Crossfields
Works Collaborated by Professionals

プロジェクトを成功に導くポイント

1. プロジェクト管理（進捗）後篇

株式会社クロスフィールド

庄司 堅太郎

クロスフィールド　レポート TOP ページへ
http://www.crossfields.co.jp/reports/index.html

Crossfields
Works Collaborated by Professionals

目次 （後篇）

はじめに（後篇）

進捗管理全体のうち、前篇では「進捗の把握」を中心に、正しい進捗把握のための仕掛け作りから報告・管理方法まで、即効性の高い確認ポイントについて説明した。ただ正しい進捗把握を行った結果として、個別タスクやプロジェクト全体が遅延していることが判明した場合には、マネジメントの立場からは早急になんらかの対処の実行が求められる。そこで後篇では、「遅延への対処」に対する確認ポイントを主に紹介したい。

遅延の対処方法としては一般的な手法を列挙しているが、やり方よりもそれぞれの実行ポイントとリスクについて主に説明している。また遅延対処は計画変更を伴うため、合わせて実施すべき組織やステークホルダーのマネジメントについても、重点的に記載している。

なお今回扱うプロジェクト事例の前提としては前篇同様、IT システム導入プロジェクト（カスタムメイド、パッケージ問わず）を想定している。

遅延に対処する

進捗管理における PDCA サイクルにおいては、前篇で述べたような正しい仕組み作り（Plan）と進捗把握（Do）が実行できれば、あとは状態分析（Check）と対処実行（Action）を実行することになる。。本篇では対処実行として回復が見込める場合と見込めない場合の手法を、実行時のポイントと共に説明する。

３章 遅延回復のための対処

遅延を把握し対処が必要であると判断された場合には、遅延回復のために計画外作業を行うことになるが、その際他のタスクとの整合性を取りながらの部分的な再計画・実行が必要となるため、スケジュール・コントロールの複雑さが増すことになる。また手法によってはプロジェクト・マネジメント上のリスクも発生するため、それらにプロジェクトマネジャーがうまく対処できなければ、遅延対応が更に遅延を招くことになってしまう。よって取りうる手法ごとの留意ポイントや発生リスクを事前に体系立てて認識しておくことはプロジェクトマネジャーにとって重要であり、本章ではこれらについて主に説明する。
なお、各手法は筆者の経験則に基づいて、考慮すべき順番に列挙している。

■ 主にチーム単位で取りうる手法

1） フロートを見つける

要件定義や外部設計等が進行して来て要件や機能が明確化されるにつれ、後続タスクの所要期間も精緻化されるため、見積もり時に計画したタスクの前後関係にも変化が生じてくる。その変化で発生した時間的猶予は「フロート」と呼ばれ、それらを再検証することで遅延回復のためのバッファとして用いることを考える。またチームリードは自チームの要件の変化を全て把握しているため、まずはその変化があった機能群について、チームリードがチーム SOC[1]の範囲内でスケジュールの再精査と対策立案を行っていくことがポイントである。
また計画時に取り切れていなかった過剰に“サバを読んでいた”安全余裕も、このタイミングで吐き出してしまい、遅延対応に当てていく、ということが必要である。

