

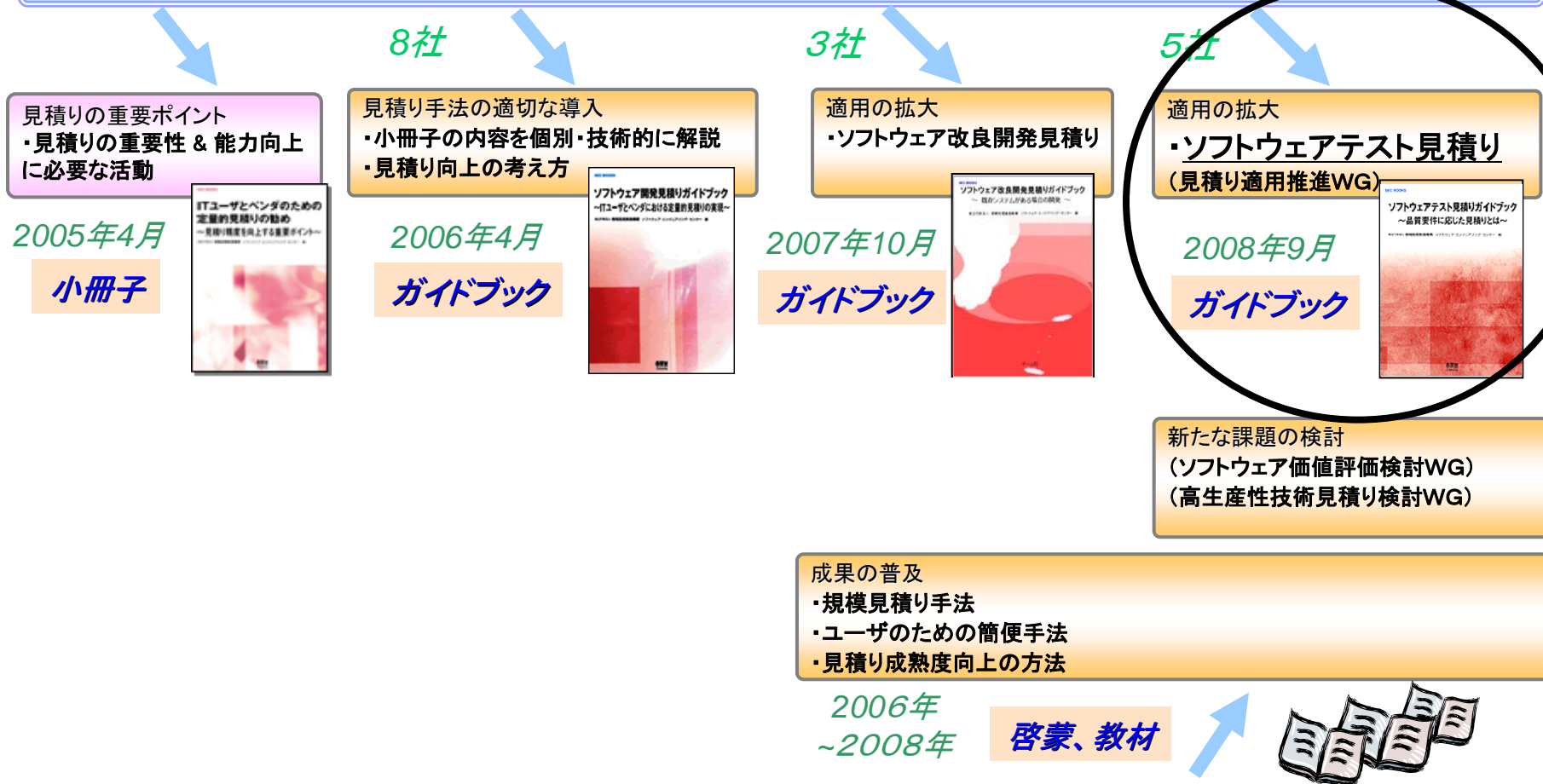
「ソフトウェアテスト見積りについて」 ～品質要件に応じた見積りとは～

2008年10月28日(火)

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
高橋 茂

見積り手法に関する活動

国内企業の成功事例 ⇒ 収集・特徴を分析



先進的な見積り手法 ⇒ 国内企業で実証・評価

本書の位置づけ

- これまでの書籍では、ソフトウェアテスト工数・コストはプロジェクト当初の見積りに含まれている前提
- 但し、テスト工数の見積りには様々な課題がある
 - テスト終了時の残存欠陥の予測が困難
 - テスト量と品質要求とのトレードオフ
 - テスト完了基準など、ユーザ／ベンダ間での合意
 - 非機能要件の把握とテスト量への反映
- 本書は、既発刊のガイドブックを補足し、ソフトウェアの品質検証と妥当性確認に関わるテスト見積り(テスト量、テスト生産性)の方法やノウハウを紹介

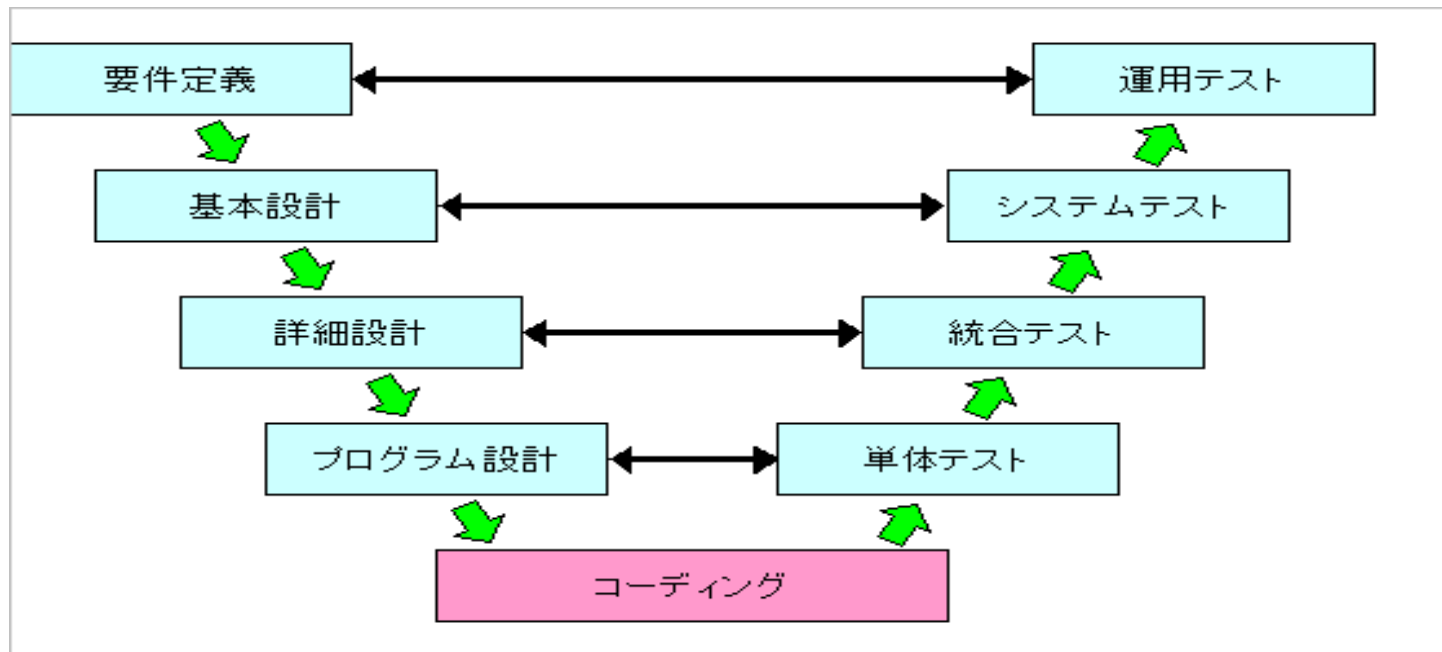
ソフトウェアテスト見積りガイドブック(目次)

