

情報システムの巨大化・複雑化と ソフトウェア工学の役割

独立行政法人 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

所長 鶴保 征城

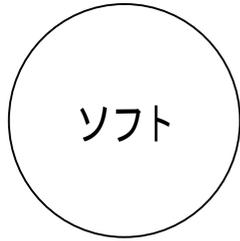
2006年6月9日

アジェンダ

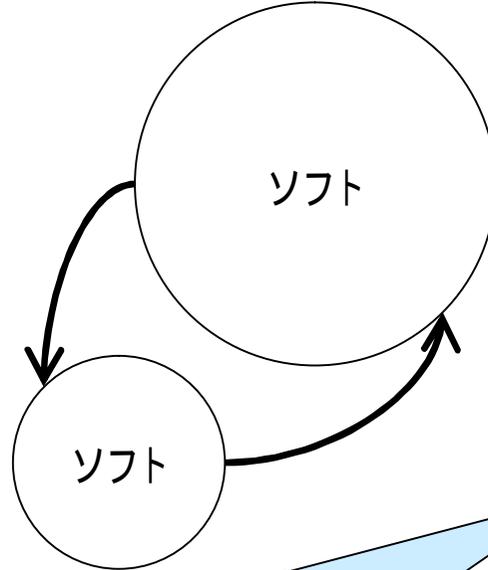
- I. 情報システムの巨大化・複雑化
- II. 人材
- III. ソフトウェアと経営の関係
- IV. 産業競争力の強化に向けて

ソフトウェア開発はだんだん複雑に

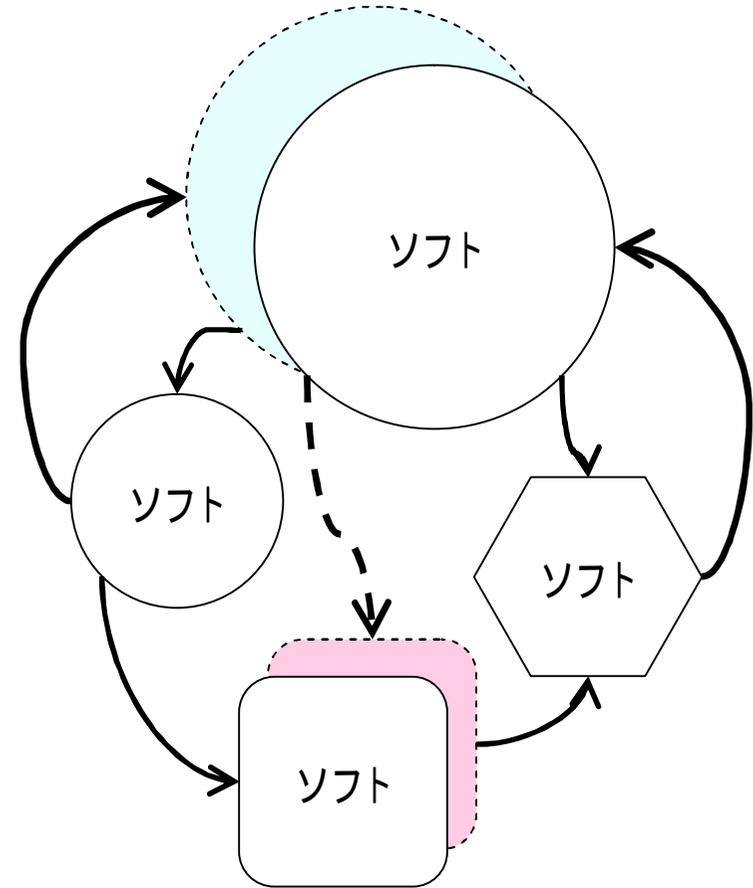
単一システム



ネットワーク化したシステム



システム of システムs

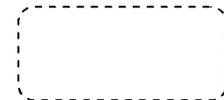


ソフトウェアの変遷:

- サイズは大規模化
- モノリシックな作りから複雑化

人の投入で問題は解決できないと分かりつつも、人月に代わる管理方法が見つからず...

追加・修正 や ラッピング



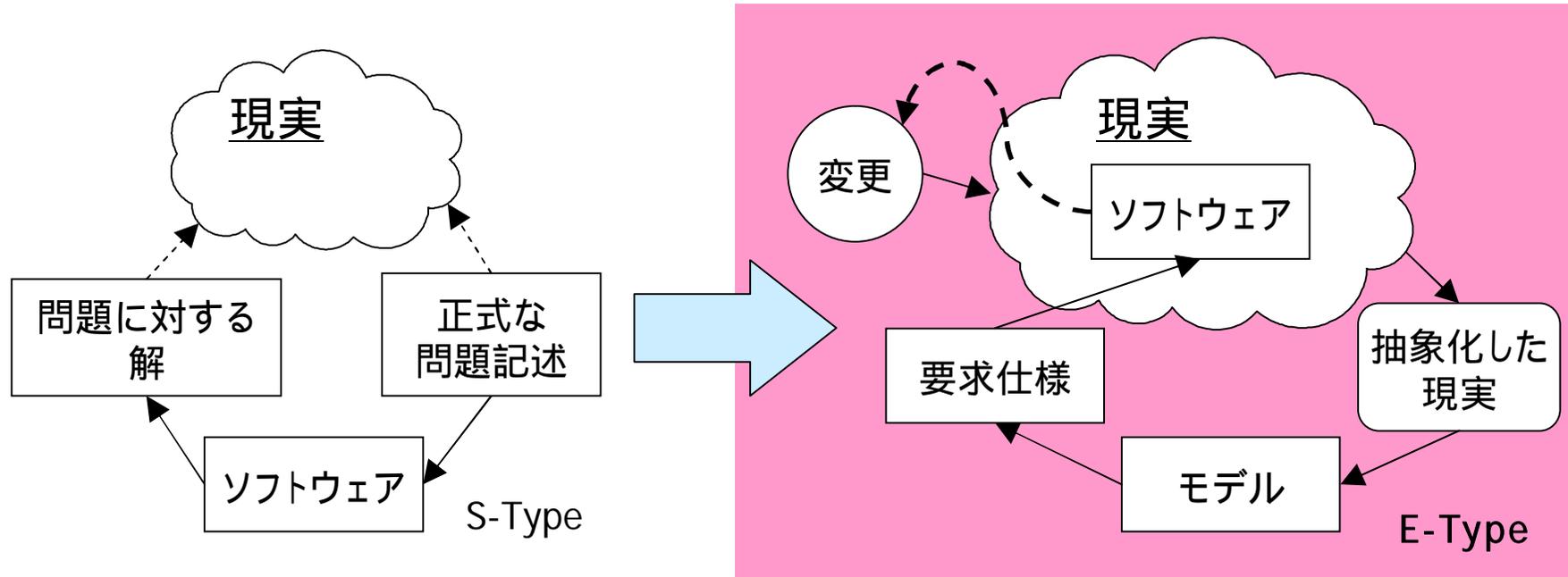
ソフトウェアは本質的に難しい...

- 複雑性 (Complexity)
 - 大きく複雑であること自体が本質的な問題である。
 - 実装レベル近くまで個々に異なる部分からなり、ゆえに、単純なサイズ増で、線形でなく非線形に複雑化する。
- 同調性 (Conformity)
 - 単純な原理は存在しない。
 - 関係する人の曖昧さや矛盾へ対処せねばならない。
- 可変性 (Changeability)
 - 常に「変更」に対する強い圧力にさらされている。
 - ソフトウェア製品は対象ビジネス、ユーザ、法、技術などからなる複雑な要素が絡み合っていてできている、それらは常に変化する。
- 不可視性 (Invisibility)
 - 物理的な世界ではなく、概念の世界でしか捉えることができない。
 - 特定の構造として顕在化することが難しい

ソフトウェア複雑化の理由

製品は「進化」が必然、現在のソフトウェアはE-type

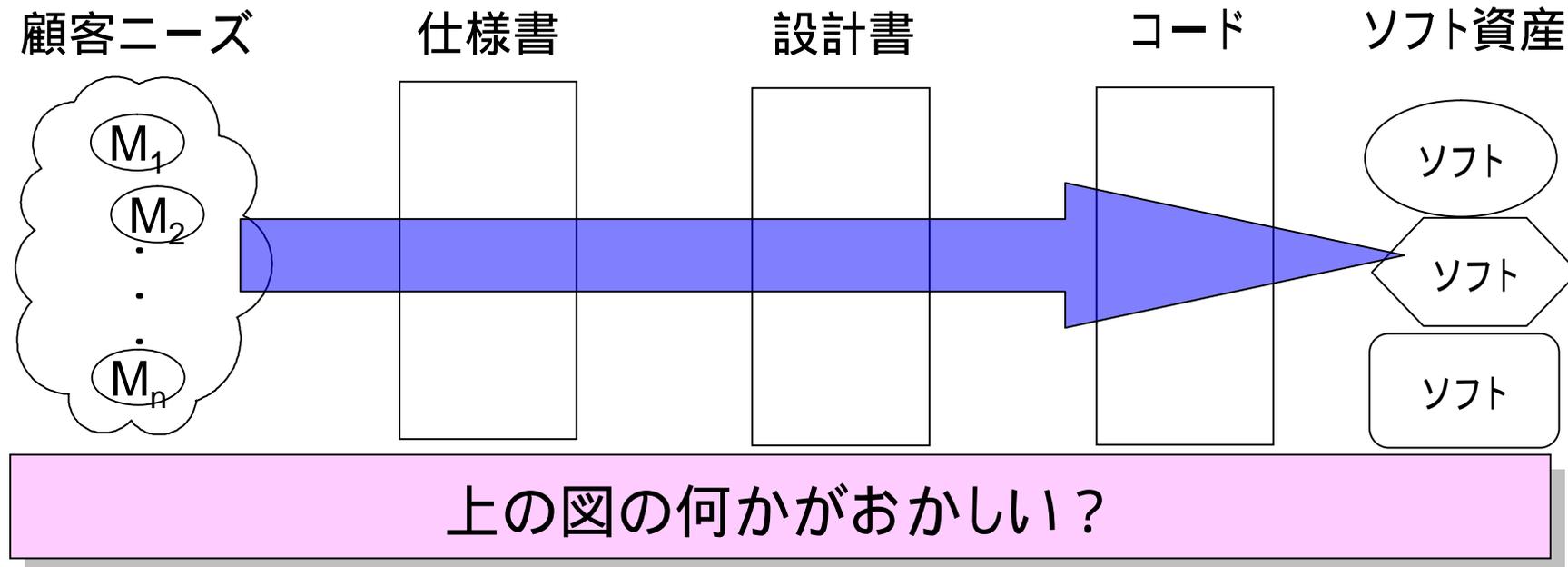
- ソフトウェアは現実の変化の影響を受けるとともに、それ自身が現実を変化させる = ソフトウェアは変化し続ける or 変化せずに次第に有用でなくなる