【リスク】
- 特になし

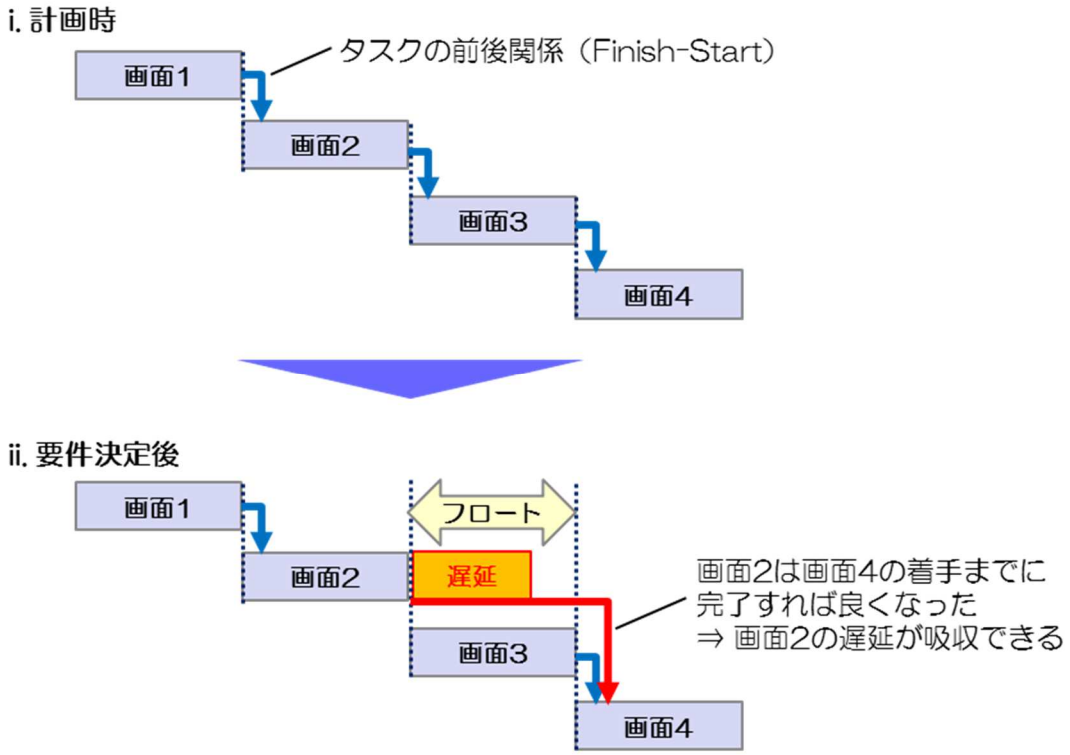
2） 残業する

既存のリソースかつ役割分担で実行するため、最も取りやすい手法であるが、やり方によってはメンバの生産性低下に繋がりやすい手法である。ポイントは「過度に継続する残業は NG」のみであり、実施する場合には期間集中型で実施し、目的や完了条件もメンバに事前に明示するなど、組織の雰囲気やモチベーション低下に最も留意すべきである。

【リスク】
- 過度な残業によるモチベーション低下、疲労による生産性低下
- （場合により）内部コストの増加

[1] スパンオブコントロールの略。各進捗管理、報告の階層ごとに明確な管理者を設置し、その範囲内での管理責任と裁量を与えること。詳しくは本レポート前篇 第１章を参照のこと。

図１：フロート検証の事例（1 機能内の複数画面設計を想定）



3） リソースの再割当を行う

プロジェクトが進んでいくと、各スタッフの得意分野やスキル、生産性が見えてくるため、本人の適正を勘案してリソースの再割当を行う、という手法である。例えば、機能ごとに担当を分けていた共通部分の設計をまとめて 1 担当に実施させる、生産性の高いメンバ（エース）に遅延部分の担当を変更する、といったものである。ここで留意すべきは「遅延部分をエースが手伝う」といったリソース補強の方法であり、この手法の一環としてよく実施されると思われるが、特定リソースへの高負荷による残業やタスク切り替え時のオーバーヘッドが発生しているような過度なマルチタスキングが発生していないか、再割当の結果が逆にボトルネック発生の要因にならないように注意が必要である。

