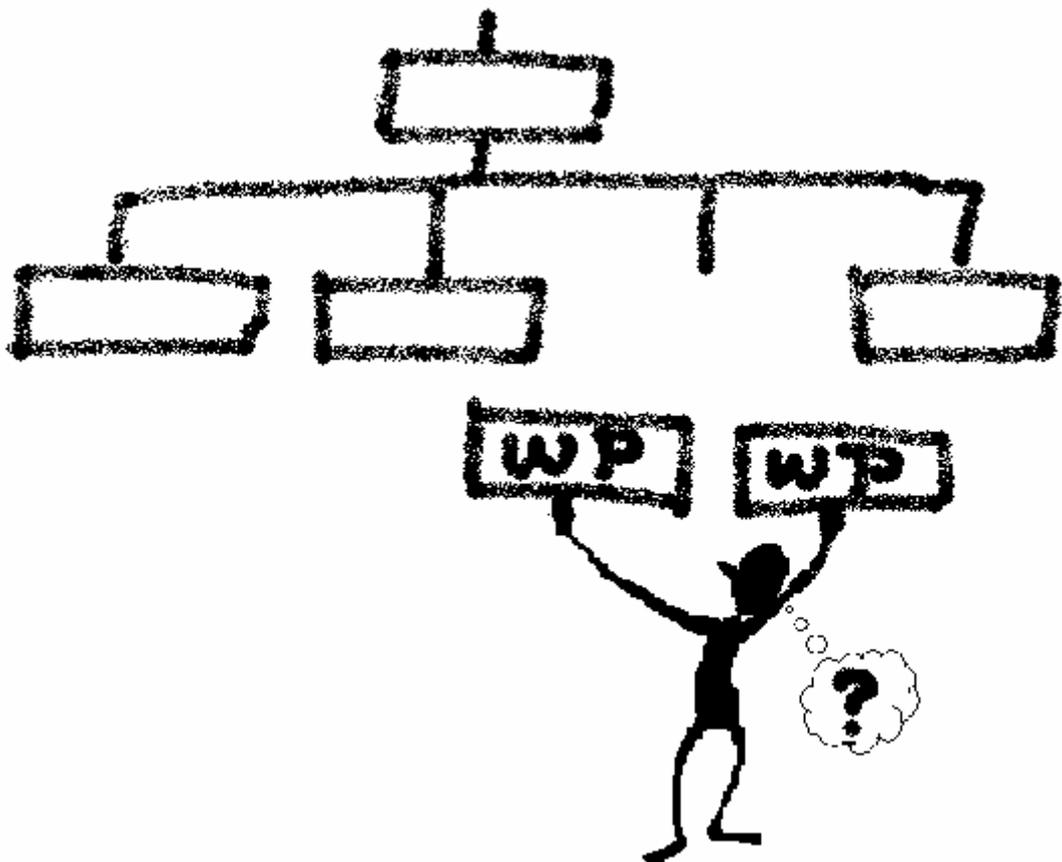


WBSの運用で困ったときに開くノウハウ集



2005年2月

第20年度 ソフトウェア品質管理研究会

第2分科会 WBSグループ

目次

はじめに	3
1. Tips1 WPの粒度をどう揃えるか	4
2. Tips2 管理の一元化(WBSとガントチャート)をどのように行なうか	9
3. Tips3 管理の一元化(WBSと課題管理)をどのように行うか	15
4. Tips4 変更をどう管理するか	19
5. Tips5 EVMにどうつなげるか	22
6. Tips6 メトリクスをどう取るか	26
7. Tips7 見積りにどう使えばよいのか	30

はじめに

本書の目的

本書は、本グループのメンバが、WBS を利用し運用していくときの実践的なノウハウを出し合い、「WBS の運用で困ったときに開くノウハウ集」としてまとめたものである。本グループでは開発現場で活躍しているプロジェクトマネージャ、品質保証やPMO でプロジェクトを周りからサポートしている人など、プロジェクトを成功させるために日夜努力している人々が参加しており、そのような様々な立場から検討されてきたノウハウがまとめられている。

今回作成した小冊子「WBS の運用で困ったときに開くノウハウ集」は、前年度の第2分科会 WBS グループが作成した「WBS 使ってますか？」(WBS 導入・運用ガイド)と併用することで、WBS を利用したプロジェクト管理に役立てるものとなっている。

想定する利用者

本書が想定している読者は、プロジェクトを遂行する方々、
経営者、プロジェクトマネージャ、プロジェクトリーダー、担当者、PMO(スタッフ)など、WBS に直接触れていない方も含め、各立場の方々に参考になるようにノウハウをまとめている。

略語集

#	略語	正式名称
1	W B S	Work Breakdown Structure
2	W P	Work Package (ワークパッケージ)
3	P M	Project Manager (プロジェクトマネージャ)
4	P L	Project Leader (プロジェクトリーダー)
5	C M M	Capability Maturity Model
6	I S O	International Organization for Standardization
7	E V M	Earned Value Management
8	P V	Planned Value
9	A C	Actual Cost
10	E V	Earned Value

1. Tips1 WPの粒度をどう揃えるか

1.1 前提

WBS 展開するにあたり、『どこまで展開すればいいの?』という疑問にぶつかる。これは、WBSの最下位層(WP)をどこまで詳細に分解するのか、分解する基準はあるのかということで、ここでは、そのWPの粒度の決め方への助言を提示する。WPの粒度を決めることは、そのWBSを利用した「プロジェクト管理」、「メトリクスへの利用」、「WBSの再利用による成熟」につながる最初のステップとなる。

1.2 問題

WBS 展開で WP まで展開する時に、WP の粒度に関して次のような疑問点が浮かび上がり、実際に WBS を展開するときの障壁となる。

- WP の粒度を揃える必要があるのか？
- プロジェクト管理に適した粒度は？
- 担当者が開発しやすい粒度は？
- メトリクスに利用しやすい粒度は？
- 一度作成した WBS を次の開発に再利用しやすくするための粒度は？
- EVM での進捗管理がしやすい粒度は？
- WP ごとに作業の難易度の差があるが、この場合の粒度は？
- WP の担当者によりコストの差が発生するが、この場合の粒度は？

1.3 解決方法

ここでは、個々の疑問点への解答として以下の解決案を提案する。

1.3.1 WP の粒度を揃える必要があるのか？

『粒度を揃えることで、管理面、作業面でプラス条件が増える』

WP の粒度を揃えるということは、

- プロジェクトマネージャやプロジェクトリーダーが進捗を管理しやすくなる
- WP を割当てられた担当者がスコープを把握しやすくなる
- メトリクスを収集し他と比較するためには、粒度が揃っていないとデータとしての意味がなくなる。
- EVM では、粒度が揃っていないと精度が落ちる

等があり、WP の粒度を揃えるということはプロジェクトを実施していくために有効である。

1.3.2 プロジェクト管理に適した WP の粒度は？

『WP の粒度は作業期間を目安とし、期間は 10 日以内とする』

プロジェクトマネージャやプロジェクトリーダーが、受け持ったプロジェクトを管理していく場合、長期間（例えば 1 ヶ月程度の粒度）で WP を作成すると、大雑把すぎる計画のため作業内容がぼやけ、見積もり精度にも欠け、進捗状況の正確な変化をリアルタイムに把握することが困難になり、その結果プロジェクト管理に問題が生じることになる。

そこで、プロジェクトマネージャやプロジェクトリーダーがプロジェクトの状況を把握しやすい状態で維持するには、WP が制御（管理）しやすいものでなくてはならない。

制御しやすく、作業内容を明確に定義できることを前提とした場合、WP の粒度は、10 日以内に完了できる規模を目安として設定することを薦める。

1.3.3 担当者が開発しやすい粒度は？

『WP は短期間で、入出力、作業内容、完了条件を明確に定義する』

作業担当者に WP を割当てると、作業スコープを把握しやすくするためには、WP は期間以外にも次の表に示された情報を定義する必要がある。

No	項目	項目の詳細	備考
1	作業名称	ワークパッケージの名称	
2	担当者	担当者名	作業を担当するメンバ名
3	予定期間	作業の開始日と終了日の予定	可能な限り 10 日以内に分割
4	入力	ワークパッケージの入力	参照するドキュメント類等
5	出力	ワークパッケージの成果物	作成するドキュメント名等
6	作業詳細	ワークパッケージの作業詳細	概略の作業実施手順等
7	注意事項	ワークパッケージ実施時の注意事項	レビューの指示、注意事項等
8	進捗報告	ワークパッケージの進捗報告方法	PL への進捗報告時期を指定
9	実績期間	作業の開始日と終了日の実績	計画に対する実績

