと下がりました。ゲームを誰でも作ることができると言われる時代の到来です。

ところが昔も今も変わらずゲーム開発はあちこちで失敗しています。開発が行き詰ってゲームのリリースが中止というニュースが流れることもあれば、大ヒット作を手掛けてきたチームの新作が延期を繰り返してようやくリリースされるも未完成の部分だらけといったことも見受けられます。

どうしてゲームの開発は失敗するのでしょうか？

黎明期のゲームはプログラマー1人によって作られ、作業の期間もせいぜい数か月でした。今ではプログラマーにプランナーにデザイナー、コンポーザーにディレクターにプロデューサー、数十人から数百人が年単位の期間をかけて作るようになっていきます。そして今のゲームは極めて複雑化しており、そのボリュームも数キロバイトから数ギガバイト以上と軽く百万倍以上に拡大しています。

こうした要因がゲームの開発を難しくしていることは間違いありません。しかし根本的な原因ではないのです。

ゲームの規模がごく小さかった初期は、ぱっと思いついたことを自分でプログラムしてみて、運よくナイスなアイデアだったら成功でした。失敗しても自分が残念なだけです。

ところが数十、数百人が関わるようになった現代のゲーム開発では、チームのメンバーが思い付きでばらばらに作ってもゲームにはまとまらず、失敗すれば大勢が困ることになります。昔も今も同様に思い付きが失敗を呼んでいます。今はそれが分かりやすく大きな問題になってしまうのです。

ここで疑問に思う方もいるのではないのでしょうか。ゲームは思い付き、つまりアイデアが大事なのではないかと。

アイデアは確かにゲームの中核です。しかし思いついたアイデアをただ作るのでは失敗してしまう可能性が大了。なぜならアイデアとは新しい提案であり、作ってみなければ分からないからです。

ゲーム開発のプロジェクト管理とは、この失敗の可能性を抑えて成功を目指すための方法です。多数のチームメンバーがまとまった方向に気持ちを合わせつつ、それぞれの個性と能力を発揮する。そんな環境を作るための方法でもあります。

さあ、ページをめくって、プロジェクト管理を使いこなすための旅に出発しましょう。機械的な支配ではなく、ばらばらの放任でもなく、生き生きとしたチームによる最高のゲームでプレイヤーを大喜びさせる未来のために。

本書の方針

本書が対象とするゲームは、コンピューターを使って動作するいわゆるデジタルゲームです。家庭用ゲーム機やスマートフォン、PCなどのプラットフォームは問いません。

またゲーム開発のプロジェクトには、新規にゲームを作る初期開発プロジェクトと、すでにリリースしたゲームをバージョンアップしながらサービスを継続していく運営プロジェクトがあります。本書では初期開発プロジェクトを中心に解説します。

読者にはプロジェクトを管理する立場のプロデューサーやディレクター、プロジェクトマネージャー、また作業割り振りに関わるパートリーダーやプランナーを主に想定しています。現場で作業されている方々にとっても、プロジェクト管理を知ることによってプロジェクトの流れを理解して働きやすくなったり、自分の業務をひとつのプロジェクトとして管理していけるようになったりするでしょう。

本書で使用する用語については、ゲーム会社でなるべく広く使われている用語を用います。プロジェクト管理では同じ意味でも様々な用語が使われているため、会社によっては用語が異なるかもしれません。どちらが正しいということではありませんので読み替えをお願いします。

本書で解説するプロジェクト管理については、基本的には標準的な方法に基づきますが、現場で行われているゲーム開発に応じて絞り込んだ内容とし、現場で利用されない煩雑な概念は避けます。

なお、本書は情報処理技術者のプロジェクトマネージャ試験に向けた内容ではなく、ゲーム開発の現場における実務的なプロジェクト管理の基礎を学んでもらうためのものです。試験対策にはそのための専門書をお求めください。

▶ 本書で得られること

- ・ゲーム開発プロジェクトの構造を理解する。
- ・ゲームをはじめとする目的があいまいなプロジェクトの管理方法を理解する。
- ・ゲームの企画立ち上げにおいて重要な要素の決定方法を学ぶ。
- ・ゲームの初期開発プロジェクトおよび運営プロジェクトについて特徴を学び、管理に役立てられるようになる。

