

# 工程管理手法CCPMの適用事例紹介

Application case introduction of process control technique CCPM

黒澤 巖\*

Iwao Kurosawa

プロジェクトにおいて、品質、コスト、納期の達成は不可欠である。その中で、納期については、近年、更なる工期短縮が求められるようになってきている。本報告では、工期短縮に有効とされている工程管理手法であるCCPM（Critical Chain Project Management）を部分的に適用した事例を紹介する。

In the project, the achievement of quality and cost and delivery date is indispensable. Recently, a further term of works shortening is requested about the delivery date in that. In this report, it introduces the case where CCPM（Critical Chain Project Management） that is the process control technique said to be effective to the term of works shortening is partially applied.

## 1. まえがき

ソフトウェア開発プロジェクトに要求されることとして、まず高品質、低コストが挙げられる。それに加えて、近年、従来よりも短い期間での製品提供が求められるようになってきている。筆者が担当することになったプロジェクトでも、従来の方式で工数見積もりを行った結果、客先指定納期にはとても間に合わない、ということがあった。（ここでいう、従来方式の工数見積もりとは、ブレークダウンされたタスク毎に安全余裕を含ませ、ボトムアップで積み上げたものを指す。そのままでは納期内に収まらないため、論理的根拠のない根性論で短縮することになりがちである。）

そこで、工期短縮に非常に効果があるとされており、筆者が数年前より情報を収集し、実践の機会を窺っていた工程管理手法であるCCPMの適用を決意し、工期短縮を図った。CCPMは、プロジェクトのタスクを実行するのは『人間中心』であることを基本理念としており、筆者が大変共感を覚えたことも、適用の要因であった。

本報告では、CCPMの概要に関する説明を踏まえた上で、その適用事例を紹介する。

## 2. CCPM概要

本章では、CCPMの概要として、以下の内容を述べる。

節	内容
2.1	CCPM概要
2.2	従来の問題
2.3	実行手順（計画時）
2.4	実行手順（進捗管理時）

### 2.1 CCPM概要

CCPMは、プロジェクトマネジメントのスタンダードとなっているPMBOK（Project Management Body Of Knowledge）にも記載されているが、まだ世の中には余り浸透していない工程管理手法である。しかし、手法の内容自体は特に目新しい訳ではなく、成功を収めてきたプロジェクトマネージャが暗黙的に行ってきた手法である。

提唱者はイスラエルの物理学者であるエリヤフ・ゴールドラット博士である。彼は、プロジェクトを遂行するにあたり最も重要な要素である「人間」の心理や行動特性を科学的に分析し、対策を体系的にまとめあげた理論であるTOC（Theory of Constraints：制約条件の理論）の一部としてCCPMを発表した。

CCPMは、プロジェクトの制約に着目したマネジメント手法であり、クリティカルパスさえ遅れなければ、最終的にプロジェクトは納期までに完了することになるという全体最適の考えを軸としている。よって、個々のタスクに安全余裕を設けることで、タスク毎の期限を守ろうとする、部分最適な従来の管理方法とは一線を画している。そのためCCPMは、2.2項のような従来存在する問題の解決に有効であり、その実行手順を2.3項、2.4項に示す。

### 2.2 従来の問題

従来の問題として、以下の(1)~(3)を挙げる。

- (1) 従来方式のスケジューリングでは計画工数の増大、及び作業遅延を招きやすい。ソフトウェア開発は人間

が行うものであり、人間の意識的／無意識的心理が影響する。

具体的には、以下のような現象が発生する。

(a) タスク毎に安全余裕をみてしまう（サバを読む）。

これは納期遵守への責任感から発生する。計画工数の増大を招き、論理的根拠のない精神論での工数削減が求められる結果となる。

(b) 与えられた時間は全て使い切ってしまう。

次回の見積もり工数を減らされないため、手抜きと思われなため、又、責任感から要求以上の丁寧な仕事を行ってしまうため、発生する。

(c) 作業開始を期限ぎりぎりまで後回しにしてしまう。

与えられた期間に余裕を感じると後回しにし、負荷を落とし過ぎたり、他のタスクとの掛け持ち作業等を行ってしまう。いざ作業を開始すると、予期せぬ問題が発生し、遅延を招いてしまう。

