

JaSST '07 Sapporo

要求仕様とテストの〇〇な関係 - モレを防ぐ -

東京エレクトロン ソフトウェア・テクノロジーズ
根本 紀之
2007/10/02



東京エレクトロン

本日の目的

ソフトウェア品質向上のための有効な手段として
要求仕様を軸とした開発プロセスの紹介をします。

要求仕様を軸とした 開発プロセスの導入と実践

発表のシナリオ

1. 問題分析と導入過程
2. 開発プロセスの説明
3. 開発プロセスの実践

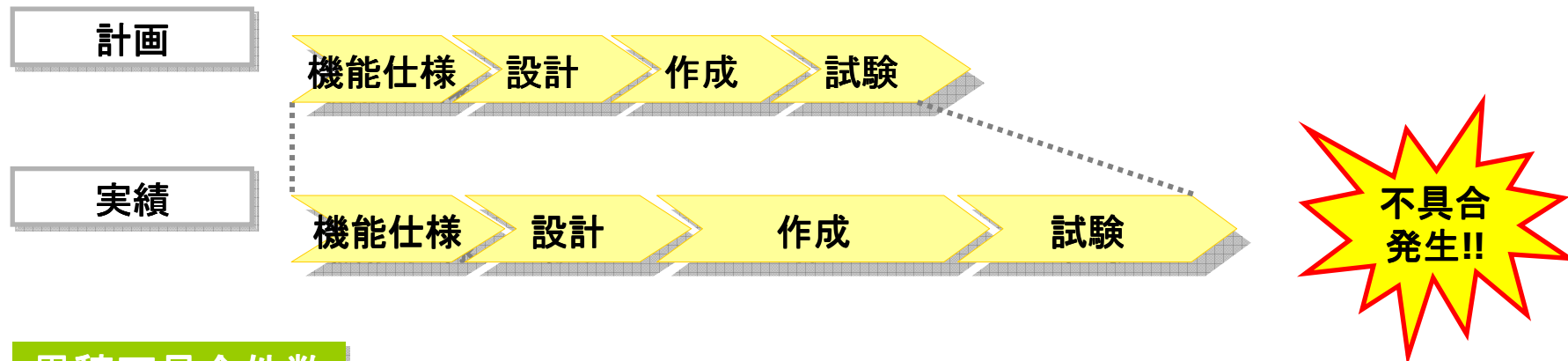
要求仕様を軸とした 開発プロセスの導入と実践

発表のシナリオ

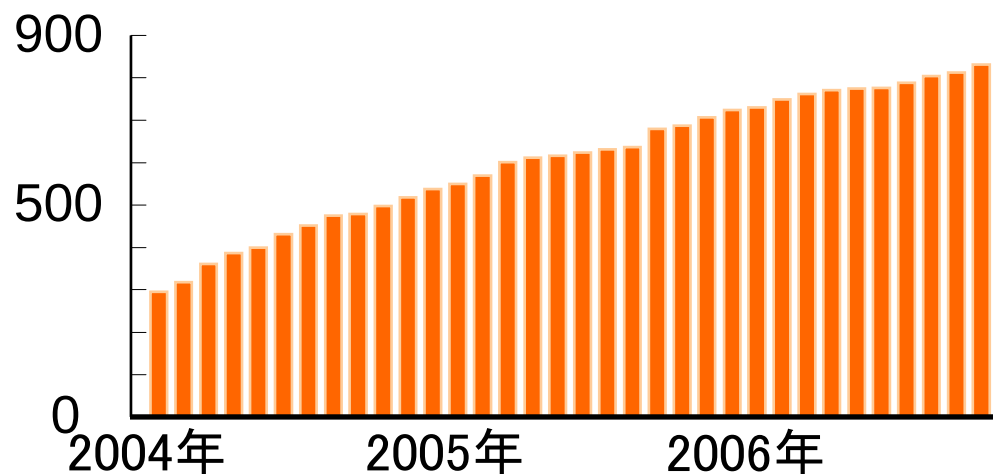
1. 問題分析と導入過程
2. 開発プロセスの説明
3. 開発プロセスの実践

要求仕様書を軸とした開発プロセスの 導入以前の開発状況

リリースの遅れや客先での不具合が多発していた



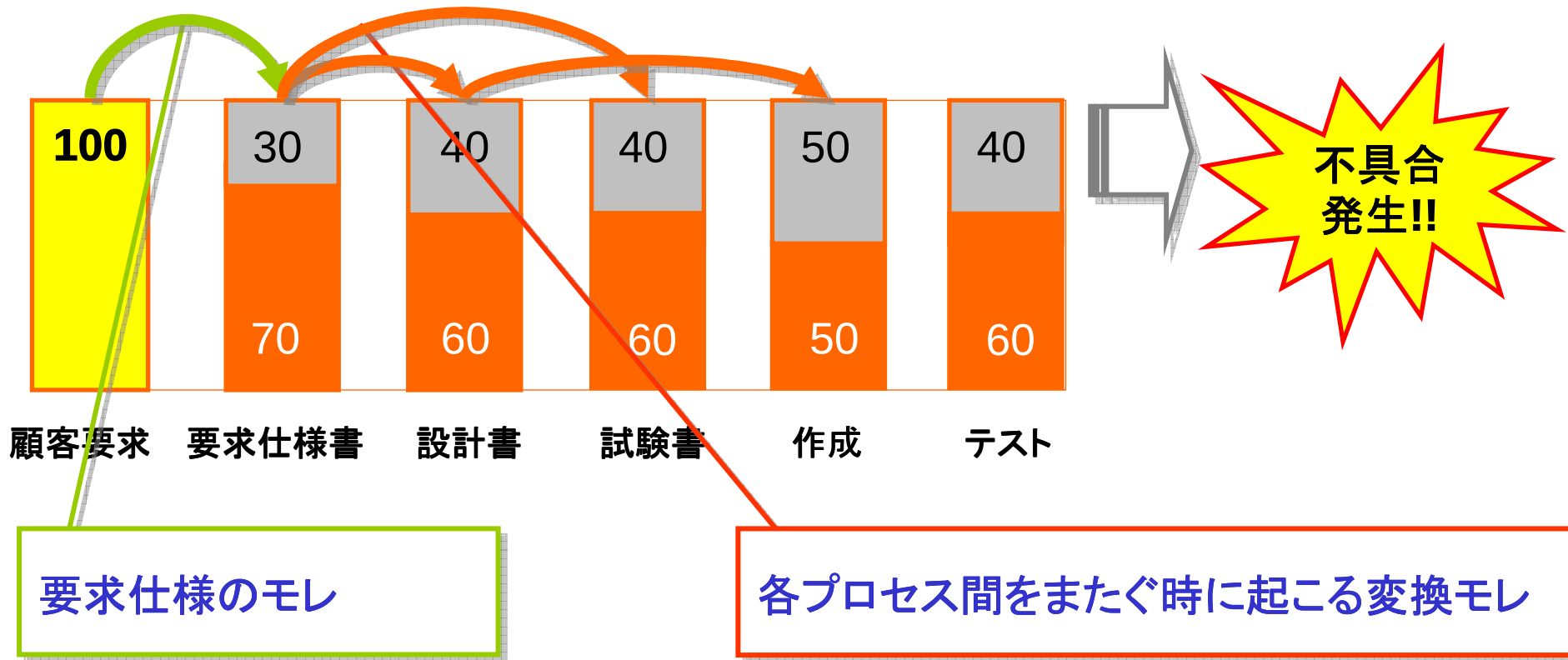
累積不具合件数



この不具合
の原因は何
だろう？

顧客要求は開発過程で漏れていき、 そのモレが最終的に不具合となる

モレのイメージ



全ての要求は仕様に反映されない

要求仕様のモレ

工場からはメールやメモ等で仕様が伝えられ、
