

# ***Lychee Redmine*** &

# **P M B O K**

Project Management Body of Knowledge



## ▶ 今回のゴール

---

本資料は以下の事が「理解できている」ことをゴールとします。

01

### プロジェクトマネジメントとは？

- プロジェクトとは？プロジェクトマネジメントとは？
- プロジェクト成功の三大制約とは？

02

### PMBOKとは何か？

- PMBOKとはどういう考え方か？
- なぜPMBOKの理解が必要か？
- 主要な構成要素の概要説明

03

### Lychee Redmineでできることは？

- Lychee Redmineと10の知識エリアの対応状況は？
- 具体的にどの機能をどう使えばいいのか？



01

# プロジェクト マネジメントとは？

PMBOKを語る前に  
プロジェクトマネジメントとはそもそも何か？

## ▶ プロジェクト？プロジェクトマネジメント？

### プロジェクトとは？

通常の業務とは異なる

**独自の**プロダクト

サービス

所産 (Results)

を創造するために実施する

**有期性**のある業務。

品質 (Quality)

資源 (Cost)

時間 (Delivery)

の3つの要素を管理し、

バランスを取りながら行う。

### プロジェクトマネジメントとは？

プロジェクトの**要求事項**を満足させるために  
知識、スキル、ツール、および技法を  
**プロジェクト活動に適用**させること。

プロジェクトを**効果的**に  
計画、実行、**コントロール**すること。

限られた時間、予算、資源の中で  
設定された目標を達成するために

計画

実行

監視

コントロール

を

行う体系的手法。

## ▶ プロジェクト成功の三大制約

---

品質 (Quality)

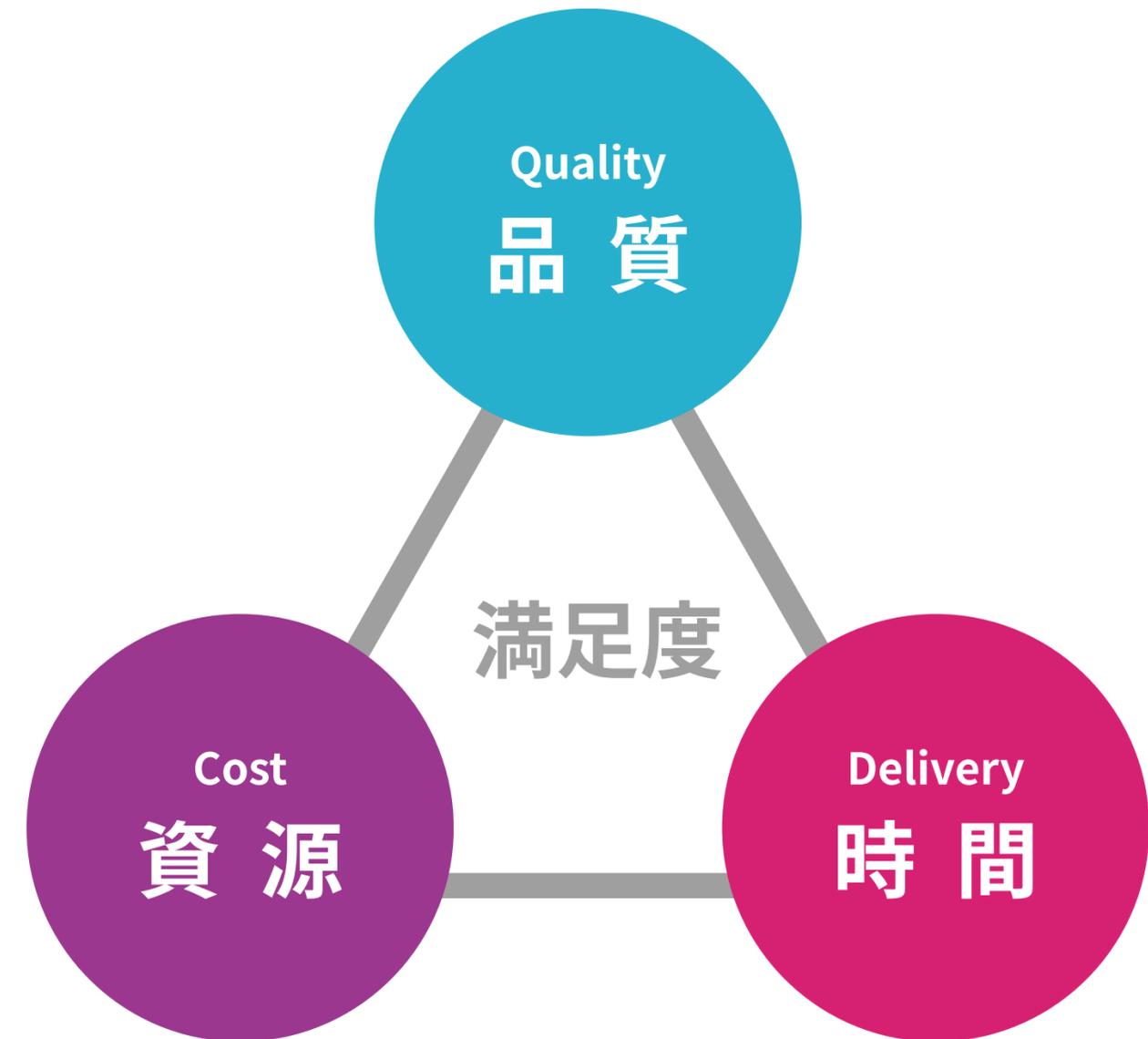
資源 (Cost)

時間 (Delivery)

の3つの要素。

この3要素は互いにトレードオフの関係にあり、**1つを変更すれば他にも影響**が出る。

この**バランス**を保ちつつ、顧客満足度を高めることが重要。



02

# PMBOKとは 何か？

マネージメントをする上で名称が出てくる「PMBOK」  
果たして一体何なのか？

## ▶ PMBOKとは？

---

### ❓ PMBOK (Project Management Body Of Knowledge)



**アメリカのPMI** (Project Management Institute) が策定・発行しています。



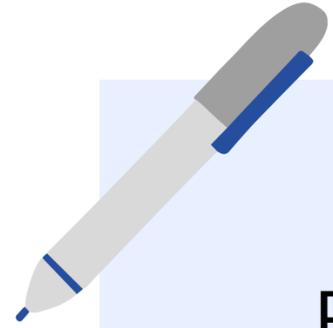
プロジェクトマネジメントの「教科書」かつ「辞書」であり、**世界中**のプロジェクトマネージャーが共通認識を持つための**標準ガイド**です。



規律的実務慣行ではなく、**模範的実務慣行**になります。

## ▶ なぜPMBOKの理解が必要？

---



PMBOKを学び、世界標準のプロジェクトマネジメントを理解すると、  
様々なお客様の状況に応じて、対応することができます。

**営業であっても、情シスであっても、開発であっても、品証であっても…  
どの業態であっても、標準ガイドとして適応が可能です。**

**理解を深めておくことで、プロジェクト管理の方法が学べます。**

それがPMBOKの理解が必要な理由なのです。

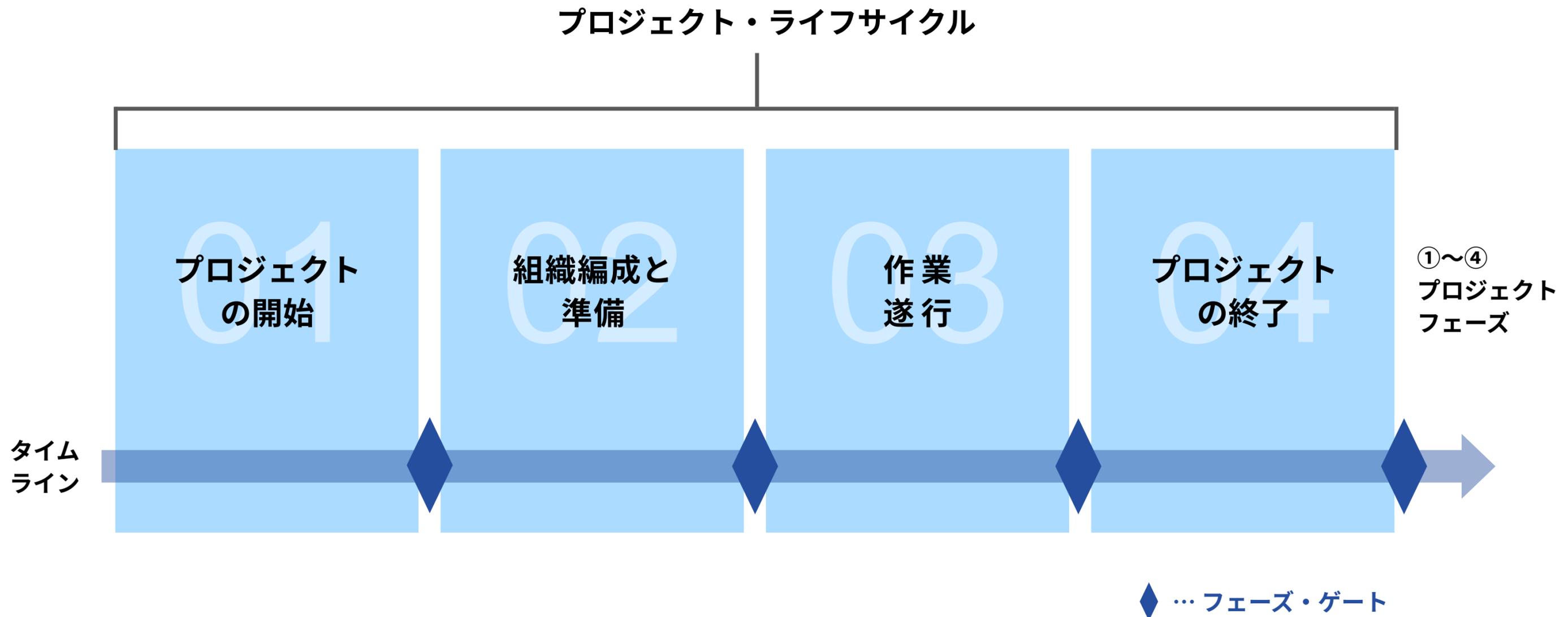


## ▶ 「PMBOKガイド」の主要な構成要素

主要な構成要素	概要
1. プロジェクト・ライフサイクル	プロジェクトの開始から完了に至るまで、プロジェクトが経験する一連のフェーズ
2. プロジェクト・フェーズ	論理的に関連のあるプロジェクトのアクティビティの集合。ひとつ以上の成果物の完了によって終了する。
3. フェーズ・ゲート	各プロジェクト・フェーズの終了時点で行われるレビュー。次のフェーズへ進めるかどうか、またはプロジェクトを継続・修正・中止するかを意思決定する。
4. プロジェクトマネジメント・プロセス	プロジェクトを成功に導くために、計画を立てて、実行し、進捗を管理しながら完了させる一連の手順
5. プロジェクトマネジメント・プロセス群	プロジェクトを成功に導くために、必要な情報をもとに、適切な手法を使って成果を得る5つのプロセス群
6. プロジェクトマネジメント知識エリア	プロジェクトをうまく進めるために必要な10の重要な要素のこと。それぞれの分野には「何をやるか」「どうやってやるか」「何が必要で、何が成果か」が整理されている。

