

Critical Chain Project Managementのすすめ～不確実性を受け入れる～

Recommendation of Critical Chain Project Management ～ Accept uncertainty ! ～

阿部 淳*
Atsushi Abe

プロジェクトを成功に導く、全てのプロジェクトマネージャは、このために日々マネジメントを行っている。プロジェクトの遂行において、その成否にもっとも影響を与えるのは人である。またその影響は、ソフトウェア開発のように、不確実性の高い成果物^(注1)を得るためのプロジェクトにおいて顕著となる。では我々は、そのことを意識してソフトウェア開発のマネジメントを行っているだろうか。

本稿では、プロジェクトの不確実性と遂行する人の行動心理と行動特性を考慮して考え出されたマネジメント手法“Critical Chain Project Management : CCPM”について紹介する。

All the project managers are managing every day, in order to lead a project to a success. In execution of a project, it is a person that affects the success or failure most. Moreover, the influence becomes remarkable in the project for obtaining a product with high uncertainty like software development. Do we consider that and are managing software development?

The project management technique CCPM is introduced in this paper. This CCPM was invented in consideration of the uncertainty of a project, and people's action psychology and behavioral trait.

1. まえがき

書店にはプロジェクトマネジメント関連書籍が数多く並び、プロジェクトマネジメントへの関心はいまも非常に高い。当社においても、人材育成計画でプロジェクトマネージャ育成が重要であることが示され、ソフトウェア開発におけるプロジェクトマネジメントの重要性は以前にも増して強く認識されるようになってきた。ではこの関心の高さが、プロジェクトの成功やプロジェクトマネージャの負荷軽減へと繋がっているであろうか。

情報処理推進機構ソフトウェア・エンジニアリング・センター (IPA SEC) の調査結果では、残念ながら成功といえるITプロジェクトの比率はここ数年あまり増加していない^(注2)。また当社におけるプロジェクトマネージャの負荷は、軽減されてきたというよりむしろ上がってきているように思える。この理由を筆者は、PMBOK^(注3)に示されているような従来型プロジェクトマネジメント^(注4)は、不確実性を極力減らすマネジメントであり、不確実性にどのように対処すれば良いかということとはほとんど取り上げられていないからであると考えている。

ソフトウェア開発プロジェクトには非常に多くの不確実性が存在する。そのため、実際にプロジェクトを遂行していくには、ある程度の不確実性を受け入れる必要がある。しかしながら、従来型プロジェクトマネジメントは、そのための手段を示してくれていないのである。

今回紹介するCCPMは、不確実性を念頭に置いて考え出されたものであり、プロジェクトを進める人の行動心理と行動特性も考慮した非常にパワフルでシンプルな実践的マネジメント手法である。ぜひそのメリットを知って頂き、ソフトウェア開発プロジェクトのマネジメントに活用して頂きたい。

- (注1) 決まっているのは要求機能だけであることが多い。実はその要求さえも曖昧で、別の人が作れば別の成果物になる不確実性の高いプロジェクトであると言える。
- (注2) ソフトウェア開発プロジェクトデータ白書2009年版63.9%、2010年-11年版63.8%
- (注3) Project Management Body of Knowledgeの略：プロジェクトマネジメントの知識体系で事実上の国際標準
- (注4) PMBOKで示されているようなプロジェクトマネジメントを本稿では従来型プロジェクトマネジメントと呼ぶ。

2. 従来型プロジェクトマネジメント

従来型プロジェクトマネジメントは、プロジェクト目標を品質 (Q)、コスト (C)、工期 (D) で設定し、これらを含めた関連管理項目を監視する。PDCAサイクルを回すことによって不確実性を極力排除し続け、個々の管理目標を達成することでプロジェクトを成功に導いていく。PMBOKにはリスク管理表のように具体的な帳票からEVM^(注5)のような手法まで多くのツールが列挙されている。しかしながら、排除しきれない不確実性がある中で、何にポイントを絞るどのようにプロジェクトをマネジメントしていけばよいかという実践的な記述は少ない。プロジェクトを成功に導くには多くの経験が必要となってくるのである。

