

ソフトウェア品質保証 入門

高品質を実現する考え方とマネジメントの要点



安田 勝通



奈良隆正

紹介者：柏 祐太郎

本日のメニュー

1. ソフトウェア品質保証の考え方

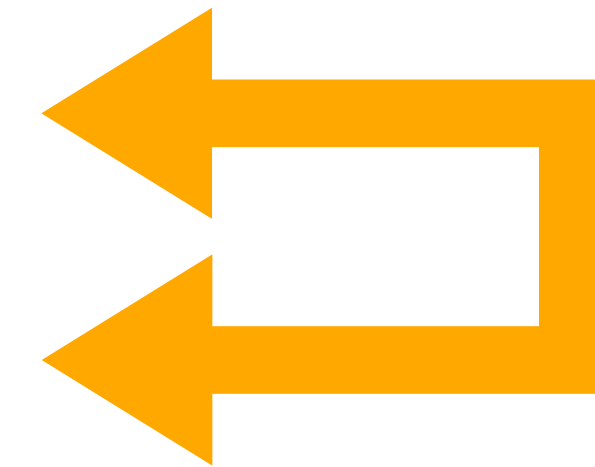
1. ソフトウェアの品質

2. ソフトウェア品質保証の問題点

3. 品質保証の考え方

4. ソフトウェア検査のあり方

5. ソフトウェア品質保証の組織と体系



順番入れ替え

»おさらい

品質とは、最終的にソフトウェアを**ユーザが使う時点**で、
ユーザのニーズを**どれだけ満足**しているか

(ただし、**ユーザが誰か**でその価値が変わる)

1.3 品質保証の考え方



品質保証の定義

- JIS, ISO, ANSI/IEEEでは,
『消費者が満足する品質を保証するための
組織的・体系的な活動』と定義
- 日本と欧米で品質保証の基本的な考え方は異なる
 - 欧米：文書化や証拠を示すことが必要
 - 日本：信用における製品を出荷してきた実績が必要
 - » 出荷した製品とそのアフターサービス, クレームやニーズのフィードバックの一連の活動を評価
 - » 本書の考え方は日本式