【リスク】
- 役割変更による切替時間および立ち上がり時間の発生
- 人的ボトルネックの発生

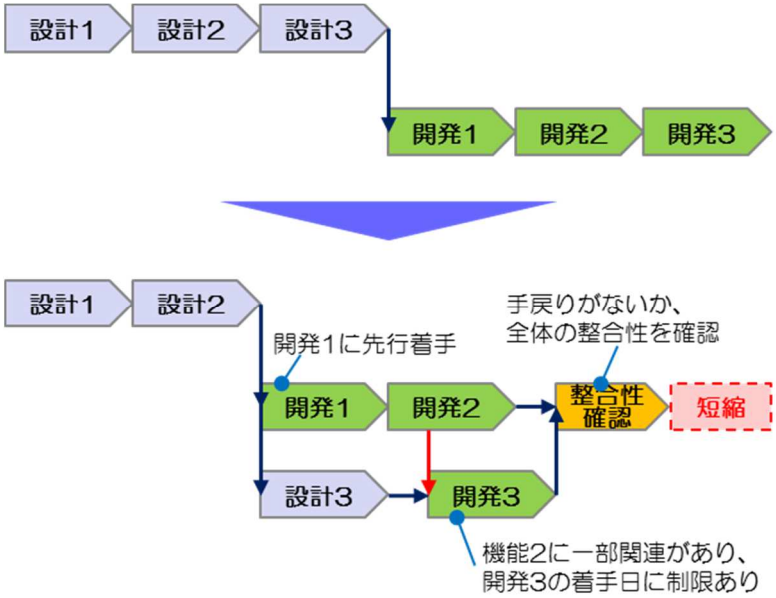
■ 主にプロジェクト単位で取りうる手法

4） ファストトラッキング（Fast Tracking）

ファストトラッキングとは、本来直列に実施される一部のタスクやフェーズに先行着手することで並列化し、トータル所要期間を短縮する手法である。

本手法のポイントは、「できるだけ手戻りの発生を少なくすること」である。元々の計画はなんらかの理由があって直列化されていたはずであり、その理由を検証し、十分な検討をもって対処を実行したい。また対策によって得られる短縮期間の一部を、手戻りのためのバッファ、または将来の手戻りを防ぐ確認期間としてあらかじめ確保しておくことが望ましい。

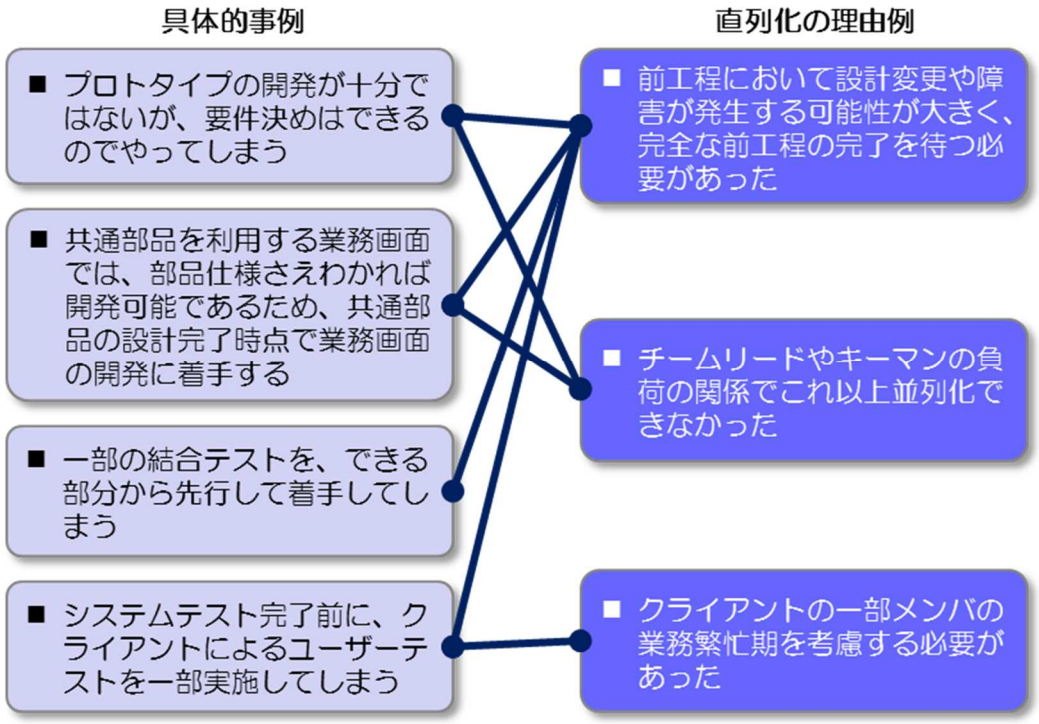
図２：ファストトラッキングの例



なお計画段階で十分にタスクが並列化されている場合は更に並列化を行うことになるため、人的リソースの追加（残業含む）も合わせて必要となる。その場合は次に述べるクラッシングの同時検討が必要である。

【リスク】
- 一部タスクの先行着手による後続タスクでの手戻りの発生
- タスク並列化によるマルチタスキングの発生及びリソースの不足
- スケジュール・コントロールの難易度増大

