

# 混乱からの目覚め ～ USDМとの出会い～

日本電気通信システム株式会社  
組込システムソリューション事業部  
岩松 洋史

# 概要

---

■ USDM作成時の弊社の取り組みをご説明します。

- USDM導入時の失敗
- USDM作成時に導入したプロセス

## ■ 目次

1. 背景
2. 問題点と原因
3. USDMの試行
4. 施策
5. 施策の結果
6. 考察
7. まとめ

# はじめに

USDMとWBSの中心前

1ヶ月間  
機能追加

2週間で、実装  
なんとかよろしく。

上司

を修正  
すればOKですね。

担当

.....

そして、3週間後

は仕様追加  
ですよ。

上司

どうすんだよ。  
処理が  
足りない...

担当

ドタバタ



現在

1ヶ月間  
機能追加

USDMで  
整理してね。

上司

の他  
も必要ですね。

担当

.....

そして、3週間後

順調だね。

上司

Bug減りました  
ね

担当

なんかうまく行った！！

# 自己紹介

---

名前: 岩松 洋史 (いわまつ ひろし)

所属: 日本電気通信システム株式会社

組込システムソリューション事業部

業務内容:

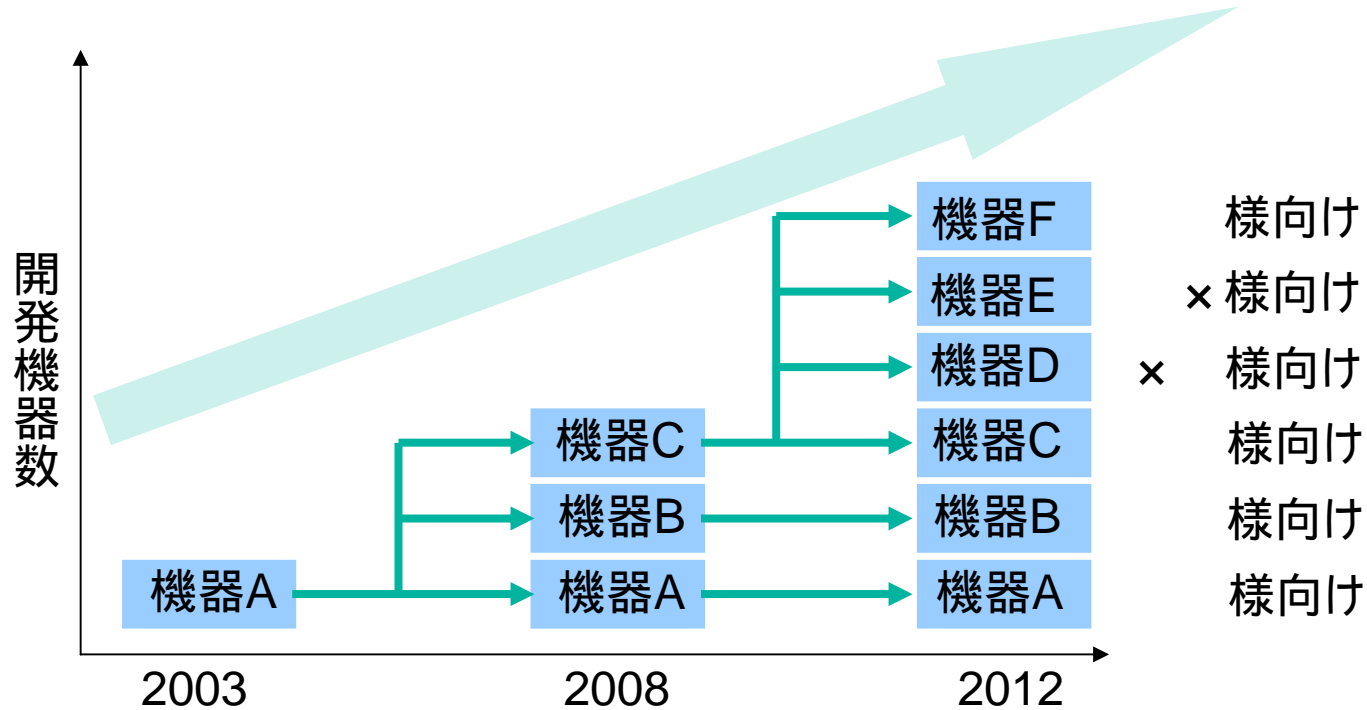
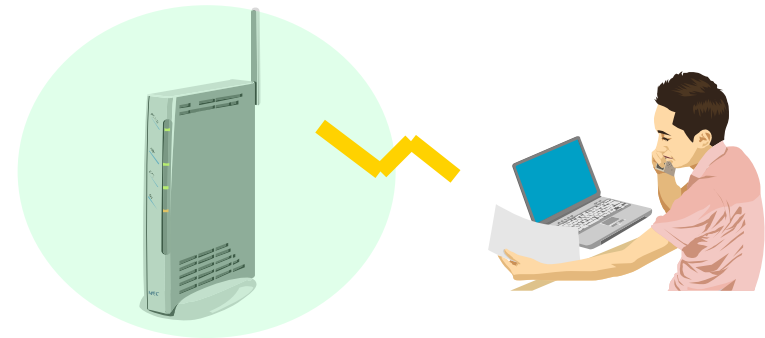
ネットワーク機器の組込みソフトウェア開発 プロジェクトリーダー



日本電気(株)我孫子事業場 (作業場所)