実際、従来型プロジェクトマネジメントにおいては、筆者の経験からも以下のような悩みを抱えてマネジメントを行うことが多い。

(1) 要求の未確定

スコープを決めるための要求がなかなか定まらず計画が立てられない。工程表も引けない。

(2) タイムマネジメントの難しさ

定期的に状況の確認や問題への助言を行うが、徐々に進捗遅れが拡大していく。その結果、工程計画と実績進捗が乖離し、何度も工程の見直しを行うことになる。

(3) 多くの管理項目

品質、コスト、工期、スコープ、リスク等、多くの管理項目 (問題発生検知のためのチェックポイント) があるため、その状況を確認するだけで多くの時間がかかる。そのため、問題を検知しても対策を打つ余裕がない。

(4) 対策実施の難しさ

リソース追加等の対策を打つ必要があっても、リソースは限られており余裕もない。他チームにヘルプを依頼しても他チームも余裕がないためなかなか協力を得られない。

(注5) Earned Value Managementの略：プロジェクトの進捗とコストの状況を価値という統一指標で把握、マネジメントする手法の一つ

(5) 実態にそぐわない報告

当初順調に進捗が進んでいても、タスク終了間際で進捗が止まる。なかなかタスクが完了しない「90%症候群」が発症する。

(6) モチベーション低下

プロジェクトマネジメント負荷が増大し、以下のような理由でチーム全体のモチベーションが低下してしまう。

- プロジェクト情報の収集、分析をすることが目的になってしまう。フォローだけ実施してマネジメントはなされない。
- プロジェクトマネージャがマネジメントをあきらめ、実作業の遂行に没頭してしまう。その結果、状況が見えなくなり、メンバーは目的や目標が曖昧な中で作業を進めていかなければならなくなる。

(7) 納期前は火の車

「前倒しで」「前倒しで」と訴えていても、結局納期前は火の車になってしまう。

ソフトウェア開発を進める上で、これらの悩みを引き起こしている主な制約や不確実性を表1に示す。

従来型プロジェクトマネジメントでは、これらに対応しきれず、工数のギャップには個々人が残業や工期のサバ読みで対応している。しかしこのように個々人が頑張ったとしても、工程を守ることはなお非常に難しいのである。その理由をCCPMでは人の行動心理・行動特性に起因していると説明している。

表2にスケジュール遅延の要因となる人の行動心理・行動特性を示す。

3. CCPM

3.1 CCPM概要

TOC^(注6)の提唱者エリヤフ・M・ゴールドラット博士が、1992年にTOCの基本原理を適用して考え出したプロジェクトマネジメント手法がCCPMである。

本手法は、企業の永続的利益獲得のため、プロジェクト工期を短縮することを目的としている。タスク間の依存関係の連なりであるクリティカルパスにリソースの重

(注6) Theory of Constraintsの略：制約条件理論

表1 ソフトウェア開発プロジェクトの制約と不確実性

項目	概説	補足
コスト制約	プロジェクトには目標コストがあり、追加リソースの投入が非常に難しい。また、目標コストと見積りコストのギャップが大きいことがある。	目標コストをメンバ全員がコミットメントしていないことも多い。目標コストをメンバが知らないこともある。
工期制約	顧客が希望する納期と、必要な工期との間にギャップがあることが多い。	目標工期をメンバがコミットメントしていないこともある。
要求未確定	顧客の要求がなかなか確定せず。部分的もしくは曖昧な要求の状況で作業を始めなければならないことがある。	そもそも全要求の文書化は難しいためある意味で確定出来ない。
タスク抽出漏れ	プロジェクトはユニークであるためタスクの抽出漏れが発生してしまうことがある。	特に初めての業務分野や新規技術の導入時。
見積り困難	作業量、作業効率ともに予測である。作業量の正確な予測は困難であるし、作業効率には個人差があり、見積り工期を画一的にまた正確に行うことは出来ない。	人を特定し、PSP等で個人の作業効率の実績を計測していたとしても、作業内容によって実際の作業効率はばらついてしまう。

表2 スケジュール遅延要因

行動心理・行動特性	概説	原因
学生症候群（一夜漬け）	納期直前まで他作業を実施してしまう。様々な業務を抱えている場合に顕著。	掛け持ち作業。 進捗への影響が不明確。
パーキンソンの法則	与えられた期間をすべて満たすまで、仕事の量は膨張する。 ・早く終わりそうであれば、手直し等でより良い品質をめざす。 ・期間内で命一杯丁寧な仕事をする。	品質追求。 エンジニア気質。
マルチタスク	オーバーヘッド時間が積算。またタイムシェアリングはタスク終了のパフォーマンスを低下させる。	非プロジェクト型組織。 プロジェクト間、タスク間の曖昧な優先順位。
早期完了の未報告	早く終わりそうでも、期限まで丁寧な仕事をしたり、別の仕事をしたりにして、作業完了を報告しない（悪気はないが確信犯的）。	早く終わると、見積りが甘いと叱責されたり、次回見積り工数が削減される可能性がある。