それをもとに各開発者が仕様書を作成し、作成→テストを行っていた。



従来の仕様書では全ての仕様を 次のプロセスに反映できない

各プロセス間をまたぐときに起こる変換モレ

3.2 ファイル圧縮方式:データ削除

データ削除によるファイル圧縮とは、通常 Ingenio ES システムにてセンサ単位で管理されるデータファイルより特定の項目データを削除することによりデータ量の低減を行う圧縮手法である。

データ削除によるファイル圧縮を行うタイミングは、Ingenio ES Server でのデータベース格納時に処理を行う。Ingenio ES Controller 内では本処理を行わない。

Ingenio ES システムにて管理している時系列データ (RAW データ) は、以下の目的に使用される。

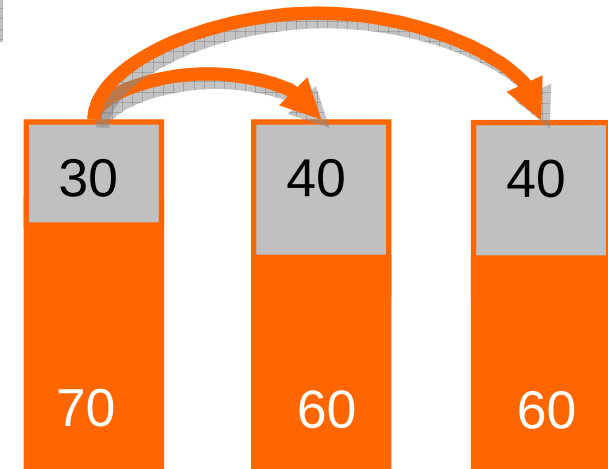
- ・ ウェハ単位 (ロット単位) 集計データの算出元データ
- ・ チャート表示用保持データ

ウェハ単位 (ロット単位) 集計データの算出元データとして使用する場合に関して、データが削除されている状態では求められる全データの集計データを算出することが出来ないこととなる。つまりウェハ単位 (ロット単位) 集計データを算出する前に削除処理を行ってはいならない。算出処理は Ingenio ES Controller にて実施される。