# 背景

- 現在の開発機器は2003年から開発
- 機器Aから派生した開発機器が増加
- 機能追加や変更で現在に至る

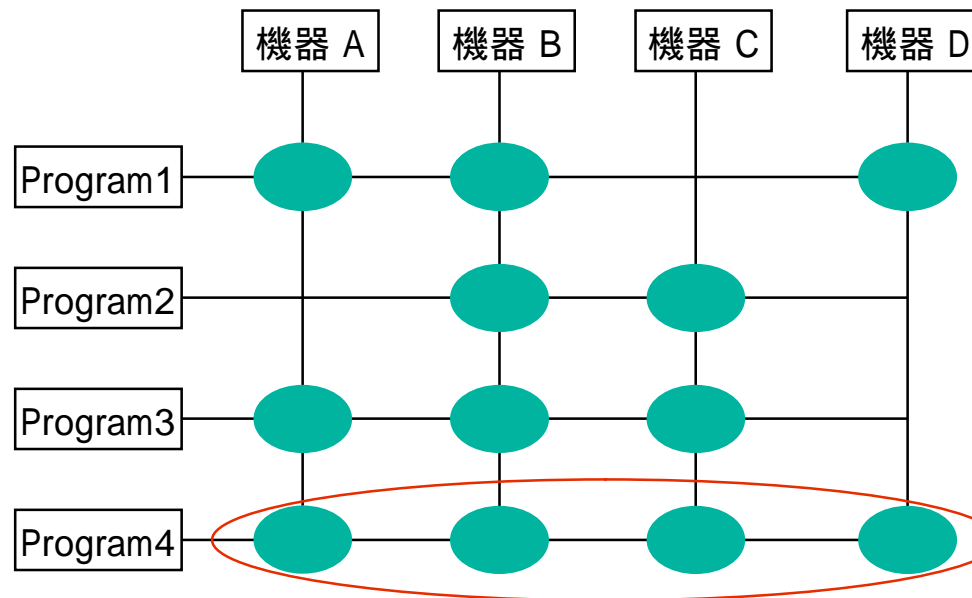


最近は派生開発  
が殆どです。

# 背景

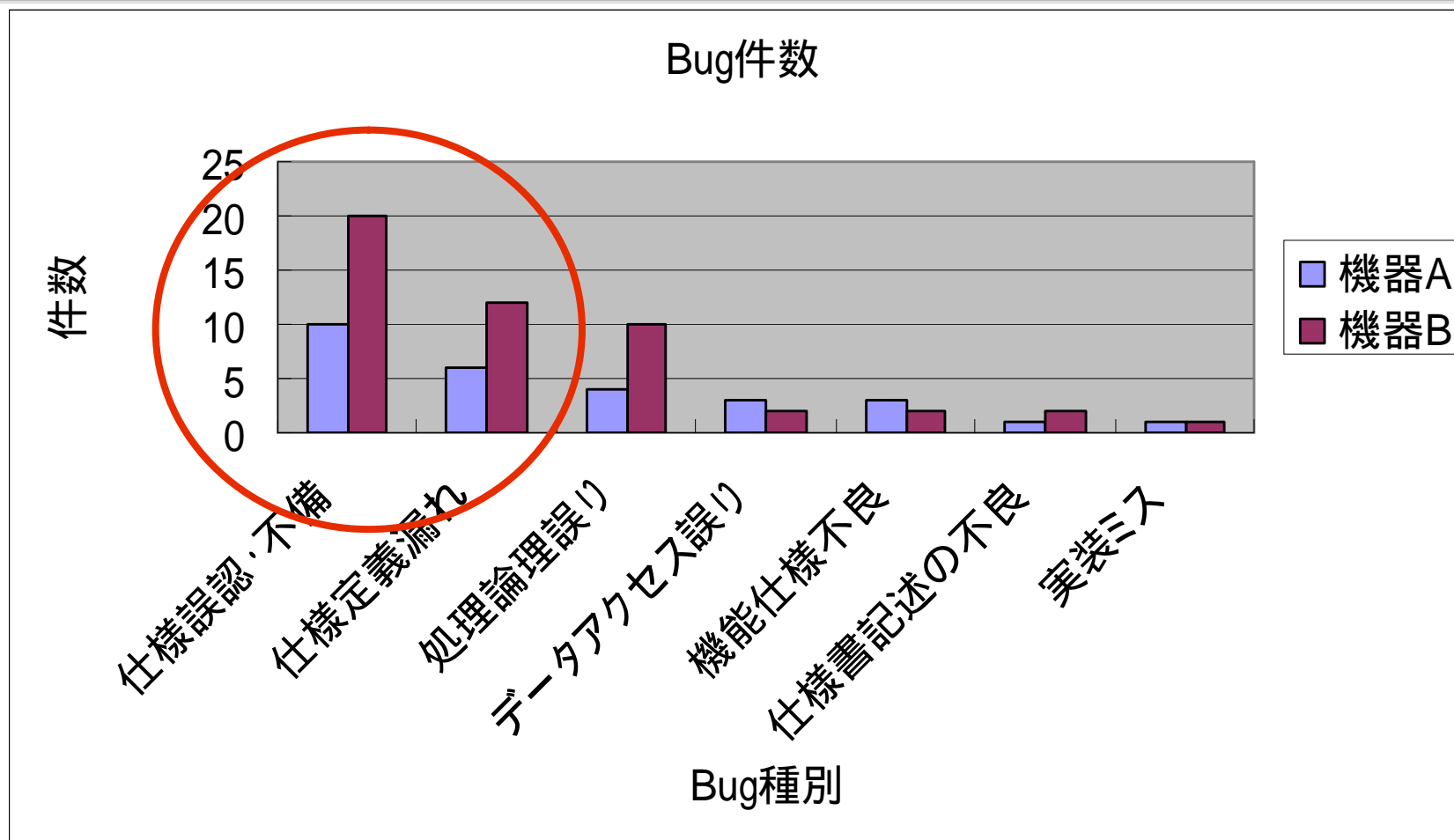
## マトリクス組織型

ある要員は複数の機器開発を掛け持ち



同じ人が複数機器の開発を担当。

# 問題点と原因



1. 仕様誤認バグが約30%
  2. 仕様定義漏れのバグが約20%
- 1, 2はいつもテストフェーズで対応

なんとか改善したい!

