2006年 ＪＳＤ研究発表会
テーマ ： マネジメント･プロセスの革新に向けて

CMMIレベル4以下のEVM
Applied EVM for under CMMI Level 4 IT Project

2006年1月14日

所属 ITコーディネータ協会
氏名 蓮尾 克彦

技術暦

1970年(S45)　早稲田大学応用物理学科卒
　　日本ユニバック株式会社（現日本ユニシス）入社
　　生産管理、住民情報などAP開発
　　通信、ディスクドライバー開発
　　システム設計方法論の開発
・1983年(S58)　UNIXサポート（カーネル改造）会社創設
・1988年(S63)　月刊パッケージソフトに解説記事　2年間連載
・1990年(H2)　米国ソフトウェアマガジン日本版発刊
　　ダウンサイジング（クラサバ）の日本への紹介
・1994年(H6)　東電ソフト（現テプコシステムズ）入社、現在に至る
　　人材開発センター（ITSS研修コース設計）
・2000年(H12)　ITコーディネータ・インストラクタ
・2004年(H16)　ITコーディネータ協会出向
　　ITC育成カリキュラム作成
　　某大学　IT業界就職対策講座　講師

■大小（1千万円～120億円）60システムの開発に参画
■国際システムダイナミクス学会所属
　　論文：SDによる戦術的BSCの解法

日本で普及しないEVM

何時、誰が、どう使うのか?

PV(Planned Value) **出来高計画値　計画時に、各作業に割りあてられた価値。**

EV(Earned Value) **出来高実績値　現時点までに完了した作業に対して、元々割り当てられていた出来高。**

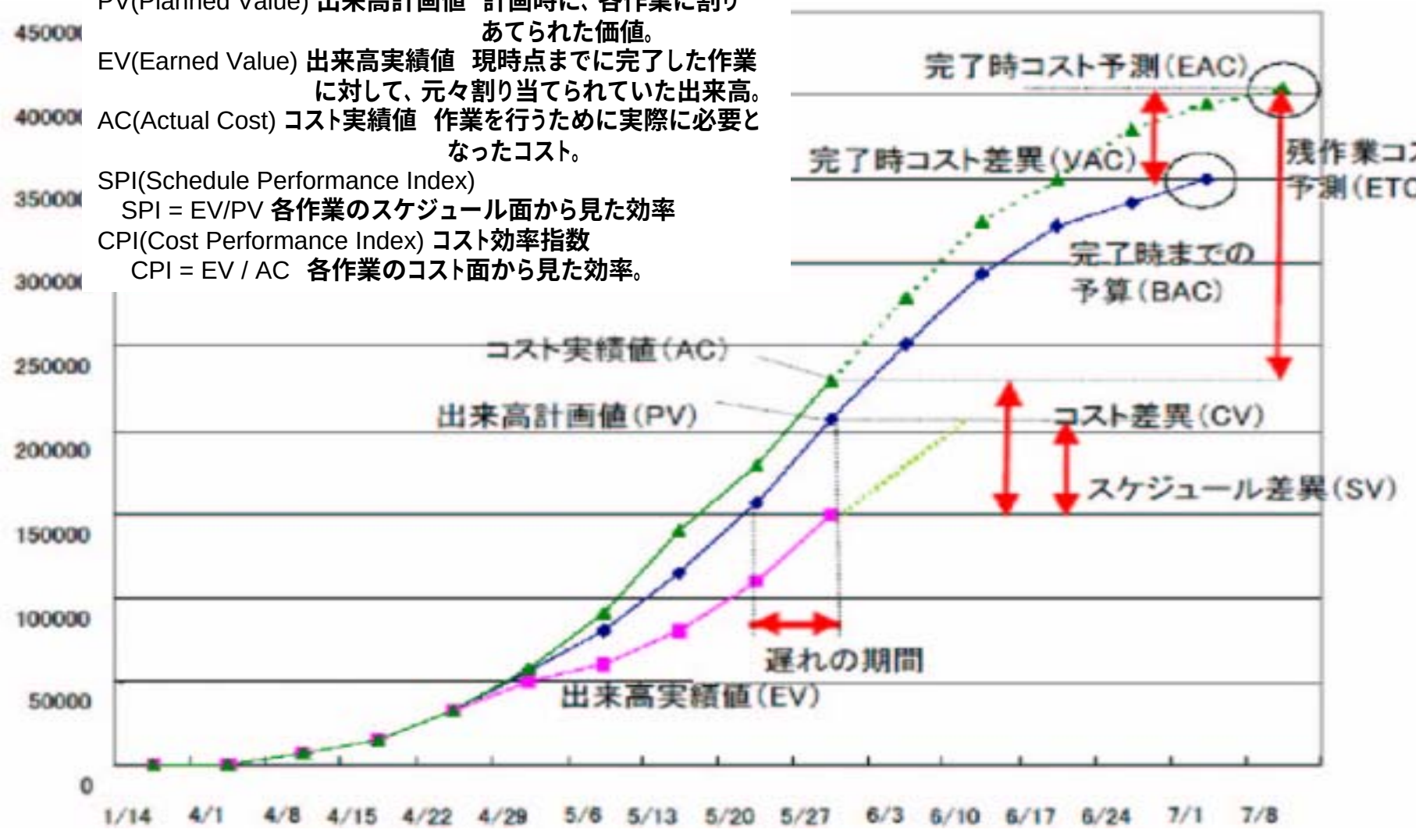
AC(Actual Cost) **コスト実績値　作業を行うために実際に必要となったコスト。**

SPI(Schedule Performance Index)

SPI = EV/PV **各作業のスケジュール面から見た効率**

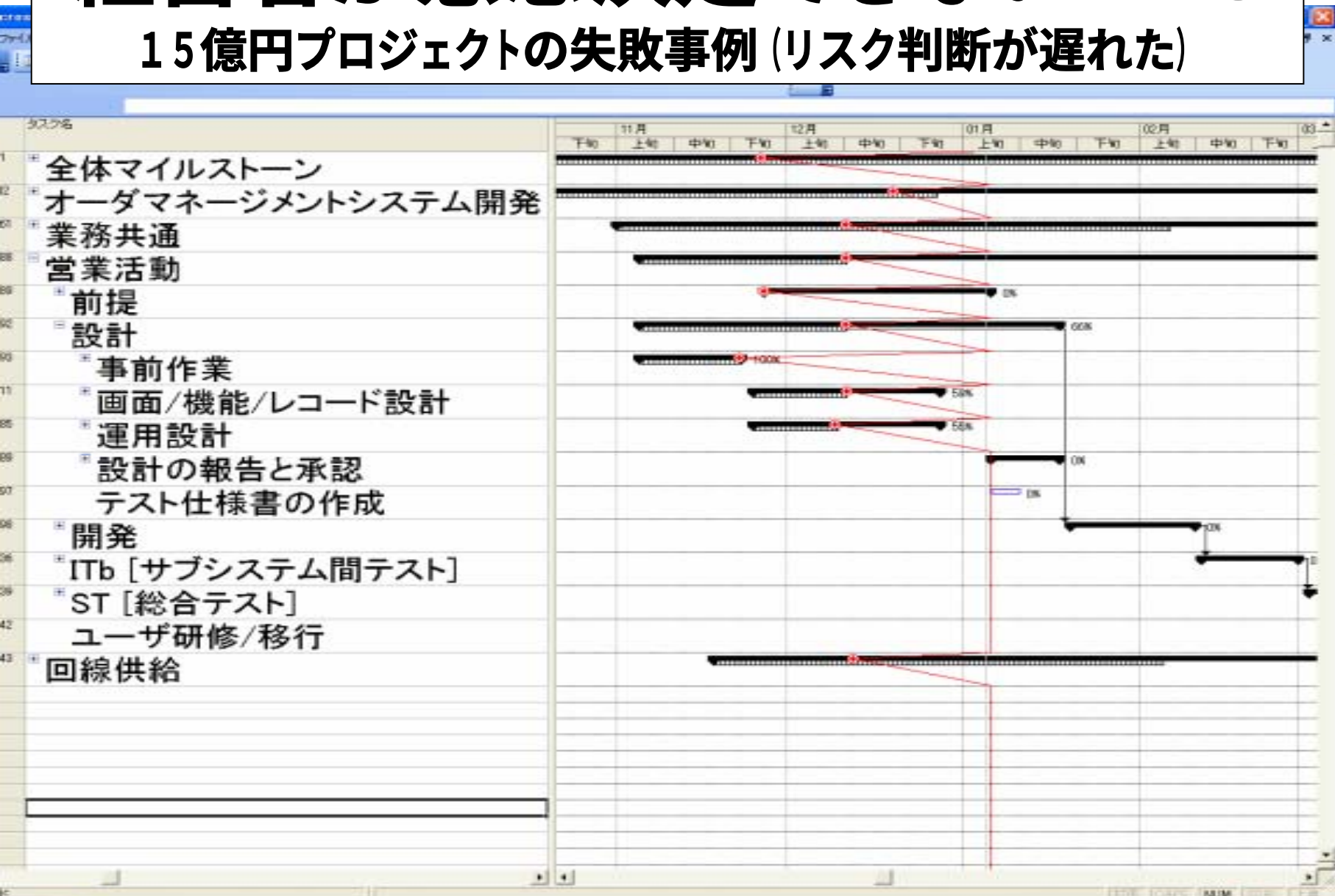
CPI(Cost Performance Index) **コスト効率指数**

CPI = EV / AC　**各作業のコスト面から見た効率。**

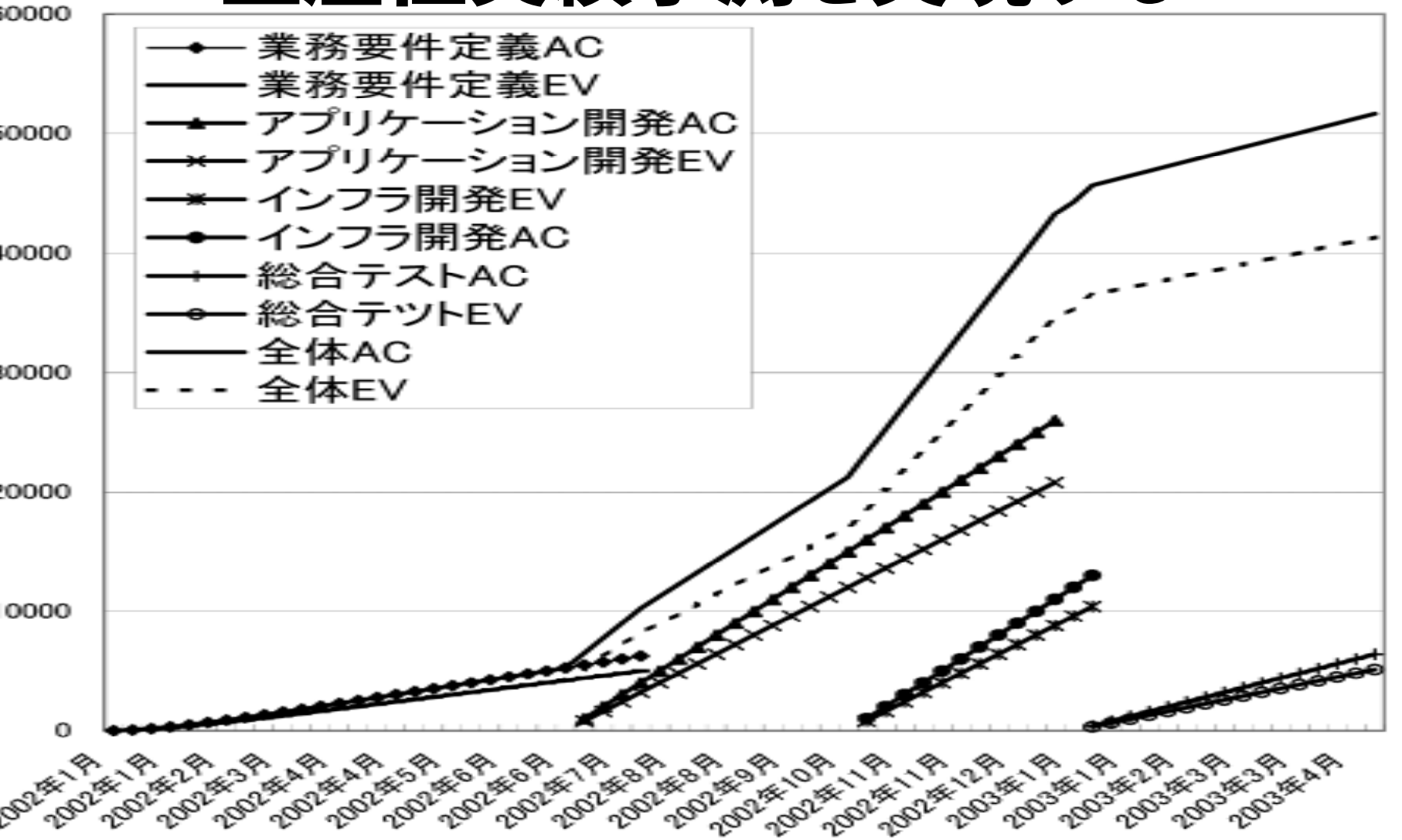


経営者が意思決定できないWBS

１５億円プロジェクトの失敗事例（リスク判断が遅れた）



アクティビティごとの
生産性実績予測を実現する

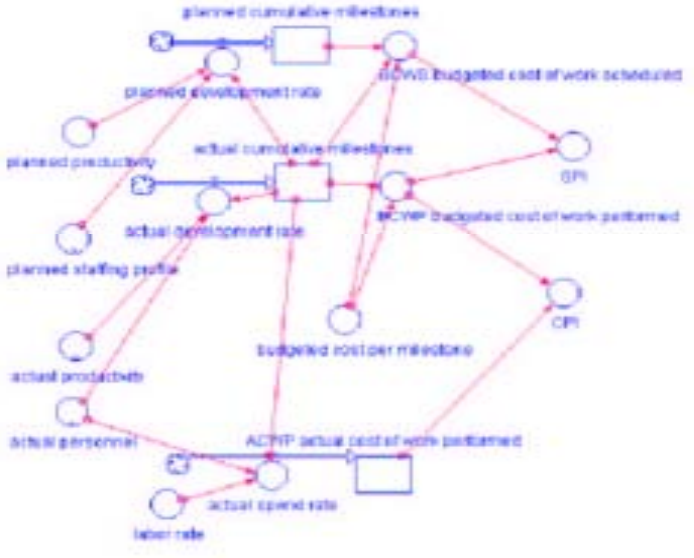


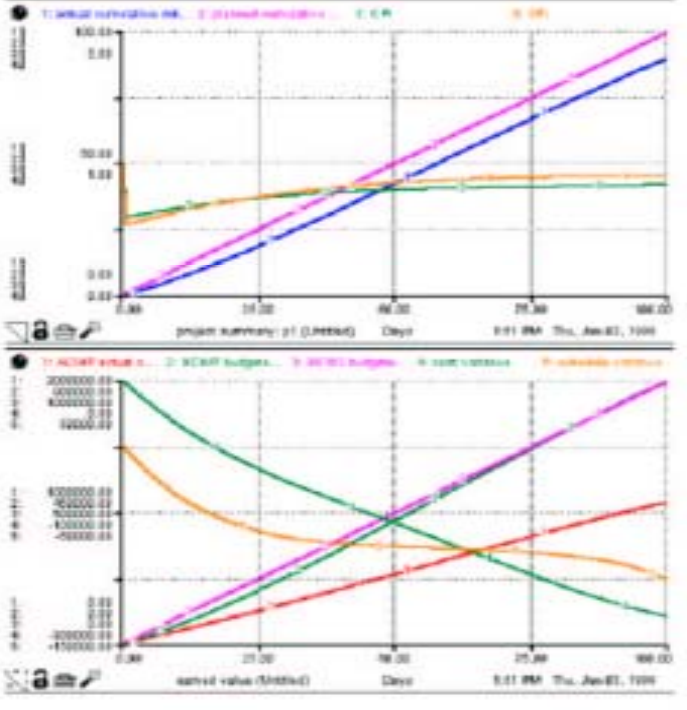
USCのEVMモデル

欧米で普及しているEVM (1998年頃から)

