



活動報告（2）第2回と第6回の活動報告「タイプ属性」 （ODC分析基礎編、応用編）

Rev.1.1

2020/10/02

瀬能 芳幸 (キヤノン株式会社)

日科技連ODC研究会

トピック

1. 第3期の振り返り
2. 基礎編・応用編
3. ワークショップの紹介

第3期の振り返り

第1回

(6/13)
基礎



第2回

(7/16)
タイプ



第3回

(8/12)
トリガー



第4回

(9/20)
インパクト・ソース



第5回

(11/14)
応用:プロセス



第6回

(12/18)
応用:タイプ分析



第7回

(1/27)
応用:トリガー分析



第8回

(2/25)

本日

(2020/10/1)
応用:手順と手法



欠陥タイプ属性

Defects type attribute

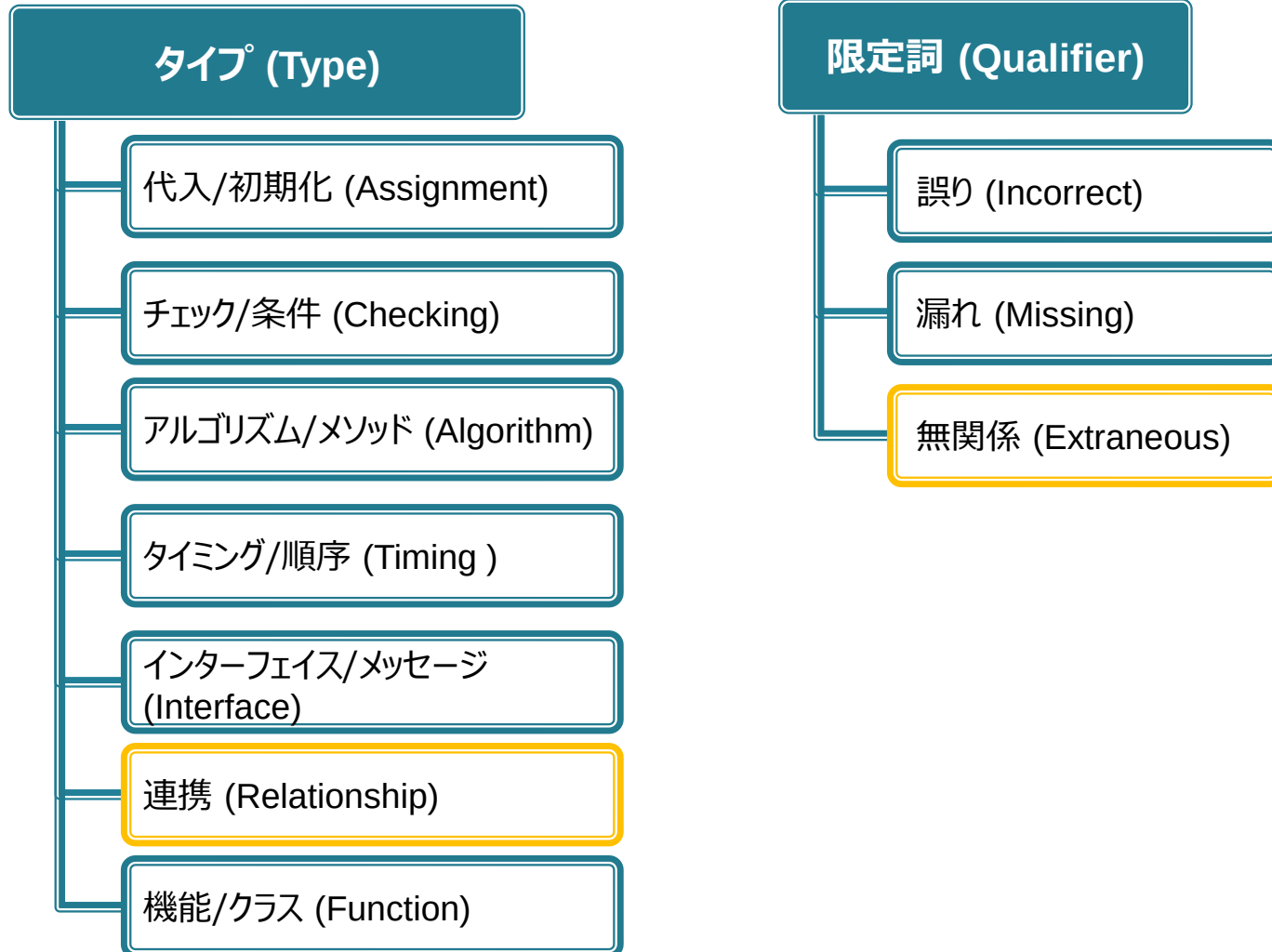
基礎編・応用編

Basic / Advance

- 欠陥タイプ
- 欠陥タイプは開発プロセス工程と関係する
- ODCはそもそもプロセス改革
- プロジェクト品質管理上、ODCをどう活用するか？

欠陥タイプ

欠陥タイプは、どんな欠陥か？



欠陥タイプは開発プロセス工程と関係する

欠陥タイプは、開発プロセスの各作業におけるミス

機能/クラス (Function/Class)の漏れ(Missing)

インターフェイス/メッセージ (Interface/O-O Messages)の漏れ(Missing)

タイミング/順序 (Timing/Serialization)の漏れ(Missing)

機能/クラス (Function/Class)の誤り(Incorrect)

インターフェイス/メッセージ (Interface/O-O Messages)の誤り(Incorrect)

タイミング/順序 (Timing/Serialization)の誤り(Incorrect)

アルゴリズム/メソッド (Algorithm/Method)の漏れ(Missing)

チェック/条件 (Checking)の漏れ(Missing)

アルゴリズム/メソッド (Algorithm/Method)の誤り(Incorrect)

代入/初期化 (Assignment/Initialization)の漏れ(Missing)

代入/初期化 (Assignment/Initialization)の誤り(Incorrect)

チェック/条件 (Checking)の誤り(Incorrect)

