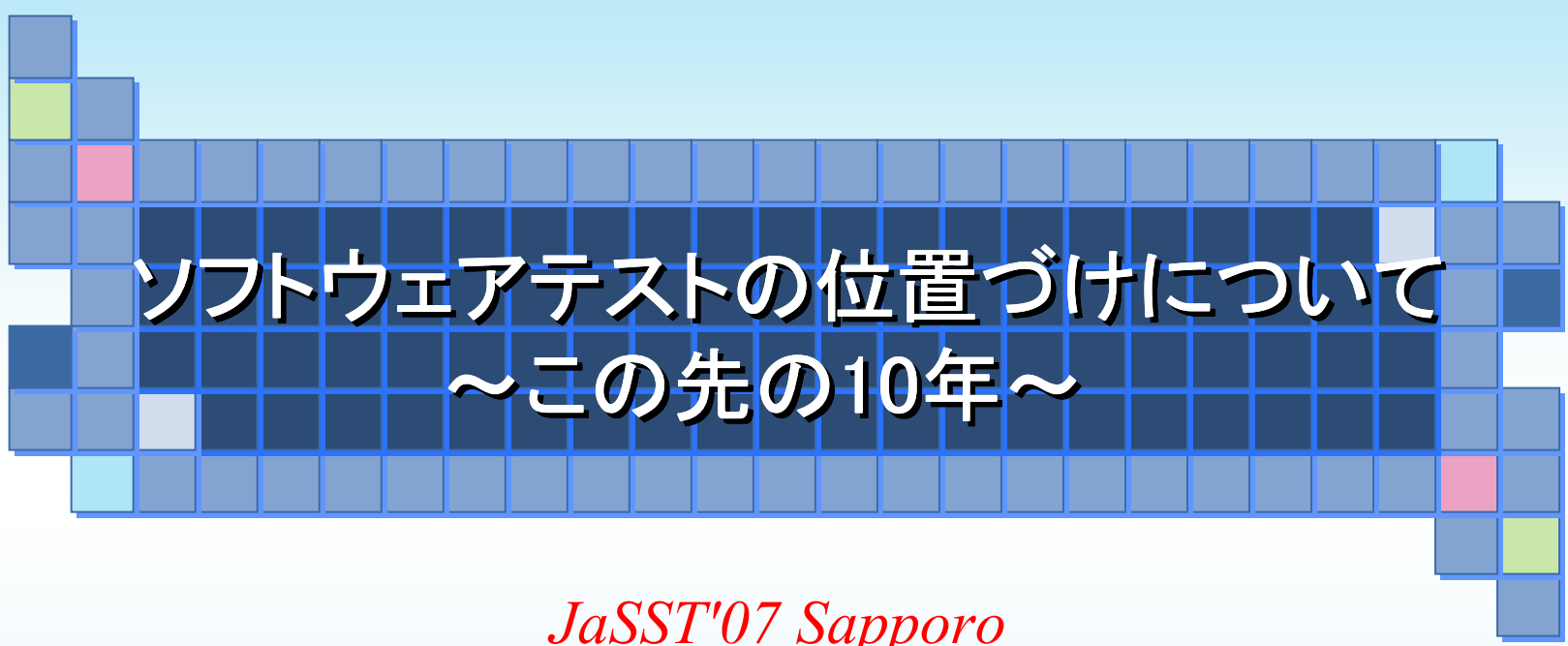




ソフトウェアテストの位置づけについて ～この先の10年～

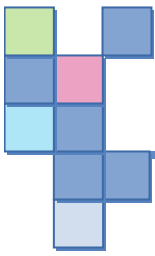
JaSST'07 Sapporo

2007.10.2

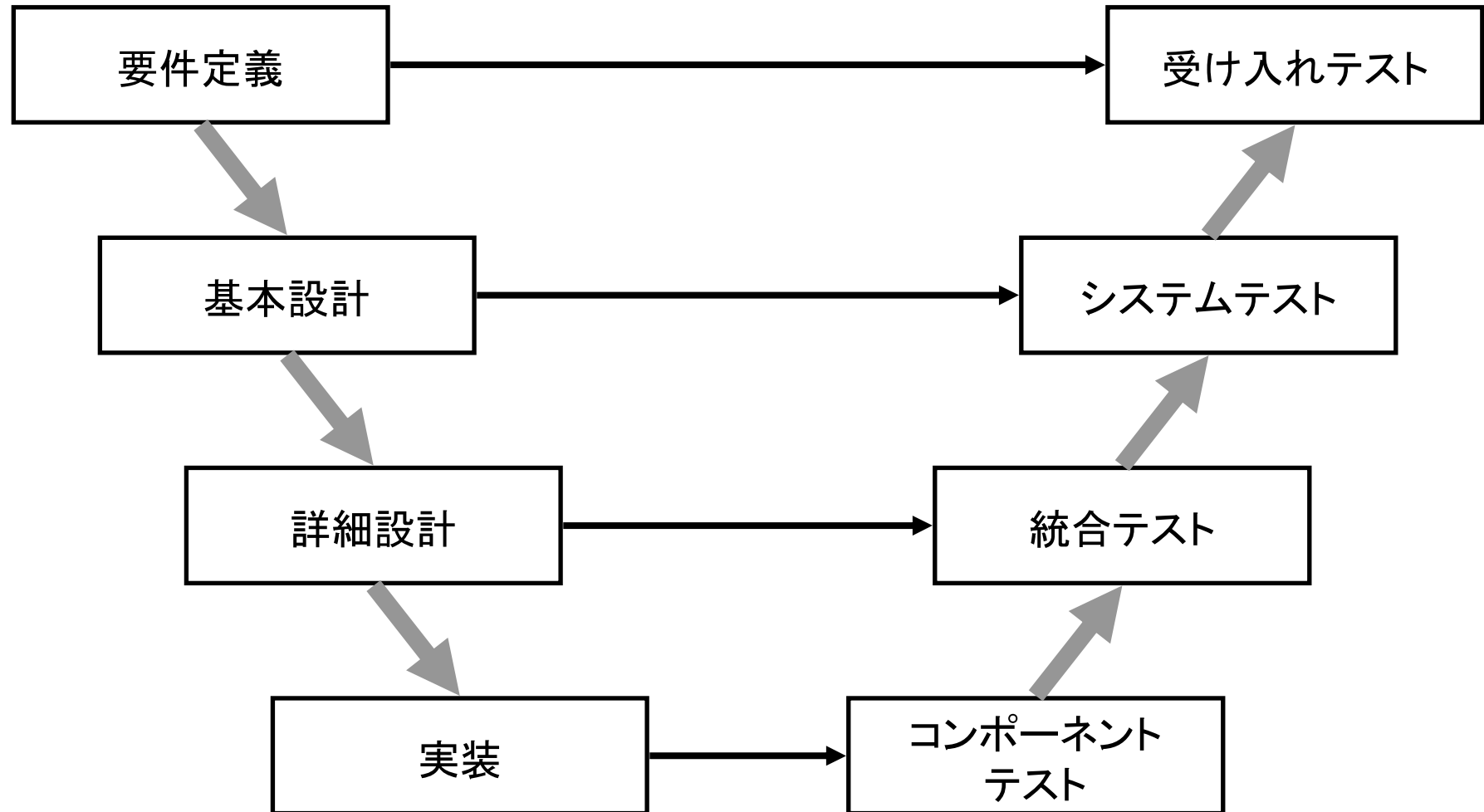
富士ゼロックス株式会社

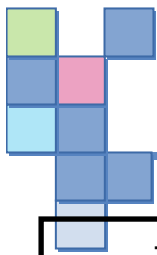
秋山 浩一

(Kouichi.Akiyama@fujixerox.co.jp)