複を加味した、クリティカルチェーンを制約条件としてマネジメントを行うのがCCPMであり、通常よりも工期を2割から3割短縮出来ると言われている。次節以降、CCPMの特徴をプロジェクトの立ち上げと実行におけるPDCAサイクルに従って説明する。

3.2 プロジェクトの立ち上げ

プロジェクトの立ち上げにおいて、非常に重要なものに目標の擦り合わせがある。CCPMではこのために、ODSCというツールを使う。プロジェクトの目標を、Objective（目的）、Deliverables（成果物）、Success Criteria（成功条件）の3つの目標で議論するのである。

ここで重要なのは、目標をただプロジェクトマネージャが設定するのではなく、ステークホルダ間（上位マネジメント層も含む）で擦り合わせを行い共有することである。ODSCがプロジェクトを遂行する上での判断基準となるため、擦り合わせ時に上位マネジメント層が参加出来なくとも、必ず承認は得る必要がある。この判断基準

が、自律したチームの土台となり、CCPMによってプロジェクトを成功させる上での一つの重要なポイントとなる。表3にODSCシートの記載例を示す。

3.3 プロジェクトの実行/PLAN

CCPMにおけるプロジェクト計画の手順を表4に示す。

表4手順1～3は、CCPMにおけるプロジェクト計画手順の特徴の一つ、バックワードスケジューリングというタスク抽出手順である。ODSCをプロジェクトの最終到達点として、逆引きで必要なタスクを洗い出していく。こうすることによって、目標達成に必要なタスクを過不足なく抽出することを狙っている。

そして手順6がCCPMのスケジューリング時に行う最大の特徴であるバッファ設定である。バッファにはプロジェクトバッファ^(注7)と合流バッファ^(注8)の2種類があ

- (注7) 工期を守るために納期直前に設定。
- (注8) クリティカルチェーン上には無いタスクのクリティカルチェーンへの合流期日を守るバッファ。合流点の直前に設定。

表3 ODSC記入シート（例）

“Critical Chain Project Managementのすすめ” ODSC	
Objectives（目的）	<ul style="list-style-type: none"> ・MSSのプロジェクト成功率を向上させる。 ・MSSのプロジェクトマネージャの負荷を低減する。 ・プロジェクトを成功に導きたいと思っている者に、広くCCPMを知ってもらう。 ・CCPM導入の敷居を低くする。 ・筆者がCCPMを推進していることを知ってもらい、CCPM導入支援の窓口となる。
Deliverables（成果物）	<ul style="list-style-type: none"> ・“Critical Chain Project Managementのすすめ”原稿 ・CCPM情報メモ
Success Criteria（成功基準）	<ul style="list-style-type: none"> ・原稿がMSS技報に掲載される。 ・CCPMに興味をもってもらいCCPMの質問を3件以上もらう。 ・CCPM導入支援を行い、適用プロジェクトを成功に導く。
その他	文才が無いので、諸先輩方に原稿チェックを何度か実施して頂く必要がある。

表4 CCPMプロジェクト計画手順

No.	概説	備考
1	プロジェクトメンバで、ODSCを達成するためには何をしておかなければならないか（タスク）を逆引きで考える。	理解しやすい言葉で「～する」と表現。
2	ひとつ前のタスクを実施するために何をしておかなければならないかをプロジェクトの始まりまで逆引きで洗い出す。	上記No. 1と同様
3	今度はプロジェクトの始まりから時系列にしたがってタスクを確認する。	成果物を作る観点から
4	リソースを決める。	担当者、設備
5	所要期間を見積もり、リソースの競合を取り除いたガントチャートを仮作成する。	クリティカルチェーン
6	バッファを設定して、CCPMガントチャートを完成させる。	