▶ 管理って何だろう？

本書では「管理」と「マネジメント」を同じ意味として扱っています。

日本語の「管理」には管理者が部下を組織の型にはめて機械的に働かせるといった抑圧的なイメージがあります。組織が主、部下が従です。

一方、「マネジメント」については一人一人を大事にする開放的なイメージがあります。これは経営学の大家であるピーター・F・ドラッカーによって書籍「マネジメント」で提唱された考え方の影響が大きいようです。

『組織とは個としての一人一人の人間に対して、何らかの貢献を行わせ、自己実現させるための手段である』と彼は述べています。つまりマネジメントにおいて組織とは人のためにあるのです。

ゲーム開発のように、チームメンバーの創造が積み重なって作品を作り上げるようなプロジェクトでは、メンバー一人一人の自己実現がゲームの成功につながります。

今までの「管理」が「マネジメント」に変わり、自己実現によるプロジェクト成功が増えていくことを祈って、本書では「管理」を「マネジメント」同様のポジティブなイメージで語っていきます。

改めて、管理とは権力者による支配体制のことではありません。

社会や組織の中で一人一人がそれぞれの能力を発揮して活躍できるようにするのが管理です。そのためには皆がやりがいを持って主体的に目的達成を目指していける状況を作らねばなりません。こうした状況は人の能力を引き出して成長させます。

参加者が活躍し、プロジェクトが目的を達成し、成果物が社会に貢献し、関わった参加者や組織が成長して己を実現していく。これこそが管理の意義です。

そうは言っても結局プロジェクトは命令されたことをやらされるだけと思うかもしれませんが。そうならないためにプロジェクト管理の方法があります。

参加者が自己実現していくには自発的に考えて動く状況が必要です。しかし参加者がばらばらに好きなことをやっているとプロジェクトは進まなくなってしまう。

そこでプロジェクトの目的を設定して参加者間で共有し、それを実現するためのコンセプトを全体的な行動指針とすることで参加者の方向性をまとめます。そして目的達成のための行動内容は参加者に任せていくのです。

人は命令されたらその命令に応じた成果しか上げなくなります。人は権限を委譲されて自由を得ることで、自発的に能力を発揮して、成長して、指示を超えた成果が上げられるようになります。

目的やコンセプトが決まっているのに自由があるのだろうかという疑問に対しては、我々の創ろうとしているゲームこそがその答えです。

ルールが決まっているからこそ自由な可能性を追求して、成長して、プレイヤーとしての自己を実現していくのがゲームです。仕事を1つのゲームに変えるのがプロジェクト管理とも言えるでしょう。

このゲームをクリアするのは決して簡単なことではありません。プロジェクト管理は安易なチートではないのです。むしろプロジェクトのトップが責任を下に押し付けたり何も考えずに思い付きだけで命令したりといった手抜きプレイを防ぐための手段です。

ゲーム開発プロジェクトは人の才能を結集することで成果物を作り上げます。プロジェクトを成功させるための攻略法はなによりも人に注目することです。そこに手間暇を惜しまないようにしましょう。

プロジェクトのリーダーには大きな権力があるものです。その力は面倒を部下に押し付けることに使うのではなく、皆を自由に活躍させるために駆使してください。それがプロジェクト管理です。

▶ 筆者の紹介

筆者は株式会社セガなどのゲーム会社にて、プロデューサー、ディレクター、プランナーとしてアーケードゲーム、家庭用ゲーム、スマートフォンゲーム、PCゲームの企画・研究・開発を担当してきました。これらで得た経験と知識をベースに、ゲーム開発におけるプロジェクト管理を解説します。

はじめに

ゲーム開発のプロジェクト管理を学び始めるにあたって、まずゲーム開発がどのように行われているのかを見てみましょう。ただしこの章で扱うのはあくまで概要ですので、詳しくは次章以降で解説していきます。

組織が作業を行うときには大きく分けて2つのパターンがあります。ひとつは決まったことを決まった人たちが決まった流れで繰り返して決まった成果を得る「定常業務」、もうひとつは臨時で集まったチームが新たな目的に向けて取り組んで新たな成果を得る「プロジェクト」です。