チャート表示用保持データとして使用する場合に関して、オペレータが参照したいデータを残せばよい。つまり参照を日常行わないデータに関してデータ削除の対象としても良いこととなる。ただし、参照対象のデータを参照したいオペレータが参照出来ない場合を考慮し、削除処理の有効/無効の切り替えが可能。

一つの文に異なる仕様がいくつも書かれている

条件や異常系が読み取りにくい



要求仕様書

設計書

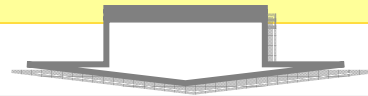
試験書

試験書と要求書の関連がないので、確認に手間がかかる。試験の目的がはっきりしない。

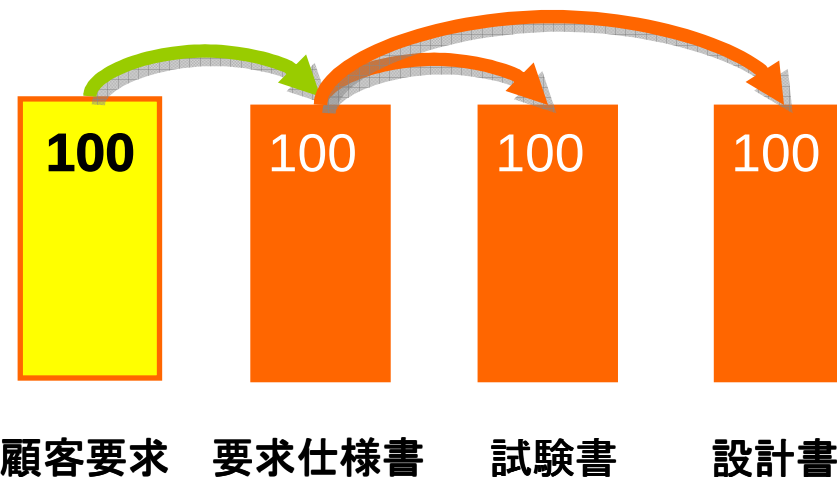
問題解決に向けて “要求仕様を軸とした開発プロセス”を導入した

問題点

- (1) 要求仕様のモレ
- (2) 各プロセス間をまたぐ時に起こる仕様変換モレ



“要求仕様を軸とした開発プロセス”を導入し問題解決を図る



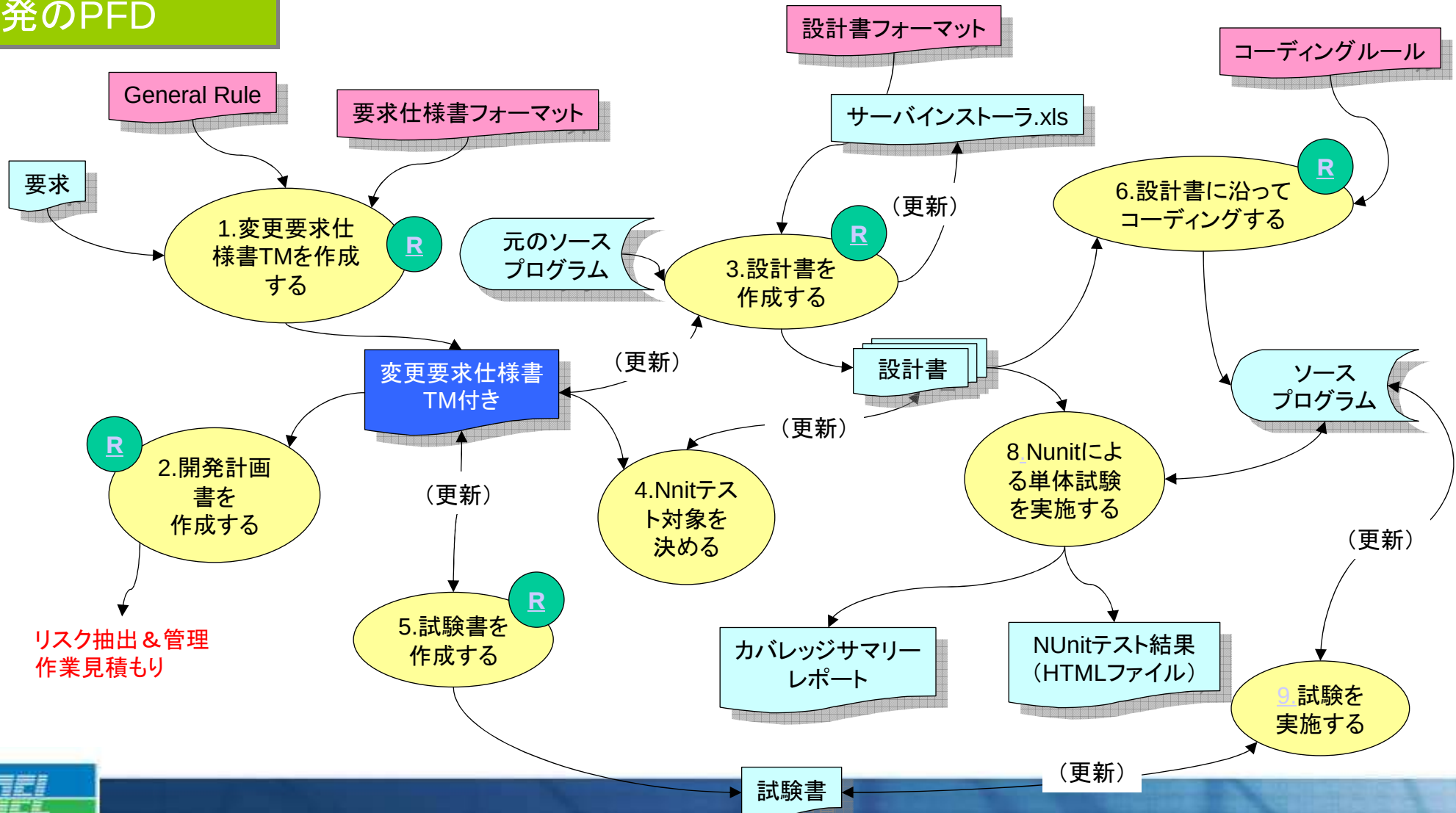
要求仕様を軸とした開発プロセスの導入

発表のシナリオ

1. 問題分析と導入過程
2. 開発プロセスの説明
3. 開発プロセスの実践

プロセスフローを定義し、 “やること”と“できるもの”を明確にする

開発のPFD



要求から理由を考え、 要求を満たす仕様を作成する

要求仕様書の特徴

- ① 要求と、その理由を必ずセットにして考える
- ② 要求から仕様を作成する
- ③ それぞれを番号で識別する

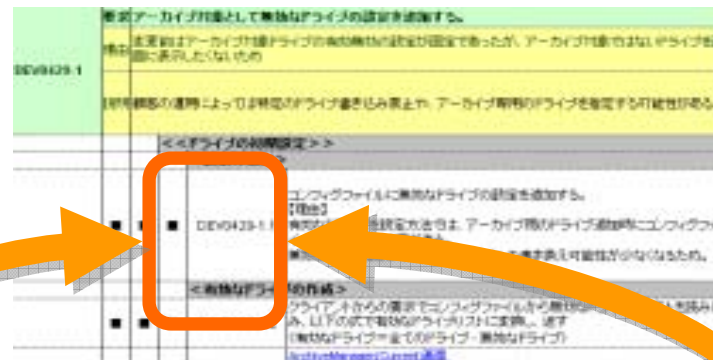
①	DEV0429-2	要求	アーカイブをフォルダ管理し、複数のアーカイブを任意の名前で保存したい
		理由	変更前は複数のアーカイブを保存する機能の使い勝手が悪い ため
		説明	複数個のアーカイブを管理する場合に 1)アーカイブ実行 2)ディスクの入れ替え、または自分でコピー 3)再アーカイブ という手順を踏む必要があった。
			<<アーカイブのフォルダ管理>>
		<フォルダの表示>	
③	DEV0429-2.1	ArchiveManager画面にTreeViewを追加しサーバーフォルダをTree表示する。	
	<フォルダの選択>		
	DEV0429-2.2	新規追加したTreeViewを用いて、フォルダを選択し、そのフォルダに存在するアーカイブをリストビューに表示する。 ArchiveManager(Current)画面	
	<<アーカイブの保存>>		
		<フォルダの指定>	