University of Southern California
Center for Software Engineering

Earned Value Model





フィードバックのないシリアルなプロセス

「成長の限界」1970年8月

- ローマ・クラブから作業を委嘱されたMITのメドウズ助教授（当時）等の研究グループの研究成果を取りまとめたものである。

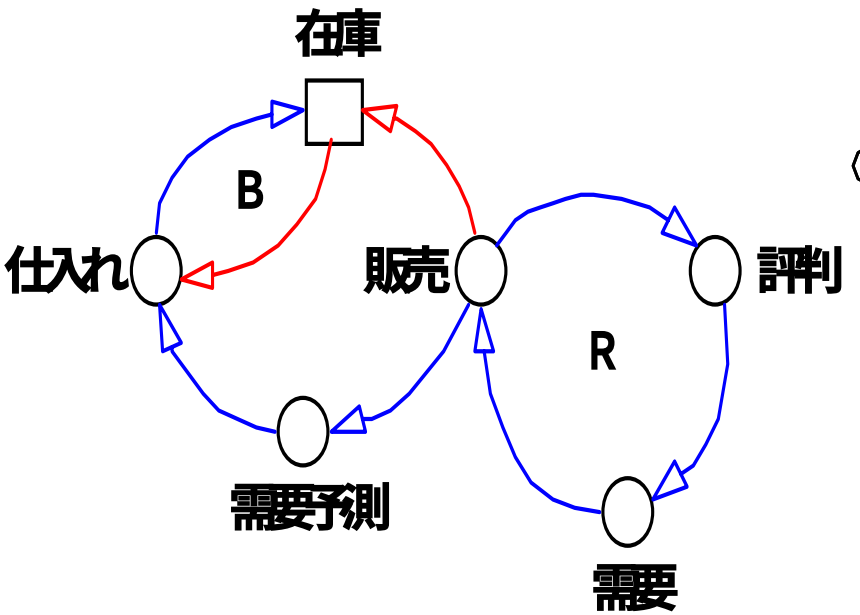
- MITのメドウズ助教授のグループは、フォレスター教授が設計したシステム・ダイナミックスに基づくワールド・モデルを原型としてワールド3を開発して用い、加速度的な工業化、天然資源の枯渇、環境の悪化、急速な人口増加、広範に広がる栄養不足の相互に関連した傾向を100年先の将来にまで分析して、人口と産業の成長予測を示した。

システム･ダイナミックスとは？

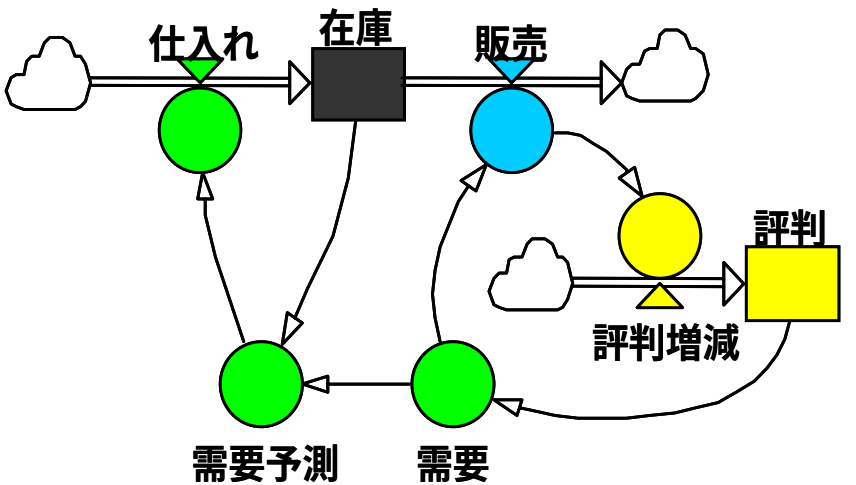
- システム･ダイナミクスは、複雑なシステムを分析して理解し、何らかの方法でそれを修正、変更するための手法です。工学分野の制御理論と同根の技術で、あえて短絡的な表現をするなら、SDは制御理論を社会系の対象あるいは問題に適用した技術と言えます。
- SDは次の二つのコンセプトからなっています。
  - フィードバック理論　：システム構造を組織化するための一般的なガイドライン
  - コンピュータ･シミュレーション　：システム構造から生じる挙動を推定するための方法

システム･シンキング、ロジカル･シンキング
コーザル･ループ、インフルエンス･ダイグラム

因果関係図
ビジネス要素と要素間の関係



ＳＤモデル
仕入･販売モデル



http://www.posy.co.jp

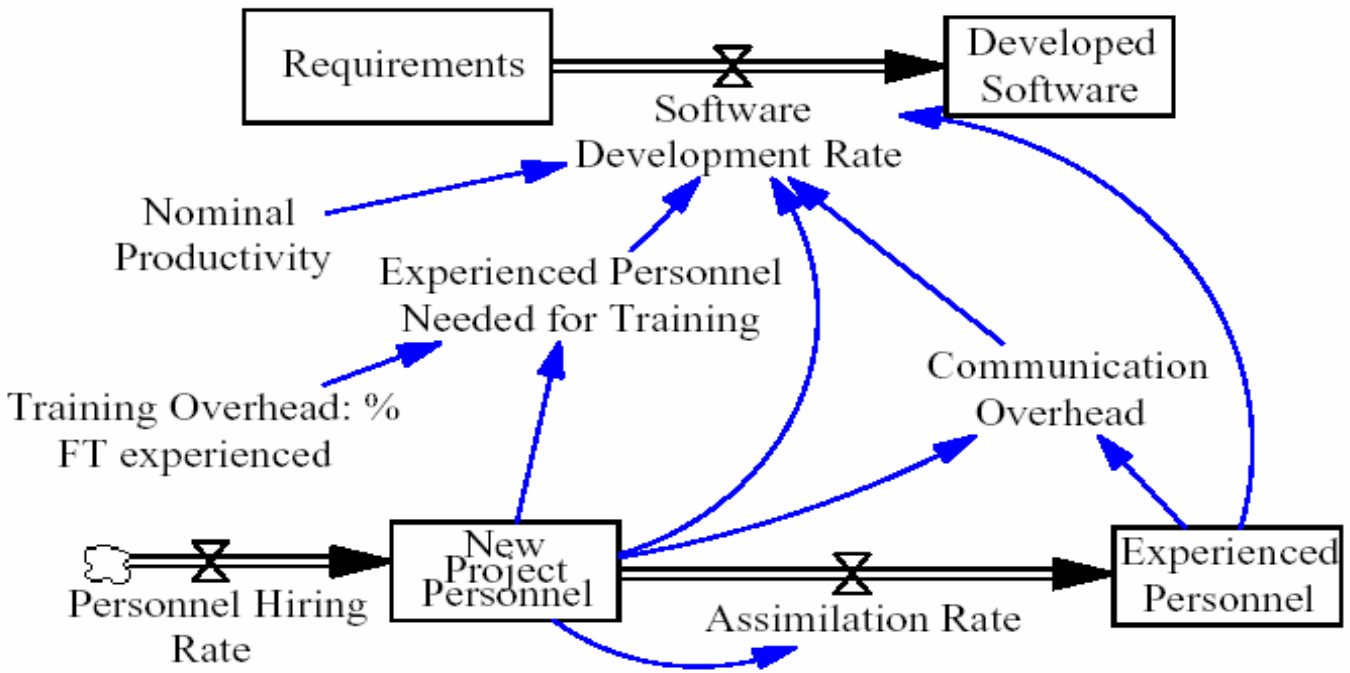
戦略シミュレーション
最適在庫管理
仕入価格リスク評価など

"*Adding people to a late project makes it later*"

Fred Brooks in *The Mythical Month-Month*, 1975

Brook's Law - Full Model

- Communication losses as team size increases



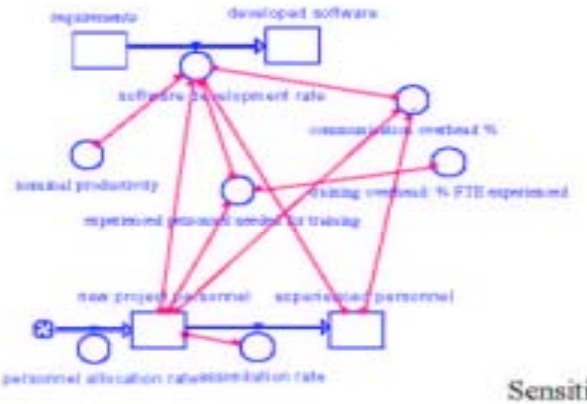
Software Development Rate = Nominal Productivity***(1-Communication Overhead/100)***
(0.8*New Project Personnel+1.2*(Experienced Personnel-Experienced Personnel Needed for Training)

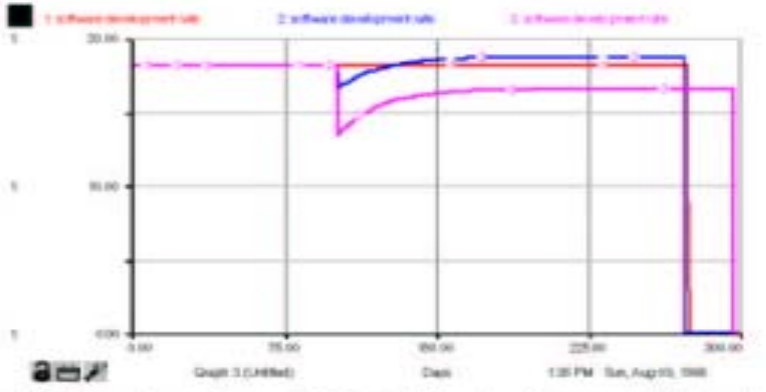
Communication Overhead = 0.06*(Experienced Personnel+New Project Personnel)^2

ブルックスの法則

University of Southern California
Center for Software Engineering

Brooks's Law Model



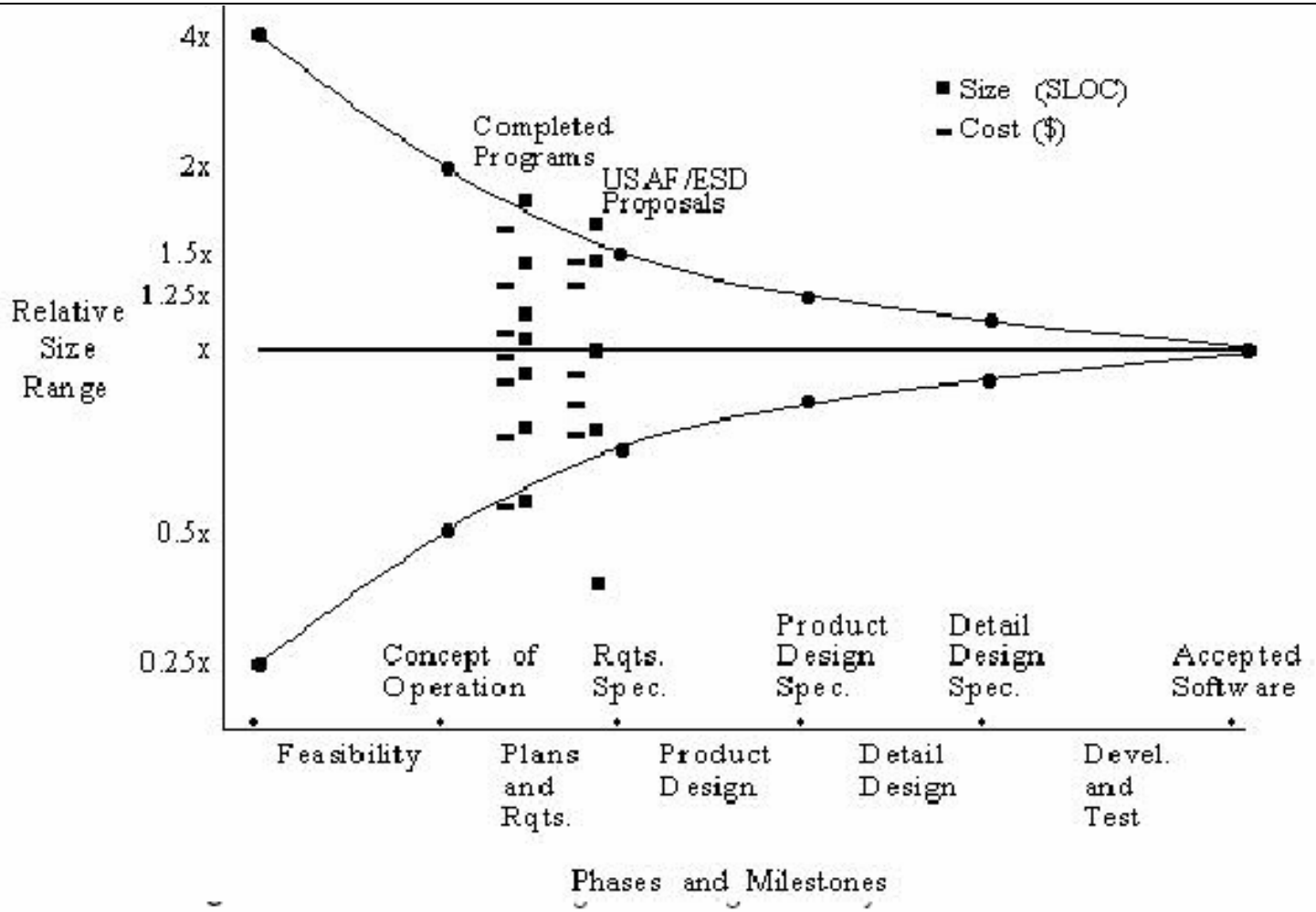


Sensitivity of Software Development Rate to Varying Personnel Allocation Pulses
(1: no extra hiring, 2: add 5 people on 100th day, 3: add 10 people on 100th day)