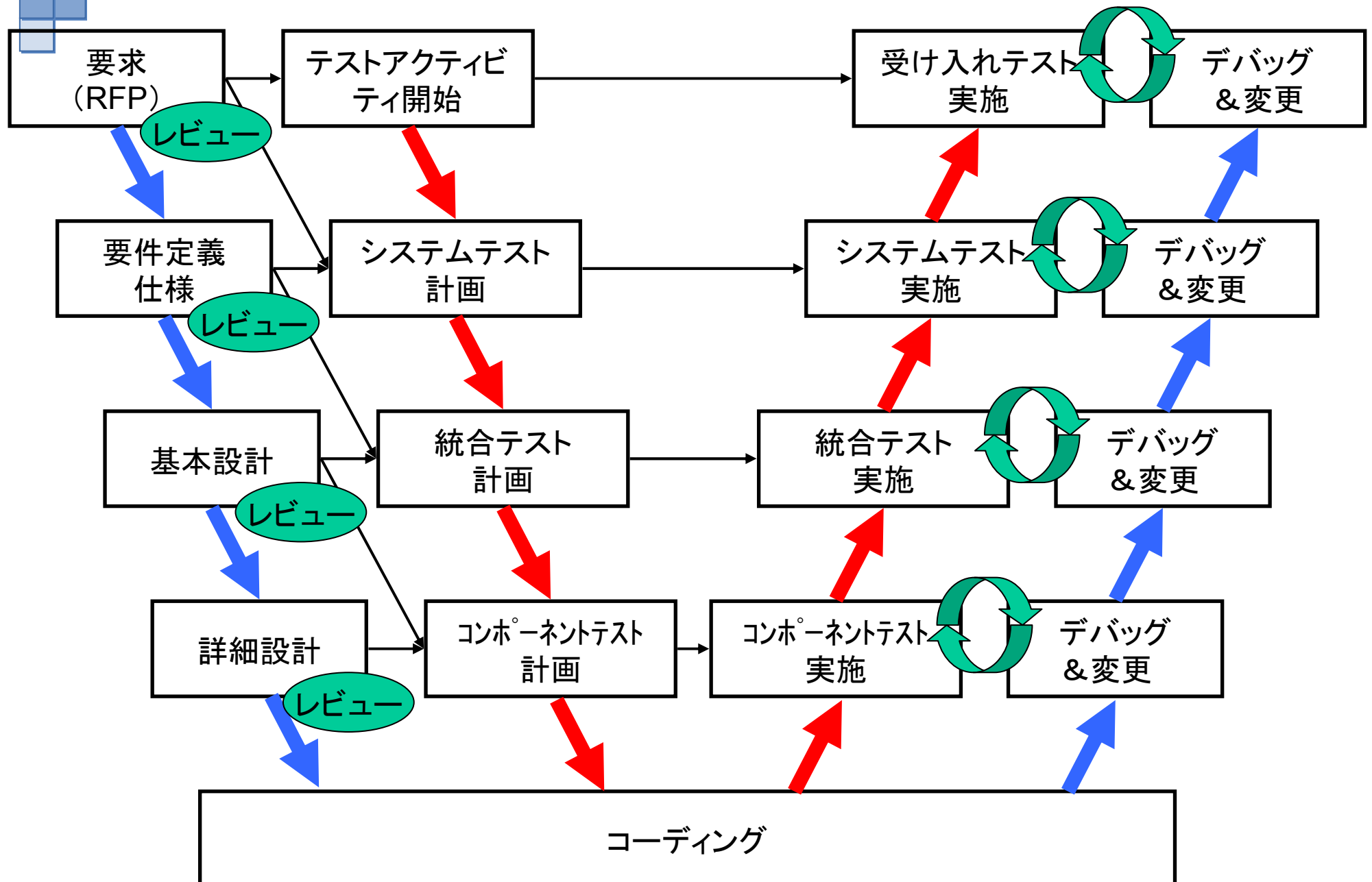


Vモデル