(第 19 年度 SPC 研究会第 2 分科会 WBS グループ『WBS 使ってますか?』から)

WP の情報に上記の情報を定義しても、作業期間が長期間で設定されていると、実施する作業の範囲（幅）が広がるため、入力から出力（成果物）を作成するプロセスが想定されるプロセスとは異なってしまう場合や、予定外の作業が入り込むことが考えられる。また、「学生症候群」と呼ばれている症状も出やすくなり、計画した期間に完了しないこと（納期遅れ）や、納期間近になって作業が集中することも考えられる。

これらの状況は、プロジェクト管理だけでなく、製品品質にも悪影響を及ぼしかねない。

このような状況になることを回避するために、WP の粒度は短期間で設定することを薦める。具体的な期間は、10 日以内を目安とする。

1.3.4 メトリクスに利用しやすい粒度は？

『有効活用できるデータを収集するには粒度を揃える必要がある』

メトリクスは、各社利用目的が異なるので、必然的に収集するデータは異なる。メトリクスに利用するデータは、より高い精度が求められ、計測結果を比較するためには、比較対象間でデータのスケールを合わせておく必要がある。データ収集を目的とした WP の粒度は、細かく展開することで、データ精度が高くなり、リアルタイム性（変化の予兆の早期検知）が向上する。また、メトリクスと WBS の関係は、「メトリクスをどう取るか」の項を参照。

1.3.5 一度作成した WBS を次の開発に再利用しやすくするための粒度は？

『WBS の再利用には WP の粒度より WBS 展開方法の比重が大きい』

既存の WBS をそのまま再利用するには、開発プロセス方式、WBS の展開ベース（プロセスベース / プロダクトベース）が同じで、尚且つ成果物までも同様なシステムで無い限り、WP レベルまでを全て再利用することは難しい。しかし、WP へ展開する直前のレベルであれば、再利用は比較的容易であると考えられる。

実際に、『Practice Standard for Work Breakdown Structure』（PMI 発行）では、様々な分野で利用できるパターンが用意されており、それをベースに WP まで展開していくことが可能である。

1.3.6 EVM を利用しやすい粒度は？

『EVM では、リアルタイム性と精度が求められ、そのためには粒度が揃っていて短期間のものが良い』

EVM では精度の高い情報と、リアルタイム性が求められる。

最初のより精度の高い情報を得るためには、WP から収集できるデータの精度が高い必要があり、それには作業期間は短期間であるということと、粒度が揃っていることが条件となる。

次にリアルタイム性とは、プロジェクトが変化する予兆を早期検知できるということで、この場合も作業期間は短期間であるということが条件となる。

また、EVM と WBS の関係は、「EVM にどうつなげるか」の項を参照。

1.3.7 WP ごとに作業の難易度の差があるが、この場合の粒度は？

『WBS 展開作業時は、実担当者ではなくモデルリソース（標準リソース）を想定して展開する』

WP ごとに難易度の差があるのは当然だが、WP 作成時に難易度を問題としてしまうのは、往々に WP に展開している時に、既に担当者を頭に思い浮かべている場合が多い。これでは思い浮かべた担当で、WP の難易度が変化してしまい、WP に割当てする期間も変化する。

しかし、本来 WBS は成果物を出力するために必要な作業を展開することが目的であり、その時点では担当者を想定する必要は無い。

ただ、リソースを想定しないと、WP まで作成しづらいのは否めない。この場合、モデルリソース（標準リソース）を想定することで難易度に左右されずに WP まで展開できる。

最終的に WP を担当者に割り当てる時に、モデルリソースと比較して WP の作業期間を調整していく手順をとっていくことが望ましい。

1.3.8 WP の担当者によりコストの差が発生するが、この場合の粒度は？

『WBS 展開作業時は、実担当者ではなくモデルリソース（標準リソース）を想定して展開する』

WP に割当てた担当者のスキルにより作業時間のコストが異なるが、WP 作成時にコストを意識してしまうのは、往々に WP に展開している時に、既に担当者を頭に思い浮かべて展開している場合が多い。これでは思い浮かべた担当で、WP のコストが変化してしまい、WP に割当てする期間も変化する。

しかし、本来 WBS は成果物を出力するために必要な作業を展開することが目的であり、その時点では担当者を想定する必要は無い。

ただ、リソースを想定しないと、WP まで作成しづらいのは否めないことで、この場合、モデルリソース（標準リソース）による展開していくことでコスト差に左右されずに WP まで展開でき、プロジェクト全体のコスト計算もモデルリソースで計算することで、コスト差の問題はなくなり、WP の粒度も担当者によるコスト差に影響を受けることがなくなる。

1.4 WBS のチェック

参考として WP まで展開した後、作成した WBS が適切に展開されているか確認するためのチェック項目の例として次に例を示した。

WBS のレビューに使用すること薦める。

【チェック項目】

- プロジェクトマネージャやプロジェクトリーダーが、制御できるレベルになっているか？

展開されているレベルが、大まかすぎず、細かすぎないレベルになっているか。
WP の粒度が、この Tips で推奨されている 10 日以内または、職場で規定されている粒度（規定の存在する場合）の範囲となっているか。

- WP は制御するために意味のあるものか？

WP の粒度が基準どおりに展開されていても、制御することに意味を持たない内容のものになっていないか、入力と出力のあるものとなっているか。

- 独立して作業ができる内容か？

他の作業に影響されず、また影響を与えない内容になっているか。

- 成果物が明確になっているか？

作業完了時に出力されるものが何か、内容はどのようなものなのかが、明確に定義されていて、割当てられた担当者が理解できるか。

- 作業の完了条件が明確になっているか？

作業がどのような状態になった時に、完了と判断するのが、プロジェクトマネージャやプロジェクトリーダーだけでなく、担当者が理解できるようになっているか

- WBS の内容は重複が無く漏れの無い集合体になっているか？

WBS は、成果物を展開して作業を定義していくので、漏れがあると期待された成果物は生成できず、重複がある場合は展開にミスがあるか、余分な作業を計画している可能性があり、WBS 自体が論理的に生成されていない可能性がある。

- WP の責任者は明確になっているか？

WP に割当てる担当者は必ずしも一人とは限らない。1 つの WP で定義された作業を複数の担当者を割当てる必要がある場合、作業の大きさに関係なく責任者が不在の仕事は、失敗する方向に進みやすい。

このような WP を定義する場合、必ずその実施責任者を決め、その責任者を中心として WP の作業を遂行していくようにする。

以上

2. Tips2 管理の一元化（WBS とガントチャート）をどのように行なうか

2.1 前提（想定する状況）

ガントチャート（線表なども含む）で進捗管理を行なっている現場に WBS を導入するケース、もしくは WBS ベースの管理を行なっているが、マネージャや上位管理者向けにガントチャートを用いた進捗報告書が必要となるケースなど。

2.2 問題

プロジェクト初期のマネジメント手法にしたがうと、スコープを決めて WBS を作成し、アクティビティ順序設定からスケジュールを決定する流れとなる。ガントチャートは一般にこの時点で作成される。

プロジェクトに変更や修正が生じなければ進捗管理はガントチャートによる予実管理でこと足りるが、現実にはそのような状況は期待できない。変更や修正は必ず発生すると考えなければならず、それらが発生したときに問題が顕在化する。

たとえば要求仕様が変更されワークパッケージの所要日数が増加するケースでは、WBS とガントチャートの両方を矛盾なく変更しなければならない。両方を同期して変更できる仕組みや運用が確立されていれば良いが、担当者の注意力に依存していると変更の頻発などによりその担当者に大きな負担がかかる。往々にして WBS とガントチャートに差異が生じ、プロジェクトに混乱を招くであろう。それが WBS の導入に抵抗する理由にもなりうる。