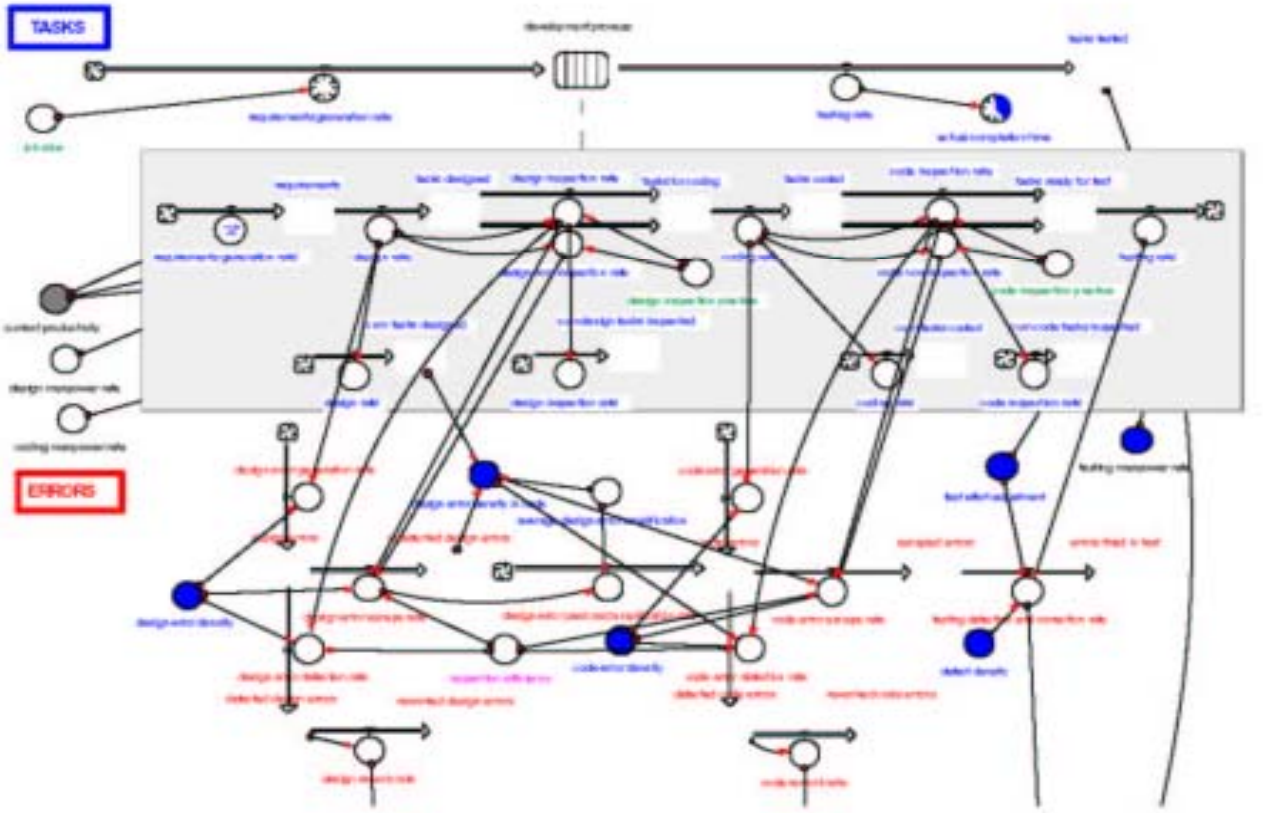
Basic model assumptions:

- new personnel require training by experienced personnel to come up to speed
- more people on a project entail more communication overhead
- experienced personnel are more productive then new personnel, on average

COCOMO II
規模と生産性

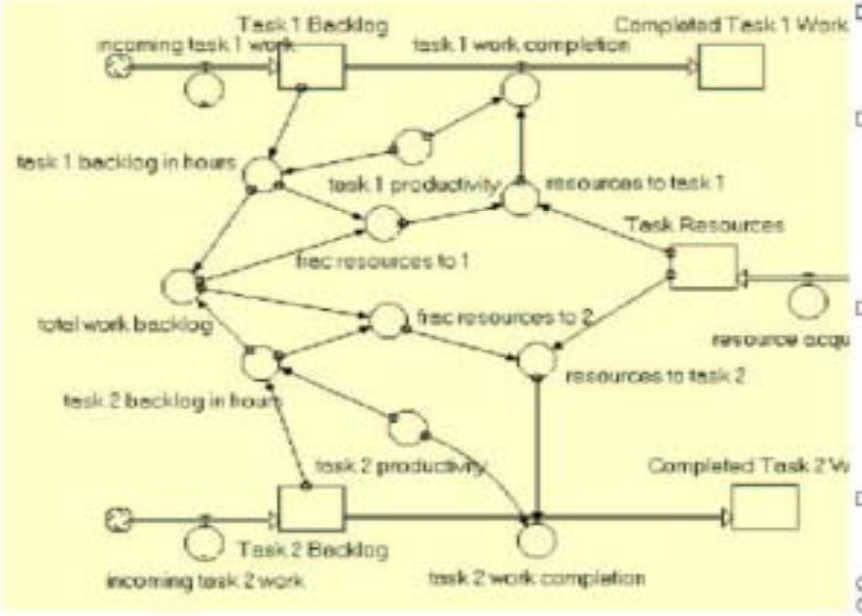


Peer Review Model Diagram



再利用のモデル
再利用技術 (クラスライブラリ)

Resource Allocation Model

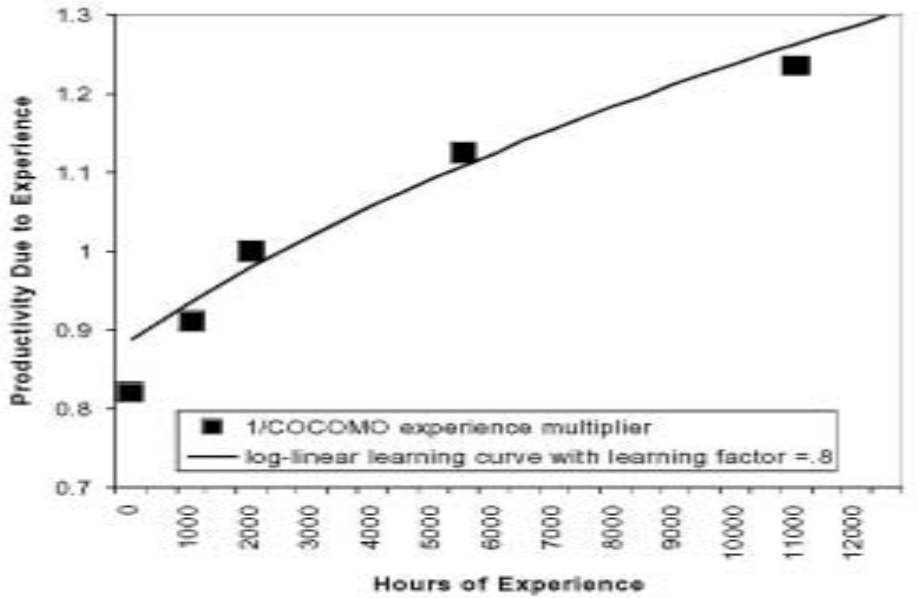


Completed_Task_1_Work(t) = Completed_Task_1_Work(t - dt) + (task_1_work_completion) * dt
INIT Completed_Task_1_Work = 0
INFLOWS:
task_1_work_completion = resources_to_task_1*task_1_productivity
Completed_Task_2_Work(t) = Completed_Task_2_Work(t - dt) + (task_2_work_completion) * dt
INIT Completed_Task_2_Work = 0
INFLOWS:
task_2_work_completion = resources_to_task_2*task_2_productivity
Task_1_Backlog(t) = Task_1_Backlog(t - dt) + (incoming_task_1_work - task_1_work_completion) * dt
INIT Task_1_Backlog = 10
INFLOWS:
incoming_task_1_work = GRAPH(time)
(1.00, 1.00), (2.00, 1.00), (3.00, 1.00), (4.00, 1.00), (5.00, 1.00), (6.00, 1.00), (7.00, 1.00), (8.00, 1.00), (9.00, 1.00), (10.0, 1.00), (11.0, 1.00), (12.0, 1.00), (13.0, 1.00)
OUTFLOWS:
task_1_work_completion = resources_to_task_1*task_1_productivity
Task_2_Backlog(t) = Task_2_Backlog(t - dt) + (incoming_task_2_work - task_2_work_completion) * dt
INIT Task_2_Backlog = 10
INFLOWS:
incoming_task_2_work = GRAPH(time)
(1.00, 1.00), (2.00, 1.00), (3.00, 1.00), (4.00, 1.00), (5.00, 1.00), (6.00, 1.00), (7.00, 1.00), (8.00, 1.00), (9.00, 1.00), (10.0, 1.00), (11.0, 1.00), (12.0, 1.00), (13.0, 1.00)
OUTFLOWS:
task_2_work_completion = resources_to_task_2*task_2_productivity
Task_Resources(t) = Task_Resources(t - dt) + (resource_acquisition) * dt
INIT Task_Resources = 4
INFLOWS:
resource_acquisition = 0
frac_resources_to_1 = task_1_backlog_in_hours/total_work_backlog
frac_resources_to_2 = task_2_backlog_in_hours/total_work_backlog
resources_to_task_1 = Task_Resources*frac_resources_to_1
resources_to_task_2 = Task_Resources*frac_resources_to_2
task_1_backlog_in_hours = Task_1_Backlog/task_1_productivity
task_1_productivity = 5
task_2_backlog_in_hours = Task_2_Backlog/task_2_productivity
task_2_productivity = 5
total_work_backlog = task_1_backlog_in_hours+task_2_backlog_in_hours

COCCOMO (プログラミングの生産性)
ケーパージョーンズ、IPA

University of Southern California
Center for Software Engineering

Learning Curve Comparison to COCOMO Multipliers



cs599 12/7/99

11

CMMIレベル4の組織能力

管理すべき知識項目として24のProcess Area

- 1)**組織プロセス実績**

  Organizational Process Performance

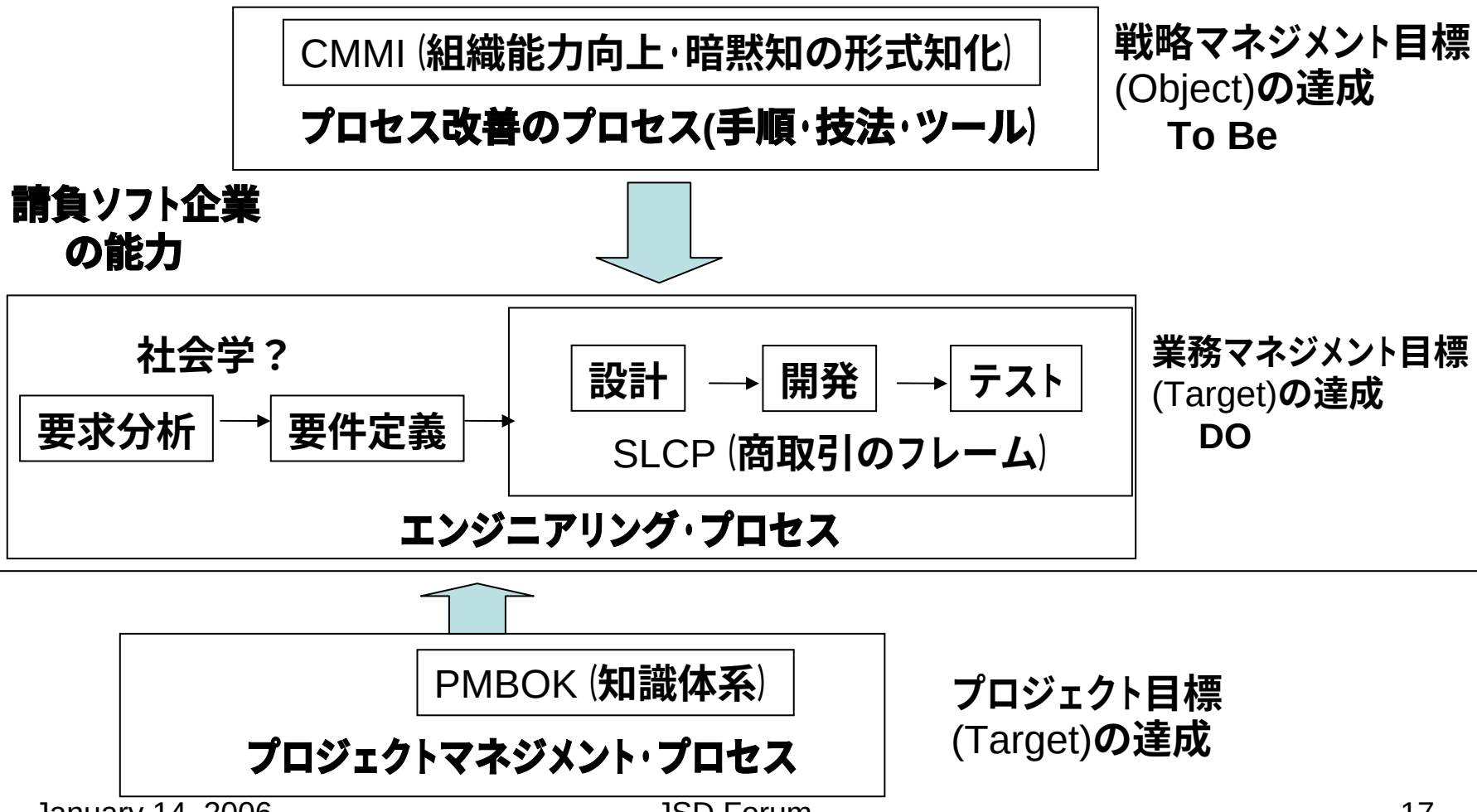
  過去の実績が組織的に蓄積できる。
- 2)**定量的プロジェクト管理**

  Quantitative Project Management

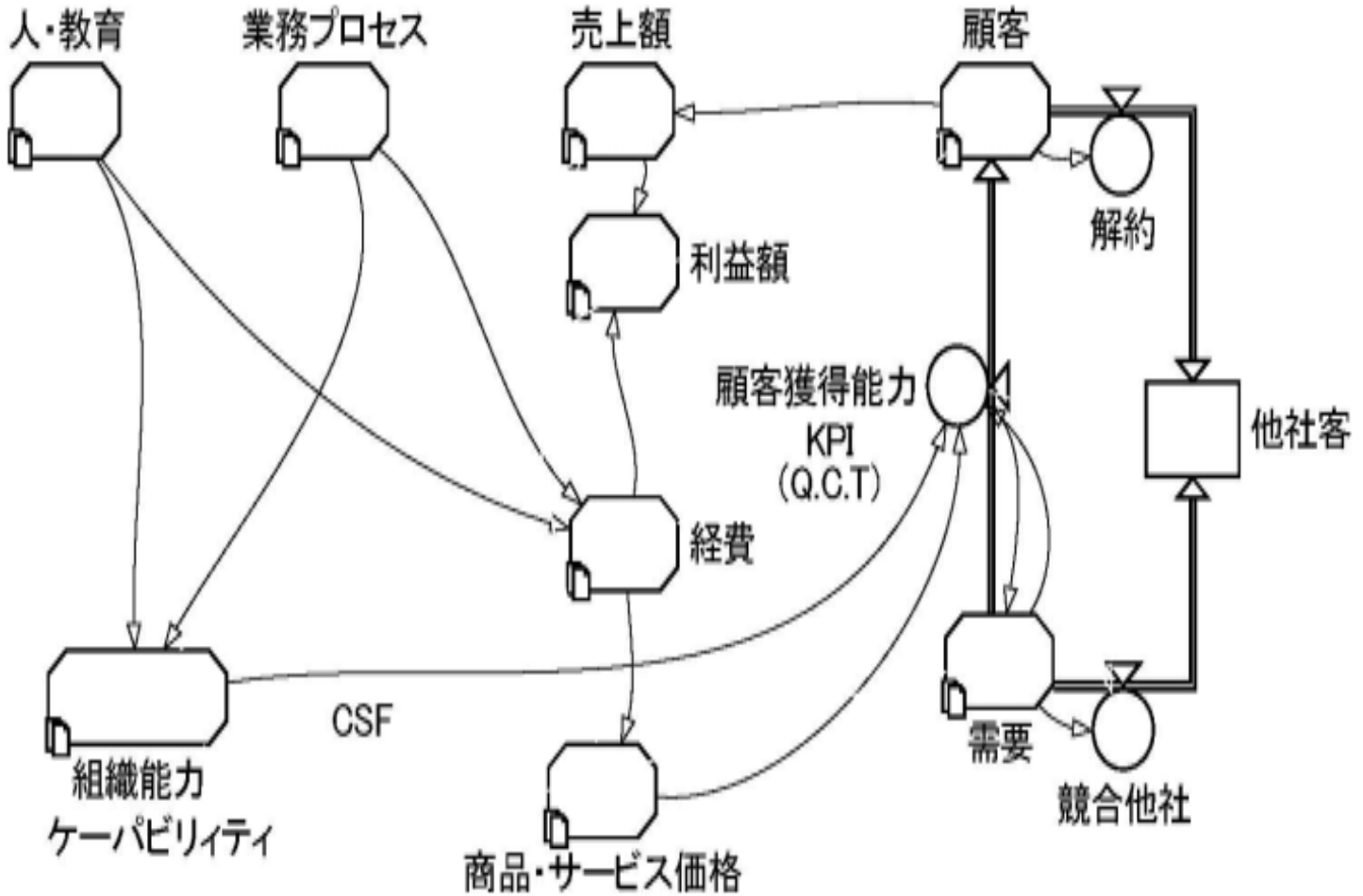
  定義されたプロセスを定量的に管理でき、品質目標およびプロセス実績目標を達成できる。