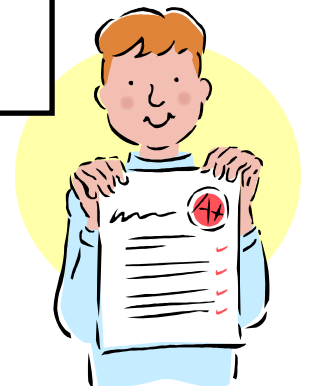
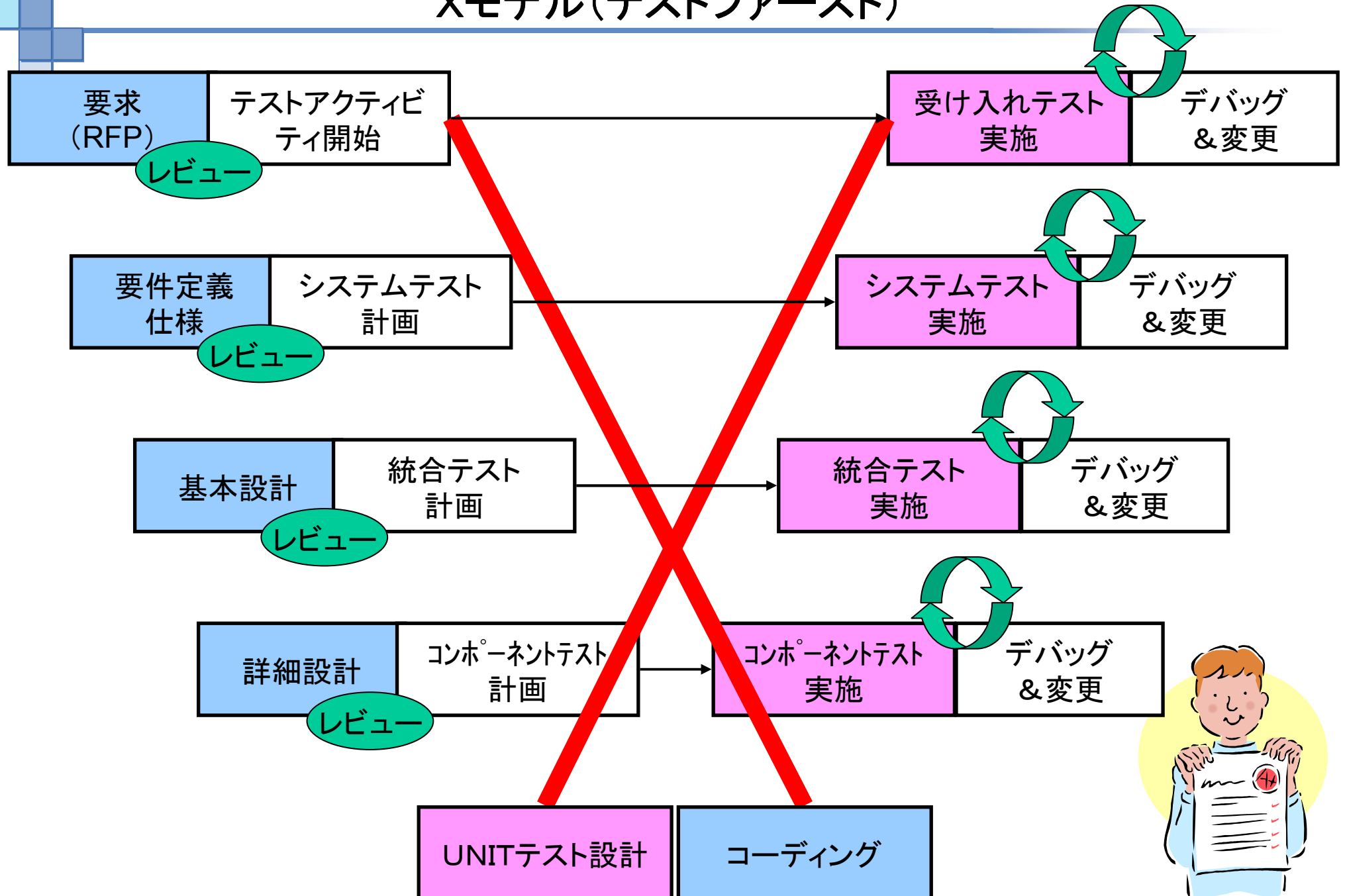




