

JaSST '19 Niigata S2) 事例紹介

エンタープライズシステムでの 品質向上プロセス

2019年7月19日(金) 14:55~15:40 (45分)



石橋 宏之 isibasi@jmacsoft.co.jp

製品品質モデル JIS X 25010:2013

2

	品質特性	概要	設計	実装
①	機能適合性 Functional suitability	要求を元に実装されていること	○	
②	性能効率性 Performance efficiency	実行時の性能や資源効率		○
③	互換性 Compatibility	他製品やシステムと機能や情報を共有・変換できること	○	
④	使用性 Usability	効果的・効率的に利用できること	○	
⑤	信頼性 Reliability	必要時に実行できること		○
⑥	セキュリティ Security	不正に悪用されずに情報やデータが保護されること	○	△
⑦	保守性 Maintainability	保守や修正がやりやすいこと		○
⑧	移植性 Portability	他のハードウェアや実行環境に移植する際にやりやすいこと		○

すべての特性の仕様化・測定は無理

- トレードオフ: 費用・納期の制約(QCD)
- 業態や案件の特性により重点を選択
 - 高品質・短納期 ➡ 高価格
 - 高品質・低価格 ➡ 長期?
 - 短納期・低価格 ➡ 低品質?
- 請負契約＝瑕疵担保責任
 - 改正民法「契約不適合担保責任」
 - 期限延長 1年→5年



品質特性の優先度

※ご自分のビジネスでの優先度を記入してみてください

4

	品質特性	受託開発	組込み	パッケージ	ゲーム	サーバー	ASP
①	機能適合性						
②	性能効率性						
③	互換性						
④	使用性						
⑤	信頼性						
⑥	セキュリティ						
⑦	保守性						
⑧	移植性						



品質を確保するための施策

施策	教育	レビュー・ インスペクション	テスト
目的	スキル向上	品質向上(欠陥除去)	品質保証
確認内容	要件・仕様通りに高品質な設計・実装ができること	要件・仕様通りに設計・実装されていること	要件・仕様通りに動作すること
例	うるう年の判定方法を正しく理解し、実装方法を身に付けていること	うるう年の判定処理が正しく実装されていること	うるう年かどうか正しく判定されること

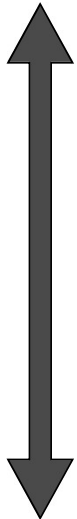
レビューの目的とメリット

- 早期に実施 ➡ 品質向上・コストダウン
 - 上流の欠陥を下流で修正 ➡ 50～200倍のコスト
 - 上流で検証 > 完了後に妥当性確認
 - ファースト成果物チェックでスキルと理解度を確認
 - 検出した問題を展開することで欠陥発生を防止
- 完成前にチェック可能
 - コーディング前にチェック
 - やり直しコストが低い
- 品質状態の可視化とプロセス改善
- 教育効果(OJT): フォロー効率を意識(引継ぎ)

レビュー手法と作業

7

コスト高
効果大



コスト低
効果小

手法	タスク					目的
	計画	準備	実施	記録	検証	
インスペクション	○	○	○	○	○	重要文書・機能、プロセス改善
チームレビュー	○	○	○	○	×	共通部品の仕様検証
ウォークスルー	○	×	○	可能	×	技術説明
ペアプログラミング	可能	×	継続	×	○	難易度の高いバグ修正、教育
ピアデスクチェック パスアラウンド	×	○	可能	可能	×	書式・誤字・体裁チェック ラフチェック
アドホックレビュー	×	×	○	×	×	中間レビュー
デスクチェック	×	×	×	×	×	ラフチェック

機能要件

	工程	内容	記載すること
①	要件定義	Why なぜ作るか	要件内容
②	基本設計	What 何を作るか	画面帳票設計、DB仕様
③	詳細設計	How どうやって作るか	実装ルール、モジュール分割

例：消費税増税

	現行	2019年10月1日(軽減税率制度実施)	
		軽減税率	標準税率
消費税率	6.3%	6.24%	7.8%
地方消費税率	1.7%	1.76%	2.2%
合計	8.0%	8.0%	10.0%

非機能要件

9

社会的影響が殆ど無いシステム	社会的影響が限定されるシステム	社会的影響が極めて大きいシステム
企業の特定期間が比較的限られた範囲で利用しているシステムで、機能が低下または利用不可な状態になった場合、利用部門では大きな影響はあるが、その他には影響しないもの。 ここでは、ごく小規模のインターネット公開システムを想定している。	企業活動の基盤となるシステムで、その機能が低下又は利用不可能な状態に陥った場合、当該企業活動に多大の影響を及ぼすと共に取引先や顧客等の外部利用者にも影響を及ぼすもの。 ここでは、企業内のネットワークに限定した基幹システムを想定している。	国民生活・社会経済活動の基盤となるシステムで、その機能が低下又は利用不可能な状態に陥った場合、国民生活・社会経済活動に多大な影響を与えるもの。ここでは不特定多数の人が利用するインフラシステムを想定している。

中項目	小項目	メトリクス(指標)	社会的影響が殆ど無いシステム		社会的影響が限定されるシステム		社会的影響が極めて大きいシステム	
			選択レベル	選択時の条件	選択レベル	選択時の条件	選択レベル	選択時の条件
性能目標値	オンラインレスポンス	通常時レスポンス順守率	順守率を定めない	トランザクションの量が少ない場合。または多い場合でもユーザに対する利用制限などが可能な場合を想定。	90%	管理対象とする処理の中で、通常時のトランザクション数の90%が目標値を達成できれば良いと想定。	99%以上	管理対象とする処理の中で、通常時のトランザクション数の99%が目標値を達成できれば良いと想定。