(d) 遅れは確実に伝播する。

タスクの開始に必要な前工程の依存関係が直列の場合はダイレクトに遅れが伝わる。並列の場合は、複数の前工程の内、1つでも遅れれば、やはり後工程は遅れてしまう。

(e) 進みは伝播しない。

例え早く作業が完了したとしても、(b)と同様の理由により作業の進みは後工程に伝播しない。

(f) 掛け持ち作業を行ってしまう。

特に方針を定めない限り、1つのタスクに集中して作業できることは珍しく、マルチタスクとなりがちである。この時、タスク切り替え時の段取り時間がオーバヘッドとなる。

(2) 計画時に正確な工期を予測することが困難である。

プロジェクトは属人性、他者工程遅延、他プロジェクト作業の割り込み等の不確実性を伴っており、正確な工期を事前に予測することが困難である。

(3) 短納期での開発が求められることが多い。

### 2.3 実行手順（計画時）

計画時の実行手順を以下に示す。又、CCPMの従来工程との比較を図1に示す。

(1) プロジェクトの目的を達成するために必要なタスクを工程の後ろから逆方向に、必要条件として抽出する。

(2) 依存関係に着目して抽出したタスクを並べる。

(3) 同時期のリソースの重複をなくし、クリティカルパスを認識する。

(4) クリティカルパスの安全余裕を切り取り、切り取った合計量の半分を工程の最後に「プロジェクトバッファ」として配置する。

(5) クリティカルパス以外については、クリティカルパスに合流するライン毎に安全余裕を切り取り、切り取った合計量の半分を、クリティカルパスとの合流直前に「合流バッファ」として配置する。（合流する作業の遅れからクリティカルパスを守る働きがある。）

### 2.4 実行手順（進捗管理時）

進捗管理時の実行手順を以下に示す。

(1) 進捗率は「あと何日（何時間）かかるか？」で測る。

(2) 遅れ具合はプロジェクトバッファの消費率（クリティカルパスのタスクの遅れ量）によって測定する（図2参照）。又、図2において、色分けされたゾーン毎に以下のアクションを行う。

緑：安全圏。特に何もしない。

黄：注意圏。対策を検討する。実施はしない。

赤：危険圏。対策を実施する。

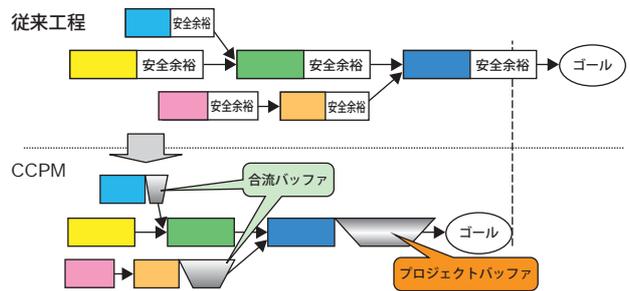


図1 従来工程との比較

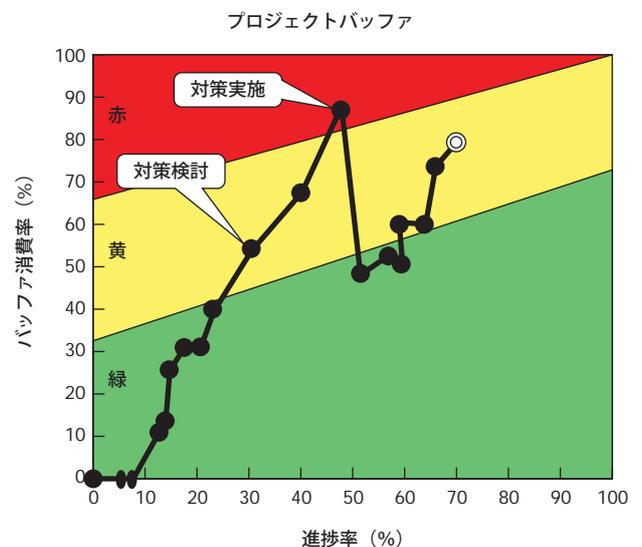


図2 進捗グラフ（例）

### 3. 適用事例紹介

本章では、適用事例紹介として、以下の内容を述べる。

節	内容
3.1	適用プロジェクト概要
3.2	実践内容
3.3	実践結果
3.4	適用事例におけるメリット
3.5	適用事例におけるデメリット
3.6	適用事例に対する改善案