目次・構成	目次・構成
第1部 総論	
第1章 ソフトウェアテスト見積りの課題	
1.1 システムの信頼性向上からみたソフトウェアテスト見積りの課題	4.2 テスト量と品質目標値
1.1.1 レビューおよびテストを鑑みた残存欠陥の予測	4.2.1 一般的事項
1.1.2 テスト量の論理的把握と品質の完全性保証	4.2.2 残存欠陥密度の設定
1.1.3 改造開発および仕様変更におけるテスト量の把握	4.2.3 レビューおよびテストでの欠陥検出戦略の統合
1.1.4 非機能要件の把握と確認	4.2.4 ソフトウェアのテスト完了基準
1.2 ソフトウェア品質と価格の関係に関する認識の統一	4.2.5 テストの網羅性とソフトウェアテスト量
第2章 開発プロセスモデルとテスト手法	4.3 テスト網羅性尺度とテスト量見積り方法
2.1 品質保証から見た開発プロセスモデルとテスト方法	4.3.1 ホワイトボックステストでの網羅性尺度とテスト量見積り方法
2.2 テストプロセスとソフトウェア開発プロセスモデル	4.3.2 ブラックボックステストでの網羅性尺度とテスト量見積り方法
2.3 テスト手法の一般的事項	4.3.3 合理的なテスト量の削減方法
2.3.1 ソフトウェアテスト方法	4.4 仕様変更量とテスト量
2.3.2 ソフトウェアテスト技法	4.4.1 仕様変更量と開発量(テスト量)との関係
2.4 ソフトウェアテストの見積りの範囲	4.4.2 見積りを行う上での仕様変更の考慮点
2.5 テストプロセスおよびテストドキュメント	4.5 テストの生産性に影響する変動要因
2.5.1 テストプロセス	4.6 非機能要件の把握とテスト見積りへの反映
2.5.2 テストドキュメントの記述項目	4.6.1 非機能要件の把握と確認方法
第3章 ソフトウェアテスト見積りでの成功と失敗例	4.6.2 非機能要件のテスト見積りへの反映
3.1 ソフトウェアテスト全般に関する事例	4.7 欠陥修正に関わる工数の把握
3.2 テスト戦略とソフトウェアテスト見積りに関わる事例	4.7.1 欠陥修正量(工数)の把握
3.3 テスト進行中における品質管理と再テスト見積り	4.7.2 欠陥修正による再テスト工数の把握
3.4 テスト戦略の重要性	第5章 ソフトウェアテスト見積り精度の向上
3.4.1 テストレベル	5.1 見積りの前提条件のモニタリングとコントロール
3.4.2 テスト戦略	5.2 見積り手法と継続的な改善
第4章 ソフトウェアテスト見積りの詳細	5.3 契約によるリスク解消の糸口
4.1 ソフトウェアテスト見積りの手順	5.3.1 見積りにおけるユーザ企業とベンダ企業の役割
	5.3.2 変動要因に関するユーザ企業とベンダ企業の調整プロセス
	5.3.3 ソフトウェアテストの段階的な見積りと多段階契約の採用
	添付資料(非機能要件の把握・確認とテスト見積りへの影響表)
	第2部 事例編
	日本ユニシス、日立製作所、東京海上日動、ジャステック、NECの5社の事例を紹介

■品質保証の観点から見たモデル

<V字型モデル>

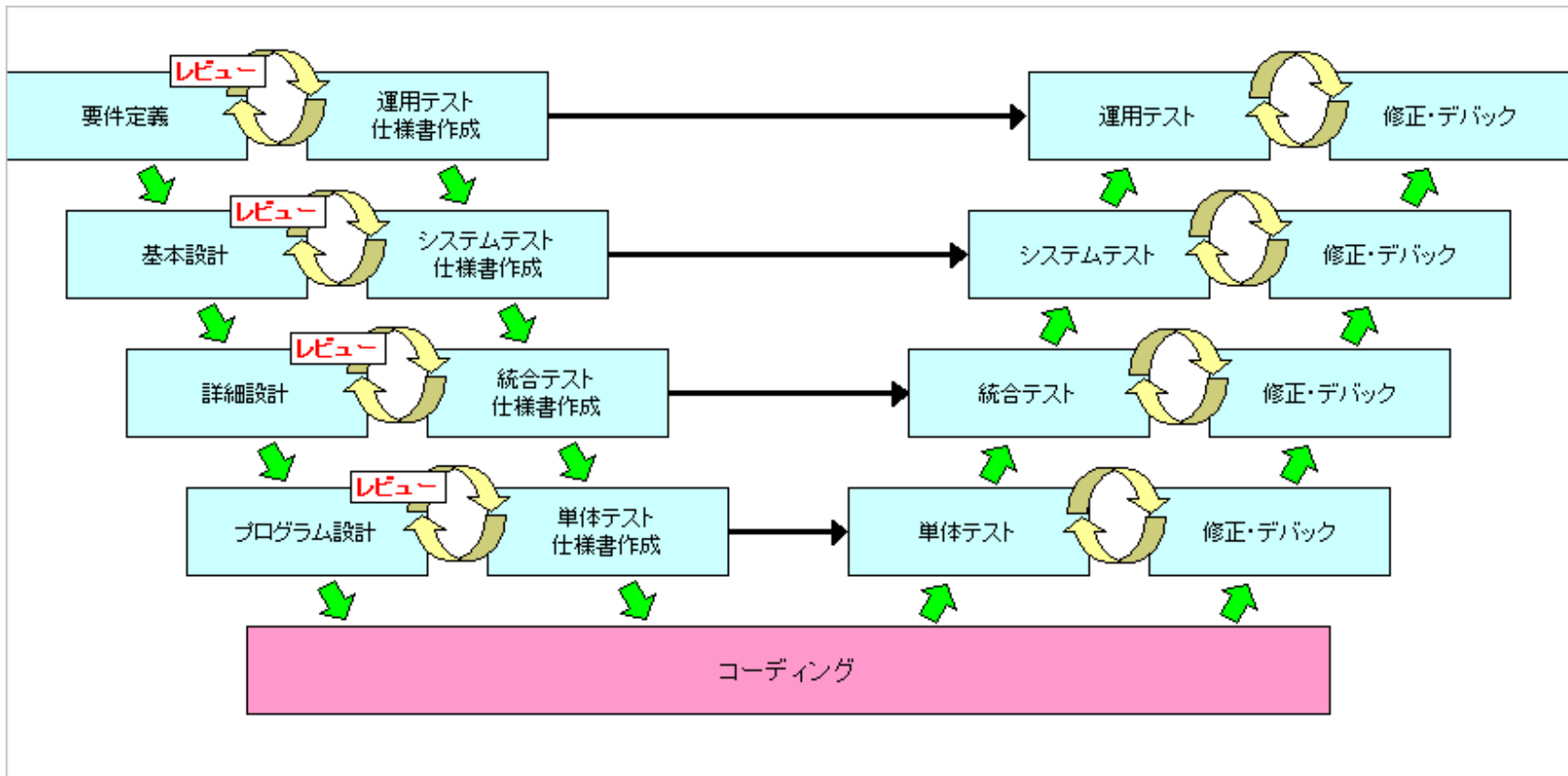
設計・実装工程で作り込まれたソフトウェアの欠陥は、当該設計・実装工程のレビューおよび相対するテスト工程で検出する。