要件定義

基本設計

詳細設計

実装

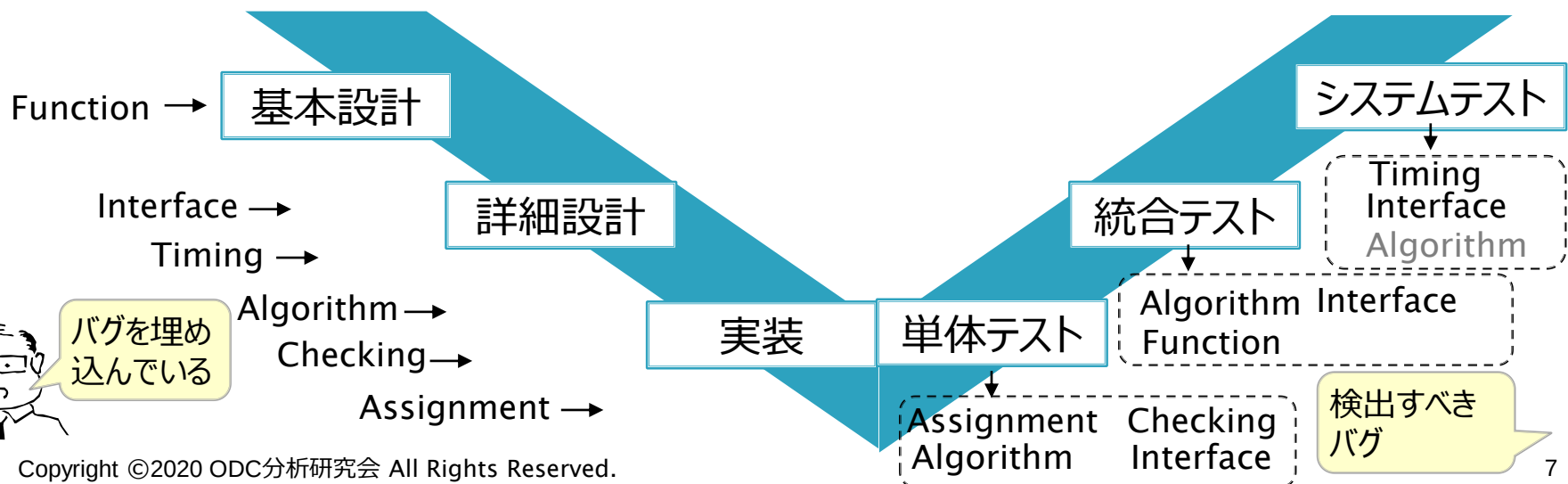
ODCはそもそもプロセス改革



2.6.1 タイプ属性と開発プロセス工程との関係

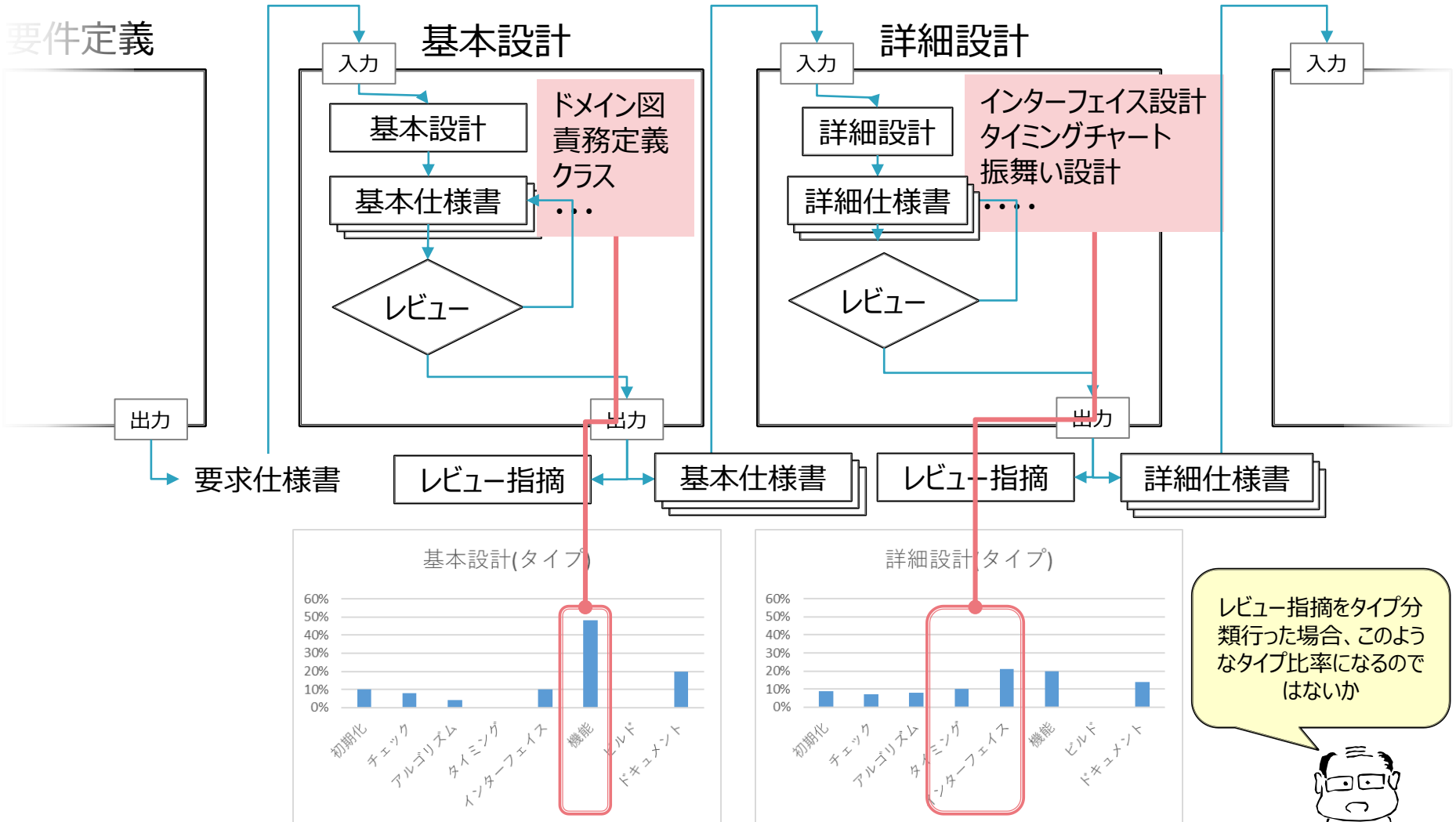
P44

	基本設計	詳細設計	コード	単体テスト	機能テスト 統合テスト	システム テスト
Assignment			✓	✓		
Checking		✓	✓	✓		
Algorithm			✓	✓	✓	✓
Timing		✓				✓
Interface		✓	✓	✓	✓	✓
Function	✓				✓	