#### 3.1 適用プロジェクト概要

表1に適用プロジェクトの概要を示す。

#### 3.2 実践内容

実践した内容を3.2.1～3.2.4の4段階に分けて記述する。又、CCPMには含まれているが、今回実践しなかった項目を3.2.5に示す。

##### 3.2.1 準備段階

- (1) 開発メンバ、上長、客先に対し、CCPMで工程管理することの宣言を行い、承認を得た。開発メンバに対しては説明会も実施した。
- (2) 目的 (Objectives)、成果物 (Deliverables)、成功基準 (Success Criteria) (併せて「ODSC」と呼ぶ) を決定した。

##### 3.2.2 工程計画段階

- (1) 必要最小限のタスクの洗い出しを目的に、ODSCに向かって、後ろからタスクを並べてネットワーク図を作成した。
- (2) 前工程に対する待ち時間を減らすこと、及び並行作業ができることを考慮してタスクへのリソース割り当てを行い、リソース間の依存関係を極力なくすようにした。これにより、他のリソースの完了を待つ必要がなくなる。クリティカルパス上に割り当てられているリソース(「クリティカルリソース」と呼ぶ)が遅れなければ、プロジェクトバッファを消費せず、プロジェクト全体は遅れない。各個人が独立プロジェクトの

表1 適用プロジェクト概要

分野	表示装置の組込みソフトウェア開発
要員 (計13人)	プロジェクトリーダー1人
	装置(1): サブリーダー1人、開発者4人 装置(2): サブリーダー1人、開発者6人
期間	装置(1): 納期までの期間6ヶ月 (従来方式見積期間8ヶ月)
	装置(2): 納期までの期間9ヶ月 (従来方式見積期間12ヶ月)
対象	装置(1): X-Window、VxWorks
	装置(2): WindowsXP (ROM-WIN)
管理	BeingManagement-CCPM
ツール	Microsoft Excel

ようになり、メンバ全体でマルチプロジェクトのような状態となる。(クリティカルリソースを周りのリソースがサポートする形が期待できる。)

- (3) 各タスクに対して、サバを読まず、問題も発生しないと仮定した時間数 (ABP: Aggressive But Possible) での時間見積もりを実施した。
- (4) クリティカルパスの合計時間数の半分を工程の最後にプロジェクトバッファとして配置した。プロジェクトバッファの終了予定日が納期を越えてしまった場合は、メンバ全員で知恵を出し合い、以下の検討を行うことにより、クリティカルパスの計画期間短縮化を図った。
  - ー更なるサバ読みチェック
  - ー事前の段取りの工夫
  - ー依存関係のつなぎ換え
  - ーリソース割り当ての変更

##### 3.2.3 設計/試験準備段階

- (1) 製作フェーズで、誰もが他をサポート(タスクへのリソース再分担)できることを目指し、設計を工夫した。具体的には、一定のフレームワーク(骨組み)を最初に構築しておき、後は段階的に肉付けしていけば、全ての機能が揃わなくとも動作確認できるように配慮した。
- (2) 各自が並行作業可能となるよう、メンバ1人ずつに対して専用の試験環境を構築した。製品は組み込みのCPUボードであるが、試験環境は比較的調達が容易なパソコン上で動作する環境とした。X-Window+VxWorksに対しては、Linuxでの模擬環境を構築した。併せて、外部装置との通信を模擬するシミュレータも作成した。

