

# CIO のための IT プロジェクト経営

## IT Project Management for Chief Information Officer

講演論文

蓮尾克彦 (Hasuo, Katsuhiko)

株式会社テプコシステムズ

hasuo-katsuhiko@tepsys.co.jp

**Abstract:** Not project manager, but CIO must be responsibility for to avoid risks and to change project plan. Monitoring performance is essential of management by Balanced Score Card, then "project performance" review by EVM is useful method for CIO and project manager. Inspect rare case in Japan, CIO decided to suspend 2 billion yen project, with what-if scenario by SD, and investigate how to improve IT development productivity.

キーワード: EVM, CMM, TOC, Critical Chain Scheduling、ブルックスの法則、パーキンソンの法則、すり合わせ

**要旨:** 開発が計画どおりに進まない時、ITプロジェクトの計画を変更するのか、或いは中断するかの決定の役割は経営者であり、企業に必要な内部統制である。プロジェクトマネージャ (PM) のマネージングの対象は開発プロセスであり、このエンジニアリングプロセスをマネージするのはオペレーショナルマネジメントであるが、経営者のマネージング対象はプロジェクトマネジメントプロセスであり、ストラテジックマネジメントである (キャプランの管理のブループ)。

ユーザ企業 (調達側) のストラテジックマネジメントの責任は CEO ではなく、CIO にある。

Work Break Down Structure (WBS) は有効であるが、経営者は「コスト」と「納期」しか理解できない。マネジメントプロセスをモニタリングする為には Earned Value Management (EVM) が必要であるが、経営者が IT プロジェクトに参画していない日本では経済産業省の推奨にもかかわらず、EVM が普及していない[1]。

EVM を導入するには Capability Maturity Model (CMM) [2] のレベル 4 以上で要求されている計画値の「実績の蓄積と評価」のプロセスが必須であるが、CMM レベル 4 以上のベンダー企業の少ない日本では EVM の普及を妨げている原因でもある[3]。CMM は 1995 年頃日本に紹介されたが、「知見」が蓄積されず、CMM の導入に失敗した IT ベンダーは多い。CMM を成功させる

には「標準生産性指標」を生産性の要素であるイネイブラーを SD で検証し、設定する方法が有効である。

生産性を向上させる為には管理プロセスだけではなく、効果的な製造プロセスとして、Theory of Constraint (TOC) にもとづく Critical Chain Scheduling and Buffer Management (CCS&BM) が有効である[4]。CCS&BM は従来型の連続型の Production 生産と非連続型の Project 開発を統合したアプローチであり、「組み合わせ」ではなく「すり合わせ」のエンジニアリングプロセスである。CCS&BM を成功させるには、パーキンソンの法則[5]を排除する「すり合わせ」の為のバッファマネジメントのテクニックと確率 50%での計画作成がポイントになる。

確率 50%でのスケジューリング作成にはメンバーとの合意が必須であり、その際プロジェクトエバリュエーションの為のツールとして SD は効果的である。

### 1. IT 開発プロジェクトにおける内部統制 (JSOX 法)

#### 1. 1 企業経営と IT 投資の統治 (ガバナンス)

成長した企業の設備投資の 1/3 は IT 投資であり (米国企業の IT ポートフォリオ[6])、経営における IT の重要

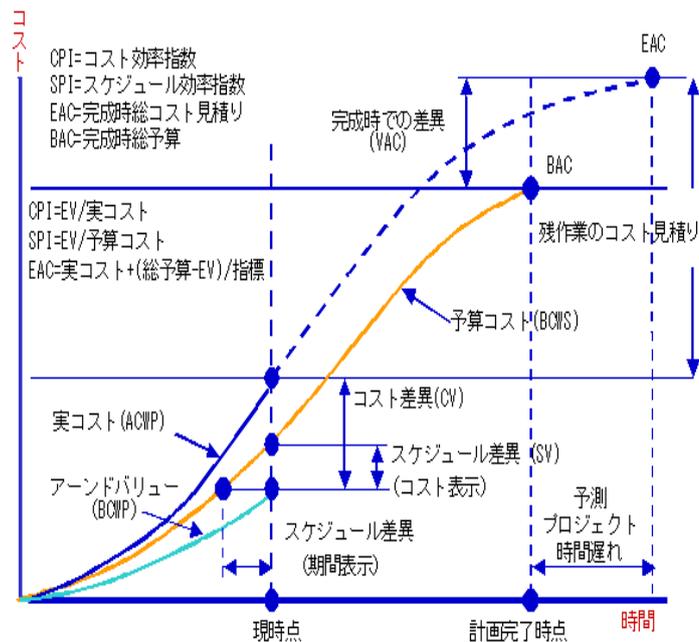


図1 研修会社のEVMの説明 (資料: プロシード社)

性は高まっていると同時に、増大する現場からの要求は増え続ける軍事費と同じであり、それに対処する為のITガバナンスの定着が企業経営の課題である。IT投資が制約される経営環境下で、システム開発の失敗で顧客の信頼を失い、業績悪化の経営責任を問われた例や、ITプロジェクトの失敗による損失で業績の下方修正を余儀なくされ株主の信頼を失ったIT企業経営者が多い。内部統制とは、発生したリスクに対してITプロジェクトを継続するか、或いは中断するかを決断を経営者がするプロセスである。

### 1. 2 IT統治のベストプラクティスとEVM

企業のIT投資の統治の為のベストプラクティスとしてControl Objectives for Information and related Technology (COBIT [7])があるが、日本では未だ定着していない。発注側の企業はCOBITの中のベストプラクティスの「IT資源調達」は必須であるが、米国国防省のような本格的な調達プロセス[8]を実現するには労力を要する。