ODCはそもそもプロセス改革

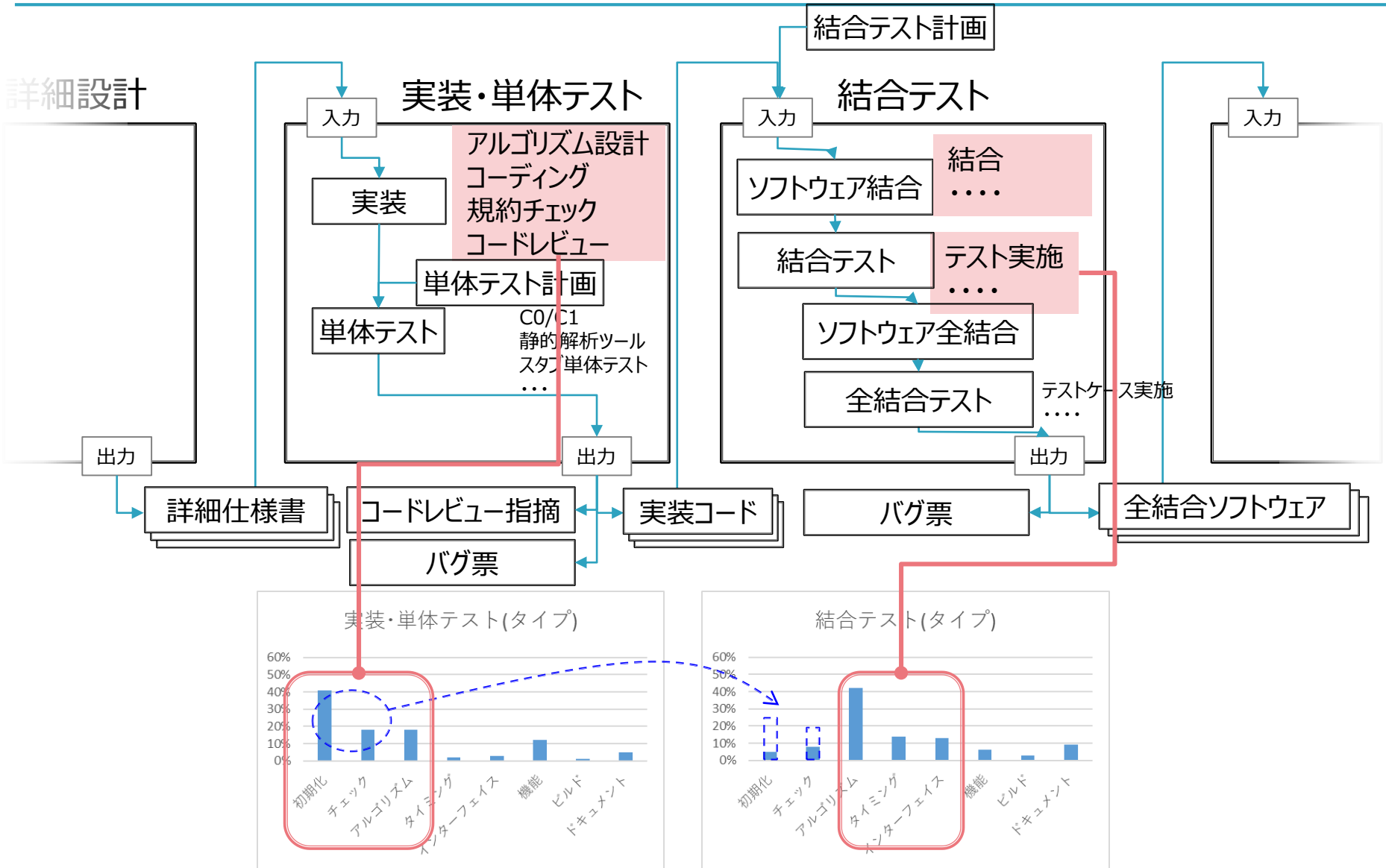
要件定義



レビュー指摘をタイプ分類行った場合、このようなタイプ比率になるのではないか



ODCはそもそもプロセス改革

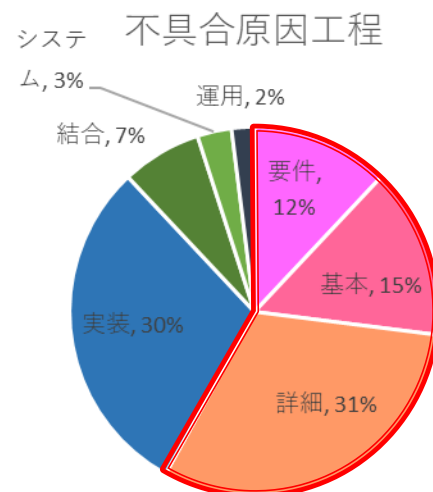


上流からの作込みが重要

▶ 上流起因のバグが58%

ソフト開発においてバグを引き起こした根本分析を行うと58%が要件、基本設計、詳細設計工程に要因がある。

出典：IPA「ソフトウェア産業の実態把握に関する調査」にみる組込み産業の現状 2013

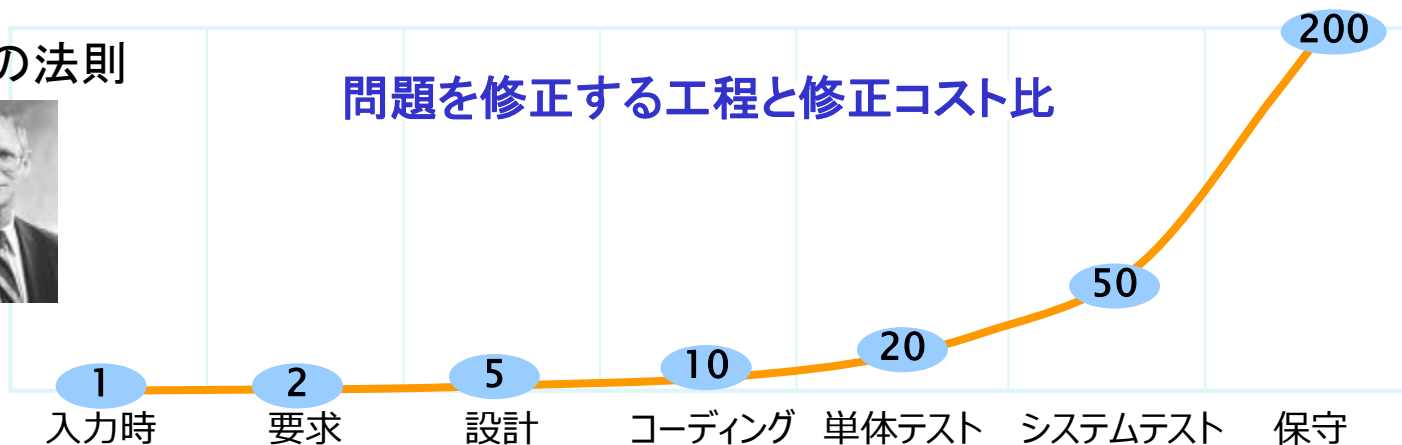


▶ 上流工程で取り除かないと下流ではバグ除去が厄介になる

ベームの法則



問題を修正する工程と修正コスト比

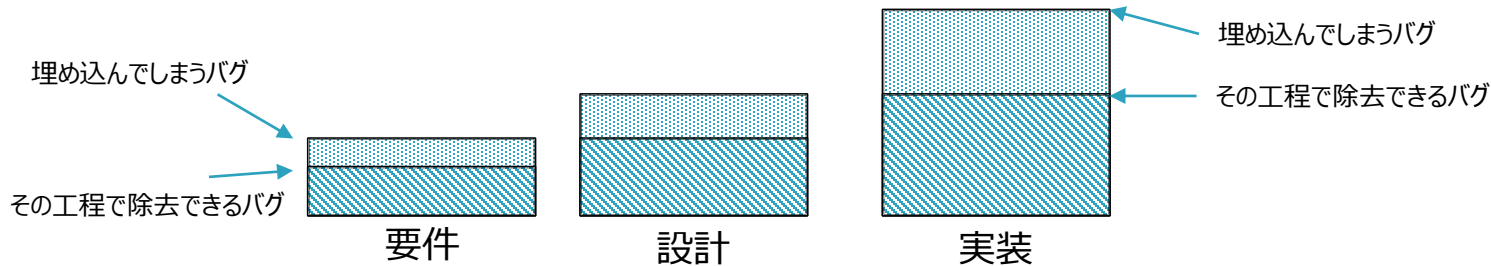


Source:

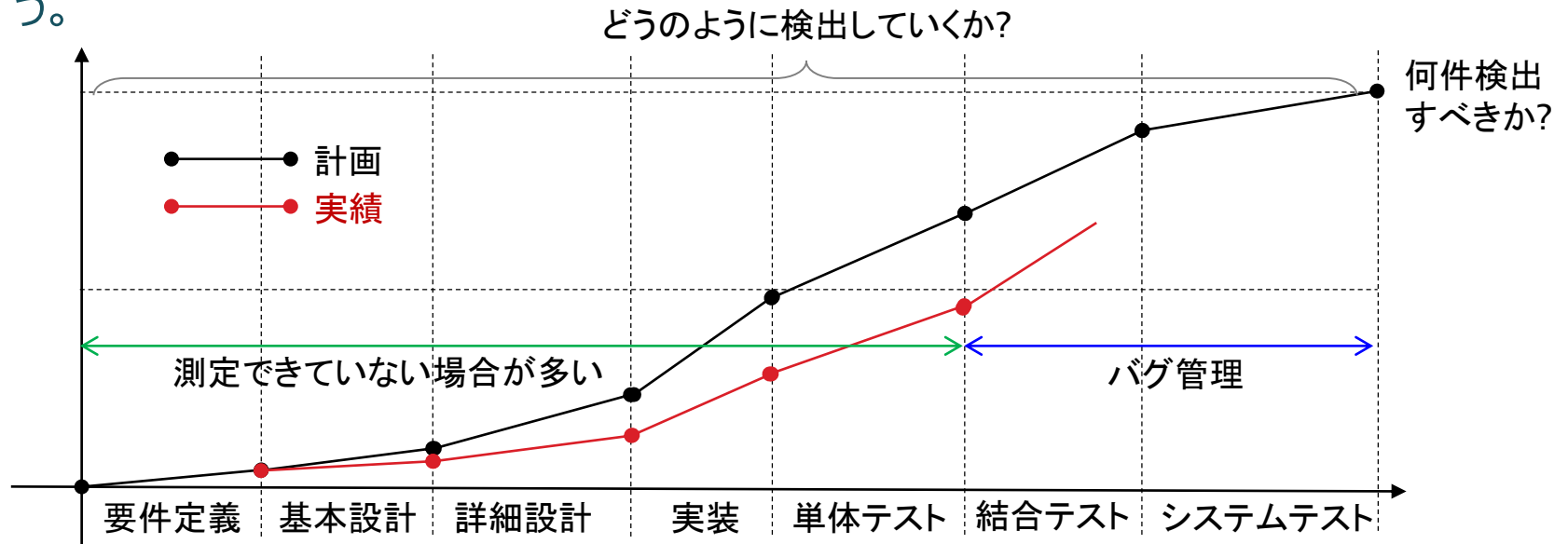
Source: Barry W. Boehm, "Software Engineering Economics" Prentice Hall
Software Engineering, IEEE Computer Society Press