第2章 開発プロセスモデルとテスト手法

<W字型モデル>

テストファースト(開発初期段階からテストを計画しながら進める)の考え方にに基づき、設計・実装工程で作り込むソフトウェアの欠陥を、当該設計・実装工程のレビューで検出し、さらに相対するテスト工程のテスト設計を開発工程の水準にあわせて同時並行して行う。



■ テスト方法の分類

● ブラックボックステスト

- 要求仕様や外部仕様を基にしてテストを設計する方法で、ユーザ視点で、外部からの入力に対するアウトプットを検証するテスト

● ホワイトボックステスト

- テスト対象ソフトウェアの内部パス、構造、設計情報に基づいてテスト設計する方法

■ 代表的なテスト技法

● ブラックボックステスト

- 同値クラステスト、境界値テスト、異常値テスト／無効値テスト、ペア構成テスト(オールペア法、直交表)、状態遷移テスト、ドメイン分析テスト、デシジョンテーブルテスト、ユースケーステスト

● ホワイトボックステスト

- 制御フローテスト／制御パステスト、データフローテスト

第3章 ソフトウェアテスト見積りでの成功と失敗例

事例5: 保留していた仕様変更をシステムテストで対応し、テスト負荷の増加と品質低下を招いた失敗事例

課題となっている状況・状態

システムテスト工程で「保留していた仕様変更と新たな仕様変更」が大量に発生したが、テスト期間や要員資源を限定されたため、結果として十分なテストが実施できず品質低下を招いた。

見積りの成功/失敗の度合い

スケジュール厳守したが、品質は大幅に低下した。

- ・ テスト密度(ケース数/ソフト規模) 見積り比 0.5倍
- ・ 本番障害発生頻度: 見積り比 5倍

教訓・メッセージ

同じ機能の仕様変更対応でも、タイミング(取り込み工程)が異なると、対応コストが増減する特質がある。仕様変更の取り込みは、システムテスト工程が一番のコスト高(棄却工数増など)となる。また、ソフトウェアの品質劣化(デグレード)を招きやすい。このような仕様変更の特質をユーザとベンダ相互に理解することが重要である。

成功と失敗例に関して 31事例を紹介

- ソフトウェアテスト全般に関する事例(10件)
- テスト戦略とソフトウェアテスト見積りに関わる事例(18件)(品質指標・目標値、コンポーネントの重要度、テストプロセス、テスト環境、テスト分担)
- テスト進行中における品質管理と再テスト見積りでの事例(3件)

第3章 ソフトウェアテスト見積りでの成功と失敗例

事例11: 標準的な品質目標値をそのまま使用して失敗した事例

課題となっている状況・状態

基盤開発で、ユーザ提示のテスト密度(50項目/KS)とテストケース抽出方法を前提に、テスト見積りを行ったが、テスト量(テストケース数)の増加となった。

見積りの成功/失敗の度合い

抽出方法が細かかったためテストケース増加となったが、幸いにも1ケースあたりのテスト実施時間が短時間で済んだため、コストへの影響は少なくて済んだ。

テスト密度(ケース数/ソフト規模): 見積り比 2.0倍
テストコスト: 見積り比 1.05倍

教訓・メッセージ

標準的な品質目標値を、そのまま、テスト対象システムに適用することはきわめて危険である。品質目標値はテスト戦略において、対象システムの特質などを見きわめ(分析・評価)、その結果をテスト見積りの入力情報にすることが必要である。