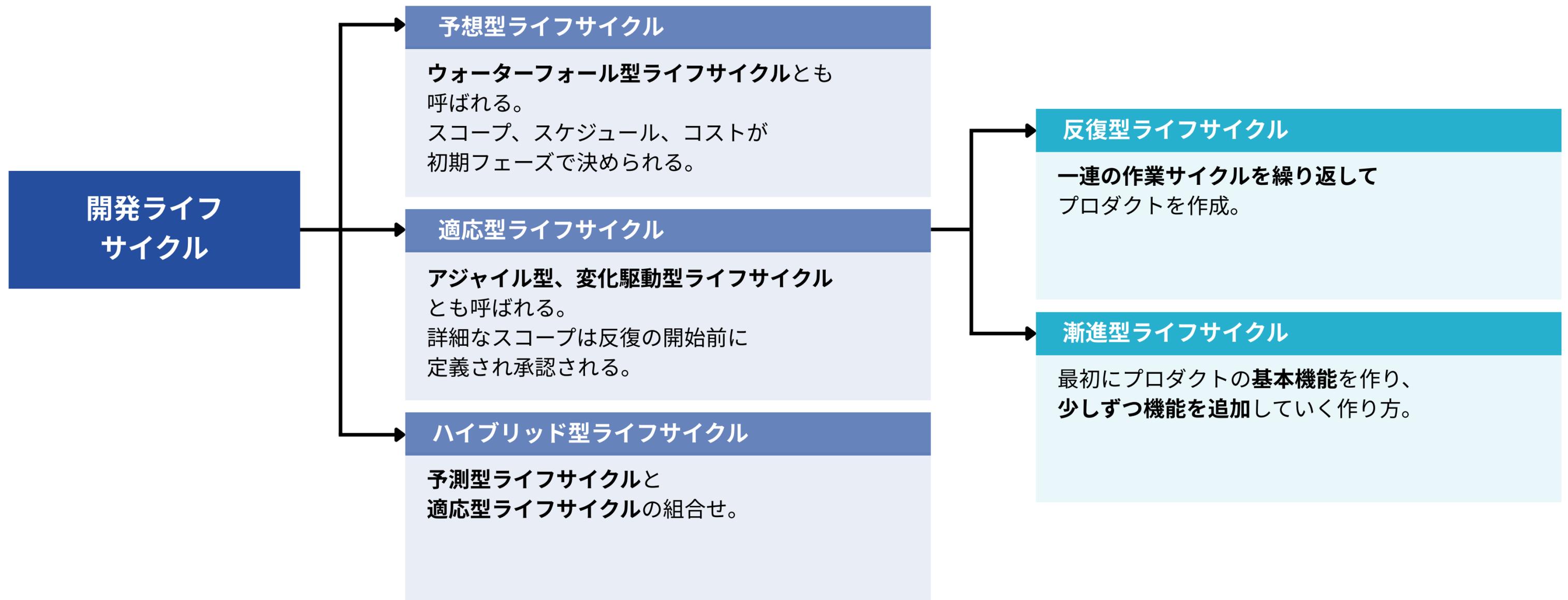
## ▶ プロジェクト・ライフサイクル

プロジェクト・ライフサイクルは、プロジェクトがその開始から完了までに経験する一連のフェーズ。



## ▶ 開発ライフサイクルの種類

開発ライフサイクルはプロジェクト・ライフサイクル内のフェーズであり、通常、製品の開発、サービス、所産に関連付けられるひとつ以上のフェーズ



## ▶ プロジェクトマネジメント・プロセスとプロセス群

プロジェクトの開始から終了までの5つの段階とそれをまとめたもの。  
5つの段階はそれぞれ「立ち上げ」「計画」「実行」「監視・コントロール」「終結」。

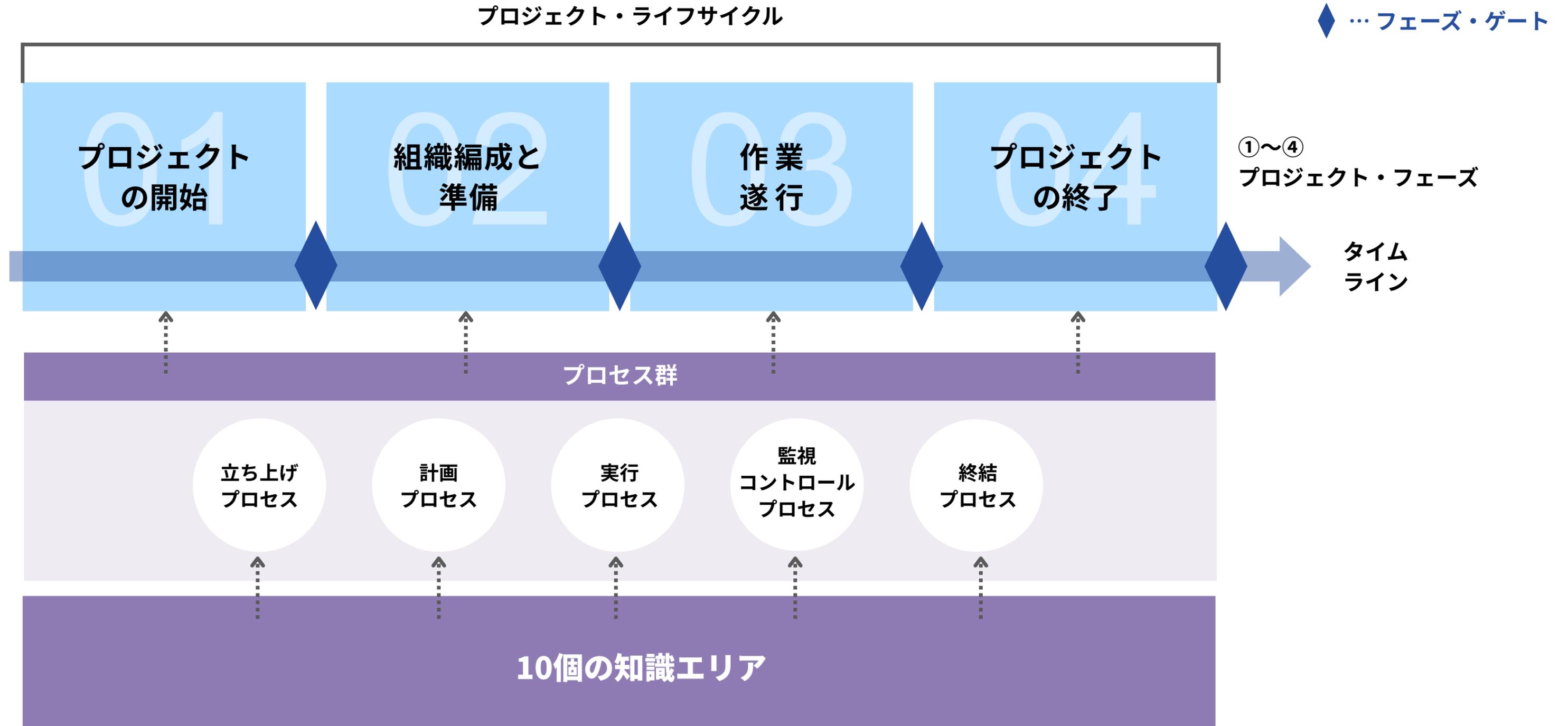
プロジェクトマネジメント・プロセス群	
1. 立ち上げプロセス群	プロジェクトやフェーズを正式に始めるために、目的や背景を整理し、実施の承認を得るプロセス群
2. 計画プロセス群	プロジェクトの目標を達成するために、作業内容・スケジュール・リスクなどを整理し、進め方を具体化するプロセス群
3. 実行プロセス群	計画に基づいて関係者と連携しながら作業を進め、成果物の作成を行うプロセス群
4. 監視・コントロールプロセス群	プロジェクトの進捗や品質、コストなどを確認・調整し、計画からのズレを管理するプロセス群
5. 終結プロセス群	プロジェクトやフェーズを正式に完了させ、成果の確認や振り返りを行って締めくくるプロセス群

## ▶ PMBOK 10の知識エリアとは？

PMBOK（第6版）に定義されており、**プロジェクトマネージャーが注意を払うべき管理上の10個の領域**のことになります。プロジェクトマネジメントに必要な知識を体系的に分類したもので、各エリアごとにプロセスやツール、成果物が定義されています。以下に各エリアとその概要をまとめます。

知識エリア	概要
1. 統合マネジメント	プロジェクト全体の計画・実行・変更を一貫してまとめる司令塔的な役割
2. スコープ・マネジメント	作業範囲と成果物を明確にして、やるべき作業を確実に実行するための管理
3. スケジュール・マネジメント	いつ何をするかを決めて、期限通りに終わらせるための時間管理
4. コスト・マネジメント	予算をオーバーしないように計画・監視・調整するお金の管理
5. 品質マネジメント	顧客が満足するレベルの成果物を、安定して出すための品質管理
6. 資源マネジメント	人・モノ・設備など必要なリソースを計画的に用意・配分する管理
7. コミュニケーション・マネジメント	会議や記録、法令もふまえて関係者に円滑に情報を伝え合うための仕組み
8. リスク・マネジメント	起こりうるリスクを見つけて分析し、備えや対応を実行・監視するための仕組み
9. 調達マネジメント	外部業者との契約や購入など、外注や物品の調達を適切に進める管理
10. ステークホルダー・マネジメント	利害関係者（顧客・上司・関係部署など）を洗い出し、関与度合いを調整し協力を引き出す仕組み

# ▶ プロジェクトライフサイクルとの関わり



03

# Lychee Redmine では？

PMBOKの「10の知識エリア」に対して  
Lychee Redmineが対応できること

# ▶ Lychee RedmineにおけるPMBOK「10の知識エリア」対応概要

Lychee Redmineが「10の知識エリア」のどの部分に対応しているかを示します。詳細は次ページから。

知識エリア	概況	Lychee Redmineでの対応の詳細
1. 統合マネジメント	○	全体管理の一部として利用可能。プロジェクト全体の統合には別途補完が必要。
2. スコープ・マネジメント	◎	作業内容や成果物に関する情報をチケットに記載・管理可能。
3. スケジュール・マネジメント	◎	ガントチャートとチケットを活用し、WBSの作成およびスケジュール管理が可能。
4. コスト・マネジメント	○	予算計画の作成は不可。予算計画以降はEVMを用いた進捗・コスト管理に対応。
5. 品質マネジメント	○	トラッカーやカスタムフィールドを利用しバグ管理が可能。テスト項目管理などは非対応。
6. 資源マネジメント	○	人的資源はリソースマネジメント機能で工数管理が可能。チーム育成や物的資源の管理には非対応。
7. コミュニケーション・マネジメント	◎	Wiki機能を活用して会議情報や使用ツールの共有が可能。
8. リスク・マネジメント	△	特定されたリスクの記録・状況管理はチケットで対応可能。それ以外のリスク分析は難しい。
9. 調達マネジメント	×	非対応。
10. ステークホルダー・マネジメント	△	関係者の整理・共有はチケットまたはWikiへの記載で対応可能。

01

# スコープ マネジメント

作業範囲と成果物を明確にして、  
やるべき作業を確実に実行するための管理

## やるべきこと

### 計画

ユーザー要求・要件を整理し、プロジェクトの範囲（スコープ）を定義する

### 計画

WBS（作業分解構成図）を作って、必要な作業やタスクを細かく洗い出す

### 実行

スコープ変更のルールを決めておく  
（例：誰が承認するか）

### 監視・コントロール

実際の成果が計画したスコープ通りかを確認する

### 監視・コントロール

スコープクリープ  
（無計画な追加作業）を防止する

## Lycheeでの対応とやり方

### 要求・要件をチケット化し、作業範囲と成果物の定義を記載する

チケットの説明欄やカスタムフィールド、チェックリストを活用する。スコープ記述書のようなストック情報はWikiに記載する。

### 作業をチケット化し、WBSを書く

チケットには、詳細な説明や担当・制約条件なども記録できる。ガントチャート機能を使うとタスクの階層構造がわかりやすい。

### 承認用のワークフローを作成し、スコープ変更時はチケットを起票して承認/否認できるようにしておく

チケットコメントと履歴機能で、この仕様になった履歴を残せる。

### 成果物の受け入れ基準をチケットのチェックリスト機能でまとめ、レビュー時に要件通りかどうかを確認する

フィードバックの内容もコメントで記載できる。

### チケットが追加されたら通知されるので、無駄なチケットかどうかを判断し、必要なら計画に組み込む

## ▶ 要求・要件をチケット化し、作業範囲と成果物の定義を記載する

作業内容や成果物に関してはチケットに情報を記載し、管理することが可能です。  
プロジェクト始める時には必ずやるべきことをチケット発行する必要があります。

### 作業内容の指示（チケット）



#### 購入ボタン制御機能を開発する

赤石 ライチ さんが約7時間前に追加. 1分未満前に更新.

ステータス:	着手前	開始日:	2025/03/24
優先度:	通常	期日:	2025/08/04 (期日まで 約4ヶ月)
担当者:	赤石 ライチ	実開始日:	
対象バージョン:	αバージョン	実終了日:	

#### 説明

自動販売機における「購入」ボタンの制御機能を実装する。  
以下の条件に応じて購入ボタンの有効/無効状態を制御する。

想定される制御条件：

- 投入金額が商品価格に満たない場合、ボタンを無効化
- 対象商品が売り切れの場合、ボタンを無効化
- 購入可能な状態のときのみ、ボタンを有効化
- 障害・エラー（例：通信断、機械エラー）が発生した際は、すべてのボタンを無効化

UI 及びロジックの両面から制御を行うことが求められる。

### 成果物の定義（チェックリスト）

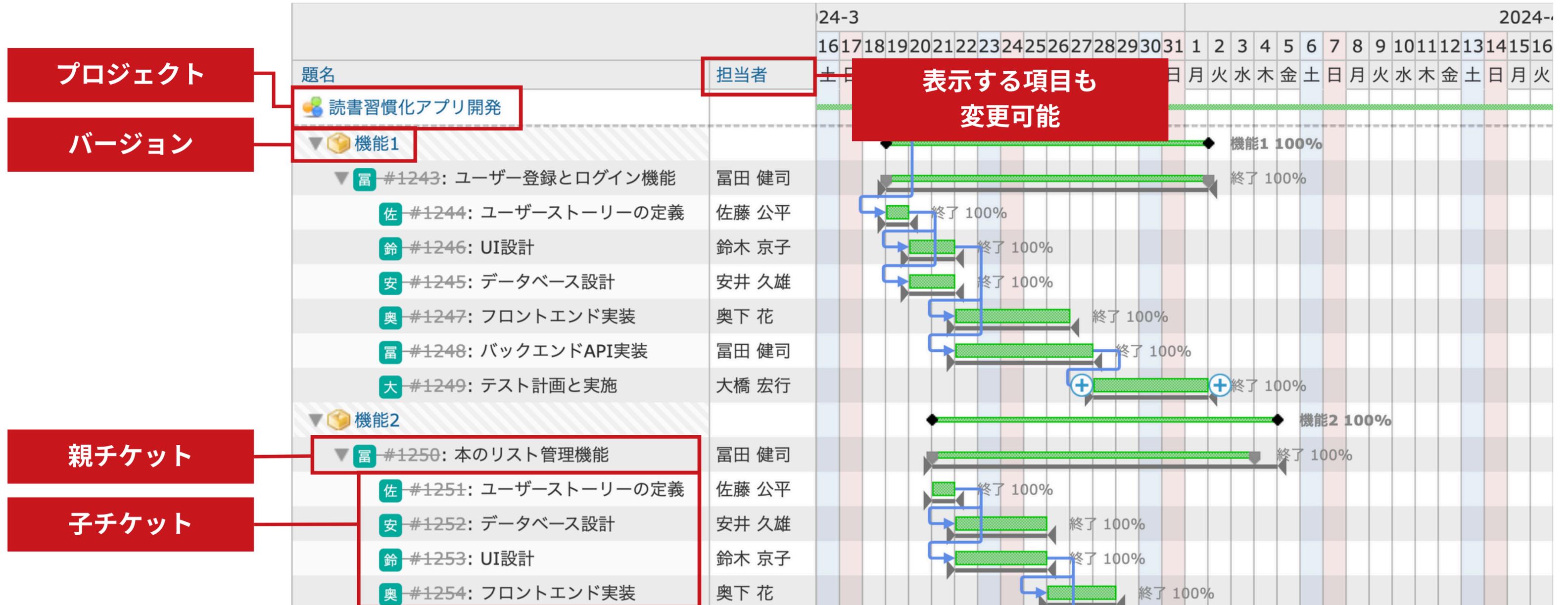
細かな作業や確認項目をリスト化して抜け漏れを防ぎ、確認しながら作業を進められます。

#### チェックリスト

- ☑ UI上で、投入金額・在庫状態・システム状態に応じて購入ボタンの有効/無効が切り替わる
- ☑ 商品ごとの価格・在庫情報を正しく参照してボタン制御できている
- ☑ 無効状態のボタンは押せない（反応しない）UIになっている
- ☐ システムエラー時にすべての購入ボタンが無効化される
- ☐ 単体テストで各条件に応じた挙動を網羅し、期待通りの制御が行われていることを確認
- ☐ UI・UXレビューを通過している（レビュー担当者がOKを出した）
- ☐ 実機またはエミュレーターでの統合テストで、制御が期待通りに動作することを確認

# ▶ 作業をチケット化し、WBSを書く

チケットには親子階層をつけることができます。ガントチャートで確認するとWBSのように見やすいです。



▶ 承認用のワークフローを作成し、スコープ変更時はチケットを起票して承認する

トラッカー機能によりチケットにワークフローを設定することができます。  
 ワークフローステータスは自由に作成できます。  
 チケットの担当者には、**次に行動してほしい人をアサイン**しましょう。  
 チケットが追加・変更されたら、担当者に通知されます。

「タスク」トラッカーのワークフロー例



「変更要求」トラッカーのワークフロー例



**変更要求の詳細はチケットに記載して残します。**  
 変更要求が承認されたら、具体的な作業チケットを追加します。

Plus One!

# チケット発行を簡単にするには？

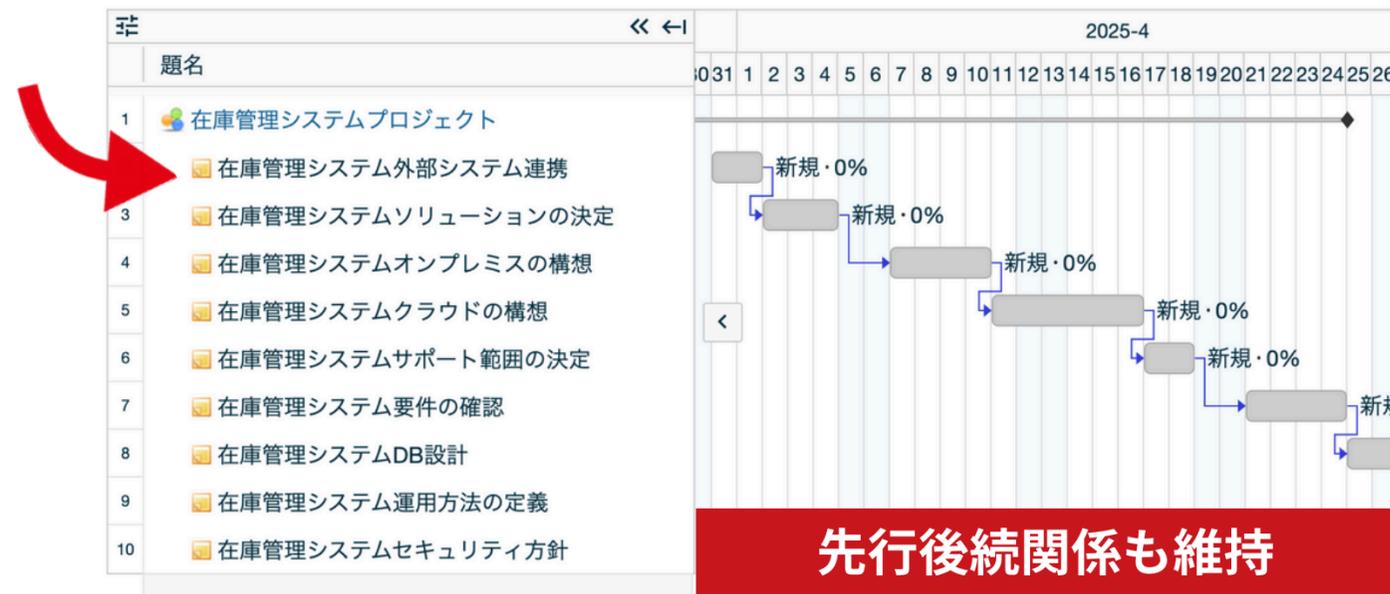
## 定型作業（チケットセット）

複数のチケットを親子、先行後続関係などの関連とともに、名前を付けてテンプレート化することができます。簡単に複数のチケットを一括生成できます。

チケットセットからチケット作成

チケットセット

<input type="checkbox"/>	#	トラッカー	題名	ステータス	優先度
<input type="checkbox"/>	1	機能	%s要件の確認	新規	通常
<input type="checkbox"/>	2	機能	%sソリューションの決定	新規	通常
<input type="checkbox"/>	3	機能	%sオンプレミスの構想	新規	通常



## 定型内容（チケットテンプレート）

よく使うチケットの書き方をひな型として登録し、再利用できる機能です。チケットを新規作成するとき、プルダウンで選択できます。

テンプレート名 \*

チケットへの適用項目:

トラッカー \*

チケットタイトル

[この項目のヘルプ](#)

チケット本文 \*

**B I U C H1 H2 H3**

### 終了条件

以下のことを実施することで本タスクを「完了」とすることができる。

- 要件定義書のレビュー結果承認完了

### 実施内容

以下のことを実施すること

- 要件定義書の作成
- 要件定義書のレビュー会

Plus One!