プロジェクト品質管理上、ODCをどう活用するか？→品質計画の策定

- ▶ 各工程で埋め込んでしまうバグは、できるだけ、その工程で除去すべき。だが全て除去することはなかなか難しい

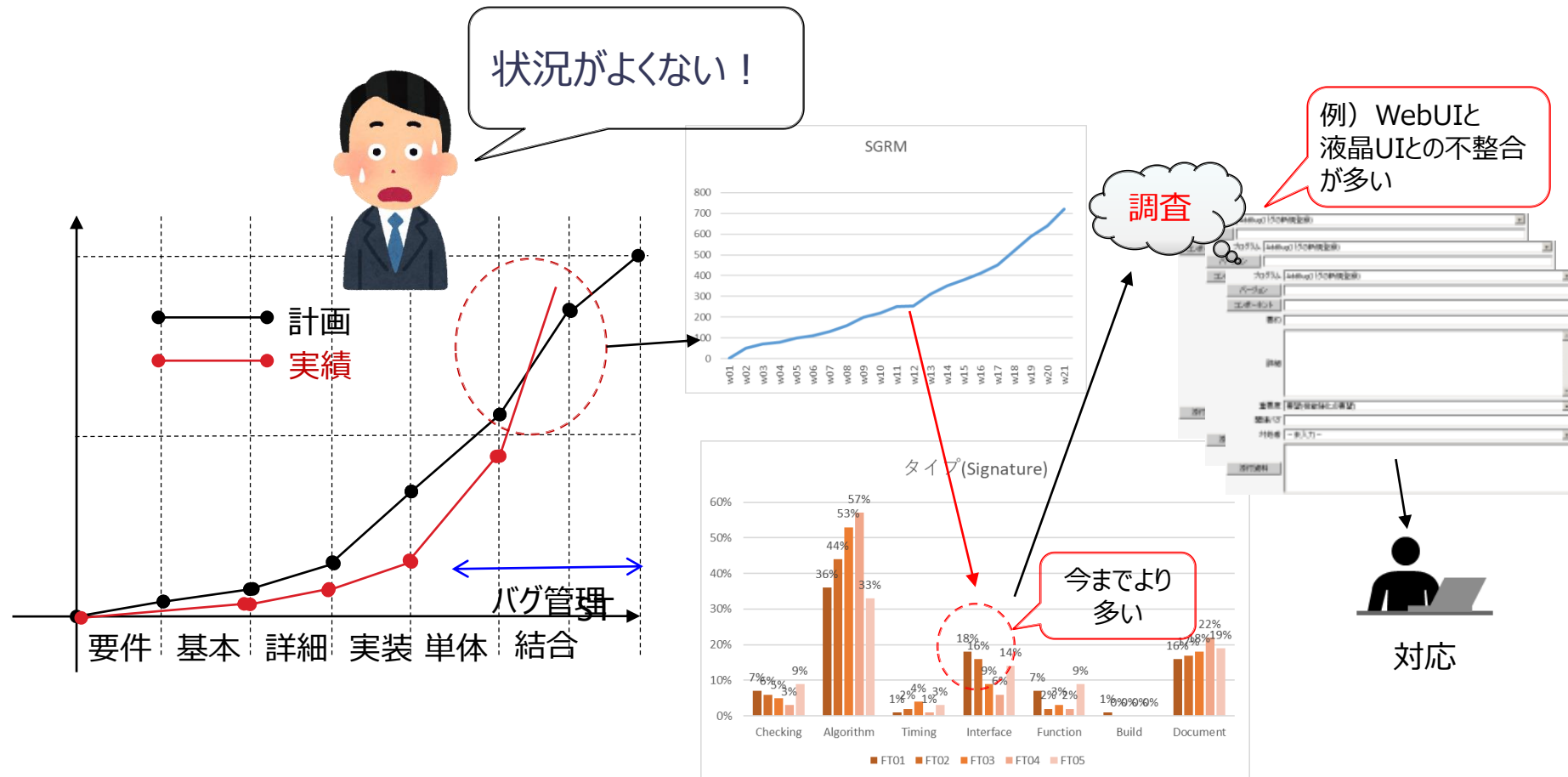


- ▶ 各工程で検出できるバグの計画、すなわち、品質計画を立て、予実管理を行う。



バグ票を調査分類し、対応策へ→分類⇒分析⇒対応

- ▶ ODC分析をライブで実施し、フィードバックすればリスクを回避、軽減させることができる！



ワークショップの紹介

Let's take a look at the workshops

実際にやってみよう

Let's try

- 分類してみよう
- 分析してみよう

バグ票：Sample0001



▶ 概要

リモートUIからFAXアドレス帳をインポートした後で電源OFF/ONするとアドレス帳がインポートされていない

▶ 内容

FAXアドレス帳を初期化した後で、リモートUIからアドレス帳をインポートする。
その後、プリンターの電源をOFF/ONするとアドレス帳がインポートされていない。

▶ 原因

リモートUIからアドレス帳をインポートした場合、リモートUIからの変更フラグがONになるが、電源OFF/ON後の復帰時にリモートUI変更フラグを確認する条件文が漏れていた。

▶ 対策

電源OFF/ON後の復帰時のリモートUI変更フラグを確認する条件文を追加した。

タイプ分類

条件文の追加なので**チェック・漏れ**(Checking Missing)

バグ票： Sample0002



▶ 概要

特殊サイズ用紙でジャムが発生時、指示通り操作してもエラー解除できない

▶ 内容

特殊サイズ用紙をADFに複数枚セットし、印刷し、ジャムが発生した場合、パネルの指示に従いカバーを開け原稿を取り除き、カバーを閉め復帰動作をとってもエラーが表示されたままとなる。

▶ 原因

中間フィード装置を制御するモジュールの特殊サイズ用紙への仕様の考慮漏れがあり、エラー復帰を適切にハンドリング出来ていなかった。

▶ 対策

中間フィード装置制御モジュールのエラー処理を特殊サイズ用紙をハンドリングするよう修正した。

タイプ分類

エラー処理モジュールの仕様が特殊サイズを考慮できておらず、仕様レビューにおいてもこれを見落としている。特殊サイズをハンドリングするように仕様変更し修正しているので**機能・誤り**(Function Incorrect)