変更管理に関する単純な問題であるが、実運用上は軽視できない。具体的なシーンを想定してよく検討しておく必要がある。

2.3 解決方法

以下のような解決方法がある。

(1) WBS とガントチャートを統合した市販ツールを導入する。

障害管理やリソース管理、課題管理などまで含めたツールが市販されている。カバーされている機能や遠隔地からの接続可否、ベンダの特性やサポート条件など、導入先部署の要件とよく照らし合わせて選択すべきであろう。

手っ取り早く一元化を実現できるメリットがあるが、カスタマイズはほとんどできないか制限が大きく、効率的、効果的な運用にこぎつけるのが難しい点には注意しなければならない。

図 2-1 に例を示す。

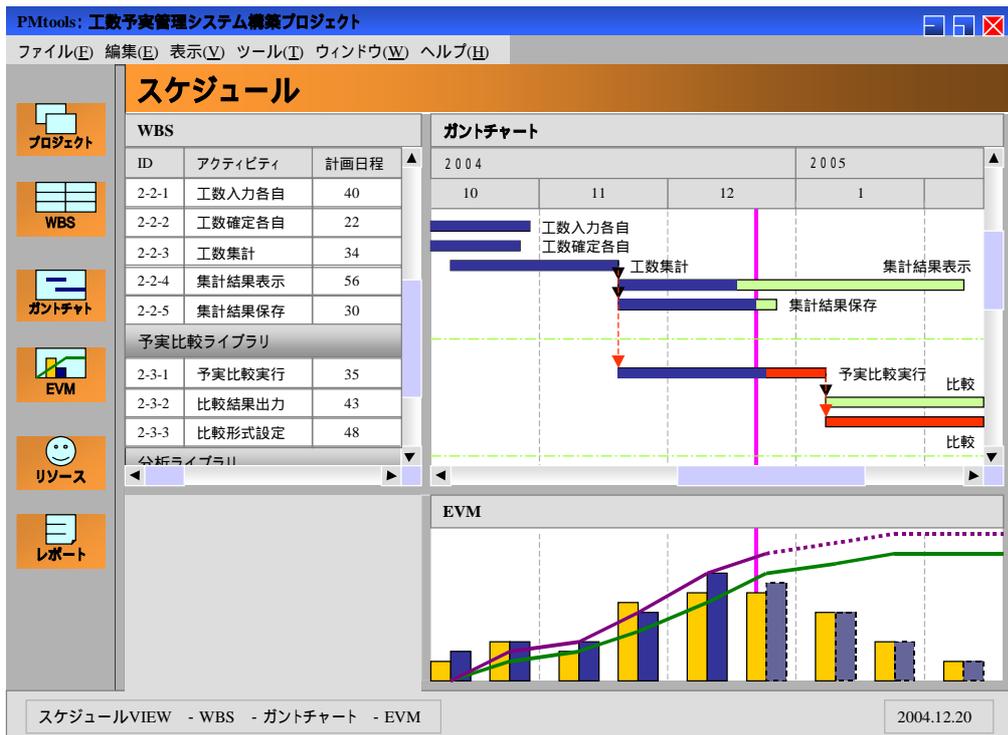


図 2-1

以下、主な市販ツールの名称および情報入手先を記す。(順序はとくに意味なし)

- ・ Primavera (Primavera、IT エンジニアリング)
http://www.ite.co.jp/products/primavera/primavera_v41/index.html
- ・ ARTEMIS7 (アルテミスインターナショナル)
http://jp.aisc.com/News/pr_040722
- ・ PMOffice (Systemcorp、IBMなど)
<http://www.systemcorp.com/jp/products/pmoffice/>
- ・ MS Projectを補完するアプリケーションパッケージ (ユーフィット社)
http://www.ufit.co.jp/pms/html/70_solutionp/10index.html
- ・ Process Director (NEC)
<http://www.sw.nec.co.jp/cced/processdirector/>

- Super-SCALE (NEC情報システムズ)
http://www.super-scale.com/
- PROMA-C (エスエムジー)
http://www.smg.co.jp/service/products/Proma-c/
- Compass Standard (シーバックス)
http://www.compass.co.jp/
- Serena (旧PVCS) Professional (セレナソフトウェア)
http://www.serena.com/jp
- nil xDTS (ニルソフトウェア)
http://www.nil.co.jp

(2) WBS とガントチャートを統合したツールを自作する。

自前で作るか、SI ベンダなどに構築を依頼する。カバーする機能にもよるが、それなりの開発規模になる。

かゆいところに手が届く使いやすい仕様が実現でき、バージョンアップや保守サービスにも柔軟に対応できるというメリットがある反面、かなりのコストがかかるため、得られる効果とのバランスを説明して予算を確保するのが難しい、そもそもそんな開発を行なう余力がない、などのデメリットがある。

図 2-2 に例を示す。

							2003年												
							8月												
NO	項目	担当 当社	P V (MD)	A C 実績 工数	進捗 率	状況	1	2	3	4	5	6	7	8	9	10	11	12	13
							金	土	日	月	火	水	木	金	土	日	月	火	水
0	マイルストーン・イベント																		
	基本設計書納品																		
	人事打合せ																		
	週次打合せ		19.5	1.5	8%	1.5MD													
	月次全体会		4.5		0%	1.5MD													
	店舗ヒアリング	全員	4	2.6	65%														
1	打合せ準備&議事録作成																		
	8/5	全員	1	1	100%														
	8/12	全員	1	1	50%														
	8/19	全員	3		0%														
	8/26	全員	2		0%														
	9/2	全員	2		0%														
	9/9	全員	2		0%														
	9/16	全員	2		0%														
	9/24	全員	3		0%														
	9/30	全員	2		0%														

図 2-2

(3) EXCEL (に代表される表計算アプリ) に一本化する。

ガントチャートではなく日付ベースの進捗管理とし、WBS と同じ EXCEL シートで、ワークパッケージごとに確認できるようにする。出来高パーセンテージもあわせて記入したり、過去や未来の予定および実績を週次で一覧表示したりするなど、視認性を上げる工夫によりガントチャートを使わないことによる抵抗をある程度抑えることができる。

簡易で使いやすくカスタマイズも容易というメリットがあるが、どうしてもガントチャートでの報告でないとダメ、という上位管理者がいたりすると日の目を見ないかも知れない。

図 2-3 に例を示す。

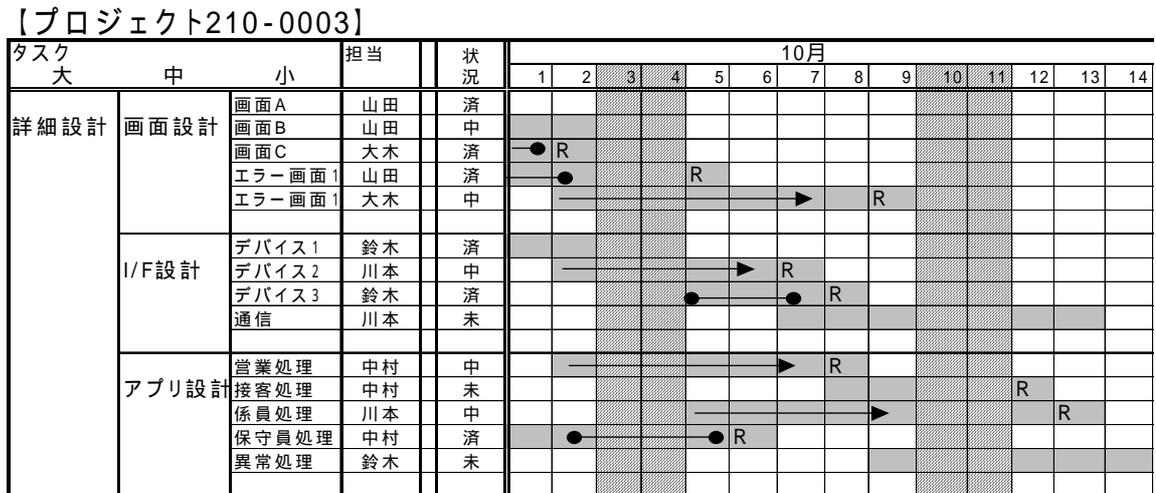


図 2-3

(4) MS-Project (に代表される進捗管理ツール) に一本化する。

MS-Project の「タスク」は階層化できるため、WBS およびワークパッケージを表現させることも可能である。純粹に WBS とガントチャートの一元化を図るだけであれば、最も簡易な方法と言える。

