

---

# ハードのものづくりの改革に学ぶ ソフトの生産革新

---

平成17年10月14日

NECシステムテクノロジー

宮下洋一

---

# 目次

---

1. SPI活動の成果についての課題
2. SPIからみたHW生産革新の成功のポイント
3. 改善事例に見るHW生産革新の考え方の具現化
4. CMM的SPI活動とHW生産革新の統合フレーム

---

# 1. SPI活動の成果についての課題

---

---

# これまでの活動経緯をふりかえると

---

- 「プロセス成熟度モデル」に取り組み始めて15年
- ISO9001に取り組み始めて10年
- JASPIC加入を契機に本格的に推進して5年
- いくつかの組織でレベル5達成
- 多くの組織でレベル3達成
- もっと多くの組織で愛踏み状態

当初期待していた成果が  
本当に達成できているか？

推進のキーワード : 「CMM推進を梃子にしたプロセス改善」

---

# しかし、こんなつづきも

---

- 高レベルを達成してもあまり変わらない
- 良くなったことが実感できない
- 目に見える成果が出ていない
- 一部の部隊、PJのみが活動している
- 目標レベルを達成したからもう良いだろう

手段(“梃子”のはずだったが)が目的化した？

何か足りなかったのか？

何か間違ったのか？

ある程度こうなることは必然なのか？

---

## もう少し具体的に

---

- データを収集するのが大変。さらに収集したデータを活用するのはもっと大変。
- 標準プロセスを定義したいが、全PJに通用するものを作成するのは大変。
- 計画重視の考え方はもっともなことだろうけれども、せっかく詳細に作った計画も、要求される仕様変更やスケジュール変更への対応でガタガタになる。
- 改善の効果を知りたいと思うが、他の大きな変動要因（スキルレベル、技術の新規性、もともと無理な計画）によるぶれの中に吸収されてしまう。
- 認証取得やアセスメント対応に特別なものを用意する

**我々の事業環境に最適なモデルだったのだろうか？**

# CMMに対する評価

## CMM推進による効果は大と言われている

- レベル1上がると10%生産性があがる
- ROIが5:1
- 品質コストが激減 … などの事例の発表

このGapは何から生まれる？

## CMMを推進してもそれほどの効果は上がらない

- 高レベル達成しても品質・生産性は良くならない
- レベル5の組織の品質は思ったほど高くない
- アセスメントではわかっていることしか指摘されない
- 高レベル達成にはよけいなコストがかかる