「内部統制のモニタと評価」だけを最優先で導入し、その際、EVMでの統制を発注先のベンダーに要求することが内部統制プロセス確立の第1歩である。発注側、提供側の双方の経営者がITプロジェクトをモニタリングし、コントロールする為にはEVMのプロセスが必須であるが、EVMの定着にはシステム開発の特性を理解し、プロジェクトエバリュエーションツールとしてSDを活用することで簡便に定着させることは可能である。

### 1. 3 ITプロジェクトの開発特性

経営者による内部統制とは、妥当な計画を作成する為のプロセスと作業をモニタリングする2つのプロセスに経営者が参加することであるが、ビルの建設と異なり、目に見えないシステム開発のモニタリングはPMにとっても、経営者にとっても困難である。目に見える成果物がでてくるのは工程がかなり進んでからになり、経営判断を仰ぐ時期を逸する。

BSC経営では、結果ではなく、パフォーマンスをモニタリングするが、モニタリングには、経営者が理解することが可能なEVMによる説明が必要である。

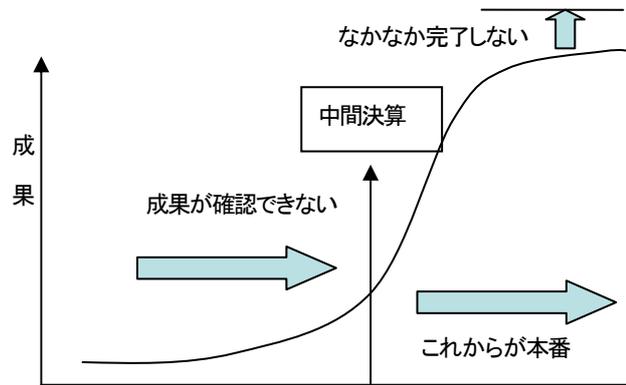


図2 ITプロジェクトの特性

発注側の経営者が請負側にEVMでの説明を要求し、ITベンダー側の経営者はEVMによるITプロジェクトの統制をすることで、リスクは回避できる。

## 2. ITプロジェクト中断事例

### 2. 1 WBSでは予測はできない

図3は約20億円の予定で進めていたプロジェクトを請負ベンダーの経営者が中止の決定した時にPMが使っていたWBSである。

既に8億円のコストを掛けたプロジェクトに更にコストを掛けるのか、或いはプロジェクトを中止するかを決断はPMの役割ではなく経営者の責任である。

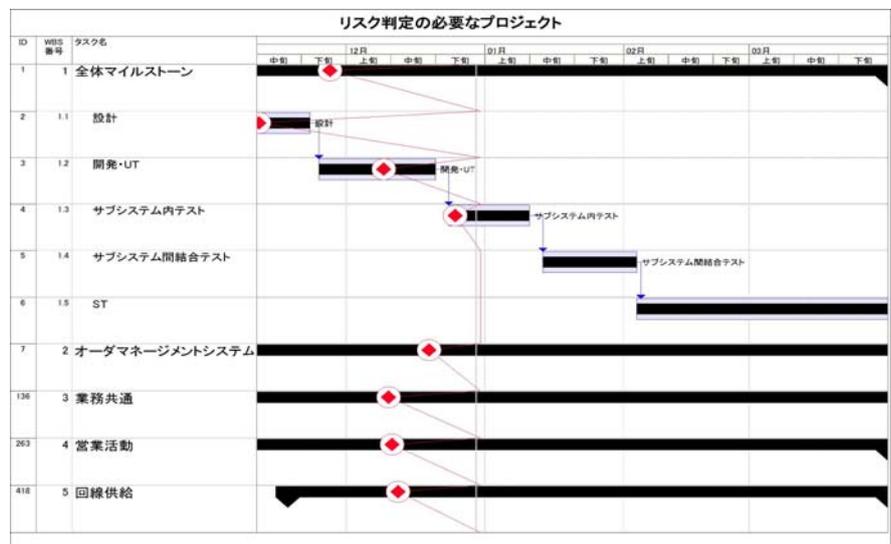


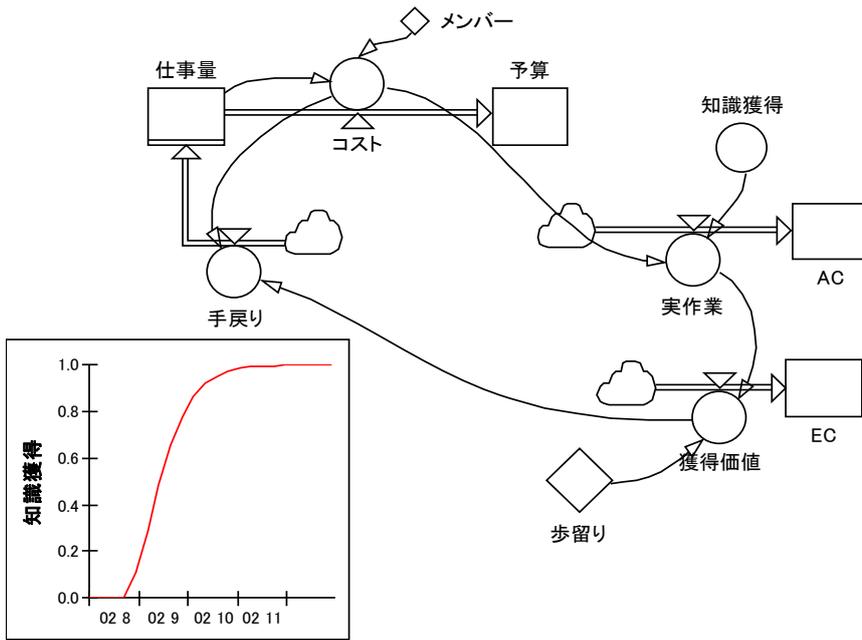
図3 プロジェクトマネージャが使用するWBS

8億円を捨てるには勇気が必要としたが、代替手段として、既存のシステムを改修し、無事本番稼動を迎え、企業としての大きなダメージは避けることができ、経営責任にはならなかった例である。