欧米の考え方

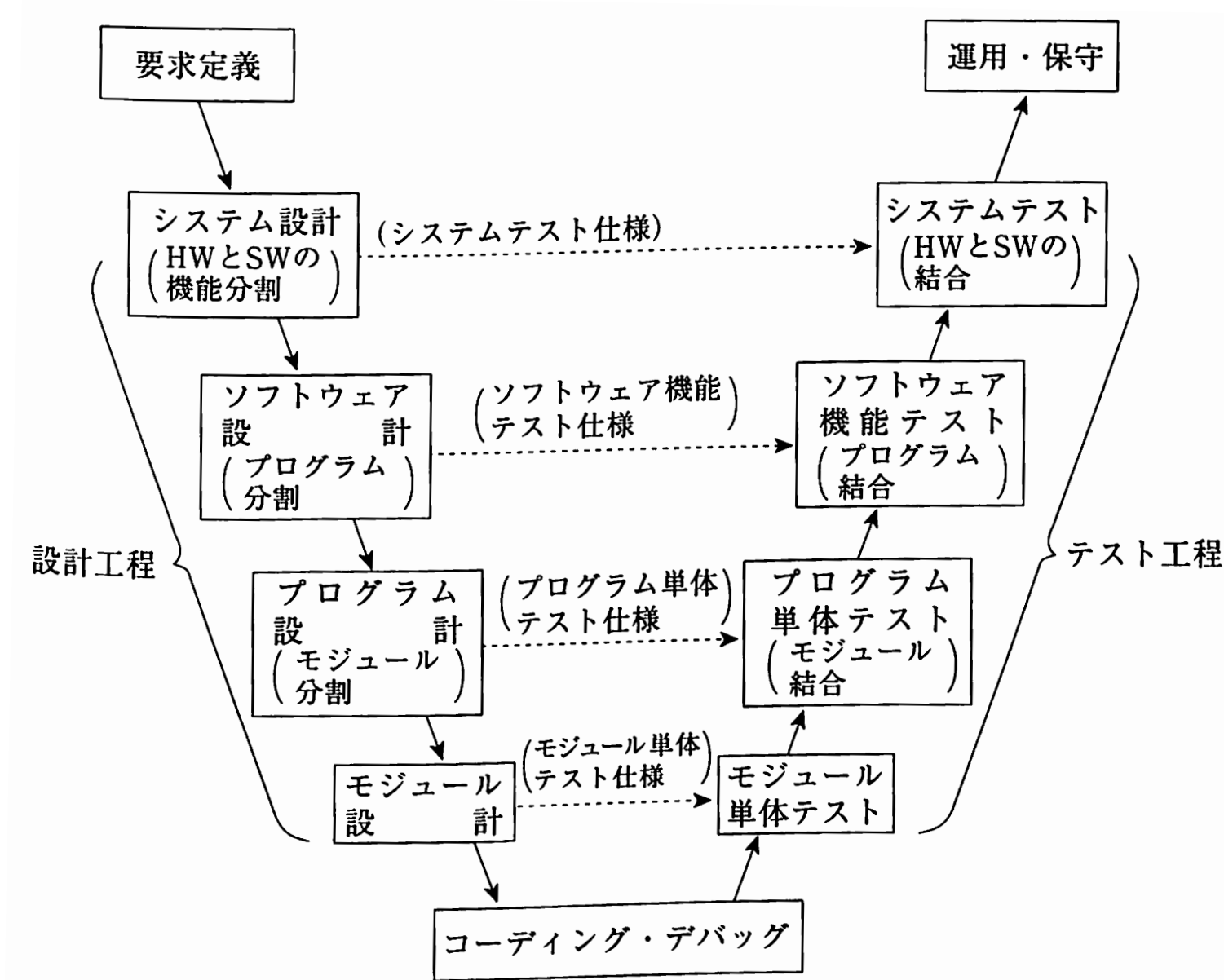
• Verification and Validation

- Verification (検証)

» 各工程が正しく実行されていることをレビューやテストで確認

- Validation (妥当性の確認)

» 最終テスト工程でユーザの要求定義をソフトウェアが満たしているかを確認



日本的品質管理

1. 全員参加

- » QC部門だけでなく，間接部門や管理部門も
- » トップから担当まで

2. 品質第一

- » 短期利益優先ではだめ
- » 不良手直しの減少→生産性向上

3. 消費者志向（マーケットイン）

- » メーカーは製品をお客様にいいはずと押しつけはだめ
- » お客様の欲するものを把握して作る

4. 事実・データで物を言う

- » 統計的方法の活用

5. 次工程はお客様

- » 自分のやる工程をキチンとやって次に回す

品質管理の進め方

- 生きた標準化
 - 品質のバラツキを抑える
 - 規格は作った以上は、徹底的に遵守させるとが必要
 - 半年に1度程度で改善（不都合は必ず出てくる）
- 応急対策と再発防止
 - プログラムの不具合修正は、当該不良原因の除去だが、根本原因の除去ではない
⇒不良の作り込み原因や検出できなかった原因を取り除く必要
- プロセスの管理
 - テストや検査のみに頼らず、設計プロセスでの品質改善が大事

日本と欧米の品質管理の違い

- 欧米：「魅力的品質」重視
 - 新しいコンセプトや独自性の強い機能が欲しい
 - » ソフトウェアを研究開発品と捉える
 - » 上流工程の要求分析や設計に重点
 - プロジェクト制度
 - » プロジェクトマネージャーが全権を握る
- 日本：「当たり前品質」重視
 - 不良をへらし，バラツキを抑える（信頼性）
 - » ソフトウェアを，ハードウェアと同様に工業製品と捉える
 - » 不良も時系列にきめ細かく集計⇒速やかな品質管理が可能
 - 工場制度
 - » 設計部門や検査部門や調達部門が分かれ，それぞれ権限を持つ（三権分立）

1.2 ソフトウェア品質保証の問題点



ソフトウェアの特質 1

- 論理の大規模集合体
 - 摩擦や腐食・劣化などの経年的変化を起こさない
 - 大規模論理集合体を正確に組み上げる技術を確立できていない
 - » 一見誰でも自由に、自己流でソフトウェアを作れるという誤解がある
 - » ハードウェアの場合は、物理学の法則に基づく力学や電磁学が必要なことは皆知っている

分割して統治せよ！！

- ・ 構造化技法（構造化分析・構造化設計・構造化プログラミング）の利用
- ・ 図式化設計記法（UMLなど）の利用
- ・ レビュー（デザインレビューやコードインスペクションなど）の実施
- ・ 系統的テスト（プログラムを分析してテスト）の実施

ソフトウェアの特質 2

- 目に見えないもの
 - 品質をどのように評価するか不確か
 - 管理がしにくい
 - » ソフトウェア開発プロセスやその問題点も見えない

可視化が重要！！

- プログラムの可視化
- 図式化設計記法の利用
- 工程進捗状況（テスト消化件数など）の定量的把握と可視化
- 品質管理状況（不良件数など）の定量的把握と可視化

- 多様なユーザニーズ
 - ユーザのニーズをいかに的確に把握し，仕様化するかが課題
 - 請負開発では，発注側でも要求仕様の取り纏めが困難
 - » ソフトウェアの大規模化
 - » エンドユーザや経営者の要求が厳しい

技術で顧客にも対応！！

- 要求分析・定義技術の確立と定着化
- プロトタイピング技術の確立と定着化
- 変更管理（どうしても発生してしまう仕様変更の対処）の徹底
- システムテスト（ユーザと同じ環境でのテスト）の実施