- ソフトウェアが変化に対応し続けるかぎり、その複雑さが増す
= 複雑さを排する努力が必要となる

厳格に保守し、現実の運用環境の変化に追従する必要がある
そうしなければ、「品質が下がった」と認識される

ソフトウェア進化とウォーター・フォールモデルの欠陥



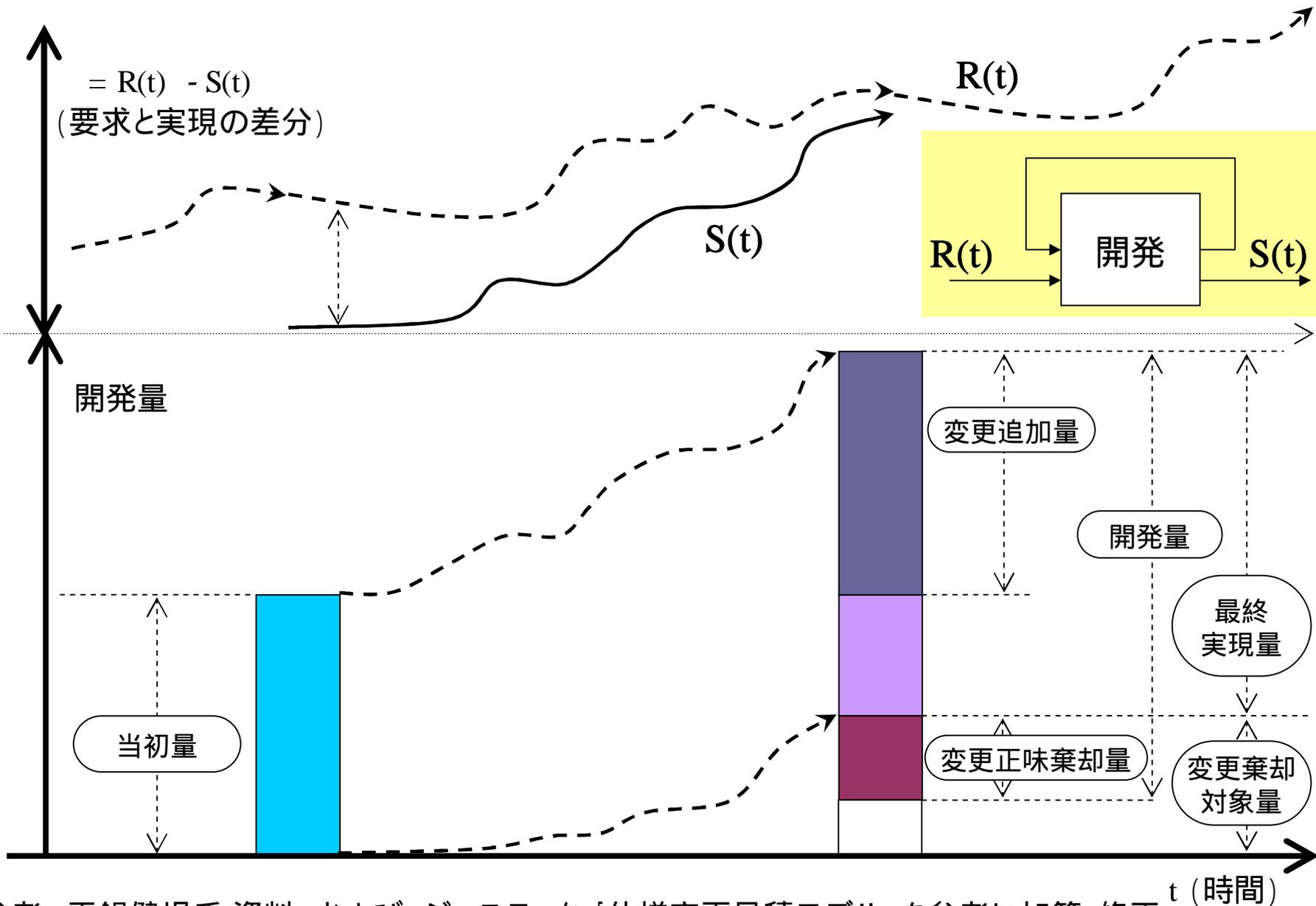
現実には、ウォーター・フォール的な「直線的開発」はできない

1. 状況は進化によって変わる

- 出来上がったソフトウェア資産が現実の一部となり、進化を加速する
- そのほかの要因でも現実が変化する

2. 進化した現実に対応するため、「見えない」要求の変化に随時対応が必要

開発プロジェクトの「成功」はMoving Target



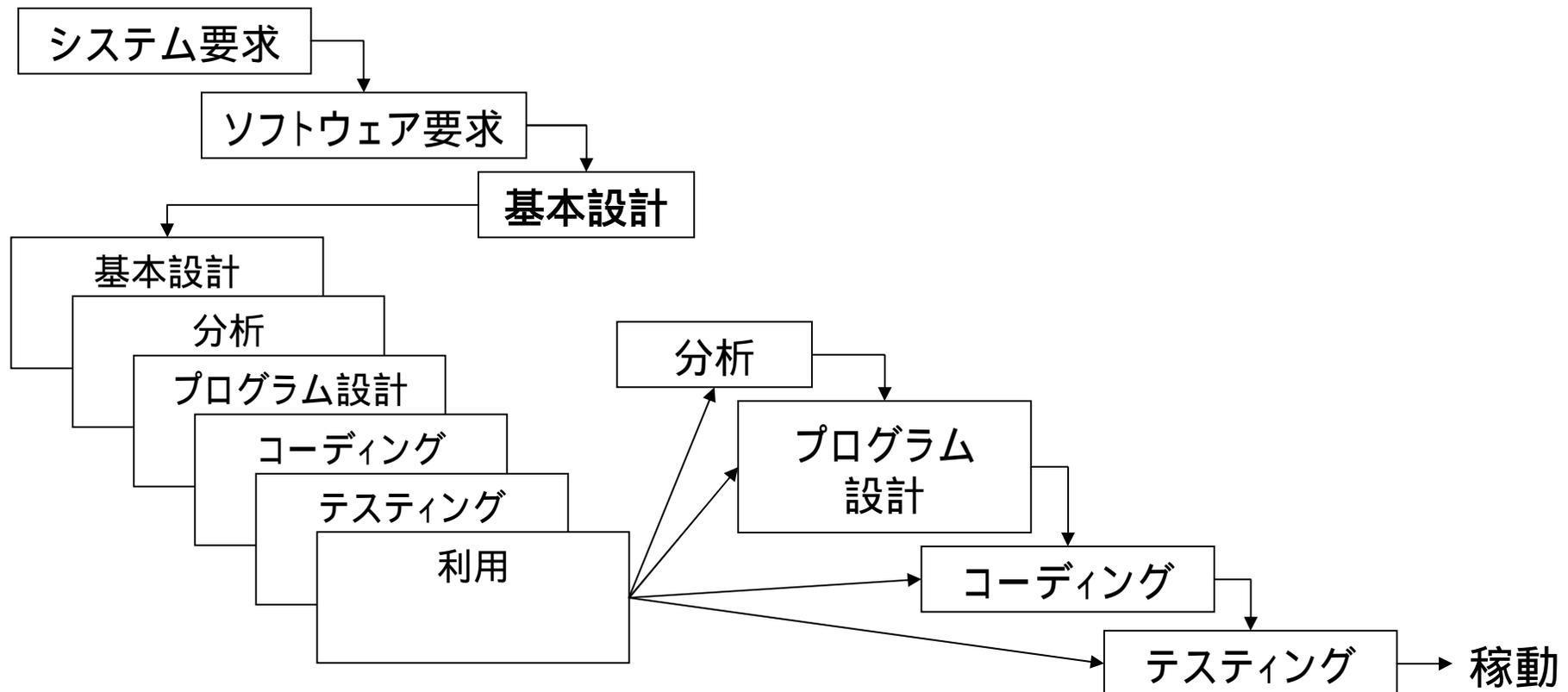
参考: 平鍋健児氏 資料、および、ジャステック「仕様変更見積モデル」を参考に加筆・修正