り、それらを設定した上でバッファマネジメントを行う。以下に、CCPMにおけるバッファの考え方と設定手順例を示す。

一般的にプロジェクトメンバは、プロジェクトを成功に導くため、タスク工期を守ろうと考える。プロジェクト経験者であれば、プロジェクト遂行に不確実性があることも理解している。そのため、その不確実性を吸収すべく、自身の個々のタスクに可能な限り“期日の余裕(以降バッファと呼ぶ)”を確保しようとする。しかしながら、このタスク工期遅延対策は、表2に示した人の行動心理と行動特性から、プロジェクト工期を守る目的に対してはあまり機能しない。

CCPMでは、この不確実性を吸収するバッファを確保しようとする考え方はそのままに、視点とその設定の仕方を変える。守るべき期日は個々のタスクの完了期日ではなく、プロジェクトの納期であることから、バッファを一つにまとめてプロジェクトバッファとして、クリティカルチェーンの後ろ、納期の直前にまとめて設定する。こうすることによって、バッファは個々人の管理から離れ、納期を守るためにプロジェクト全体でマネジメントするプロジェクトバッファとなるのである。同様に、クリティカルチェーンへの合流点の直前に設定するのが合流バッファである。

バッファ設定の手順はプロジェクトの目的に応じて様々であるが、ここではHP^(注9) 1点見積りをベースにした設定手順の例を紹介する(図1)。

(1) 工期見積り

タスク毎にHPで1点見積りを実施し、見積り工期を設定する。必要に応じて、プランニングポーカー^(注10)等のツールも活用。

(2) クリティカルチェーン積算工期

クリティカルチェーンの見積り工期の積算が、納期と比して現実的なレベルか確認し、現実的でなければ、見積り精査、リソースの追加投入等で、見積り工期を短縮させる。このとき、プロジェクト目標や組織方針に合致していることも確認する。

(3) 目標実施工期

見直した個々のタスクの見積り工期の半分を各タスクの目標実施工期とする。

(4) バッファ設定

組織方針に沿った比率でバッファサイズを決める(推奨は、目標実施工期の半分^(注11))。

(5) プロジェクトバッファ設定

クリティカルチェーン上のバッファを積算して納期の直前にプロジェクトバッファとして設置する。また、非

- (注9) Highly Possibleの略。高い確率で完了可能な工期。
- (注10) アジャイル開発で用いられる見積り手法。チームで作業規模の擦り合わせを行いながら見積る。
- (注11) HP工期の1/2で、タスクを完了出来る確率はほぼ五分五分であるとの仮定のもと、クリティカルチェーン上の“各タスク工期のバッファ：HP工期の1/2”を積算したもののさらに1/2をプロジェクトバッファとすれば、五分五分の確率でバッファを消費した上で、工期を守ることが出来るとの考え方に基づいている。