# 問題点と原因

---

## 1. 仕様誤認のバグが多発

➡ 関連(同様)機能の実装(修正)漏れ

- 要求の背景(目的、理由)が理解できておらず  
要求の影響範囲が特定できていない

## 2. 仕様定義漏れのバグが多発

➡ 既存のプログラムを変更したことによる他機能でのデグレード

- 開発母体の影響範囲が特定できていない

USDMを使うことで  
「要求」「仕様」「実装」を整理！



# USDMの試行

## USDMを書いてみました

要求	A	.....	
	理由	.....	
	説明	-	
要求	A-1		
	理由	-	
	説明	-	
<カテゴリ名>			
仕様	A-1-1		
要求	A-2		
	理由		
	説明	-	
<カテゴリ名>			
仕様	A-2-1		
<カテゴリ名>			
仕様	A-2-2		

要求と仕様が  
結び付かない

これって、  
関数単位？  
プログラム単位？

理由の所に  
仕様を書いている？

レビューしやすい文章に  
なっていない...

# USDMの試行

このままでは問題点の解決にならない・・・

要求	A	.....	
	理由	.....	
	説明	-	
要求	A-1		
	理由	-	
	説明	-	
<カテゴリ名>			
仕様	A-1-1		
要求	A-2		
	理由		
	説明	-	
<カテゴリ名>			
仕様	A-2-1		
<カテゴリ名>			
仕様	A-2-2		

要求仕様を書き写したら、  
要求と仕様が混在

->仕様誤認のバグは防げない

要求仕様の動詞から  
仕様に落とせない

->仕様の定義漏れの対策に  
ならない

# USD Mの試行

## USD Mのカイゼンポイントについて見直し

### カイゼンポイント

仕様や理由に要求事項が混在

プログラム単位 / 関数単位の記載が混在

何に影響があるのか記載ではわからない

要求から仕様に結び付かない

### 疑問点

要求と仕様の違いは？

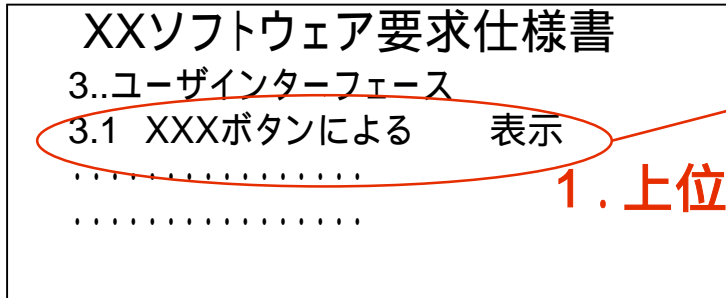
追加と変更の違いは？

影響範囲の特定の仕方はどうする？

仕様をソースコードレベルで書き落とすとはどうする？

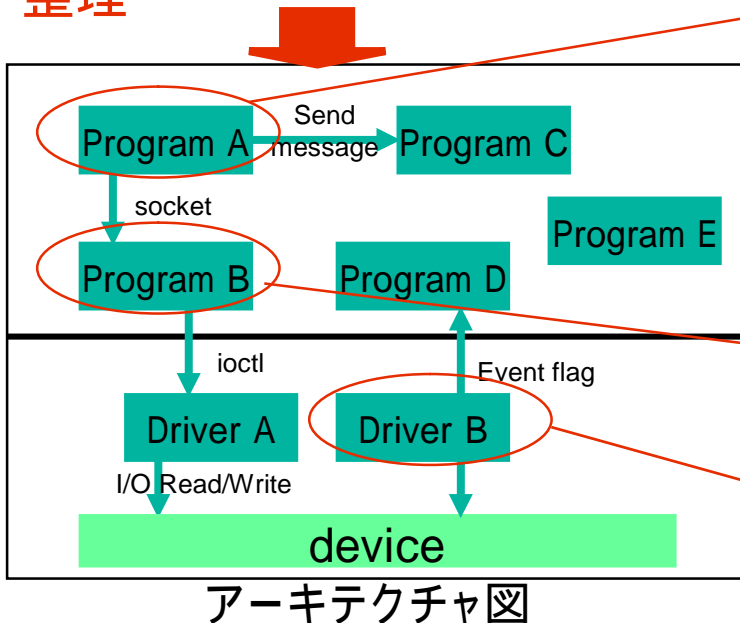
# 施策( のカイゼン)