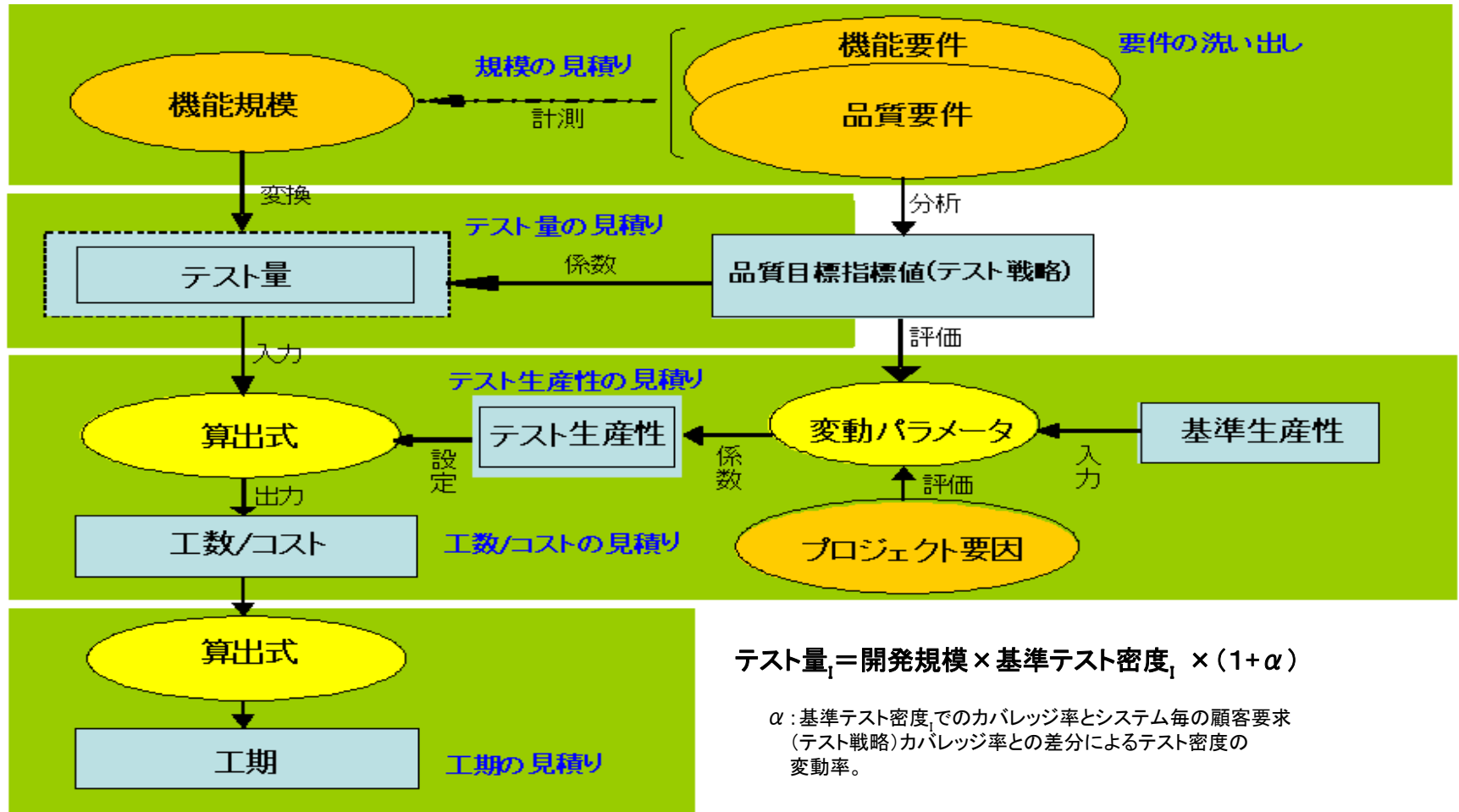
■ テスト戦略の必要性

- 事例を鑑みて、見積り精度向上のためには、プロジェクトの見積り時点で、テスト戦略を定め、これを見積りのインプットにすることが重要

■ テスト戦略の具体例

- **テスト技法の選択**・・・同値クラス、ペア構成(直交表)、状態遷移など
- **品質指標・目標値の設定**・・・テスト密度、テスト網羅率、欠陥密度など
- **テスト環境**・・・自動化テストツール、結果照合ツールなど
- **テスト分担**・・・ユーザとベンダの役割など
- **テスト進行中における品質管理**・・・信頼度成長曲線、テスト完了基準など

■テスト見積りの基本手順



■ テスト戦略に基づく品質目標値

- 品質目標値は、次のような品質指標に基づいて、テスト戦略策定を通して設定
 - 残存欠陥密度、工程ごとの欠陥検出密度
 - テスト完了基準
 - テスト網羅率

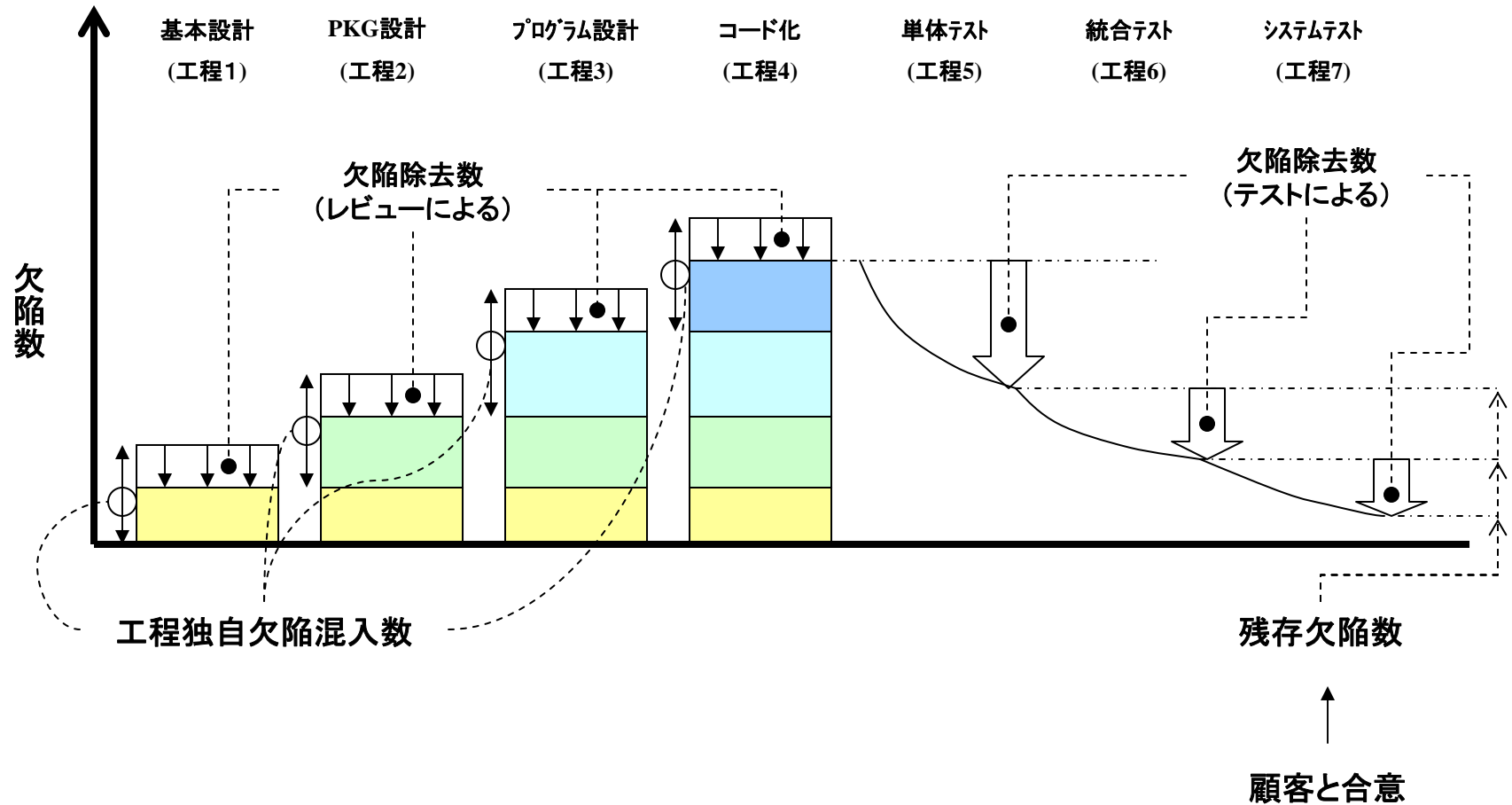
■ 残存欠陥密度(欠陥検出戦略)の設定

- 出荷時の残存欠陥数を測ることは困難
 - 出荷後に発見した欠陥数を測定するしかない
- ユーザと出荷時の残存欠陥密度を合意する必要がある
 - 現実にはソフトウェアテストをどこまで実施するかということを利用者と合意することになる
- 工程ごとの欠陥除去数と残存欠陥数との関係は次頁のとおり