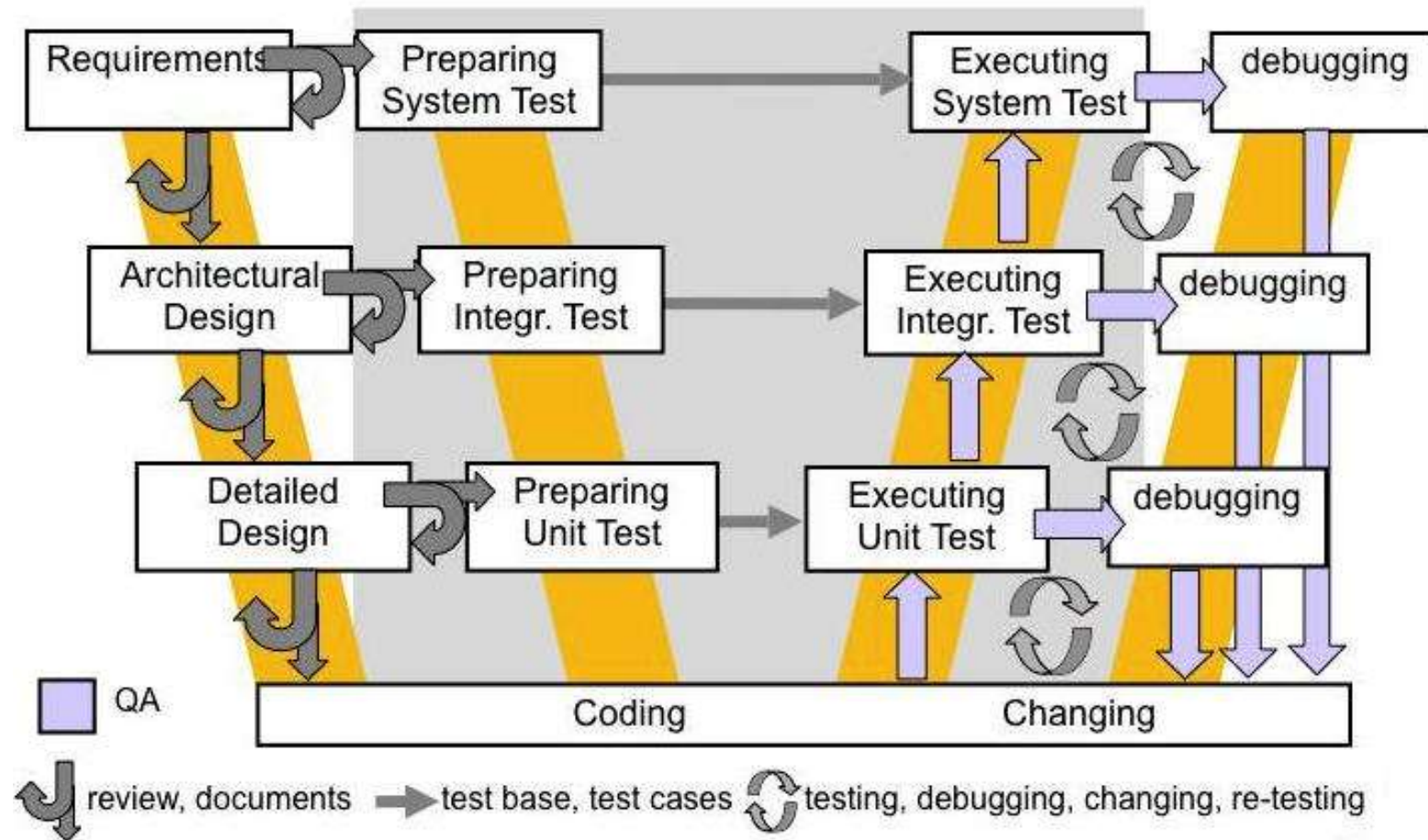
非機能要求グレード2018 システム基盤の非機能要求に関するグレード表 2018年4月



IPA「システム構築の上流工程強化(非機能要求グレード)」 <https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html>

システム開発モデル(Wモデル)

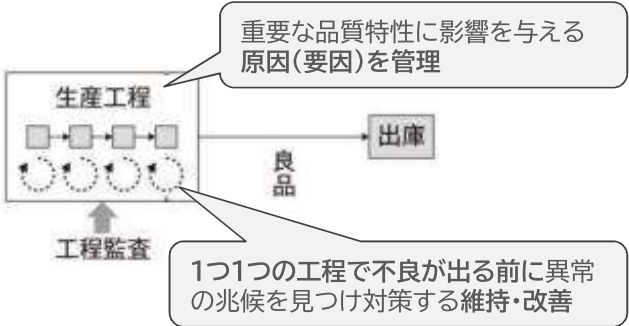
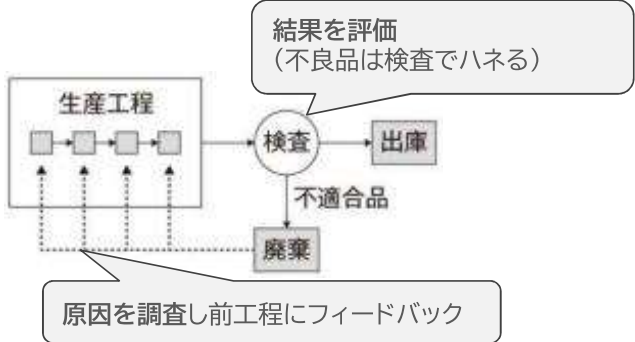
10