##### 3.2.4 進捗把握段階

- (1) 残作業時間報告表を作成し、各メンバに毎日残り作業の予想時間数を入力してもらった。1日に扱うタスクはほぼ1つなので、入力比較的容易であった。毎日の繰り返しが、見積もりの訓練となり、精度の向上につながることを期待した。なお、同フォーマットの表に、後の集計のための実績時間入力も行った。
- (2) 残作業時間報告表を基に、プロジェクトリーダーが工程管理ツールへ値の入力を行い、プロジェクトバッファ消費率を更新した。更新は毎週の定例ミーティングに合わせ、週に一度以上行った。
- (3) 現在実行中のタスクよりも後工程のタスクを先に実行した方が効率が良いことがしばしばあったが、使用していたツールの関係上、後工程の進捗を入力するこ

とができなかった。そのため、便宜的に後工程のタスク工数を減らすことにより対処し、プロジェクトの完了予測日を認識できることを優先させた。

- (4) プロジェクトバッファ消費率の監視を実施した。セオリーどおりに進捗グラフの『値』により対策立案／実施を行った。さらに工夫点として、進捗グラフの『傾斜』により、先の予測を行い、たとえその時点が危険圏でなかったとしても、右上がりの傾斜が急であれば、対策を実施した（図2参照）。
- (5) クリティカルパスの工程を守るため、余裕のあるリソースが、クリティカルリソースの遅れ分を再分担する方策を取り入れた。従来と比較して、CCPMによる工程表ではクリティカルパスと非クリティカルパスの進捗度合いと完了予定日が分かり易く、クリティカルパスへのリソース再分担の量が決めやすいことも一助となった。
- (6) 週一回の進捗ミーティングでは、発言しやすい雰囲気作りが心掛け、メンバの積極的な発言を促し、リスク対策、現状問題対策、ノウハウ等が柔軟に出てくるようにした。これを行った背景としては、CCPMを実践する上で、進捗報告に関しては性善説が基になっており、クリティカルパスの進捗を守るために、メンバ間の信頼関係と、メンバ全員の協力が不可欠であることが挙げられる。

### 3.2.5 非実践内容

今回は、ほぼ初めての導入ということで、CCPMのフルセットは適用せず、いくつかを省略した。その内容は以下の通りである。

- ・合流バッファの挿入と管理
- ・複数プロジェクトの管理

### 3.3 実践結果

- (1) 装置(1)は、客先納期までに完了することができた。しかも、残業、休日出勤は従来に比べ格段に少ないと実感できるほどであった。様々な要因が絡んでいるとは言え、単純計算で2ヶ月（12.5%）の工期短縮が実現できたことになる。
- (2) 装置(2)は、残念ながら途中でCCPMによる管理を断念せざるを得ない状況になってしまった。仕様変更が余りにも多発し、自社努力の許容量を超えてしまい、プロジェクトバッファ消費量が甚大となってしまったためである。しかし、外部要因による工程遅延分を考慮した再スケジュールを客先に承認頂き、結果的に再設定された納期を守ることができた。

### 3.4 適用事例におけるメリット

#### (1) 管理労力の軽減

クリティカルパスさえ遅れなければ、最終的にプロジェクトは納期までに完了することになるため、プロジェクトリーダは、プロジェクトバッファ消費率に重点を置いて監視すればよく、管理労力が軽減された。

#### (2) 遅れ具合、完了予定日の可視化

従来のガントチャートのイナズマ表現は、各タスクの遅れ具合は把握できるが、全体としての遅れ具合や完了予測日が把握しづらい点がある。これと比較してCCPMでは、遅れ具合がプロジェクトバッファ消費率として可視化でき、又、工程表上で完了予定日も分かるため、臨機応変な対策実施のトリガとして有効であった。

#### (3) 先手管理が実施可能

完了予定日をよりの確に把握し、事前に対策をとりながらプロジェクトを進めることができたため、先手管理が実施でき、極端に無理をせずに納期を守ることができた。

#### (4) 協力体制の確立

進んでいる人が遅れている人を随時カバーする協力体制ができた。遅れているタスクに闇雲にリソースを投入するのではなく、プロジェクトを納期通りに完了する為に一番効果的な箇所に絞って投入することで、合理的な根拠を持った協力量針を策定、実践することができた。

### 3.5 適用事例におけるデメリット

#### (1) ABP時間見積もりが困難

当初、ABP時間見積もりが難しく、技術者のプライドから短く見積もり過ぎ、遅れが出やすい傾向にあった。そして、リカバリが思うように進まない、モチベーションが下がってしまうという現象を引き起こした。