第4章 ソフトウェアテスト見積りの詳細 **SEC**

Software Engineering
for Mo·No·Zu·Ku·Ri

設計・実装工程とテスト工程での欠陥除去数と残存欠陥数との関係



ジャステック社からの出典

■ テスト網羅性の尺度とテスト量見積り方法

● ホワイトボックステスト

- 命令網羅率(C0),分枝網羅率(C1),条件網羅率(C2)
- 開発規模、基準テスト密度、上記網羅率などからテスト量を算出

● ブラックボックステスト

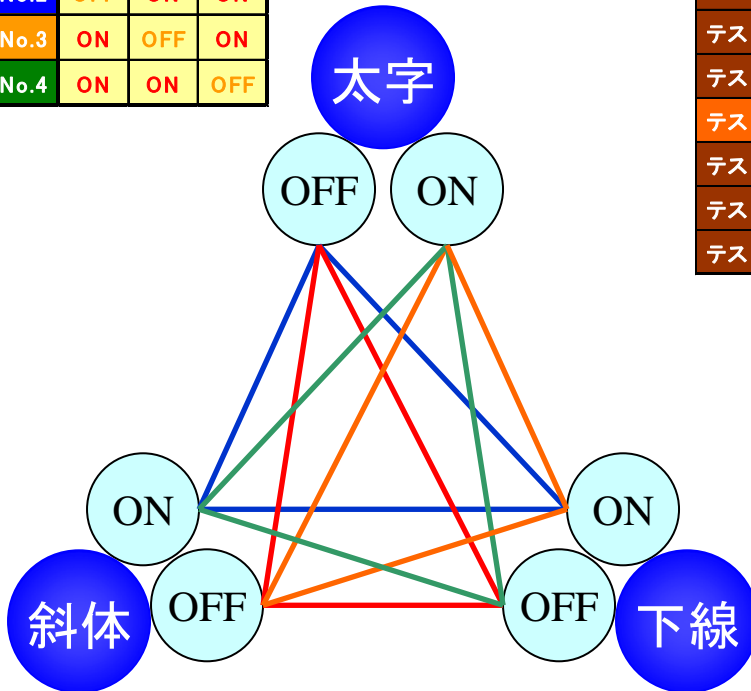
- ユースケース網羅、状態遷移網羅など
- $\text{テスト網羅率} = (\text{実際にテストするテスト項目の数} \div \text{抽出されうる全てのテスト項目の数}) \times 100$
- テスト項目の抽出対象:要件定義、設計書、マニュアル類

第4章 ソフトウェアテスト見積りの詳細

合理的なテスト量削減方法(例:直交表)

(直交表が網羅する組合せ)

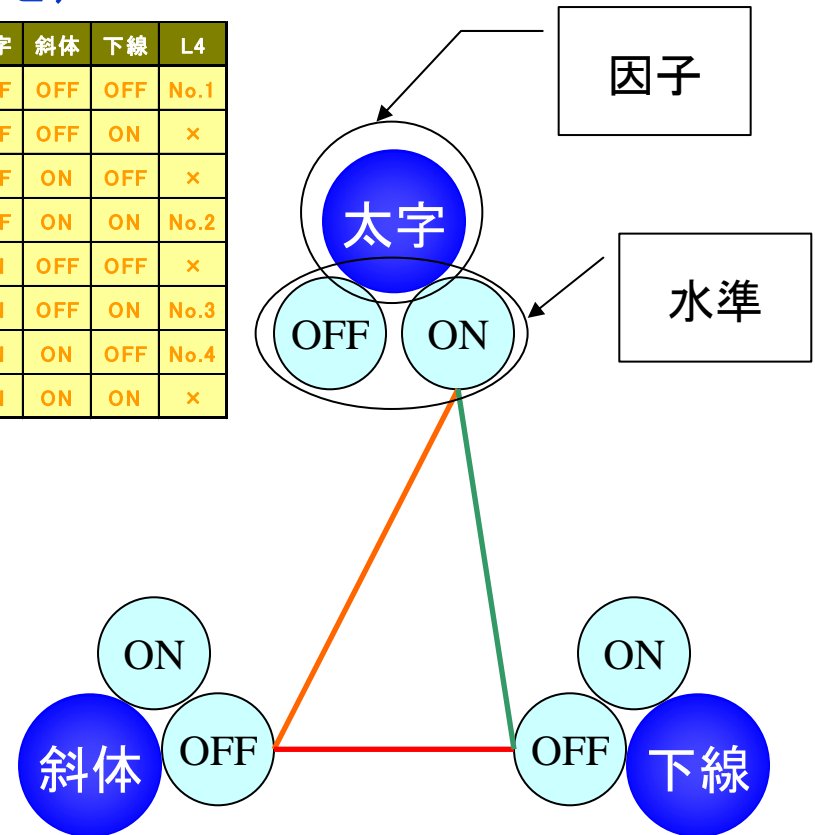
L4直交表	太字	斜体	下線
テストNo.1	OFF	OFF	OFF
テストNo.2	OFF	ON	ON
テストNo.3	ON	OFF	ON
テストNo.4	ON	ON	OFF



2機能間の組合せが全て存在している
(この状態を組合せ網羅率100%と定義する)

(全組合せ)

全組合せ	太字	斜体	下線	L4
テストNo.1	OFF	OFF	OFF	No.1
テストNo.2	OFF	OFF	ON	×
テストNo.3	OFF	ON	OFF	×
テストNo.4	OFF	ON	ON	No.2
テストNo.5	ON	OFF	OFF	×
テストNo.6	ON	OFF	ON	No.3
テストNo.7	ON	ON	OFF	No.4
テストNo.8	ON	ON	ON	×



全組合せでは直交表に存在しない組合せも出現する
よく見ると直交表では現れないNo2, 3, 5は単機能テスト内容

(出典) 「ソフトウェアテストにおける直交表の活用」 (富士ゼロックス株式会社 秋山浩一)

第4章 ソフトウェアテスト見積りの詳細

■ 非機能要件の把握とテスト見積りへの反映

資料「非機能要件の把握・確認とテスト見積りへの影響表」(保守性1/4)

品質特性	副品質特性	詳細分析ニーズ (測定目的)	測定項目	項目定義と測定方法	測定尺度と導出式	解釈	プロセス							見積変動対象 (○該当)				
							組織統制	要求定義	要件定義	設計	プログラム作成	システムテスト	運用テスト	保守・運用	100126 (○該当)	テスト量	テスト生産性	
保守性	解析性	活動記録保有能力の評価 (故障原因の解析のため)	データログ実装率	故障の原因となった特定の操作を見つけるために、記録することになっているデータ項目の中、実際に記録されるデータ項目数の割合。	実勢に記録されたデータ項目数 / 記録することになっているデータ項目数	0~1 高いほど良い			②	③			⑤		○			
			状況監視データ取得成功率	ソフトウェアの状態を記録したモニターデータについて、保守担当者が取得しようとしたケースの中、取得できた割合。	モニターデータに(取得できなかった数 / 全ケース)											○		
		故障原因解析能力の評価	診断機能実装率	仕様で要求される診断機能のうち、実装された診断機能の比率	診断機能実装数 / 要求数													
			故障原因判明率	登録されている故障のうち、原因が分かっている故障の数の割合。														
		故障解析時間	故障解析にかかった平均時間。															
		解析容易性の評価	保守ドキュメント充足	解析容易性向上につながる保守ドキュメントについて、実際に用意できているドキュメント数。以下に具備候補の保守ドキュメント例を挙げる。 ・機能仕様書 ・DBクロスリファレンス ・データ項目クロスリファレンス ・トランザクションリファレンス ・変更手順書(組織変更、制度変更、限度額変更等)	実際ドキュメント数													
			トレースツール利用率	実装機能をトレースする際に、トレースツールを利用できた機能数の割合。	ツールを用いてトレースした実装機能数 / トレースした実装機能数	高いほど良い	①		②	③	④	⑤	⑤	⑤		○		
			プログラムソースコメント率	プログラムソースに組み込んだコメント行の割合。	実装したコメント率 / 組織で定めている標準コメント率	0~1 高いほど良い	①		②		⑤							

測定項目* (約180項目) について、
テスト量およびテスト生産性に
影響する項目を識別した。

* : 機能性、信頼性、使用性、効率性、
保守性、移植性、運用性、障害抑制性

①: 要求仕様の決定(ユーザの役割) ②: 要求を実現するための仕様提示(ベンダSE、またはユーザSE)
③: 設計仕様への盛り込み(および確認) ④: 実装結果の確認 ⑤: 最終確認