ワークショップの紹介

Let's take a look at the workshops

実際にやってみよう

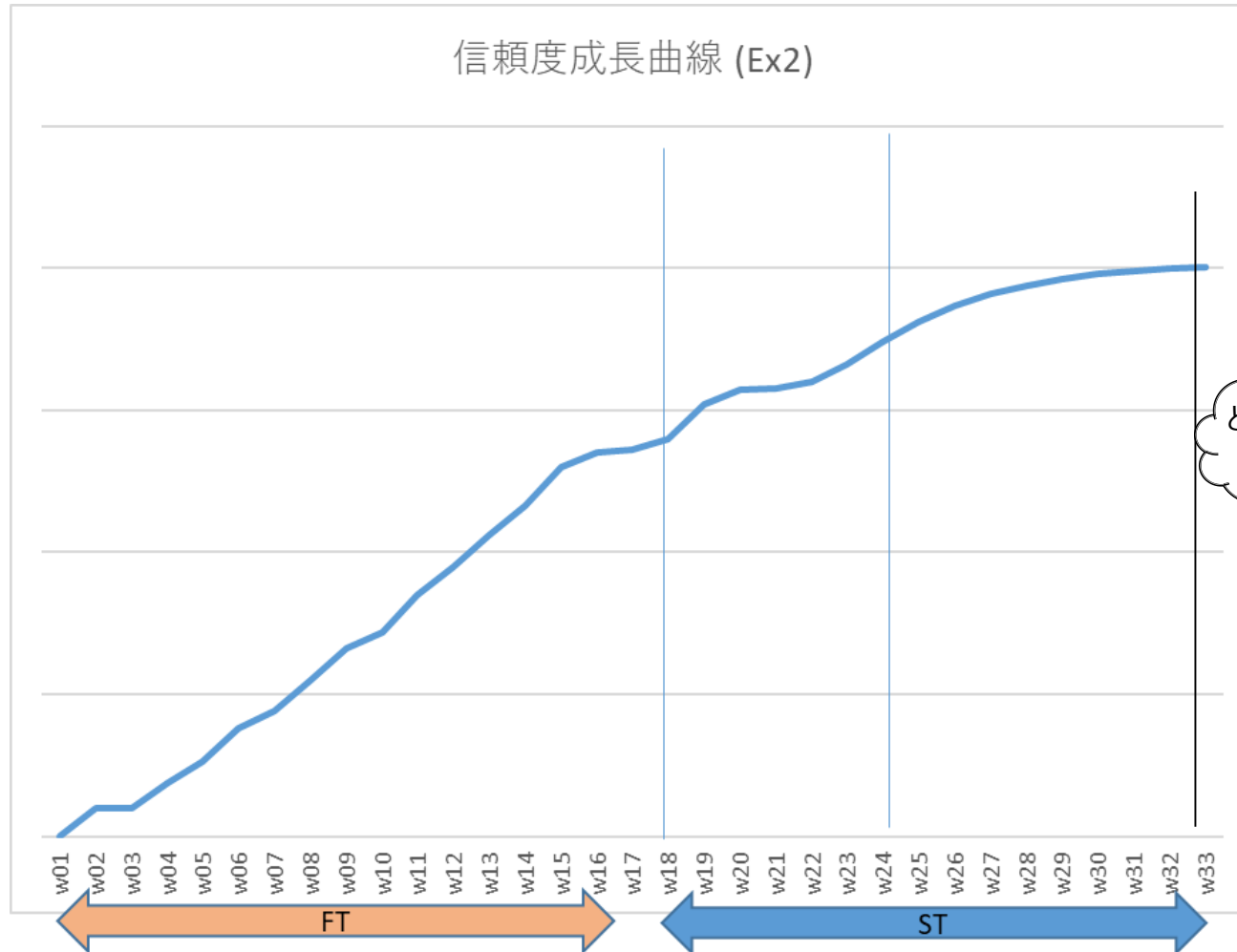
Let's try

□ 分類してみよう

□ 分析してみよう

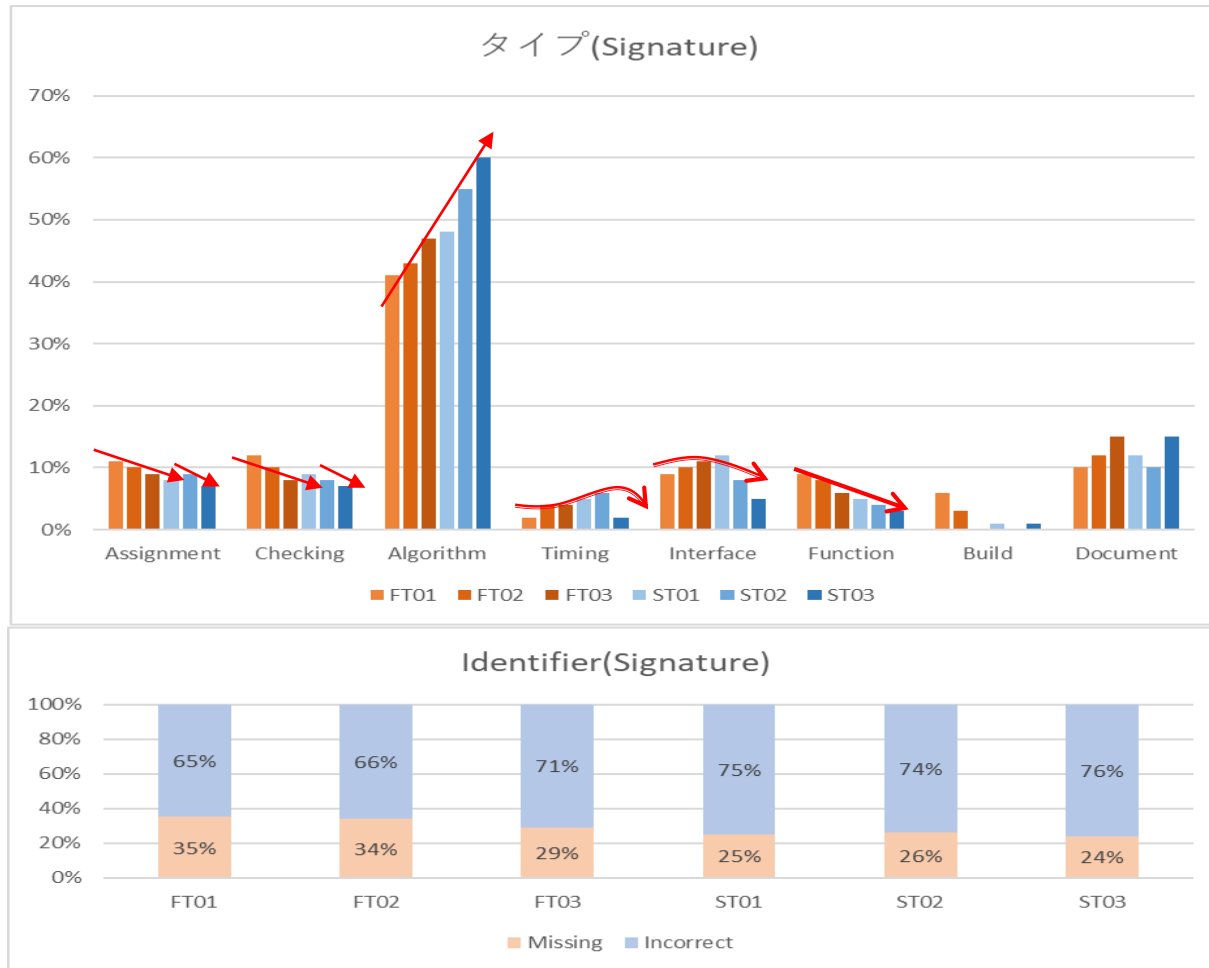
ODC タイプ分析ワークショップ-例①

バグ収束を確認し、審議会で承認され出荷されました。



タイプ比率の変化-「あるべき姿」

- 完了した工程がしっかり行われていれば、あるべきシグナチャー示すはずです。