	DEV0429-2.3	アーカイブの保存時のフォルダの指定方法として、TreeView上でフォルダの操作(作成、名称変更、削除)を行い、任意の保存先のフォルダを選択する。	

② 要求を満たす仕様を作成する

仕様をトレースする

- ① 各仕様毎に設計、テストケースを作成する
- ② 仕様番号でモレが無いかチェックを行う

要求仕様書



要求仕様書#	DEV0429-1.1
ソース名	IngenioConfig.xml
要求仕様書#	DEV0429-1.2
ソース名	ArchiveService.asmx.cs
要求仕様書#	DEV0429-2.1
ソース名	DefineDatabaseState.cs
ソース名	ArchiveDatabaseInfo.cs
ソース名	DatabaseInfo.cs
ソース名	TelArchistory.cs
ソース名	FrmArchiveManager.cs
ソース名	ArchiveService.asmx.cs
要求仕様書#	DEV0429-2.3
ソース名	FrmArchiveManager.cs
要求仕様書#	DEV0429-2.4
ソース名	FrmArchive.cs
要求仕様書#	DEV0429-2.5
ソース名	InstalledAn

詳細設計書

仕様カテゴリ	仕様番号	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.1	2.2	2.3	2.4
無効なドライブの設定	DEV0429-2.1													
	DEV0429-2.2													
	DEV0429-2.3													
アーカイブのファイル管理と圧縮の制御	DEV0429-2.4													
	DEV0429-2.5													
	DEV0429-2.6													
コンバート実行	DEV0429-2.7													
	DEV0429-2.8													
	DEV0429-2.9													
	DEV0429-3.0													
操作履歴の表示	DEV0429-3.1													
	DEV0429-3.2													
	DEV0429-3.3													
	DEV0429-3.4													
	DEV0429-3.5													
	DEV0429-3.6													
	DEV0429-3.7													
	DEV0429-3.8													
	DEV0429-3.9													
	DEV0429-4.0													