図３：ファストトラッキングの事例



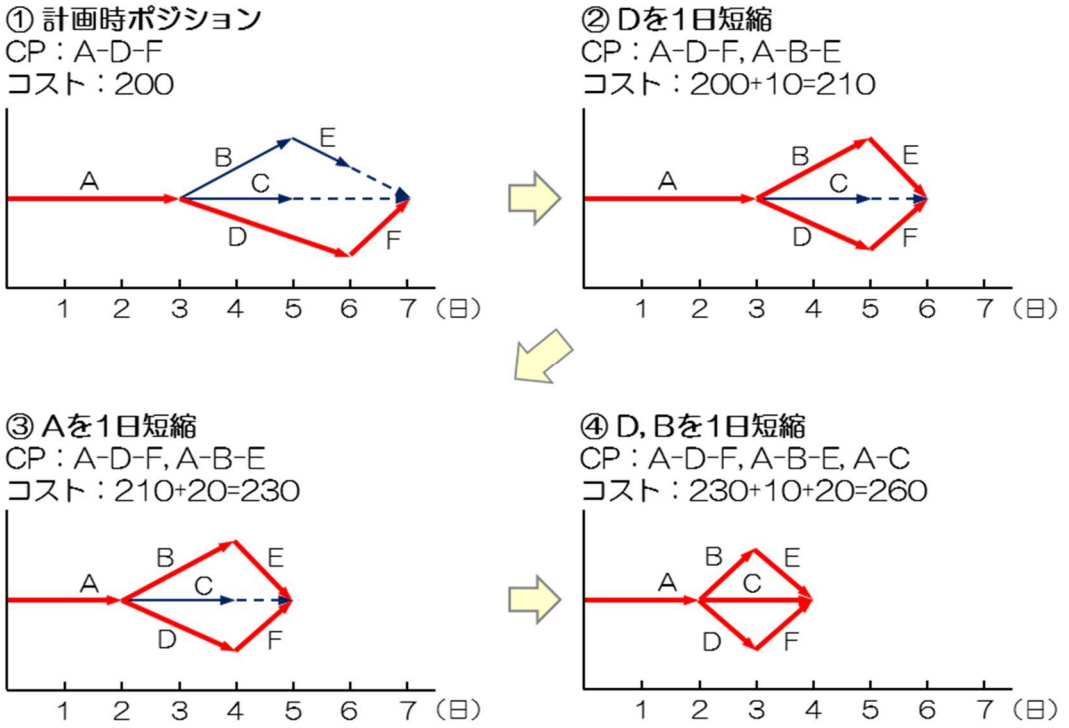
5） クラッシング（Crashing）

クラッシングとは、リソースを追加しつつもコスト増を最小に押さえながら期間短縮を図る手法である。この手法を採用するケースで多いのは、要員追加によるクリティカルパス上タスクの所要期間短縮を図る場合であるが、人的リソース以外でもハードウェアやツールの導入などによる生産性向上、所要期間短縮もクラッシングには含まれる。よって追加の金銭的コストを最小限に留めるためにはどのリソース（モノ・ヒト（組織））に投入すれば最も効果的なのかをプロジェクトで十分な検討する必要がある。更にそのコスト追加のためのステークホルダー間事前合意も、プロジェクトマネジャーの重要なマネジメント業務である。

図４：クラッシングコストの算出と、簡単なクラッシングの例[2]

アクティビティ	所要期間（日）		コスト		コスト/時間勾配
	通常	クラッシュ	通常	クラッシュ	
A	3	2	30	50	20
B	2	1	40	60	20
C	2	1	20	80	60
D	3	1	30	50	10
E	1	1	40	40	0
F	1	1	40	40	0

※ コスト/時間勾配：単位時間あたりのクラッシングコスト



※1：CP＝クリティカル・パス
※2：コスト/時間勾配の小さいタスク順にクラッシング

[2] ドラガン・ミロセビッチ著、PMI 東京支部訳、「プロジェクトマネジメント・ツールボックス」、鹿島出版会

クラッシングにおける人的リソース追加時の第１のポイントは、新規調達要員に求めるスキルセットを具体的かつ明確にしておくことである。筆者の経験上、プログラマ、テスターレベルの調達では大きなズレはなかったが、特にチームリードや PMO といった上位者の調達には注意が必要で、プロジェクト管理の経験があるからといって、進捗管理や課題管理の個別実務経験がなく、想定していた役割レベルの業務をこなせるようになるまで相当の時間を要したことが何度かあった。とはいえ急にプロジェクト外部からリソースを調達することは上位者になればなるほど難しいため、妥協点があるにせよ「ここだけは外せない」というスキル要求は明確にしておくべきである。