このうちゲームの開発は主にプロジェクト形式で行われます。

プロジェクトと定常業務

	定常業務	プロジェクト
目的	前から決まっている目的	新しい目的
作業内容	決まったことを繰り返す	新しい作業を始める
作業者	前から決まったとおりの人	臨時で集まった人

ゲーム開発プロジェクトの流れ

ゲーム開発のプロジェクトは様々な段階を経ながら進んでいきます。どのような段階があってそれぞれ何が求められているのかを知ること、プロジェクト管理を理解しやすくなります。詳しくは次の節から見ていきましょう。

ゲーム開発プロジェクトの大きな流れ

計画

- ① プロジェクト立ち上げ
- ② 企画立案
- ③ 開発計画
- ④ 収支計画
- ⑤ 企画承認
- ⑥ チーム編成

開発

- ⑦ プロト版
- ⑧ α版
- ⑨ β版
- ⑩ マスター

製品リリース

- ⑪ 内部承認
- ⑫ 外部承認
- ⑬ リリースと不具合対応

運営

- ⑭ 運営と開発
- ⑮ 運営継続判断

製品リリースまでの流れ

ここではゲーム開発におけるプロジェクトの立ち上げから製品リリース（発売やサービス開始）までを扱います。プロジェクトの段階は計画、開発、製品リリースと進んでいき、オンラインゲームの場合は製品リリース後に運営が行われます。

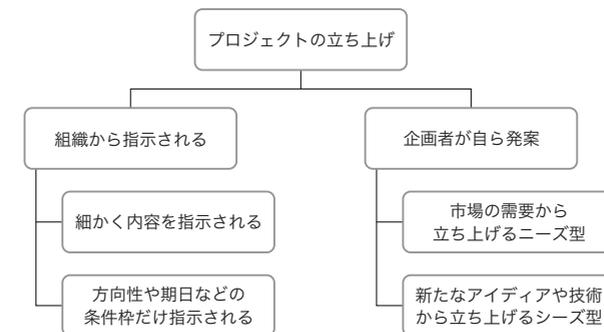
計画

まずはプロジェクトそのものの立ち上げから解説していきます。

① プロジェクト立ち上げ

プロジェクトの立ち上げには様々なパターンがありますが、市場の需要を見て企画する「ニーズ型」と企画者のアイデアや新技術から立ち上げる「シーズ型」に大きく分けることができます。

そこからのさらなる区分として「組織から命じられて企画するパターン」と「自ら企画するパターン」があります。なお、命じられて企画するパターンの中にも、プロジェクトの内容を厳密に指定される場合と、市場や期日など大枠の条件（条件枠）を指定されてその枠内での企画を求められる場合があります。



一般的に企画としてイメージされるのは企画者個人がアイデアから立ち上げるパターンではないかと思いますが、そうした企画を通すのはかなりの難問です。組織が求める様々な条件をクリアせねばならないからです。実際には命じられた枠内でシリーズ続編を開発するといった企画が主流になります。

- 必要性能を満たすか
- 納期は間に合うか
- 使いこなせるか

開発費を算出

- 予算内か

技術案

- 新たな技術が必要か
- 必要な技術をどのように確保するか

ワークフロー案

- データ作成などの開発段取りをどのように進めるか
- 制作物をどのような段取りで検証するか

開発計画の例：

名前	職種	1月	2月	3月	4月	5月	6月	7月	8月	9月
A	プロデューサー	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
B	ディレクター	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
C	プログラムリーダー	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
D	プログラマ	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
E	プログラマ	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
F	プログラマ	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
合計		6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5

例：月に半分だけこのプロジェクトをやっているので0.5

数字は人月工数

開発計画では月ごと、人ごとに人月工数を並べていく

④ 収支計画

プロジェクトの想定売上から想定費用を引くと想定利益になります。この想定利益が基準以上になるように調整するのが収支計画です。会社によって計算方法や基準が大きく異なりますので、ここでは簡単な概要を説明します。

想定売上は、パッケージ販売、ダウンロード販売、アイテム課金などの販売形態によって想定パターンや計算が異なります。想定費用とされるものは主に開発費、販促費(広告宣伝費)、間接費(光熱費や家賃、間接業務¹の人件費など)です。