#### (2) 再スケジュール時のバッファ消費率設定方針誤り

仕様変更／追加の度に再スケジュールを行い、プロジェクトバッファ消費率を0%に戻してしまったため、メンバの危機感が薄れてしまった。（図3において、バッファ消費率が最初に注意圏に入ってから0%に戻っている部分参照。）

#### (3) 工程毎に進捗グラフの傾斜傾向注意

コーディング工程と試験工程では、進捗グラフの傾き具合が違うことが多いため、工程の移行直後は傾向を見誤らないよう、注意が必要である。

#### (4) 極端な短納期プロジェクトには不向き

ABPの合計だけで納期を迎えてしまうような場合等、納期までの期間が短すぎるプロジェクトには向かない。

#### (5) 協力拒否の可能性

性善説を基にしているため、悪意があれば、計画工数

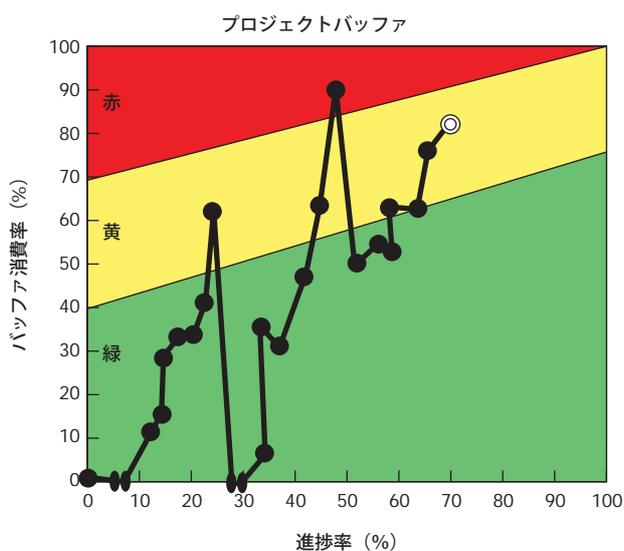


図3 進捗グラフ（装置(1)の途中実績）

を長めに設定したり、進捗を遅めに申告して、タスクの再分担を免れることができってしまう。（幸いにして、本事例では発生していない。）

### 3.6 適用事例に対する改善案

#### (1) ABP時間見積困難対策

ABP時間見積もりが難しい点に関しては、過去や直近の実績値から今後のタスクの残作業時間を見積もる方法を積極的に取り入れることにより、改善が期待できる。

#### (2) 性善説対策

性善説を基にしている点に関しては、残作業時間の申告値を鵜呑みにせずに、上記(1)により時間算出の根拠を確認する必要がある。しかし、日頃からのコミュニケーションに留意し、信頼関係を築いておくことを忘れてはならない。

#### (3) 再スケジュール時方針

仕様変更／追加時の再スケジュールに関しては、その

時点での残作業だけでプロジェクトバッファを作成し直して消費率を0%にするのではなく、追加作業の延長分を元のバッファに追加して消費率を考える必要がある。可能であれば、客先ともバッファ消費状況を共有し、悪化原因を認識してもらいながら、共に解決策を出し合うことが望ましい。

## 4. むすび

CCPMの概要と、手探りながらも1つのプロジェクトを完遂させることができた適用事例について述べてきた。この事例にも、まだまだ改善の余地があるが、人間をプロジェクトの中心として理解し、実践できる手法として、手ごたえを感じており、今後も実践を続けて行きたいと考えている。本報告が、CCPMになじみのない読者に興味を持ってもらい、又、これからCCPMを実践しようとしている読者には少しでもお役に立てれば幸いである。

### 参考資料

- (1) 目標を突破する実践プロジェクトマネジメント（岸良裕司 中経出版）
- (2) PMプロジェクト・マネジメント クリティカル・チェーン（中嶋秀隆、津曲公二 日本能率協会マネジメントセンター）
- (3) クリティカルチェーン（エリヤフ ゴールドラットダイヤモンド社）
- (4) TOCクリティカル・チェーン革命（稲垣公夫 日本能率協会マネジメントセンター）