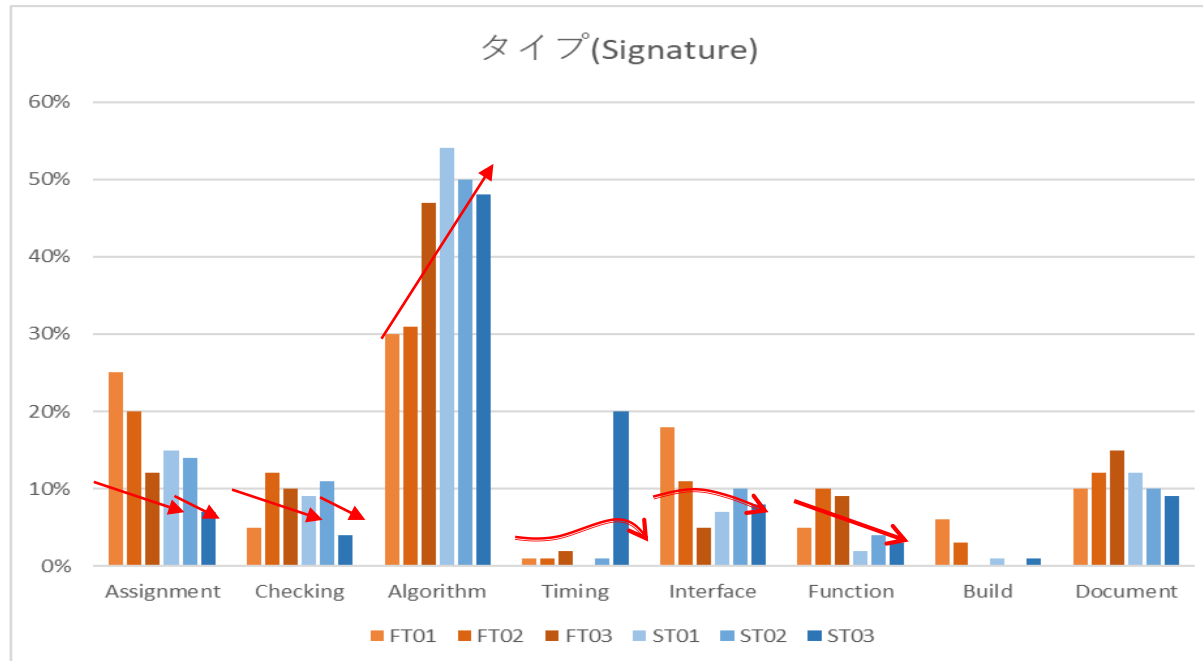
あるべきシグナチャーに
ほぼ近いですね。



しかし、実際は？

タイプ比率の変化-実例

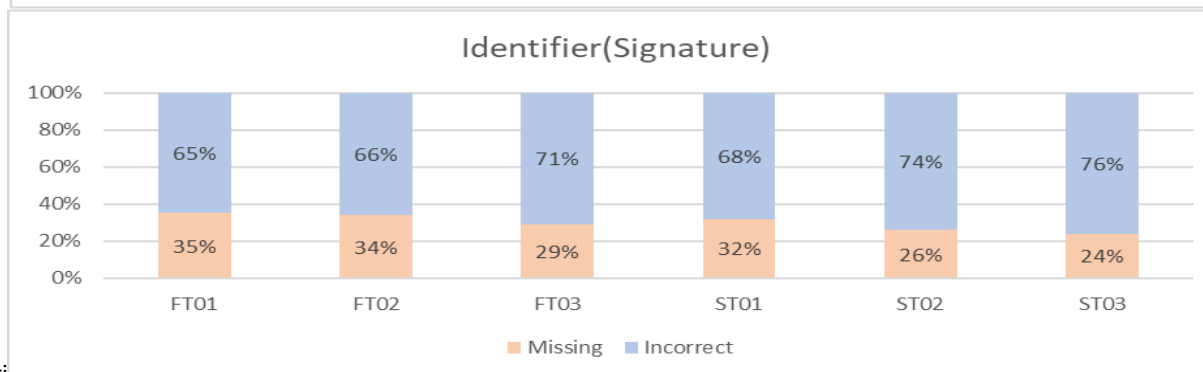
- あるべきシグナチャーの変化を赤線で示しています。



実際のシグナチャーと比較し、プロジェクトに何が起きているか？

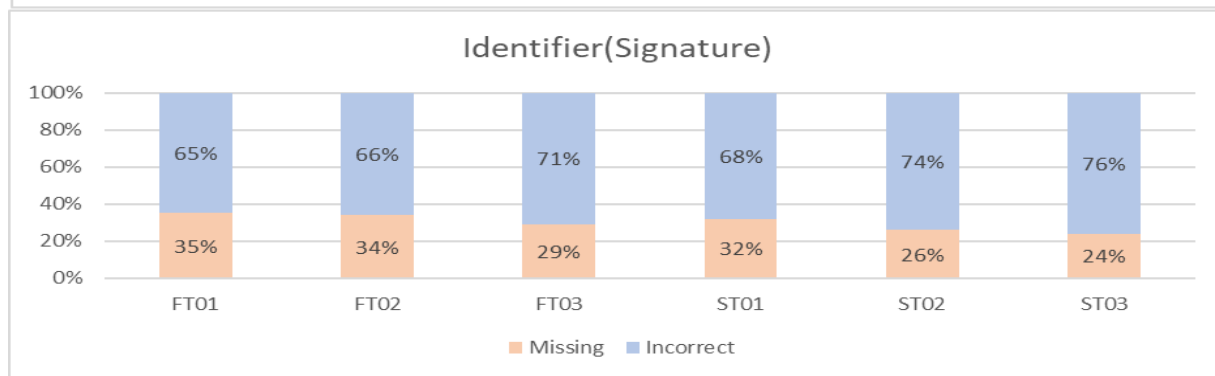
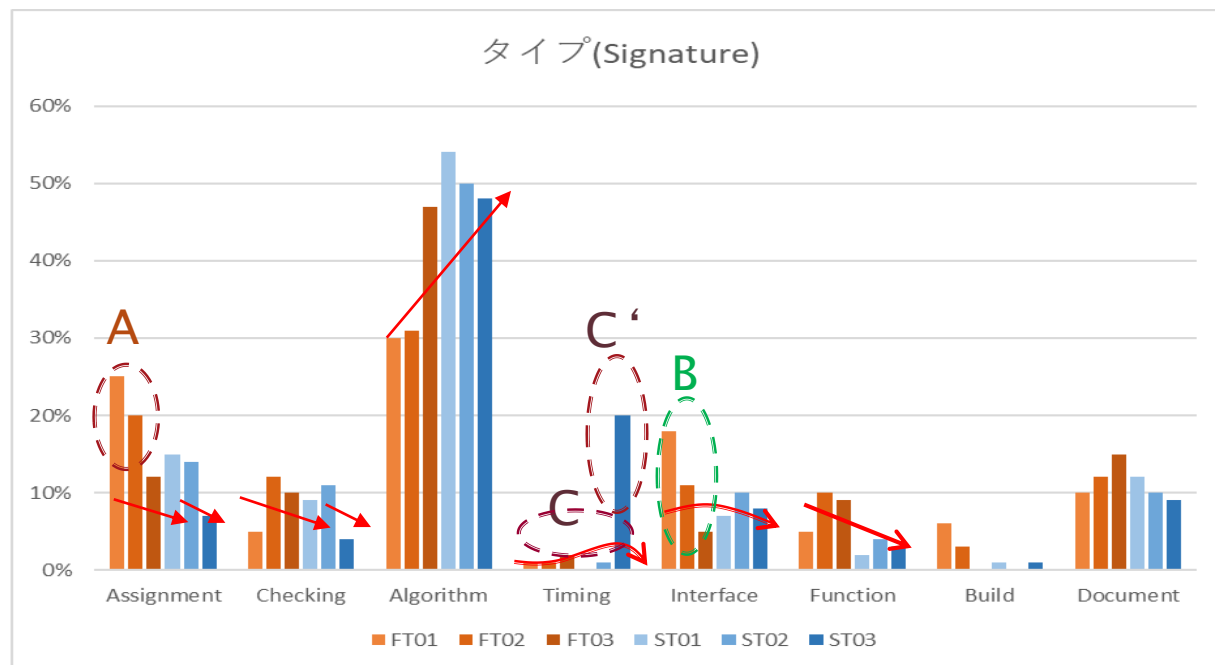