すでに MS-Project を使っている部署ではハードルがきわめて低いというメリットがある。一方、ワークパッケージに付随する承認ルートやメトリクスなどのプロパティ、参照するドキュメントなどへのリンク、それにワークパッケージの変更履歴などまで一元管理することは難しい。

図 2-4 に例を示す。

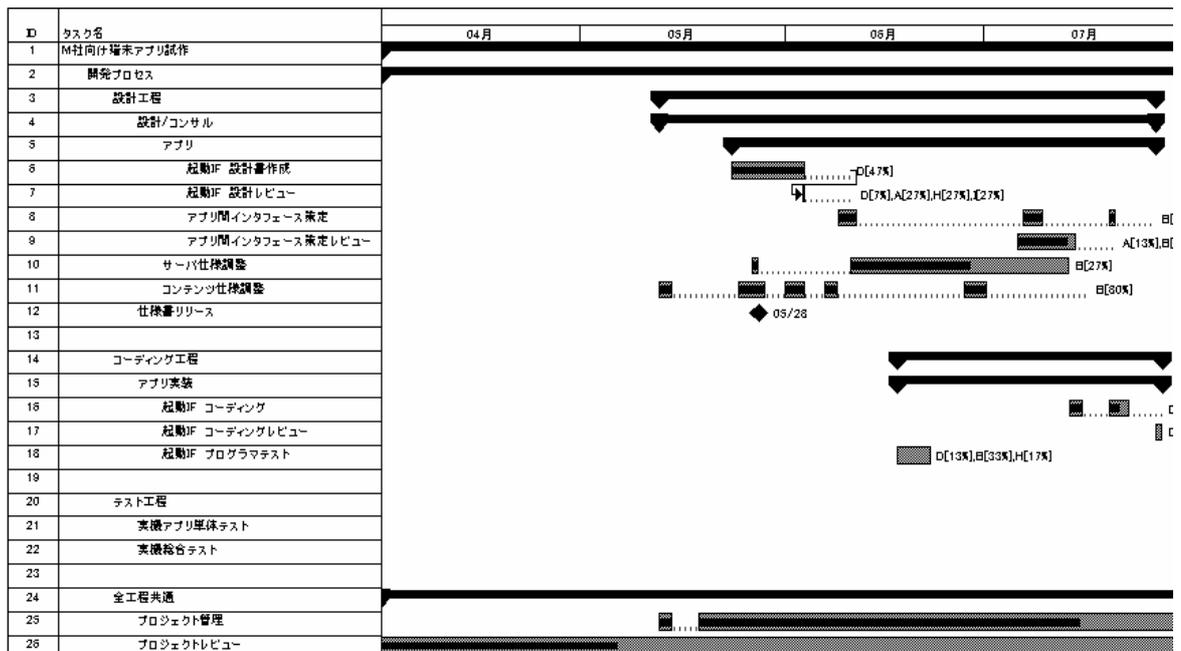


図 2-4

以上、当該部署およびスタッフ部門でメリットとデメリットを比較検討し、それぞれの状況に合わせてベストな解決方法を選択するのが良い。

2.4 補足（進捗管理について）

本稿では WBS とガントチャートを一元化するための方法にフォーカスを当てたが、それはあくまでも枠組みであり、導入しただけで効果が出るわけではない。有効活用し運用の実をあげるためには、WBS を用いた進捗管理について正しく理解する必要がある。

適切に構築された WBS は網羅性と非重複性があるため、進捗率の数値に対する信頼性が高い。ワークパッケージの粒度を一定内に揃えることにより、さらに精度が向上するとともに遅延した場合のワーニングが早く、リカバリが容易になるといいうメリットがある。

ところが、それでもなお遅れるプロジェクトは遅れる。なぜだろうか。組織により事情が異なるが、一般的には以下のような点について進捗管理に関する認識が不足しているのではないかと考えられる。

(1) 工数見積りは確率的な値である

工数は実際には、一般的に見積り値として使う最頻値をピークとした山型の確率分布となる。ただし左右対称ではなく、左側（工数が少ない方）は広がりがなくすぐにゼロとなるのに対し、右側（工数が多い方）はピークの数倍までブロード状に広がりを持つ（ベータ分布と呼ぶ）。つまり確率的な平均は最頻値ではなく、より工数が増える方向にシフトすることになる。

したがって、プロジェクトマネージャは各ワークパッケージに予定日数の数倍遅

延する潜在リスクが（一定の確率で）あることを認識し、そのリスクの影響度を算定、および顕在化した際の対策を用意しておくべきである。

(2) クリティカルパスを重点管理する

ガントチャートはたんなる線表ではない。ワークパッケージごとの依存関係を明確にし、どの部分の遅延がどこに影響するか、リカバリの時間的余地がどれだけあるか、ということを表すことができる。

プロジェクト全体にわたるリカバリ余地が最も少ないワークパッケージの順序連携(パス)をクリティカルパスと呼ぶ。これはそれぞれのワークパッケージの最早、最遅開始日および終了日を計算し、その差分であるフロートを求めることで確定できる。

クリティカルパスの遅延はプロジェクト全体の遅延を決定づけるため、きわめて重要であり、重点的に管理されなければならない。逆に言えば、クリティカルパスを使ってプロジェクトをコントロールすることができる。プロジェクトが進行するにしたがい、クリティカルパスが別のパスに移ることもあるので注意が必要である。

(3) 進捗管理を大局的にとらえる

進捗管理を局所的にとらえているマネージャが多い。どのツールも基本的にはプロジェクトのある時点の状況を切り取って（つまり局所的に）示してくれる。予定より遅れているワークパッケージがあれば、リソースを追加してキャッチアップするか、当該ワークパッケージと依存関係にある他のワークパッケージのスケジュールを調整すればよいと考えるであろう。しかし、ここまでは局所的な観点からの言わば対症療法にすぎないのである。

進捗管理は大局的にとらえなければならない。前述の通り、プロジェクトに変更はつきものである。ワークパッケージによっては複数回の変更が入ることもあるし、遅延に次ぐ遅延というケースもあるだろう。同じ担当者があちこちで遅延を引き起こしている例も珍しくはない。これらの背景を、あるいは履歴をじゅうぶんに考慮しなければならない。それをせず、すべてのワークパッケージを一律にあつかえばまた同じ落とし穴にはまるだけである。これまでにどのような変更や遅延が発生したか、その原因は何だったか、対策は有効だったか、といったことを把握し、各々のワークパッケージの特性を理解した上で適切な対策を取る。大局的な視点からの進捗管理とはそういうことである。

なお、こうした視点からの経験則の一部を定式化し、コスト管理と統合した手法が EVM である、とすることができる。

以上のような進捗管理に関する正しい理解と、最適なツール選択はプロジェクト運営における車の両輪であり、いずれが欠けてもうまくはいかない。両方そろってはじめて安定したプロジェクト運営が可能となることを、プロジェクトマネージャや品質管理担当者は理解しておくべきである。

以上

3. Tips3 管理の一元化（WBS と課題管理）をどのように行うか

3.1 前提（想定する状況）

WBS を導入したが、WBS と課題管理を切り分けて運用してしまっているケース、問題が発生して追加作業が発生した際、WBS にどのように反映してよいのかわからないケースなど。