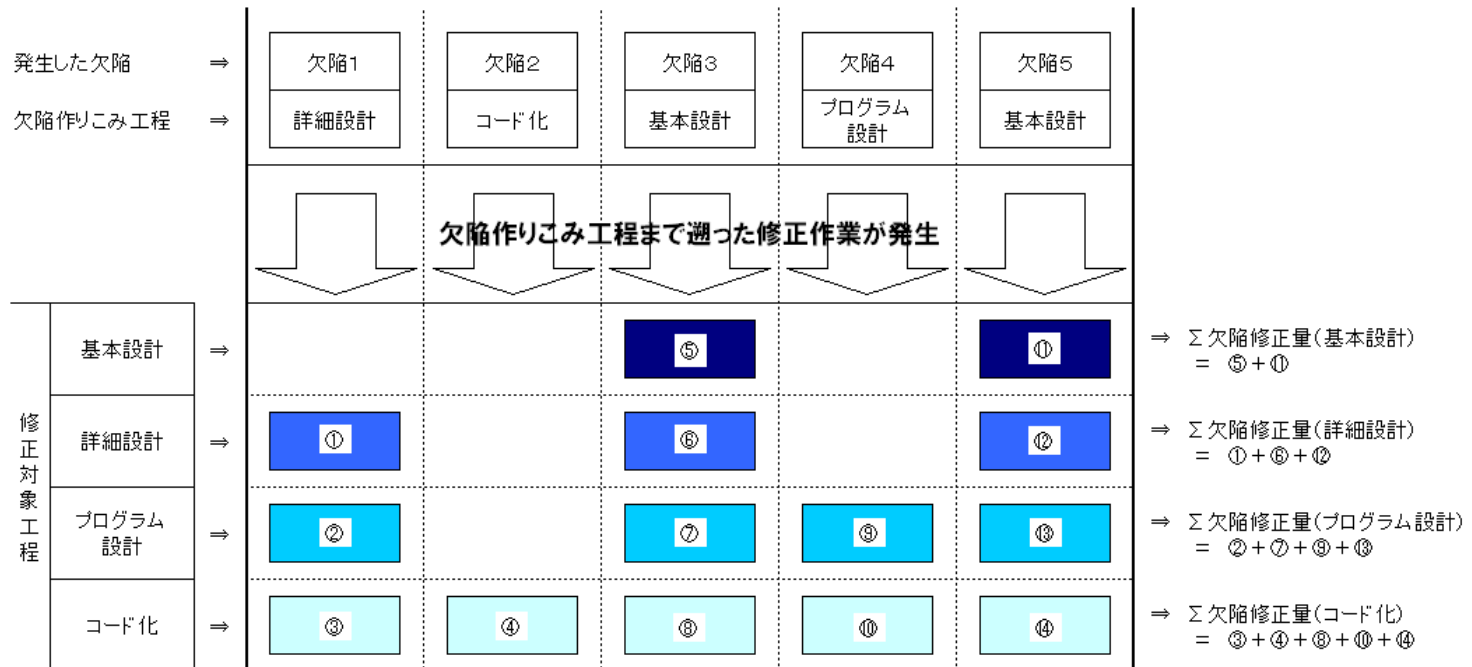
■ 欠陥修正に関わる工数の把握

- 欠陥修正量は、作り込んだ工程での修正量と先行する工程の欠陥修正によって修正される量の総和(下図参照)

□ 欠陥修正工数 I = Σ 欠陥修正量 I × 欠陥修正生産性 i

- さらに、欠陥修正量を入力として再テスト工数を算出

□ 再テスト工数 I = Σ 欠陥修正量 I × 再テスト生産性 i



第2部① 日本ユニシスの事例

～品質強化への取り組みと留意点～

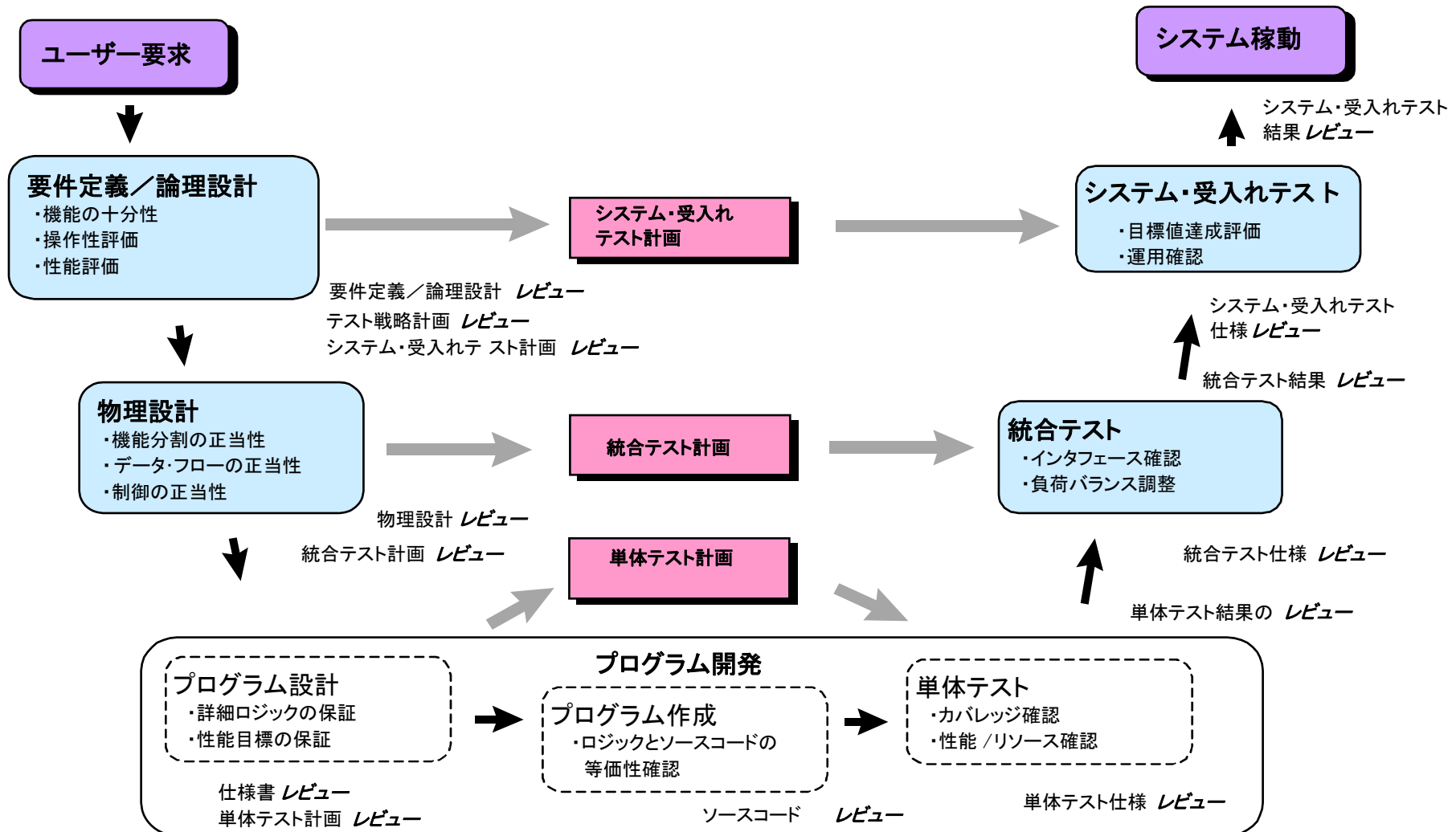
■ 概要

- 日本ユニシス社では、「VモデルとV&V(検証および妥当性確認)」を基本の考えとし、大きく設計のフェーズとテストのフェーズに分けて、遵守すべき手順と管理指標を設定。
- また、更なる品質強化へ向け、改めてテスト技術にフォーカスするとともに、Wモデル開発の推進に取り組んでいる。
- 本書では、開発工程の大きな割合を占めコストに大きく影響するテストに焦点をあて、見積り、及びテスト進行中の品質制御と再テスト見積りに関し、留意している事項について紹介。