---

## 2. SPIから見たHW生産革新の 成功のポイント

---

何か違うのだろうか？

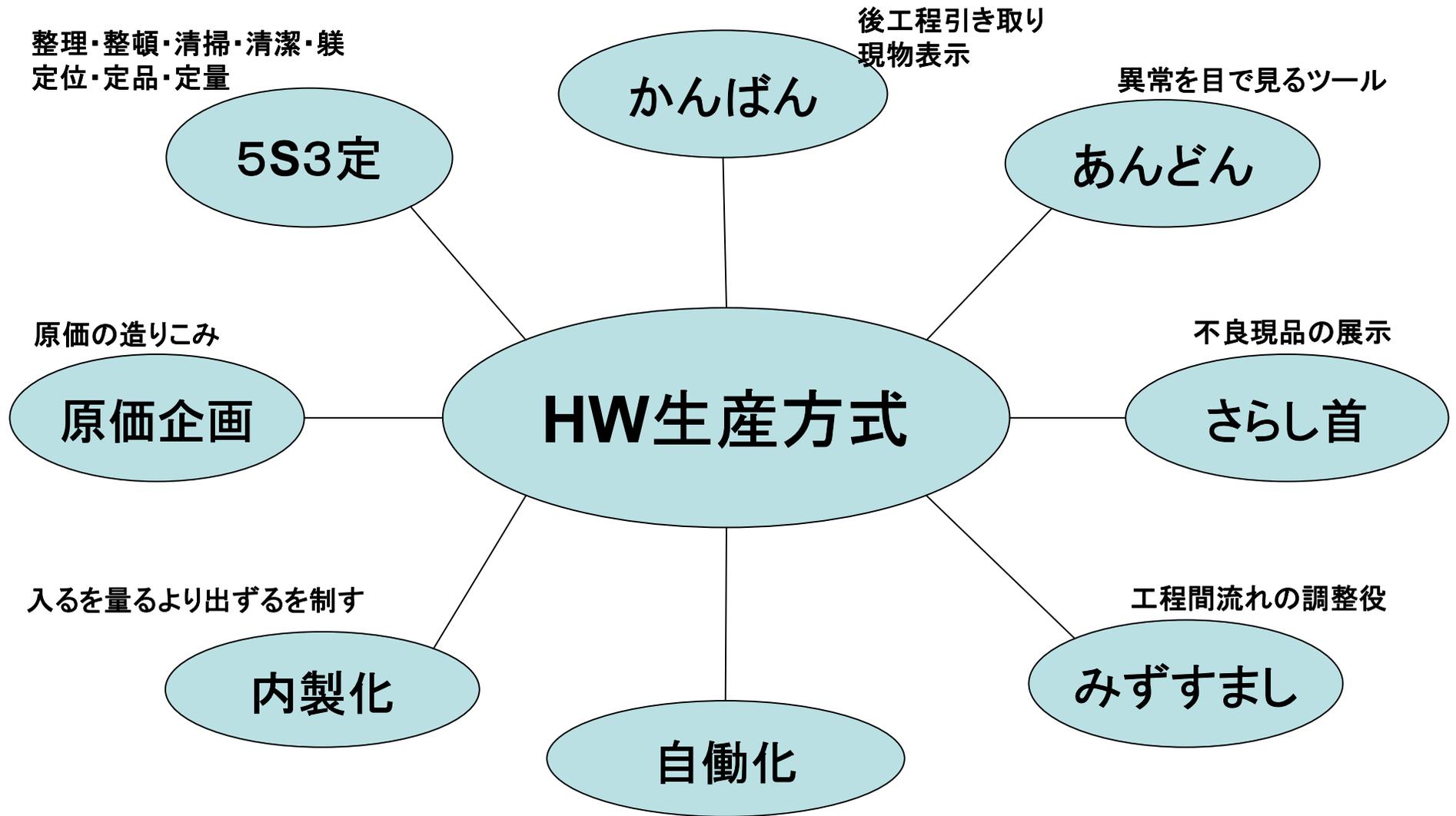
方法の違い？

分野の違い？

適性の違い？

隣の芝生は青いだけ？

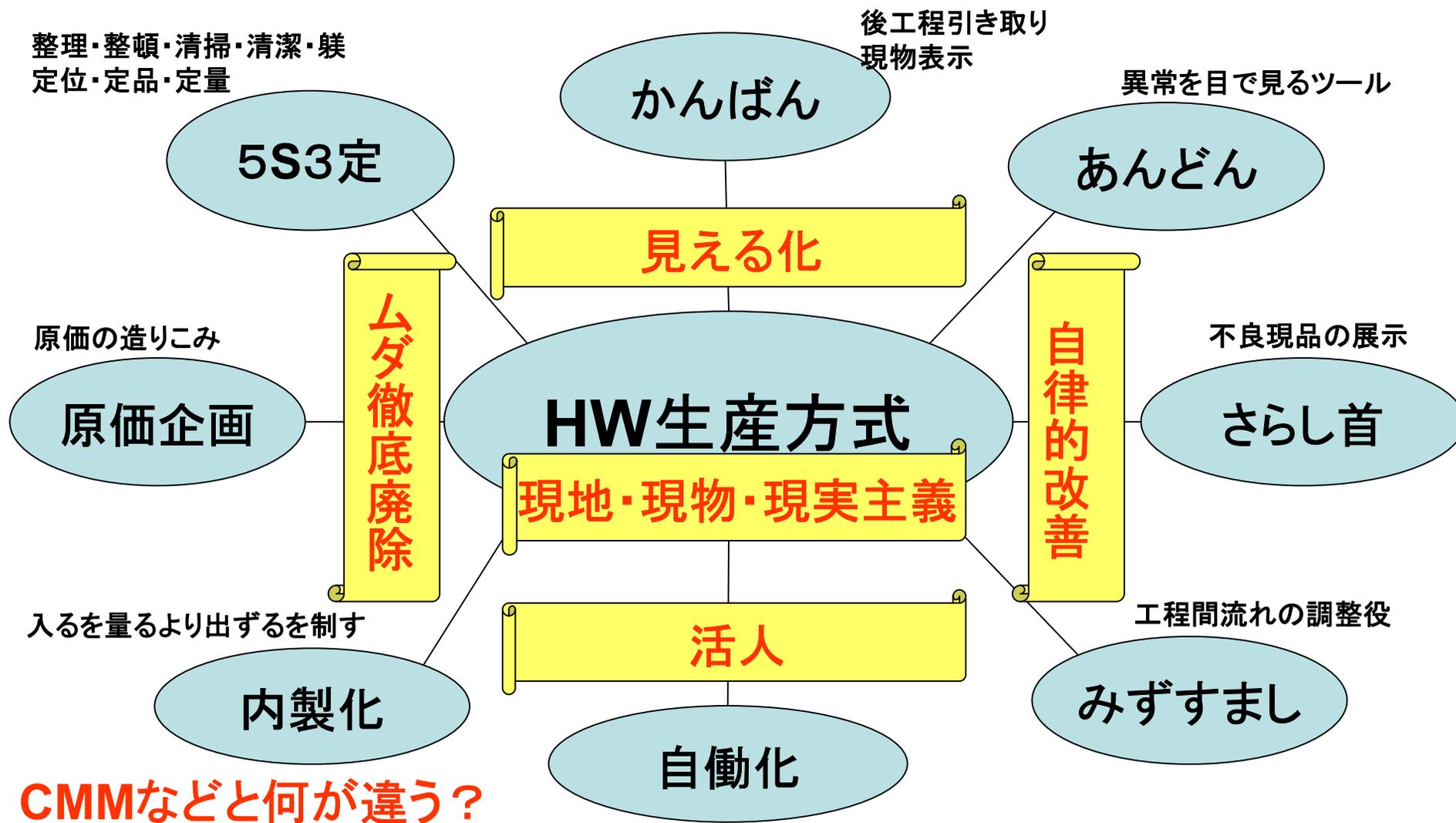
# HW生産方式のキーワード



これらから何を学ぶべきか？

異常を検知して止め、後工程に不良を伝えない

# HW生産革新の成功要因



CMMなどと何が違う？  
何が同じ？

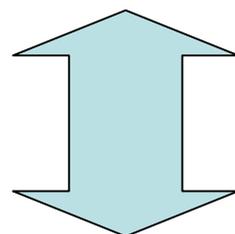
異常を検知して止め、後工程に不良を伝えない

# 徹底的なムダ廃除

## HW生産革新成功のキーワード

作業 = 「働き」 + 「動き」

「働き」は付加価値を生むもの、「動き」はムダ



## SPIからみた考察

- ・作業を「働き」と「動き」に分ける考え方はわかりやすい軸
- ・ムダな作業に加えて、ムダな成果物の洗い出しも重要
- ・ムダ廃除に焦点を当てた活動は成果も出しやすい
- ・ムダ廃除というキーワードはSPIの中ではあまり使われていない  
足りないものを追加するという方向に目が向いている