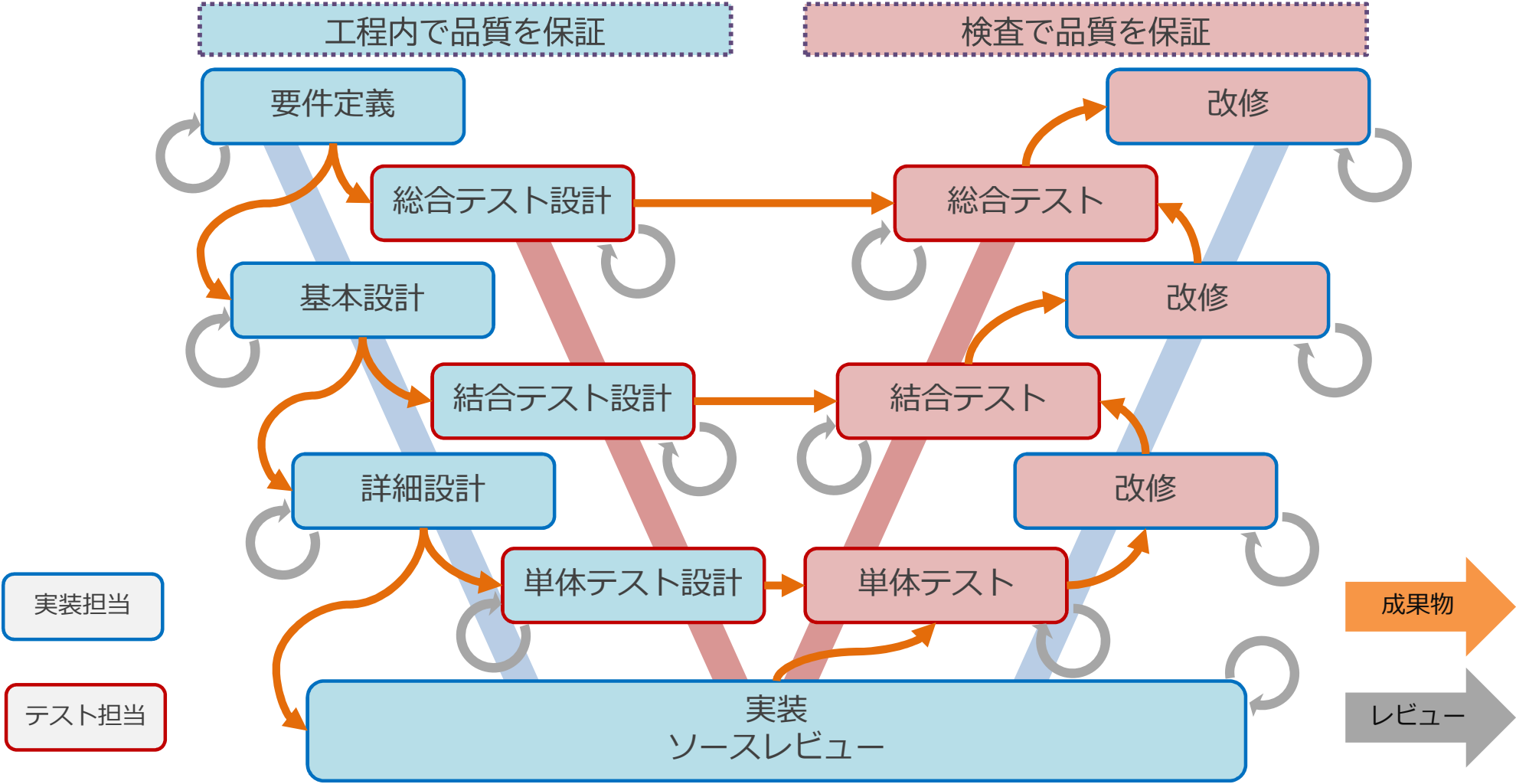
W-model - test process parallel to the development process JST2004 Dr. Andreas Spillner

自工程完結：各工程で品質向上

11

	工程内で品質を保証	検査で品質を保証
基本的考え方	不良品は作らない	不良品は出ることが前提
主な活動	【課題達成型/未然防止】 	【問題解決型/流出防止】 
メリット	- 課題が見える、自ら改善サイクルが回る(スパイラル・アップ) - 品質、生産性向上、コスト低減 - 作業する人の働きがいが出る(モチベーション・アップ)	- お客様へ渡す状態で検査できる - 初期品質評価に適している - 各工程で生じる「ばらつき」が累積された状態で検査できる - 運用が比較的容易
デメリット	管理すべき要因の抽出・特定に技術力が必要であり、運用が難しい。	- モチベーションが下がる(検査員がフィードバックすると現場がイヤな思いをする、検査員も嫌がられるのでフィードバックしたくなくなる) - 検査コスト(設備・人)がかかる - 耐久品質(経年劣化)は検査困難