仮説を立てましょう。



ワークショップ

タイプ比率の変化-分析はどうか？



FT開始直後の特徴

A:「代入・初期化」が多すぎる、単体テスト不足だったことが懸念される

B:「インターフェース」が多すぎるー再設計の影響か？リスクがあるので対策要)

FT開始後の特徴

C:「タイミング」が検出されるべきだが、検出されない

ST開始後の特徴

C:終盤まで「タイミング」が少なすぎる。システム動作不安定が懸念

ST終盤の特徴

C':タイミングは最後20%に達する

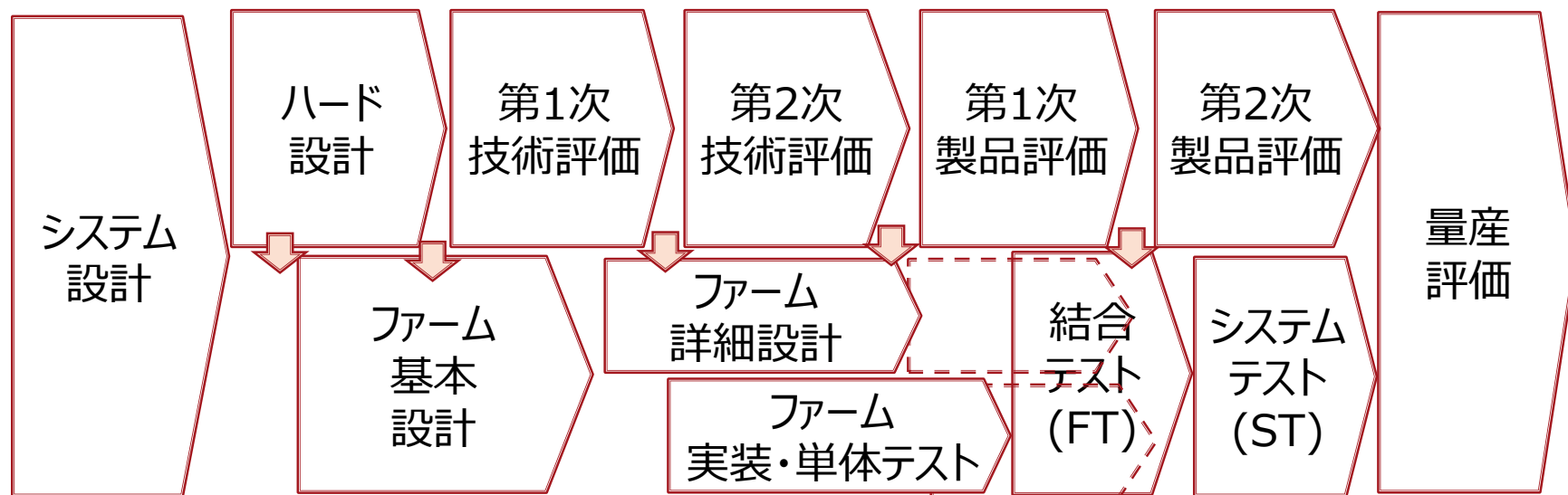
リスクあり ↑

B:部品(A)はST終盤やっとまともに動き始めた。他のタイミング問題が今後出る可能性が高い。

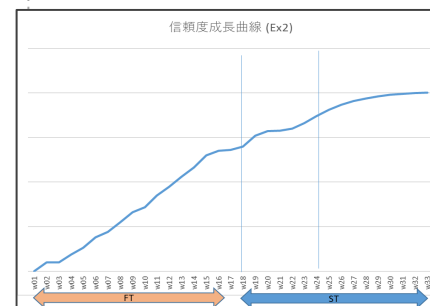
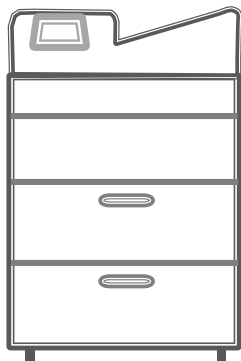
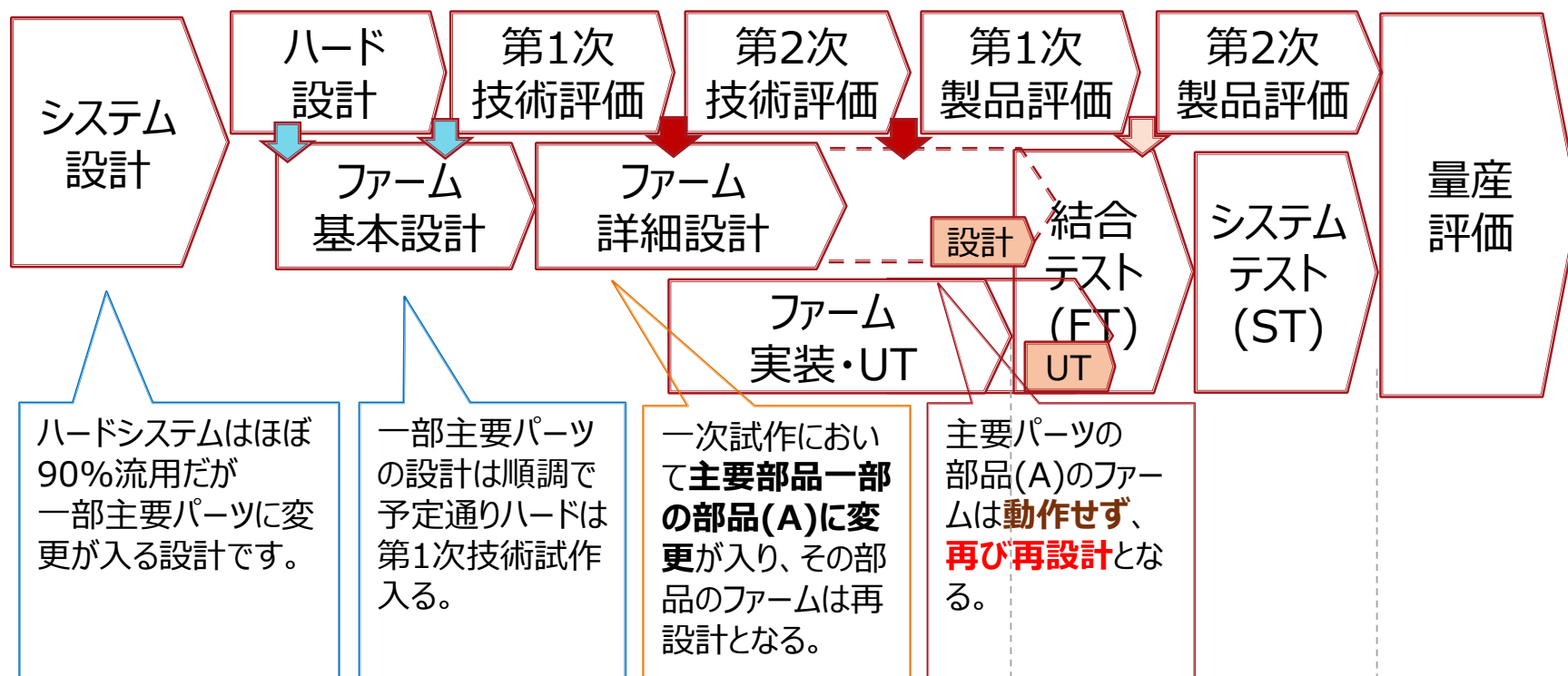