利益は当然ながら黒字が求められますが、ただ黒字であればいいということはありません。利益が1円でもあれば数字上は黒字ということになりますが、それでは会社経

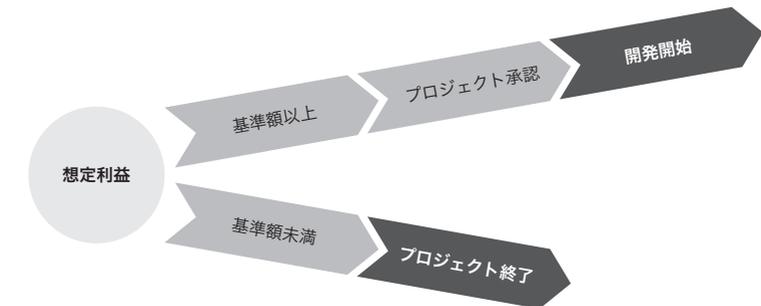
1：売上に直接関係する開発・製造・営業・マーケティングなどの業務が直接業務であり、間接業務とはそれ以外の人事・総務・経理・法務などの業務のこと。利益を算出するには、これら間接業務の費用も含んで考える必要がある。

営の諸経費を満たすことができないからです。費用に対して一定割合以上の金額といった利益基準額を超えることが求められます。



想定利益が基準額を超えるのが難しい場合は、仕様を一部削除して費用の削減を検討したり、広告宣伝の規模を下げてみたり、もっと強気の売上を想定できないかといった調整を行います。

仮にどうしても基準を超えられない場合、プロジェクトは承認されずに中断となってしまう可能性が大了。



⑤ 企画承認

企画書、開発計画、収支計画が揃ったところで、そのプロジェクトを開始するための組織内承認を求めることになります。なお組織や承認段階によっては、ここで広告や広報などの販促計画が求められることもあります。

組織の上位者は、

- 技術的な実現性
- 組織戦略における必要性
- 商売的な実現性
- 経営方針との合致
- 収支

などの検討と評価を行って承認するかどうかを判定します。ここで承認してもらえないとやり直しですが、最悪の場合は企画が没になってしまいます。

ただし、承認が下りなかったとしても企画が悪いとは限らないのも事実です。上位者がたまたまその時にやりたいと思っていたことと企画がマッチするのかどうかという運の要因も大きいので、確実に利益が上がりそうなプロジェクトだからといって承認され

● 人月単価の例

社員 20 名 社員の月給 30 万円 給料以外の費用 1,000 万円 / 月の場合

人月単価：

$(\text{全社員の給料合計 } 20 \times 30 + \text{給料以外の費用 } 1,000) / 20 = 80 \text{ 万円}$

人件費は人月工数と人月単価をかけることで算出できます。

プロジェクトの利益を計算する場合、この人件費の見積もりによって参加できる人数も決まってきます。

● 人件費

人件費 = 人月工数 × 人月単価

▶ プロジェクトのための組織（チーム）

プロジェクト実行のため、組織の中に新たな臨時組織を編成したものがチームと呼ばれます。プロジェクトは基本的にこのチーム単位で進められ、動いていきます。

▶ チームの種類

ひとくちにチームといっても、参加するメンバーによって細かい呼称が異なります。

会社の社員のように、組織に正式所属しているメンバーで開発する場合を内製と呼びます。これに対して、組織外の作業者に発注する場合を外注と呼びます。現代のチームは内製や外注をミックスした様々なメンバーの混成が主流です。

なお、プロジェクトを丸ごと外注し、組織内にはチームを置かず開発を行わないケースもありますが、本書の趣旨から外れますので最小限の説明とします。

チームの例

種類	構成
内製チーム	社員オンリーのチーム
内製+外注チーム	社員+別会社のチーム
フリーランス	会社に属しないフリーランスによるチーム
内製+外注+フリーランス……	様々なメンバーの混成
外注	チームは別会社にあつて、組織内には外注管理者のみが置かれる