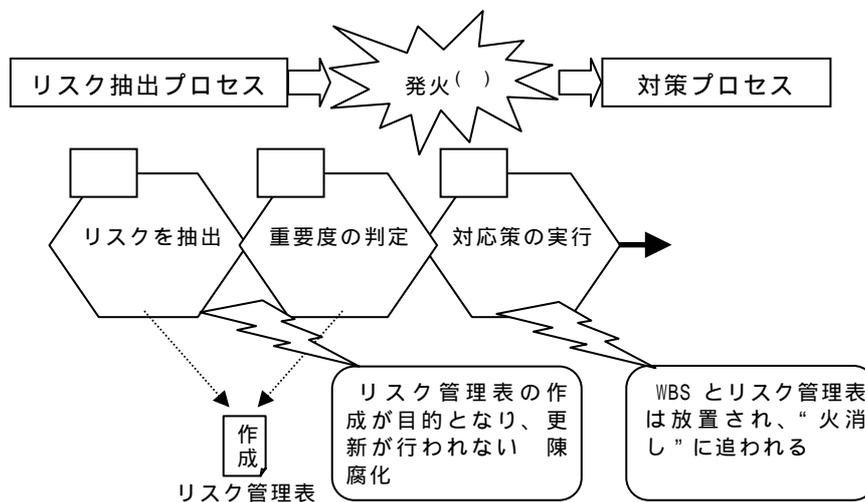
3.2 問題

現状は、WBS と課題管理を全く別物として運用しているプロジェクトがまだまだ多いと想定される（下図）。

このような運用をする場合、以下のような問題がある。

- ・プロジェクト開始時に想定されるリスクを抽出するが、リスク管理表を作成したことで満足してしまう。
- ・プロジェクト途中でリスク管理表が更新されないため、リスク発火^()時点では内容が陳腐化しており、対応策を実施する際に使い物にならない。
- ・プロジェクト開始時にはWBSに基づいて実施しているが、ひとたびリスクが発火^()すると、WBSとリスク管理表は放置され、以後は無計画な“火消し”に追われる。

現状の課題管理プロセス



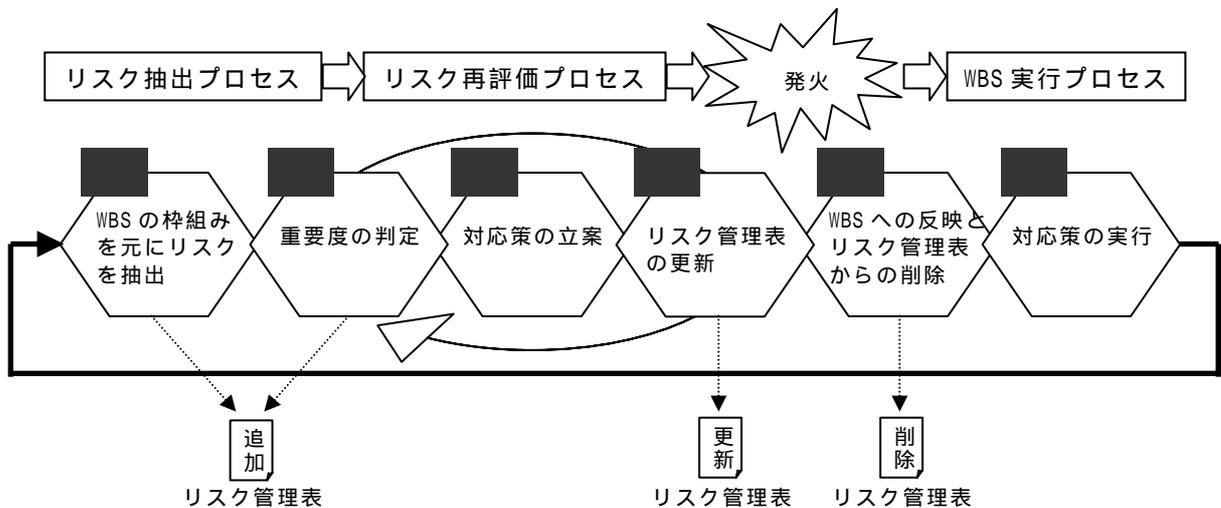
()発火：想定していたリスクが実際に発生すること

3.3 解決方法

前述の問題を解決するために、以下のようなステップで WBS の枠組みを基にした実践的な課題管理の手法を紹介する（下図）。

課題管理のプロセスでは、～ のプロセスで顕在化していない課題（以下、リスクと記載がある場合はこの意味）に対して、と のプロセスでは、発生済みの課題（以下、問題点と記載がある場合はこの意味）に対して、WBS を活用して対処できるようアプローチを行う。

管理を一元化した課題管理プロセス



（プロセス ） WBS の枠組みを基にリスクを抽出

WBS の大項目または中項目レベルの内容に対し、それぞれリスクを抽出する。

（プロセス ） 重要度の判定

抽出したリスクに対して、影響度と確率を加算（または積算）し、重要度を判定する。

影響度 = 実際にその事象が発生した場合の被害の大きさ、確率 = リスクが顕在化する可能性の度合いとし、3～5 段階を目安に設定する。

表 1 . リスク管理表例

WBS 項目名	リスク	影響度	確率	重要度
契約				
要件定義	要件定義書が顧客に承認されるのに時間がかかる	2	1	3
	顧客の社内での意見調整が出来ず、要件が固まらない	2	5	7
外部設計				
内部設計				
プログラミング				
単体テスト				
ハードウェア導入				
結合テスト				
システムテスト				
ユーザー教育				
検収				
システム稼働				

重要度が高い項目は、事前に対応策を立案して「対応策一覧表」などの形で明確化しておくが良い

(プロセス) 対応策の立案

あらかじめ設定した値を超える重要度となったリスクに対して、対応策を立案する。

表 2 . 対応策一覧表例

リスク	要因	対応策
顧客の社内での意見調整が出来ず、要件が固まらない	<ul style="list-style-type: none"> ・ 客先キーマン不在 ・ 客先キーマン複数 ・ 客先方針があいまい ・ P M の経験不足 	<ul style="list-style-type: none"> ・ 顧客責任による免責を契約書に明記 ・ 同上 ・ 要件定義フェーズを別契約とする ・ 打合せ初回は経験豊富な さんが同行する

(プロセス) リスク管理表の更新

リスクについては、プロジェクト開始前の想定から立ち消えになったか、逆に発生率が増加したか、あるいは対応策が必要になるほど悪化したのか(問題点に変化)状況を常に把握する。

すなわち、プロジェクト期間中は、プロセス からプロセス までのサイクルを常に繰り返すこととなる。

(プロセス) WBS への反映とリスク管理表からの削除

リスクに対する対応策が必要になった場合、あるいはリスクとして想定していなかった問題点が発覚した場合、WBS のワークパッケージとして含め、WBS を更新する。

リスクに対する対応策を WBS に反映した後、リスクは現実のものとなり、問題点に変化して WBS のワークパッケージに含まれているので、リスク管理表からは削除する。

(プロセス) 対応策の実行

対応策は、通常の WBS 内の作業として実行・管理する。

3.4 補足

3.5.1 WBS と課題管理の一元化が必要とされる背景について

プロジェクトマネジメントの主要項目である課題管理の重要性は、プロジェクトマネージャを筆頭とするプロジェクト関係者の全てが認識していると言っても過言ではない。しかし、実際のプロジェクトにおいて、課題管理の手法を体系的に理解し、実践しているプロジェクトマネージャは限られている。

この問題は、プロジェクトマネージャ個人に限った話だけではない。組織レベルにおいても、課題管理の重要性をひたすら繰り返すだけの管理部門 (PMO、品質保証部門) と、日々の業務に手一杯との言い訳で改善に手をつけない現場のプロジェクトメンバーとの温度差がそれである。特に、前者は「CMM や ISO の審査にパスする最低限の基準をクリアするため。」といった組織的な動機付けなどの理由によるところが大きい。

すなわち、個人においても組織においても、“プロジェクトを成功に導く”という本来の目的が置き去りにされ、表面的な課題管理を行っているに過ぎないという背景がある。

3.5.2 リスクを定期的に再評価するプロセスサイクルについて

一般的に、リスクはプロジェクト期間中に影響度や確率が変化する。このため、リスク一覧表の重要度は常に更新することが望ましい。

冒頭に示した課題管理のプロセスサイクルは、このようなプロジェクトマネジメント特有の状況を考慮し、一定期間 (または工程) ごとにサイクルを回してリスクを再評価し、プロジェクトが失敗する可能性を最小限にとどめるためのものである。