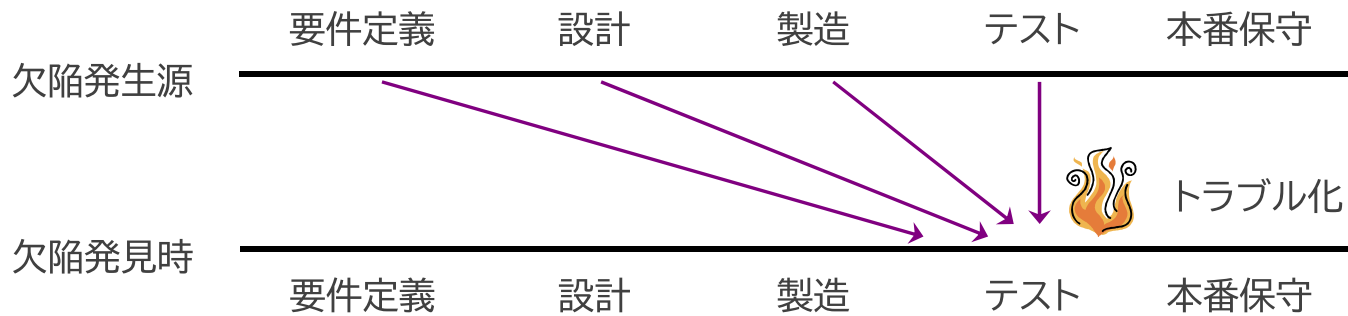
Wモデルでの品質保証



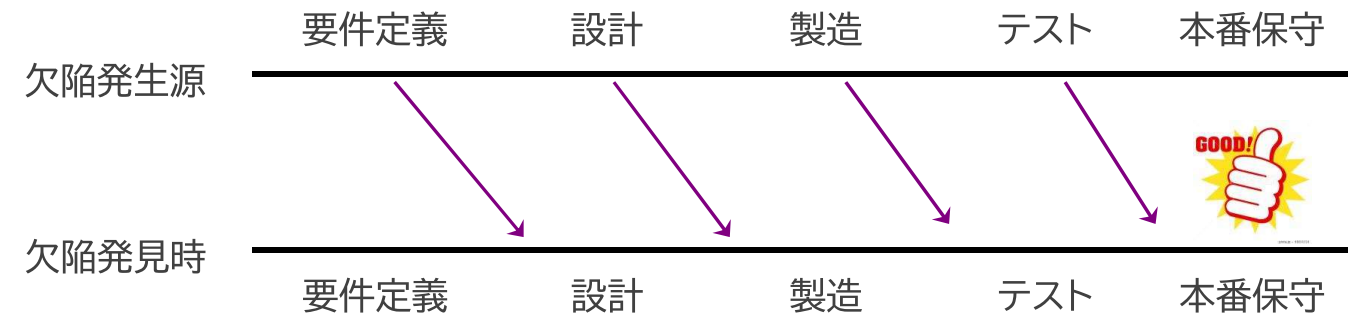
発生した問題は発生フェーズで

13

❖各工程での検証を省略した場合



❖各工程で検証を実施した場合



自工程完結のポイント

14

1. 「目的・ゴール」をはっきりさせる
2. 「最終的なアウトプットイメージ」を明確に描く
3. 「プロセス／手順」をしっかりと考え、書き出す
4. 次の「プロセス／手順」に進んでよいかを判断する基準を決める
(クライテリアによる工程移行判定)
5. 正しい結果を導き出すために「必要なもの」を抜け・漏れなく出す
6. 仕事を振り返り、得られた知見を伝承する

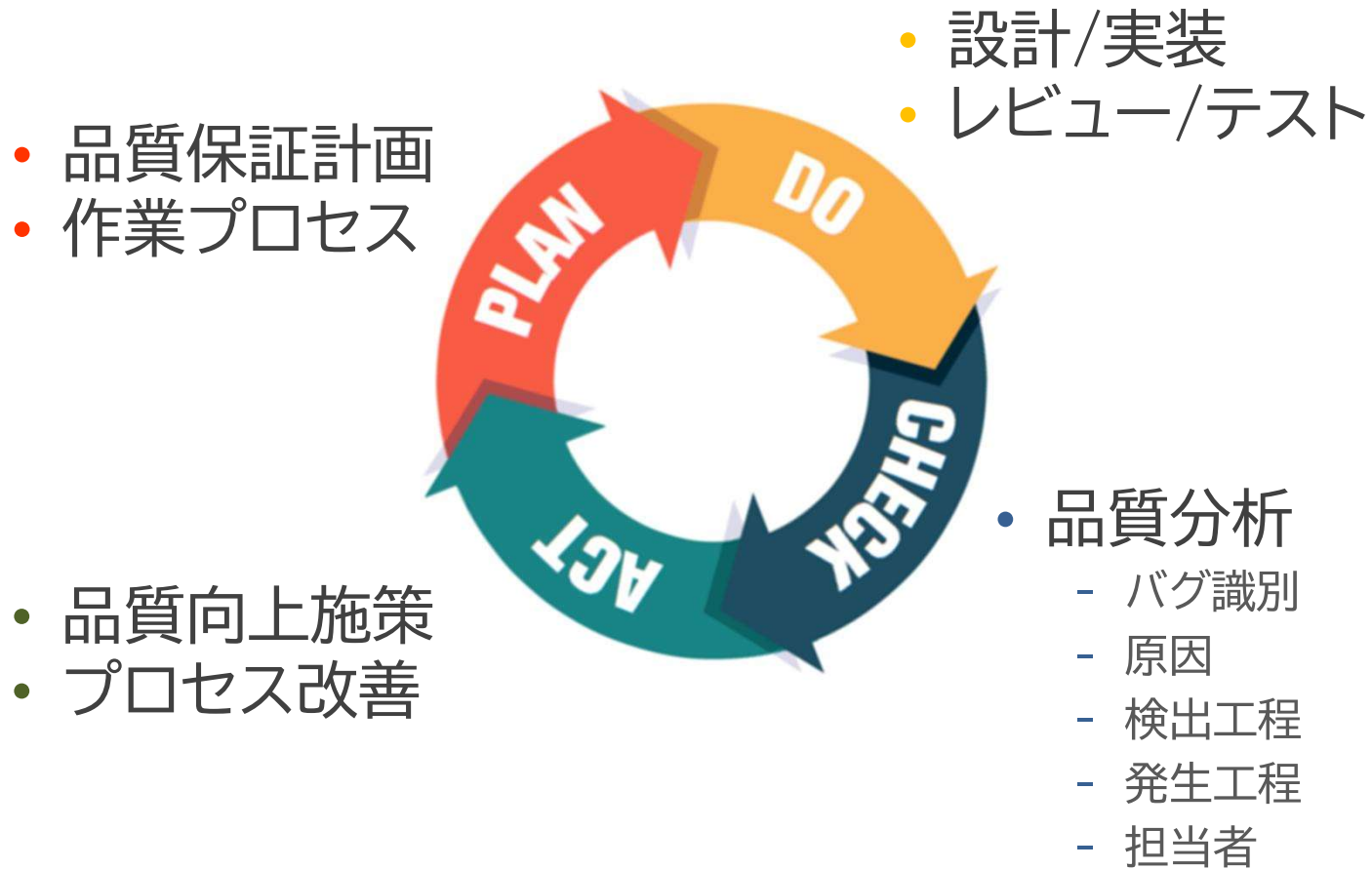
クライテリアによる工程移行判定(例)