# ソフト開発での「働き」と「動き」の例

	「働き」 (付加価値を生む作業と成果物)	「動き」 (ムダな作業と成果物)
要件定義	<ul style="list-style-type: none"> <li>顧客との直接的な要件確認</li> </ul>	<ul style="list-style-type: none"> <li>要件確認ミスによる再調整(後戻り)</li> <li>要件の理解や確認のための情報収集</li> <li>顧客による要件確定までの待ち時間</li> </ul>
設計	<ul style="list-style-type: none"> <li>お客への納入物である設計書の作成</li> <li>品質が確認されている設計の再利用</li> </ul>	<ul style="list-style-type: none"> <li>設計ミスによるやり直し(後戻り)</li> <li>同じ機能の重複した設計</li> <li>お客に納入しないドキュメントの作成</li> <li>協力会社への説明、教育</li> <li>上位設計完了までの待ち時間</li> </ul>
製造	<ul style="list-style-type: none"> <li>お客への納入物であるプログラム作成</li> </ul>	<ul style="list-style-type: none"> <li>プログラミングミスによるやり直し(後戻り)</li> <li>同じ機能の重複したプログラミング</li> <li>仕様変更によるプログラミングやり直し(後戻り)</li> </ul>
テスト	<ul style="list-style-type: none"> <li>品質保証</li> </ul>	<ul style="list-style-type: none"> <li>協力会社からの受け入れ確認テスト</li> <li>システムバグ修正に伴う再テスト</li> <li>バージョンアップに伴うリグレッションテスト</li> </ul>
PJ管理	<ul style="list-style-type: none"> <li>PJ計画承認</li> <li>PJ課題解決</li> </ul>	<ul style="list-style-type: none"> <li>PJ進捗データの収集</li> <li>PJ進捗会議による状況報告</li> </ul>

---

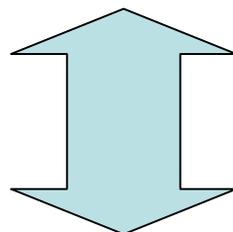
(生産革新成功要因) **現場・現物・現実主義**

---

## HW生産革新成功のキーワード

蓄積された膨大なデータに依存するのではなく、  
発生時に現場、現物で事実を把握すると真因がわかり処置できる

注：報告時では情報が散逸し、  
改善のタイミングを逃す



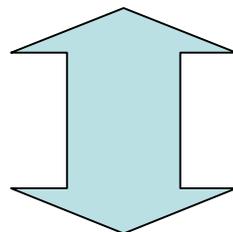
## SPIからみた考察

- データを収集しても役にたっていない実情へのメッセージ
- HW生産ラインでも、「標準化」が進んでいる(業務標準)から可能。  
かんばんを使ってもジャストインタイムの機構が機能しなければ、  
滞ったかんばんの数が異常かどうかは即座にはわからない

# 自律的改善

## HW生産革新成功のキーワード

計画主導型からの転換＝標準化、システム化、自律化  
作業の繰り返し性を作り、仕事を標準化してシステム化。  
異常のみ管理すれば良くなるので、管理工数は少なくて済む。  
ほとんどの場合はかんぱんやみずすましで自律的に流れる



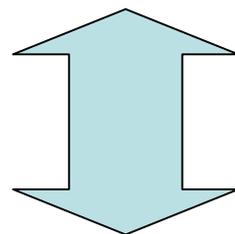
## SPIからみた考察

- ・計画重視といいながら要件変更への対応に四苦八苦する状況に対する一つメッセージ
- ・計画で押さえるべき点と現場で自助的調整解決していく点の見極め
- ・計画重視の考えと自律化の両面が必要。  
その重みはPJの特性によって変わる

# 見える化

## HW生産革新成功のキーワード

「見える化」による問題の顕在化→だれでもが共有化→即改善  
かんばん、あんどん、さらし首・・・など現物と一体となった状況の見える化



## SPIからみた考察

- 「見える化」は「見せる化」ではない
- 誰が見ても問題が明らかになるような仕組みが狙い
- PJ進捗管理票やPJ状況報告書などは「見せる化」の手段。  
見せるために、報告、データ集計などのムダな作業が必要になる。
- 「見える化」により現場での速やかで的確な判断ができる

---

### 3. 改善事例に見るHW生産革新の 考え方の具現化

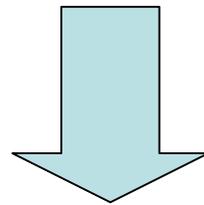
---

---

# HW生産革新的活動の組み込み

---

- HW生産革新の成功要因は、SPIでつまずいている課題とオーバーラップしている
- SPIを成功させるためのヒントがある
- HWの方法のそのままの適用は無理であるが「改善」を志向する同様な試みをすることは有望



**重点PJを設定し個別に改善に挑戦**

# 改善のねらい目

	「働き」 (付加価値を生む作業と成果物)	「動き」 (ムダな作業と成果物)
要件定義	<ul style="list-style-type: none"> <li>顧客との直接的な要件確認</li> </ul>	<ul style="list-style-type: none"> <li>要件確認ミスによる再調整(後戻り)</li> <li>要件の理解不足による情報収集</li> <li>顧客への説明不足</li> </ul>
設計	<ul style="list-style-type: none"> <li>お客への納入物である設計書の作成</li> <li>品質が確認されている設計の再利用</li> </ul>	<ul style="list-style-type: none"> <li>設計書の作成</li> <li>設計書の確認</li> <li>設計書の修正</li> </ul>
製造	<ul style="list-style-type: none"> <li>お客への納入物であるプログラム作成</li> </ul>	<ul style="list-style-type: none"> <li>プログラムの作成</li> <li>プログラムの確認</li> <li>プログラムの修正</li> </ul>
テスト	<ul style="list-style-type: none"> <li>品質保証</li> </ul>	<ul style="list-style-type: none"> <li>プログラムのテスト</li> <li>プログラムの確認</li> <li>プログラムの修正</li> </ul>
PJ管理	<ul style="list-style-type: none"> <li>PJ計画承認</li> <li>PJ課題解決</li> </ul>	<ul style="list-style-type: none"> <li>PJ進捗確認</li> <li>PJ進捗会議による状況報告</li> </ul>