### 2. 2 計画のwhat-ifシナリオ

仕事量は獲得すべきシステムの価値で示し、金額に換算する。週当たりの仕事量はメンバーの週当たりの単価とするが、成果にかかわらずコスト(費用)が発生する。予算はこのコストの集計額であり、契約金額になる。仕事量がゼロになった時点が仕事の完了時点である。作業をするには、作業環境に関する知識の獲得が必要であ

り、習熟するには一定の期間がかかる。知識獲得コストも含め作業実績の金額が AC となる。実作業の中で完成した量が獲得価値となり、未完了の方は手戻りとして再度作業する(図 4)。



仕事量：作業量の金額換算  
 予算：仕事量が0になるまでのコスト集積  
 コスト：週あたりの人員コスト  
 (80人×35万円/週)  
 AC：実作業のコスト  
 実作業：習熟度を考慮した週当りの作業実績の金額換算  
 知識獲得：能力をフルに発揮するまでの習熟曲線  
 EC：完成した成果物の金額換算  
 獲得価値：実作業の中の完成分の金額換算

図4 EVのSDモデル

このモデルでは、実行計画はメンバーが環境に習熟するために約1ヶ月の知識獲得時間を設定してある。この結果10億円の実行計画に対して、20%のプロジェクトリザーブを設定し、予算は12億円を確保した。作業量が0になる4月13日が完了日である。このプロジェクトリザーブの金額はPMと経営者だけが知っており、メンバーには伝えていない(図5)。

### 2.3 継続した場合の what-if シナリオ

遅れの原因は①要員不足、②能力不足、③歩留まりが悪い、の3種類であるが、③の場合は前工程に原因があり、解決は困難である。発注者側のリーダーやサブリーダーは50億円規模の開発の経験者であり、請負側のPMも修羅場をくり抜けたベテランであるが、必死の努力にも関わらず歩留りは改善しなかった。現時点までの歩留りはACとEVの比から求められる。既に8億円消費し、獲得価値は6億円相当であるから歩留りは75%である。

図6は歩留りが75%の場合の将来をシミュレートした結果である。この結果、EVの価値は変わらないので、10億円のままであるが、費用(予算)は15.7億円かかり、完了日は6月16日になる。これはリザーブを含んだ予算(12億円)を3.7億円超過し、納期は2ヶ月の遅れになる。請負側のWBSを解析して、約800のアクティビティについて各サブリーダーからヒヤリングを約1ヶ月かけて実施し、今後の予測をし、積み上げた予測金額はこれより低い。

プロジェクトを中断したくない請負のPMの常として予測より低い金額を提示する。

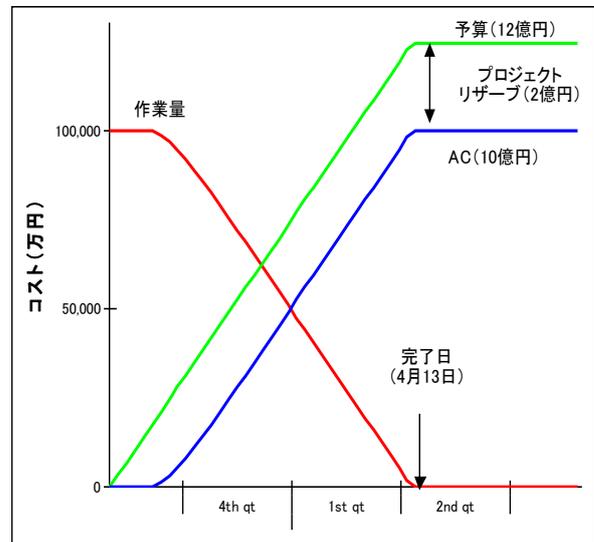


図5 実行計画

## 3. 経営者のリスクコントロール

### 3.1 プロジェクト継続の判断基準

経営者はEVMでプロセスをモニタリングし危機を感知することができる。プロジェクトが危機に直面した時、コストが増えるメンバーの増員の決定はPMの権限ではない。EVMでプロジェクトの状態をモニタリングした後、リスクに対処する決定をするのは経営者の役割であるが、その時の判断基準はPMと統一しておくことが必

要である。

ブルックスの有名な法則「adding people to late project makes it later」[9]はプログラム分割が発生する特別な場合であり、トップダウン設計の手順で開発を進めている場合、プロジェクトが危機に直面した時、経営者がメンバーを追加することで納期を守ることはできる。しかし、追加した時点から30%のコミュニケーションロスが発生し、生産能力は70%にダウンし、コストは更に増大する。