管理のダブル･ループ

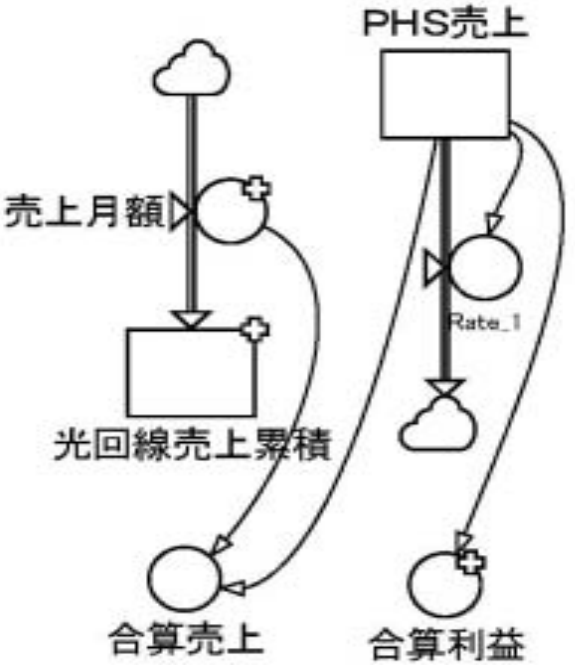
戦略(Strategy)と業務(Operation)のマネジメント（キャプラン）

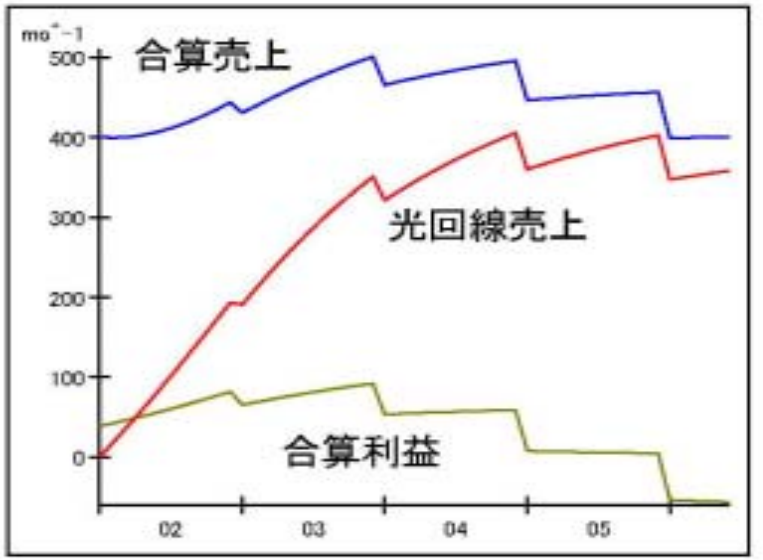


企業能力をマネジメントするBSC

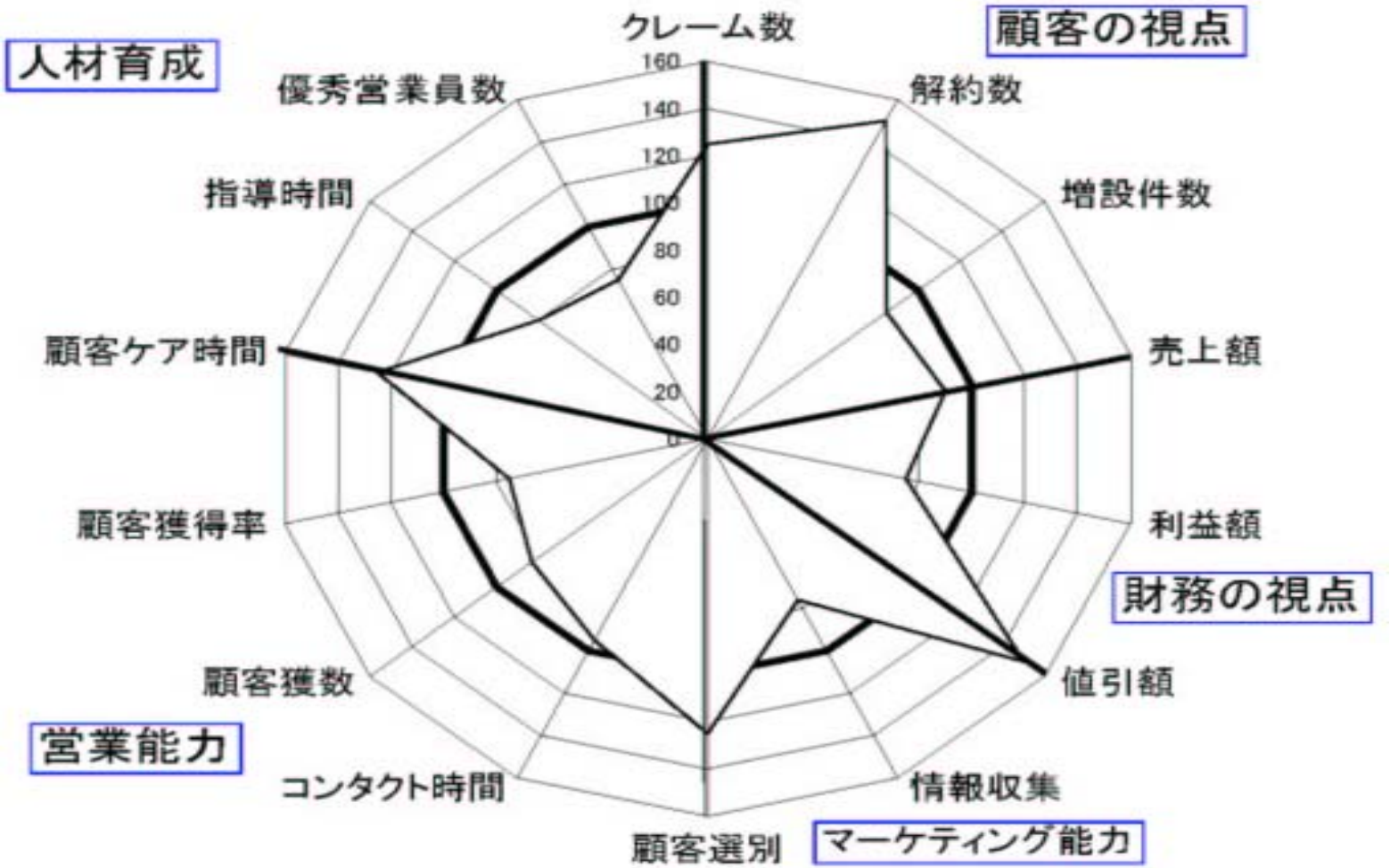


能力を基本とした結果
財務諸表はプロセス能力の結果である



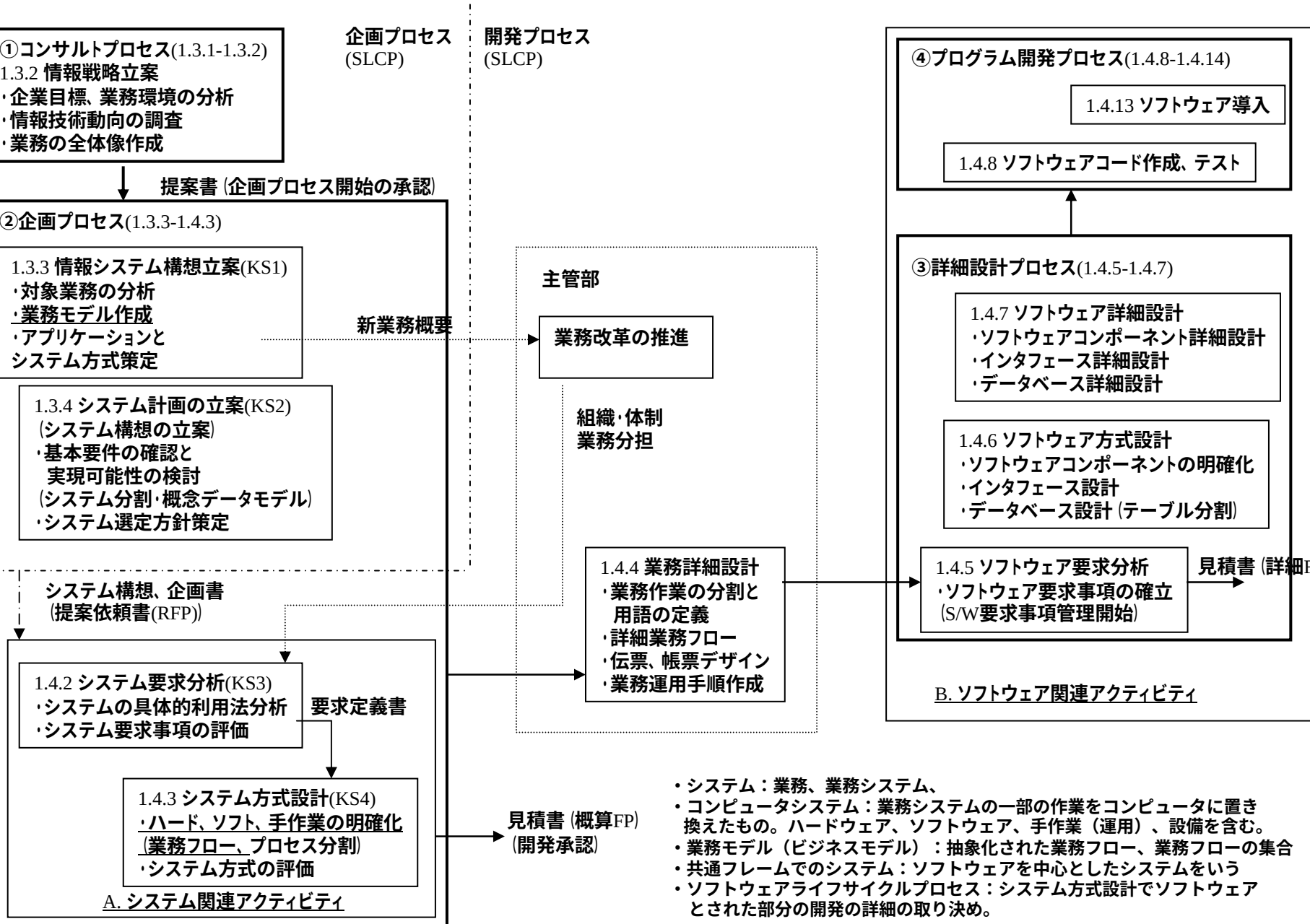


社長の操縦席


クレーム数
顧客の視点
人材育成
優秀営業員数
解約数
指導時間
増設件数
顧客ケア時間
売上額
顧客獲得率
利益額
財務の視点
顧客獲数
値引額
営業能力
コンタクト時間
情報収集
顧客選別
マーケティング能力
160
140
120
100
80
60
40
20
0

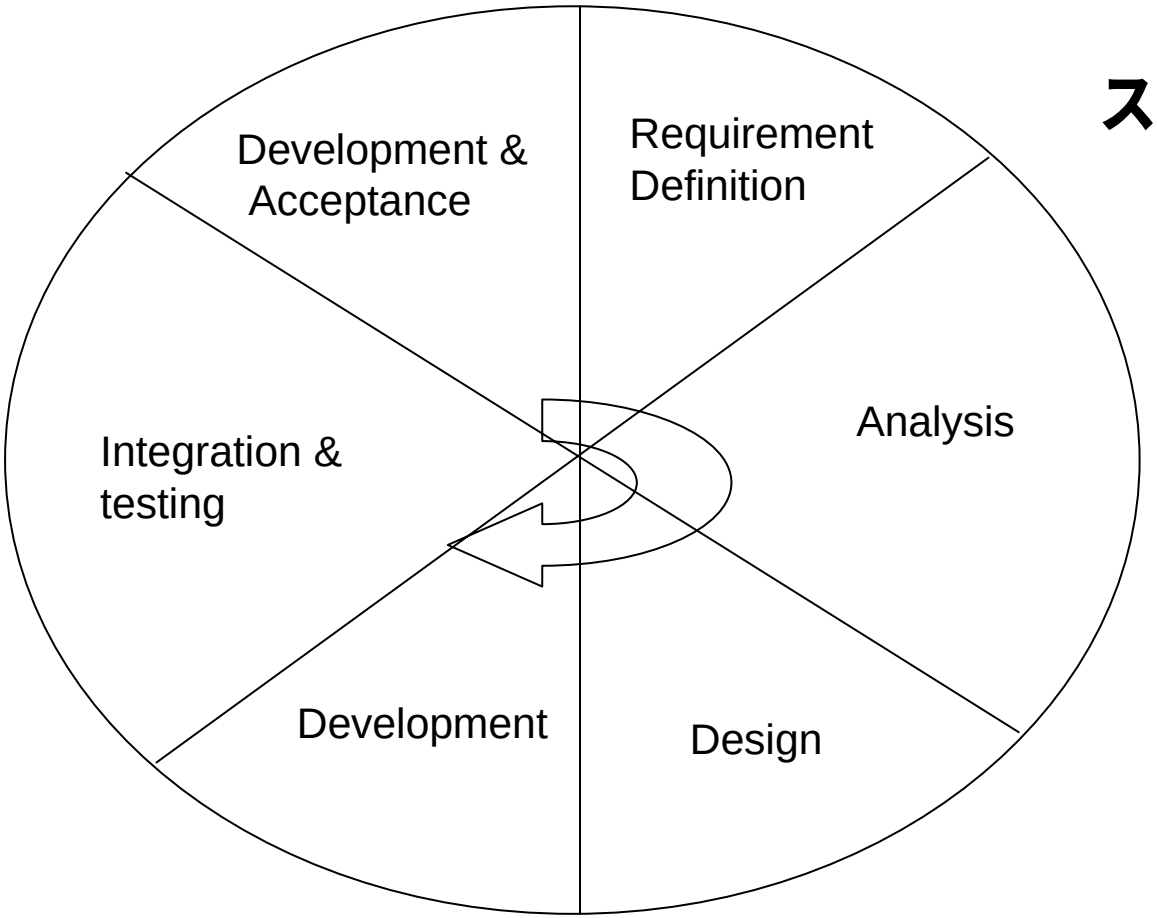
新プロセスとSLCPの関係 (2000/4/5)