この部分を  
最小化する  
(ムダなものにする)

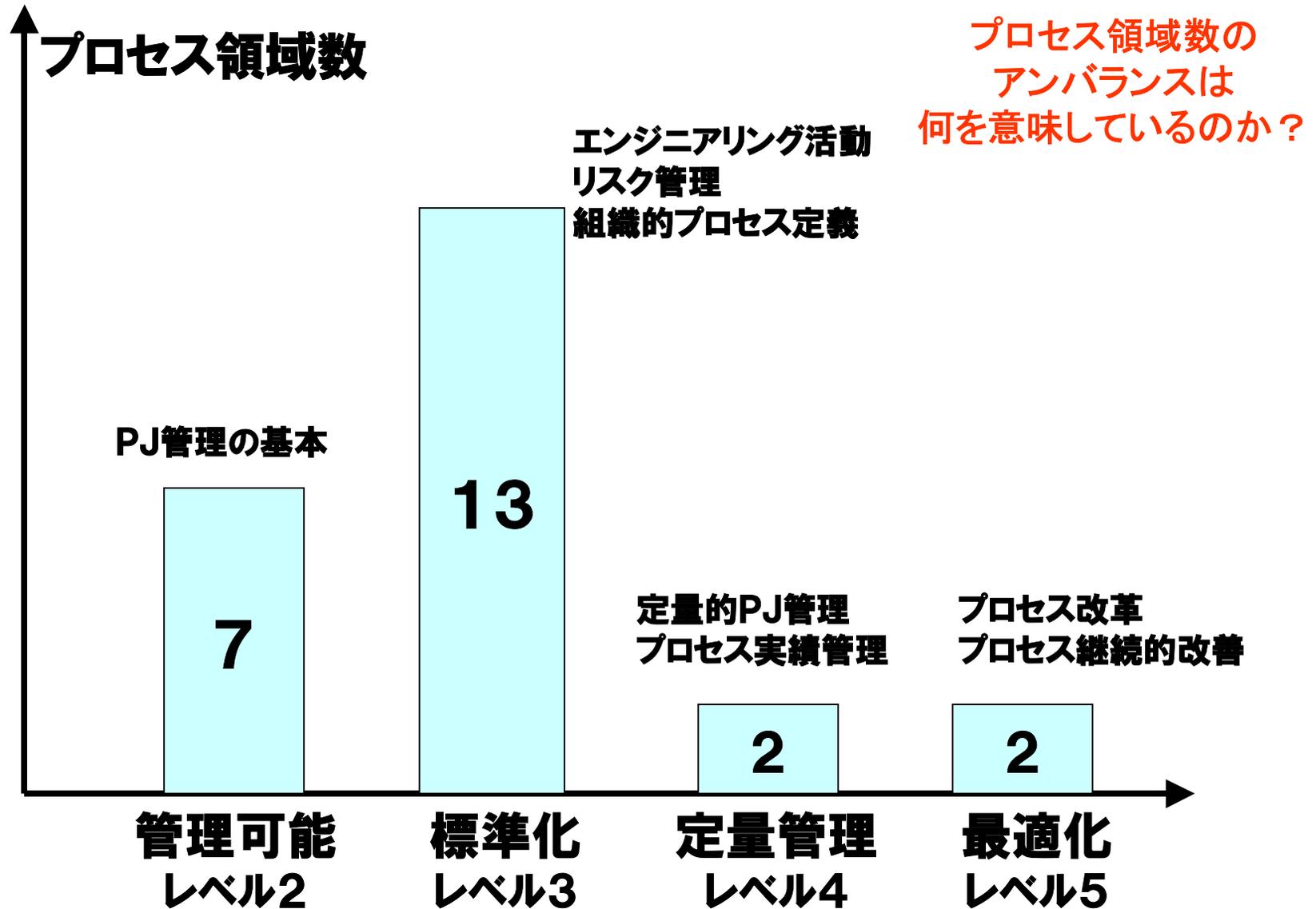


---