第２のポイントは、「戦力の逐次投入」は非常に効率が悪い、ということである。新規要員は必ずプロジェクト説明や要件引継等のスタートアップ時間が必要となり、習熟度の向上は学習曲線（S 字曲線）をたどるため、見積時に想定した生産性を有する戦力となってくれるまでは相応の時間を要する。そのような中、新規要員のスキル不足やスタートアップ期間中の遅延拡大、別問題の発覚等でまたすぐに要員追加が必要となってしまい、いつまで経っても根本的な対策が打てない、という状況をよく見かける。よってクラッシングを考える場合には、小規模な要員追加を複数回行うよりも、ある程度先を見越して、一定の余裕リソースを加えたまとまった要員を一度に追加してしまう方が、結果的に時間・コストの節約に繋がる場合も多いことを十分留意しておくべきである。

その他、今後プロジェクトの目的や作法をまとめたもの（プロジェクト・チャーターの一部や事務的な手続き方法等）をプロジェクト・スターターキットのような名称で準備しておけば、追加リソースの受入を段階的に繰り返し行う場合でも、効率化と一定の品質を保つことができる。

【リスク】
- 新規調達リソースのスタートアップ時間の発生
- 複数回の調達による時間的・コスト的オーバーヘッドの発生

4 章 遅延回復が見込めない場合の対処

前章では当初スコープを確保しながら回復が見込める場合の対処を説明したが、本章では許容されるコスト追加によっても回復が見込めない場合の対処方法について説明する。これらの対処はプロジェクト当初のスコープまたは納期を変更するものであるため、プロジェクトが中止に追い込まれるという最悪の結果を避けるためにも、合わせてステークホルダーマネジメントも慎重に行う必要がある。

■ スコープ縮小と稼働日延期、段階稼働

この対処を検討する状況においては、プロジェクトで遵守すべき QCD の指標にスコープの“S”を合わせた QCDS で考えると、品質（Q）は落とすことはできず、コスト（C）はすでに追加済みという状況であり、よってスコープ（S）と納期（D）について調整を行うことになる。
調整には、対応工数を削減するための**①スコープ縮小**、追加の実行日数を稼ぐための**②稼働日延期**の 2 軸があり、またその二つの複合として、一斉稼働を必ずしも必要としなくてよい場合には、展開組織や機能について段階的に稼働させる**③段階稼働**が挙げられる。どの対応を取るのかは、プロジェクトの状況とクライアント側の稼働要件を中心に、精緻に各種検討ポイントを検証した上で決定する必要がある。また段階稼働は、一部の組織または機能において当初稼働予定日に稼働実績を部分的に得られる一方、部分稼働期間内の業務フローの計画やシステム上のデータ整合性担保、複数回の移行・切り替えが必要となり計画・実行の難易度も上昇するため、より注意深い検討が必要となる。

検討ポイントについては様々な状況が想定されるため一概には言えないが、ここでは比較的どの状況にも共通するであろう、プロジェクト内で検討すべき主な項目を挙げる。

表 1：スコープ縮小、稼働日延期、段階稼働における主な検討ポイント

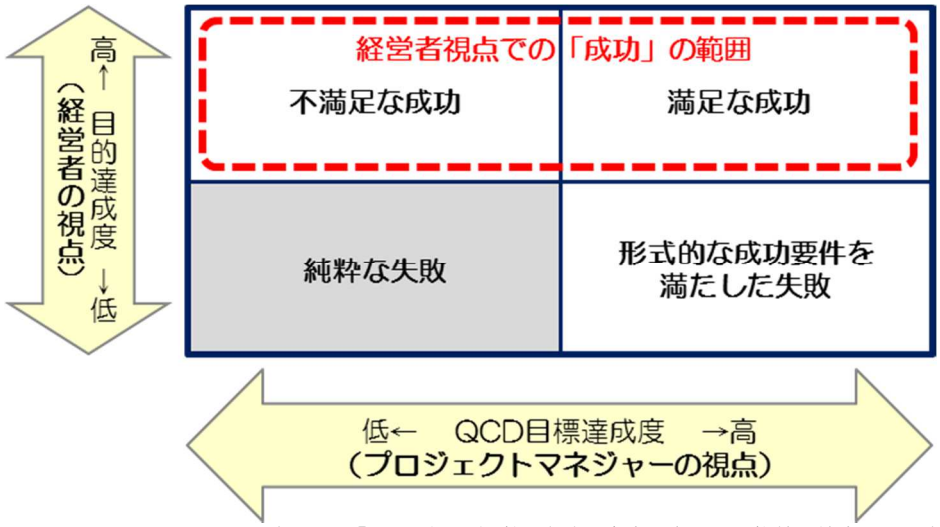
分類	検討ポイント	各対処における検討要否		
		① スコープ 縮小	② 稼働日 延期	③ 段階 稼働
システム 機能	業務オペレーションの継続性	○	○	◎
	インターフェースのデータ整合性	○	○	○
	現行システムの稼働延長		○	○
	現行システムとの並行稼働			◎
	段階稼働期間中のデータ整合性			◎
	段階稼働期間中のバージョン管理			◎
移行・切替	データ移行、システム切替の実現性	○	○	◎
	ユーザーへの周知・トレーニング実施	○	○	◎
運用保守	運用保守体制の構築		○	○
マネジメント	プロジェクト延長に伴うコスト確保		○	○
	プロジェクト延長に伴うリソース確保		○	○
	ステークホルダーとの合意	○	○	○