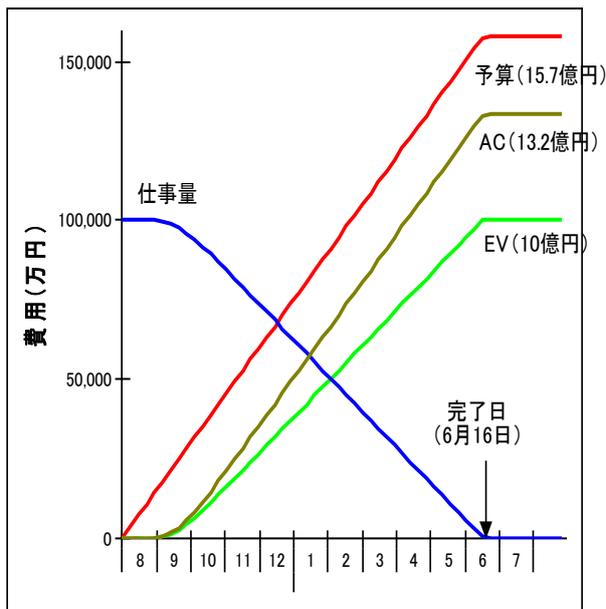
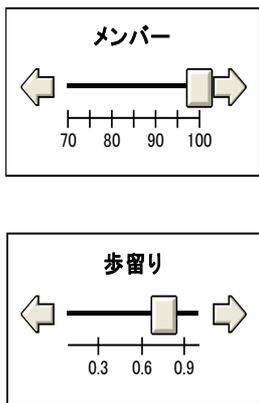


図6 継続した時のWhat-if シナリオ

図7はブルックスの法則を日本のIT開発に適用した時のSDモデルであり、このモデルを使用し、要員を追加した時のコストを見積もることができる(ブルックスの法則の定量的解法[10])。

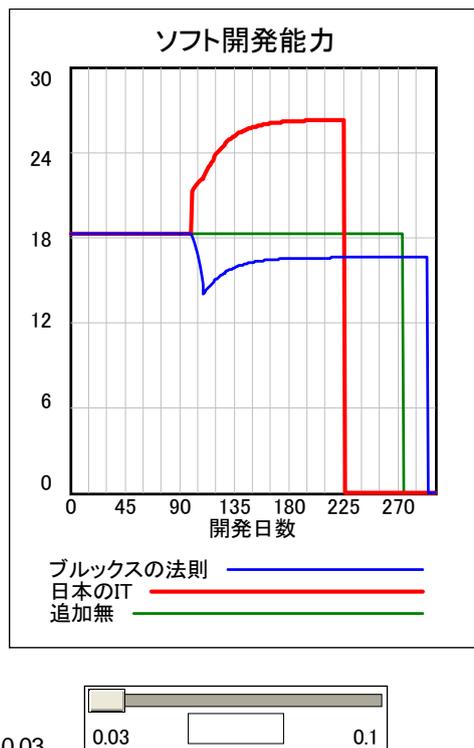
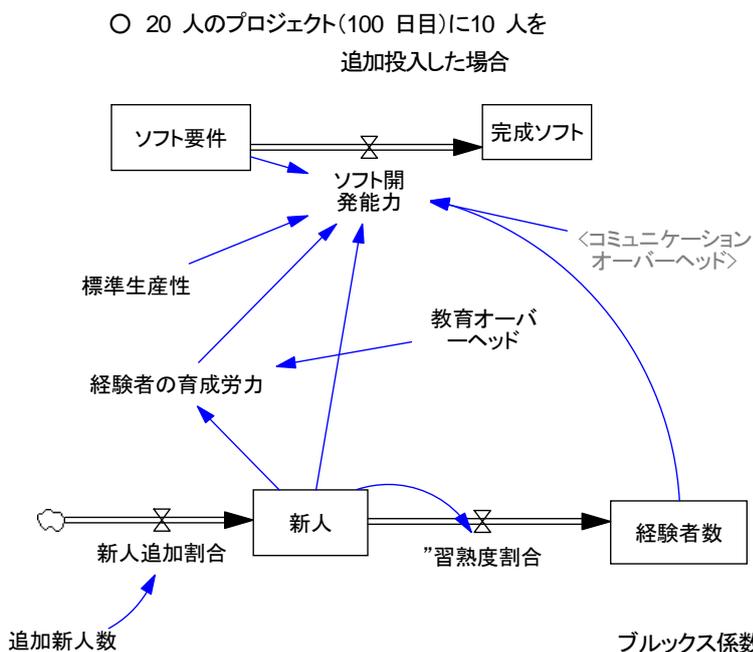


図7 日本のIT開発におけるブルックスの法則

### 3. 2 計画時のリスクの低減

PMが作成したプロジェクト計画の承認は経営者の重要な役割である。仕事の量は規模(サイズ)と複雑性であり、仕事の量が増えれば工数も工期も増える。しかし、計画に対して納期、コストの実績の確率分布は正規分布にならず、ベータ分布になることが知られている。正規分布では形状はベル型と固定であるが、ベータ分布は形状そのものが確率で決まり、unknown(見積り範囲外)の作業に起因する為、規模からだけの単純な計算では導けないことを意味する(図8)。

ブルックスの第2法則では工数はサイズの2/3乗であるが、これは正確ではないことが、COCOMO[11]や国防省の予算で生産性を調査している研究機関から既に発表されている[12]。IPAの「ソフトウェア開発見積ガイド」[13]によれば日本のITベンダーはサイズと実績のデータしか持っておらず、見積の根拠が不明であり、誤差が