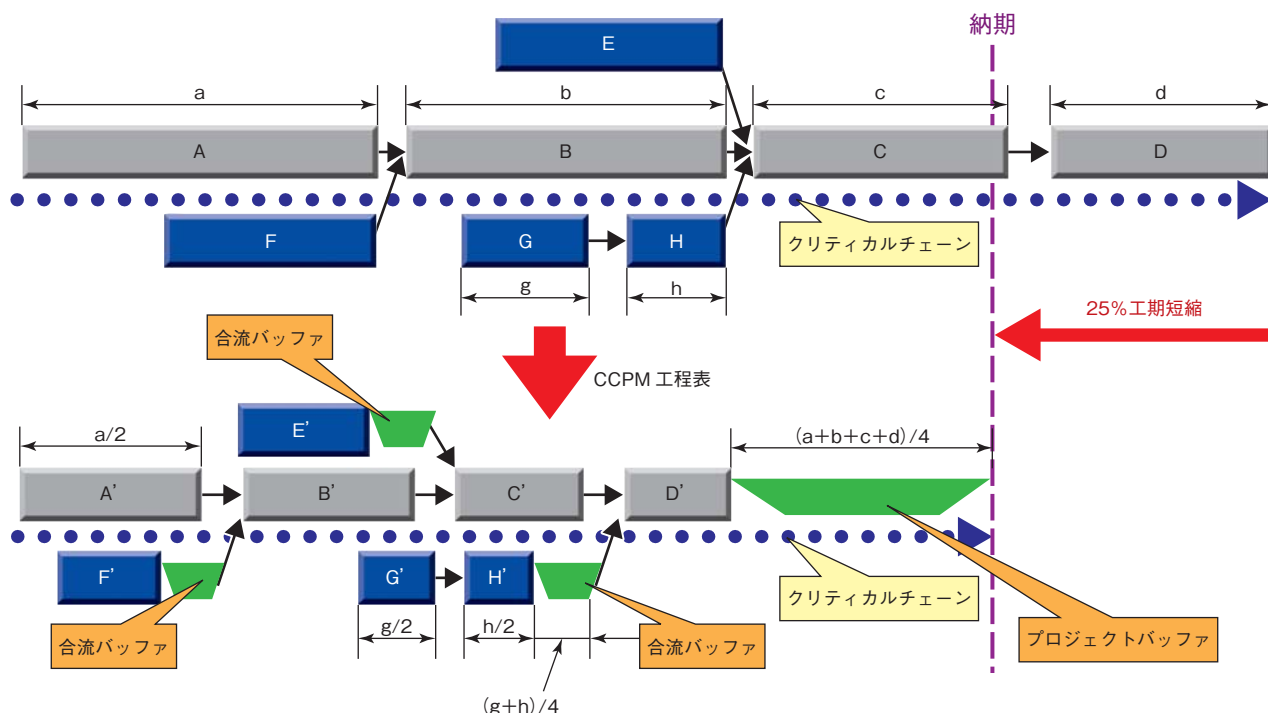


図1 CCPMバッファ設定

クリティカルチェーン上のバッファを合流点まで積算して合流点の直前に設置し合流バッファとする。

3.4 プロジェクトの実行/DO

タスクの担当責任者は、各タスクの進捗状況を残日数で報告する。これによって、マネージャが知りたい実態に近い進捗状況を計測することが可能となる。

残日数をガントチャートに反映して、プロジェクトバッファの残量を算出し、バッファ傾向グラフにプロットする（図2にCCPMにおけるガントチャート表示例、図3にバッファ傾向グラフの例をそれぞれ示す）。

作業を遂行する中で、適切なタイミングで資源投入予告（作業着手予告含む）^(注12)等のマネジメントを行う。

3.5 プロジェクトの実行/CHECK

プロジェクトマネージャは、バッファ傾向グラフ（図3参照）でプロジェクトバッファ残量の変化を確認して、対策ACTIONのトリガを検出する。また、付帯情報も監視しプロジェクト状況変化の兆候（以降、シグナルとい

う）を掴むようにする。付帯情報を用いたシグナルの例を表5に示す。シグナルが発生した場合は、詳細状況（問題点、ヘルプ要求）の確認をタスク担当者に行う。

3.6 プロジェクトの実行/ACTION

プロジェクトマネージャは、バッファ傾向グラフ（図3）にプロットされた領域に応じて対策検討や対策の実行を行う（プロット領域に応じたACTIONを表6に示す）。

CCPMは、人の行動心理と行動特性に着目して遅延を抑えるとともに、発生した遅延を可能な限り早く検出して対策を打つ先手マネジメントによって納期順守、工期短縮を達成する手法である。早めに遅れを検出するため、チーム内での対策時間が確保されて対策を打ちやすくなる。さらに、いくつかのプロジェクトにおいてCCPMでの成功事例が示されれば、助け合いの風土も広がり、チーム間や部門間での助け合いが活発化して対策の幅も広がっていく。