TSIの新プロセス



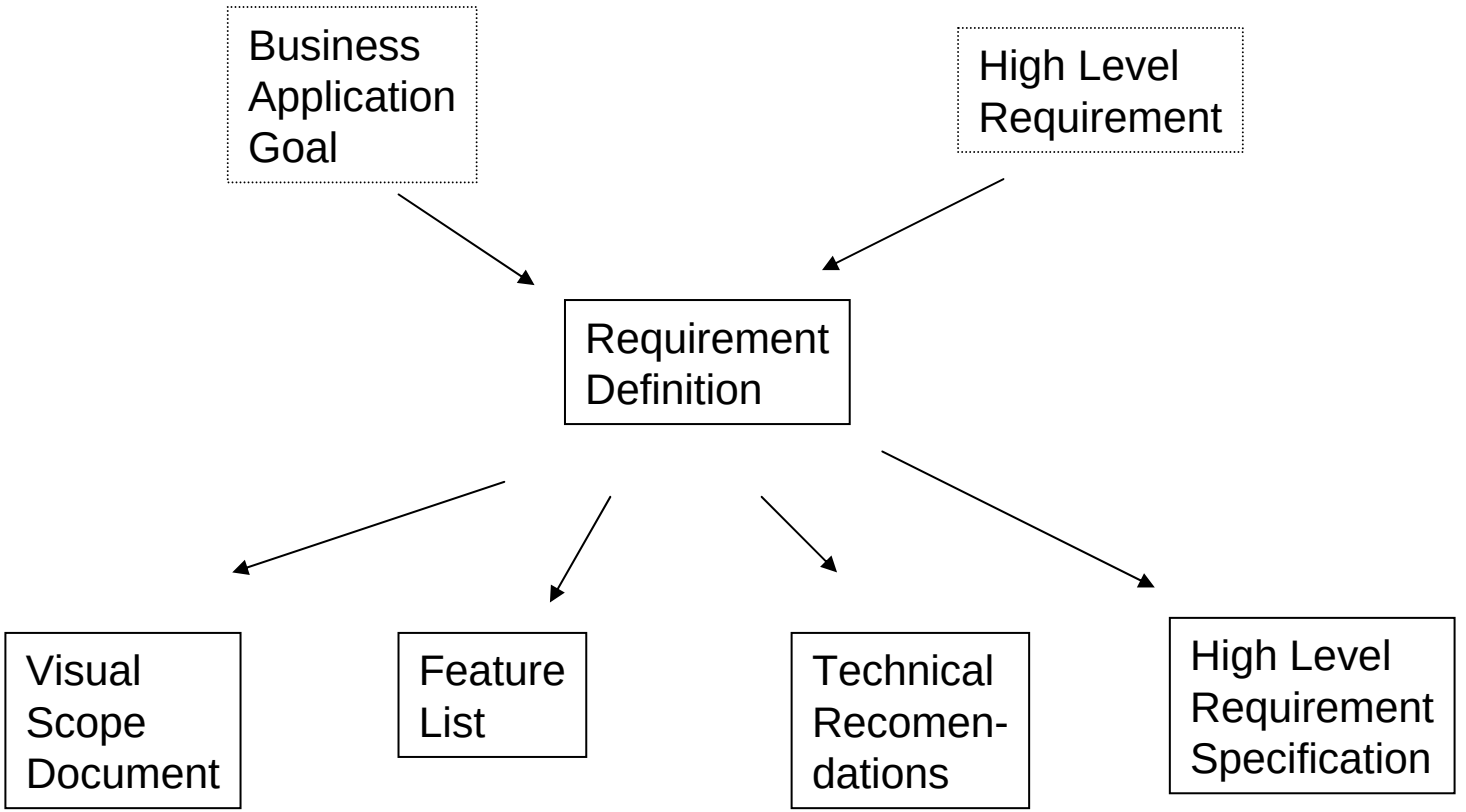
- システム：業務、業務システム、
- コンピュータシステム：業務システムの一部の作業をコンピュータに置き換えたもの。ハードウェア、ソフトウェア、手作業（運用）、設備を含む。
- 業務モデル（ビジネスモデル）：抽象化された業務フロー、業務フローの集合
- 共通フレームでのシステム：ソフトウェアを中心としたシステムをいう
- ソフトウェアライフサイクルプロセス：システム方式設計でソフトウェアとされた部分の開発の詳細の取り決め。

SLCP
(管理プロセスとエンジニアリングプロセス)

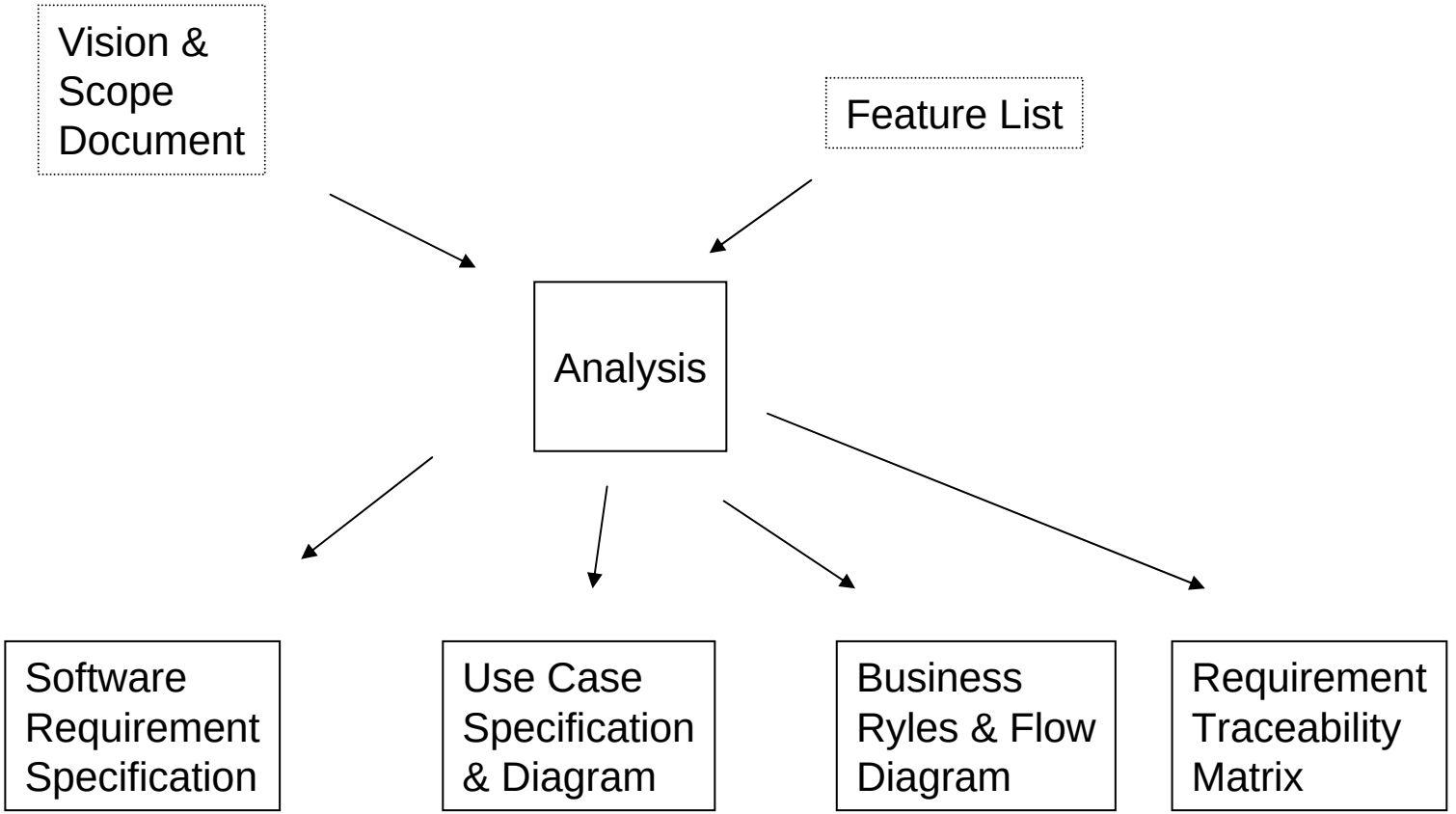


スパイラル開発

Requirement Definition
管理プロセス(SLCP)との整合性



Analysis


Vision & Scope Document
Feature List
Analysis
Software Requirement Specification
Use Case Specification & Diagram
Business Ryles & Flow Diagram
Requirement Traceability Matrix

IT開発プロセスのTOC

Theory of Constrain ゴルードラッド1980前半

- **工場の生産性はボトルネック工程以上の能力にはならない。・・・「ザ・ゴール」**

  ドラム・バッファ・ロープ(DBR)

- **制約条件を見え易くし、制御する。**

  クリティカル・チェーン

何を・何に・どうやって変えるか

継続的改善

松本憲洋：モデル・ベースト経営、JSD学会誌,2003

プログラミングの生産性の把握の困難さ
テクノロジは進歩している

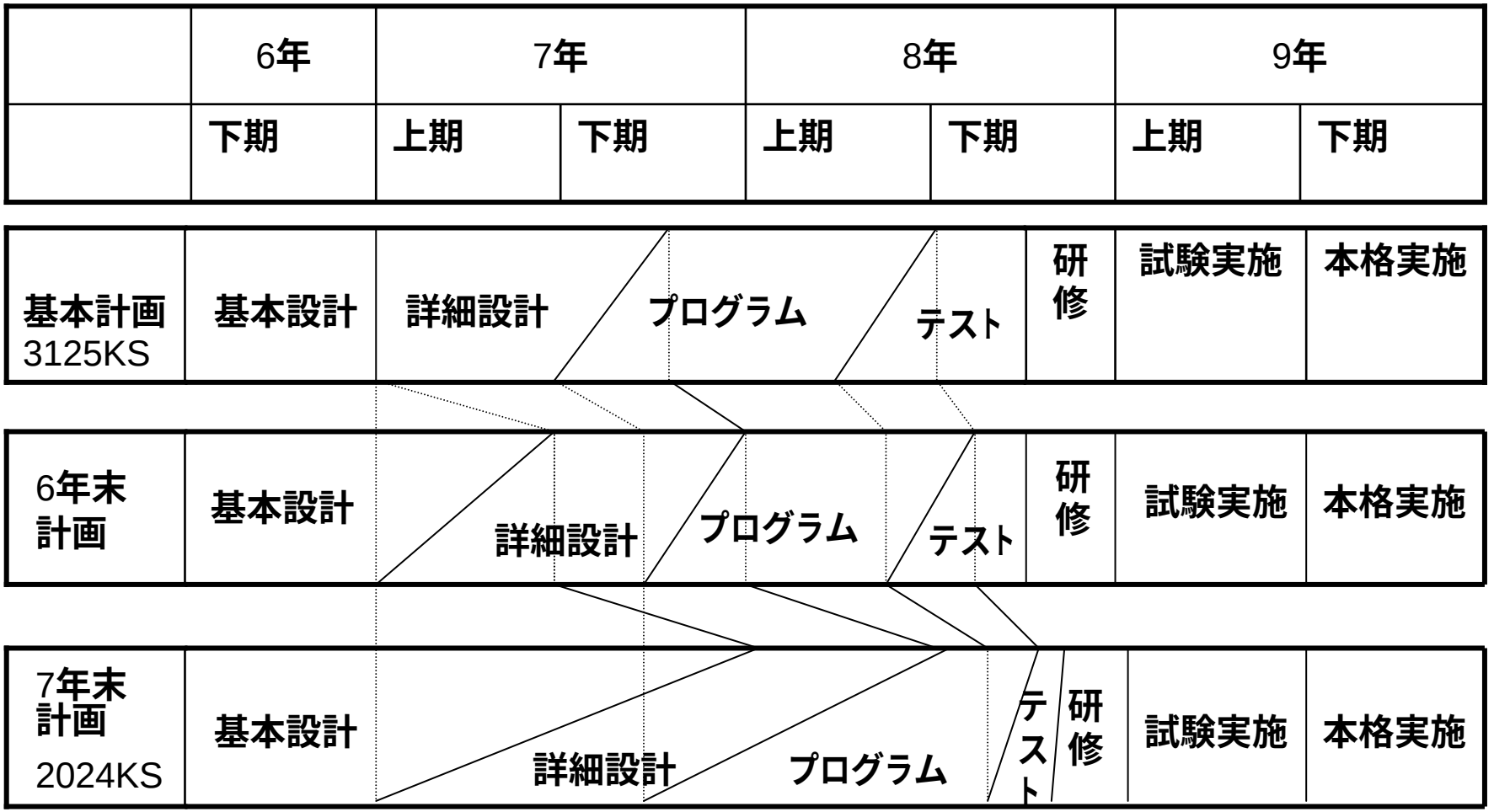
- **パンチカード時代**
- 机上デバッグ
- コンパイル1回
- テスト　2～3回
- **画面エディター時代**
- １０～２０ショット
- 3回／1日　大型汎用機昭和６０年頃
- 1ショット／2時間　UNIX機
- **最近の**IDE
- エディターで構文チェック
- ソースコードデバッガー
- メモリーリーク
- リグレッションテスト環境

ソフトウェア・テスト工学の講座のない日本
テストツール開発ベンダーが育たなかった日本

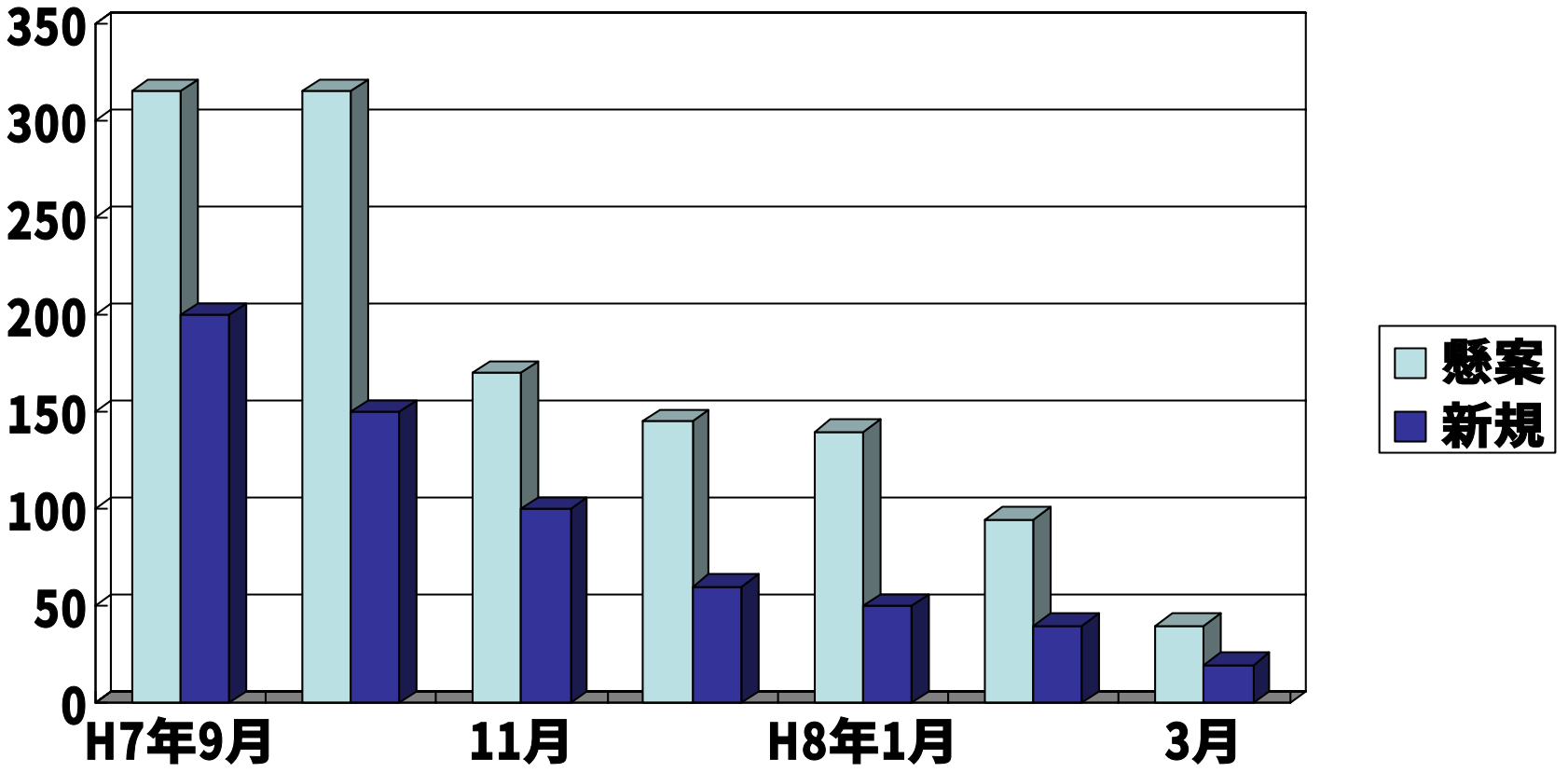
- 生産性・見積もりができない
- 金融で半年、公共システムで3ヶ月、製造業で1ヶ月
- 歴史のあるソフトウェアテスト工学学会・・・情報処理学会から分派しなかった。
- 日本では4，5人
- カバレッジ率と品質
- デバッグとテストの違い・・・オフショア開発
- リグレッションテストと生産性
- 潜在欠陥数・・・組み込み系では実績がある