大きい。

より正確な見積をするためには生産性に関する因果関係を明確にする必要がある (図9)。

経営者が IT プロジェクト計画を承認する時、PM に対して生産性の根拠となるイネイブラー (1. 開発技法、2. 開発手順、3. 開発ツール) を明らかにさせることがリスクを軽減する最適な方法である。

#### 4. 内部統制のための管理プロセスとツール

##### 4.1 パフォーマンスレビュー

経営者が IT プロジェクトを統制する為にはリスクの少ない計画の作成と進捗をモニタリングできることが条件となる。

経営者が判断できるのは利益と納期であり、PM は経営者に対しては EVM で説明することが必要であるが、EVM での説明は結果だけではなく、なぜその結果になるかの説明が重要である。この因果関係を説明するには図 10 のように因果関係を SD でモデル化したジェネリックモデルが有効である (図 10)。

このジェネリックモデルを要件定義、設計、開発、テスト等のプロセスごとに作成し、結合することでプロジェクト全体を表現することができる (次世代型の PERT[3])。このツールを

使用し、各要素を変化させ、ITSS や PMBOK が要求している経営者が参画したプロジェクトエバリュエーション (パフォーマンスレビュー) のプロセスが内部統制には必須である。

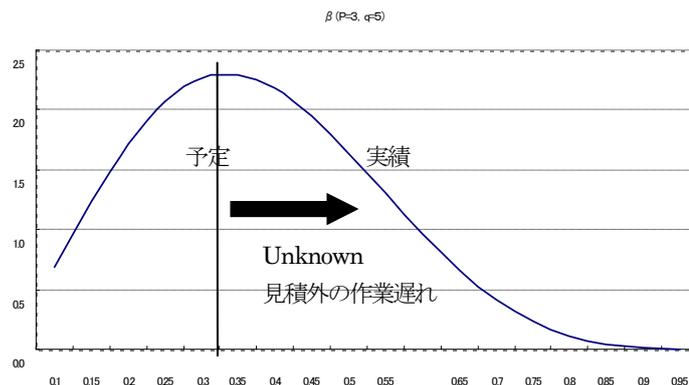


図8 見積りできないベータ分布

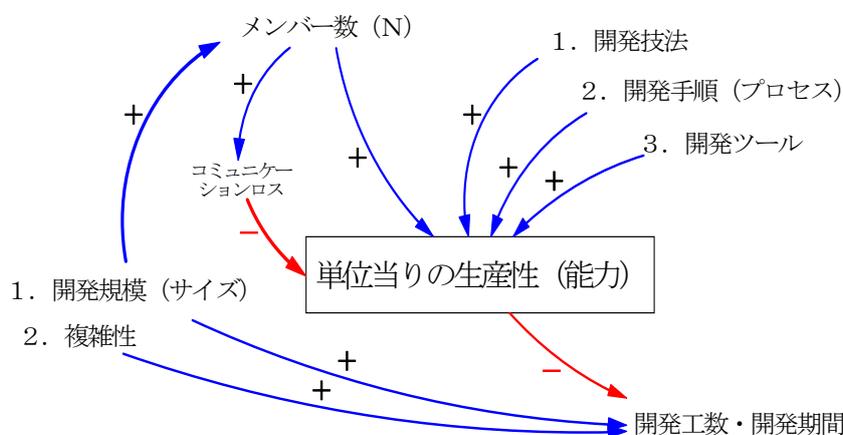
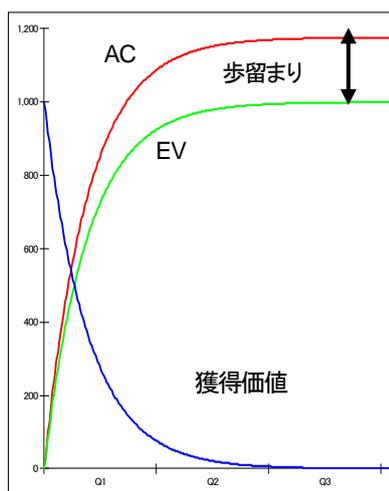
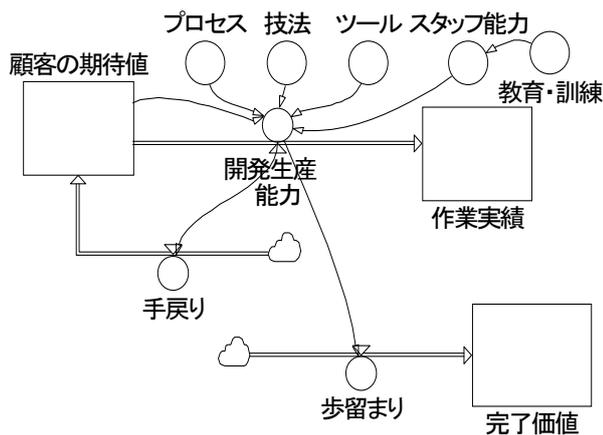


図9 生産性の要素とイネイブラー



生産能力のイネイブラー  
 ① プロセス  
 ② 道具 (ツール)  
 ③ 技法 (メソッド)  
 \* 「手戻り」は前工程の品質に依存する

図10 EVMのジェネリックモデル

##### 4.2 標準生産性指標の蓄積と分析プロセス

経営者が参加したプロジェクトパフォーマンスレビューを実施する為には、過去の実績にもとづいた計画のベースの情報が必要であり、蓄積する情報は過去のプロジェクトの実績であるが、プロジェクト解散後はこれらの情報は飛散し、組織に残ることがない。CMM レベル3 では情報の組織的蓄積プロセスを要求しているが、蓄積