Waterfall Model 再考

W. Royce, “Managing the Development of Large Software Systems: Concepts and Techniques,” 1970.

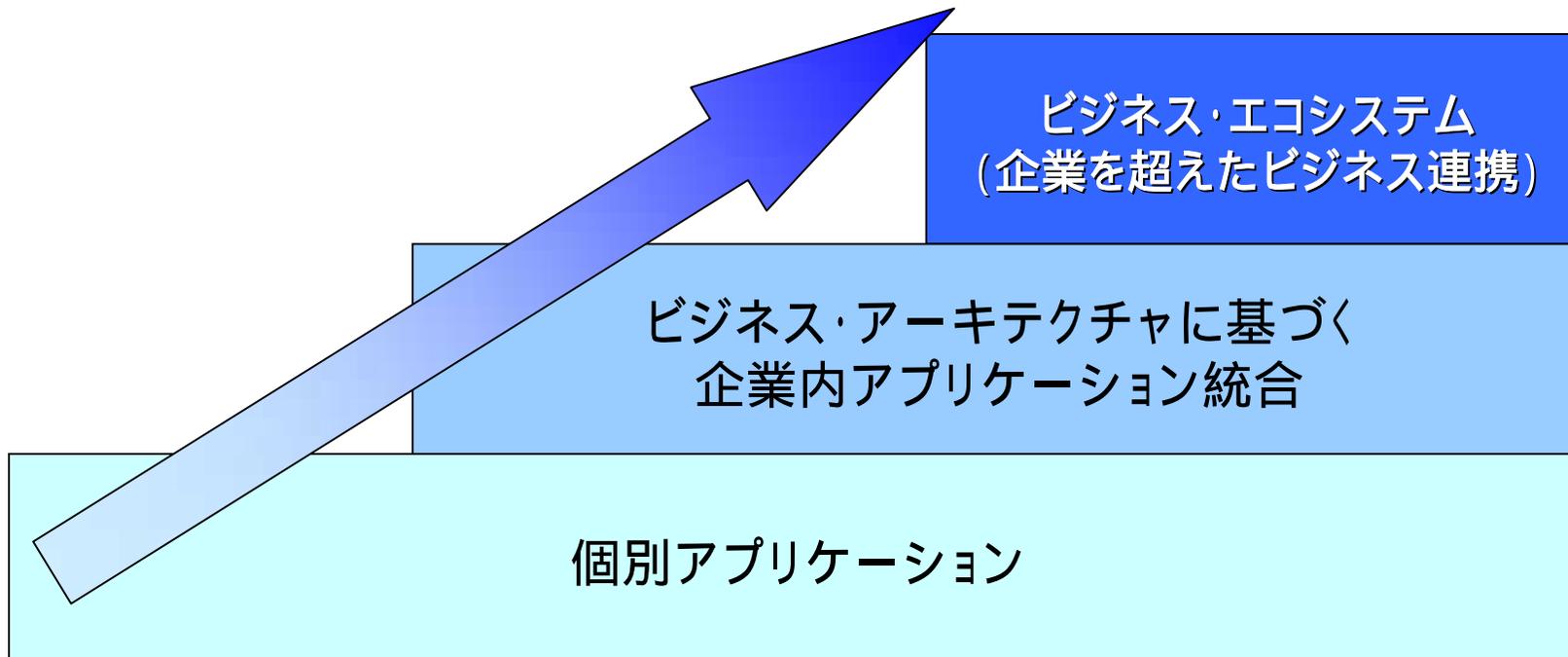
順次的なステップの実行が現実的な仮定でないことは実は認識されていた

- 「基本設計」フェーズの導入：要求の後、分析の前に入れる
- 「基本設計」以下を2度実施：1回目「問題発見に注力」、2回目「解決」とする



今後のソフトウェア開発イメージ

- 既存のソフトウェア資産からの進化
- “個別業務システムからビジネス・エコシステムへ”
 - BPM, SOA等を活用して徐々に「全体最適」を目指す方法



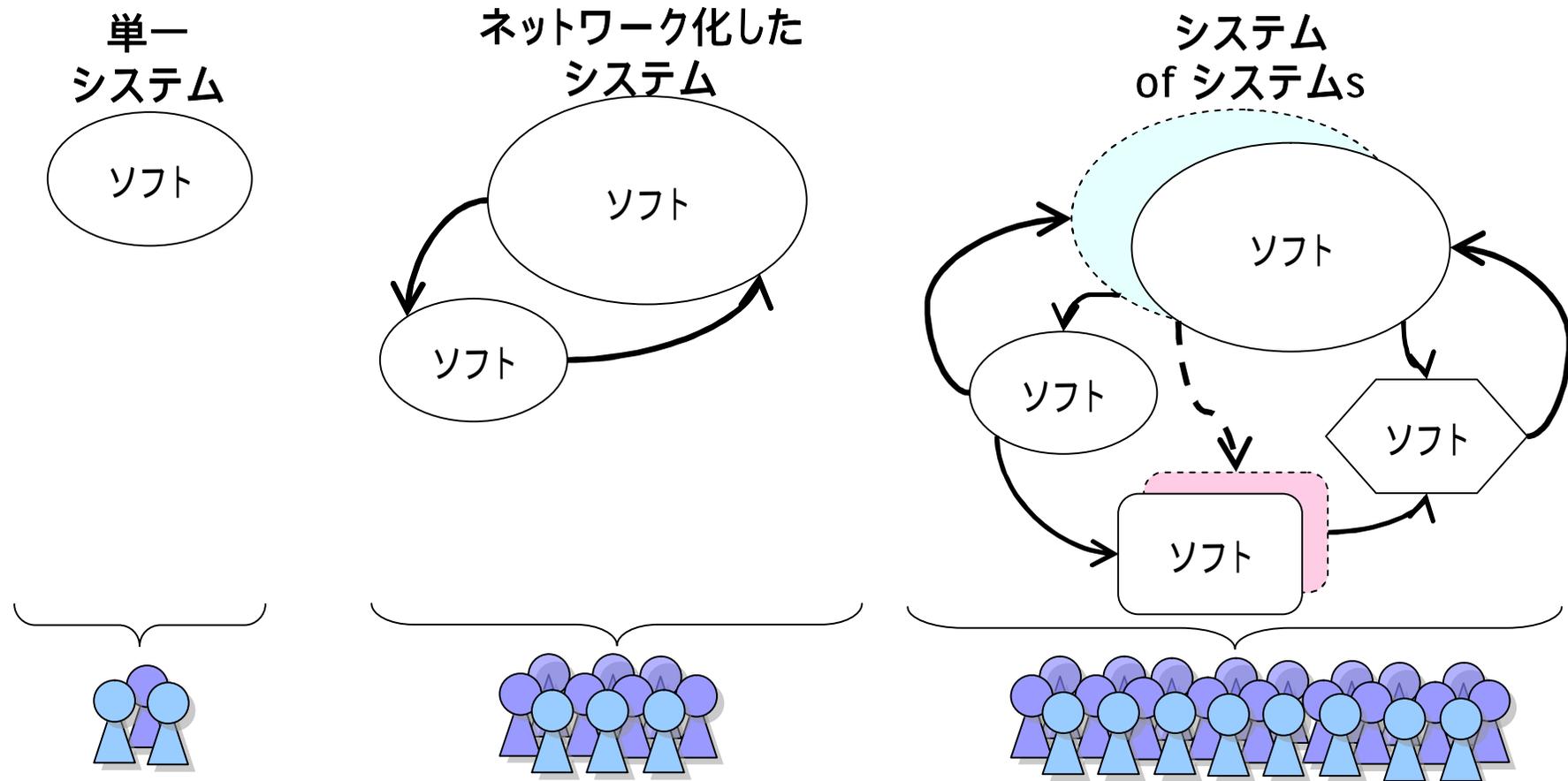
- より広範囲：企業経営にもより大きなインパクト(リスク)
- より現実へ密着：迅速、柔軟、コスト安を望まれる
- ➡ 「信頼」できるソフトウェア・システムが望まれる

- MOT (技術経営) の中心はMOS
 - ソフトウェアは経営にとって必要な技術要素
 - リスク: 東証問題、組込みシステムのトラブル...
 - リターン: IT投資が経営指標の改善に結びつく
- 現場と経営者のギャップは依然大きい
 - 現場の抱えている問題やリスクを経営者は理解できない
 - 結果としてリスクがクライシスに...
- 開発の課題
 1. 開発量の拡大
 2. 工期短縮
 3. システムの複雑化
 4. 人材不足 (プロフェッショナル)

➡ 信頼性の確保が現状のテーマ

どう対処してきたか? どう対処するか?