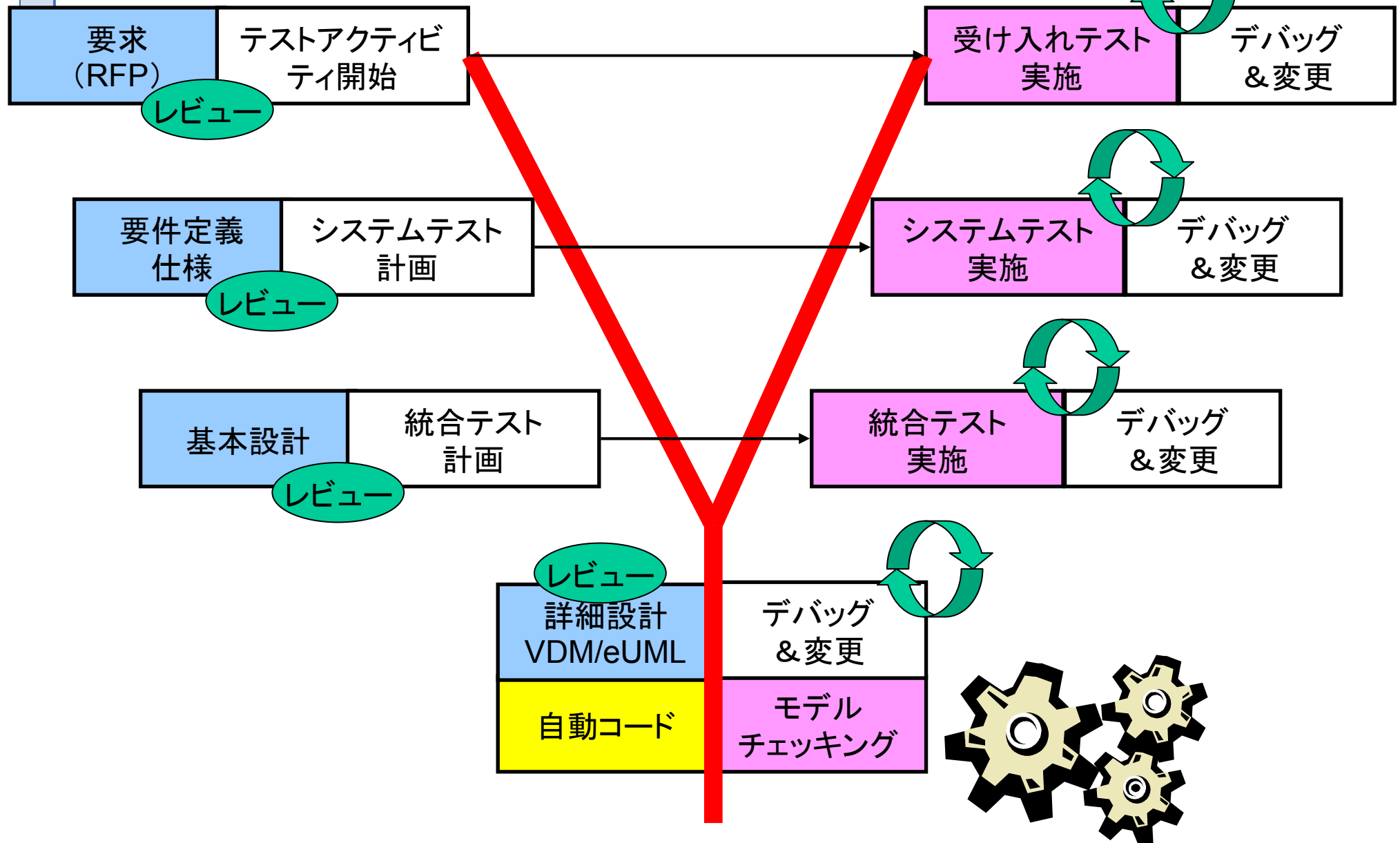
Wモデル