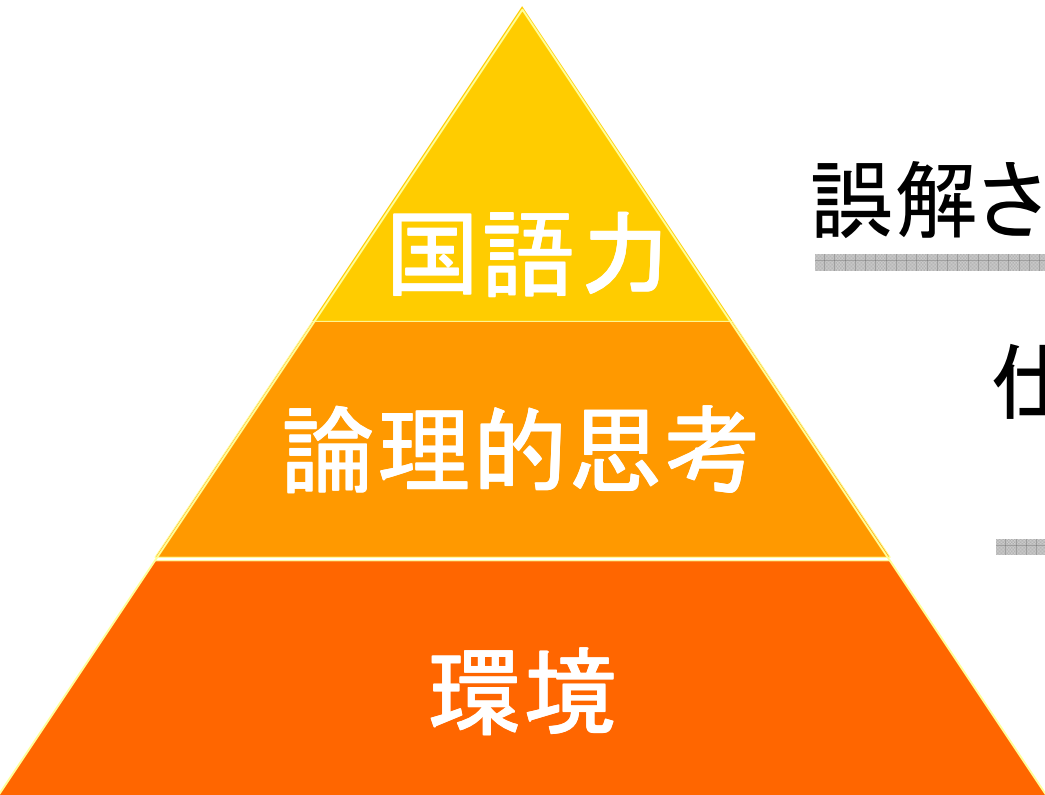
試験仕様書

要求仕様を軸とした 開発プロセスの導入と実践

発表のシナリオ

1. 問題分析と導入過程
2. 開発プロセスの説明
3. 開発プロセスの実践

要求仕様を支える仕組み



国語力

誤解されない仕様を表現する国語力

論理的思考

仕様を整理する
ロジカルシンキング

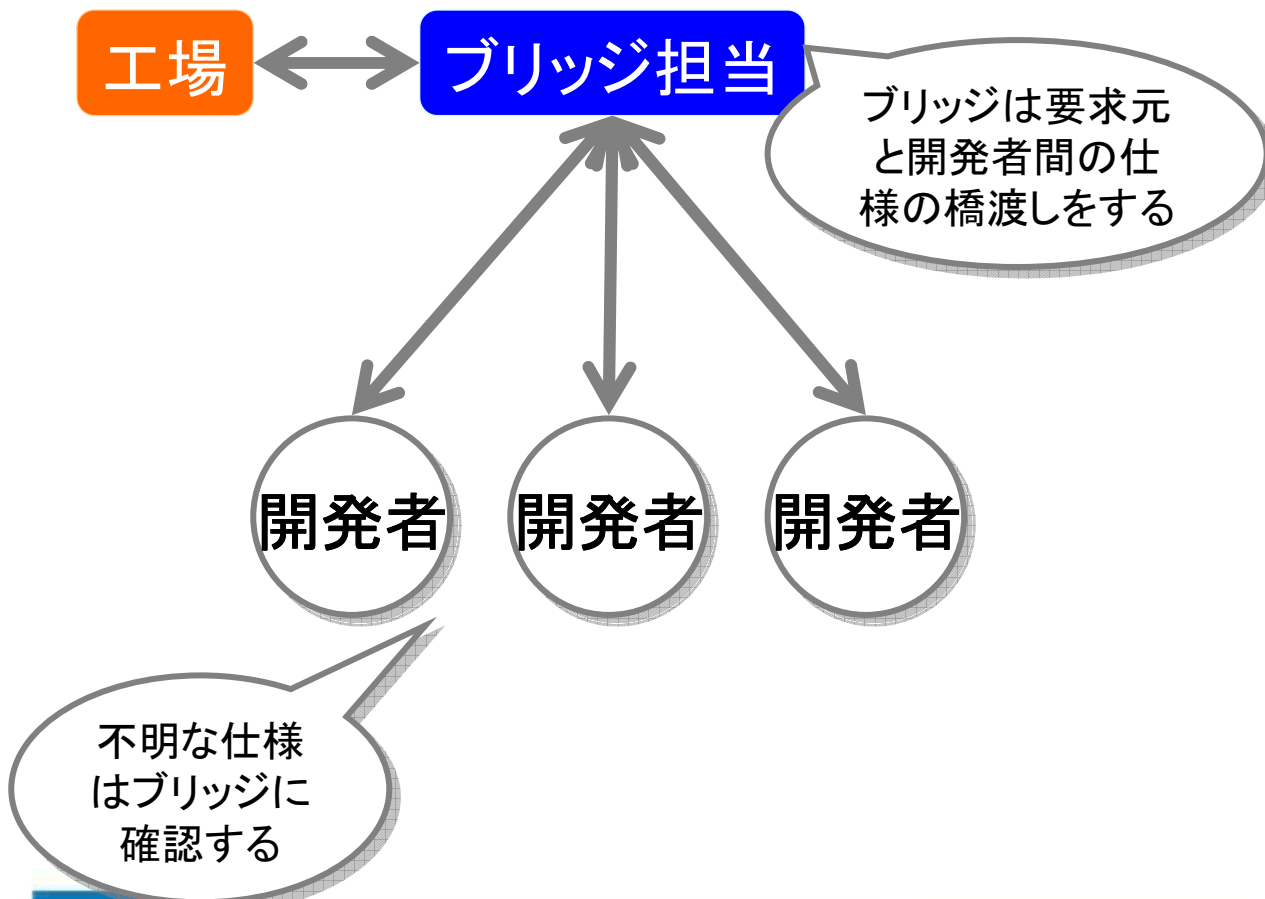
環境

要求と理由を聞ける
環境・体制の構築

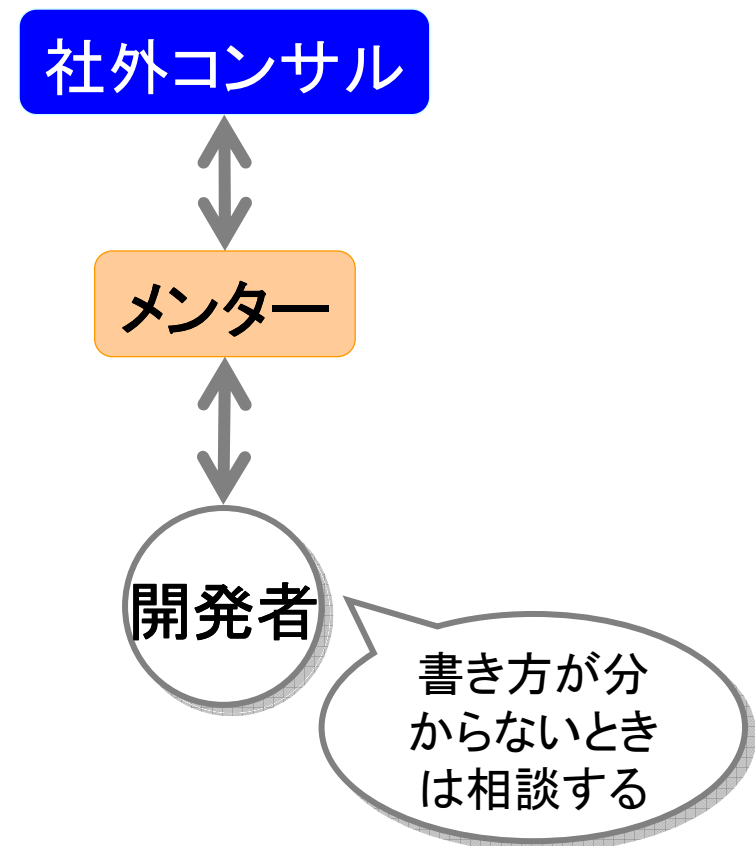
- ① 仕組みを理解する
- ② 意識しながら練習を積み重ねる

モレのない要求仕様書を書く ～環境・体制の準備～

環境の構築



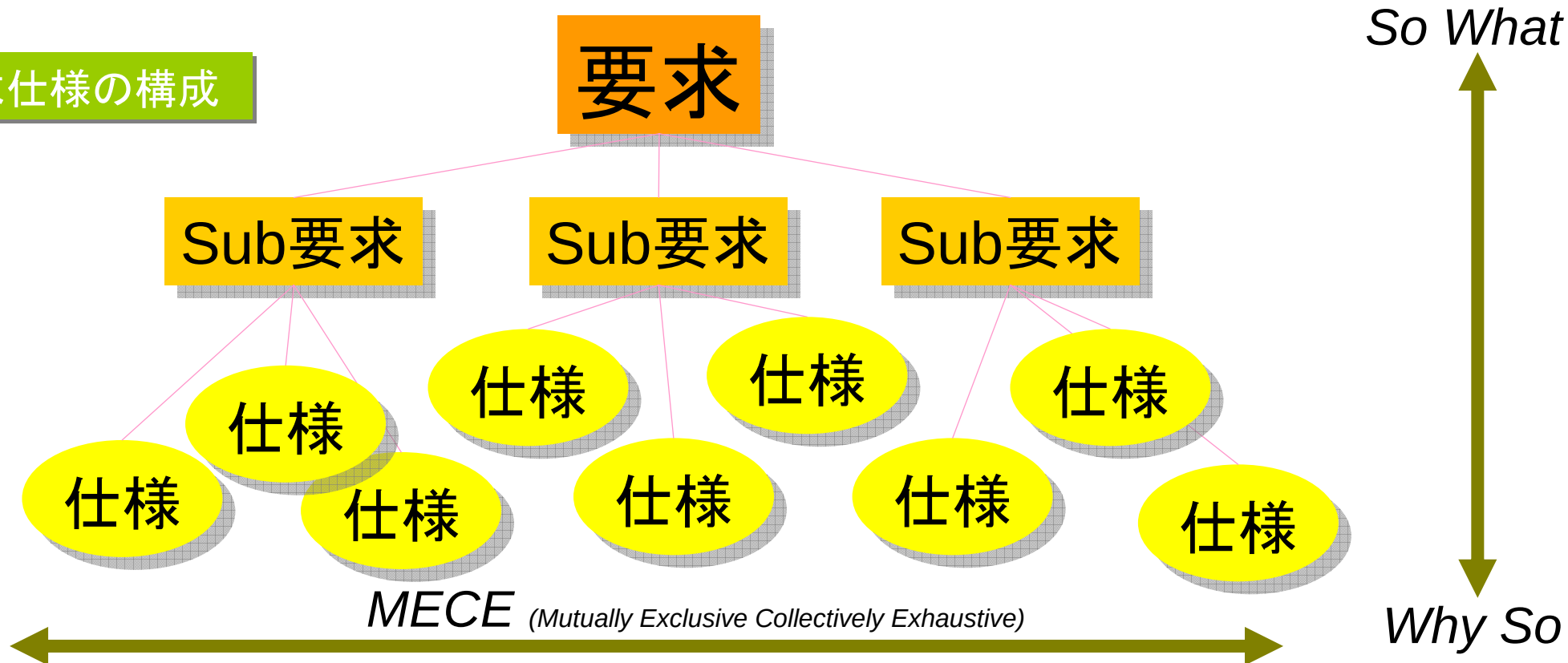
教育



モレのない要求仕様書を書く ～論理的思考～

要求仕様はロジックの集積なので、それをモレなくダブリなく整理する

要求仕様の構成



モレのない要求仕様書を書く ～国語力～

要求

Sub要求

Sub要求

仕様

仕様

仕様

仕様

仕様

仕様