には工数がかかり、生産性は向上しない（レベル2からレベル3で約3%の向上率）。CMM 導入に失敗した企業の多くは、必要な情報が何かを定義せず、収集蓄積にコストを掛けすぎた例が多い。

表1は汎用機、COBOLの時代からUNIX、C言語、更にWEB技術へとテクノロジーの変化に対応してきた時のベテランPMの経験知である。作業実績の工数だけではなく、ベテランがもつこのようなイネイブラーに関する暗黙知こそが組織が引き継ぐべき真の情報である。

空白部分はデルファイ法、EASE 協調フィルタリング法[14]などで、欠損値を埋める。イネイブラーの実績情報収集には大規模なシステムを必要とせず、コストもかからない。

イネイブラーとともに「歩留り」に関する情報が計画時に必要になる。歩留りは前工程の品質に依存し、前工程のドキュメントの量と質から依存関係の情報を分析でき、すでにこのような情報を蓄積しているITベンダーがある。この蓄積と分析プロセスが組織的に機能したとき、見積もり精度は次第に高まり、プロジェクト計画の概算見積りは規模と実績から見積もった場合よりリスクが少なくなる。CMMの狙いとする組織能力を高めるには図11の因果関係を理解し、内部プロセスを組織に実装することが必要である。

表1 ベテランの暗黙知

1) C言語の修得の修得時間	
アセンブラ経験者	50 時間
COBOL 経験者	150 時間
2) VBの修得時間	
C 言語経験者	20 時間
COBOL 経験者	50 時間
UNIX 開発環境構築知識修得時間	
C 言語経験者	100~150 時間
3) WEB技術の修得時間	
C 言語経験者	30~50 時間

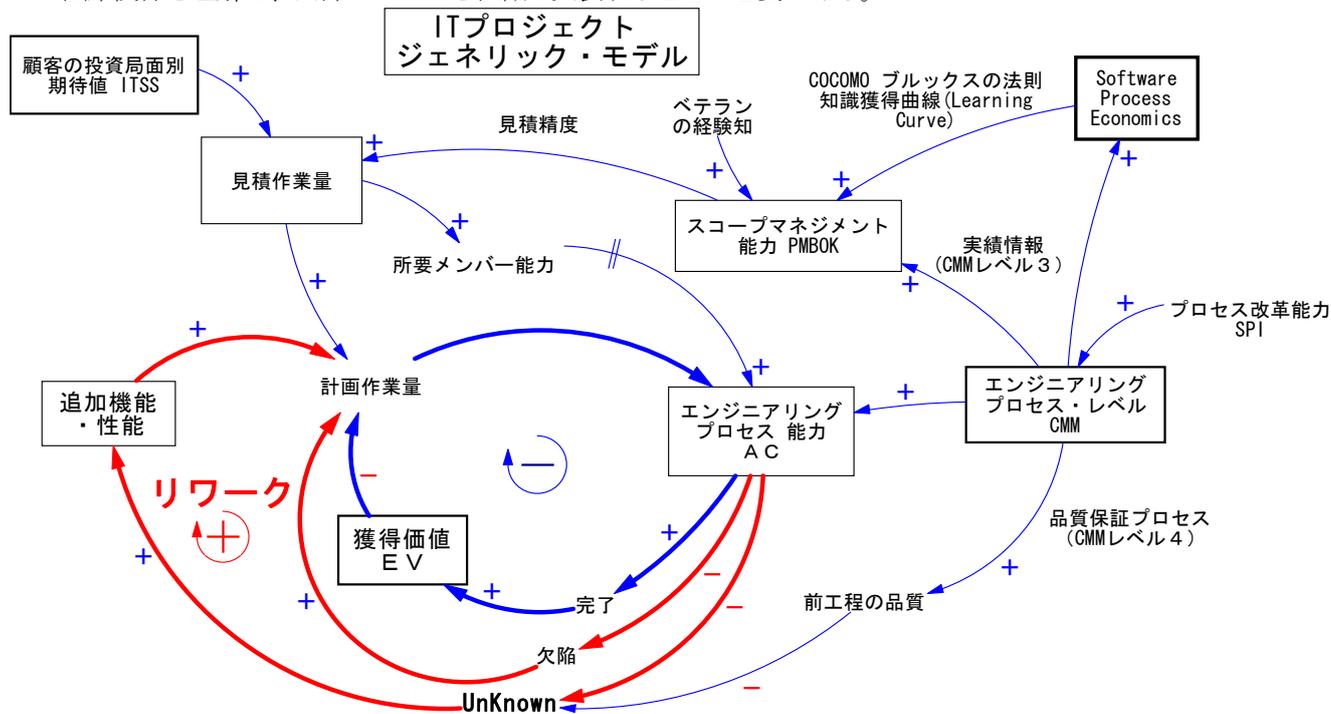


図11 組織能力を高める為のモデル

## 5. ITプロジェクト開発の生産性の改善

### 5.1 連続生産と非連続生産

管理プロセスを充実することは重要ではあるが、生産性を高めることはできない。CMM レベル4を達成しても、生産性の低い製造ラインでは生産性は向上しない。生産性を高めるにはイネイブラー（プロセス改善、技法の導入、ツールの導入）などへの設備投資が必要であり、生産性の高い製造ラインやツールへの設備投資の決定はPMではなく、経営者の重要な役割である。ITプロジェクトは自動車や電機製品などの連続生産(Production)と異なり、毎回開発条件が異なるのが非連続型の生産である。1980年代ソフトウェア工場を目指し、AD サイクル[15]など、Production 生産（いわゆる「ウォーターフォール」式のプロセス）を試みたが、各社ともに失敗に終わっている。Production 生産の成立する条件はモジュール化し、その後工程での組み合わせである。このためにはモジュール化能力のある技術者を必要とするが、十分訓練された米国の技術者でも正しくモジュール化された