▶ ゲーム開発に携わる職種

プロジェクトでは臨時に組織（チーム）が編成されます。

ゲーム開発のチームには様々な職種が集まり、職種ごとに異なる役割を持って協力し合うことでプロジェクトを進めていきます。通常の組織では上下で関係が成り立ちますが、プロジェクトのチームでは役割で成り立つ点が大きな特徴です。

それぞれの職種と業務内容を見ていきましょう。

● プロデューサー

- ・プロジェクトの総責任者
- ・企画を立ち上げる役
- ・人事や収支に責任を持つ



● ディレクター

- ・開発の総責任者
- ・ゲームの中身に責任を持つ
 - 面白さ
 - ビジュアル
 - ストーリーなど



● プランナー

- ・ゲームの仕様をまとめる¹
 - アイディア
 - ストーリー
 - キャラ
 - システム
 - バランス調整



● プログラマー

- ・プログラムを作る
 - バトル
 - CG
 - 通信
 - システムなど



1：ゲームの仕様設計に特化した場合はプランナーと区別してゲームデザイナーと呼ばれる。

- タスクの依存順：
依存元のタスクは、依存先のタスクよりも優先度が高くなる

これらを総合すると、より必要性が高く、よりリスクが大きく、より依存されているタスクが高い優先度になります。

▶ タスクの優先度の例

以下はコンセプトを「ドリフトが気持ちよいレースゲーム」、マイルストーンの目標を「ドリフト操作の気持ち良さを確認する」とした場合の例です。

目標達成に必須であり、かつ実現できなかった場合にコンセプトが破綻して致命的なリスクとなるタスク「車のドリフト機能を作る」が優先度 S となっています。そのタスクが依存しているタスクも S、その他、気持ち良さにとって重要なタスクが A です。

優先度	タスク
S	車の基本操作を作る
S	車のドリフト機能を作る
S	基本的な車を 1 台デザインする
S	ドリフトできる基本的なコースを一か所デザインする
A	車の走行音を鳴らす
B	キャラクター別に車をデザインする
B	車に乗るキャラクターをデザインする
B	BGM を鳴らす
B	キャラクターにしゃべらせる
C	キャラクターが登場シーンで踊る

▶ タスクはプロジェクト

タスクはそれぞれが以下のようなプロジェクト的要素を持っています。

- 目的
- 期間
- 実行者
- 内容 (成果)

つまりタスクはひとつひとつが小さいプロジェクトといえます。

タスクの担当者が小さなプロジェクトの責任者として各要素を把握・管理していくようにすると、プロジェクトは自律的に進むようになります。

そのためにも、作業者が言われたことをやるだけという状況は避けて、プロジェクトチームの全員が自己管理できるようにしていきましょう。

▶ タスク管理の破綻で起こる「デスマーチ」

タスク管理が破綻して、プロジェクトの終わりが見えなくなった状況をデスマーチと呼びます。

デスマーチはプロジェクトにおける最悪の状況です。タスクは爆発的に増えていき、タスクを終わらせてもマイルストーンが進展せず、チームメンバーを増員してタスクに割り振っても状況が改善しません。

開発は延期が重ねられ、最終的には中止されるか、目的から遠く離れた低品質の成果物が残されます。チームは疲れきって、未来につながる経験は残らず、つらい思い出だけが記憶される結果となります。

デスマーチ：



- 開発の延期
- 中止になることも
- 製品の質にも影響



- やることだけがが増えていく
- 業務が進展しない



- 肉体的、精神的負担
- 正しい成長が得られない

デスマーチは以下のような状況が複合して発生します。どれかひとつでも思い当たるものがあれば注意すべきでしょう。

- コンセプトを達成するために必要な人員が全く足りていないが、それをチームが把握していない、もしくは人員不足をあきらめている
- プロジェクトの期間が本来必要な期間よりもはるかに短い
- 目的に対して適切なコンセプトが設定できておらず、コンセプトに沿ったタスク作業を進めても目的を達成できない
- そもそもコンセプトが実現不可能なのに、それを無視してタスク設定されている

また、基本的にはデスマーチも突発的に発生するものではありません。突入する前には必ず何らかの兆候が表れます。

例として以下のような状況がデスマーチの兆しです。業務中にこれらの状況に気づいたらできるだけ迅速に解消できるよう努めましょう。