要件定義プロセス (9752人月)

Derive, Elicit, Extract ->仕切り・にぎり(Engage Management)



懸案事項



懸案事項の分類

業務政策	5%	資材譲渡代金の相殺方式 譲渡導入による工事残品の会計方式 技術協議工程における営業･配電部の分担	主管部門	
業務設計	30%	各店の管理方式の統一 標識･設備名の統一 共架改善業務の機械化範囲 地中線の管理項目の範囲 電柱番号振り出し、ジュイント終端の管理方法		
ソフト設計	40%	担当者コードの体系 設計書のセキュリティ方式 設計変更履歴方式 図面補正のタイミングと管理の整合性 設計図書の長期管理と検索方式 業務標準パターンの設定		システム部門
基盤設計	25%	一次記憶の使用法の統一 サーバー運用管理ソフトの決定 複数営業所またがりDB設計 ポップアップとリストボックス設計基準		

フィードバックのある米国海軍調達

手戻りのあるIT開発モデル

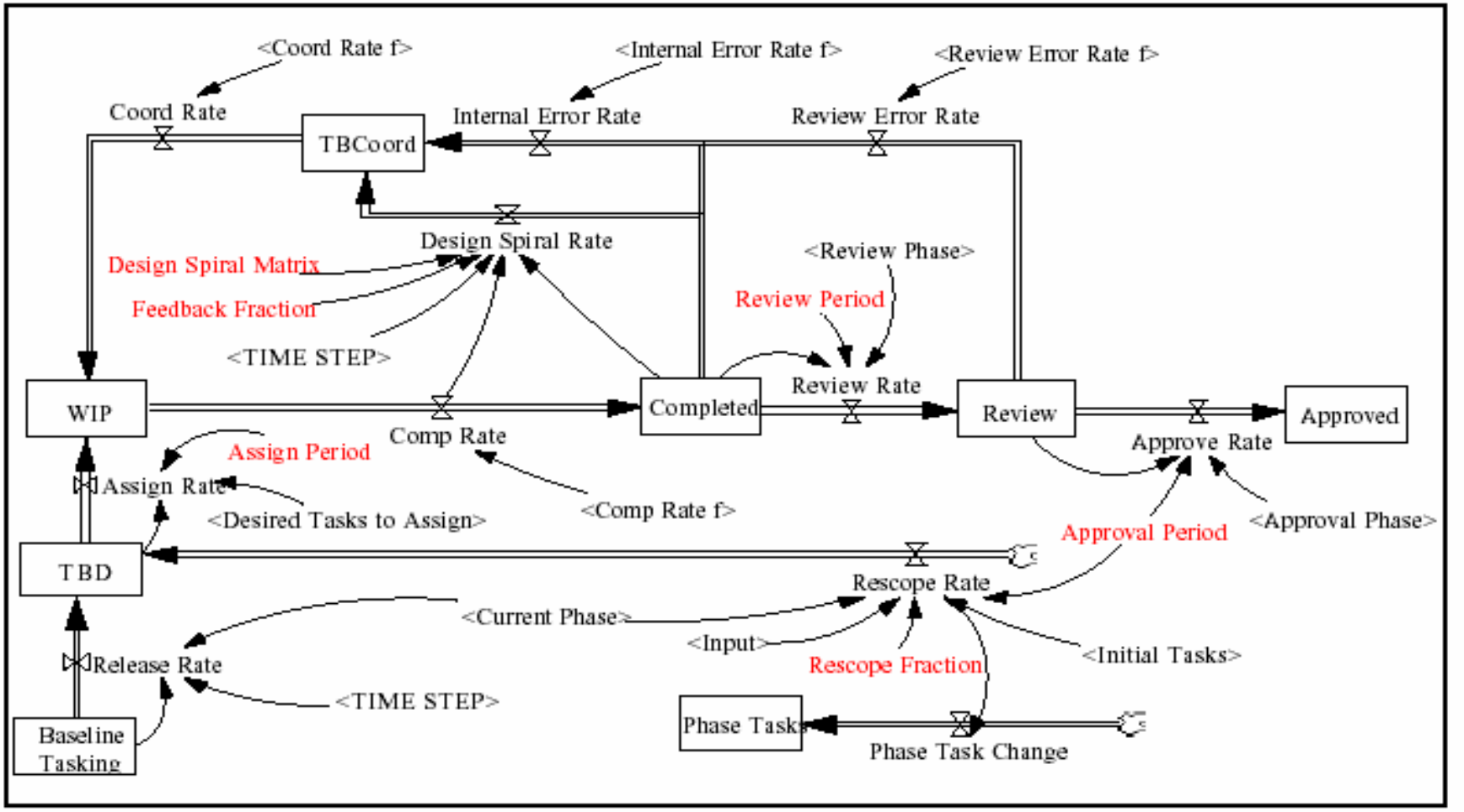
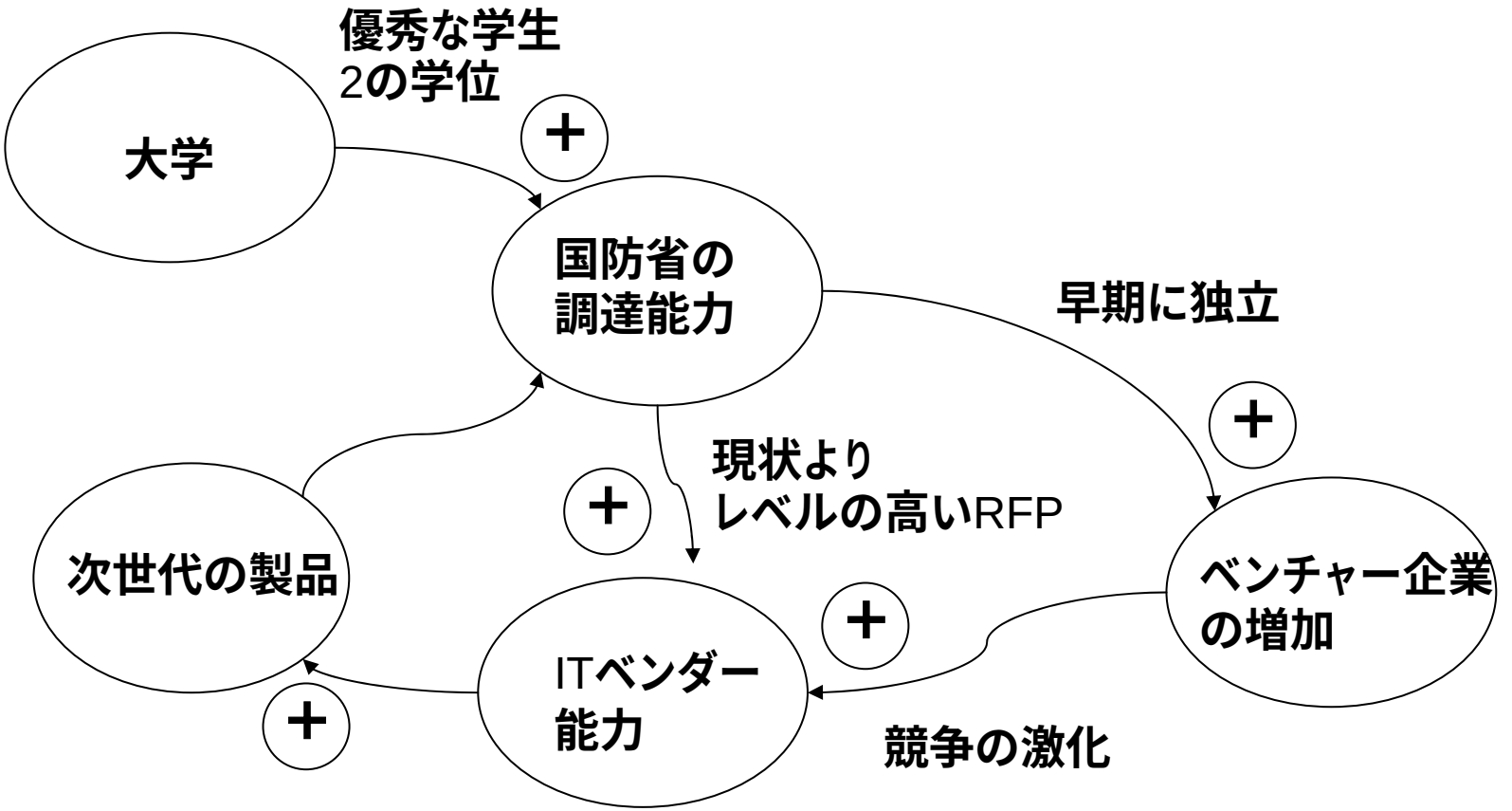


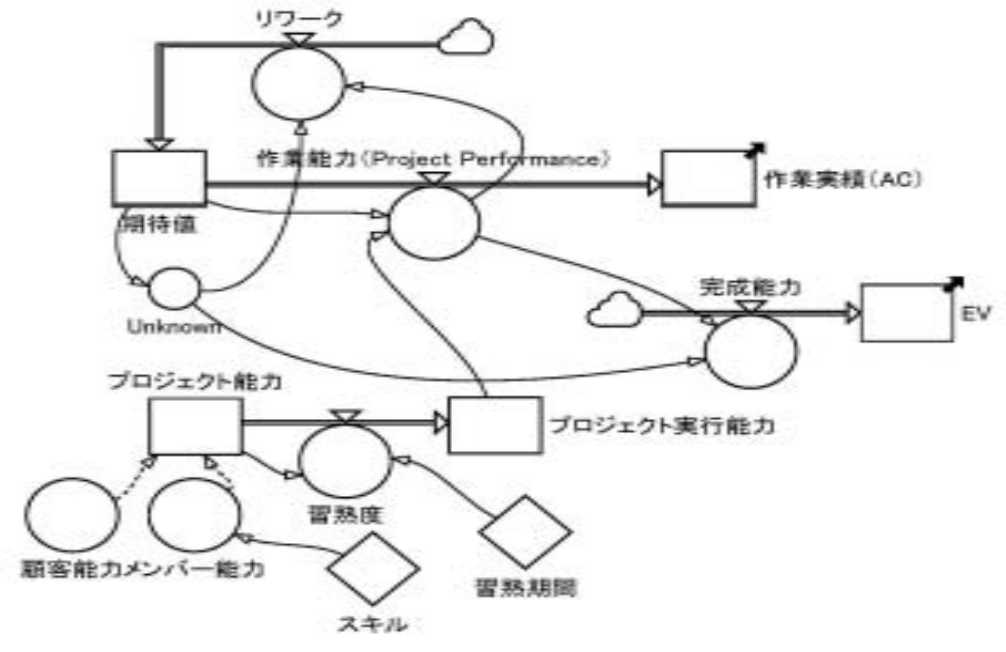
Figure 12 Naval Design Process Model Task Accomplishment Structure

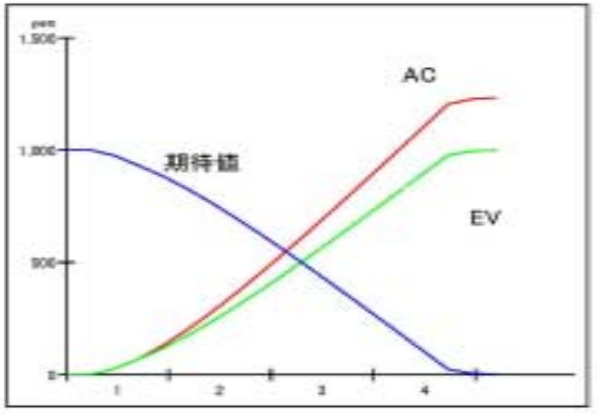
日本と異なるコンピュータの調達環境

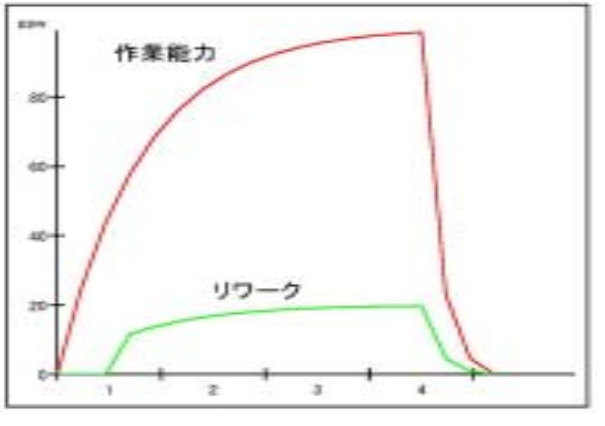


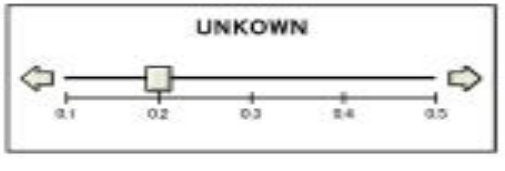
システム開発のジェネリックモデル

リワーク、習熟度を考慮したサブモデル









ジェネリックの要素

1） **期待値**(Expectation EP)：**顧客が一定の費用で、期限内に獲得したい機能や品質（単位：円）**

2） **作業能力**(Project Performance per week)：**一定期間内に価値を完成する能力（単位：円／週又は月）**

3） **獲得価値**(EV)：**一定期間内に完成したEPに見合った作業価値（単位：円）**

4） Unknown：**作業中に発見される未知の作業（単位：円）**

価値を高める追加機能ではなく、当初想定した価値を維持する為には必要な機能。発生量は前工程の品質に依存し、前工程の欠陥（洩れや間違い）が原因である。

1/3位作業に入って時点で発見される。

ITSSを使った期待値の計算

顧客の投資局面（職能・能力別単価）

職種	スキル	達成度レベル
コンサルタント	経営戦略 ビジネス戦略策定	4〜7
ITアーキテクト	課題整理／分析 ソリューション設計	4〜7
アプリケーション スペシャリスト	コンポーネント設計 ソリューション構築	1〜6
プロジェクト マネジメント	マネージング	3〜7

職種別人工　＝仕事量と難しさ　÷　作業期間

期待値（金額）＝ Σ（職種別人工　×　職種別単価）

ITSSは
米国IBMのリストラクチャリングの成功事例

- ハードウェアビジネスからソリューションビジネスの事業特性の再構築
- 顧客の視点
- 成果主義・・・達成度評価
- 技術者の知識共有
- 保守はない

ITスキル標準（概要）

Tサービス・プロフェッショナル
成の基盤構築に向けて
(Ver 1.1)2003.7
キル標準の必要性
辞書としてのスキル標準
プロフェッショナル育成への貢献
キャリアパスのモデル事例