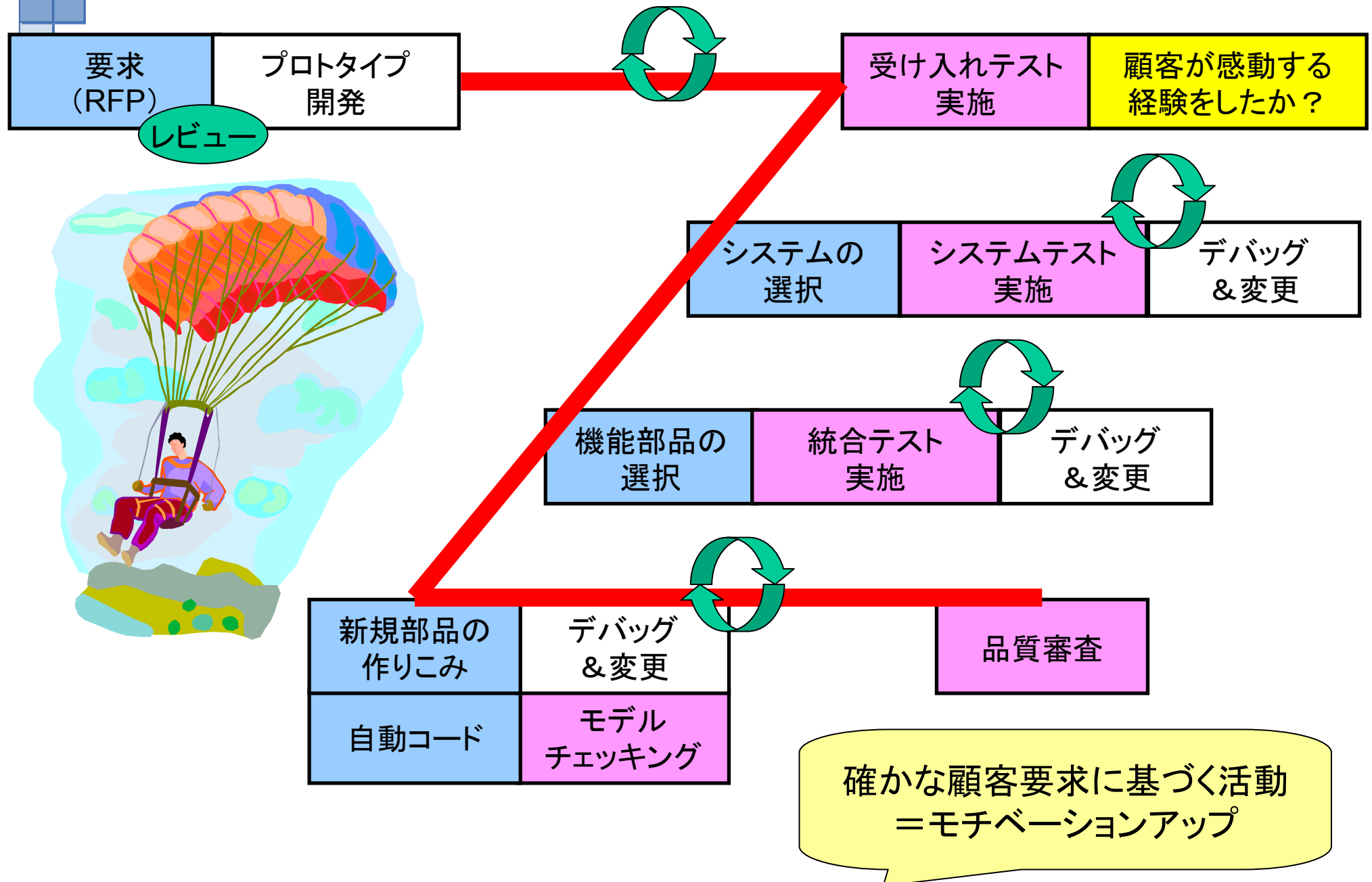
Xモデル(テストファースト)