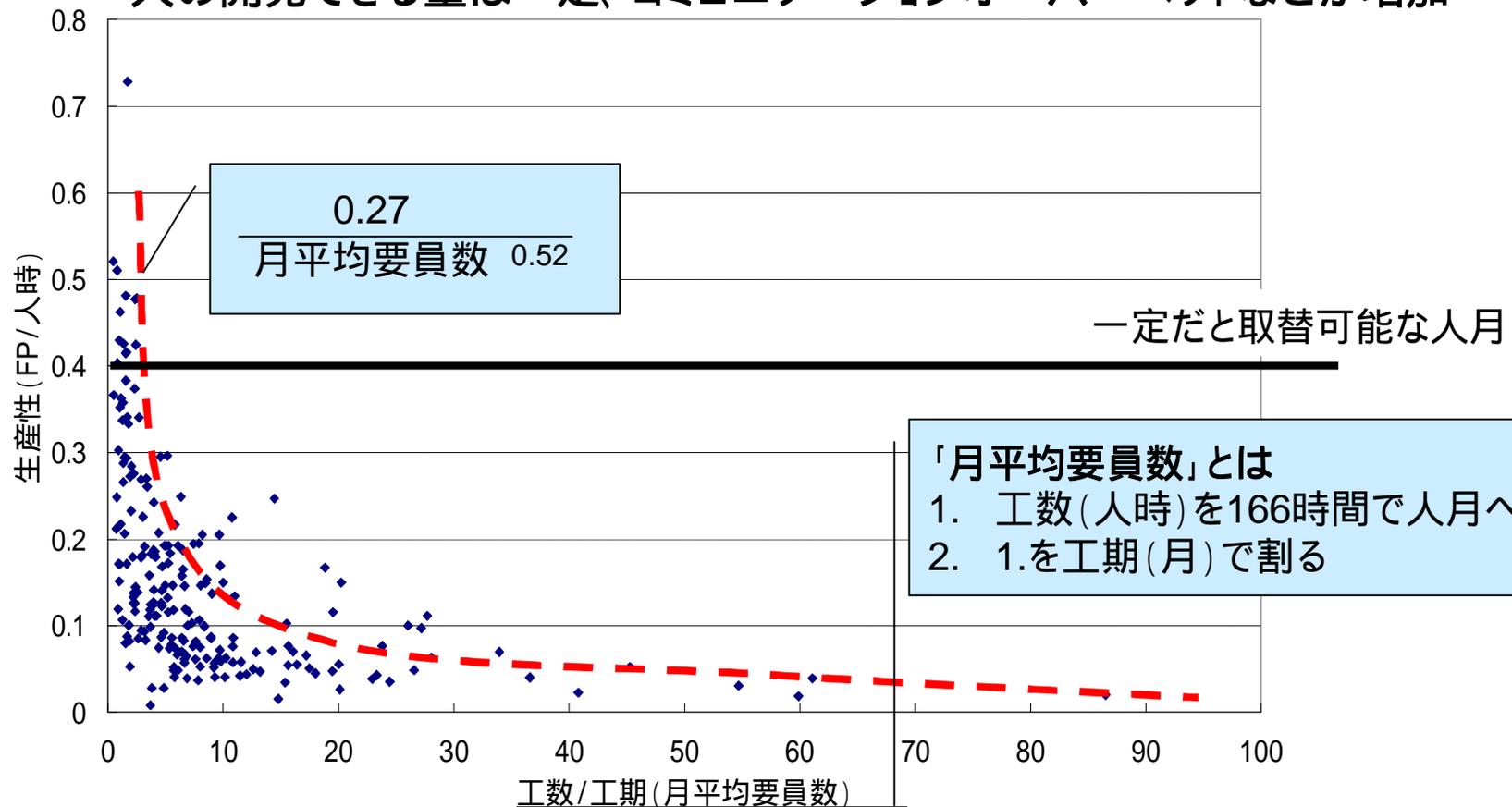
- 「少数精鋭 + 技術力」ではなく、人海戦術



- 独自の工夫 = “請負派遣!?”
 - コンプライアンス? 文字通りの請負の方向へ

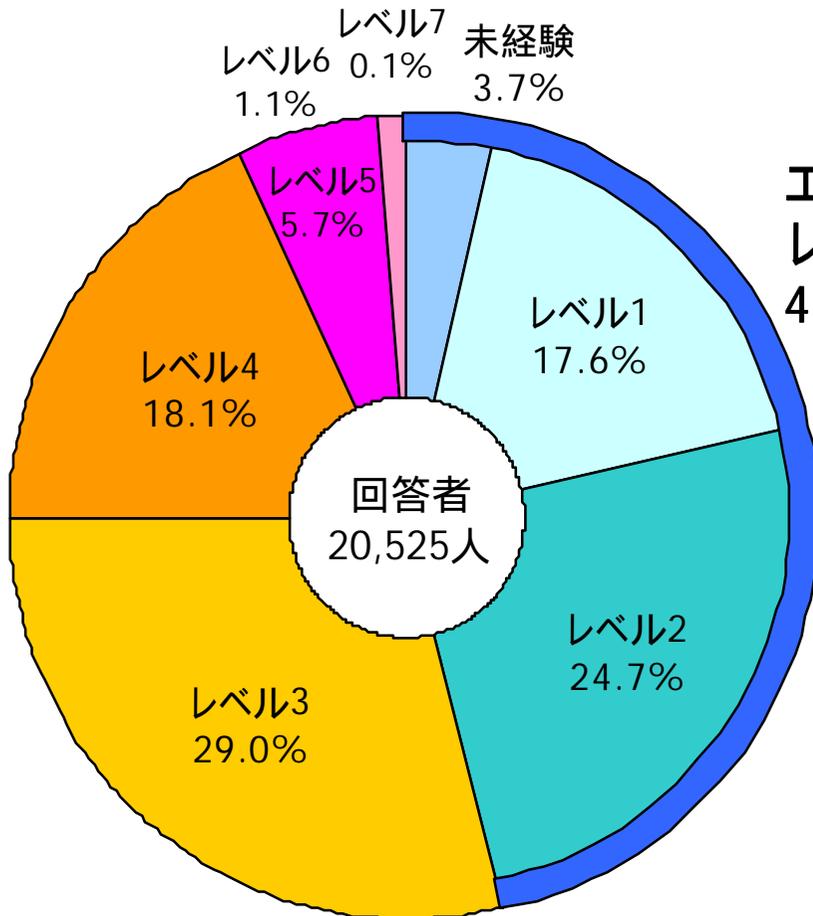
ブルックスの法則は健在

- “遅れているソフトウェア開発への要員追加はさらに開発を遅らせる”
 - 人数10倍になると生産性は 1/3
 - 20人月 = 4人 × 5ヶ月だが = 2.5ヶ月ならば、8人？
否：1.4倍は働かなければならない
人の開発できる量は一定、コミュニケーションオーバーヘッドなどが増加



ソフトウェア人材：プロフェッショナルが不足している

- 現状の戦力：全体の半数近くのスキルレベルは「エントリレベル」



ITSSのスキルレベルで見た2万人のスキル実態

出典：日経ITプロフェッショナル 2005年10月
2万人のスキル実態調査より

- 未来の戦力：
新卒社員は即戦力としては期待薄？

- 人口減 情報系学生減！？
- IT企業の人気も下降？
- 専門の学生は0.5万人/年

エントリ
レベル
46%

	情報系 (専門)	情報系 (非専門)	その他 (手段)	合計
(万人)				
学部	1.54	3.50	2.20	7.24
修士	0.43	0.65	0.10	1.18
博士	0.11	0.13	0.21	0.26
合計	2.08	4.28	2.51	8.68

出典：東京大学 大学院情報理工学系研究科
数理情報学専攻 武市 正人 教授



IT投資と経営効果について分析が進んできた

1. IT投資は経営にプラス効果
2. 組織特性(分権化の程度、熟練労働の利用、知識共有など)はIT投資と相関
3. IT投資はITの直接使用に関する投資だけでなく多くのインタンジブルな組織投資が必要
4. IT投資はインタンジブルな組織投資と組み合わせさせて、より大きな経営効果を生む

ソフトウェア・エンジニアリングと経営効果の関係は?

- ソフトウェア・エンジニアリングへの投資は経営にプラスか?
- 環境、組織特性などの要因とソフトウェア・エンジニアリングや経営は関係あるか?
- どのように関係があるか?

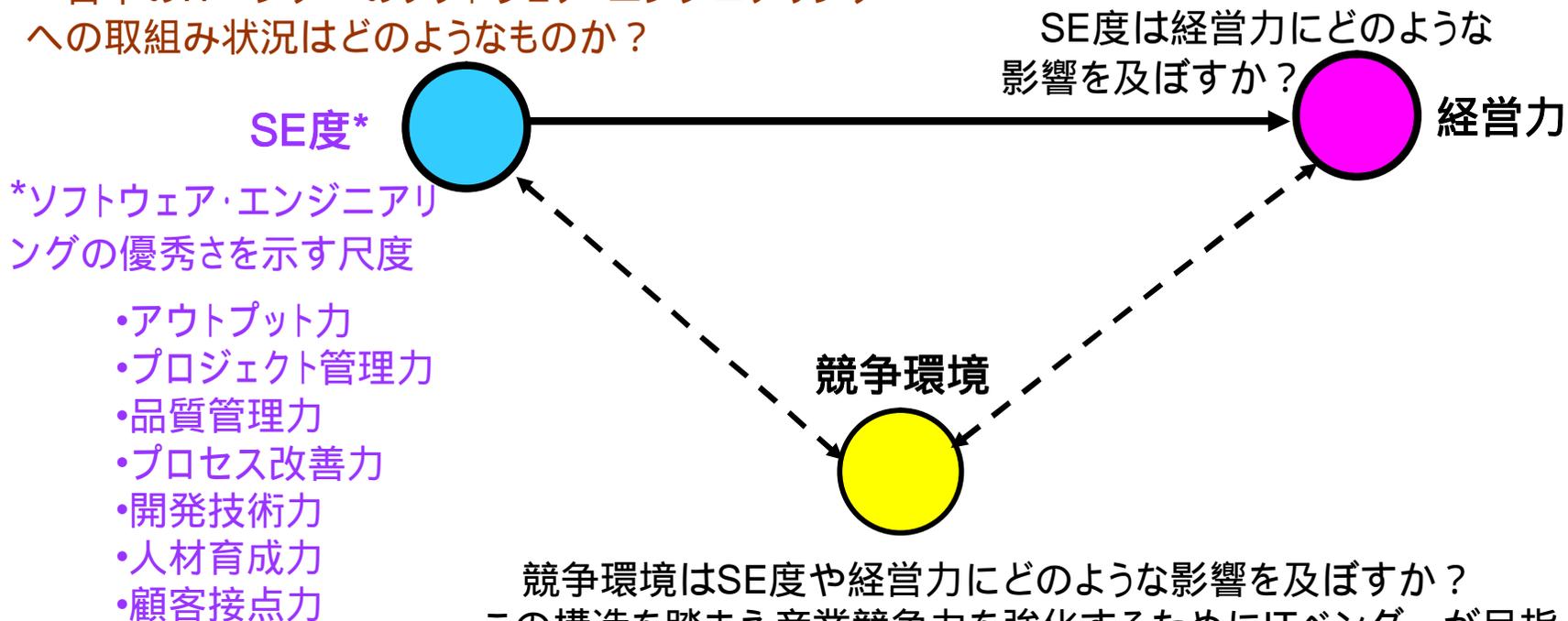
「SE度調査」

調査概要

1. 調査対象：日本の大手ITベンダー企業(237社)
2. 有効回答：55社(メーカー系17社、ユーザー系15社、独立系23社)
3. 調査規模：特サビ調査 SE/PG数 = 347,005人(H16)
SE度調査2005 55社推定SE/PG数 = 109,544人 32%