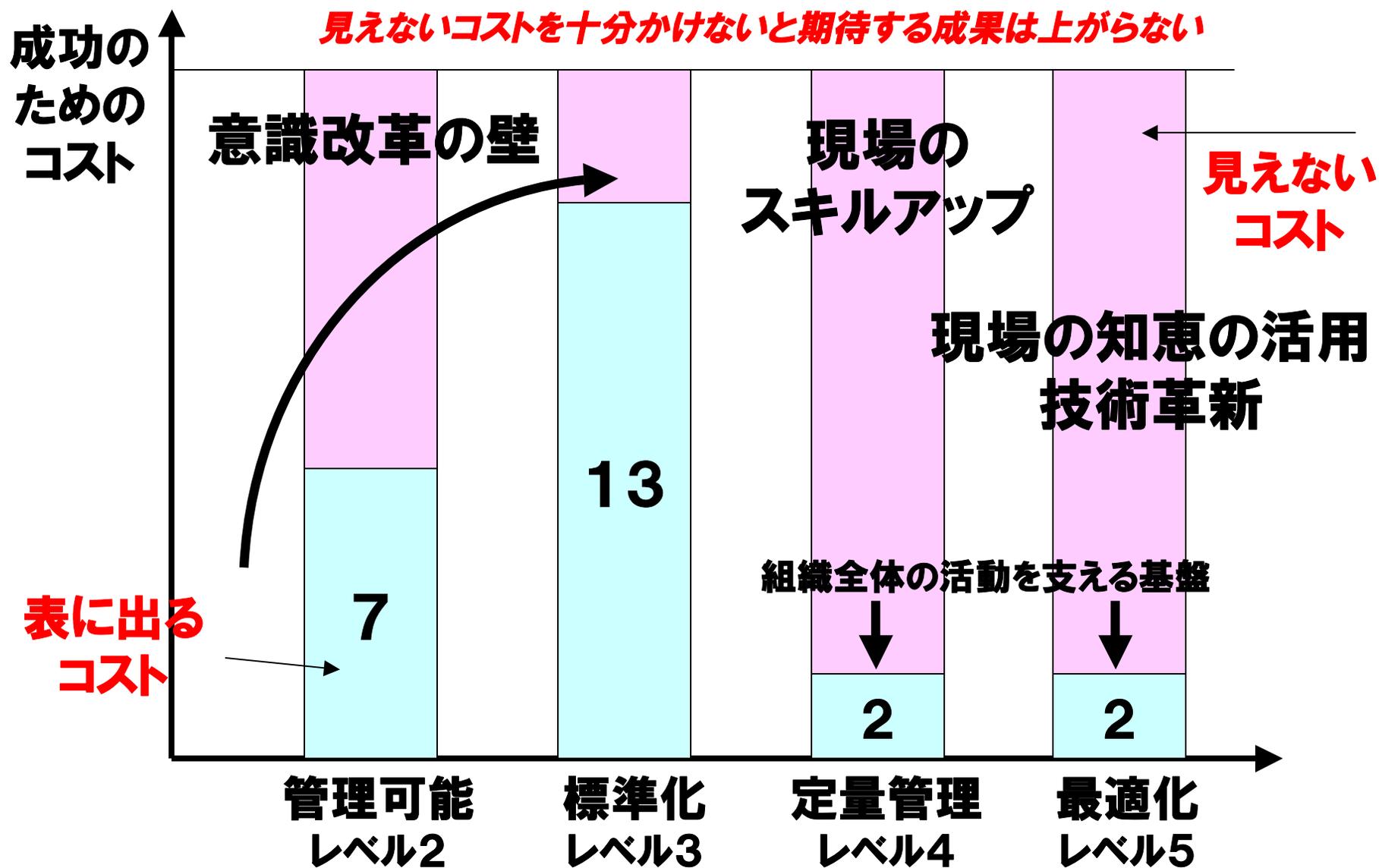
## 4. CMM的SPI活動とHW生産革新の 統合フレーム

---

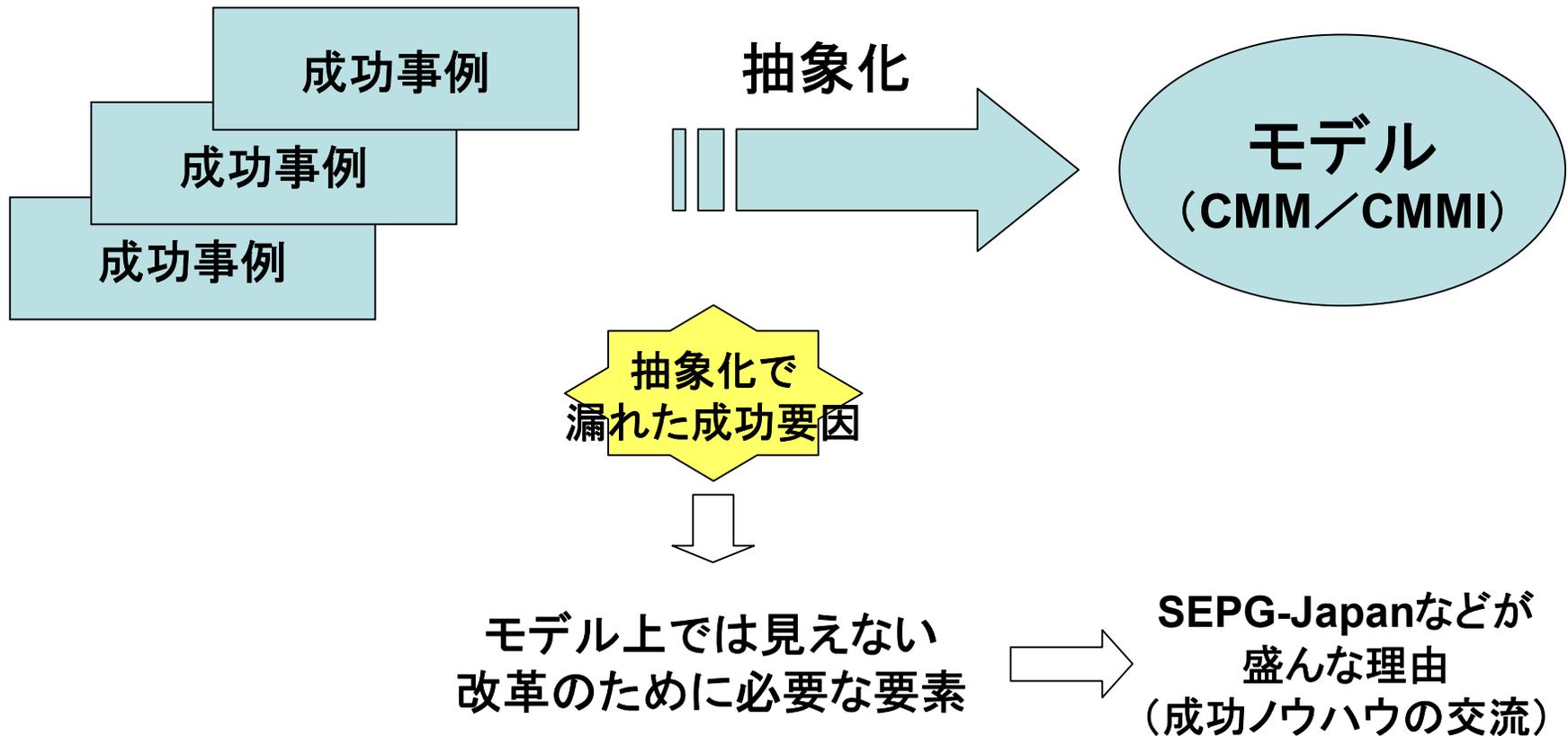