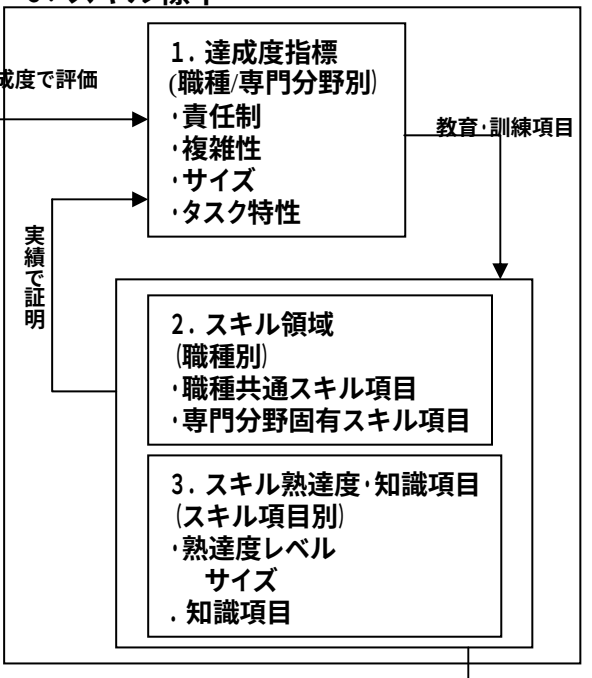
2．ITSSガイドの構成

資産業省　IT人材の育成
w.meti.go.jp/policy/it_policy/index.html

2．スキル・フレームワーク（職種と達成度）

職種	専門分野	レベル
マーケッティング	・マーケティングマネージメント	5〜7
	・販売チャネル戦略	4 〜 6
	・マーケットコミュニケーション	3 〜 6
セールス	・訪問型コンサルティングセールス	1 〜 7
	・訪問型製品セールス	1 〜 6
	・メディア利用型セールス	1 〜 5
コンサルタント	・BT (Business Transformation)	4 〜 7
	・IT	4 〜 7
	・パッケージ適用	4 〜 6
ITアーキテクト	・アプリケーション	4 〜 7
	・データサービス	5 〜 7
	・ネットワーク	5 〜 6
	・セキュリティ	5 〜 6
	・システムマネジメント	5 〜 7
プロジェクトマナジメント	・システム開発/AP開発/SI	3 〜 7
	・アウトソーシング	6 〜 7
	・ネットワークサービス	3 〜 6
	・eビジネスソリューション	6 〜 7
	・ソフトウェア開発	3 〜 6
ITスペシャリスト	・プラットフォーム	1 〜 6
	・システム管理	1 〜 6
	・データベース	1 〜 6
	・ネットワーク	1 〜 6
	・分散コンピューティング	1 〜 6
	・セキュリティ	1 〜 6
アプリケーション スペシャリスト	・業務システム	1 〜 6
	・業務パッケージ	1 〜 6
ソフトウェア デベロップメント	基本ソフト	1 〜 6
	ミドルソフト	1 〜 6
	応用ソフト	1 〜 6
カスタマサービス	・ハードウェア	1 〜 5
	・ソフトウェア	1 〜 5
	・ファシリティマネジメント	1 〜 6
オペレーション	・システムオペレーション	1 〜 5
	・ネットワークオペレーション	1 〜 5
	・カスタマサポート	1 〜 5
エデュケーション	・研修企画	4 〜 6
	・インストラクション	3 〜 6

3．スキル標準



4．研修ロードマップ(2003.7)

ITSSに対応した研修体系の参照モデル
・カリキュラムではない
・研修コース設計は専門企業でやる
1）研修コース群（職種別体系図）
実務能力の向上の為の知識習得
・レベル
　未経験レベル
　エントリレベル(レベル1〜2)
　ミドルレベル(レベル3〜4)
　ハイレベル(レベル5〜7)
・研修分野
　テクノロジ
　メソドロジ
　プロジェクトマネジメント
　ビジネス/インダストリ
　パーソナル

判断資料：達成度レベルの熟達度についての補足資料

達成度レベルの考え方

〇企業における職種・専門分野の重要度
- **・業務としての重要性**
- **・業務としての難易度**
- **・ビジネスへの貢献度**

〇企業が必要とする技能の難易度
- **・技術面での技能習得の難易度**
- **・修得・熟達までの期間**

達成度指標

・職種・専門分野の各「実務能力」レベルのプロフェッショナルとして認められる為に必要な成功裡に完了した経験と実績の要件を定義

成功裡に完了しない
- **・開発中である**
- **・契約に影響した**
  裁判になる、和解する
- **・サービスイン（本番稼動）しない**
  中止、中断になる

達成度指標事例（プロジェクトマネジメント）

1．責任性：実施した業務における責任の範囲を規定
プロジェクト全体責任、サブプロジェクト責任、
チームリーダの責任、メンバーとしての実施責任

・プロジェクト規模（サイズ）とリスク度（複雑性）の組み合わせによりプロジェクト責任者の経験レベルを設定

2．複雑性：業務遂行の難易度を示す業務内容・環境と経験数を規定

・複雑性の指標はプロジェクトを成功裡に完遂させる為には避けられぬリスク要素

①プロジェクトのリスク
- **－契約の履行に影響する要素・要因**
- **－品質、コスト、進捗に影響する要素・要因**

②ミッションクリティカルなプロジェクト
- **－プロジェクトの不成功が社会/企業において大問題を引き起こす可能性がある要素・要員を秘めたプロジェクト**

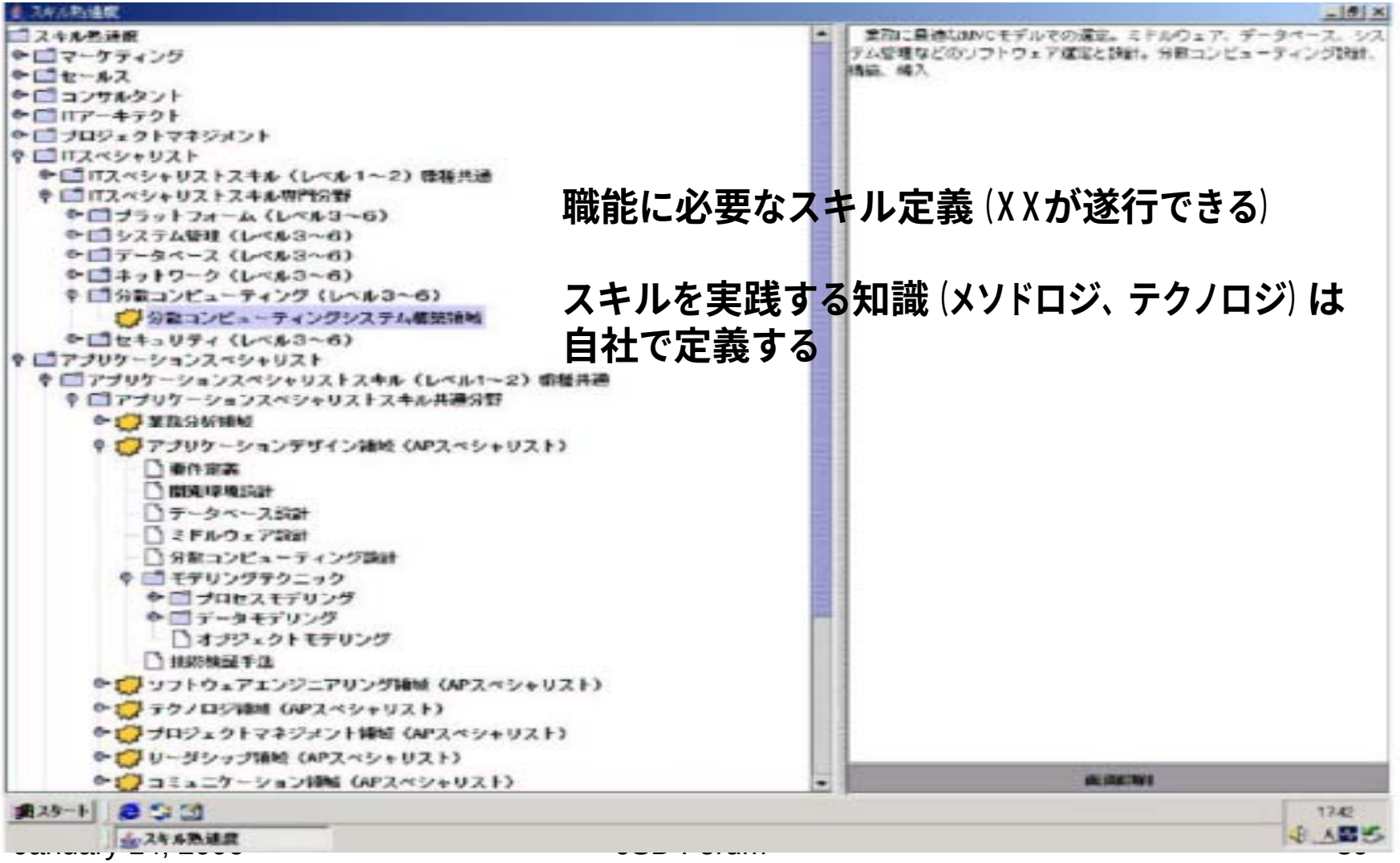
3．サイズ：ビジネス規模やプロジェクト規模を規定

4．タスク特性：プロフェッショナルとして要求される専門性の活用に関する規定
- **・プロジェクト内専門性活用**
- **・外部・内部（コミュニティ）活動・論文**
- **・後進の指導・育成**

複雑性の事例
- **・システム要件の複雑性**
- **・システムデザインの複雑性**
- **・複雑なアプリケーション要件**
- **・複雑なプロジェクト体制**
- **・国際的なプロジェクト**

ITSSスキルナビゲーター

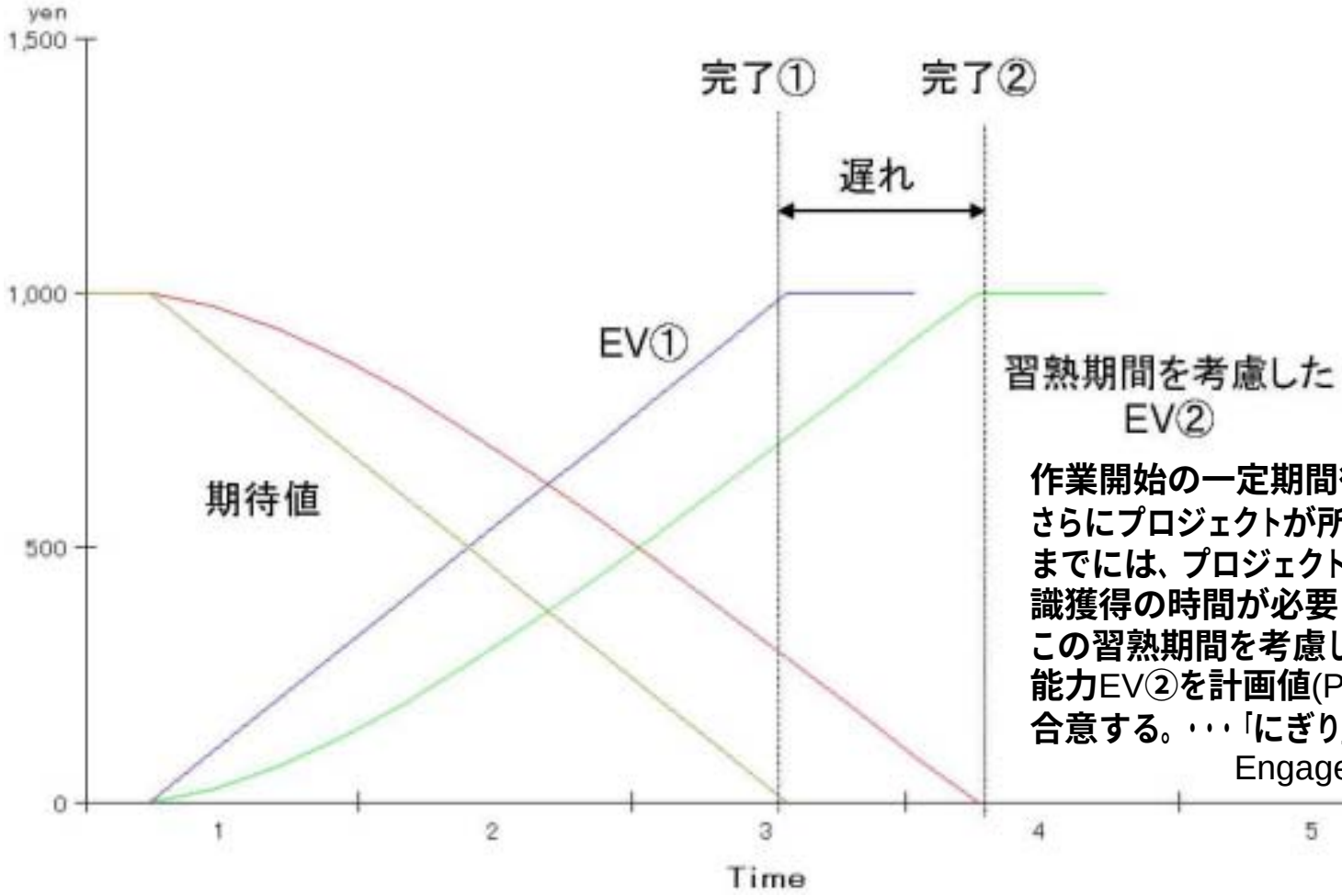
http://www.vector.co.jp/soft/other/java/se361915.html