SE度の基本仮説

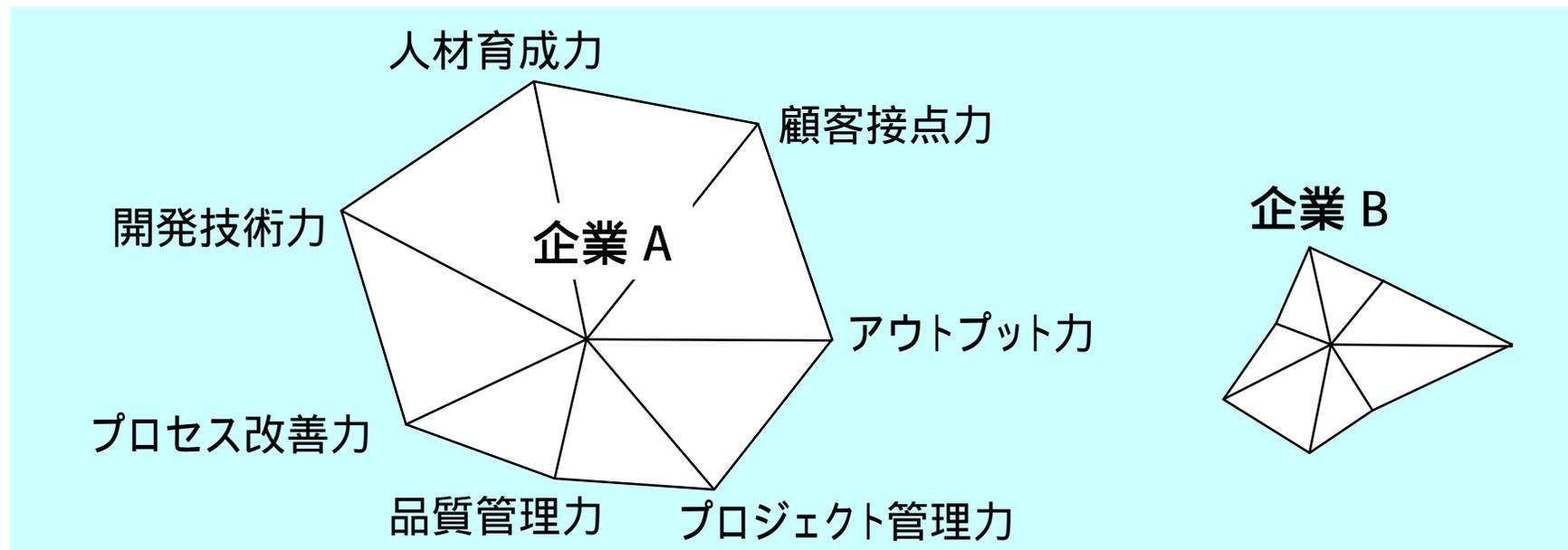
日本のITベンダーのソフトウェア・エンジニアリングへの取組み状況はどのようなものか？



競争環境はSE度や経営力にどのような影響を及ぼすか？
この構造を踏まえ産業競争力を強化するためにITベンダーが目指すべき方向および産学官の役割は何か？

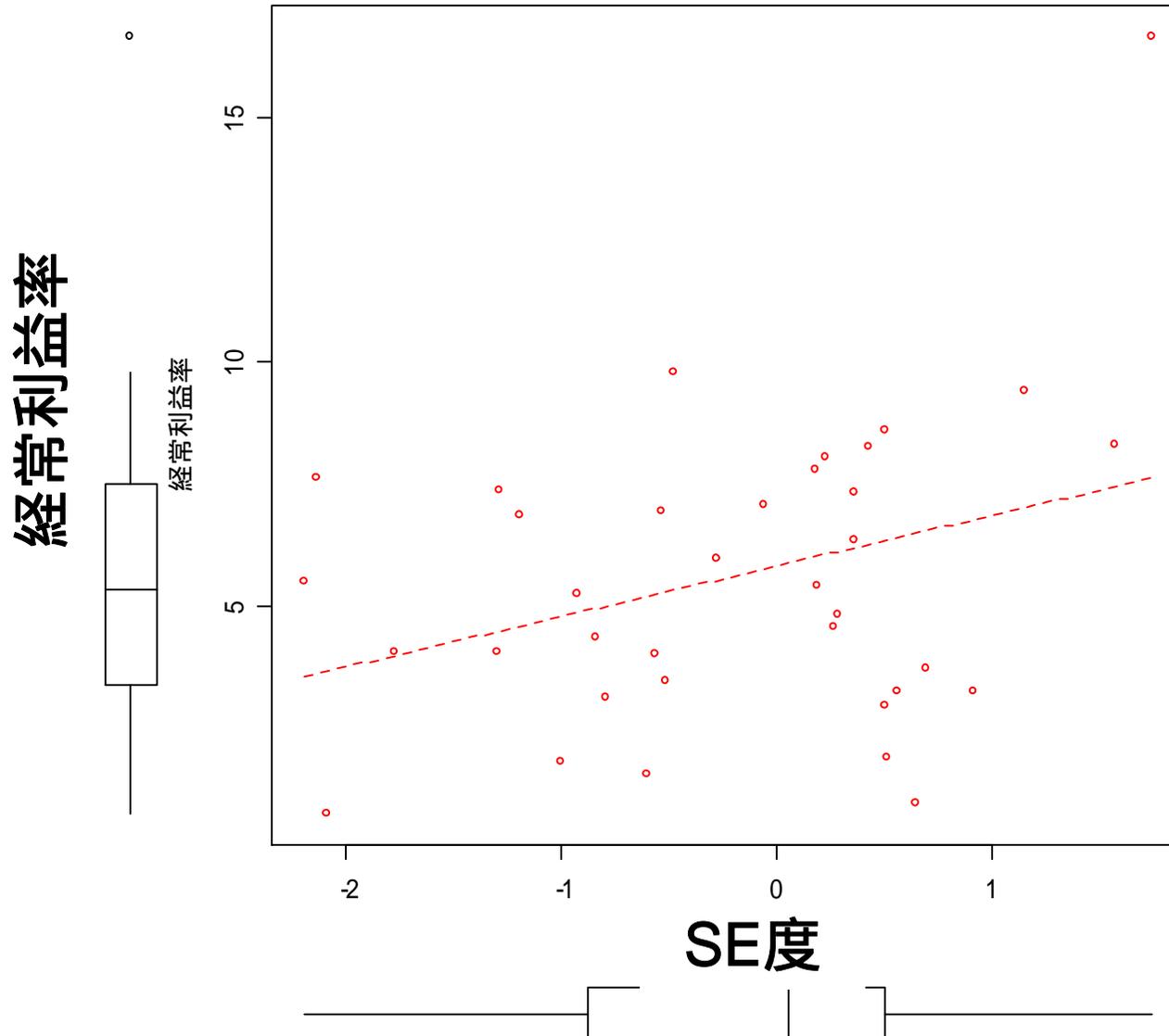
各企業のソフトウェア・エンジニアリングへの取組状況

- SE度の分析
 - SE度を7つの因子でコンセプトを定義
(20名+の有識者への事前インタビューなどをベース)
 - 有効回答55社を企業ごとに対比し特性を数値化
 - 標準偏差による得点(相対的な強さ)
 - 因子ごとの特徴把握(強み・弱み)