## 要求事項を分解



1. 上位の要求項目を抽出

2. 機能分割を行い、アーキテクチャ図に整理

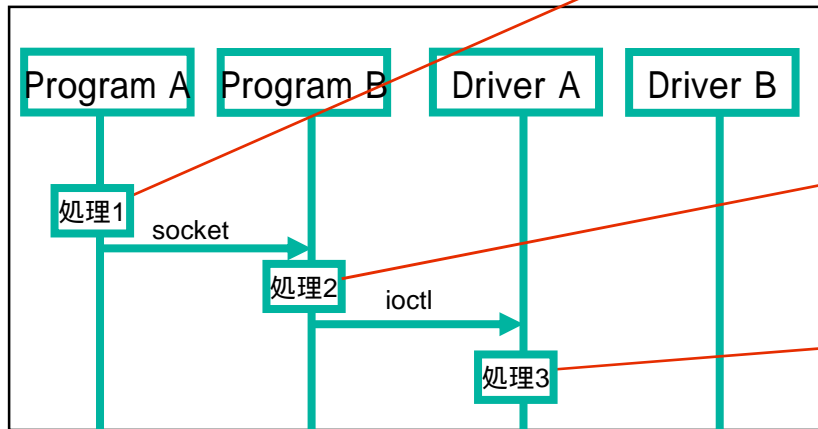
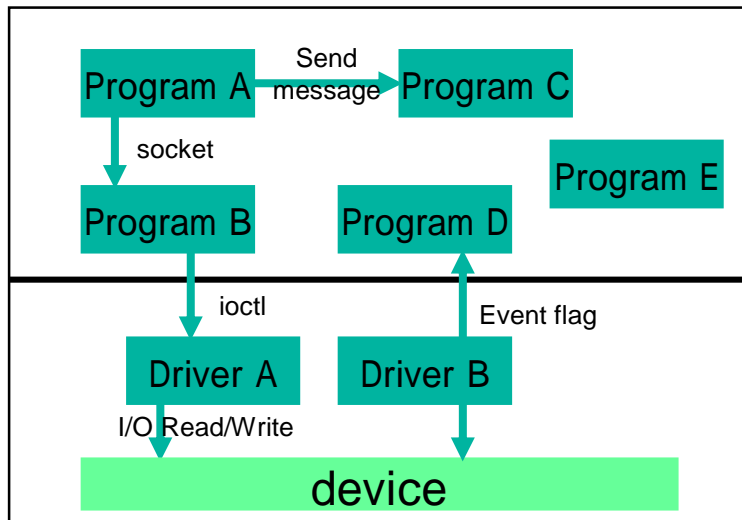


3. 関連するプログラムの要求に分解

要求	MENU	ユーザが画面を見ながら装置の各種設定を変更できるため
理由		使用頻度の高い機能を、ユーザーに使いやすくさせるため
説明		
要求	MENU01	SETボタン押下でメニューを表示し、
理由		本製品には、SELECTボタンとは、プロトタイプ開発で実装した
説明		
仕様	<アクション: 変更>	
<input type="checkbox"/>	MENU01-01	メニュー表示の条件を「SELECTボタン長押し」に変更する。
<input type="checkbox"/>	MENU01-02	メニュー項目決定(階層移動)の条件を「SELECTボタン長押し」に変更する。
<input type="checkbox"/>	MENU01-03	メニュー項目切替(カーソル移動)の条件を「SELECTボタン長押し」に変更する。
仕様	<イベント: 追加>	
<input type="checkbox"/>	MENU01-04	「SETボタン長押し(2秒経過)」のイベントを追加する。
仕様	<イベント: 削除>	
<input type="checkbox"/>	MENU01-05	「SELECTボタン押下」のイベントを全て削除する。
仕様	<アクション: 変更>	
<input type="checkbox"/>	MENU01-06	ボタン押下によるメニュー表示シーケンス中にメニューが表示されている時の振る舞いを変更する。 [変更前]: メニューを消去する。 [変更後]: メニューのカーソルを移動する。
<input type="checkbox"/>	MENU01-07	「自動LED消灯」からの復帰後、「SETボタン押下」によるメニュー表示シーケンス中にメニューが表示されている時の振る舞いを変更する。 [変更前]: メニューを消去する。 [変更後]: メニューのカーソルを移動する。
仕様	<状態遷移: 追加>	
<input type="checkbox"/>	MENU01-08	メニューの表示/非表示と同じタイミングで電源ボタン押下によるメニュー表示シーケンス中にメニューが表示されている時の振る舞いを変更する。 [変更前]: メニューを消去する。 [変更後]: メニューのカーソルを移動する。
要求	MENU02	電源ボタン押下で表示中のメニューを消去する。
理由		ユーザビリティを考慮すると、ワンタッチでメニューを消去できるようにする。
説明		
仕様	<アクション: 追加>	
<input checked="" type="checkbox"/>	MENU02-01	「電源ボタン押下(Edge)」イベント発生時メニューを消去する。
<input checked="" type="checkbox"/>	MENU02-02	「電源ボタン押下(Edge)」イベント発生時メニューを消去する。
要求	MENU03	一定時間タイムアウトで表示中のメニューを消去する。
理由		メニューを使用しない状態が続いた時は、メニューを消去する。
説明		