# チケットにはコメントや操作ログを残せる

大澤 が4ヶ月前に更新

Lycheeクラウドの利用規約、修正完了しました。レビューをお願いします。

<保存先フォルダ>

https://

<該当資料名>

Lychee Redmine利用規定\_onCloud\_20190401.docx

<検討資料>

https://

<改定内容>

1. 全体

Lychee Redmine on Cloud → Lycheeクラウド

だれが

いつ

ド  
い  
員書  
加  
注意  
づく損害についてはこの限りではありません。」

萬谷 が4ヶ月前に更新

- 期日を 2019/01/31 から 2019/02/11 に変更
- ステータスを Doing から Done (確認待ち) に変更
- 進捗率を 40 から 80 に変更

なにをした

レビュー確認中

大澤 が4ヶ月前に更新

- 期日を 2018/12/31 から 2019/01/31 に変更
- ステータスを Done (確認待ち) から Doing に変更
- 担当者を 水口 から 大澤 に変更
- 進捗率を 80 から 40 に変更

#10



チケットには、自由に書き込めるコメント欄があります。

「作業の進捗報告」「質問への回答」「決まった方針のメモや議事録」など、やり取りを残す場所です。

投稿すると日時と投稿者が自動で記録され、あとで自分や管理者が編集・削除しても履歴に痕跡が残ります。

#15

コメントが追加されると、担当者やウォッチャーに通知を飛ばす設定もできます。

#14

02

# スケジュール マネジメント

いつ何をするかを決めて、  
期限通りに終わらせるための時間管理

## やるべきこと

## Lycheeでの対応とやり方

## 計画

アクティビティを定義する



WBSを分解し、アクティビティ単位でチケット化する  
チケットは無限に階層化可能。

## 計画

タスク所要時間を見積もる  
(人日・時間など)

見積もった予定工数をチケットに記載する  
ガントチャートのオートスケジュール機能を使えば、  
予定工数から自動的にスケジュールを引いてくれる。

## 計画

タスクの依存関係を明確にする  
(例：「Aが終わらないとBが始められない」)

チケットに先行後続をつける

## 計画

スケジュールを可視化する



ガントチャート機能を使う

## 計画

重要な工程の遅延を防ぐ  
クリティカルパスを把握する

ガントチャート上でクリティカルパスを明確化できる

## 監視・コントロール

定期的に進捗状況を確認し、  
遅延があれば対策する

遅延の確認にはイナズマ線機能  
以前の計画との差分の確認にはベースライン機能を使う  
ベースラインで、どの段階で計画が変更されたのかがわかる。

## 監視・コントロール

スケジュール変更が発生したら  
関係者と共有するスケジュールが変更されたら担当者に通知される  
定例mtgでガントチャートを確認し、共有する



## ▶ 見積もった予定工数をチケットに記載する

各チケットの予定工数を見積もり、入力しておきましょう。

オートスケジュール機能を使えば、開始日を指定して予定工数から自動的にスケジュールを引いてくれます。

The image shows two side-by-side screenshots of a project management software interface. The left screenshot shows a task list with estimated work hours being entered. A red box highlights the '40' entered in the '予定工数' column for 'タスク #4434: プロセス設計'. A red arrow points to the right screenshot, which shows the same task list with Gantt bars generated for each task. A red box highlights the Gantt bars for the first three tasks, with the text 'オートスケジュールでガントバーも一発設定' (Auto-schedule Gantt bars in one click). A red callout box with the text 'ダブルクリックで簡単入力' (Simple input with double-click) points to the '40' in the left screenshot.

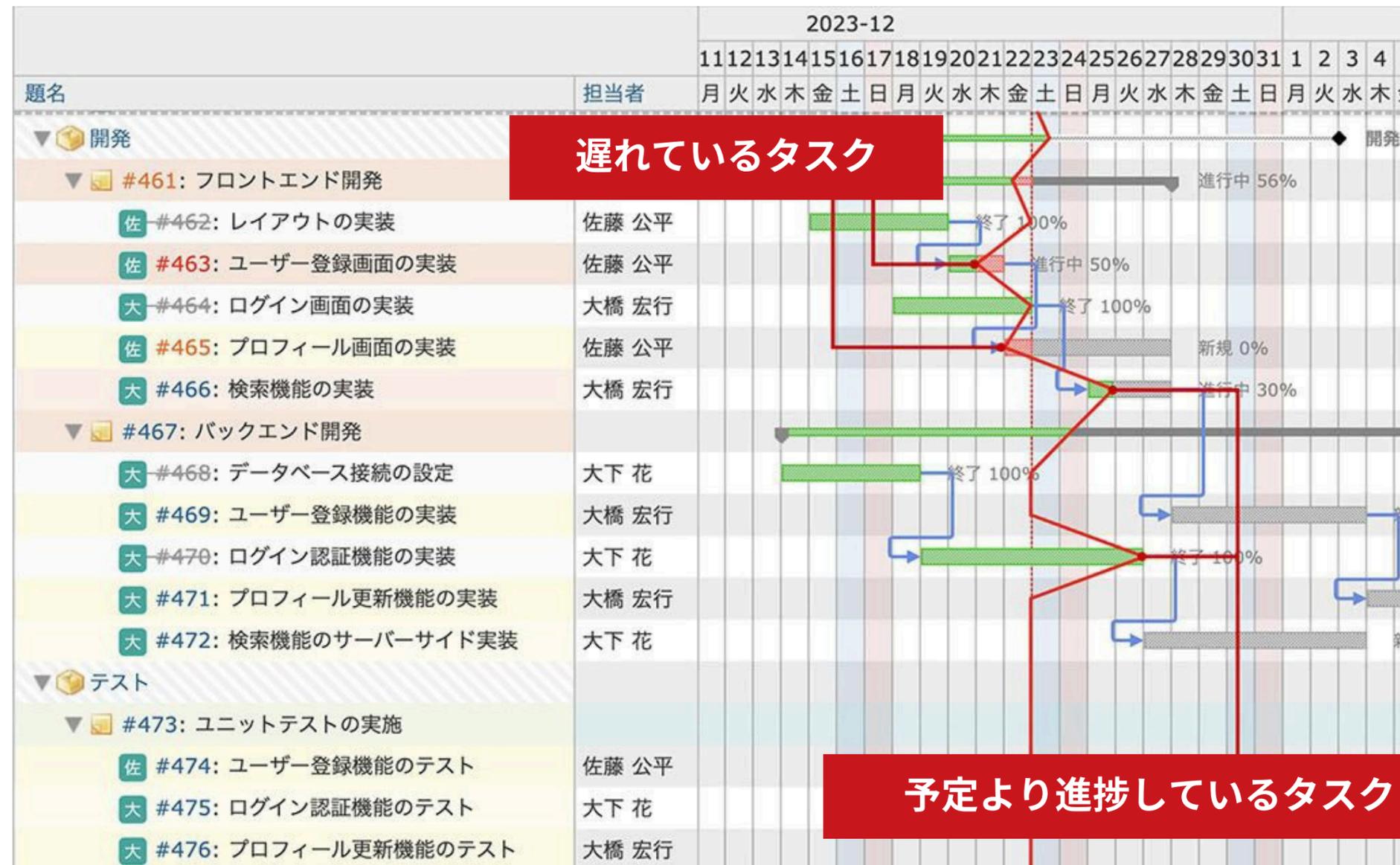
題名	担当者	予定工数	金	土	日	月	火	水	木	金
社内システム開発										
販売会計システム開発										
要件定義										
タスク #4431: 要件の洗い出し		40								
タスク #4432: システム化方針の検討		24								
タスク #4433: 要件定義書作成		32								
設計										
タスク #4434: プロセス設計		40								
タスク #4435: 移行設計		40								
タスク #4437: 画面帳票設計		40								
タスク #4439: 共通部品設計		40								
タスク #4440: DB設計										
タスク #4441: スケジュール作成										

ダブルクリックで簡単入力

オートスケジュールでガントバーも一発設定

## ▶ イナズマ線を表示させ、遅れているタスクを見える化

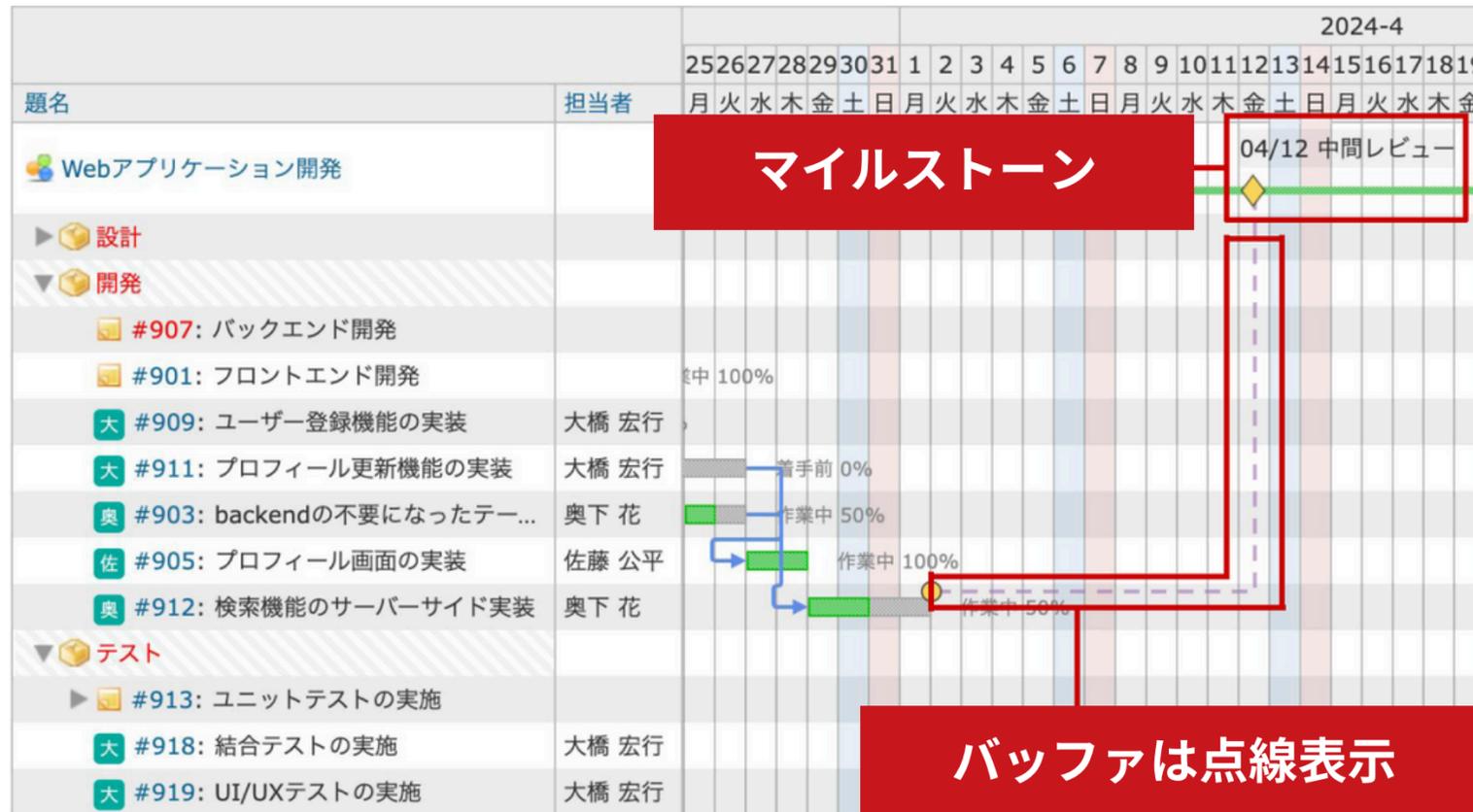
今日の日付をベースに進捗状況を可視化します。遅れているタスク（作業）がピンポイントでわかります。



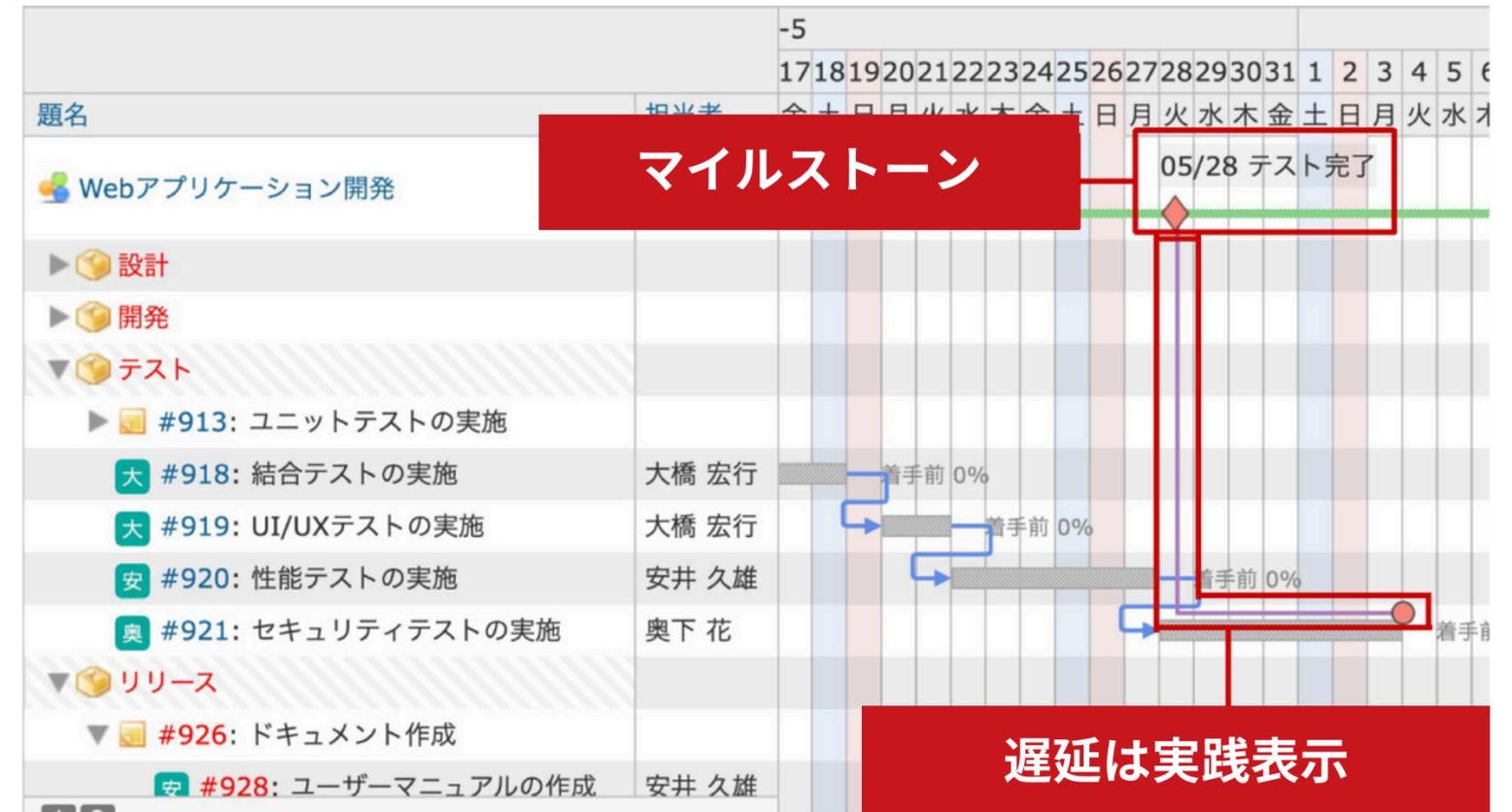
## ▶ 重要なイベントをマイルストーン設定

プロジェクトの重要な期日にマイルストーンを設定できます。  
 タスクと紐付けることでマイルストーンまでのバッファを可視化できます。

バッファがある時



遅延している時



# ▶ カンバンを利用した状況更新

カンバンで更新してもガントチャートにすぐ反映されます。  
 作業担当者はカンバンで、プロジェクトマネージャーはガントチャートで管理できます。

担当者	新規	作業中
佐藤 公平 <span>佐</span>		#1307 性能テスト アプリケーションの性能評価と
予残 80H 実 0H 80H 未完 80H		
富田 健司 <span>富</span>	性能テスト #1309 ストレステスト実施 5/22 システムが限界状態まで耐えられるかを確認するストレステスト	セキュリティテスト #1306 修正措置実施 検出されたセキュリティ問題の作業
予残 32H 実 0H 32H 未完 32H	16H	
大橋 宏行 <span>大</span>	性能テスト #1310 修正措置実施 5/24 性能テス	
予残 32H 実 0H 32H 未完 32H		
奥下 花 <span>奥</span>	セキュリティテスト #1304 脆弱性スキャン実施 5/10	

**DRAG & DROP!**

取り掛かり始めたら  
「作業中」に移動しよう!