再評価の内容として、以下が挙げられる。

- (1) リスクのどの項目が立ち消えになり、新たにどのようなリスクが発生したか
- (2) リスクに対する対応策が変化したか
- (3) リスクに対応するためのリソースは十分か、対応時の費用および期間に問題ないか

以上

4. Tips4 変更をどう管理するか

4.1 前提（想定する状況）

- ・ WBS の修正が必要となったケース
- ・ 新たな WBS の定義が必要となったケース
- ・ 作成した WBS の一部が不要となったケース

4.2 問題

プロジェクトを進めていく中で変更は往々にして発生する。その際、単に最新の WBS に置き換えていくという手法では現在の状況しか把握できないため、過去の履歴や当初計画との差異比較などが困難になる。一方、当初の計画に固執しすぎてもプロジェクトをマネジメントするという本来の目的に支障をきたすことになる。

また、変更に対して、個々のプロジェクトリーダーが異なるアプローチをとっている場合は、プロジェクト完了後も組織としてのノウハウが蓄積していかない。そのため、どのように WBS を更新していくかについて、統一した対応方法を検討していく必要がある。

4.3 解決方法

上記問題に対して以下のような対応を行う。

最新の状態を正しく把握するためには、変更が発生するたびにその都度 WBS を更新していく必要があるが、その際、過去の履歴を確認するため、不要となった WP であっても削除は行わず、追加のみで更新していくことが有効である。つまり一度作成した WP は最後まで残しておき、新たな WP を追加していくことで最新の計画に更新していくという方法である。

また、変更によって影響を受けた WP はその内容（変更理由、要件、影響範囲、期間）を正しく確認できるようにしておく必要がある。WBS によってそれらの情報を一括して管理するか、もしくは、管理表などにリンクして適切に参照できるように管理する。

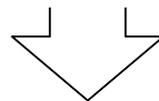
以下に Excel にて管理を行う例を挙げる。

EXCEL での管理

		追加された WP は新たに定義する		不要となった WBS も削除しない					
担当者	開始日	終了日	変更NO	備考					
4.業務A	1.新事業コア	1.フレームワークの検討	佐藤	9/1	9/15				
		2.システム機能定義		9/5	9/20				
		3.概念DB設計		9/8	9/20				
	2.機能A	1.ユーザ要望書	佐藤	9/20	9/25				
		2.定義書レビュー		9/26	9/30				
		3.レイアウト作成	中井	10/1	10/14				
		4.項目定義	中井	10/15		11			追加
		5.桁数調整	中井	10/15		11			追加
	3.機能B	1.ユーザ要望書	高山		10/20	12			削除
		2.定義書レビュー	高山		10/20	12			削除
		3.レイアウト	高山		10/20	12			削除
	3a.機能C	1.ユーザ定義書	高山	10/20		12			追加
		2.定義書レビュー	高山	10/20		12			追加
		3.レイアウト	高山	10/20		12			追加
		4.出力制御	高山	10/20		12			追加
	4.明細表示	1.機能定義	1.ページA	佐藤	10/23				
2.ページB			佐藤	10/23					
2.レイアウト									

変更は、新たな WBS を定義する

変更管理表にて
変更の詳細を管理



No	済	内容	受付日付	依頼者	依頼書No.	影響範囲		作業着手		見積工数		影響(変更)範囲			備考	
						WBS No	WBS名	開始	終了	小計	内訳	期限	資料	PGM修正		総合リスト
11		コードの追加 (コード追加により「機能A」 に新たなWPを定義する)	2004/10/15	鈴木	11	4.2.4	項目定義			0.8	0.5	10/31				
						4.2.5	桁数調整				0.3	11/10				
12		出力機能の変更 (機能要件の変更のため、 「機能B」を削除し、新たに 「機能C」として定義する。)	2004/10/20	鈴木	12	4.3	機能B									
						4.3a	機能C			0.5	0.5	11/14				
合計										人月						

仕様変更の概要と
その受付日付

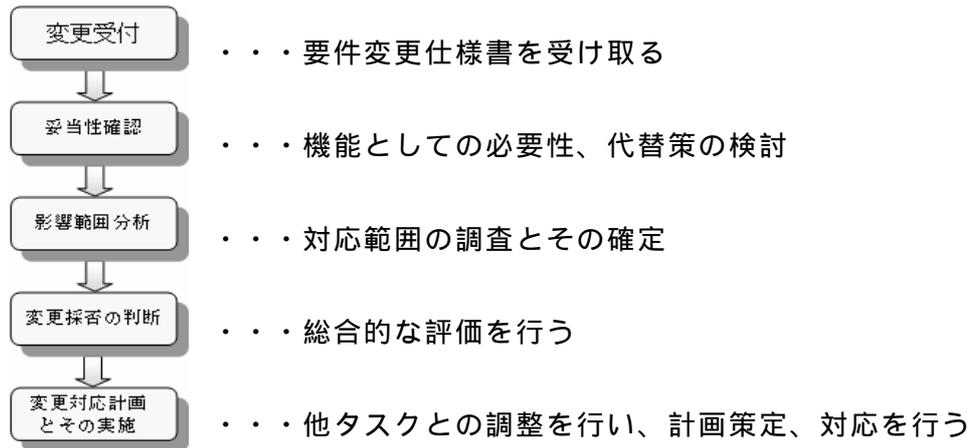
影響する
WBS を記述

詳細をそれぞれ作成

4.4 補足

ここでは WBS の変更を前提としたケースについて取り上げたが、まず変更要求が発生した際には、その影響範囲を把握して対応を十分に検討する必要がある。その結果、WBS の修正を行う場合には、上記で示した通り当初 WBS を活かす形での管理を推奨する。

変更管理のプロセス



以上

5.4 補足

(1) E V Mを使う上での一般的なメリット等

メリット

- 遅れの状況把握が可能
 - ・ 予定通りの工数を投入していて、遅れている
 - ・ 予定通りの工数を投入できなくて、遅れている
 - ・ 予定以上の工数を投入しているのに、遅れている
- 計画時の実現性を検証可能（P VはS字カーブを描いているか）

P Vは通常ではS字カーブの形になる。これは、必要とされる工数はプロジェクトの開始時は少なく、中盤で増加し、終盤は収束するためである。従って、なだらかなS字を描いていない場合は、特定の時期に作業が集中している可能性があるため見直しが必要である
- 開始の早い段階での将来予測が可能
- 担当者間のコミュニケーションプロセスの向上

留意点

- E V Mはスケジュール進捗を把握する手法
品質管理およびリスク管理も、並行して行うことが必要
- プロジェクト担当者、作業員、E V M実施担当者間で進め方について合意しておくことが必要（進捗率の計上ルール等）
- 計画変更時の速やかな反映が必要（変更管理）

教訓

プロジェクトの遅れは、ある日突然には発生しない。徐々に遅れていく（傾向の把握が重要）

E V Mを利用することにより、早期に、かつ定量的に傾向を把握可能

以上

6. Tips6 メトリクスをどう取るか

6.1 前提（想定する状況）

- ・ WBS を導入したが、どのようなメトリクスを取ればよいか分からないケース
 - ・ WBS の導入に伴いデータが蓄積されてきたが、うまく活用できていないケース
 - ・ WBS 導入に伴うメリットを、定量的に示すことを問われているケース
- など。

6.2 問題

WBS を導入することによって、メトリクスが取りやすくなる、ということは理解できるが、具体的にどのようなメトリクスを取ればよいか分からない、という問題がある。EVM を利用した進捗管理などでは、WBS をベースに具体的にどのようなメトリクスを取って、どのように使えばよいかということが確立されているが、それ以外ではどのようにメトリクスを取りどのように活用するかについて、整理されたものがあまりない。

また、WBS 導入に伴うメリットを定量的に示すように問われる場合があるが、どのようなメリットがありどのように効果を示せばよいか分からない、という問題を抱えている場合もある。