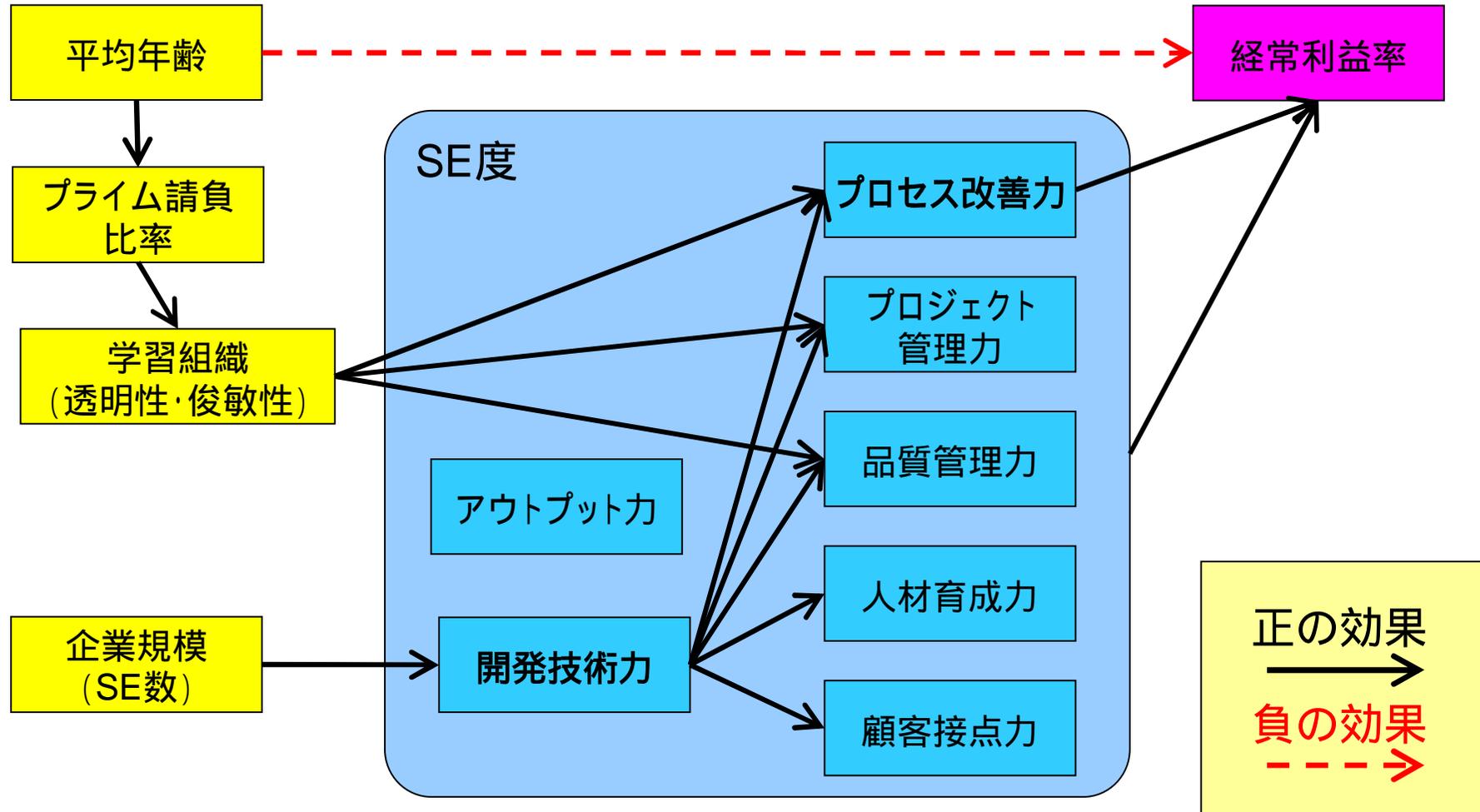
例：企業ごとのSE度をレーダーチャート化したもの

SE度は経常利益率に貢献



環境はどのようにSE度や経営に影響を及ぼすか？

- ソフトウェア・エンジニアリングの実践力(= SE度)の改善は、経営力を高める
- SE度と経営力の関係には補完する要素(組織風土、規模、顧客基盤など)がある



“補完要素、SE度、経常利益率の因果関係”の例

産業競争力強化に向けて

1. 産学官の連携

- 産業界、学界の間のギャップにおける価値創造
- 仲介者としてのSEC

2. 見える化

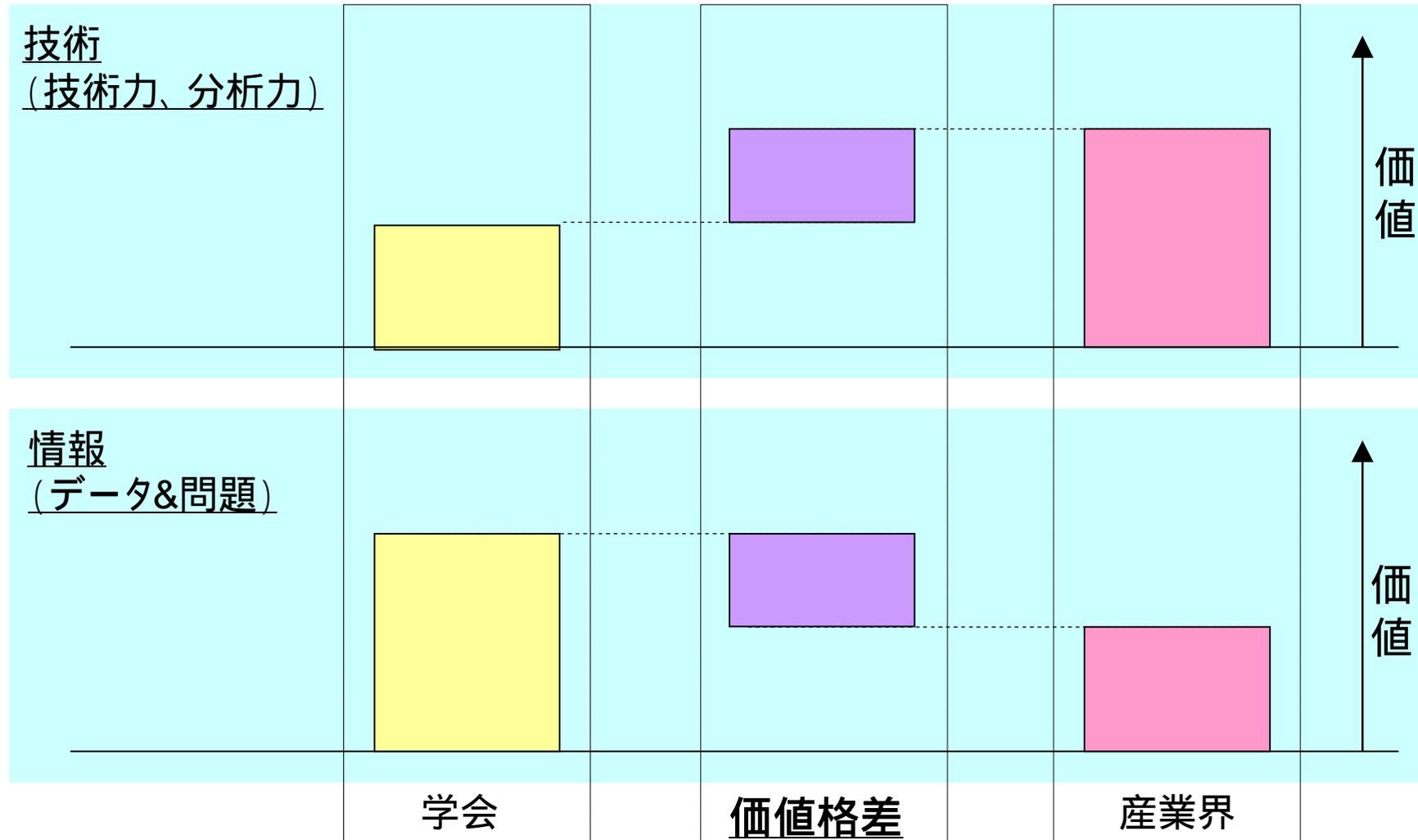
- リスク・マネジメント
 - リスクはなくなる、プロジェクトは常にリスクを抱える
- 「組織」としての能力へ

