

卒論

22J5-114 野々村空河

2025 年 10 月 12 日

1 EVM 理論を応用した個人タスク進捗管理モデル

1.1 EVM 理論の概要

EVM (Earned Value Management) は、計画値 (PV), 実績値 (EV), 投入量 (AC) を比較することで、進捗効率と完了見通しを定量的に評価するプロジェクトマネジメント手法である。本研究ではこれを作業時間 (分) 単位に再定義し、個人タスク進捗管理へ応用する。

Table 1: EVM 理論の基本定義 (時間単位モデル)

記号	名称	定義式	説明
E	Estimate	入力値	見積所要時間 (分)
A_t	Actual	実績ログ累積	時点 t までの累積実績 (分)
D	Deadline	(締切日 - 作成日)	期間 (日)
PV_t	Planned Value	$E \times \frac{t}{D}$	理想的進捗量 (分)
EV_t	Earned Value	A_t	実際に完了した作業量 (分)
SPI_t	進捗効率指数	$\frac{EV_t}{PV_t}$	1 未満で遅れ, 1 超で先行
R_t	残作業量	$E - A_t$	未完了部分 (分)
$pace_{7d}$	実績ペース	過去 7 日平均	実績速度 (分/日)
$requiredPace_t$	必要ペース	$\frac{R_t}{D_t}$	締切までの必要ペース (分/日)
EAC	完了見積	$today + \frac{R_t}{pace_{7d}}$	予測完了日

1.2 個人タスク管理への対応

表 2 に EVM 理論の要素とアプリ変数の対応を示す。これにより EVM 指標をリアルタイムに更新し、個人レベルで進捗・遅延・完了予測を評価できる。

Table 2: EVM 理論項目とアプリ変数の対応

EVM 理論項目	アプリ内変数	算出処理ファイル
E	<code>estimatedMinutes</code>	TodoInput / Firestore
A_t	<code>actualTotalMinutes</code>	updateStats.js
PV_t	理想進捗	updateStats.js
SPI_t	<code>spi</code>	updateStats.js
$pace_{7d}$	<code>pace7d</code>	updateStats.js
$requiredPace_t$	<code>requiredPace</code>	updateStats.js
EAC	<code>eacDate</code>	updateStats.js
$riskLevel$	<code>riskLevel</code>	updateStats.js

1.3 日次割当最適化モデル

EVM 指標を基に遅延リスクの高いタスクを抽出し、1 日の作業キャパシティ内で作業時間を動的に割り当てる。割当アルゴリズムは以下の通り定式化される。

$$\text{maximize } \sum_i w_i \cdot a_i \quad (1)$$

$$\text{subject to } \sum_i a_i \leq Cap, \quad (2)$$

$$0 \leq a_i \leq R_i, \quad (3)$$

$$w_i = \alpha \cdot lag_i + \beta \cdot \frac{required_i}{\max(required)} \quad (4)$$

ここで、 a_i はタスク i への割当時間、 R_i は残作業量、 lag_i は理想進捗との差、 $required_i$ は必要ペース、 Cap は 1 日の上限時間（dailyCap）を示す。

1.4 モデル図（概念）

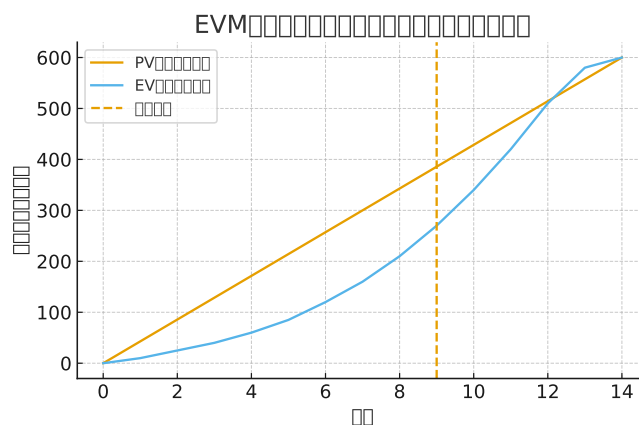


Figure 1: 個人タスクにおける EVM モデル概念図

緑線（EV）は実績進捗，橙線（PV）は理想進捗を示す。EV が PV を下回る場合，タスクは遅延傾向にある。これらの値から SPI および EAC を算出し，翌日の割当を自動調整する。

1.5 理論的主張

本研究では，EVM 理論における指標（EV, PV, SPI, EAC）を時間単位に再定義し，個人タスク単位の進捗管理モデルとして拡張した。さらに，EVM 指標に基づく日次割当最適化アルゴリズムを提案し，ユーザーが自律的に進捗を管理できる仕組みを実現した。