15

次工程移行 条件	条件内容	判断の根拠	判定
条件1	基本設計書と詳細設計書が対応づけられていること(施策1) 施策1)基本設計書で示したカスタマイズ仕様について、漏れなく盛り込まれていることを確認する。	新旧機能一覧にて、現行機能踏襲/一部機能改修/新規機能を明確にしてレビューを実施した。	○
条件2	基本設計書の内容がすべて反映されていること(施策1、2) 施策1)基本設計書で示したカスタマイズ仕様について、漏れなく盛り込まれていることを確認する。 施策2)内部レビュー前に、設計者自身がセルフレビューを行う。	レビュー実施時に基本設計書を参照して漏れが無いことを確認した。	○
条件3	レビューでの指摘事項がすべて対処済みであること(施策3) 施策3)レビューについては、要件定義書で示された業務内容と突合し、抜け漏れがないことを確認して品質を担保する。 レビューのインプットとして、「新旧機能一覧」、「基本設計書」、および必要に応じて「現行システムのソースプログラム」を使用する。	レビュー記録表での指摘事項はすべて対応をおこなったことを確認した。 問題課題の残がないことを確認した。	○
条件4	レビューでの指摘事項は、必要に応じて横展開を行い対処済みであること(施策4) 施策4)レビュー指摘事項は、必要に応じて横展開を行い、効果(横展開で検出したバグ数)を記録する。	指摘事項は、横展開事項の判断を行い、必要に応じて横展開を実施し、対処を確認した。	○
条件5	バグ数が目標値内であること。目標値外の場合は明確な理由があること(施策5) 施策5)レビューの結果、バグ目標値範囲内の指摘事項を摘出する。	バグ数は目標値を下回った。目標87件に対して67件(77%)。目標範囲は69～106件であった。 現行機能からの移行がほとんどであり、作業手順を作成して進めたため、品質維持できたものとする。	△

レビュー/テストでのPDCA

16



品質保証施策(例)

17

詳細設計書

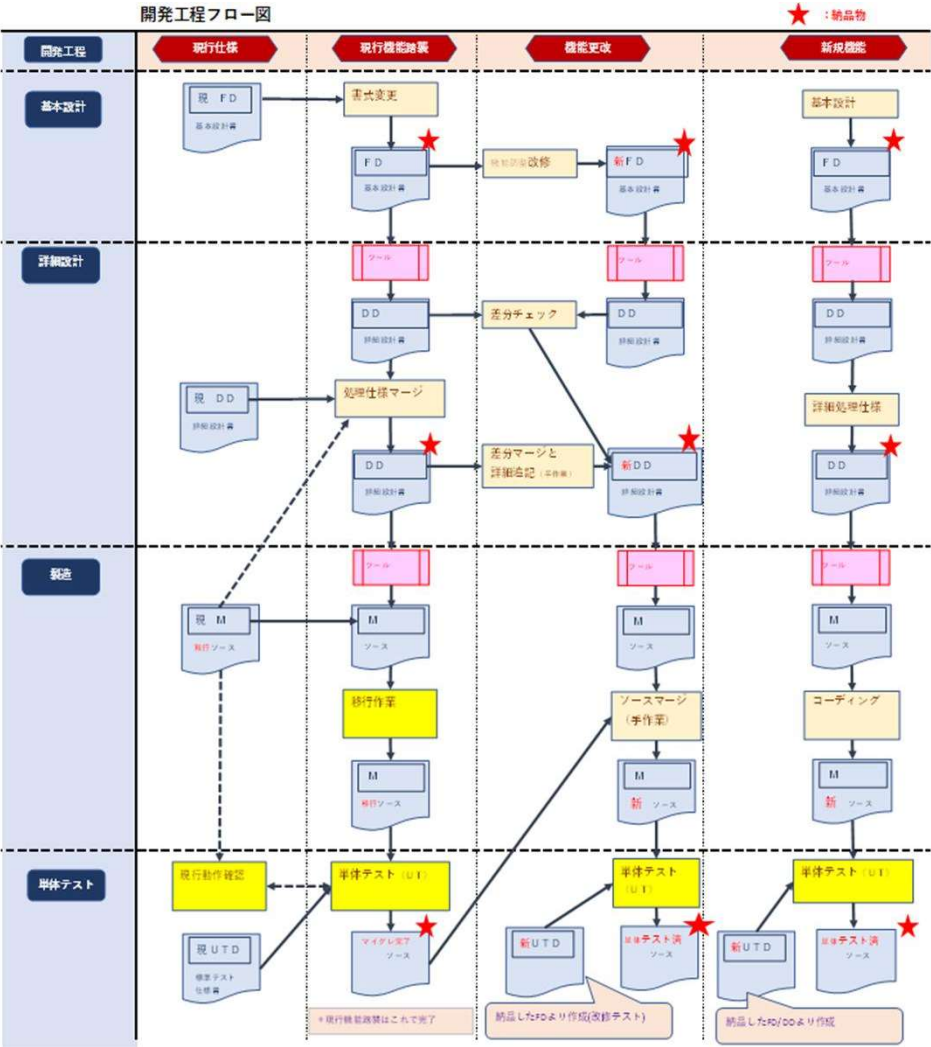
- 今回採用する移行プロセスでは詳細設計はプログラム品質に影響しないため、作業品質の確保を重点目標とする
- ファースト成果物チェックで OK であれば、同じ担当者の成果物はサンプリングチェックのみ
- ファースト成果物チェックで NG の場合
 - ファースト成果物を改修して XXX 殿で指摘が改訂されていることを確認
 - ジェイマックスソフトで改修後のファースト成果物をリチェック
 - 同じ担当者のセカンド成果物をジェイマックスソフトでチェック
 - OK であれば同じ担当者の成果物はサンプリングチェックのみ
 - NG の場合、XXX 側で原因分析と再発防止策を実施
 - 3 件以降も指摘がある場合は担当者変更等の抜本策を XXX 殿で検討する

プログラムソース

- XXX 殿で CDI を実施しており、ジェイマックスソフト側での全件 CDI は 過剰品質 となる
- 品質を確保するために CDI レビューチェックリストを作成する(XXX 殿・JMAC 調整)
- ファースト成果物チェックは XXX 殿製造担当者全員に対して実施する
- セカンド成果物チェックの進め方は詳細設計と同様
- 品質レベルが低い場合の対応は詳細設計と同様
- 移行作業エビデンスチェックも CDI と同様に実施する