# CMMIの構造



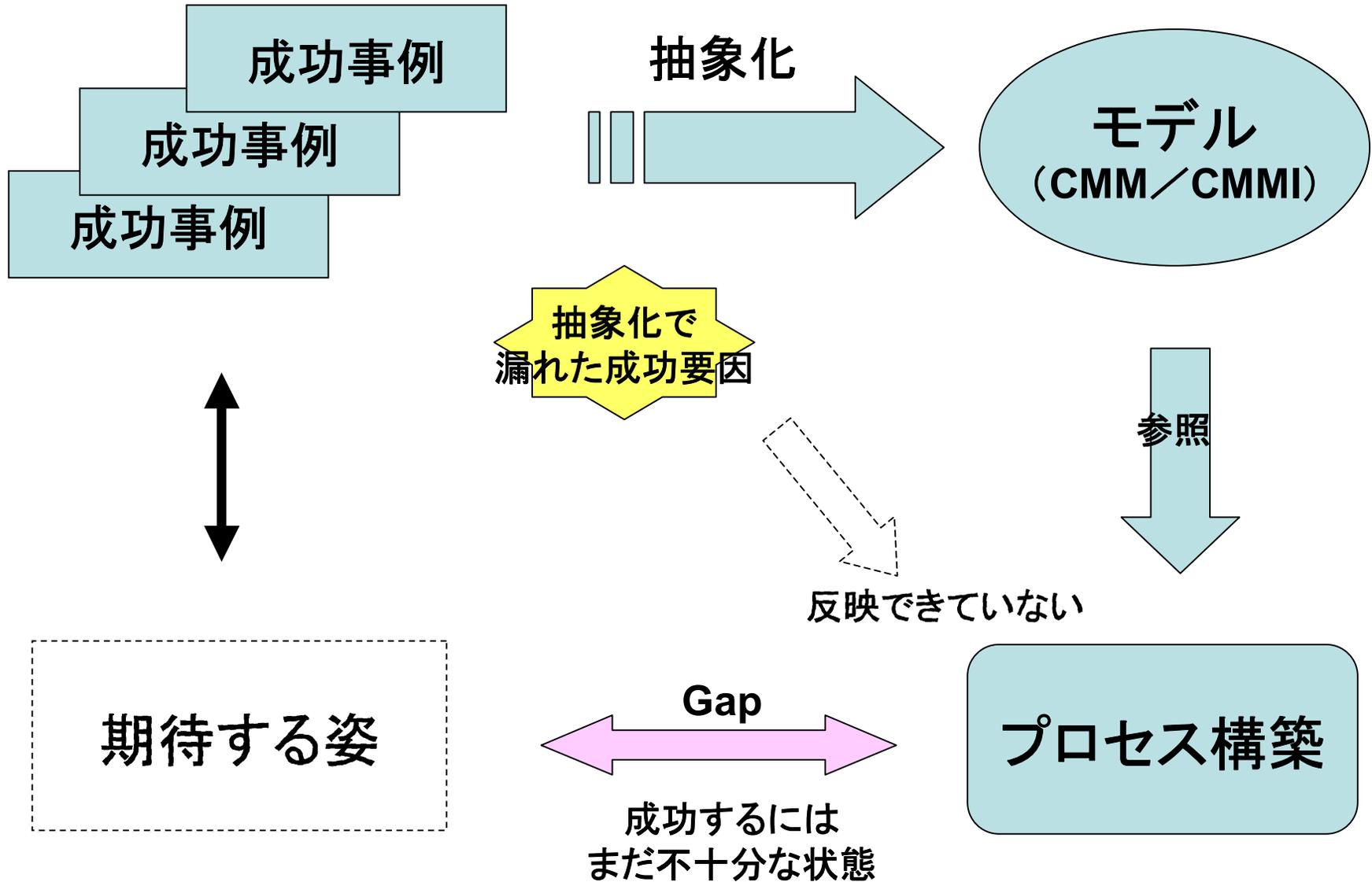
# CMMI実装のコスト