6.3 解決方法

上記のような問題に対して、ここでは、メトリクスを設定し活用する具体的な例（下記5つ）を示すことで解決の一助としたい。

- 見積りに関するメトリクス
- 進捗管理に関するメトリクス
- WBS の粒度に関するメトリクス
- 後戻り工数の削減に関するメトリクス
- 計画変更に関するメトリクス

メトリクスを決め活用するための手順としては、下記の Step1～Step4 の方法をとるとよい。以下に示す具体例についてもこの Step に沿って整理している。

(Step1) メトリクスを取る目的を明確にする（ここが最も重要）。

(Step2) その目的が達成できているかどうかを判断することが出来るメトリクスを定める。

(Step3) そのメトリクスを得るためにプロジェクトで測定しておく必要があるデータを決める（WBS 導入によって既に蓄積されているデータがあれば、それを利用できる）。

(Step4) メトリクスを活用する方法を定義する。

6.3.1 見積りに関するメトリクス

目的	見積りの予実差を小さくしたい。
メトリクス	計画と実績の工数の差異
測定するデータ	各 WP の計画工数 各 WP の実績工数
活用方法	<p>WBS で展開されている各 WP に対する工数を見積り、その工数を合算したものがプロジェクト全体の見積り工数になる。完了時 WBS の実績工数も同様に計算する。</p> <p>その差異を把握し、その原因分析を行い次回のプロジェクトの見積り時にフィードバックすることで見積りの精度を高めることができる。</p> <p>原因分析においては、継続的に収集した過去のプロジェクトのデータを利用したり、WBS のレベル毎の計画と実績の工数の予実差を見たりして、原因を的確に分析することが可能になる。</p> <p>(Tips7 見積りにどう使えばよいのか 参照)</p>

6.3.2 進捗管理に関するメトリクス

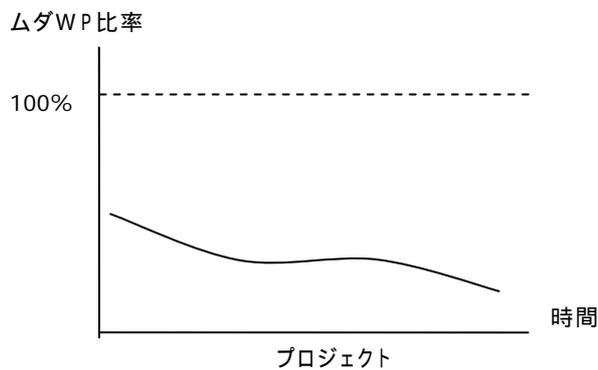
目的	進捗の遅れを把握し、必要な対策をとりたい。
メトリクス	スケジュール差異 (SV)、スケジュール効率指数 (SPI) コスト差異 (CV)、コスト効率指数 (CPI) 完了時コスト差異 (VAC)、乖離率 等
測定するデータ	各 WP の計画コスト、実績コスト 各 WP の計画工数、実績工数
活用方法	<p>プロジェクト全体または WBS で定義した各レベルでの進捗を、EVM の各メトリクス(SV、CV 等)で把握する。</p> <p>把握した遅れ (乖離) に応じて、必要な対策を打つことができる。</p> <p>(Tips5 EVM にどうつなげるか 参照)</p>

6.3.3 WBS の粒度に関するメトリクス

目的	WP をプロジェクト管理に適した適切な粒度に揃えたい。
メトリクス	WP の基準日数と計画日数の差異 (差異の平均) WP の基準日数と実績日数の差異 (差異の平均)
測定するデータ	各 WP の計画日数 各 WP の実績日数
活用方法	<p>WBS を用いたプロジェクト管理を効率的に行うために、WP の粒度をある一定の水準に揃えることが望ましい。</p> <p>そのため基準として定めた粒度 (WP の日数) と WP の計画または実績の日数の差異を把握する。</p> <p>その差異を分析し次のプロジェクトにフィードバックすることで WBS としての質を高めていくことができる。</p> <p>(Tips1 WP の粒度をどう揃えるか 参照)</p>

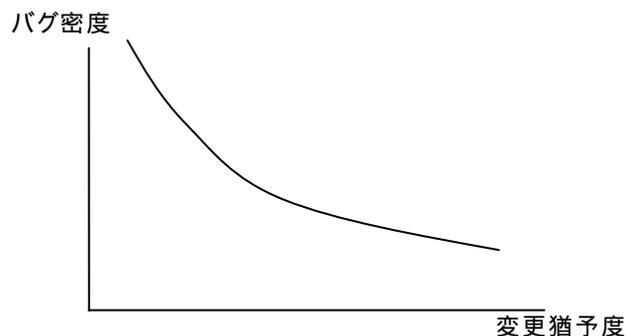
6.3.4 後戻り工数の削減に関するメトリクス

目的	後戻り工数を減らしたい。
メトリクス	全 WP の工数に対するムダ WP 工数の比率
測定するデータ	各 WP の工数 プロジェクトの中で結果としてムダになった WP の工数
活用方法	プロジェクトの中で結果としてムダになった WP を識別し、プロジェクト全体に対するムダ WP 工数の比率を把握する。 継続的に過去プロジェクトのムダ WP 比率を把握して（下図参照）その原因の分析を行う。その結果を次のプロジェクトの計画へ反映することで、後戻り工数を減らしていく。



6.3.5 計画変更に関するメトリクス

目的	変更への対応を計画的に行う。
メトリクス	変更猶予度（実施の何日前に計画が変更されたか）
測定するデータ	WP の変更日 （変更された）WP の開始日 （変更された WP に関連する機能の）バグ密度
活用方法	突発的な計画変更が多い場合、バグが多くなると仮定する。変更された WP について、その変更日と実際の WP の開始日の差異の平均を変更猶予度と定めて、変更された WP に関連する機能のバグ密度との相関のデータを継続的に蓄積する（下図参照）。 それをもとに計画変更は何日前までに行うべきといった指標を設定し、突発的な変更が入らないようにコントロールしていく。



6.4 補足

(1) メトリクスを取る負荷に配慮する

メトリクスを取る際には、計測するための負荷（手間）が障害になるというケースが多く見られる。さまざまなメトリクス活用を狙って、多くのデータを細かく取ろうとすればするほど、プロジェクトのメンバにかかる負荷は高くなる。

これに対しては、WBS を使い管理するときに、データ収集や分析なども効率的に出来る仕組みを備えたツールを利用することなどで、なるべく新たな手間をかけずに自然にデータが取れるような配慮が欠かせない。

また、何でもメトリクスとして取るというのではなく、目的に合ったもののみを取るようにして余計な手間をかけないようにしたい。プロジェクトのメンバに対しては、それを取るによってどのような活用が出来るかをきちんと説明して理解を得ることも忘れてはならない点である。

(2) メトリクスを取るというのは手段である

本稿ではメトリクスをどう取るかという側面から論じてきたが、本来はプロジェクトを正しく把握し制御&管理するために、データ（メトリクス）を手段として使うというように考えたい。「こういうデータが取れそうだから何かに使えないか」とメトリクスありきで考えるやり方もあるが、「プロジェクト管理に関するあらゆる局面において、データ（メトリクス）を基盤としてよりよく制御できる方法がないか」という考え方をしていくとよいと思われる。

以上

7. Tips7 見積りにどう使えばよいのか

7.1 前提

開発現場で常に大きな課題になっているのが見積りである。

ここでは、次のようなケースを想定している。

WBS に基づいて管理を行っているが、経験や勘での見積りを行っているので精度が悪い。よって、WBS を用いて見積りを行いたい具体的などのようにしていいかわからない。

7.2 問題

見積りを行う際にボトムアップ的なアプローチとして WBS による積算により見積りする方法がある。ただし、WBS を用いた見積りを行うとなると下記のような疑問点にぶつかり、導入の妨げになることが考えられる。

- ・ WBS で何が見積れるのか？
- ・ 見積りにどう利用すればよいのか？
- ・ WBS で見えていない部分はどを見積ればよいのか？
- ・ プロジェクト計画時の見積りに使えるのか？