(注12) CCPMではこの資源投入予告のマネジメントをリソースバッファ管理という。

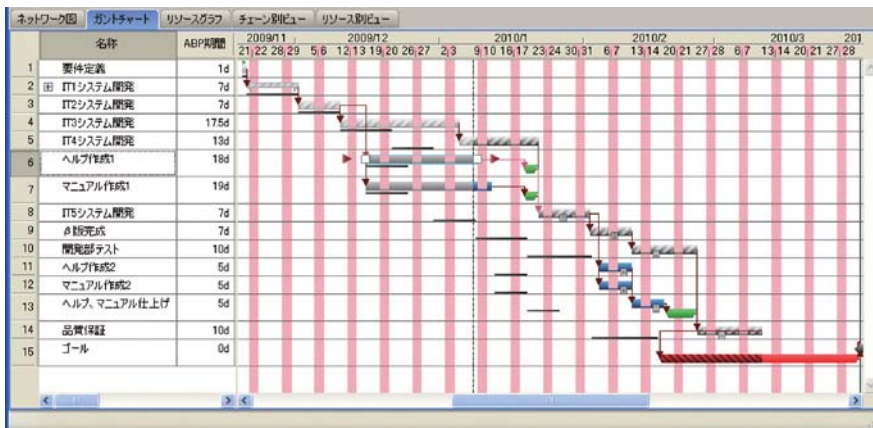


図2 CCPMガントチャート

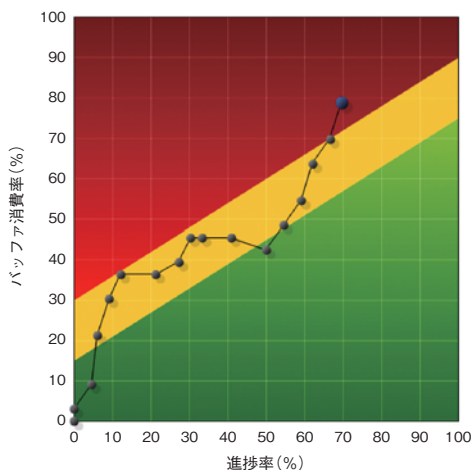


図3 バッファ傾向グラフ

表5 付帯情報シグナル

付帯情報	シグナル (例)
残日数	残日数変更なしが3日連続する。
	残日数増が2日連続して発生。
	残日数が1日に1ずつ順調に減少していくことが1週間続く。
バッファ傾向グラフ	プロットを結んだ線の変化(量、傾き、傾向)。

表6 対策ACTION

シグナル (プロット位置)	アクション名称	対策詳細
緑色領域	—	なし
黄色領域	対策準備	遅延防止/挽回対策を検討し、対策実行に向けた準備を実施(部門間調整、ネゴ等の取り付け等)。
赤色領域	対策実行	上記検討対策を実行する。

4. 期待効果

CCPMを導入することによる期待効果^(注13)を表7にまとめる。

2章で述べた7つの悩みも以下のように解消される。

(1) 要求の未確定

プロジェクトの開始段階の適切なタイミングで区切り、その時点の情報で要求を一旦FIXする。そこまでの要求でタスクの洗い出しを実施し、不足事項や不明点は段階的詳細化を行いながらプロジェクトを進める。結果として生じる工期のギャップはバッファで吸収することによって解消する。

(2) タイムマネジメントの難しさ

残日数情報を収集してバッファ傾向グラフの更新をおこない、バッファの変化状況を監視するのみ。進捗遅れはバッファで守られているため、進捗状況の更新による工程上の期日のシフトはあるが、工程表そのものの見直しは発生しない。

(3) 多くの管理項目

マネージャの平時の主たるチェックポイントはバッファ傾向グラフのプロジェクトバッファ残量の変化のみ。他の管理項目は、計測しプロジェクトの目的と状況に応じて適宜確認する。

(4) 対策実施の難しさ

対策を打つべきタイミングが明確になるとともに、早く問題の発生を検出することが出来るようになり、先手マネジメントで従来よりも余裕を持って対策を打つことが出来る。また、助け合いの文化が醸成されれば、協力依頼に対応してくれる可能性も高まる。

(5) 実態にそぐわない報告

パーセント管理ではなく、残日数マネジメントを実施することにより、工期に対する状況を把握しやすくなり、90%症候群を引き起こす可能性も軽減される。

(注13) プログラムマネージャが、複数プロジェクトの状況をバッファ傾向グラフで概観し、各プロジェクト状況を瞬時に判断できることから、プログラムマネジメントの洗練化（柔軟なプロジェクト間協力）も期待できる。

(6) モチベーション低下

共通の目標ODSCに向かい、プロジェクトマネージャが適切なレベルのマネジメントをおこなうことにより、ステークホルダ間でコミュニケーションとコラボレーションが推進され、チームワークが高まり、メンバーのモチベーションアップにつながる。

(7) 納期前は火の車

バッファマネジメントによる先手マネジメントで、負荷が平準化され納期前の過負荷も軽減される。

5. 導入時の課題

筆者がCCPMをプロジェクトに適用するにあたって悩んだ点と現時点での筆者の考えを述べる。

(1) 工期見積りで多点見積りを実施すべきか

筆者はバッファ工期の根拠を持ちたいとの思いから、当初は多少手間が掛かっても3点見積り^(注14)を用いてバッファ設定を実施していた。しかし今現在は、不確実性を受け入れるという考え方もあって、工期見積りで多点見積りを実施する必要はないと考えている。