例は 70%であり、訓練の不十分な日本の SE では正しくモジュール化される例は 20%以下である。正しくモジュール化されていない場合、後工程では想定外の作業が発生し、単純な組み合わせ作業では済まなくなり、その結果、生産性は大幅に悪くなる。

近年 Production 生産と Project 生産を統合したゴールドラッドの制約理論にもとづく Critical Chain Scheduling & Buffer Management(CCS&BM)が IT の開発プロセスとして着目されている。CCS&BM は完全なモジュール化を前提としない「すり合わせ」方式であり、トヨタの生産方式が原点であり、この生産方式は連続型の生産と非連続型の生産を統合した管理が特長である。

IT 開発での「すり合わせ工程」はアジャイル開発方式(助け合い方式・・・熟練者の技術を移転する)ともいわれ、米国で導入している IT ベンダーは多い

CCS&BM を成功させるには安全率を取り除いた確率 50%のスケジューリングを作成する。確率 50%の日程は努力すれば達成可能な日程であり、95%の安全率を見込んだ時より、計画上の日程を約 30%短縮することができる(図 12)。

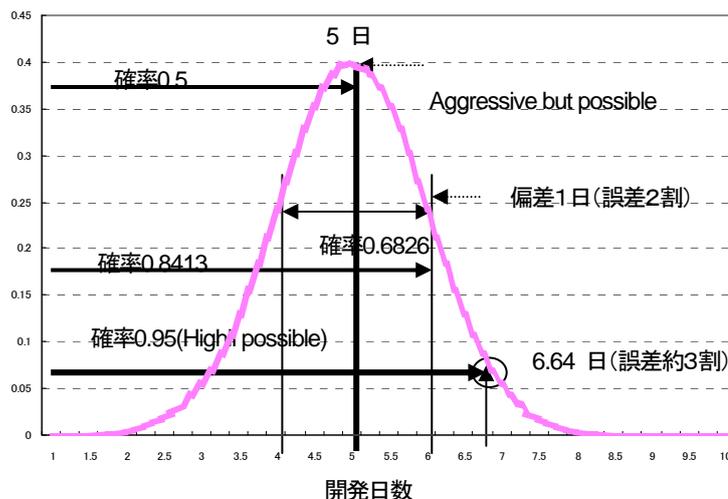


図 12 納期 5 日の確率 50%スケジューリング

### 5. 2 生産性を阻害するパーキンソンの法則

非連続型の生産の場合、毎回開発環境が異なり、見積り作成が困難であるが、請負型のプロジェクトではパーキンソンの法則(Work expands to fill the time allowed)が作用し、さらに生産性を阻害する。

プロジェクトの失敗を経験するとメンバーも下請も工程ごとにリスクを考慮し、安全率を過大(確率 95%以上)に積み上げる。この時約 30%の納期の誤差が発生するが、複雑な下請け構造の IT 業界では、2 次下請け、3 次下請けと、それぞれ 30%のサバが積算され、更にサバが増幅される。その結果、全体では大きな「サバ」ができる。このサバの大きな計画にパーキンソンの法則が働き、例え予定より早く作業が終了しても納期は早まることはない。これらのサバ

を含んだ実績が次の見積りに使われる為、さらに生産性は悪くなり、また実際の工数を把握することができない。

これを避けるにはクリティカルなプロセスは確率 50%でスケジューリングして、工期

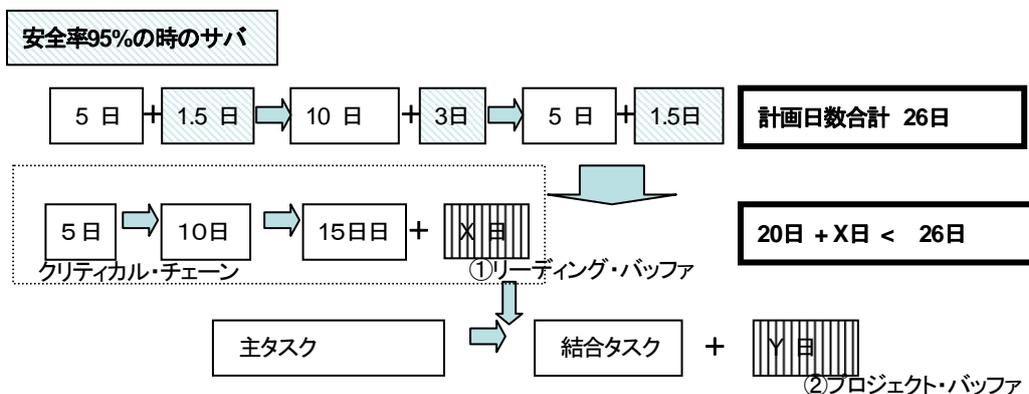


図 13 クリティカルチェーンスケジューリングとバッファ

を短縮し、リーディング・バッファとプロジェクト・バッファを設定し、遅れを取り戻す CCS&BM の導入が必要である。この時、バッファプロセスは組み合わせやテストではなく「すり合わせ」の工程になる(図 13)。バッファを大きくとり、すり合わせる方式は、過去日本のベテランの PM のやり方と同じである。