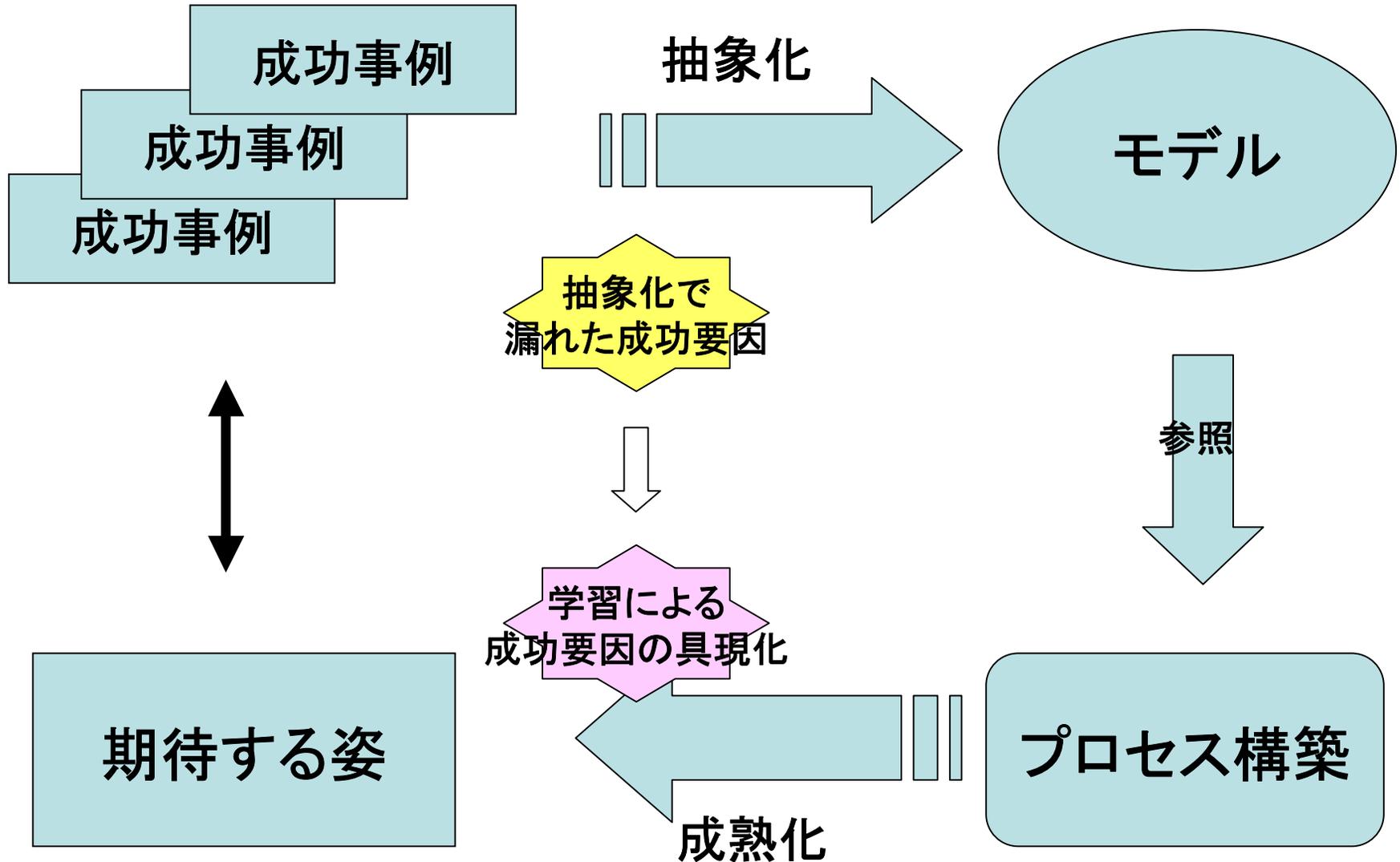
# モデルの限界



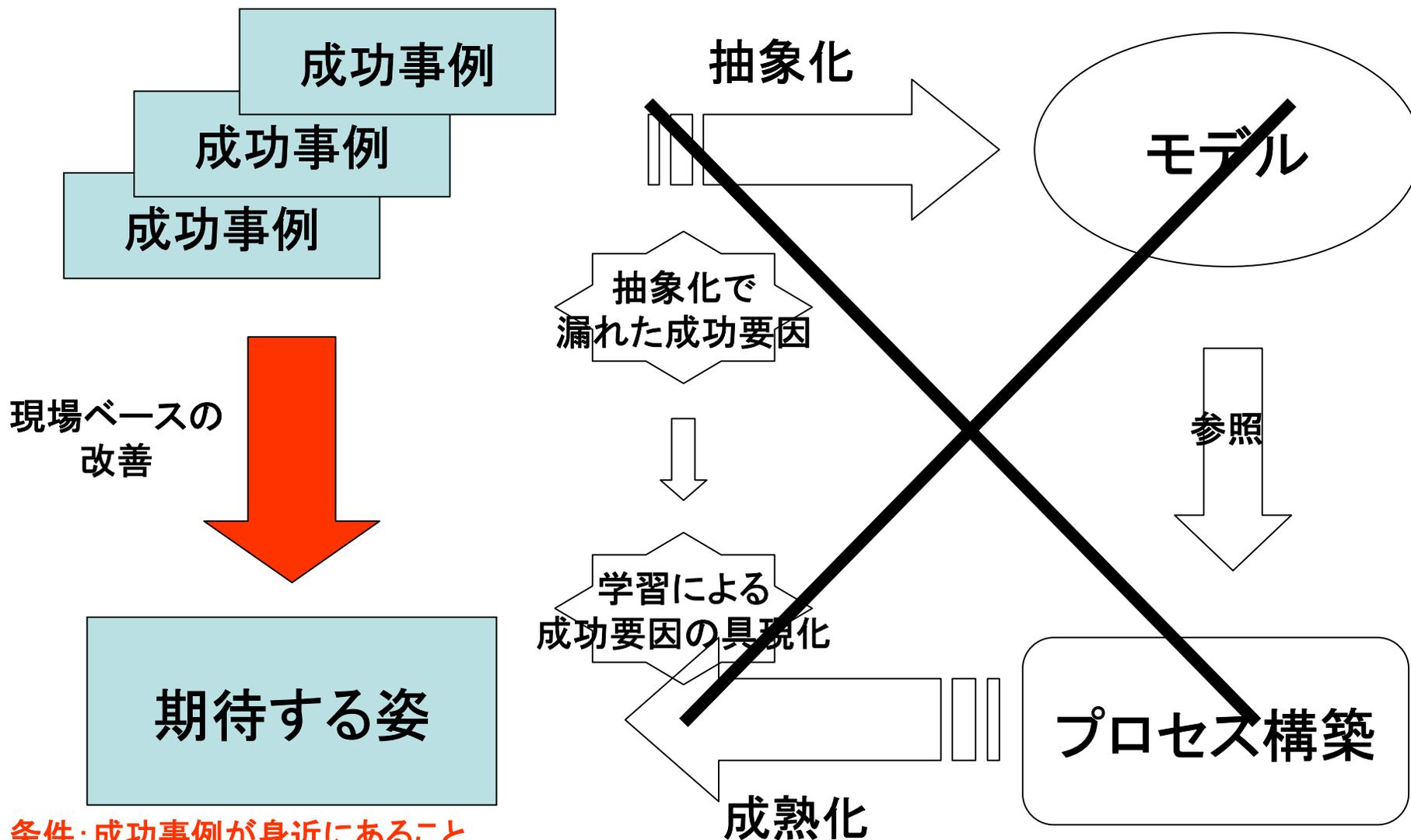
# CMM方式の改善(1)