担当者	新規(8)	
佐藤 公平 <span>佐</span>		#2806 タスク ストレステスト実施
予残 0H 実 0H 0H 未完 0H		
富田 健司 <span>大</span>	性能テスト #2806 ストレステスト実施 4/4 システムが限界状態まで耐えられるかを確認するストレステスト 極端な高負荷状態におけるシ...	セキュリティ #2806 検出 作業
予残 0H 実 0H 0H 未完 0H	H	
大橋 宏行 <span>富</span>	性能テスト	
予残 0H 実 0H 0H 未完 0H		
奥下 花 <span>大</span>	セキュリティテスト #2810 脆弱性スキャン実施 4/4 自動ツールを使用したセキュリティ脆弱性のスキャン	
予残 0H 実 0H 0H 未完 0H	H	

#2806 タスク ×

ストレステスト実施

プロパティ コメント/履歴 作業時間

説明

システムが限界状態まで耐えられるかを確認するストレステスト  
 極端な高負荷状態におけるシステムの耐久性・挙動を確認する。

- 本番環境に類似したステージング環境を使用
- データベース：本番相当のスキーマ・初期データあり
- 外部サービス：通知・ログ収集などMock化（または制限付き連携）

プロジェクト\* Webアプリケーション開発

ステータス\* 新規

担当者 大橋 宏行

優先度\* 通常

カテゴリ -

対象バージョン 開発

親チケット #2805 性能テスト

ブロックング いいえ

**CLICK!**

チケットをクリックすると  
その場で編集可能

## ▶ ロードマップ機能でバージョンごとの進捗状況を確認

### ロードマップ

[+ 新しいバージョン](#)

#### 自動販売機開発プロジェクト\_ソフト - αバージョン 進行中



期日まで 約4ヶ月 (2025/03/24 - 2025/07/18)

 0%

14 チケット (0件完了 - 14件未完了)

#### 関連するチケット

自動販売機開発プロジェクト\_ソフト - 機能 #3030: 購入ボタン制御機能を開発する...

自動販売機開発プロジェクト\_ソフト - 機能 #3037: おつりボタン制御機能...



#### 自動販売機開発プロジェクト\_ソフト - βバージョン 進行中



開始日まで 約4ヶ月 (2025/07/21 - 2025/10/03)

 0%

14 チケット (0件完了 - 14件未完了)

#### 関連するチケット

自動販売機開発プロジェクト\_ソフト - 機能 #3044: 冷蔵温度管理機能...

自動販売機開発プロジェクト\_ソフト - 機能 #3051: 商品排出機能...



プロジェクトの全体的なスケジュールを俯瞰できる機能です。

バージョンに紐づいたチケットを一覧表示し、進捗や期日を可視化できます。

## ▶ ダッシュボードで期日が近いチケットを確認し進捗遅れを防止

カスタムクエリを使って期日が近いチケット一覧をリスト化しておけば、ダッシュボードにアクセスするだけで状況が分かります。

### STEP.1

【事前準備】カスタムクエリを作成

プロジェクト 活動 チケット 作業時間 ガントチャート ガントチ

チケット

設定したいフィルタ条件

▼ フィルタ

- ステータス
- 期日   日

フィルタ追加

オプション

✓適用   カスタムクエリを保存

CLICK!

### STEP.2

ダッシュボードを定期的に確認

### ダッシュボード + パネル追加

#### 期日が近いチケット一覧

条件に合ったチケットを一覧で表示

#	トラッカー	題名	担当者	期日	進捗率
3034	タスク	要件定義作業	河崎 美雨	2025/04/06	<div style="width: 100%;"></div>
3027	タスク	要件定義作業	重田 登	2025/04/06	<div style="width: 100%;"></div>

一覧で見る

03

# コスト マネジメント

予算をオーバーしないように  
計画・監視・調整するお金の管理

## やるべきこと

## Lycheeでの対応とやり方

## 計画

プロジェクトの予算を立てる  
(人件費、ツール代、外注費など)



プロジェクトの立てた予算をプロジェクトの概要から設定する  
EVM機能の予算に利用する。

## 計画

各タスクのコストを見積もる



チケットの予定工数とコストグループを設定すれば自動で計算する  
コストグループは、作業者をまとめて時間単価を設定する機能。

## 計画

実際にかかったコストを記録・集計し、  
予算と比較する



作業担当者が日々チケットに作業時間を記録しておく、EVM機能  
を使って確認できる  
コストパフォーマンス (CPI) を確認する。

## 監視・コントロール

コスト超過しそうなときは、スコープ  
調整・リソース調整を検討する



EVMのコストパフォーマンス指標 (CPI) を確認する。  
CPIが悪化している場合はスコープ・リソース・スケジュールの調整を行うこと。

## 監視・コントロール

定期的にコストレポートを作成して  
関係者に報告する

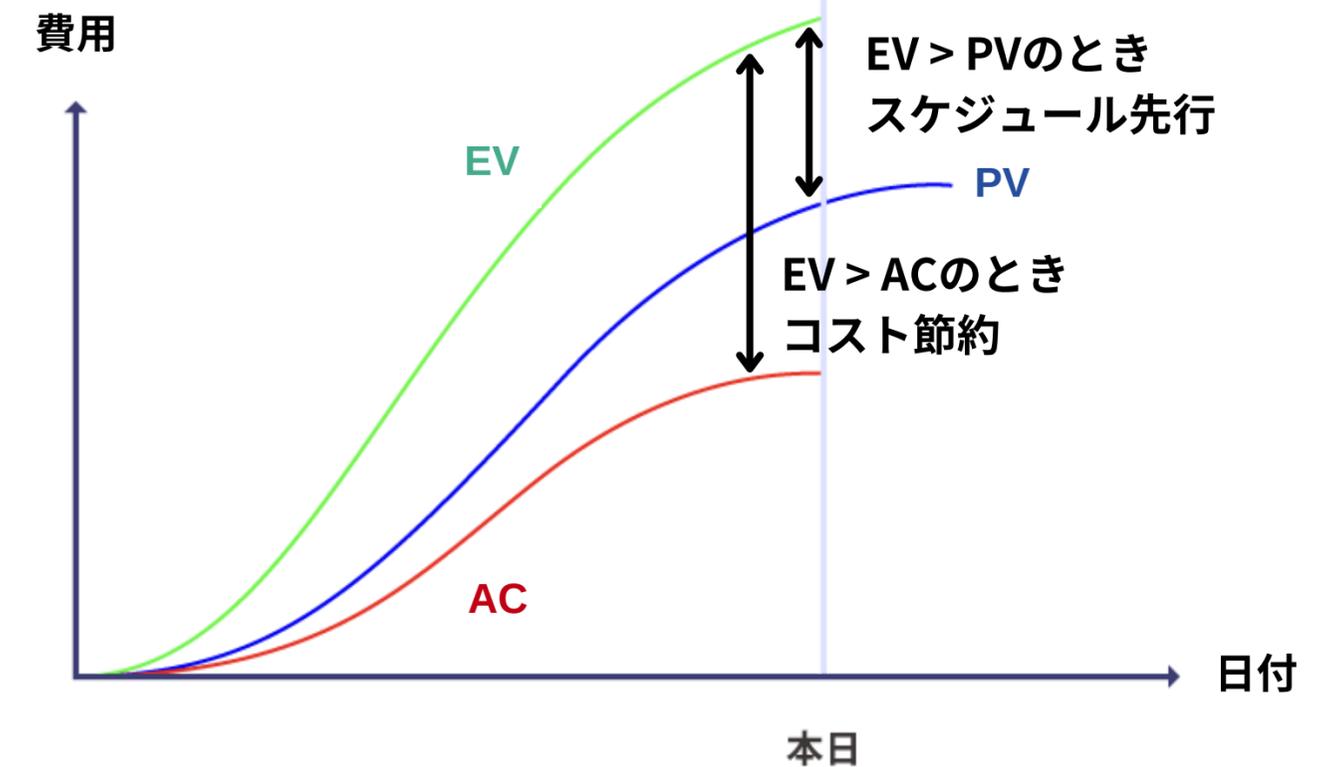
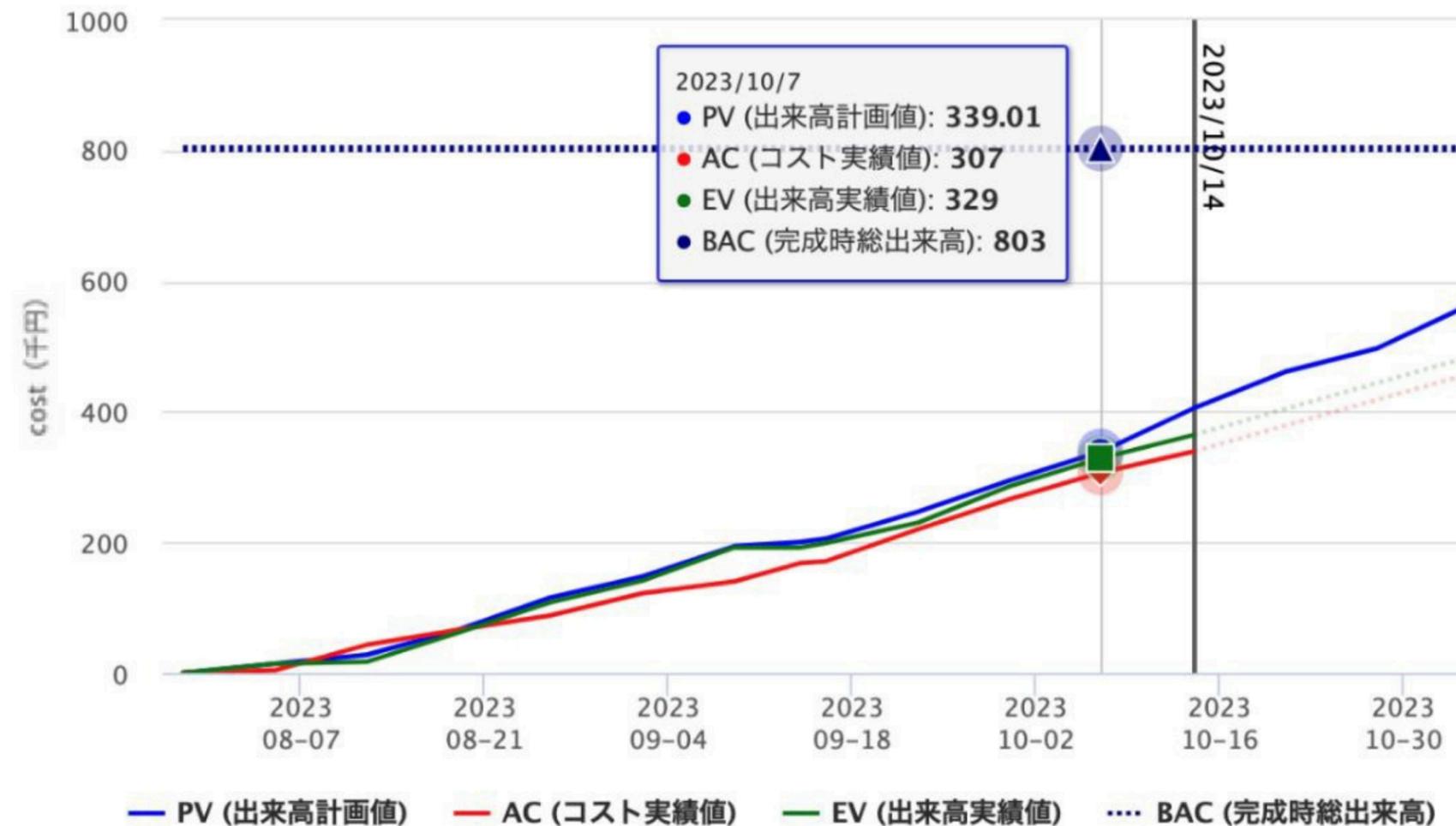


定例会議でEVMを共有する  
プロジェクトレポート機能でEVMを設定しておけば  
週次・月次でスナップショットを自動作成する。

# ▶ EVM機能を活用してコスト・マネジメントを行う

👑 プレミアムプラン・ビジネスプラン

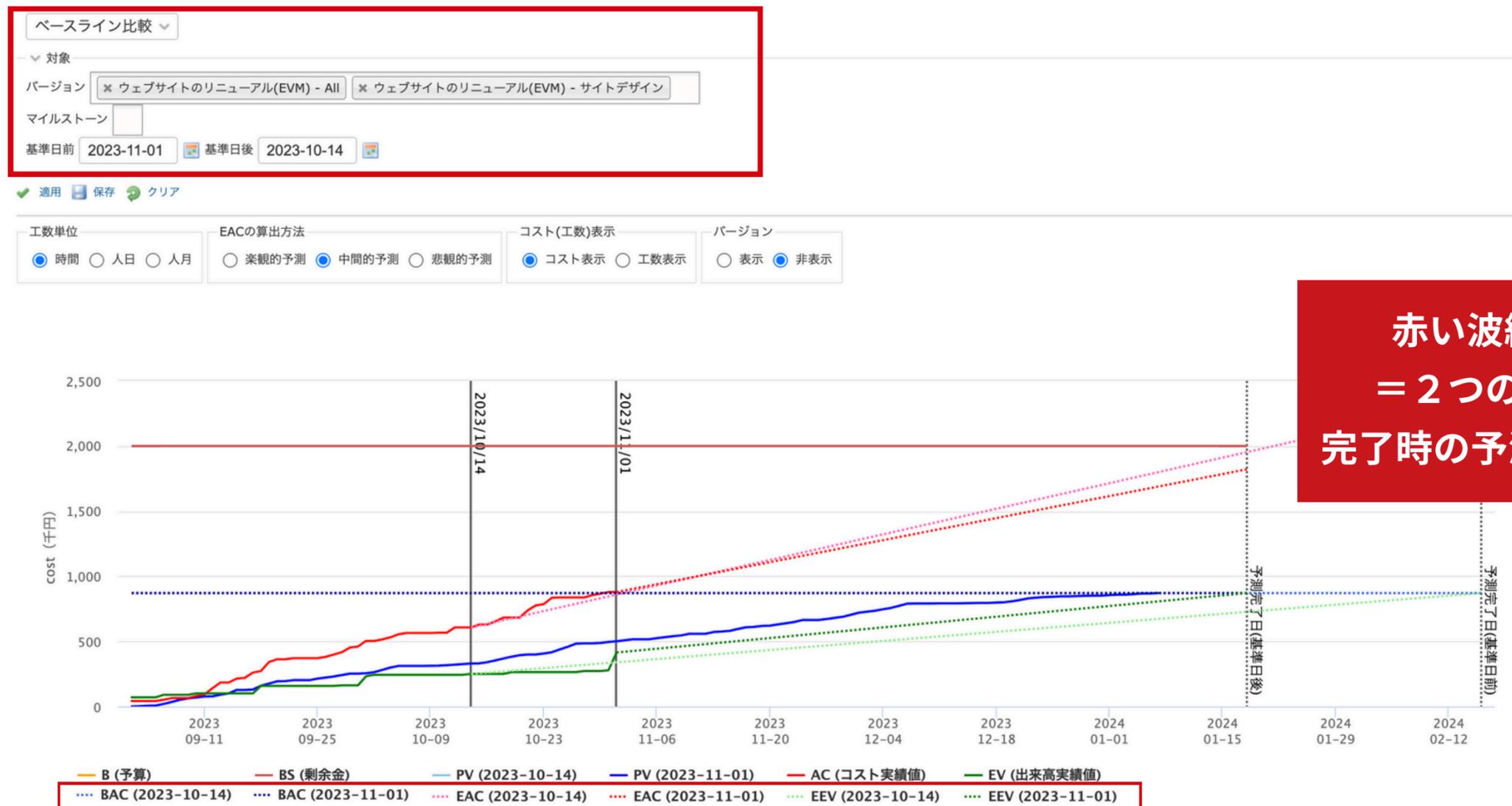
チケットに登録した期日や工数のデータを元に EVM（出来高管理）を集計し計算、グラフにします。  
 総コスト（EAC）が当初予算（BAC）の範囲に収まるか、現在の開発ペースから予測できます。