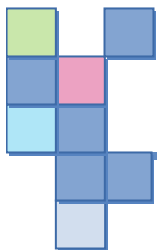


Yモデル(形式仕様記述・モデルチェッキング)

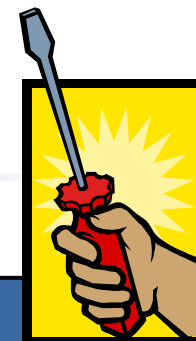


Zモデル(顧客体験による要求の獲得)



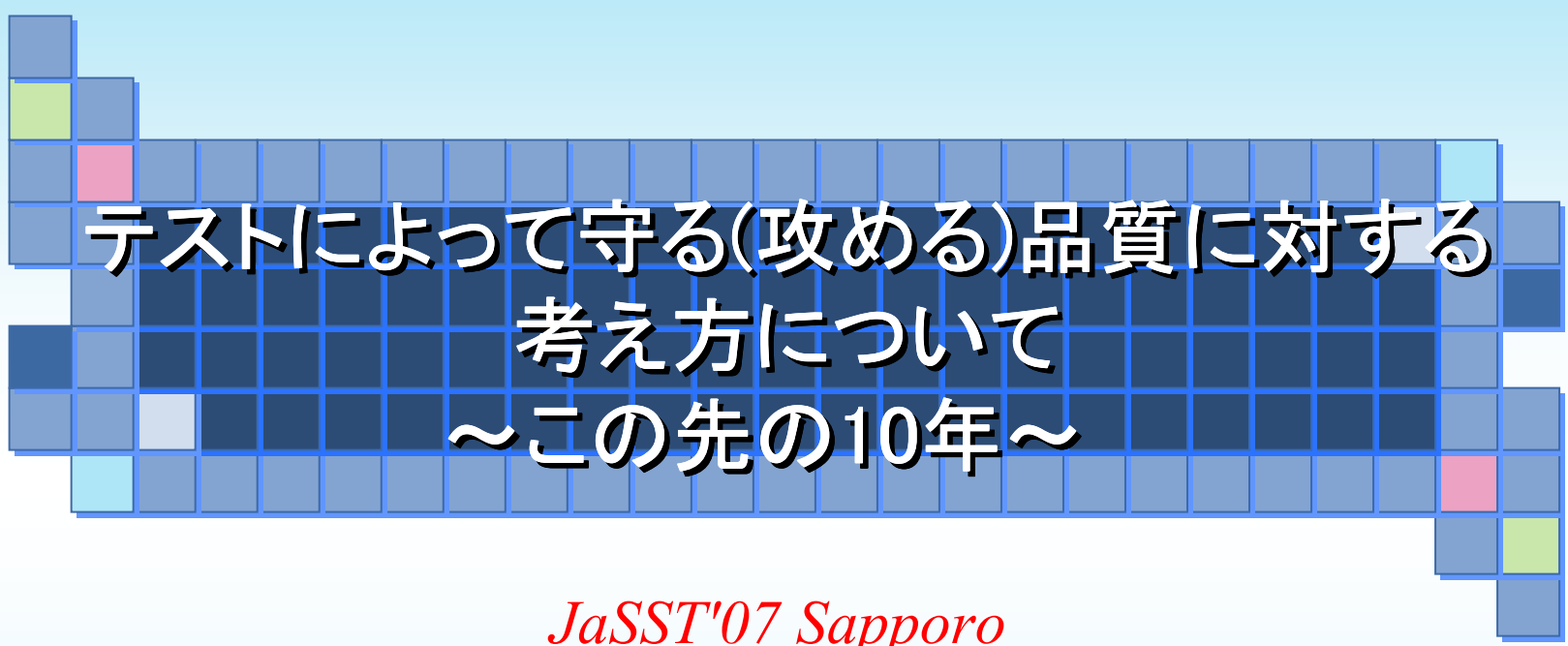


プロセスを変革させる力を持った手法・ツールの出現が鍵



プロセスモデル		概要	手法・ツール
ウォーターフォール		各フェーズで品質確認を行い審査にパスしないと次工程に移行不可	レビュー、工程移行審査、品質点検
Vモデル		設計工程に対応するテスト工程を明確にし、Verification(検証)とValidation(妥当性確認)を分離	ホワイトボックステスト手法 ブラックボックステスト手法
Wモデル		テスト計画の遅れを解消するとともに設計レビューを明確に位置づけている	設計レビューとテスト網羅性をチェックするツール FV表、Mind Map、インベントリ
この先の10年	Xモデル	コーディングに先立ちテスト設計を実施するテストファーストを採用	アジャイル、TDD、xUnitツール
	Yモデル	形式仕様記述やExecutable UMLによりコードを自動生成・自動テスト	VDM、MDA、xUML(BridgePoint)、SPIN
	Zモデル	顧客要求を「顧客感動」という視点ではじめに確認し部品を選択することでシステムを構築	Ruby on Rails,Smarty/PHPなどによるView分離、シミュレーション、部品化