- 個人依存度大
 - 個人の能力差により生産性や品質が数倍異なる
 - 個人の能力を定量的に評価することが困難
 - » 特に外部委託では、委託先の開発要員の能力を適切な把握は難しい

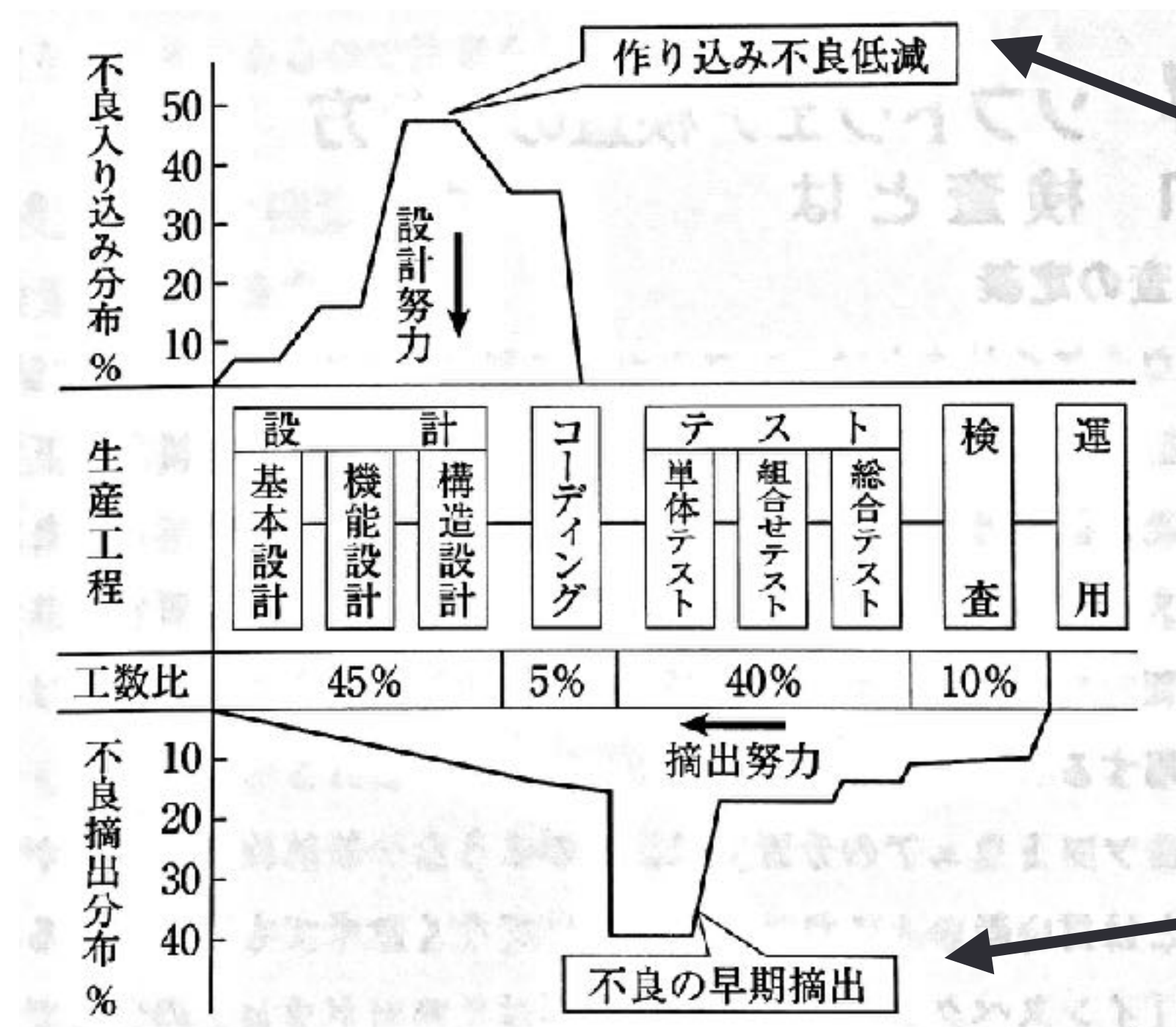
職人芸からの脱却！！

- ・ 開発の標準化の実施（標準開発手順の確立，部品化など再利用の促進）
- ・ 開発の自動化の実施（開発支援ツールの導入）
- ・ 個人能力の向上（基本動作の徹底，教育の強化，モラルの向上）

不良低減のための方針

14

- 不具合による手戻りは、効率と品質のボトルネック
 - コーディング量(100ステップ)に対して1～5件の不具合
 - 全工程の30～50%をテストに費やしている
 - » 仮に不具合が0であったとしても、ある程度のテストは必要



現在のソフトウェア工学では、設計段階での不良の作込み0は困難

『特に大事！！』 by 著者

- ・デバッグの効率化
- ・見逃し不具合の減少⇒品質向上