# ▶ 2つの基準日でコストや生産性を「ベースライン比較」

👑 プレミアムプラン・ビジネスプラン

「ベースライン比較」を使い比較する2つの基準日を入力すると、2つのEVMを重ねて表示します。工程（ベースライン）ごとに進捗・コスト・生産性を比較分析したり、要員増減、開発スコープ変更のタイミング前後で、進捗・コスト・生産性に変化が発生したか否かを分析できます。

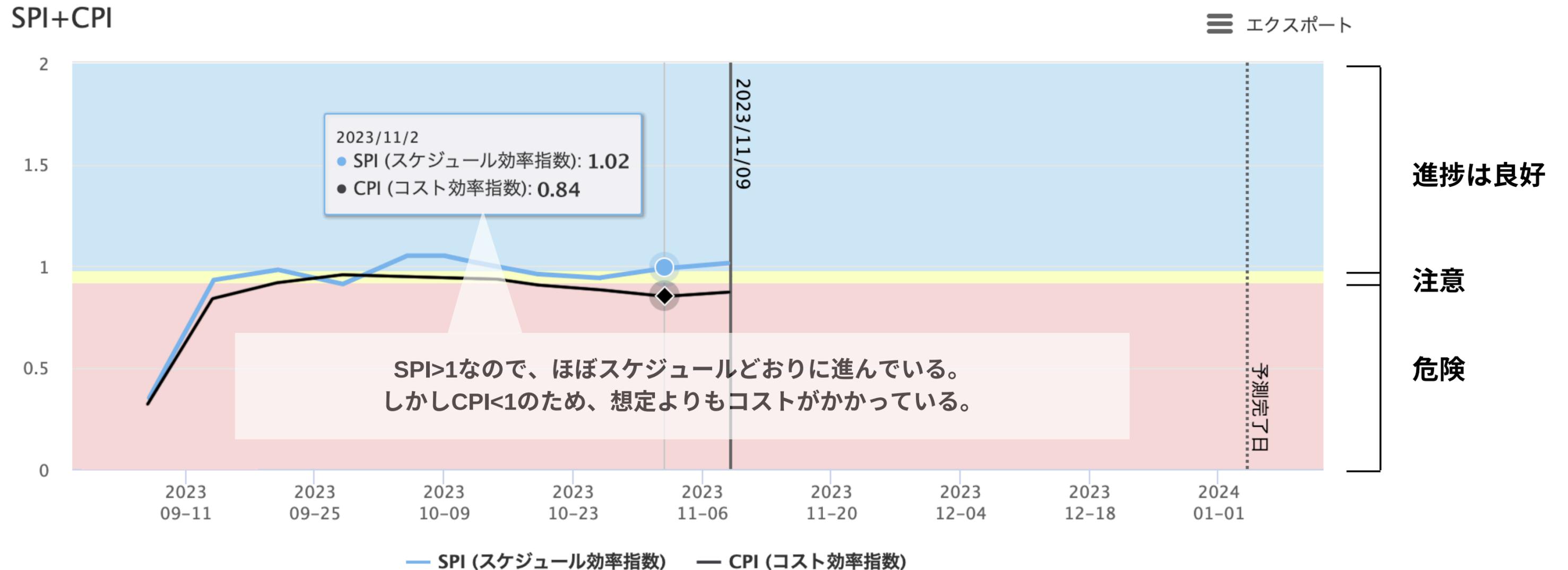


赤い波線とピンクの波線  
= 2つのスケジュールでの  
完了時の予測コスト(EAC)を比較

# ▶ EVMのスケジュール効率指標とコスト効率指標

👑 プレミアムプラン・ビジネスプラン

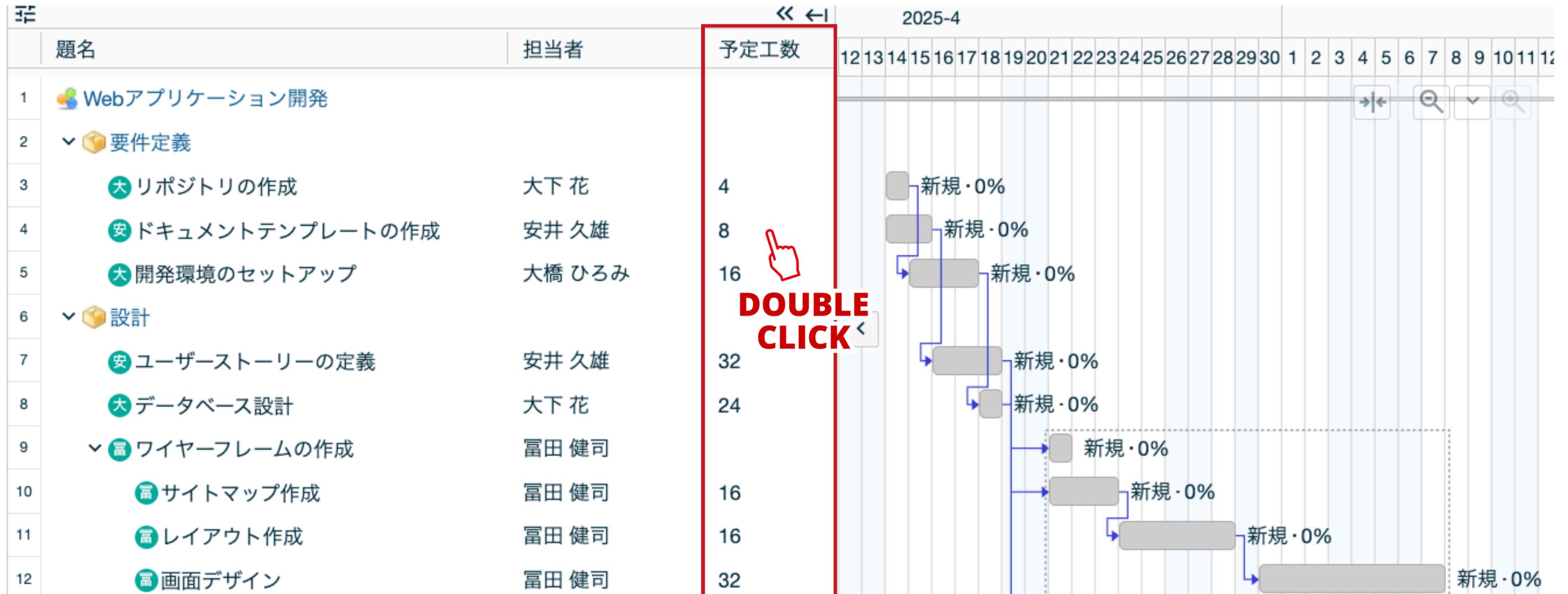
現在の開発ペースから、納期遅延やコスト増のリスクを示す、SPI（スケジュール効率化指数）とCPI（コスト効率化指数）を表示できます。青は進捗が良好、黄色は注意、赤は危険ゾーンです。



## ▶ EVM機能を使うには？

作業の予定工数、作業実績、単価・経費を入力しましょう

ガントチャートで予定工数を入れましょう。ダブルクリックで表計算シートのように編集できます。  
予定工数は、作業日数で割って1日あたりの工数を自動で算出します（例：15時間 ÷ 5日 = 1日3時間）。



# ▶ EVM機能を使うには？

作業の予定工数、作業実績、単価・経費を入力しましょう

担当者は、タイムマネジメント機能の「作業実績」を利用して、作業にかかった時間を入力しましょう。

作業実績 < 2025/03/01 ~ 2025/03/31 > 今日

ソート条件 題名 | 担当者 Select...

+ プロジェクト追加 + タスク追加 + 前月のプロジェクトを引き継ぐ

					2025/03												
題名	ステータス	予定工数	作業分類	合計	1土	2日	3月	4火	5水	6木	7金	8土	9日	10月	11火	12水	13木
<b>すべての作業</b>				83:00			8:00	8:00	8:00	8:00	8:00			8:00	8:00	8:00	8:00
<b>Webアプリケーション開発</b>				83:00			8:00	8:00	8:00	8:00	8:00			8:00	8:00	8:00	8:00
プロジェクト工数				11:00													
#2754	ドキュメントテンプレートの作成	新規	8:00	設計作業	8:00		8:00										
#2755	ユーザーストーリーの定義	新規	32:00	設計作業	34:00			8:00	8:00	8:00	8:00			2:00			
#2752	リポジトリの作成	新規	4:00	設計作業	4:00									4:00			
#2756	ワイヤーフレームの作成	新規															
#2757	ワイヤーフレームの作成 サイトマップ作成	新規	16:00	設計作業	18:00									2:00	8:00	8:00	
#2753	開発環境のセットアップ	新規	16:00	開発作業	8:00												
間接作業																	

**DOUBLE  
CLICK**



## ▶ EVM機能を使うには？

作業の予定工数、作業実績、単価・経費を入力しましょう

「コスト・マネジメント」機能を使えば、メンバーの単価や経費の登録ができるようになります。

### メンバーの単価をグループごとに設定

管理 > 「コストグループ」から設定できます。

#### コストグループ

+ 新しいコストグループ

名称	単価	
エンジニア	2,500	削除
マネージャー	3,000	削除
テスター	1,800	削除
デザイナー	2,000	削除
アルバイト	1,100	削除

### プロジェクトの「経費」を登録

単価以外の経費も入力しましょう。

合計金額: **900,000円**

年月日	名称	金額	説明
2025-03-31	サーバー費用	100,000円	
2025-03-14	UI/UXデザイン費用	500,000円	
2025-03-04	テスト外注費	300,000円	

以上で、EVMを利用する準備が整いました！

04

# 品質 マネジメント

顧客が満足するレベルの成果物を、  
安定して出すための品質管理

## やるべきこと

## Lycheeでの対応とやり方

## 計画

品質基準を定める  
(例：エラー率、納品条件、ユーザビリティなど)



## Wikiに指標・許容値・計測方法を明文化する

プロジェクトレポート機能の「品質指標」で KPI（エラー率、C-SAT など）を登録し、自動集計されるように設定する。

## 計画

チェックリストや  
レビュー基準を事前に準備しておく

チケットの説明欄にレビュー観点を記載したり、  
チェックリストに受入条件を登録する

## 実行

成果物のレビュー・テストを  
計画通りに実施する

マイルストーンを「レビュー」「システムテスト」などで切り、  
期間・担当をガントチャートで可視化する

テスト項目の管理は外部テスト管理ツールを使うこと推奨。

## 監視・コントロール

不具合が出た場合は原因を分析して、  
再発防止策を立てる



## 不具合はバグトラッカーでチケットを起票する

「原因区分」「発生工程」などのカスタムフィールドを用意しておくこと、原因分析に役立てられる。要件と不具合チケットを関連づけして、トレーサビリティ担保にも。

## 監視・コントロール

品質改善のためのフィードバックを  
収集し、プロセスに反映する

CPI・不具合数の推移・レビュー指摘件数などを定点観測する  
ダッシュボードやプロジェクトレポートで定期的に確認する

## ▶ トラッカー「バグ」を活用して品質マネジメントを行う

不具合の早期発見や是正には、「見える化」して継続的に分析・改善することが重要です。  
不具合や問題点に関する情報は、チケットのトラッカー「バグ」「課題」などにして蓄積しましょう。

### 原因や傾向分析のために カスタムフィールド（項目）を活用

例えば、下記のようなカスタムフィールドを作っておけば、  
不具合を定量的に分析しやすくなります。

項目名	選択肢・入力例
不具合分類	仕様漏れ / 実装ミス / テストケース不足 など
影響度	高 / 中 / 低
バグ再現性	常に再現する / 高頻度で再現する / 条件付きで再現する / 再現困難 / 未確認

バグ #2815 未完了

編集

時間を記録



#### アプリが起動しない

赤石 ライチ さんが7分前に追加. 1分未満前に更新.

ステータス: 新規 開始日: 2025-03-31

期日:

実開始日:

実終了日:

不具合分類: 仕様漏れ 影響度: 高

バグ再現割合: 常に再現する

#### 説明

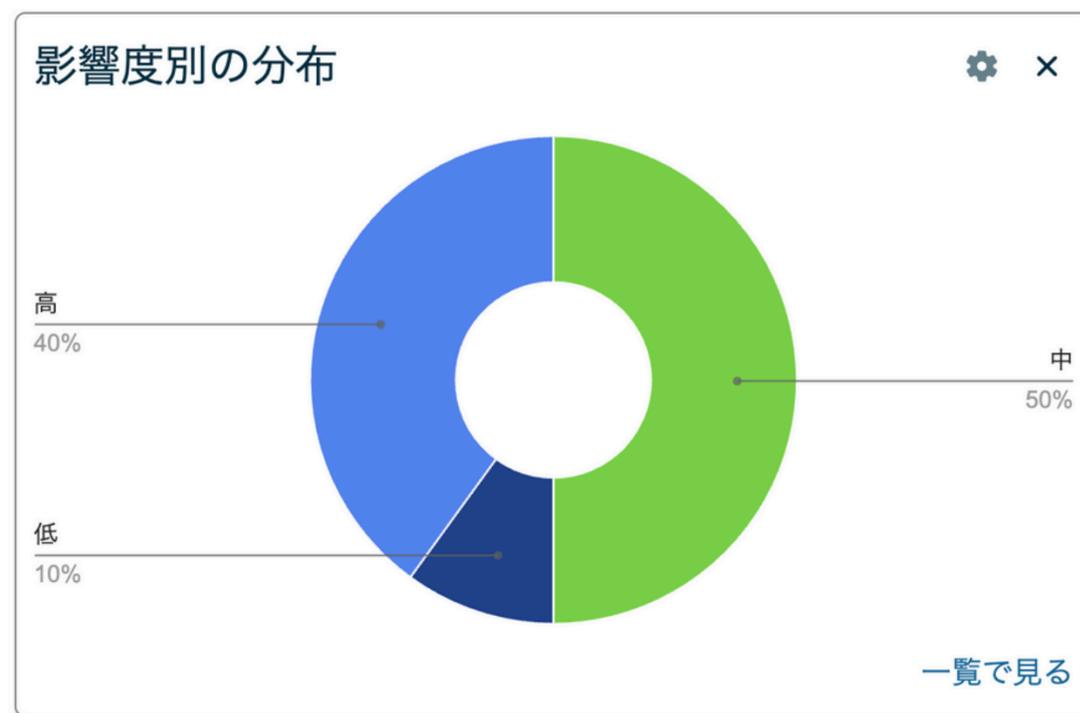
アプリを起動しようとする時、スプラッシュ画面の後にフリーズし、そのまま強制終了され  
特定のOSバージョン（Android 13）で100%再現され、複数ユーザーから報告があります。  
デバッグログによると、初期化時の設定ファイル読み込みに失敗して例外が発生している様