職能に必要なスキル定義（ＸＸが遂行できる）

スキルを実践する知識（メソドロジ、テクノロジ）は自社で定義する

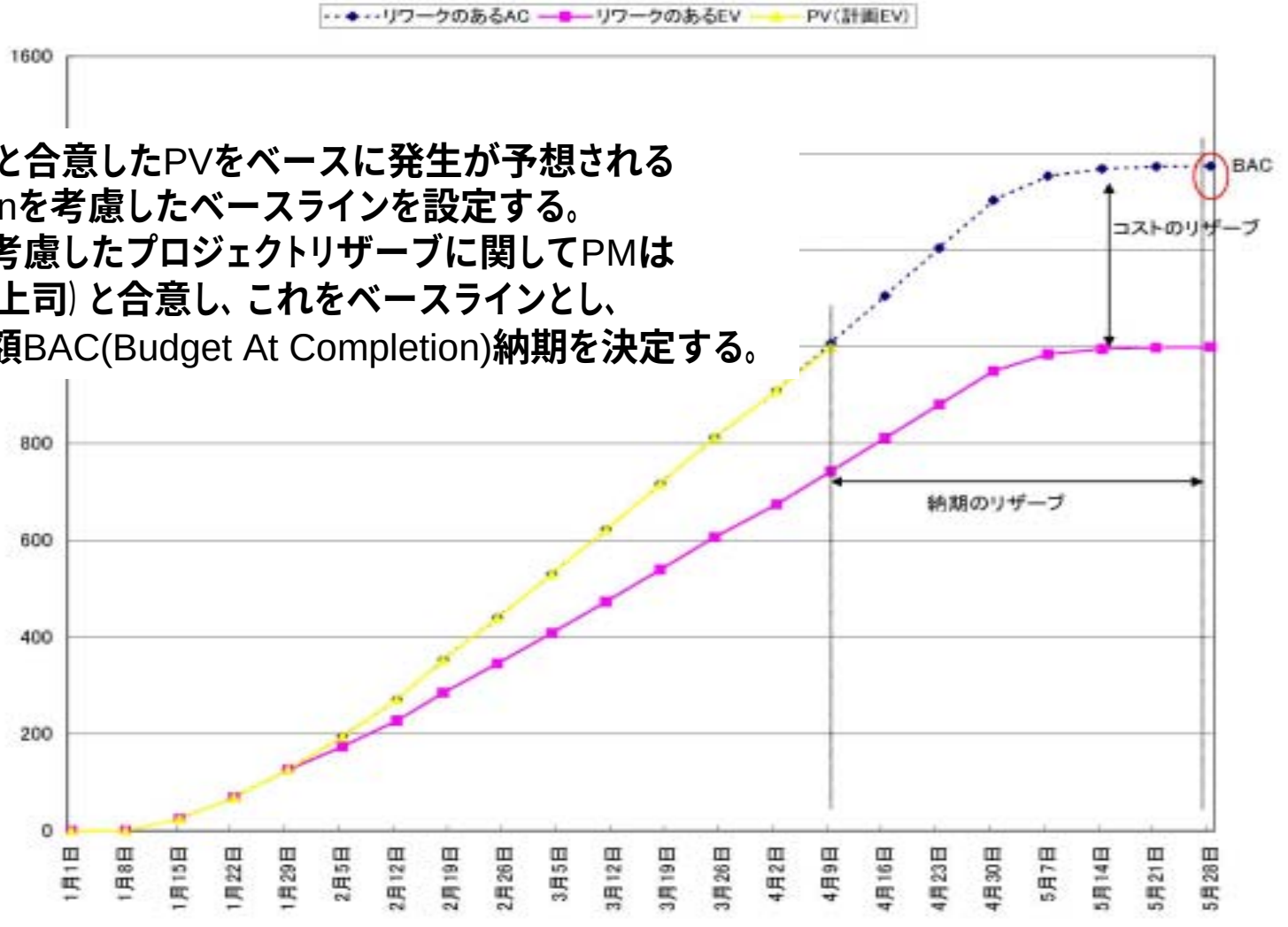
ベースラインの作成(1)
新人SEの計画の2割増



作業開始の一定期間後にEV①が得られる。
さらにプロジェクトが所定の能力を発揮するまでには、プロジェクトメンバーが所定の知識獲得の時間が必要となる。
この習熟期間を考慮したプロジェクトの実行能力EV②を計画値(PV)とし、PMはメンバー合意する。・・・「にぎり」「仕切り」

Engagement Management

ベースラインの作成(2)
中堅SEの計画の1割増

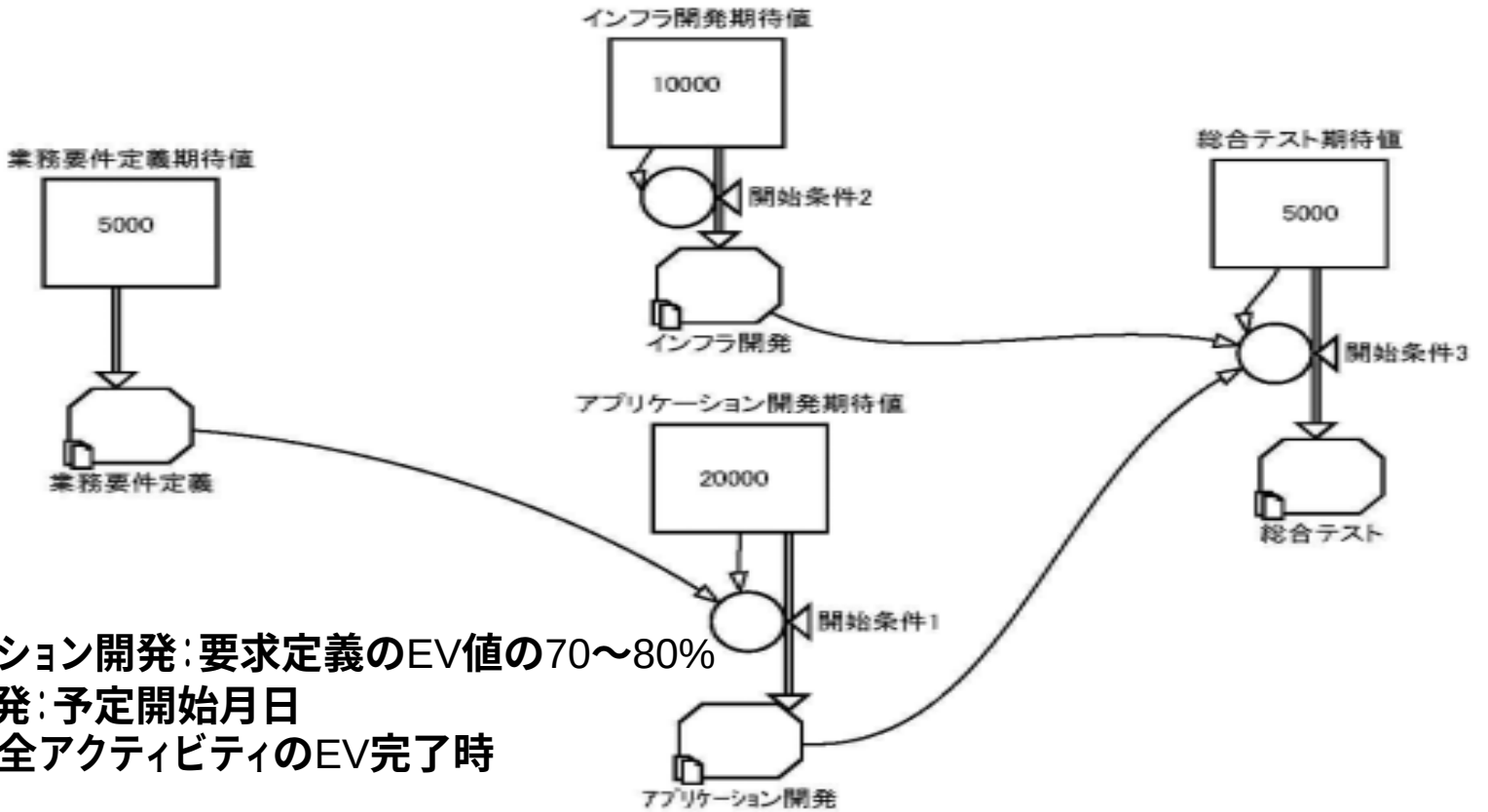


メンバーと合意したPVをベースに発生が予想される
Unknownを考慮したベースラインを設定する。
リスクを考慮したプロジェクトリザーブに関してPMは
経営者（上司）と合意し、これをベースラインとし、
予算金額BAC(Budget At Completion)**納期を決定する。**

Jan

次世代型PERT

ジェネリックを複数組み合わせたプロジェクトモデル



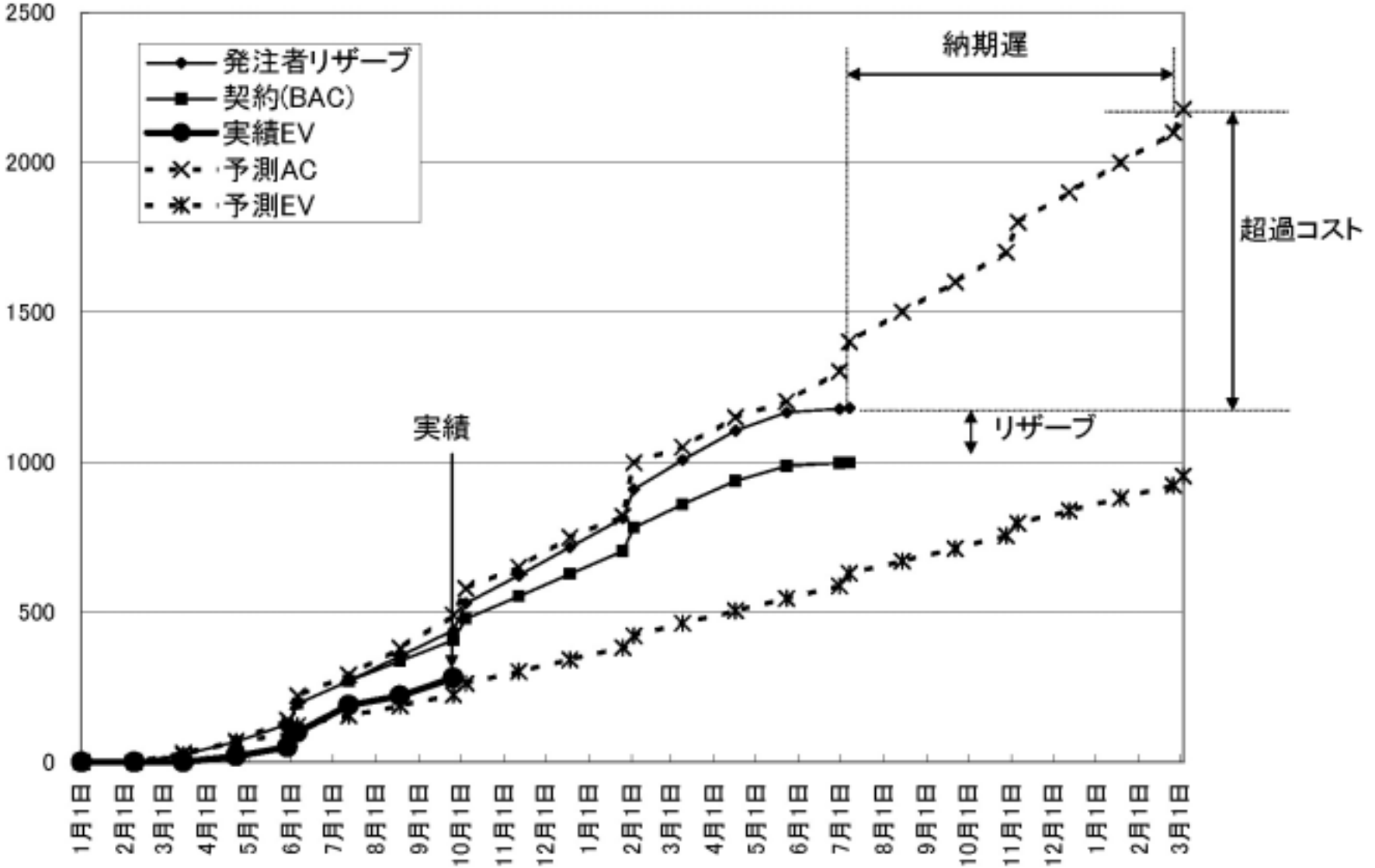
開始条件

・アプリケーション開発：要求定義のEV値の70～80%

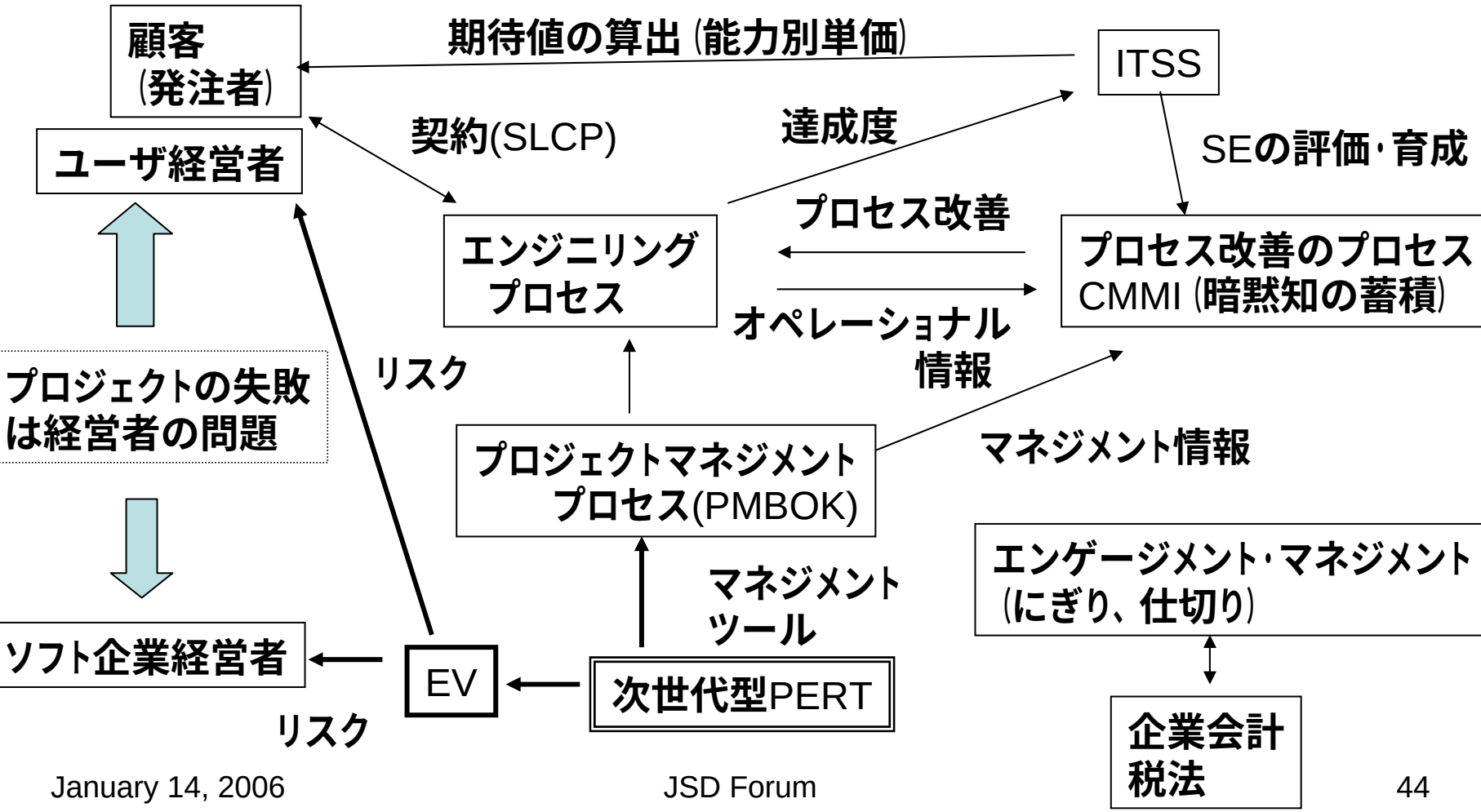
・インフラ開発：予定開始月日

総合テスト：全アクティビティのEV完了時

実績EVを反映した予測



EVマネジメント・プロセス
請負ビジネスの近代化



次世代型PERT(CMMのレベル向上)

Project Evaluation & Review Technique

Project Management Office

NGPERTを使ったエバリューエーション

外部ベテランの参加（デルファイ法）

1人の経験知では不足（金融と製造では違う）

NGPERTを使ったリスク管理

マネジメント情報の蓄積

プロセス･技法･ツールの改善

経営者の参加
リスクの判断

下請けからの脱皮
（技術力向上）

暗黙知の形式知化

組織能力の向上

終わりに

- **計画の重要性・・・どうやって作るか**
- **計画の**Evaluation & Review **のテクニック**
- **モニタリング・・・プロジェクトのパフォーマンス**
- **コントロール・・・パファーマンス・ギャップの分析**
- SD**ツール・・・　これからさらに改良される**

ベテランのメンタルモデル
(ダイナミックスの考え方)

- SDでノウハウをビジュアル化する
- マネジメント情報の予測と実績を蓄積
- オペレーショナル情報 (生産実績) 収集はコストが掛かり、利用価値が低い