### 5. 3 CCS&BM のリスクマネジメント

メンバーや下請は 50%の確率で計画することを嫌がるが、計画に含まれる「サバ」を排除するのは PM の役割であり、説得するには確率 50%でスケジューリングした場合、リスクがどのくらいになるかを説明する必要がある。

SD ツールにはリスクアセスメントの機能が標準で用意されており、図 14 は 5 人のプログラマーが 1 本のプログラムを 5 日間の予定で作るとき、50%の確率ではどうなるかをミュレーションした例である。モンテカルロ法を使った乱数を発生させた結果、早い人で 3 日間、遅い人で 8 日間になる。しかし、1 人で 5 本のプログラムを 25

日の予定で作業した場合、同様の条件でシミュレーションした結果、最速で 17 日間、最遅で 25 日間であり、計画内に収まる。[10]

早く終わった人が遅れている人を「助ける」アジャイル開発では、遅れを取り戻し、さらに熟練者の技術を新人に移転することができる。

#### 5. 4 バッファマネジメントのテクニック

IT プロジェクトを成功させるには内部統制のプロセスの充実だけではなく、オペレーショナルプロセスのマネジメントテクニックが必要である。

特に CCS&BM の成功はリーダーの経験と能力に依存する。この能力はバッファマネジメントにおけるベースラインマネジメント（仮決めテクニック、構成管理）やエンゲージメントマネジメント（メンバーや下請けとのにぎり、仕切り）で必要とされるが、これらのテクニックは実践やワークショップでの訓練を経て身に付く。

CMM 導入失敗のように管理プロセスの導入だけで成果は得られないことを教訓に、テクニックを磨き、失敗を繰り返さないことが必要である。

#### 6. 結言

30 数年前に、経営者と一体になって積極的に IT 導入をし、世界に先駆け、最先端のシステムを構築し、高度成長を成し遂げたのが日本の企業である。しかし、その後ベテランの PM（システム部長）に全てを任せ、経営者（CIO）が IT 導入に参画しなかったことが、IT を効果的に使った欧米に遅れをとり、現在の日本の不況を招いた一因でもある。これまで、業務知識と製品知識獲得に偏重し、生産技術やプロジェクト経営技術を追求してこなかったことは IT ベンダーの経営者の責任であり、経営者が理解できる言葉で説明を行ってこなかった日本の IT 技術者の怠慢でもある。

企業経営において IT の果たす役割はますます大きくなり、IT 導入失敗により経営者は顧客や株主に対しての結果責任をとらされる。JSOX 法を契機に、経営の視点で統制のために開発された欧米のベストプラクティスの導入の好機である。計画や意思決定のプロセスにおいて、マネジメントツールとしての SD の活用は日本では未開拓の分野であるが、期待は大きい。

#### 参考文献

- [1]EVM : EVM 活用型プロジェクトマネジメント、IPA、2003
- [2]CMM : Carnegie Mellon University Software Engineering Institute (SEI)
- [3]CMMI レベル4以下のEVM : 蓮尾克彦、JSD 研究会、2006
- [4]CCS&BM : Francis S.Patrick、Focused Performance、1998
- [5]パーキンソンの法則 : 英国政治学者、1960 年頃
- [6]米国の IT ポートフォリオ : Harvard Business School Press、2001
- [7]COBIT : Information Systems Audit and Control Association、1996
- [8]Naval Design Process Model : Naval Engineers Journal、1996
- [9]The Mythical Man-Month : F.Brooks、Datamation、1974
- [10]ブルックスの法則の定量的解法 : 蓮尾克彦、BPD 分科会、2006
- [11]Constructive Cost Model : Barry Boehm、1981
- [12]Overall Group Productivity : USC Center for Software Engineering、1998
- [13]ソフトウェア開発見積りガイド、IPA、2006
- [14]EASE 協調フィルタリング法 : 奈良先端大学が開発した手法
- [15]Application Development Cycle : IBM 社、1989

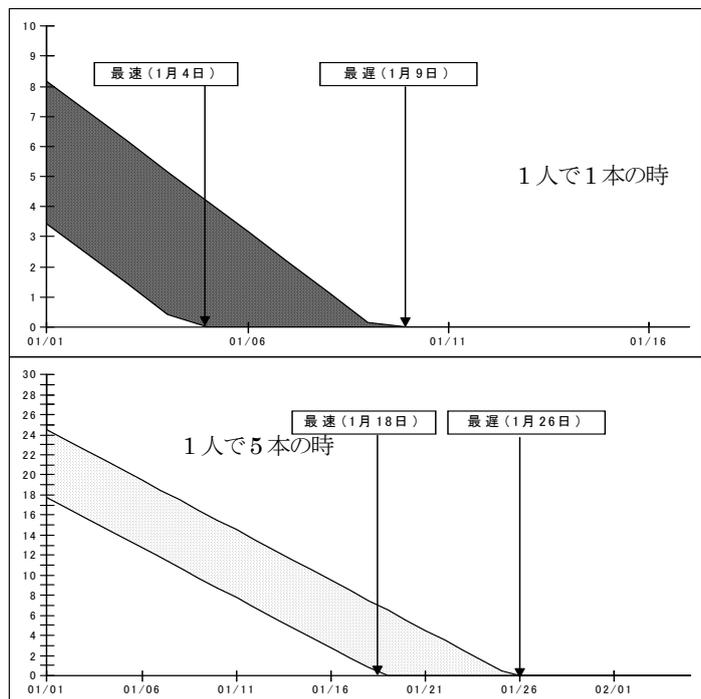


図14 確率50%でのリスクシミュレーション