## ▶ ダッシュボード機能を活用して品質マネジメントを行う

ダッシュボードではチケットの情報を元に、今どんなバグが多いのか・どこに問題が集中しているのかをひと目で確認できる、グラフやチケットリストの表示ができます。

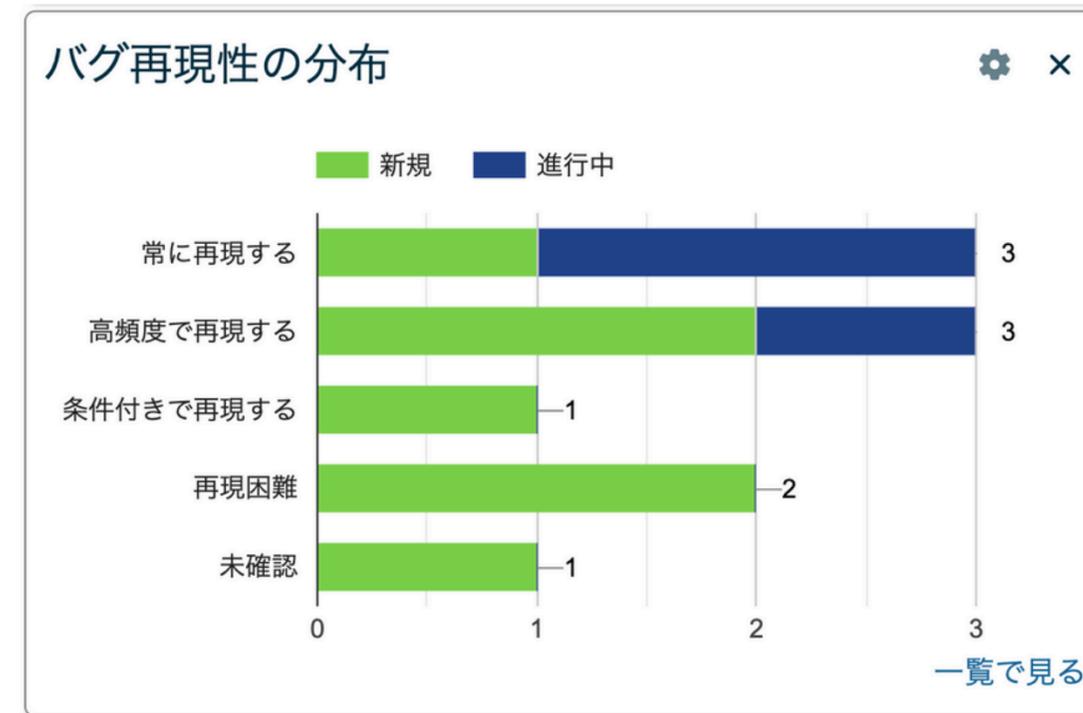
### 影響度別の分布（円グラフ）

影響度「高」に分類されたチケットの割合を可視化し、深刻な影響を与える不具合の比率を把握する目的で使用します。「高」の割合が高い場合、リリース品質や仕様設計に関する見直しが必要と判断できます。



### バグ再現性の分布（棒グラフ）

「常に再現する」「高頻度で再現する」が多い場合、修正優先度が高いバグが多いと判断されます。ステータスを横軸に取れば対応状況も見えます。



## ▶ ダッシュボード機能を活用して品質マネジメントを行う

### 未対応の「高影響 × 常に再現する」(チケットリスト)

未対応チケットの中でも「影響度が高く」「常に再現する」バグの件数を抽出し、優先対応リストとして表示します。

現時点で最も早急に修正すべき不具合を明確化するのに役立ちます。

#### 未対応の「高影響 × 常に再現する」リスト

#	トラッカー	ステータス	題名
2818	バグ	進行中	スクロールが途中で止まる
2816	バグ	進行中	パスワード変更が反映されない
2820	バグ	新規	検索結果が表示されない

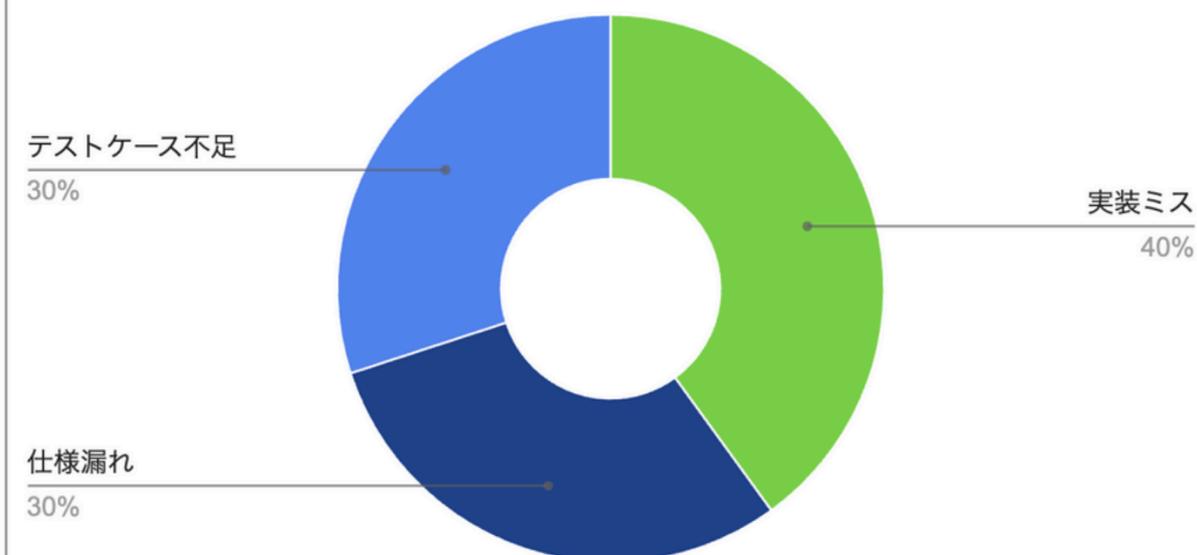
一覧で見る

### 不具合分類ごとの割合 (円グラフ)

プロセス改善を検討する際に役立ちます。

「仕様漏れ」が多い場合は、要件定義やレビュー工程に課題がある可能性があり、「テストケース不足」が多い場合は、テスト設計の見直しや網羅性の強化が必要です。

#### 不具合分類ごとの割合



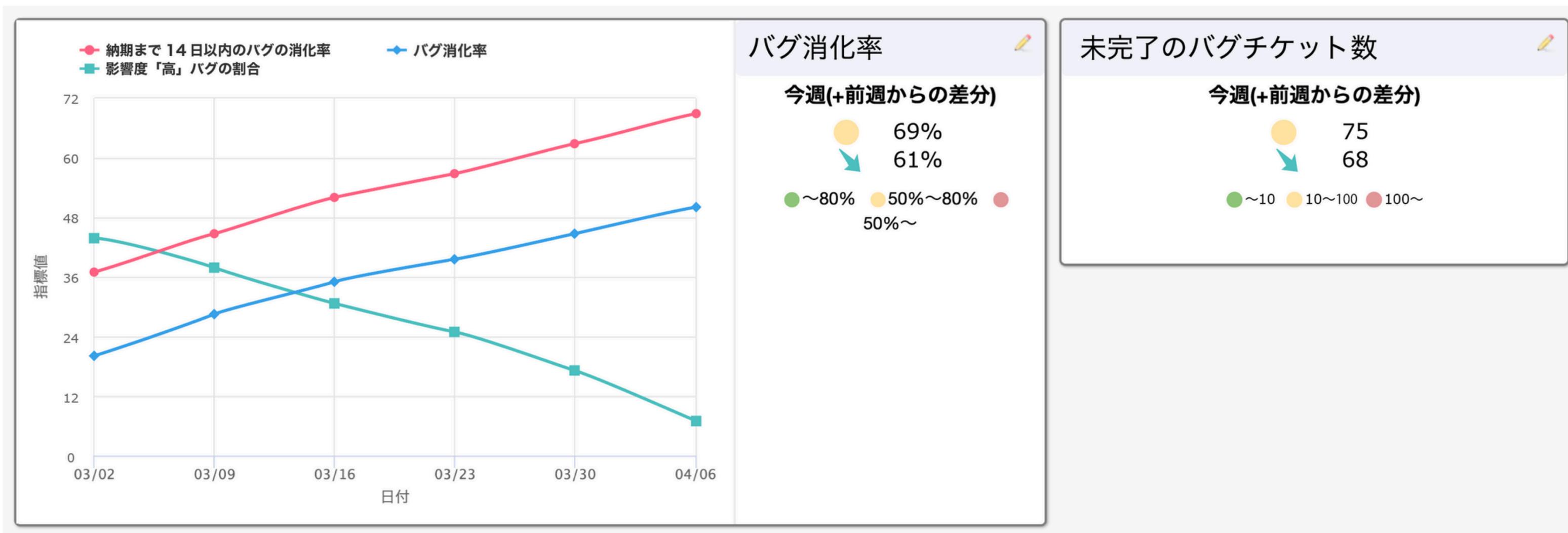
一覧で見る

# ▶ プロジェクトレポート機能を活用して品質マネジメントを行う



ビジネスプラン

プロジェクトレポートでは、「時系列での推移」や「目標とのギャップ」にフォーカスして、プロジェクト全体の品質状態・リスク・対応状況の定点観測に使えます。



# ▶ プロジェクトレポート機能を活用して品質マネジメントを行う

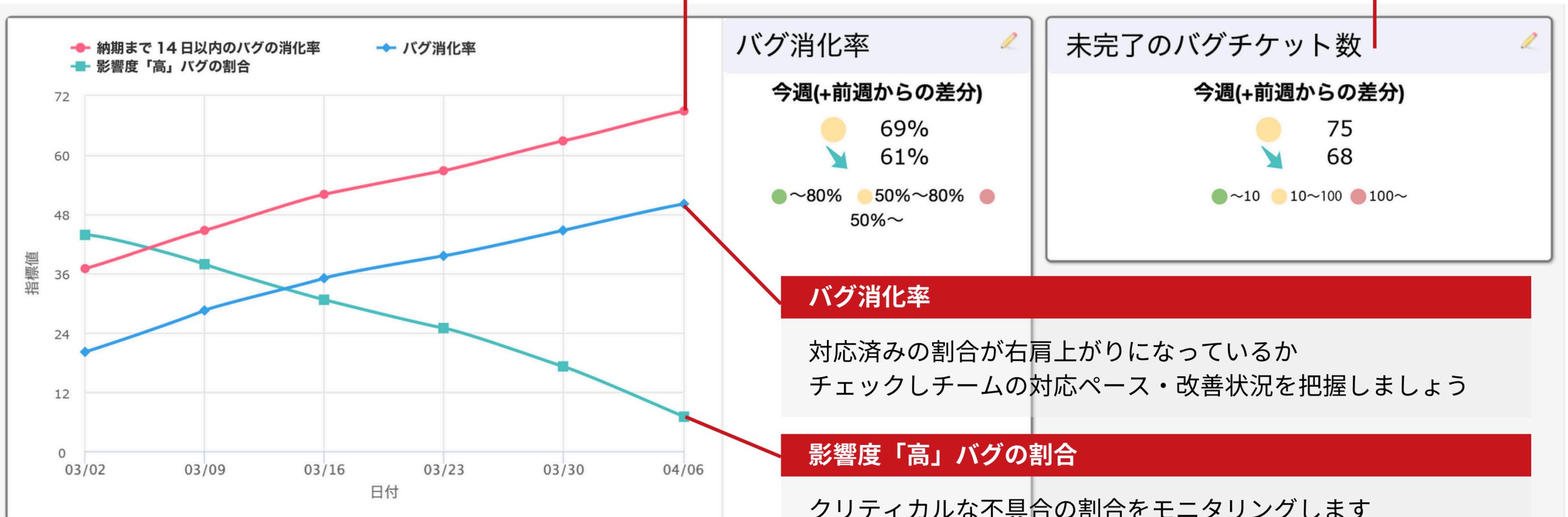
📁 ビジネスプラン

## 納期まで14日以内のバグの消化率

期日が近いバグの消化率はリリース直前のリスクがわかります  
比率が低ければ人員を増やすなどの対処が必要です

## 未完了のバグチケット数

先週比較で消化状況がわかります



### バグ消化率

対応済みの割合が右肩上がりになっているか  
チェックしチームの対応ペース・改善状況を把握しましょう

### 影響度「高」バグの割合

クリティカルな不具合の割合をモニタリングします  
優先対応できていれば、時間の経過につれて下がります

## ▶ プロジェクトレポートを使うための指標設定

### バグ消化率

プロジェクトレポートでは、指標（計算式の定義）と閾値を設定しておくことで、自動的に計算＋信号表示をします。

今回の計算式は

$$\frac{\text{完了済みのバグチケット数}}{\text{バグチケット総数}}$$

消化率なので指標が大きいほど良く

閾値は

～50%

50～80%

80%～

名称\* バグ消化率

説明

指標種別 品質指標

チケットフィルタ(分子) 追加したいフィルタを選択してください

チケットフィルタ(分母)  分子と同じ条件にする  個別の設定をする  
追加したいフィルタを選択してください

指標テンプレート バグトラッカー消化率

選択されたトラッカーの総チケット数に対し、完了したチケット数の割合

指標値判定 大きいほど良い

単位 %

指標値定義\*

- 80.0 %以上
- 50.0 %以上 80.0 %未満
- 50.0 %未満

## ▶ プロジェクトレポートを使うための指標設定

### 納期まで14日以内のバグの消化率

計算式は

$$\frac{\text{期日が14日以内の完了済みバグチケット数}}{\text{期日が14日以内のバグチケット総数}}$$

消化率なので指標が大きいほど良く

閾値は

～50%

50～80%

80%～

名称\*

説明

指標種別

チケットフィルタ(分子)

期日   ✕

チケットフィルタ(分母)  分子と同じ条件にする  個別の設定をする

指標テンプレート

選択されたトラッカーの総チケット数に対し、完了したチケット数の割合

指標値判定

単位 %

指標値定義\*

- 80.0 %以上
- 50.0 %以上 80.0 %未満
- 50.0 %未満

## ▶ プロジェクトレポートを使うための指標設定

### 影響度「高」バグの割合

計算式は

$$\frac{\text{未完了かつ影響度「高」のバグチケット数}}{\text{バグチケット総数}}$$

未完了の割合なので指標が小さいほど良く

閾値は

～30%

30～50%

50%～

名称\* 影響度「高」バグの割合

指標種別 品質指標

チケットフィルタ(分子) 追加したいフィルタを選択してください

トラッカー 等しい

バグ

影響度 等しい

高

チケットフィルタ(分母)  分子と同じ条件にする  個別の設定をする

追加したいフィルタを選択してください

トラッカー 等しい

バグ

指標テンプレート カスタム

指定されたカスタムフィールドの数値の割合で判定する指標です。  $\frac{\text{分子のカスタムフィールドの総和}}{\text{分母のカスタムフィールドの総和} \times \text{営業日数経過率}}$

分子 未完了のチケット数

分母 総チケット数

営業日数経過率の考慮 しない

指標値判定 大きいほど良い

単位 %

指標値定義\*

- 30.0 %以上
- 10.0 %以上 30.0 %未満
- 10.0 %未満

## ▶ プロジェクトレポートを使うための指標設定

### 未完了のバグチケット数

計算式は

$$\frac{\text{未完了のバグチケット数}}{1}$$

1

未完了の数なので指標が小さいほど良く

閾値は

～10

10～100

100～

名称\* 未完了のバグチケット数

指標種別 品質指標

チケットフィルタ(分子) 追加したいフィルタを選択してください

トラッカー 等しい バグ

チケットフィルタ(分母)  分子と同じ条件にする  個別の設定をする

指標テンプレート カスタム

指定されたカスタムフィールドの数値の割合で判定する指標です。  
分子のカスタムフィールド / 分母のカスタムフィールドの総和

分子 未完了のチケット数

分母 1

営業日数経過率の考慮 しない

親子チケットを計算に含める

指標値判定 小さいほど良い

単位 数値

指標値定義\*

- 100 以上
- 10 以上 100 未満
- 10 未満

## ▶ プロジェクトを横断して指標を確認する

複数のプロジェクトを統括している方（例：PMOや品質保証部門）には、全体レポートで指標を確認していただくことも可能です。設定した閾値をベースに信号表示で状況がわかるので、赤●や黄色●のところは注意して確認しましょう。