真打の登場はこれから



テストによって守る(攻める)品質に対する 考え方について ～この先の10年～

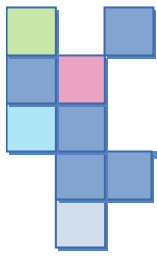
JaSST'07 Sapporo

2007.10.2

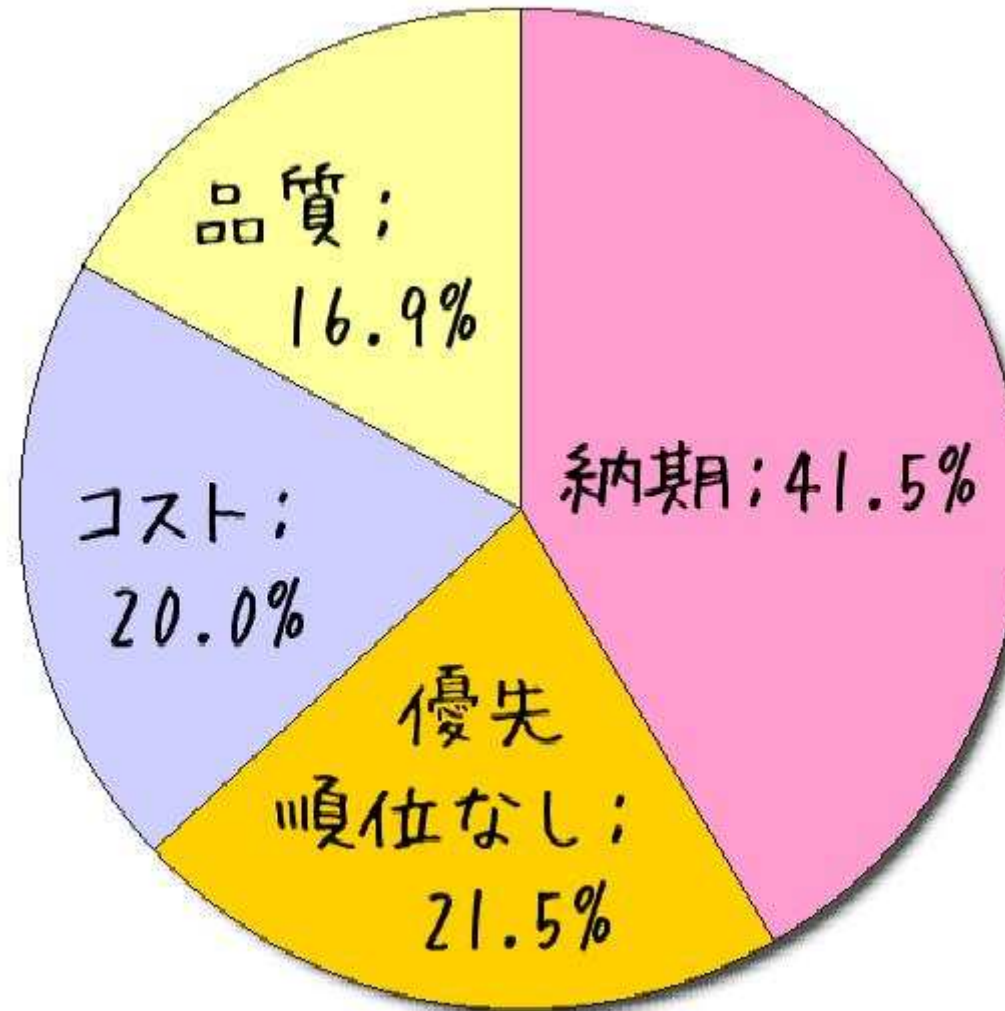
富士ゼロックス株式会社

秋山 浩一

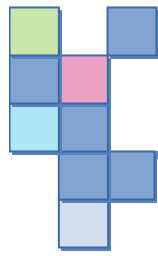
(Kouichi.Akiyama@fujixerox.co.jp)