# 施策 ( のカイゼン)

## アーキテクチャ図からシーケンスに展開

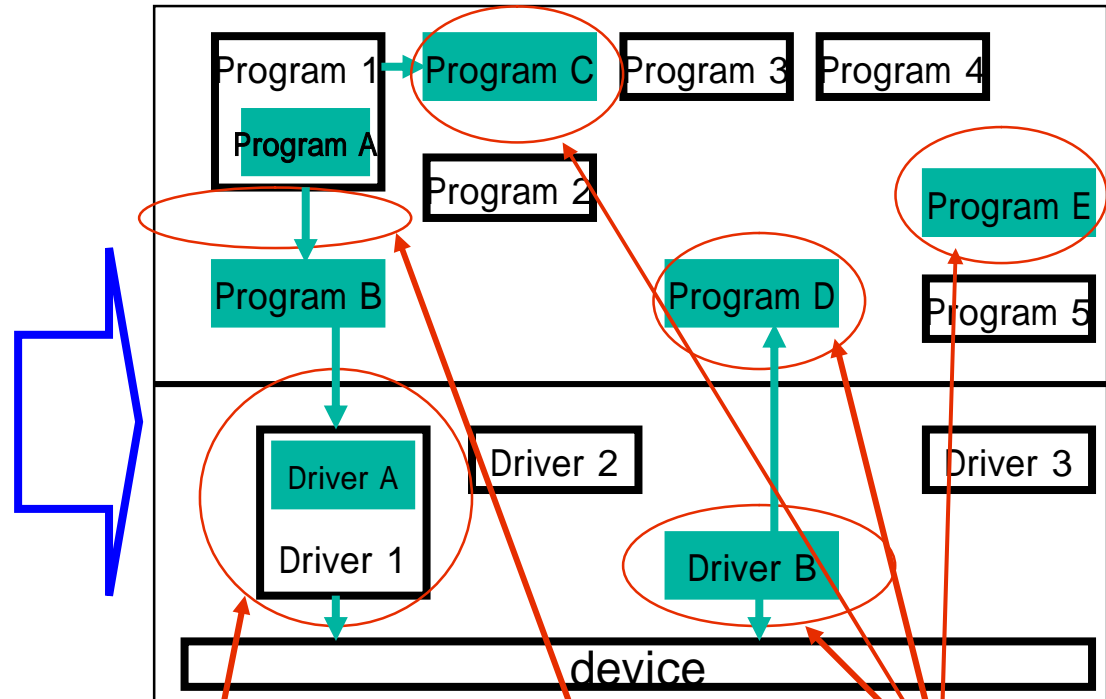
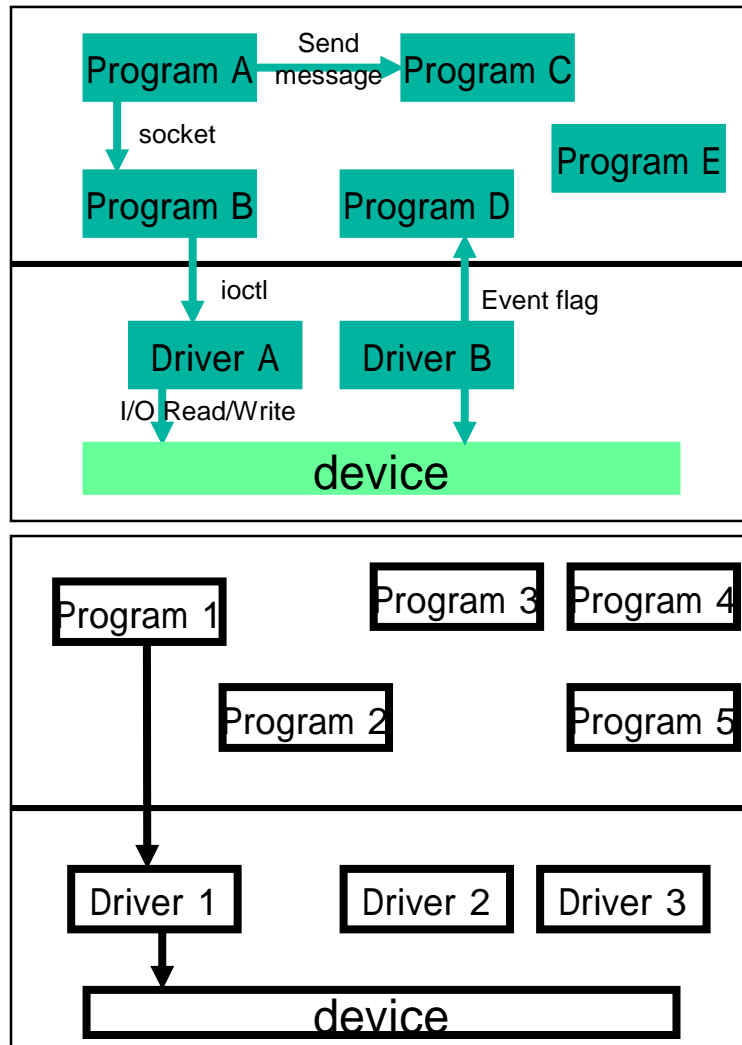


### 4. 仕様を抽出

要求	MENU	ユーザが画面を見ながら装置の各種設定を変更できるメ
理由		使用頻度の高い機能を、ユーザーに使いやすくさせるため
説明		
要求	MENU01	SETボタン押下でメニューを表示する
理由		メニュー表示の条件を「SELECTボタ
説明		メニューとは、プロトタイプ開発で実装した
仕様	<アクション: 変更>	
<input type="checkbox"/>	MENU01-01	メニュー表示の条件を「SELECTボタン長
<input type="checkbox"/>	MENU01-02	メニュー項目決定(階層移動)の条件を「S
<input type="checkbox"/>	MENU01-03	メニュー項目切替(カーソル移動)の条件を
仕様	<イベント: 追加>	
<input type="checkbox"/>	MENU01-04	「SETボタン長押し(2秒経過)」のイベント
仕様	<イベント: 削除>	
<input type="checkbox"/>	MENU01-05	「SELECTボタン押下」のイベントを全て削
仕様	<アクション: 変更>	
<input type="checkbox"/>	MENU01-06	ボタン押下によるメニュー表示シーケンス 既にメニューが表示されている時の振る舞 [変更前]: メニューを消去する。 [変更後]: メニューのカーソルを移動する
<input type="checkbox"/>	MENU01-07	「自動LED消灯」からの復帰後、「SETボタ
仕様	<状態遷移: 追加>	
<input type="checkbox"/>	MENU01-08	メニューの表示/非表示と同じタイミング
要求	MENU02	電源ボタン押下で表示中のメニューを消
理由		ユーザビリティを考慮すると、ワンタッチで
説明		
仕様	<アクション: 追加>	
<input checked="" type="checkbox"/>	MENU02-01	「電源ボタン押下(Edge)」イベント発生時の
<input checked="" type="checkbox"/>	MENU02-02	「電源ボタン押下(Edge)」イベント発生時の
要求	MENU03	一定時間タイムアウトで表示中のメニュー
理由		メニューを使用しない状態が続いた時は、
説明		