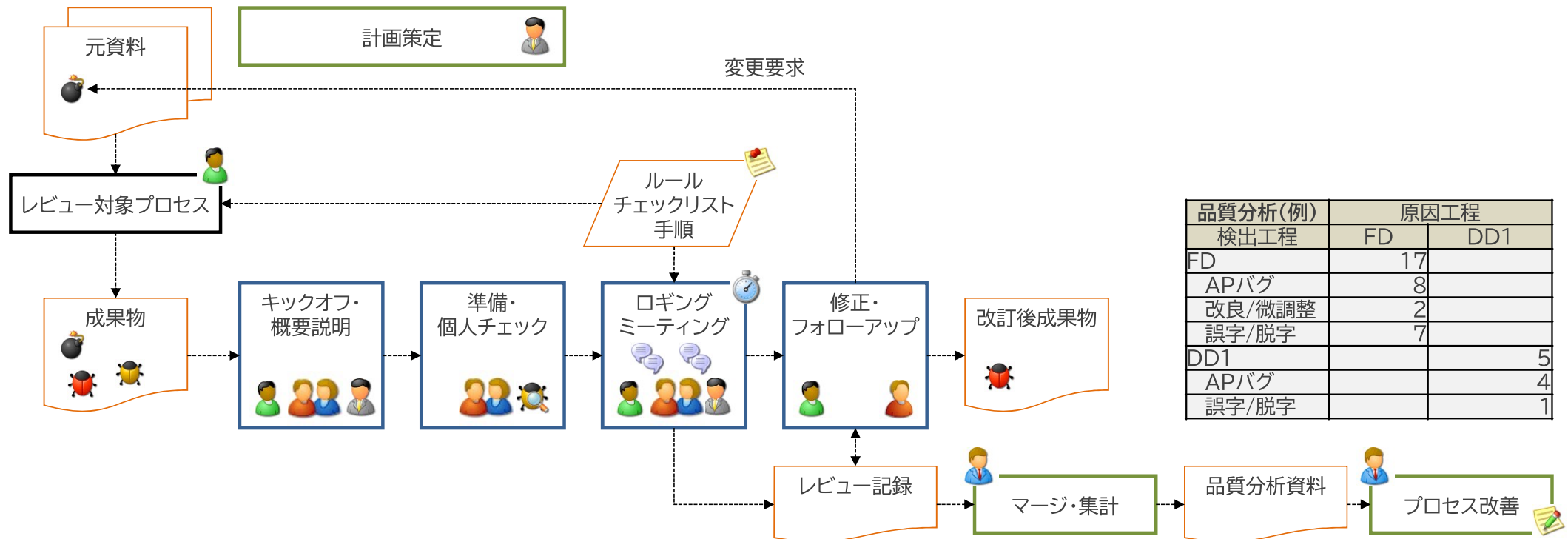
作業プロセス(例)

18



レビュー手順と品質情報収集

19



【レビュー記録項目(例)】※テーラリングあり

管理番号

指摘情報: 検出工程/指摘日/指摘者/レビュー対象/版/指摘箇所/指摘内容

対応情報: 対応内容/対応者/対応予定日/対応日

検証: 対応結果確認日/確認者

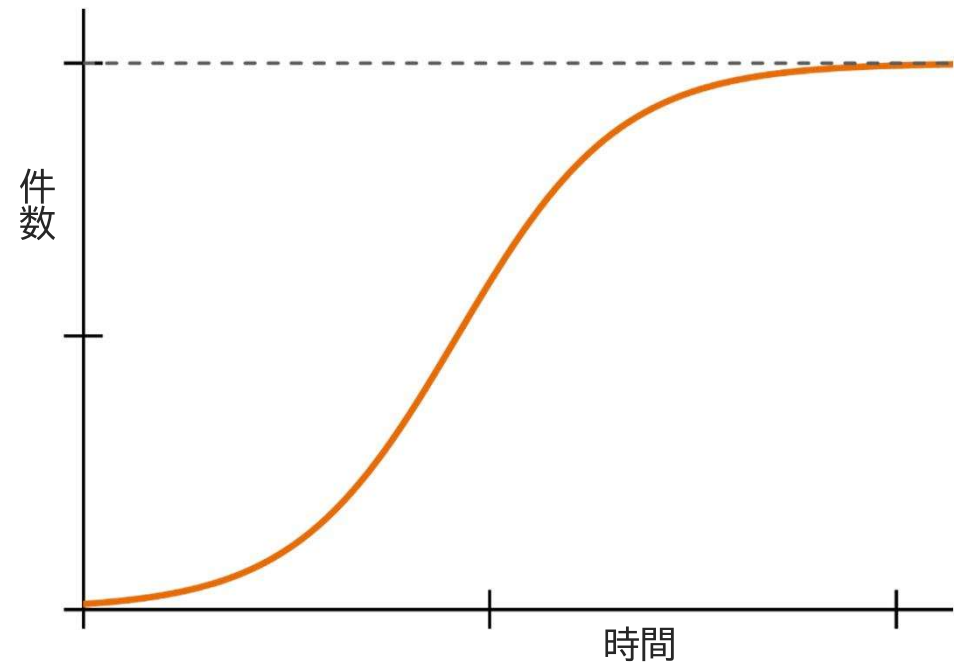
分析: バグ識別/原因区分/原因工程

横展開: 要否/対応完了日

ソフトウェア信頼度成長曲線

20

- 勝手にバグは収束しない
- 収束するのは理由がある
 - バグ内容を分析して対策実施
 - 初期バグの解消
 - 同件バグの事前対応
 - 習熟効果
 - 機能要件
 - 非機能要件
- 拡散する場合は？
 - 炎上/デスマーチ