3. 組織能力の向上

- 柔軟な組織、学習する組織へ
 - トップの意識改革
 - 中間管理職の再教育

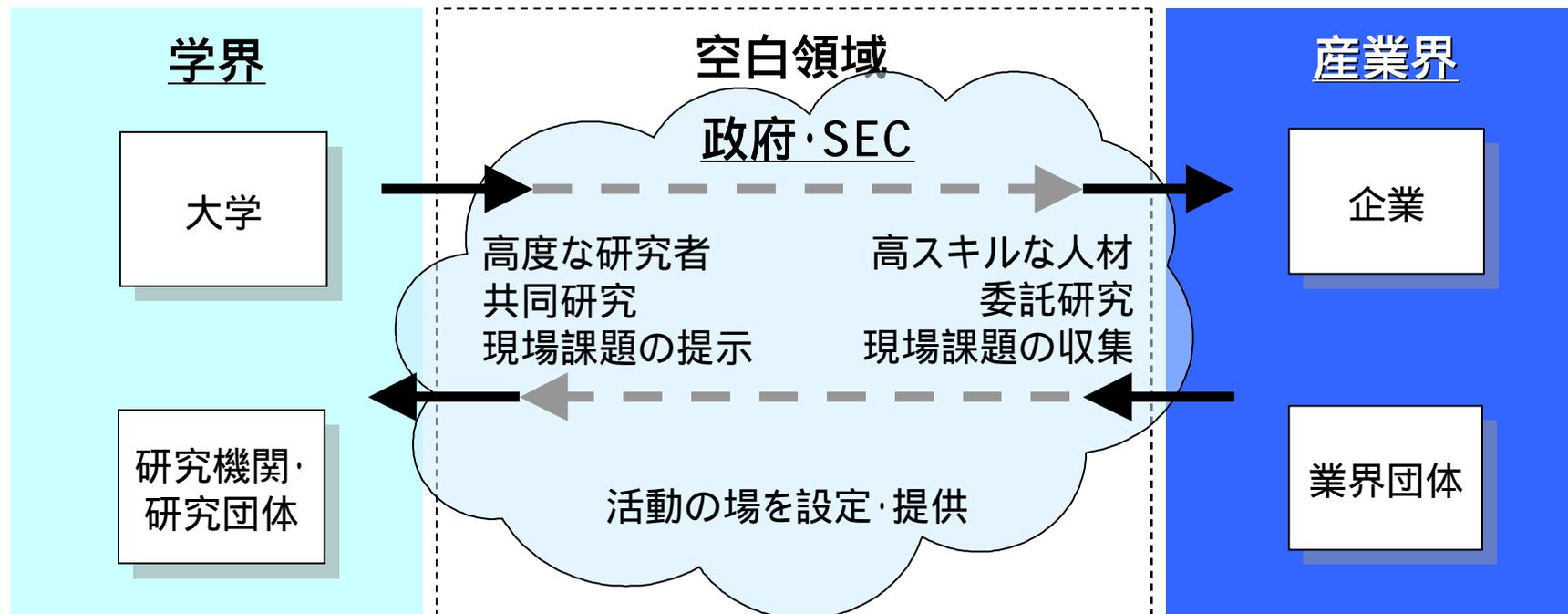
産学官連携における各自の役割

- 産学連携を仲介する価値：「格差」を利用した鞘取り
 - 格差となる価値を提供するべく、それぞれの差を認識し結ぶ



産学官の連携の“要”としてのSEC

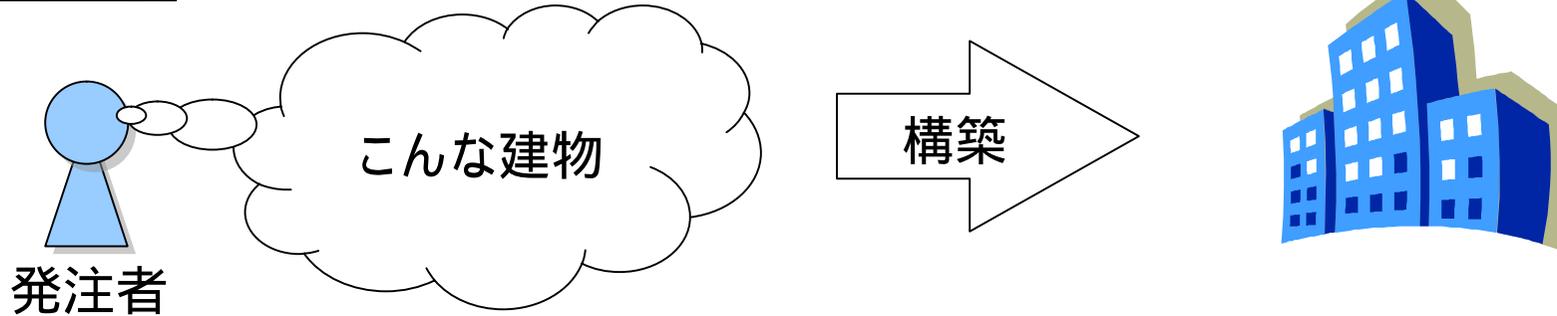
- 産学連携の“触媒”となり、連携を補助する
 - 現状：企業の研究は減り、大学のR&Dは現場課題とマッチングに課題
 - 将来：価値創造のポジティブ・サイクルが回る仕組み



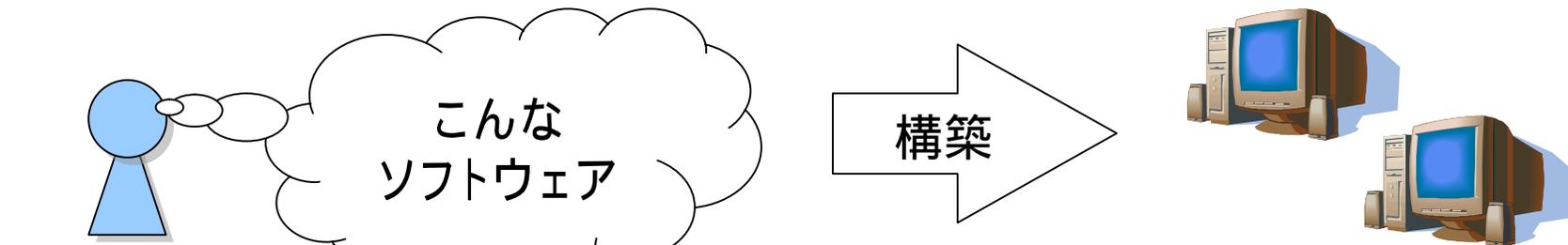
ソフトウェアは「見えない」?