# 施策 ( , のカイゼン)

要求仕様から作成したアーキテクチャ図と開発母体のアーキテクチャ図をマージ



変更と追加が混在  
(影響の出る場所)

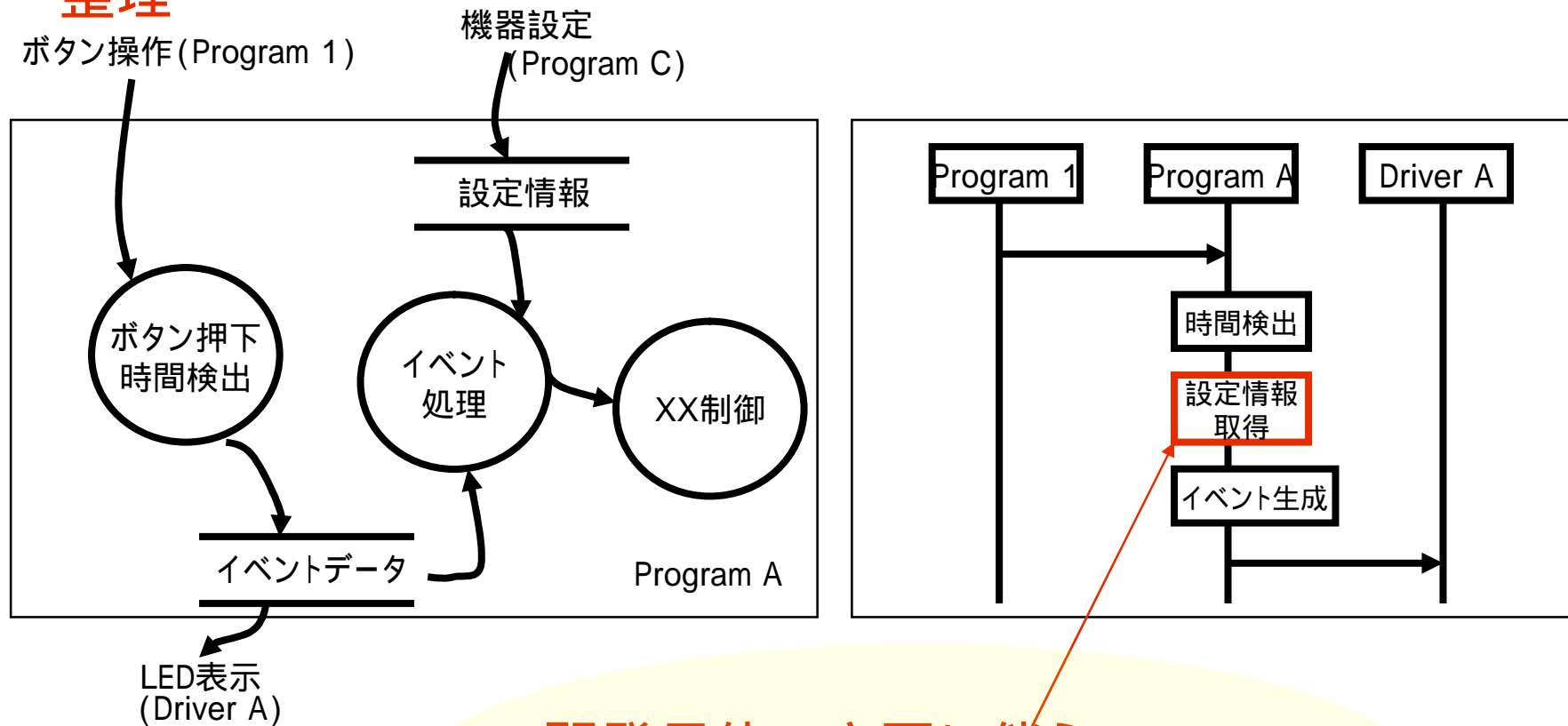
変更  
(影響の出る場所)