第2部① 日本ユニシスの事例

～品質強化への取り組みと留意点～

標準的な開発モデル：VモデルとV&V（検証および妥当性確認）



第2部② 日立製作所の事例

～品質マップによるプログラムの早期品質確保～

■ 概要

- 日立社における品質指標を活用したテスト進行中における、品質管理と再テスト見積りの事例、特に、プロジェクト自身の実績データを活用した品質マップによるプログラムの早期品質確保の事例を紹介。
- 品質マップとは、縦列に現象を、横列に原因などでマトリクスで表示して、不良摘出の実績データを記入したもの。
- 本事例では、品質マップによる不良の収束性を分析することにより、追加テストの方針を導き出すことができた。

第2部② 日立製作所の事例

～品質マップによるプログラムの早期品質確保～

品質マップによる分析とその評価結果の例

原因 現象	規約		定義・設定			編集			SQL 出力処理					カウンタ操作 処理抜け			判定			仕様誤り		
	C ₁	C ₂	T ₁	T ₂	T ₃	E ₄	E ₅	E ₆	K ₁	K ₂	K ₃	K ₄	K ₅	K ₆	K ₇	K ₈	K ₉	K ₁₀	K ₁₁	S ₁		
画面表示形式不正	6	2	16	2					3													
画面項目出力不正	2		5						2				3			1						
表示不正	4		3				2	2	3	3						3						
結果不正	1		1																			
エラーチェック不正		1	1						1													
画面遷移不正				1					1													
異常終了													3									
操作不能																						

不良の傾向	評価結果
不良の偏り ・画面表示形式・項目出力不正 ・表示不正	画面形式や編集処理などの不良に偏っており、判定処理などのロジックに起因した不良が抽出されていない。(テスト項目の十分性を確認)
抽出不足 ・エラー処理不良が少ない	エラーチェック処理に関する抽出不足 結果不正の不良の抽出不足

第2部③ 東京海上日動システムズの事例

～テスト品質と障害発生状況の分析～

■ 概要

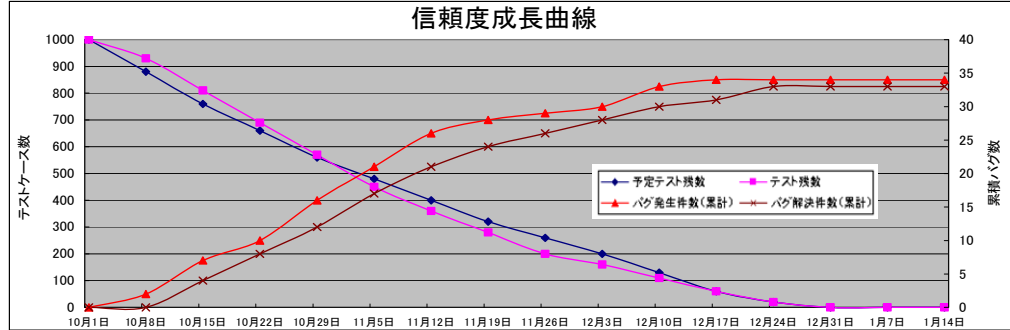
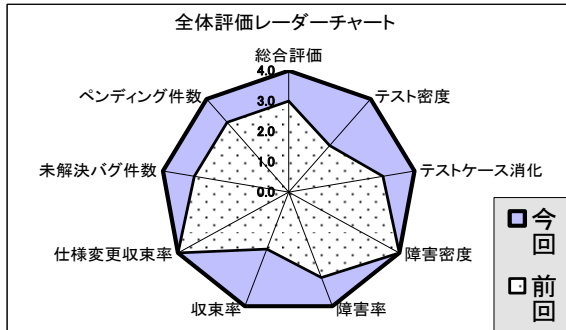
- ソフトウェアテストの品質とテスト工程の見積りの関係について、社内プロジェクトの実績をもとに、ソフトウェアテストの品質と生産性や本番での障害発生状況に関する分析結果を紹介。（「品質評価報告書」の導入）
- テスト工程の見積りのブレを早期に是正する仕組みとして、テスト工程のなかで中間評価を行い、品質を低下させる要因の芽を早期に発見し、対応要員・期間の確保などの調整を行うこと、要件変更・ペンディング状況などの残作業も明確にし、開発チーム内外と調整を行う取り組みを紹介。

第2部③ 東京海上日動システムズの事例 ～テスト品質と障害発生状況の分析～

品質評価報告書の例

タイトル

会社名		案件名	ああああああああ	規模(Kstep)	25.0	作成日	
所属名		本番予定日	2008年2月1日	工数(人月)	25.0	開発主担当	フルネーム
案件No	XXXXXXXX	子案件No	XXXXXXXXXXYY	発注No	ZZZZZZ	開発主担当TEL	(Tel: 1234)
全体評価	信頼度成長曲線						



項番	評価指標	評価点		指標	実績値				見解
		前回 中間品質評価 X月X日実施	今回 最終品質評価 X月X日実施		ST	OT	合計	残数 / 総数 / 残割合	
1	テスト密度	2	4	xx~xx件/Kstep (システム毎に設定)	ST: 34.0 OT: 6.0 合計: 40.0	(件/Kstep) (件/Kstep) (件/Kstep)			当初のケースの挙げ方では機能による偏りがあったため、中間評価時に見直しを行った。その結果、テスト密度は標準実績値より高くなったが、十分なテストが出来たと評価する。
2	テストケース消化	3	4	中間: 残数 / 総数 (残40%) 最終: 残数 (0) / 総数	残数: 0 総数: 1000 残割合: 0%				中間評価時は想定より早いペースでテストケース消化していたが、障害検出率が低かった。テストケースの挙げ方に問題があると判断し、ケース見直しを実施し、ケースを追加した。残ケースは大量テスト、性能テスト、プレ本番環境でのリモコン対象画面の確認のXケースであり、現在最終確認中。X月X日に完了予定。
3	障害密度	4	4	xx~xx件/Kstep (システム毎に設定)	ST: 1.2 OT: 0.2 合計: 1.4	(件/Kstep) (件/Kstep) (件/Kstep)			平均値より少ないが、単純改訂が多いためであり問題ない判断する。 平均値より多いがテスト初期で環境不備によるものが多発したため、バグ数としては平均的。 平均値より多いのはXX機能の品質不足と判断し、品質向上施策を追加実施し改善。
4	障害率	3	4	xx~xx% (システム毎に設定)	ST: 3.5% OT: 2.7% 合計: 3.4%				中間評価時は障害率が1%と低かったが、テストの観点を見直しケースを追加してテストを実施した結果、標準的な数値となった。 障害発生率が低いためテストケースを見直したが、内容・範囲とも問題なし。単純改訂によるものと判断。
5	収束率	2	4	中間: 60~70% 最終: 100%	ST: 100.0% OT: 80.0% 合計: 97.1%	目標値: 30 目標値: 5			テストケースの消化に応じて障害検出も収束しており、問題はない。 X月X日の週前後に山があるが、テストケース内容の見直しを行ったためであり、問題ない状況と判断する。
6	仕様変更収束率	4	4	果積抽出件数 / 推定障害総数 (抽出目標値) × 100。問題抽出十分性評価	残数: 0 総数: 12 残割合: 0%				オーナテスト時に振舞のレイアウトについて、仕様変更が発生。全体への影響は少ないと判断し、対応を行い完了しており、問題はない。
7	未解決バグ件数	3	4	中間: 残数 / 総数 最終: 残数 (0) / 総数	残数: 1 総数: 34 残割合: 3%				未対応の障害はない。 未対応の障害は1件であるが、本番稼働はC/Oの2W後であり、対応は完了する見込。 画面項目の誤字があるが、社内利用であり次回対応でオーナで解消済み。
8	ペンディング件数	3	4	未解決懸案事項 / 総発生件数。C/Oへの影響評価	残数: 3 総数: 3 残割合: 33%				ペンディング事項は別紙記載のX件。いずれもC/O直後には影響なし。
9	総合評価	3.0	4.0						テスト内容、障害検出状況ともに問題ないと判断する。 テスト段階でXX機能に要件変更がいくつか発生し、その分障害の収束も遅れた。追加テストを実施し、問題ないレベルに達したと判断する。