(2) バッファサイズ比率

当初は、バッファサイズ算出のための比率についても悩んだ。ほとんどの教科書で、目標実施工期の半分である50%を推奨しているが、そもそも個々のタスクに余裕などないとの考えから、バッファを削ることへの理解がなかなか得られない場合があった。そのような導入当初においては、工期が許せばクリティカルチェーンの積算工期と同期間をそのままプロジェクトバッファとして設定することを薦める。

(3) バッファ可視化の是非

CCPMの適用を議論する中で、“バッファをステークホルダ（メンバー等）に見せるか否か。”という質問が講習会等でよくなされていた。現時点での筆者の考えからいえば、バッファはオープンにした方が良いと思っている。

(注14) 楽観的見積り値、最確見積り値、悲観的見積り値

表7 CCPM導入期待効果

項目	概説	備考
作業効率向上	目的の擦り合わせによってメンバー間で目的が共有され、現場が全体最適を考えて行動するようになり、プロジェクト全体での作業効率が向上。	個別最適⇒全体最適
メンバー自律	プロジェクトにおける判断基準（ODSC）が明確になることによって、あまり干渉しないマネジメントが可能となり、メンバーが全体最適に向かって自律して作業を進められるようになる。	メンバーの成長に繋がる。 “やれ” ⇒ “やる”
マネジメント負荷軽減	平素の実行フェーズ、監視コントロールフェーズにおいては、干渉しないマネジメントによって、マネジメント負荷が軽減。	工程表の改定頻度も大きく減少（原則無し）。
状況報告負荷軽減	干渉されないマネジメントの下、平素は実行タスクの残日数見込みの報告のみとなり、報告負荷が軽減。	

以前は、バッファをメンバに見せずにマネジメントを実施していた時期もあるが、最近は短工期工事が多く、計画の時点で短い各タスク工期にメンバの理解が得られない可能性も高い。またそんな状況でメンバにバッファを見せなければ、信頼関係を築けずCCPMに期待される効果である助け合いの風土は生まれないと考えるからである。

(4) EVMとの共存

EVMは顧客要望等の時勢もあって、MSSにおいてもほとんどのプロジェクトで使用されている。しかしながら、バッファでプロジェクト状況を把握しようとするCCPMにおいてはEVMを使用するマネジメントはそぐわないという意見が一般的であり、その算出方法も確立していない。

筆者の考えでは、EVMはプロジェクトマネジメントを行う上で非常に有益なツールであることから、その計測は並行して行い具体的な対策検討局面での遅延分析やコストマネジメント、顧客から要求がある場合の報告等に使用することが適当であると考えている。

本稿では紹介しないが、CCPMと共存させる場合のEVM諸元算出方法についても見解を持っているので、活用するにあたっては相談して頂ければと思う。

6. むすび

CCPMについては、MSS技報Vol.19において工程管理手法の適用事例として既に紹介されている。しかしながら、まだまだ社内に浸透していないのが現状である。筆者が本稿で改めてCCPMを紹介したのは、本手法がソフトウェア開発プロジェクトのマネジメントに非常にマッ

チしていると考えているからであり、より多くのソフトウェア開発にCCPMを導入すべきであるとの思いからである。まずは本手法の特徴とメリットを知り、本手法の活用を検討して貰えればと考えている。

導入にあたっては、様々な障害や迷う部分もあるかと思う。最初から工期短縮は目指さず、まずは工期厳守に向けたツールとして活用し、MSSのプロジェクト成功率の向上に結びつけて貰えれば嬉しい限りである。また、それに向けて筆者もできる限りの支援をしていきたいと考えている。

参考図書

- (1) クリティカルチェーン—なぜ、プロジェクトは予定どおりに進まないのか？
(エリヤフ・ゴールドラット著、ダイヤモンド社)
- (2) TOC／CCPM標準ハンドブック (西原 隆著、秀和システム)
- (3) マネジメント改革の工程表 (岸良 裕司著、中経出版)
- (4) 全体最適のプロジェクトマネジメント (岸良 裕司著、中経出版)
- (5) プロジェクトマネジメント 実践編 (中 憲治、総合法令出版)