開発の現状

実際の開発では、ハードウェアの新規開発では試作が段が多段階で、ファームウェアへの仕様が多段階になることで、反復型になることが多い。

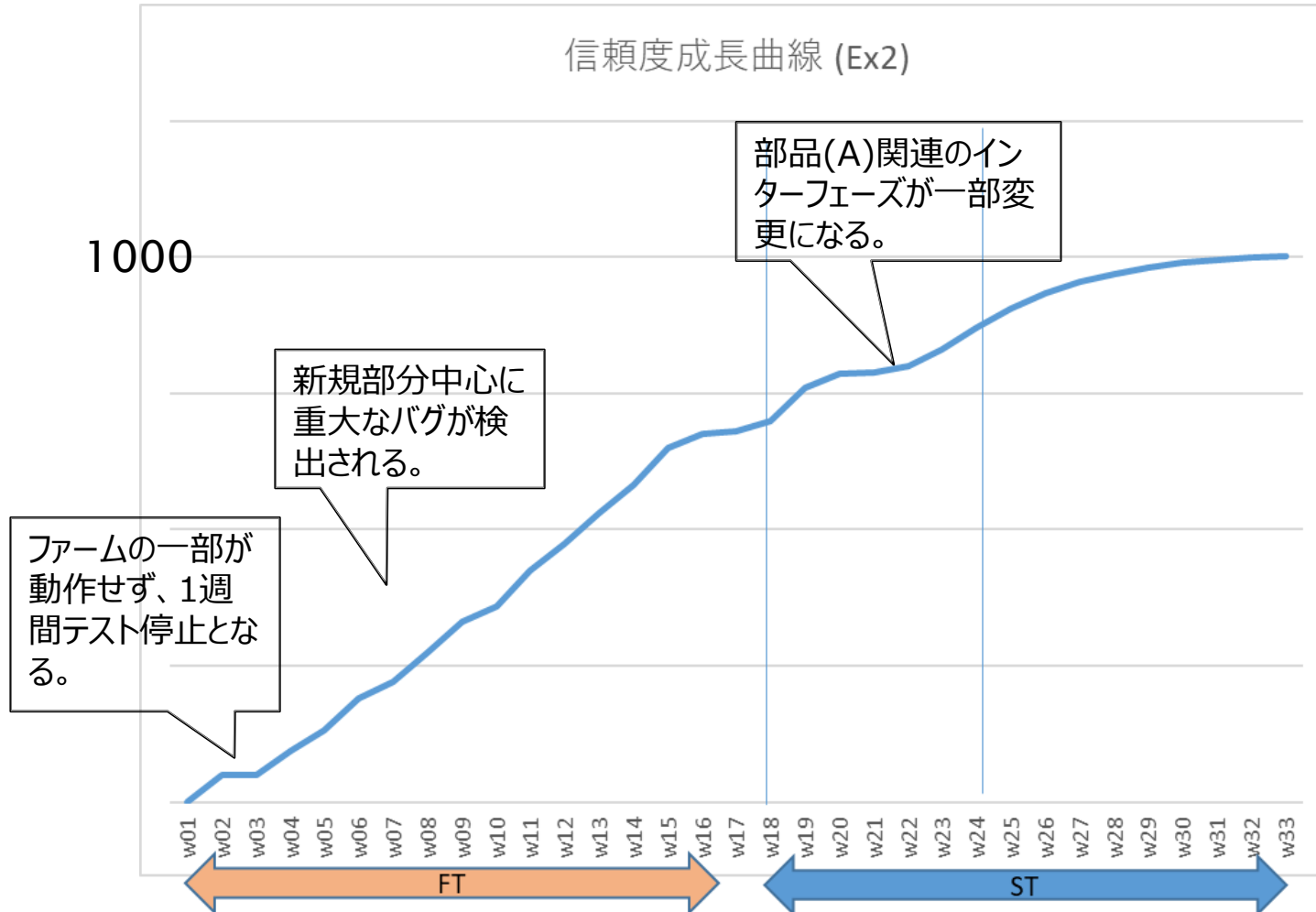


プロジェクト①の経緯

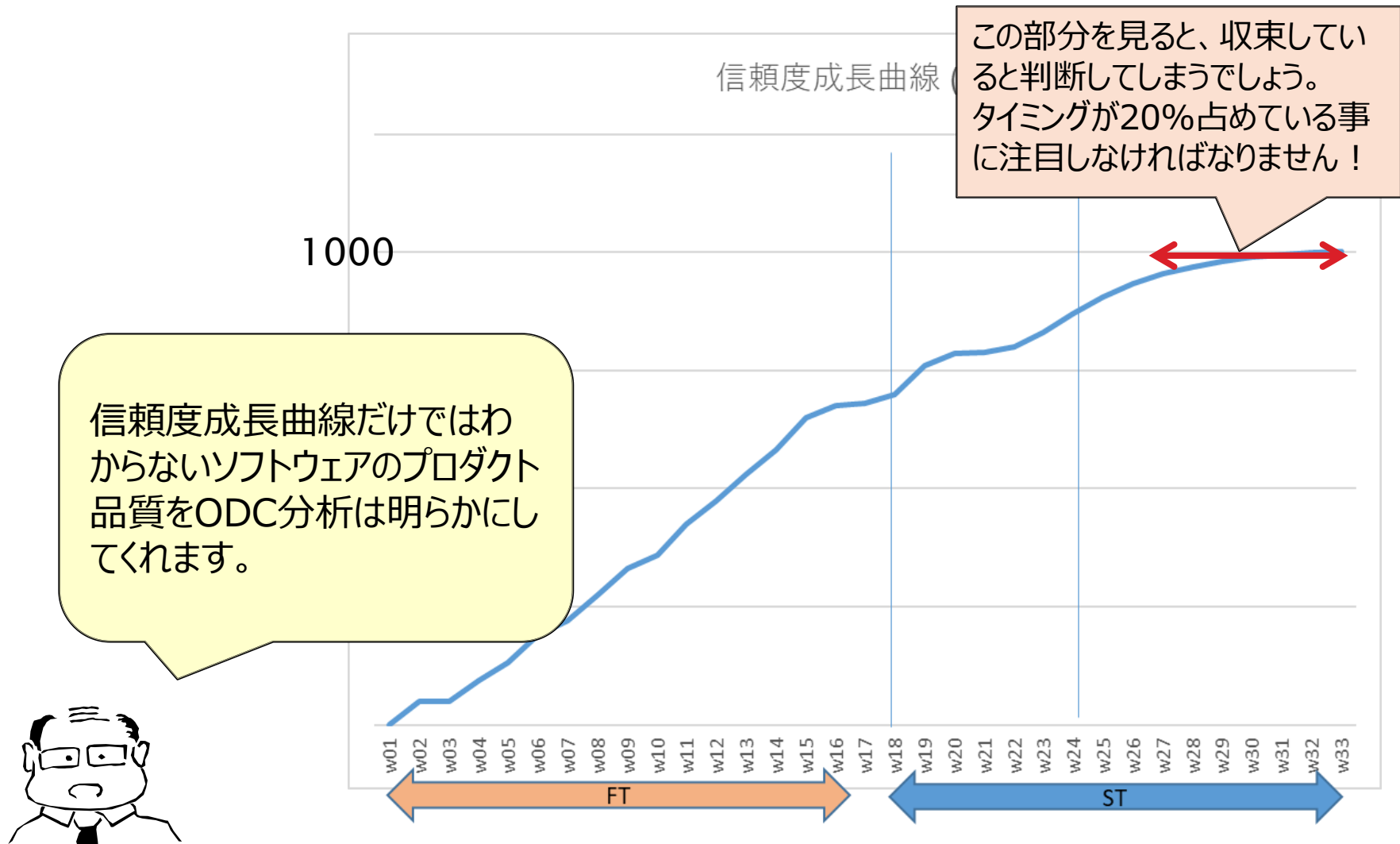


ODC タイプ分析ワークショップ-例①

- 以下の信頼度成長曲線をとりました。一部状況を補足します。



まとめ：ODC分析でわかるセマンティックス



ご清聴ありがとうございました