タイミングによる品質分析・改善施策

21

タイミング	品質分析の視点	分析手法	改善施策
初期分析 ・初回成果物 ・進捗10%	①体裁・粒度のチェック ②標準化・規約のチェック ③品質管理ルールの徹底	・内容分析	スキル/理解不足⇒教育 手順自体の誤り⇒完了分への展開 環境問題⇒環境改善
中間分析 ・進捗30% ・進捗50% ・進捗70%	①初期レビューでの指摘が再発していないか ②バグ・指摘の傾向分析で原因推測・対策 ③実施した対策は次の分析での改善を確認	・予実分析 ・傾向分析 ・ゾーン分析 ・収束度分析 ・原因分析	原因＝前工程⇒再レビュー 原因＝現工程⇒トレーサビリティ確認 実施した改善の効果を測定
最終分析 ・進捗100%	①現工程での品質目標を達成していること ②品質向上のために実施した施策の有効性 ③バグが収束していなければ追加対策を実施	・予実分析 ・傾向分析 ・ゾーン分析 ・収束度分析	横展開と対応のリソース

傾向分析

- 検出したバグの傾向により対策を実施
- 発生する問題を想定して、改善策を事前に検討し、原因区分を決める

原因区分	改善策
仕様間違い	前工程の成果物を再レビュー
仕様漏れ	トレーサビリティプロセスの徹底
仕様理解不足	仕様説明を実施
共通部分の間違い	共通部分のレビュー確認
共通部分理解不足	共通部分の説明

- 傾向分析例

データの個数 / 指摘日	※原因区分1						
※バグ識別	コーディング誤り	既存バグ	設計(記述)不足	コーディング不足	設計(記述)誤り	その他	総計
APバグ	6	4	5	4	5	0	24
改良/微調整	12	0	1	0	0	0	13
誤字/脱字	2	1	0	0	2	0	5
仕様通り	0	0	0	0	0	1	1
重複バグ(カウント外)	0	0	2	0	0	0	2
総計	20	5	8	4	7	1	45

- 成果物品質だけではなく、作業品質にも注意(例：コピー後の訂正漏れ)

傾向分析(例)

23

- 仮説を検証するため分析/分析だけで対策は出ない
- 仮説「横展開・習熟によりバグは減少している」
- 分析「担当者・週別に検出したバグをカウント」
- 累積では改善効果が分からないため、期間で区切って変化を確認

The screenshot shows a spreadsheet with columns for bug identification, detection stage, and dates. A pivot table is being configured on the right side of the screen.

対応者	2018/5/7 - 2018/5/13	2018/5/14 - 2018/5/20	2018/6/11 - 2018/6/17	2018/6/18 - 2018/6/23	総計
担当者	1	10	19		
検出工程	4	8	30		
検出日	4	6	10		
検出者	4	5	12		
ドキュメント名	2	2	15		
Ver/版	1	1	3		
指摘箇所			23		
総計	13	4	48	16	32

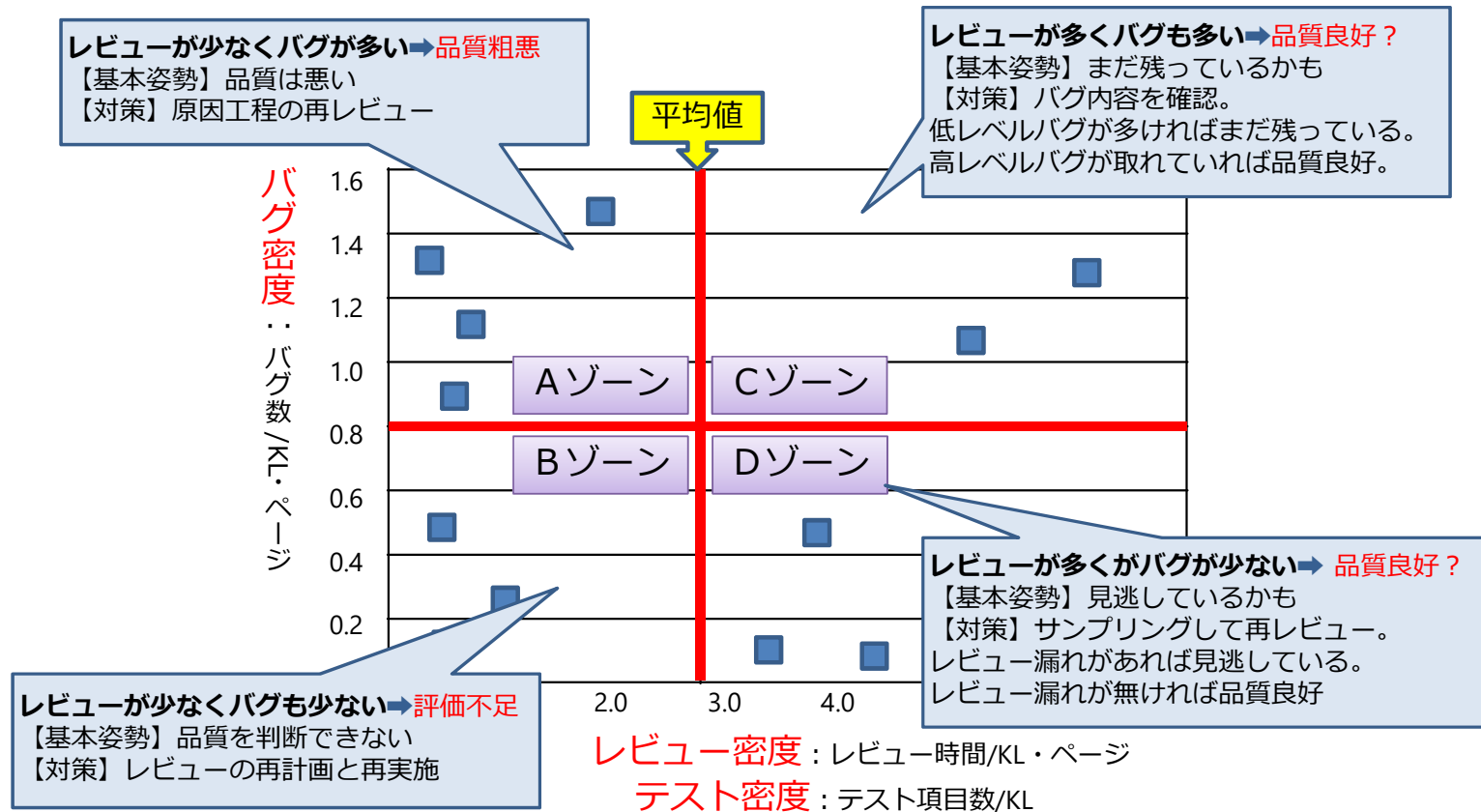
Pivot Table Configuration:

- レポートに追加するフィールドを選択してください:
- 検索:
- フィールド一覧:
- 次のボックス間でフィールドをドラッグしてください:
- フィルター: ※バグ識別, ※検出工程
- 列: 指摘日
- 行: 対応者
- 値: 個数 / 指摘...
- レイアウトの更新を保留する

ゾーン分析

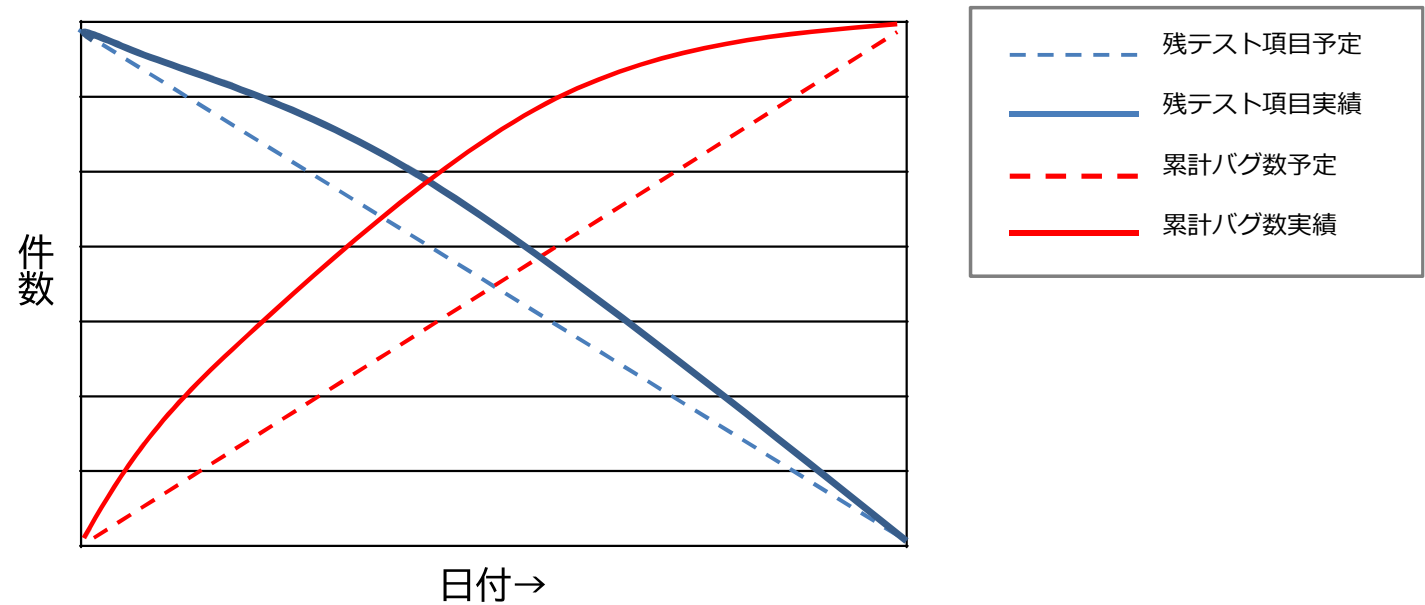
24

- 相対的な品質状況を把握する



収束分析

- 課題把握とバグ収束状況の評価



進捗	バグ数	品質状況	進捗状況	チェックポイント
先行	少	○	○	テスト項目が粗い？
先行	多	○	○	順調？
遅れ	少	○	×	進捗リカバリー
遅れ	多	×	×	要因分析と対策実施

現場に必要なのは「心理的安全」

26

仕事のタイプ	ルーチン業務	複雑な業務	イノベーション
例	組み立て工場	医療/システム開発	製品開発
手順	固定	混合(固定/一時的/新規)	指針のみ
目標	効率性/信頼性	品質/安全性	価値創造
組織学習目的	不断の改善	問題解決	革新/発見
失敗頻度	ほとんど発生しない	起こりうる	ひんぱん
重要ポイント	TQM/SPC/行灯	心理的安全 Psychological safety	プロセス/多様性
重点課題	管理/評価	率直/情報共有	集中探査/支援

システム開発(複雑な業務)を遂行するためには失敗共有が必要
そのために**心理的安全が重要**



失敗を許容して組織的に学習

- 高信頼性組織(High Reliability Organization)研究
 - 失敗が許されない組織でどうやって失敗をゼロに近づけるか
 - 航空機産業、病院、原子力発電所など
- 「**失敗を許容する文化**」を作ることが重要
 - 失敗が罰せられる環境では人々は失敗を隠す
 - 失敗こそ組織にとっては学習の源
 - 隠されると組織は成長の機会を失う
 - 失敗が許容されるから無責任にふるまうという意味ではない
- [重要]失敗の当事者だけでなく**組織全体で原因分析**
- 個人の責任にしないことが失敗を減らすことにつながる
- 組織全体が学習することで成長していく
- **コミュニケーション**➡**心理的安全**➡**失敗共有**➡**組織的学習**➡**高品質**

- 優先する品質特性はシステムによって違う
- レビュー対象によって手法を使い分ける
- 機能要件だけでなく非機能要件も確認
- 品質を作り込むために各工程でチェック
- プロセス改善のために手順を文書化
- 品質分析により改善施策を実施する
- 失敗を共有するために心理的安全性を保証
- 品質は結果ではなく「作り上げるもの」