追加

開発母体の影響を受ける所を特定

# 施策( , のカイゼン)

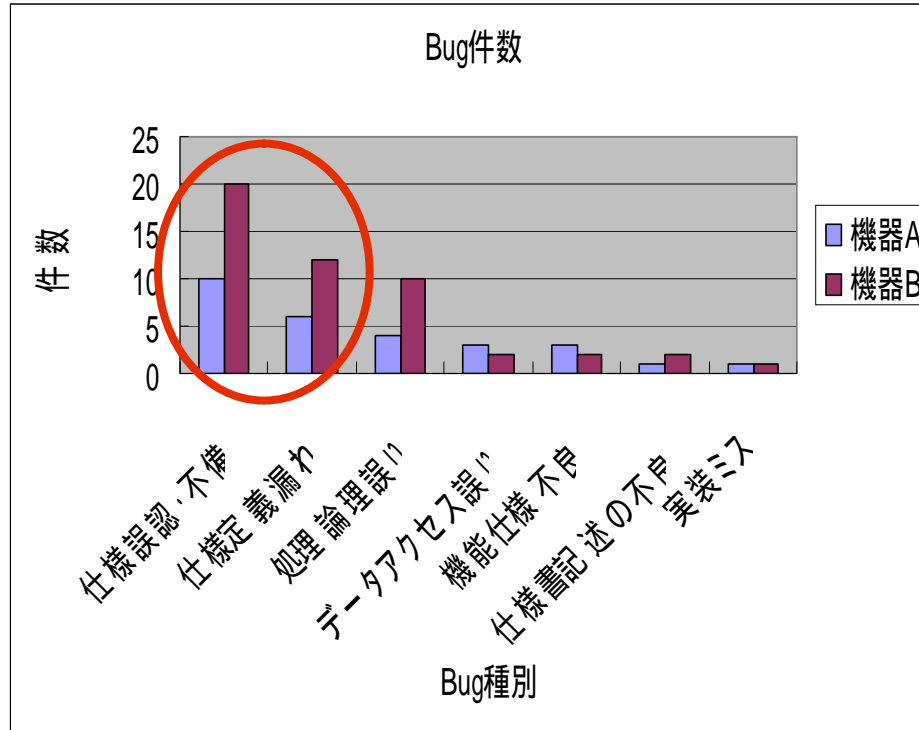
## 機能ブロック間のデータフローとタイミングをDFDとシーケンス図で整理



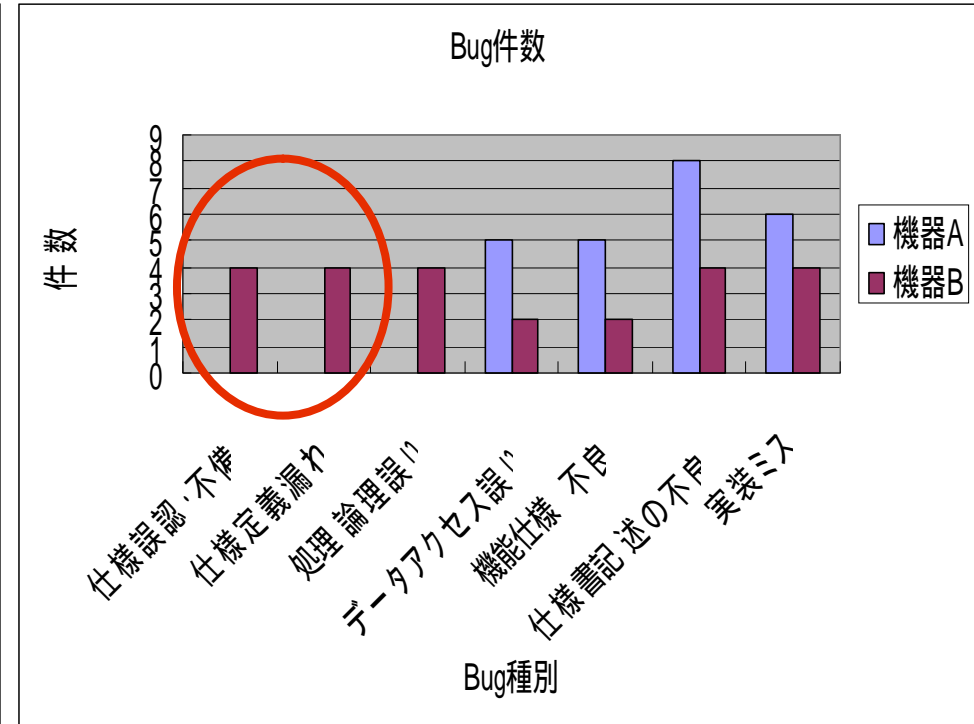
開発母体の変更に伴う  
影響箇所特定  
得られた結果をUSDMにマージ

# 施策の結果(定量的効果)

## USDM適用前



## USDM適用後



仕様誤認や不備によるバグが  
7件/KLから1件/KLに減少



## 施策の結果(定性的効果)

---

### 問題点1:仕様誤認のバグが多発

- 実装漏れのバグが7件/KLから1件/KL以下に減少した。
- 今まで仕様化できていなかった所を仕様として明記することができた。

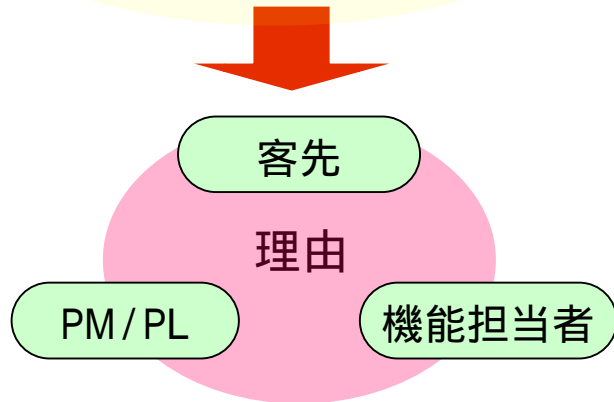
### 問題点2:仕様定義漏れのバグが多発

- デグレードのバグが4件/KLから1件/KLに減少した。
- アーキテクチャ図に追加と変更を整理することで影響範囲の見える化ができるようになった。

# 施策の結果 (作成したUSDMの確認)

## 作成したUSDMを開発要求元とレビュー

理由の記載を重点的に開発要求元とレビューを行う。  
**要求に対する何故の部分**を関係者で共有する



要求事項の実装範囲や影響範囲を関係者で認識

要求	MENU	ユーザーが画面を見ながら任意の各種設定を変更できる
	理由	使用頻度の高い機能を、ユーザーに使いやすくさせるため
	説明	
要求	MENU01	SETボタン押下でメニューを切り替え可能
	理由	メニュー項目決定(階層移動)には、SELECTボタンを使用する
	説明	メニューとは、プロトタイプ開発で実装した
仕様	<アクション: 変更>	
<input type="checkbox"/>	MENU01-01	メニュー表示の条件を「SELECTボタン長押し」に変更
<input type="checkbox"/>	MENU01-02	メニュー項目決定(階層移動)の条件を「SELECTボタン」に変更
<input type="checkbox"/>	MENU01-03	メニュー項目切替(カーソル移動)の条件を「SELECTボタン」に変更
仕様	<イベント: 追加>	
<input type="checkbox"/>	MENU01-04	「SETボタン長押し(2秒経過)」のイベントを追加
仕様	<イベント: 削除>	
<input type="checkbox"/>	MENU01-05	「SELECTボタン押下」のイベントを全て削除
仕様	<アクション: 変更>	
<input type="checkbox"/>	MENU01-06	ボタン押下によるメニュー表示シーケンス時にメニューが表示されている時の振る舞いを変更 [変更前]: メニューを消去する。 [変更後]: メニューのカーソルを移動する