仕様

要求	DEV0429	要求	アーカイブデータをユーザーが任意の名前で任意のフォルダに保存し、管理したい。
		理由	以下の点で使い勝手が良くなかったため 1) アーカイブ名が固定だったため、ユーザーが任意の名前を使用することができなかった 2) アーカイブファイルは1ドライブに1ファイルと限定されていたためドライブの1個しかアーカイブデータを作成することができなかった。 3) フォルダ毎に管理ができなかった。
		説明	
Sub要求	DEV0429-1	要求	アーカイブ対象として無効なドライブの設定を追加する。
		理由	変更前はアーカイブ対象ドライブの有効無効の設定が固定であったが、アーカイブ対象ではないドライブを画面に表示したくないため
		説明	
		<<ドライブの初期設定>>	
		<設定の追加>	
		■ DEV0429-1.1	コンフィグファイルに無効なドライブの設定を追加する。
		<有効なドライブの作成>	
		■ DEV0429-1.2	クライアントからの要求でコンフィグファイルから無効なドライブのリストを読み込み、以下の式で有効なドライブリストに変換し、送す (有効なドライブ=全てのドライブ-無効なドライブ) ArchiveManager(Current)参照
Sub要求	DEV0429-2	要求	アーカイブをフォルダ管理し、複数のアーカイブを任意の名前で保存したい
		理由	変更前は複数のアーカイブを保存する機能の使い勝手が悪いため 複数のアーカイブを管理する場合には 1) アーカイブ実行 2) ディスクの入れ替え、または自分でコピー 3) 再アーカイブ という手順を踏む必要があった。
		説明	
		<<アーカイブのフォルダ管理>>	
		<フォルダの表示>	
		■ DEV0429-2.1	ArchiveManager画面にTreeViewを追加しサーバーフォルダをTree表示する。
		<フォルダの選択>	
		■ DEV0429-2.2	新規追加したTreeViewを用いてフォルダを選択する。
		■ DEV0429-2.3	選択されたフォルダに存在するアーカイブをリストビューに表示する。 ArchiveManager(Current)参照
		<<アーカイブの保存>>	
		<フォルダの保存>	
		■ DEV0429-2.4	選択されたフォルダにアーカイブを保存する。
		<アーカイブの表示>	
		■ DEV0429-2.5	選択されたフォルダにアーカイブを表示する。