プロジェクト	進捗	テスト密度 (数 / KLOG)	バグ密度 (数 / KLOG)	バグ件数	バグ放置件数 (5日以上)	マイルストーン					
						3月	4月	5月	6月	7月	8月
▼ ABC 商事	101% ●	67 ●	3 ●	61 ●	11 ●						
● 販売管理システム開発	108% ●	150 ●	6 ●	35 ●	7 ●		設計		試験		
● 営業支援システム開発	100% ●	50 ●	3 ●	18 ●	3 ●		設計	▼工程会議	試験	▼受入検査	
● 問合せ管理システム開発	91% ●	20 ●	1 ●	8 ●	1 ●			▼設計会議 バージョン1		▼設計会議 バージョン2	
▶ ● 顧客管理システム開発	86% ●	120 ●	5 ●	32 ●	10 ●		▼工程会議 バージョン1		▼受入検査	▼工程会議	▼受入
									バージョン2		

05

# 資源 マネジメント

人・モノ・設備など必要なリソースを  
計画的に用意・配分する管理

## やるべきこと

## Lycheeでの対応とやり方

### 計画

プロジェクトに必要な人的・物的リソースを特定し、入手計画を立てる



チケットに予定工数を入力することで人的リソースは可視化できる  
物的リソースは管理できない。

### 計画

チームの体制や役割を定義し、責任分担を明確にする



1. チーム体制をWikiにまとめる
2. ロールと権限機能で役割を作成し権限範囲を制御する
3. チケットには担当者をアサインして、誰がどのチケットを担当しているかを明確にする

### 実行

チームメンバーの動機づけやトラブル解消に対応する



対応していない

### 実行

チームのスキルやパフォーマンスを向上させる取り組み（育成・サポート）を行う



対応していない

### 監視・コントロール

メンバーの作業負荷を調整し、偏りがないかを管理する



1. ガントチャートの工数見える化機能で稼働時間を超える負荷を調整する（計画の調整）
2. リソースマネジメント機能で担当者別の稼働率と生産性を確認する（実績の確認） ※プレミアムプラン以上の契約が必要です。

# ▶ リソースマネジメント機能で資源マネジメントを行う

👑 プレミアムプラン・ビジネスプラン

人的資源（＝チームメンバーの作業負荷や稼働状況）に関するマネジメントが可能です。  
一方で、物的資源の管理やメンバー育成といった領域は対象外となります。

各メンバーに対して「予定工数」と「実績工数（作業実績）」を登録・可視化することで、稼働状況の過不足を把握したり、プロジェクト全体の工数計画を最適化するのに役立ちます。

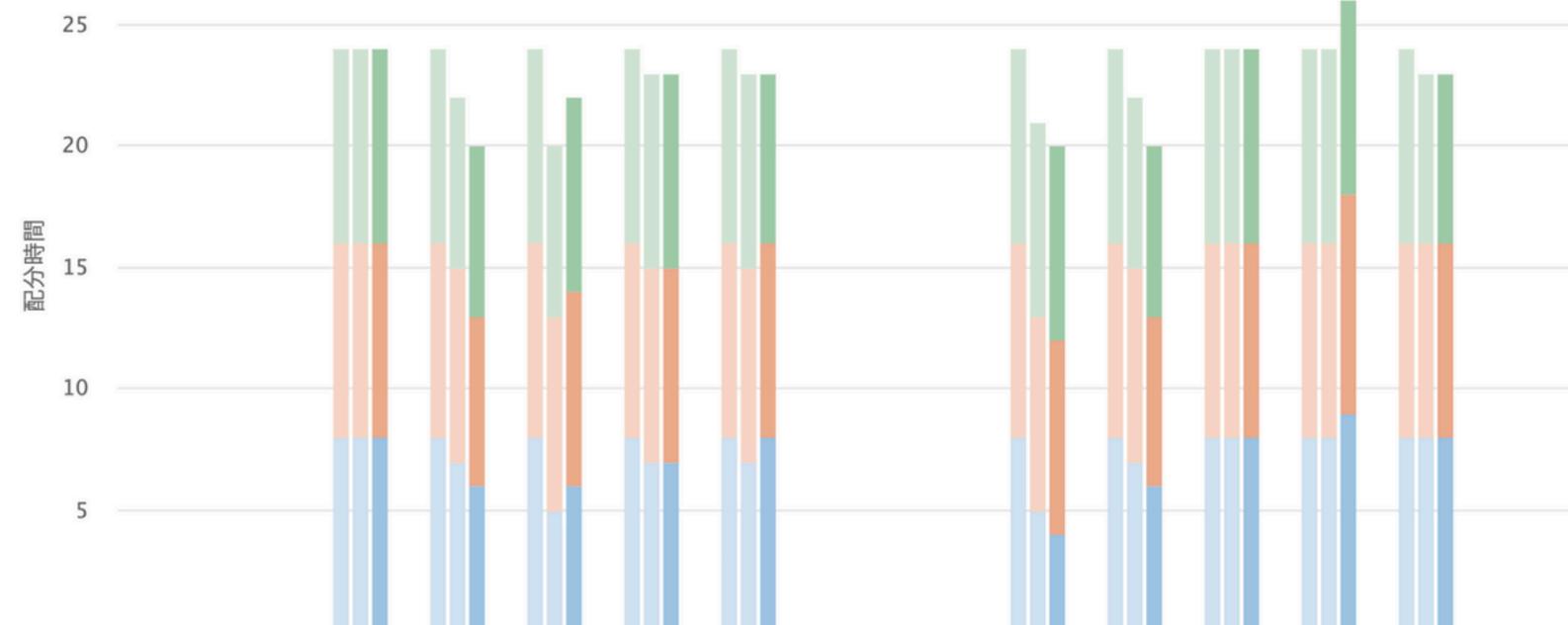
## 工数タブで予定と実績を比較しよう

### ■計画時

予工（予定工数）と予作（作業予定）を比較し、差が大きければ調整対象としてチェックします。

### ■実行中

予工と実作（作業実績）を比較すると、進捗が思ったより進んでいない／やりすぎている、などが明らかになります。



ユーザー		3/1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
大下花	予工			8.00	8.00	8.00	8.00	8.00			8.00	8.00	8.00	8.00	8.00	
	予作			8.00	7.00	7.00	8.00	8.00			8.00	7.00	8.00	8.00	7.00	
	実作			8.00	7.00	8.00	8.00	7.00								
富田健司	予工										8.00	8.00	8.00	8.00	8.00	
	予作										8.00	8.00	8.00	8.00	8.00	
	実作			8.00	7.00	8.00	8.00				8.00	7.00	8.00	9.00	8.00	
佐藤公平	予工			8.00	8.00	8.00	8.00	8.00			8.00	8.00	8.00	8.00	8.00	
	予作			8.00	7.00	5.00	7.00	7.00			5.00	7.00	8.00	8.00	8.00	
	実作			8.00	6.00	6.00	7.00	8.00			4.00	6.00	8.00	9.00	8.00	

予工と予作の差が大きい

実作が大きい＝遅れている？

# ▶ リソースマネジメント機能で資源マネジメントを行う

👑 プレミアムプラン・ビジネスプラン



担当者	3/1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
大下 花			100.00	87.50	100.00	100.00	87.50					100.00	100.00	87.50	
富田 健司			100.00	87.50	100.00	100.00	100.00					100.00	112.50	100.00	
佐藤 公平			100.00	75.00	75.00	87.50	100.00					50.00	75.00	100.00	112.50
平均			100.00	83.30	91.70	95.80	95.80					83.30	83.30	100.00	108.30

手が空いている

作業負荷が高い

## 稼働率タブでチームの健全性を把握する

稼働率を定期的を確認することで、誰に負荷が偏っているか、計画と現実のギャップを把握しやすくなります。

稼働率 = 予工 ÷ 実作 で計算されます。

- 100%に近い  
予定通り作業が進んでいる
- 100%を超えている  
作業が詰まりすぎている・想定以上に負荷がかかっている可能性がある
- 50%を下回る  
手が空いている、または実績入力漏れかもしれない

## ▶ リソースマネジメント機能を使うには？

開始日・期日、予定工数、作業予定、作業実績を入力しましょう

開始日・期日と予定工数はガントチャートで、作業実績はタイムマネジメント機能で入力しましょう。(p18-19参照)

自分が担当しているチケットの開始日・期日・予定工数を元に、「作業予定」が配置されます。

担当者は、自分のスケジュールを鑑みながら、作業予定を組み立てます。

計画段階で1日の稼働時間を超えた無理な計画を修正することができます。

作業予定 1日の稼働時間: 8:00

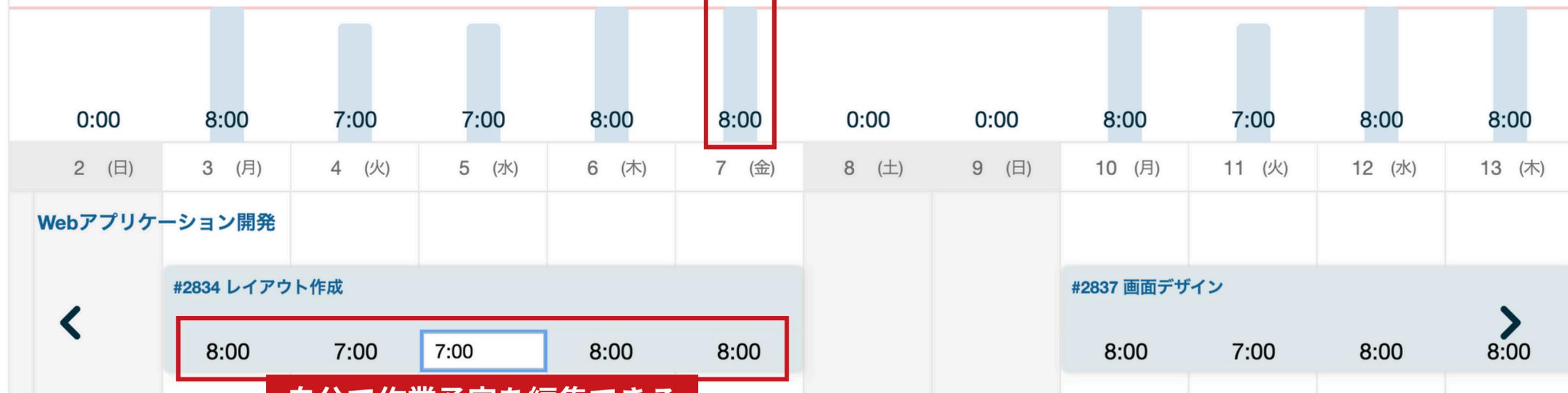
今日

担当者

大下 花

2025年03月

1日の稼働時間を超えないように



自分で作業予定を編集できる



06

# コミュニケーション マネジメント

会議や記録、法令もふまえて  
関係者に円滑に情報を伝え合うための仕組み

## やるべきこと

## Lycheeでの対応とやり方

## 計画

誰に、何を、いつ、どの手段で伝えるかを整理したコミュニケーション計画を作る



## Wikiに、関係者・頻度・手段を公開し通知

「遅延／障害時の連絡先・期限」やエスカレーション先を記載する。公開範囲が限られた情報は、ロールと権限で閲覧範囲を制御する。

## 実行

定例会議・報告書・チャットで情報共有を行う



## 「定例会議」チケットを作成し、議事録をコメントで追記していく

関係者をウォッチャに設定して共有する。

## 実行

ステークホルダーごとに適切なレベルで報告する



## プロジェクトレポートで報告テンプレートを作成しておき、定期的に共有する

## 実行

情報の誤解や抜け漏れがないよう、記録（議事録・メモ）を残す



## 決定事項は必ずチケット or Wikiに残すようメンバーに徹底する

## 監視・コントロール

フィードバックを受けたら、内容を確認して必要に応じて対応する



## チケットのコメントにフィードバックを集約する

# ▶ Wiki機能でコミュニケーション・マネジメントを行う

プロジェクトルール、仕様情報、会議の議事録などの情報を一元管理できます。関係者全員が常に最新情報にアクセスでき、口頭やチャットでの伝達漏れを防げます。編集履歴が自動で残るため、誰がいつ何を更新したかも追跡可能です。通知機能やリンク共有を併用すれば、関係者との情報共有がよりスムーズに行え、プロジェクト内のコミュニケーションの透明性と効率が向上します。

## 自動販売機開発プロジェクト憲章

プロジェクトの全体的な概要を記載し、プロジェクトの方向性、目的、役割、責任を明確にする文書です。この項目を適切に埋めることで、プロジェクト全体の方向性を明確にし、関係者間の共通理解を促進できます。また、プロジェクトの成功に向けた重要なガイドラインとなります。

### 1. プロジェクトの基本情報

- プロジェクト名：プロジェクトを特定できる正式な名称。
- プロジェクト開始日と終了日：スケジュールの明確化。
- 作成日：この憲章を作成した日付。

### 2. 背景・目的

- プロジェクトの背景：プロジェクトが立ち上がった理由や課題。
- 目的と目標：達成したい具体的な成果や成功基準。
- 戦略的価値：プロジェクトが組織や顧客にどのように貢献するのか。

### 3. スコープ（範囲）

- プロジェクトに含まれる内容：実施する作業や対象。
- 除外される内容：プロジェクトの範囲外とする事項を明確化。

### 7. リスクと制約

- 想定されるリスク：プロジェクト成功の妨げとなる可能性がある要因。
- 制約条件：時間、予算、スコープなどの制約。

### 8. プロジェクト管理計画

- 進捗管理方法：報告の頻度や形式。
- 変更管理：プロジェクトに変更が生じた場合の対処方法。
- 品質管理：成果物の基準やレビュー体制。

### 9. 承認と署名

- プロジェクト憲章を承認する主要なステークホルダーの署名欄。

### 10. 参考資料・補足情報

- 関連文書：提案書やビジネスケースなどのリンク。
- 用語集：専門用語や略語の説明。

### 11. 会議体制

- デイリーミーティング：チーム内での状況を毎朝15分確認の時間を設ける。場所はチームに任せる。など
- 月例ミーティング：毎月の報告を実施する。など

## ▶ チケットの「コメント」でコミュニケーション・マネジメントを行う

状況を報告する際にはチケットのコメント機能を使ってコミュニケーションを円滑に進める事ができます。

### コメントで日々の状況報告

コメントは日報の代わりにもなります。作業内容や変更内容を残しておけば、他のメンバーも検索して当時の状況を知ったり、類似案件などの解決策検討のヒントを得ることができます。

履歴 コメント プロパティ更新履歴

赤石 ライチさんが約11時間前に更新・編集済み

- 残工数(Lychee) を 4.0 にセット

動作確認をお願いします

林 美琴さんが約10時間前に更新・編集済み

動作確認環境は下記でよろしいですか？  
動作検証サーバ10

Redmine 検索: 環境 プロジェクトへ移動...

プロジェクト 活動 チケット 作業時間 ガントチャート5.0 [プレα] ガントチャート チャート ニュース

検索

環境 全プロジェクト での単語 □ タイトルのみ

✓

チケット  ニュース  文書  更新履歴  Wikiページ  メッセージ  プロジェクト

オプション

検索

結果 (2) チケット (2)

3チケットのフィルタを適用

- 互換性テスト- サポート #3202 (新規): Lycheeシリーズの影響テスト  
他のLycheeシリーズに影響がないか確認する  
2025/06/10 07:43
- 商品管理システム(Kanban)- 機能 #2487 (新規): 開発環境のDocker化  
2025/06/10 07:36

## ▶ 「メッセージBOX」でコミュニケーション・マネジメントを行う

チケットへの「コメント」をチャット形式でやり取りができます。

### メッセージBOX

#3200 2025/06/10 07:58 ECサイト構築 >> 互換性テスト  
Redmine2.5の動作確認  
進捗率を 60 から 90 に変更

#3202 2025/06/10 07:58 Lycheeシリーズの影響テスト  
関連している バグ #3201: Ruby 2.3...