<input type="checkbox"/>	MENU01-07	「自動LED消灯」からの復帰後、「SETボタン」を押下するとメニューが表示される
仕様	<状態遷移: 追加>	
<input type="checkbox"/>	MENU01-08	メニューの表示/非表示と同じタイミングでメニューを消去する
要求	MENU02	電源ボタン押下で表示中のメニューを消去する
	理由	ユーザビリティを考慮すると、ワンタッチでメニューを消去できるようにする
	説明	
仕様	<アクション: 追加>	
<input checked="" type="checkbox"/>	MENU02-01	「電源ボタン押下(Edge)」イベント発生時にメニューを消去する
<input checked="" type="checkbox"/>	MENU02-02	「電源ボタン押下(Edge)」イベント発生時にメニューを消去する
要求	MENU03	一定時間タイムアウトで表示中のメニューを消去する
	理由	メニューを使用しない状態が続いた場合は、メニューを消去する
	説明	

## 1) USDM導入に関して

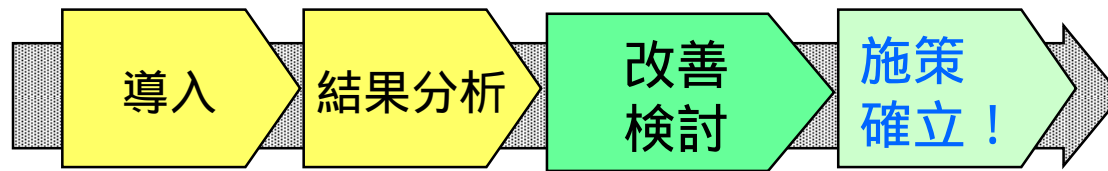
- USDMは追加と変更をばらすツールとして有効。
- 仕様を階層構造で落とすには設計技術も絡める必要がある。
- 理由を関係者が共有することで、変更の範囲を強く意識出来る。

## 2) 今後について

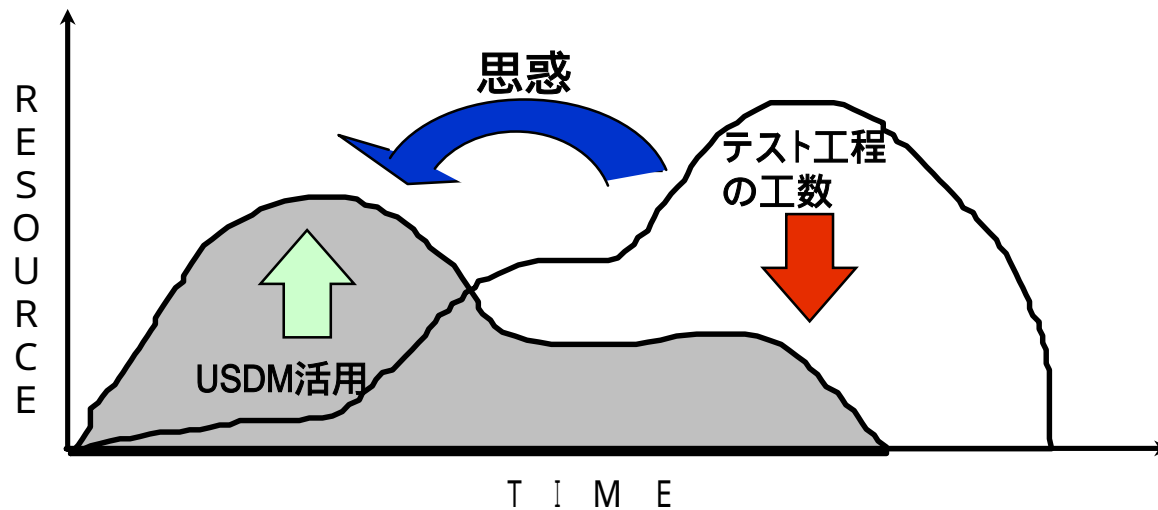
- 成功事例を増やし、適用範囲を開発作業全体に広げて行く。

# まとめ

## USDMを導入して要求仕様書の影響範囲を整理 混乱状態からの脱却



要求仕様書の整理を行わず  
ソースコード修正をしていたため、  
開発終盤で混乱！



USDMで要求仕様書の整理  
要件開発手法、現有資産を  
活用！

客先要求の分析作業と  
開発母体の把握作業の  
導入

アーキテクチャ図

+

シーケンス図

+

DFD

=> [影響範囲の見える化]

# まとめ

USDMとWBSの前

2週間で、実装  
なんとかよろしく。

上司

を修正  
すればOKですね。

担当

.....

そして、3週間後

は仕様追加  
ですよ。

上司

どうすんだよ。  
処理が  
足りない...

担当

ドタバタ



今後

変更内容を把握

USDMに整理

担当

じゃあ、  
はB君に。

上司

だけでなく、  
の修正も  
必要ですね。

.....

そして、3週間後

お客様に  
褒められちゃった。

上司

HAPPY!

今日、飲み  
に行きませんか？

担当

---

ご静聴ありがとうございました。

Empowered by Innovation

**NEC**