○：検討必要、◎：十分に検討必要（難易度高）

■ プロジェクトの価値を最大化する対策をとる

プロジェクトとして更なるスケジュール延期を伴う遅延対応策を決定し実行に移すためには、プロジェクト関係者での検討・合意はもちろんだが、最終的にはステアリングコミッティでのプロジェクトオーナーによる承認を経る。従って、対策はプロジェクトオーナー（＝経営者層）の視点に合致するものでなければならず、たとえ当初計画通り進まなかったプロジェクトだとしても、今後いかにプロジェクトが生み出す価値がビジネスに貢献するか、という視点が重要となる。確かにプロジェクトマネジャーからみれば一義的にはQCD 目標達成度を上げることが必要とされる。ただしこのような状況下ではプロジェクトオーナー視点から考えることで、納期を大きく遅らせても高い品質を守ることや、コストを更につぎ込んでも短納期を守る、スコープを大幅に縮小して品質を担保した基本機能のみを予定どおり稼働させる、といった「QCD 達成度は低くともビジネスに最も貢献する」対策を取るべきである。

図５：経営者視点に基づくプロジェクトの「成功」の定義[3]



※ 日本 IBM 「SCM と IT 経営・実践研究会」資料より抜粋、筆者にて図を再作成

■ キーマンの意識統一を図る

プロジェクトオーナーの判断を仰ぐような規模のプロジェクト決定となると、利害関係者も多くなり、関係者間で必ずしも一枚岩となった対策案の策定ができない可能性もある。もしそうなった場合、ステアリングコミッティでの意志決定に思わぬ影響を及ぼしたり、対策実行時に一部の組織から十分な支援が得られなかったりということが十分に考えられる。よってプロジェクトで対策案を検討する際には、関係各署の中で影響力の強いキーマン全員に必ずプロジェクト案に賛同してもらえるよう、十分に調整、説明し、キーマン全員の意識統一を図っておくことが重要である。このようなステークホルダーマネジメントは、プロジェクトの通常時は実行されているものの、大幅遅延時などは遅延対策に忙しくおろそかにされがちであるため、留意すべきである。

[3] 日本 IBM 栗山 敏、SCM と IT 経営・実践研究会「経営者視点に基づくプロジェクトの成功・失敗の考察」、2010 年８月

最後に

前篇・後篇の２回にわたり、できるだけクラシカルな進捗管理手法に沿ってスケジュール・コントロールのポイントを説明してきたが、近年では EVM（アーンドバリューマネジメント）や CCPM（クリティカルチェーン・プロジェクトマネジメント）などの特色のある管理手法を採用するプロジェクトも増えているようである。PMBOK もそれらを取り入れつつ日々進化しているが、教科書にいくら手法が増えようとも「その手法をどのようにプロジェクトに適用し、実行まで落とし込むか」という、プロジェクトマネジャーの試行錯誤や経験・ノウハウの蓄積がなければ、それら手法の最大限の活用は難しいと考えている。今回は紙面の都合で割愛したが、筆者の経験でも CCPM をプロジェクト最後半の大規模リカバリ策に利用したところ有効に機能し、実行面で貴重な経験を得たことがある。やはり実際に実行してみるとコツがわかってきて、次回に向けた改善点も見えてくるため、自分の中で手法の利用方法がブラッシュアップされてくることを感じた次第である。プロジェクトマネジャーの方々においては、プロジェクト状況や特性を鑑みながら、本レポートで紹介したポイントや新しい手法を積極的にプロジェクトに適用いただき、スケジュール・コントロールにおいての知見を深めていただければと考えている。