# CMM方式の改善(2)

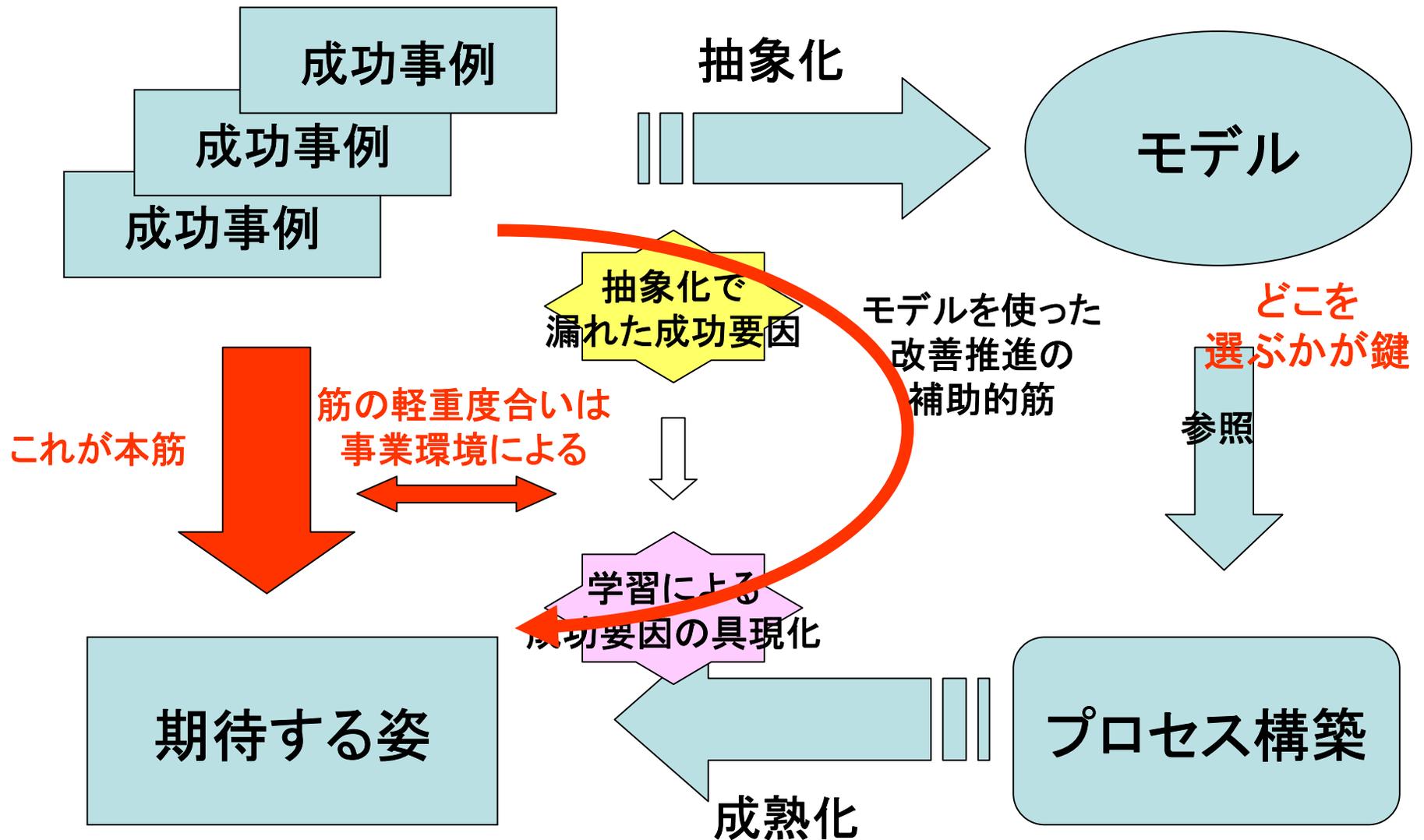


# モデルを使わない改善



条件: 成功事例が身近にあること  
モデルが叫ばれなかったところは皆このタイプだった?

# プロセス改善の統合フレーム



# CMM方式 vs HW生産方式

	CMM方式	HW生産方式
管理	PJ計画主導型管理	自律調整型管理
	前工程重視	後工程引き取り
経営者参画	マネジメントレビュー	現場指導
標準化	制度化できる範囲 (全PJ遵守が原則)	ベストなやり方 (Gapを埋めていくのが改善)
改善の焦点	プロセス重視	人を活かす
プロセス学習	標準の教育	隣から学ぶ、トレーニング
改善の主役	管理者、SEPG中心	現場中心
プロセス維持	明文化による継承	組織文化として伝承
発祥	米国で体系化	日本で体系化

# CMM方式 vs HW生産方式

	CMM方式	HW生産方式
制度化の速度	早い	遅い
現場での定着度	浅い	深い
適応する分野	グローバル拡大型事業 狩猟型事業改善 非繰り返し型(PJ型)作業	国内成熟型事業 農業型事業改善 繰り返し型作業

---

# 今後目指すべき方向は？

---

× 和才洋才

組織の基本方針が不明確  
形ばかりを追うタイプ  
(失敗する典型例)

△ 和才洋魂

CMMを組織の基本方針にし  
時組織のノウハウを当てはめるタイプ  
(そこそこ成功するがすぐに停滞する)

○ 和魂洋才

自組織で築いて来たQMSを  
CMMで補強するタイプ  
(成功する例だが大成功かどうかは疑問)

◎ 和魂和才

自組織で築いて来たQMSを  
自らの力で改善するタイプ  
(大成功を期待できるが土俵が必要)

? 洋魂洋才

M&A型で改善を志向するタイプ  
(大化けするかもしれないがリスクも高い)