(中間品質評価: ホスト品質評価) 4点: このままテストケースを消化し品質確保可能
(最終品質評価) 4点: 品質確保十分。懸念無し

3点: 品質確保可能だが若干の軌道修正が必要
3点: 品質確保したが調整要素有り

2点: 品質確保不安有り。追加テスト要
2点: 品質確保不安有り。追加テスト要

1点: 品質確保不十分。工程見直し要
1点: 品質確保不十分。やり直し

第2部④ ジャステックの事例

～ソフトウェアの生産管理に基づくテスト見積り～

■ 概要

- ジャステック社の見積モデルの基本アルゴリズムは、生産物量見積り方式および生産性見積り方式から成り立ち、さらにおのこの方式において開発環境の違いや品質要求の多寡による変動を吸収する「環境変数」と呼ぶパラメータを導入している。
- ソフトウェアのテストに焦点を当てて、テスト見積り、ソフトウェアテストの管理およびソフトウェアテスト効率の改善などに対するジャステック社の取り組みを紹介。

第2部④ ジャステックの事例

～ソフトウェアの生産管理に基づくテスト見積り～

環境特性変動要素評価表（生産性に影響する外部環境変数）の例

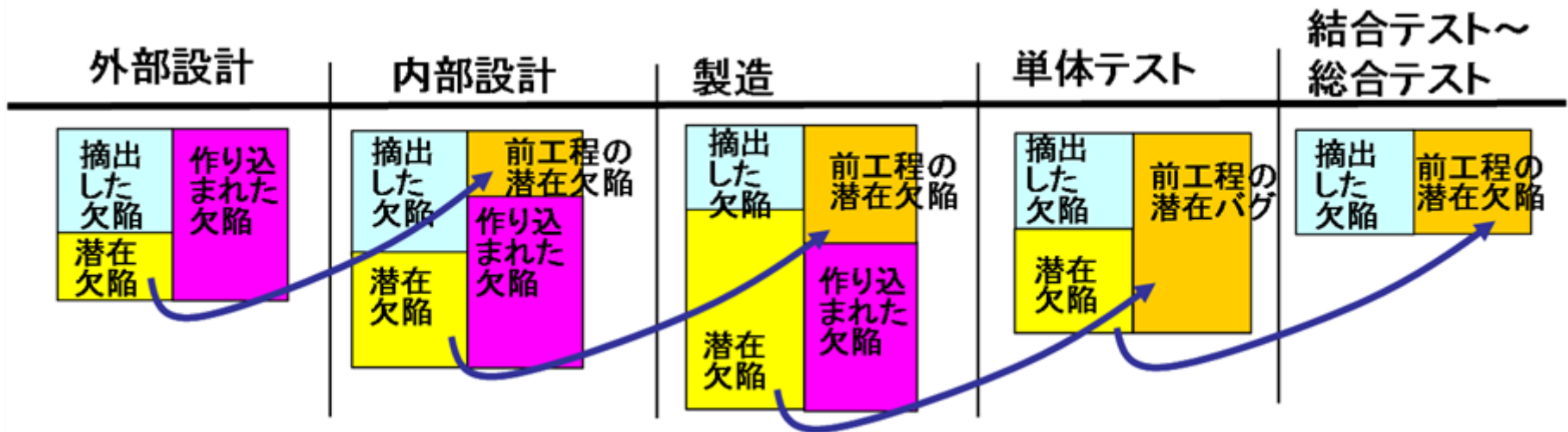
主特性	副特性	変動要素	ベースラインからの変動率(%)			
			要件定義	設計	製作	テスト
業務特性	業務ナレッジ	顧客の開発対象業務に対する業務ナレッジが生産性に及ぼす影響	-10 ～ 50	-10 ～ 10	—	-10 ～ 10
ハードウェア特性	安定度/信頼度/使用度	システムもしくは製品となるハードウェアの安定度・信頼度	—	-5 ～ 5	—	-5 ～ 5
ソフトウェア特性	安定度/信頼度/使用度	システム/製品に組み込む他社作成ソフトウェアもしくはCOTSの安定度・信頼度	—	-5 ～ 5	—	-5 ～ 5
コミュニケーション特性	顧客窓口特性	意思決定能力(期限遵守、決定事項の覆る度合)	-10 ～ 20	-10 ～ 10	—	-7 ～ 7
	工期の厳しさ	基準工期(月) = $2.7 \times (\text{人月})^{1/3}$ に対し ▲30%限度とした短期化度合	0 ～ 10	0 ～ 10	0 ～ 5	0 ～ 7
	コミュニケーション基盤	開発拠点分散、資料など情報共有、電子媒体・システム具備など物理的基盤充実度	-10 ～ 10	-5 ～ 5	-3 ～ 3	-3 ～ 3
	レビュー体制	無駄なレビュー(重複多段階など)の排除およびレビュー効率向上への工夫度合	-5 ～ 5	-5 ～ 5	-3 ～ 5	-5 ～ 5

■ 概要

- NEC社における品質指標をもとにした上流工程からの品質管理とソフトウェアテストの取り組みについて紹介。
- NEC社では上流工程からの品質管理としてソフトウェア品質会計制度と呼ぶ管理手法を採用し、上流工程からの欠陥管理、ソフトウェアテストの計画ならびに評価を行っている。
- 本書では、ソフトウェア品質会計制度を中心にその概要と上流工程、下流工程それぞれの工程における適用方法、適用効果について紹介。

第2部⑤ 日本電気(NEC)の事例 ～ソフトウェア品質会計制度の適用～

開発工程における欠陥の作り込みと抽出



ご清聴ありがとうございました。