7.3 解決方法

上記各疑問点に対して解決方法を提案する。

7.3.1 WBS で何が見積れるのか？

工数の合計値、またはスケジューリングが終わっていれば期間が見積ることが可能である。

7.3.2 見積りにどう利用すればよいのか？

WBS で展開されている各ワークパッケージに要する工数を算出し、その各ワークパッケージに要するものを合算したものがプロジェクト全体の見積りになる。

また、第 1 レベル、第 2 レベルでどれくらい工数がかかるのかといった見積りも可能である。

WBS を作成したことで作業内容が明確になっている。この作業内容により、作業の難易度を推測し、見積りの際に作業難易度を考慮することもできる。

7.3.3 WBS で見えていない部分はどを見積もればよいのか？

WBS を作成することにより、何が見えていないかが明確になるはずである。

見えている部分の見積りをしたうえで、見えていない部分は今までのプロジェクトの状況により推測し見積りが行える。継続的な WBS の利用により、見えていない部分の推測能力が成熟してくることが予想される。(詳しくは補足 7.4.1 参照)

当初の見積りが大幅にずれたことで上司に報告する状況は珍しくない。このよ

うなときに、WBS を用いて見積りを行っていたら説得しやすいのではないだろうか。例えば、「ここまででは仕様が確定していたので WBS を作って、それに基づき見積りを行いました。ただ、未確定だったのが WBS のこの部分です。見積りの時には考慮していたのですが、思ったより未確定部分が大きかったのでリソースが足りなくなりました。」というような言いまわしができる。「何とか見積りました」と言っても、説得できるとはいいがたい。

7.3.4 プロジェクト計画時の見積りに使えるのか？

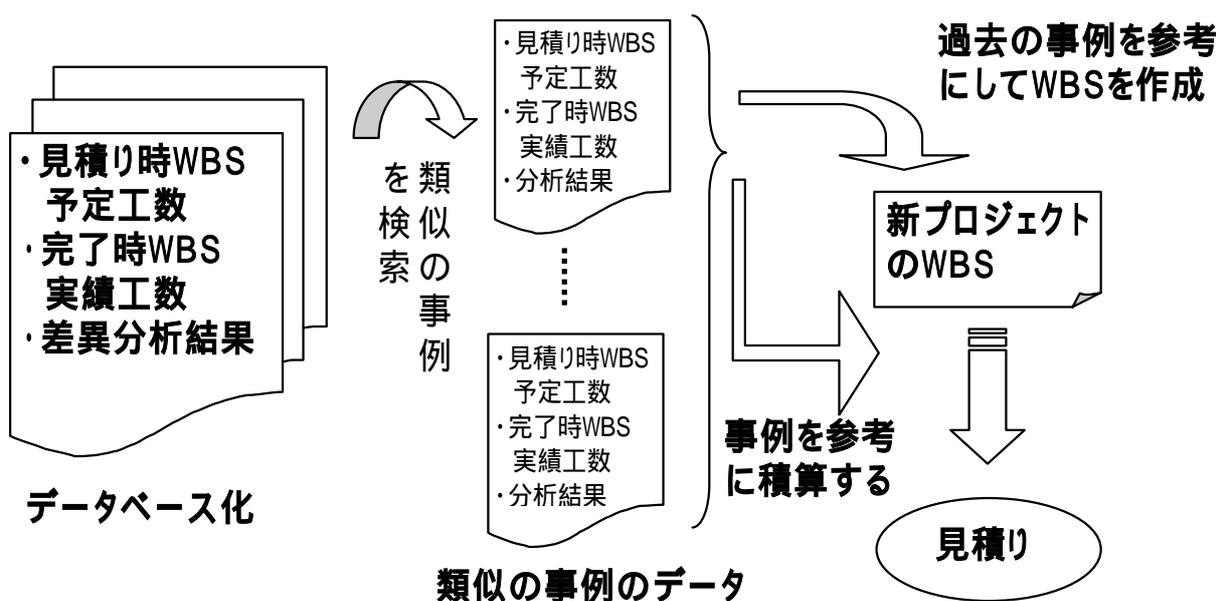
プロジェクト計画時に WBS の作成が可能であれば使うことができる。ただし、通常プロジェクトでは計画時に WBS が作れないことがほとんどである。このような場合は、トップダウン的なアプローチでの見積りを行う。(詳しくは補足 7.4.2 参照)

7.4 補足

7.4.1 継続的な WBS の利用について

WBS を用いた見積りを行う際に、過去の事例を参考にすると精度が良くなるであろう。過去の事例を継続的に蓄積し、利用する仕組みを作ると見積りの精度が成熟してくることが考えられる。具体的に蓄積するデータとしては、「見積り時の WBS」と「予定工数」、「完了時の WBS」と「実績工数」、「差異分析結果」が考えられる。(下図参照)

ここでの「差異分析結果」とは、「予定工数」と「実績工数」の間になぜずれが生じたのかの原因分析を行なった結果である。ここで考えられる原因としては、「WBS の作成がうまくできていないことによるずれ」と「各ワークパッケージの工数見積りがうまくできていないなどの積算によるずれ」などである。このような、原因を的確に抽出し次プロジェクトへ反映させることがよい。

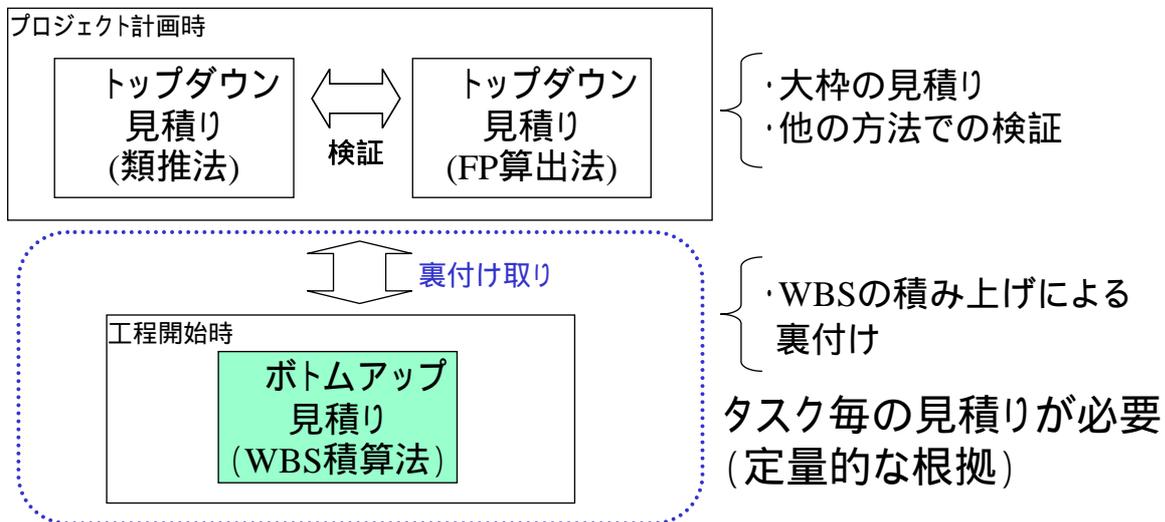


7.4.2 トップダウン的な見積りとの併用

トップダウン的な見積りには類推法と FP(ファンクション・ポイント)を用いた算出法がある。類推法とは過去の類似した事例から工数を推測する方法である。また、FPを用いた算出法とは、FPよりシステムに含まれる機能数を計算し規模を求め、これを用いて開発の生産性から工数を算出する方法である。

このようなトップダウン的な見積りは大枠での見積りであり、あくまでも概算でしかない。このような見積りの値でプロジェクトを進めるとずれが大きくなることは明らかである。

そこで、WBS が作れた時点で詳細な見積り(ボトムアップ的な見積り)を行い、トップダウン的な見積りとボトムアップ的な見積りをつき合わせて、裏付けを取ることが有効である。(下図参照)



類推法:過去の類似事例から工数を推測

FPを用いた算出法:FPを計算し、開発生産性から工数を算出
大枠での見積り

WBS積算法:WBS展開した項目ごとに工数算出、積み上げ
詳細な見積り

以上