同一の動作や名称に対して異なる表現を使わない

誤解を生まない語順で書く。

曖昧な表現を使わない。

- ・論理的に構成されているのでモレ・ダブリが見えやすい
- ・1つの仕様は50字程度なので、レビューもしやすい

フォーマットへの追加

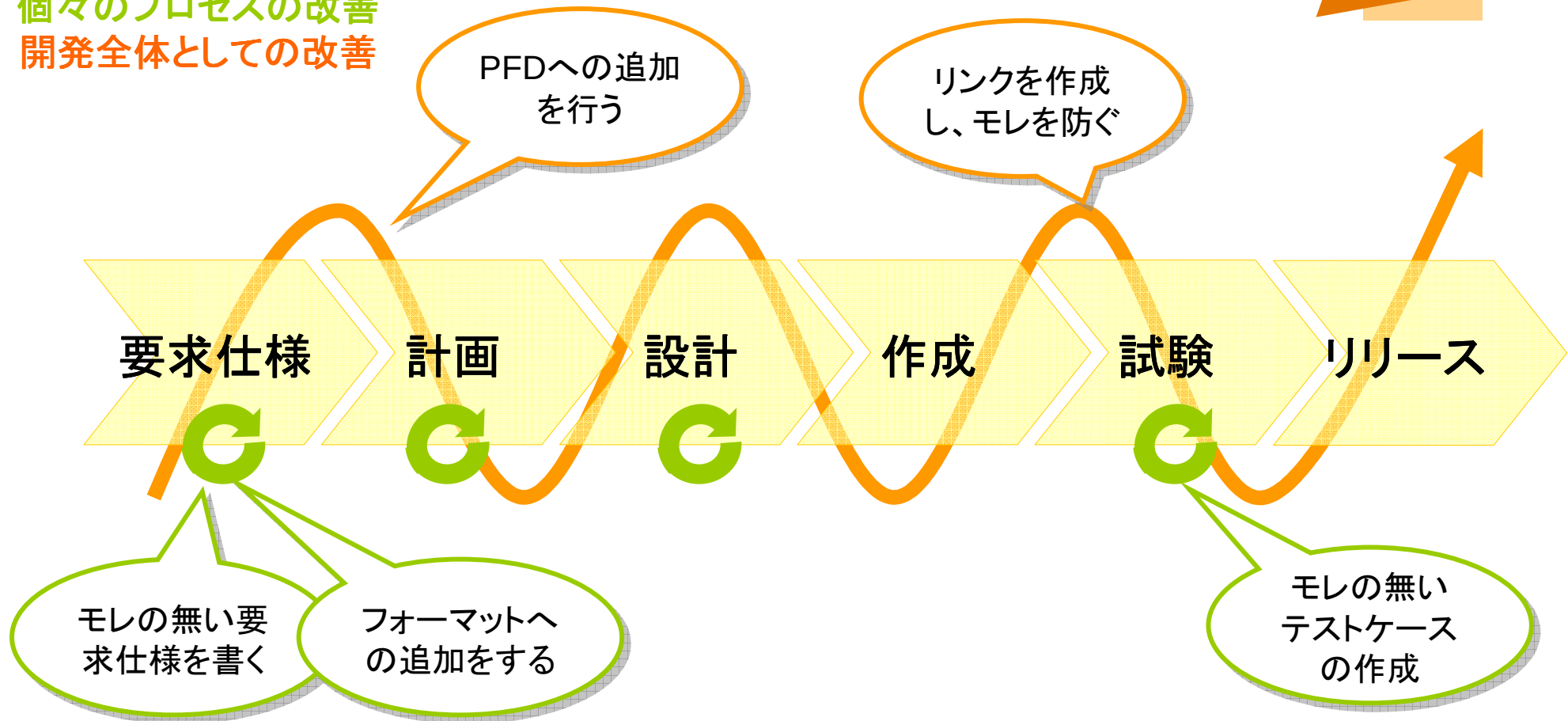


19

実践を継続していく



個々のプロセスの改善
開発全体としての改善



個々のプロセスだけではなく、開発全体の最適化を継続して行う

まとめ

1. 要求を仕様化するためには仕組みが必要であり、その仕組みを理解し実践することで完成度を上げていくことができる
2. 仕様を整理して書くことにより、設計やテストケースからモレを減らすことができる
3. 個々のプロセスの改善も必要であるが、上流工程を中心とした全体としての最適化がモレの減少につながる

要求仕様の完成度が上がれば テストケースも作成しやすい

開発プロセス導入前後の違い

	Before	After
テストケース	抜き出しにくい	抜き出しやすい
テストの目的	曖昧	明確
モレ	漏れやすい	漏れない

仕様が1行に
書かれているので

仕様が明確
なので

仕様と番号
でリンクして
いるので

開発者は試験書作成時に効果を実感している!!