システム企画時におけるQCDの優先順位



『ソフトウェアメトリックス調査2007』 JUASより



QCDの定量化

QCD	定量化	次元数
Q	商品品質を表す式は存在しない(しいて言えばSN比?) 品質 = ??	多次元
C	ソフトウェア開発コストは工数とほぼ同義(工数に人月単価を掛けたもの) 工数の見積りは、FPなどから求めた規模から、 工数 = $\alpha \times \text{規模}$ または、COCOMOなどでは、 工数 = $\alpha \times \text{規模}^\beta$ ※ 規模が大きくなると指数関数的に工数が増える	一次元 or 二次元
D	工期は、COCOMOなどでは、工数と高い相関を持つといわれており、 工期 = $\alpha \times \beta \sqrt{\text{工数}}$ などの式となる。SECデータベースでは、 工期 = $1.8 \times \text{工数}^{0.38}$	二次元

CとDの式は、『ソフトウェア開発見積りガイドブック』 IPA/SEC編より引用

品質が多次元とは？

◆ ロバート・L・グラス(『ソフトウェア開発55の真実と10のウソ』より)

- ✓ ソフトウェアの品質とは、優れたプログラムが備えるべき七つの属性の集合である。七つの属性とは、移植性、信頼性、効率性、使用性(人間工学)、検証性、理解性、変更容易性である。

◆ ISO9126品質特性

- ✓ 機能性、信頼性、使用性、効率性、保守性、移植性



品質は多次元
(多変数)

バグだけじゃない！

そんな品質をテストによって守る(攻める)には？

◆ トム・デマルコ(『Controlling Software Projects』より)

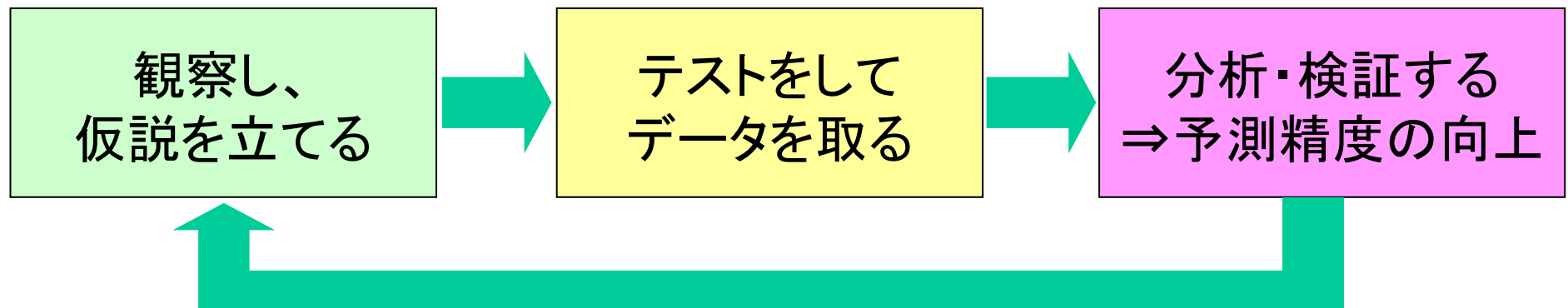
- ✓ You can't control what you can't measure.
- ✓ 測定することにより管理下に置く(正しい状態をキープ)ことができる

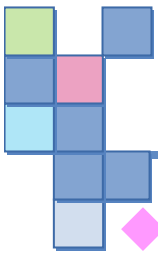
◆ 多次元の品質を測定する方法(Qの定量化へのチャレンジ)

- ✓ 区分原理(事実の収集＝良い・悪いではなく数値で集める)
 - たくさんの品質特性を漏れなくダブリ無く(MECE)分けてそれぞれを定量化する
- ✓ 多変量解析
 - まずは仮説を立ててデータを当てはめて検証してみる
 - 最終的には数値解析(マハラノビスの距離を求めるなどしてパターン解析)

◆ テストで品質を予測できるようになる ← 目的

- ✓ 品質をテストによって守る(攻める)ためには予測ができるようになる必要がある
- ✓ 「果因(結果⇒原因)」の異常対応から、「因果関係」を理解し予測・助言可能な世界へ





品質が予測できるようになったら.....。

- ◆ Control(管理) → Management(マネジメント) → Strategy(戦略)へ
 - ✓ 品質のどこにFocus & Deepするかをテスト部門が決定する
 - ✓ 経営者の仕事を奪う！？
- ◆ ただし、一歩ずつ進む(Neurath's boat)
 - ✓ それぞれの組織には歴史がある
 - 海原で修理しながら船を進めるようなもの
 - みな海の上の船(完璧な土台の上に建っているわけではない)
 - 成功している企業のツールは使っていなくても自社にあった別のもので代用している可能性
- ◆ テストの責任
 - ✓ 品質を予測・助言する = 予測・助言結果に対する責任の発生
 - 機会損失
 - 過剰品質(cf. 正しい品質)
 - 財務への影響
 - ✓ それでも、他にできる人がいないのでやるしか.....ない！？