#3114 2025/06/10 07:57 画面仕様書

#3066 2025/06/10 07:57 商品ディスプレイ制御

#3039 2025/06/10 07:57 製品保管・ディス Pens 機構

#2489 2025/06/10 07:56 データ修正ツールの作成

#2488 2025/06/10 07:56 ● 外部公開APIの作成

#2487 2025/06/10 07:56

2025/06/10 07:43  
残工数(Lychee)を 4.0 にセット  
動作確認をお願いします

林 美琴 さん 2025/06/10 07:48  
林 動作確認環境は下記でよろしいですか？  
動作検証サーバ 10

2025/06/10 07:53  
そこで検証してもらって問題ないです。

林 美琴 さん 2025/06/10 07:58  
林 進捗率を 40 から 80 に変更

林 美琴 さん 2025/06/10 08:03  
林 他のLycheeシリーズへの影響を確認しました。  
問題があったので別のチケットを作成します。

ここにコメントを入力... 送信

### チケットのコメント

赤石 ライチさんが約11時間前に更新・編集済み #1  
• 残工数(Lychee)を 4.0 にセット  
動作確認をお願いします

林 林 美琴さんが約10時間前に更新・編集済み #2  
林 動作確認環境は下記でよろしいですか？  
動作検証サーバ 10

更新内容はコメントにも反映

07

# リスク マネジメント

起こりうるリスクを見つけて分析し、  
備えや対応を実行・監視するための仕組み

やるべきこと

Lycheeでの対応とやり方

計画

リスク・マネジメント計画を作る  
(目的・役割・評価基準・レビュー頻度を定義)



Wikiにリスク計画の目的・担当・レビュー日を記載し、更新時は全員へ通知する

計画

プロジェクトの不確実性やリスクを洗い出し、リスク登録簿を作る



リスク管理用の専用プロジェクトを作り、洗い出したリスクをチケット起票する

計画

リスクが実際に発生したときの対応フロー  
(トリガーとアクション)を決めておく



リスクトラッカーを作成し、ワークフローで「検知→対応中→完了」のステータスを設定する

計画

各リスクの「発生確率」と「影響度」を評価して、優先度を決める



リスクチケットに「発生確率 (%)・影響度 (5段階)」項目を設定し、カンバンで優先順に並び替える

計画

リスクの対応策を決定する



リスクチケットに「回避／軽減／転嫁／受容」項目を入力し、対応タスクを子チケット化する

監視・コントロール

リスクの状況を定期的に見直し、新たなリスクの洗い出しを行う



「未対応リスク」一覧をダッシュボードに表示させ、定例会議で確認する

## ▶ 専用プロジェクトを作成しリスク・マネジメントを行う

リスク管理用のプロジェクトを「専用プロジェクト」として立ち上げ、体系的に管理しましょう。  
Redmineはもともとバグや課題管理に強みがあるため、過去の不具合や障害をベースに、リスクの傾向分析を行いやすいという利点があります。

### ▼ ブックマークプロジェクト

Webアプリケーション開発 

Webアプリケーション開発\_リスク管理用 

専用プロジェクトを作成

DR指摘期日超過ランク割合



✓適用  クリア  カスタムクエリを

<input type="checkbox"/>	#	トラッカー	ステータス	優先度	題名	担当者	更新日	
<input type="checkbox"/>	2522	リスク	新規	通常	利用者に検索できない	安西 博満	2025/06/10 07:36	...
<input type="checkbox"/>	2519	リスク	進行中	通常	在庫管理画面	安西 博満	2025/06/10 07:36	...
<input type="checkbox"/>	2492	リスク	解決	通常	ユーザ権限管理	宮本 吉之助	2025/06/10 07:36	...
<input type="checkbox"/>	2491	リスク	進行中	通常	OSSライブラリ修正	富田 健司	2025/06/10 07:36	...
<input type="checkbox"/>	2489	リスク	フィードバック	通常	データ修正ツールの作成	赤石 ライチ	2025/06/10 07:36	...
<input type="checkbox"/>	2486	リスク	進行中	通常	レスポンスデザイン	宮本 吉之助	2025/06/10 07:36	...
<input type="checkbox"/>	2478	リスク	進行中	通常	商品ページ速度改善	赤石 ライチ	2025/06/10 07:36	...

(1-7/7)

特定トラッカーを管理して分析しやすく

## ▶ リスクトラッカーでリスク管理を行う

トラッカーも専用のものを作り、カスタムフィールドやチケットテンプレートを設定しましょう。

「影響度」「発生率」「原因」などのカスタムフィールドを設定しておくことで、蓄積データをもとに分析が可能になります。

説明欄には、「リスク詳細」「トリガーポイント」「予防対策」「発生時の対策」など、リスク管理に必要な項目を記載できるように準備しておきます。



### 同一書籍の重複登録によるデータ肥大化の懸念

赤石 ライチ さんが2日前に追加. 1分未満前に更新.

ステータス:	新規	開始日:	
担当者:	-	期日:	
		予定工数:	
影響度:	中	リスクの原因:	仕様漏れ
発生率:	中		

#### 説明

#### リスク詳細

現在の書籍登録機能では、ユーザーがISBNコードを用いて検索・登録する形式を採用しているが、同一ISBNの書籍であっても重複登録されてしまう恐れがある。複数ユーザーによる登録やUIの導線ミスなどにより、DBに同一データが複数存在することが懸念される。このまま放置すると、データ肥大化、UXの低下、統計の精度劣化などの影響が出る可能性がある。

#### トリガーポイント

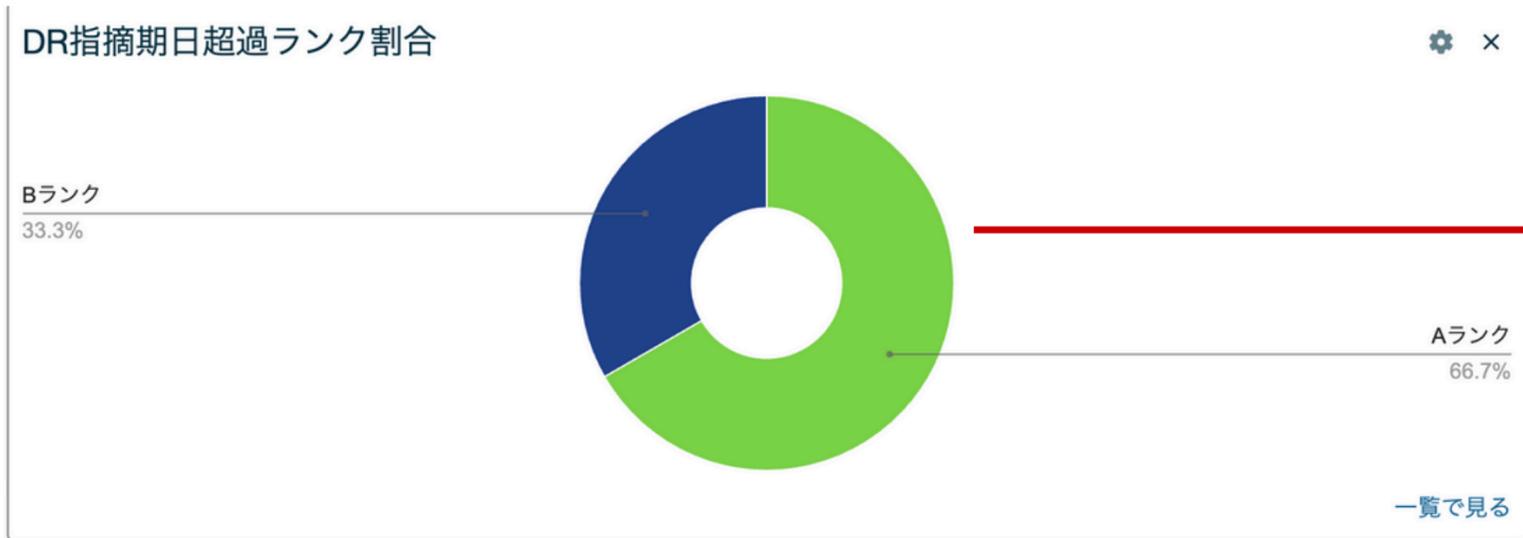
- 書籍一覧に同じタイトルのデータが複数登録されている
- ユーザーから「同じ本が何冊も出てくる」との問い合わせ
- 登録書籍数が異常に増加している（例：1日100件超）

#### 予防対策

- 書籍登録時にISBNの重複チェックを実装
- 既存書籍が存在する場合は「既に登録済み」のUI表示を行う

## ▶ ダッシュボードでリスクを見える化する

カスタムクエリでリスク項目を抽出・分類すれば、対応漏れの検知や重大リスクの優先対応も簡単に行えます。このクエリをもとにして「ダッシュボード」機能を使えばリスクの見える化ができます。



### 指摘事項の修正期日が超過した割合

レビュー指摘チケットで、修正期日を超過したものをランク毎（Aが最上位）に円グラフで割合を可視化。

高ランク指摘が残っている＝大きな課題が残っており早急に対処が必要。

DR指摘期日超過一覧

#	プロジェクト	トラッカー	ステータス	題名	担当者	フェーズ	期日	DR指摘ランク
2198	プロジェクトAA	DR指摘	確認待ち	CCCCCC	河崎 美雨	DR2	2024/09/23	Aランク
2196	プロジェクトAA	DR指摘	進行中	AAAAA	河崎 美雨	DR2	2024/09/13	Aランク
2197	プロジェクトAA	DR指摘	進行中	BBBBBB	河崎 美雨	DR2	2024/09/19	Bランク

一覧で見る

### 内容も確認できるようチケット一覧も併記

## ▶ プロジェクトレポートでリスク状況を見える化する

「指標」を作れば、「プロジェクトレポート」機能において分析・早期警告が実現できます。

プロジェクト	進捗	テスト密度 (数/KLOG)	バグ密度 (数/KLOG)	バグ件数	バグ放置 件数 (5日以上)	マイルストーン					
						3月	4月	5月	6月	7月	8月
▼ ABC 商事	101%	67	3	61	11						
販売管理システム開発	108%	150	6	35	7		設計		試験		
営業支援システム開発	100%	50	3	18	3		設計	▼工程会議	試験	▼受入検査	
問合せ管理システム開発	91%	20	1	8	1			▼設計会議	バージョン1	バージョン2	▼設計会議
▶ 顧客管理システム開発	86%	120	5	32	10		▼工程会議	バージョン1	バージョン2	▼受入検査	▼工程会議
											▼受入

### 進捗の遅れがひと目で分かる

プロジェクトのチケットの進捗の割合を指標とすると、プロジェクトの進捗具合がひと目で分かり、リスク状況が見える化されます。

08

# 調達 マネジメント

外部業者との契約や購入など、  
外注や物品の調達を適切に進める管理

**Lychee Redmineは、調達（購買・契約）をワークフロー化／財務連携まで行う領域は得意ではありません。**

大まかな調達の進捗状況をチケット化し共有する程度には使えますが、詳細は専用ツールのご利用をおすすめします。

## やるべきこと

## Lycheeでの対応とやり方

### 計画

外注やツール購入など、プロジェクトに必要な外部調達の範囲を定める



対応していない

### 計画

調達先（ベンダー）の選定基準や契約方法（固定価格、出来高制など）を決める



対応していない

### 計画

契約書の作成や納期・品質条件などを明確に定義する



対応していない

### 実行

ベンダーとの連絡・調整・トラブル対応を行う



対応していない

### 監視・コントロール

納品物の検収・支払い処理を行い、契約通りかを確認する



対応していない

09

# ステークホルダー マネジメント

利害関係者（顧客・上司・関係部署など）を洗い出し、  
関与度合いを調整し協力を引き出す仕組み

Lychee Redmineの利用はタスクとスケジュールの可視化といった行動管理に限定し、ステークホルダー登録簿や満足度の集計など情報管理はスプレッドシートや専用ツールに任せることをおすすめします。

## やるべきこと

## Lycheeでの対応とやり方

立ち上げ

ステークホルダー（利害関係者）を洗い出し、影響度や関心度をマッピング



対応していない

計画

各ステークホルダーが求めている情報や成果を把握する



対応していない

計画

関心や懸念に合わせた対応を計画する（説明会、個別フォローなど）



対応していない

報告にはチケットやガントチャートの情報を利用できる。

監視・コントロール

要求や期待値に対する満足度を継続的に確認し、ギャップが出ないように調整する



タスク化できない粒度のものは対応していない

「UI デザイン」「性能」「納期」などはチケットで管理できる。

監視・コントロール

意見・要望をプロジェクトに適切に反映させる



要望チケットを作成し、承認フローを経て作業化する

要望と作業チケットは関連リンクさせておく。

## ▶ Wiki機能でステークホルダー・マネジメントを行う

Wiki機能を活用することで、ステークホルダーとの共通認識を整理・共有できます。  
また、チーム全体が誰に何を伝えるべきかを把握しやすくなります。  
更新履歴を活用すると、やりとりの変遷も追いややすく、合意形成の過程を可視化できます。

### ステークホルダー一覧（2025年3月現在）

氏名/部署	役割	主な関心事項	連絡手段	備考
田中 剛（営業部）	発注元・プロジェクト責任者	コスト、納期、現場での使いやすさ	Slack・メール	週次報告を希望
鈴木 麻衣（製造部）	製造現場代表	組み立てやすさ、メンテナンス効率	月次会議、メール	組立工数データあり
吉田 健（経理部）	支払い・会計管理	初期投資額、ランニングコスト	メール	数値資料を重視
中村 悠（外部取引先A社）	決済ユニット提供	API連携、通信仕様、安全性	直接連絡（Teams）	契約範囲の確認が必要
木下 亮（PM）	開発プロジェクト管理者	全体スケジュール、リスク管理	毎朝定例、Redmine通知	承認フローをWikiで管理

### 合意済み事項の記録

- 2025/03/12：デザインVer.3を基本案として決定（田中氏・鈴木氏）
- 2025/03/18：決済ユニットはA社製を継続使用（中村氏確認済）
- 2025/03/20：製造コスト上限は一台あたり25万円以内（吉田氏）

### 連絡体制とルール

- 定例会議：毎週月曜 10:00～（Zoom）
- 急ぎの連絡：Slackの「#proj-vend」チャンネルに投稿
- 資料共有：Redmineのファイル機能、またはGoogle Driveの共有フォルダ
- 合意事項・議事録は本Wikiに随時反映する

### 現在の確認待ちポイント（2025/03/27 時点）

- 決済ユニットAPI仕様（中村氏）→ 3/29までにドラフト提示予定
- メンテナンスパネル位置の修正要望（鈴木氏）→ 設計チームが対応中
- コスト試算版Ver2のレビュー依頼中（吉田氏）→ 回答待ち（3/28目標）

10

# 統合 マネジメント

プロジェクト全体の計画・実行・変更を  
一貫してまとめる司令塔的な役割

## やるべきこと

## Lycheeでの対応とやり方

### 立ち上げ

プロジェクト憲章を作成し、目的・ゴール・制約条件を明文化する



作成したプロジェクト憲章はwikiに記載し、プロジェクトメンバーが閲覧できるようにしておく

### 計画

プロジェクトマネジメント計画書を作成する（全知識エリア計画のハブになる）



計画書の一部（スコープやスケジュール）はガントチャートを確認する

### 実行

各知識エリアの計画・実行・成果を統合して一貫性を保つ



**（各知識エリアの項目にて解説済み）**  
ガントチャート・EVM・未対応リスクチケット一覧・プロジェクトレポートを確認する。

### 監視・コントロール

変更要求（スコープ変更など）を受けたら、影響を評価して承認フローを通す



**変更要求トラッカー＋ワークフロー**  
**（受付→評価→承認→実装）**を設定しておく  
承認後は作業チケットを関連付ける

### 終結

プロジェクト全体の完了報告・振り返りを実施し、成果物を正式に引き渡す



ガントチャートやEVM、プロジェクトレポートは完了報告書がふりかえりの材料として利用できる



株式会社アジャイルウェア