- ソフトウェアは本当は「よく見える」のでは....。

建築の例



ソフトウェア開発の例

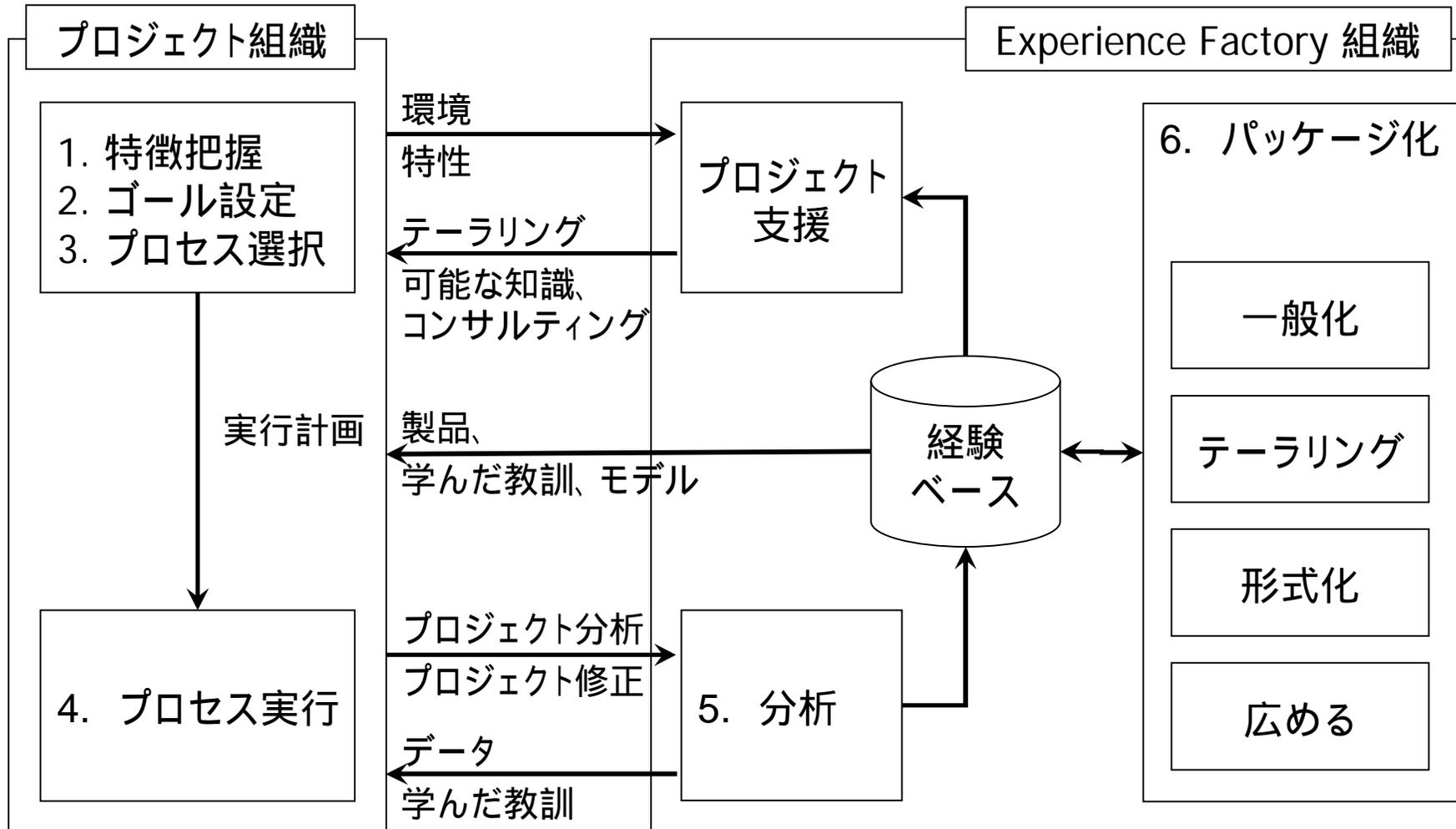


ここはどちらも実は見えない

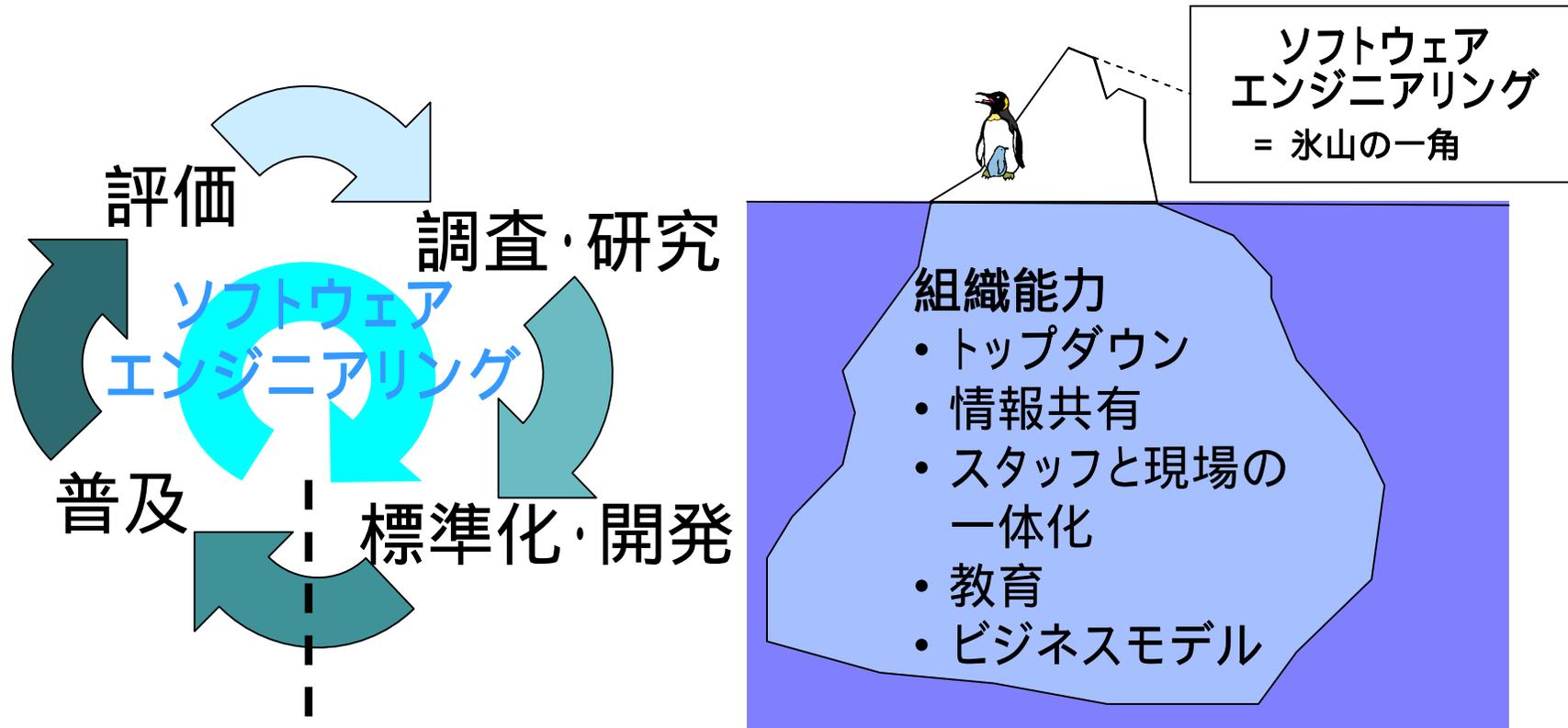
ここは建物は壊したりしないと確認不能
ソフトウェアは全部見える

ソフトウェア開発で「見える」ものをどう活かす？

- 「見える化」が組織知へのカギ



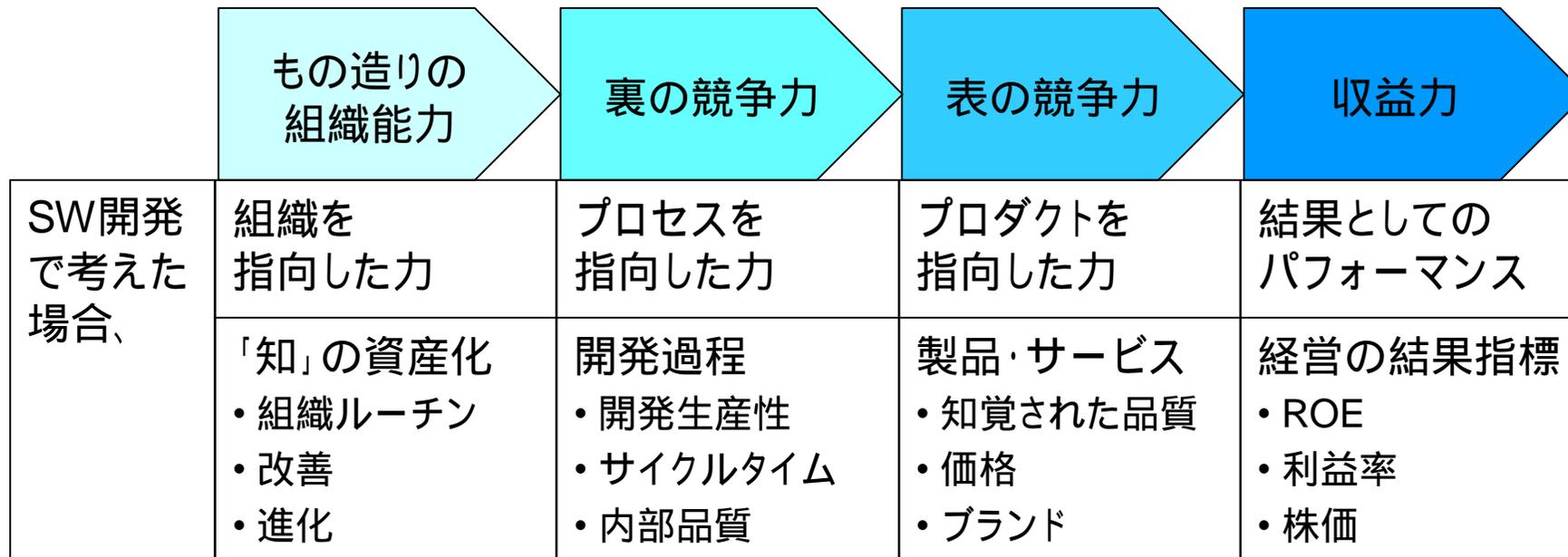
ソフトウェア・エンジニアリングを 企業収益向上に結びつけるために



- 経営層からのトップダウンでの取組みが不可欠
- 本社スタッフと現場が一丸となって取組むことが必要

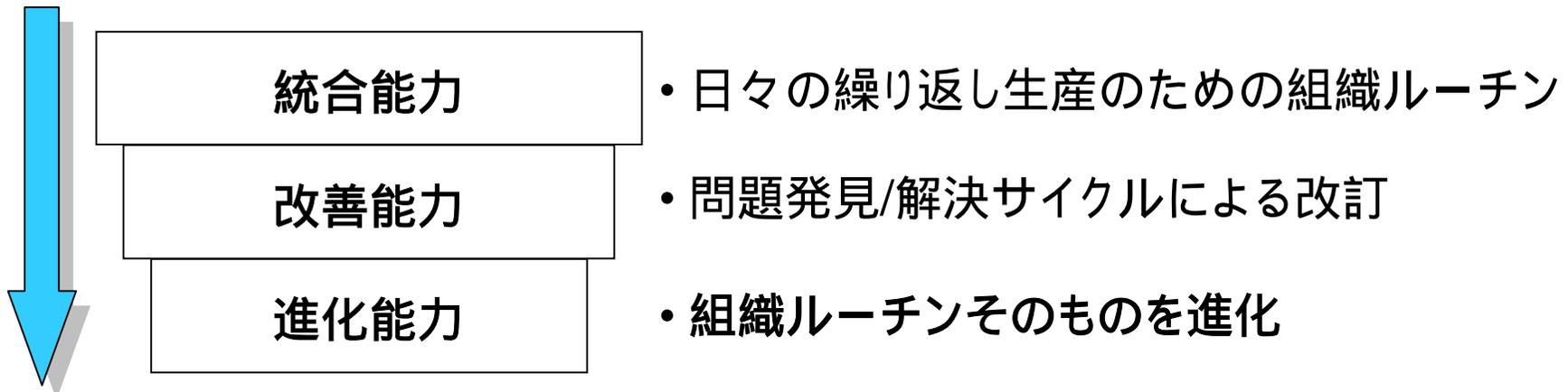
ソフトウェア開発「組織」としての競争力

競争力の多層構造



- モノ (= プロダクト) を作るだけでなく、
- 開発過程 (= プロセス) だけでもなく、
- それらの関連付け、知の資産 (= もの造りの組織能力) とし、最終的に収益力へ結びつく「競争力」を作っていくことが必要

組織能力の「深さ」



もの造りのための組織能力:

“組織ルーチンをきちんとまわす”
一度つかんだら離さない、組織としてのメモリー保持メカニズム

と同時に

“優れたやり方を取り入れ進化させる”
不具合・改善点は、どんどんルーチンに取り入れメモリーする



組織

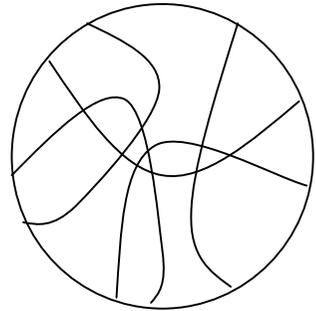
1. 個々の“心構え”
2. 全体としての**実力差**

企業の大小と技術移転

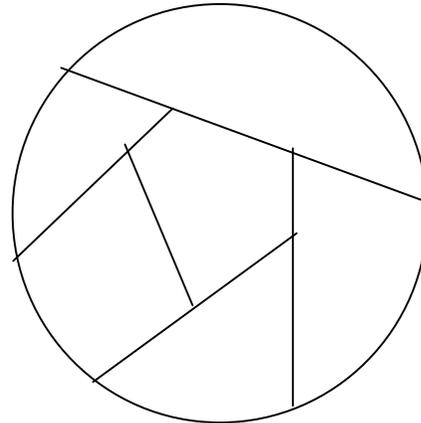
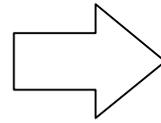
規模：小



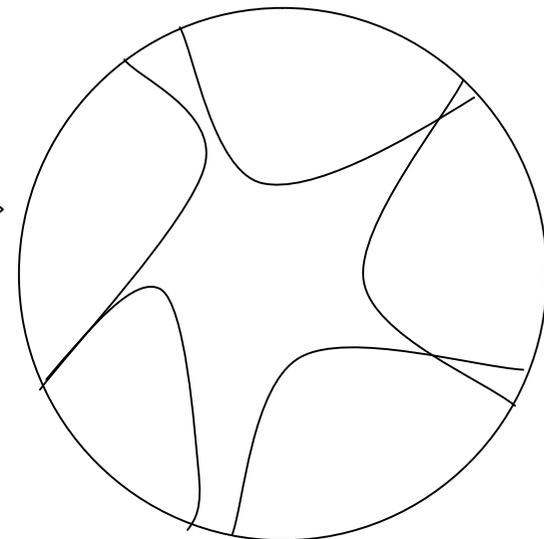
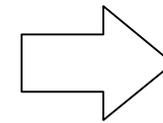
規模：大



要素機能が
オーバーラップ



狙い(建前)
要素機能を
分担 効率化



成熟し、要素機能の
隙間がある(個別最適)

アジャイル的な取組み

- 情報化投資はまだまだ
- 千差万別
- 人間力、統合力

ベストプラクティスや
モデルベースの取組み

- 情報化投資は一巡
- 専門分化、技術力(R&D)
- 制度、仕組み、資格

ご清聴ありがとうございました

独立行政法人 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

所長 鶴保 征城

2006年6月9日

<http://sec.ipa.go.jp/>