

生涯バグの統計的予測による品質水準値の算出方法の
標準化および汎用的な統計的品質分析シートの構築

2018年10月11日
株式会社NTTデータ アイ
中村 泰幸

もくじ

1. はじめに
2. 背景
3. 改善したいこと
4. 制約条件
5. 改善策の概要
6. 改善策の実現方法
7. 運用について
8. 改善による効果
9. 今後に向けて

1.はじめに

生涯バグ数（※）を統計的に予測し、その生涯バグ数を基にした品質水準値の算出方法の標準化を行いましたのでご紹介します。

合わせて汎用的な統計的品質分析シートを構築しましたのでこちらについてもご紹介いたします。

※生涯バグ数・・・ソフトウェアライフサイクル全体で検出される全バグ数

<対象組織>

官公庁が運用するオンラインシステムの業務開発を行う組織（複数の開発グループあり）

<プロセス改善活動の特徴>

- 開発のためのCMMI（CMMI-DEV）に基づく改善活動を実施
- 統計的手法を活用した管理を実施

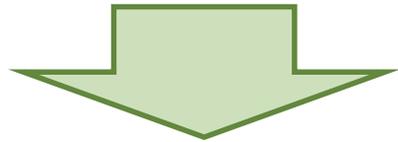
2.背景

2.背景

従来、統計的な予測モデルの構築は、個々のシステム開発グループごとに策定

2.背景

従来、統計的な予測モデルの構築は、個々のシステム開発グループごとに策定



開発グループ、
SEPG双方の検
討、運用にかかる
負荷が大きい

2.背景

従来、統計的な予測モデルの構築は、個々のシステム開発グループごとに策定

開発グループ、SEPG双方の検討、運用にかかる負荷が大きい

組織全体で使用できる標準的な予測モデル構築が急務

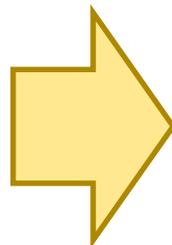
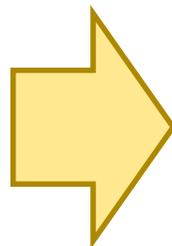
3.改善したいこと

3.改善したいこと

従来

①品質水準値の算出方法が開発グループ毎に異なる

②サービス開始後6か月バグ密度を予測する統計的モデルがない



あるべき姿
(ニーズ)

品質水準値の算出方法を統計的手法を使って標準化

サービス開始後6か月バグ密度を予測し、コントロールできる統計的予測モデルを構築

3.改善したいこと

従来

③統計ツールの利用は、
統計手法の知識が必要
となるため、統計の専
門家が分析を実施

④分析方法は、開発グ
ループ毎に異なる

あるべき姿
(ニーズ)

分析シートへ統計に関す
る数式を組み込むことで
統計ツールを使用しなく
ても誰でも分析可能

開発グループによらず使
用可能な汎用的な分析
シートを構築

4.制約条件

4.制約条件

①開発グループに負荷を与えないようにするため、新たなデータの収集はしないこと（例 スキル、難易度など）

②統計的予測モデルのパラメータ数は、最小限とすること。
（パラメータ数が多いと実運用時、開発グループのデータ収集に負荷がかかるため）

③CMMIモデルからの制約条件

- 目標（サービス開始後6か月バグ密度）との関係が数式的に結びついている統計的予測モデルであること。
- コントロールできるパラメータが 1つ以上あること。
- パラメータのうち1つ以上は、管理図でばらつきがコントロールされていること。
- 統計的予測モデルは、95%以上の精度があること

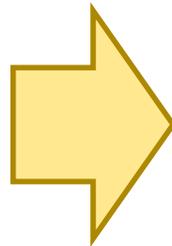
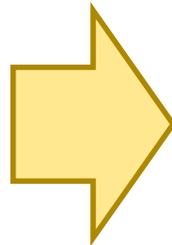
5.改善策の概要

5.改善策の概要

あるべき姿
(ニーズ)

①品質水準値の算出方法を統計的手法を使って標準化

②サービス開始後6か月バグ密度を予測し、コントロールできる統計的予測モデルを構築



改善策

(a)生涯バグを統計的に予測するモデルを構築

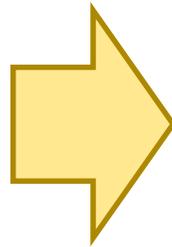
(b)工程毎の抽出バグ数 = 生涯バグ × 工程比率により各工程の品質水準値を算出

(c)生涯バグ - 抽出バグ = 残存バグにより、サービス開示後6か月バグ密度を予測

5.改善策の概要

あるべき姿
(ニーズ)

②サービス開始後6か月
バグ密度を予測し、コン
トロールできる統計的予
測モデルを構築



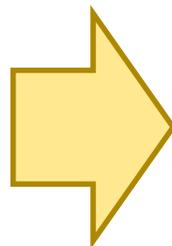
改善策

(d)工程毎の目標バグ数
を抽出するために必要な
試験項目数 (コントロー
ルするためのパラメー
タ)を算出する統計的予
測モデルを構築

5.改善策の概要

あるべき姿
(ニーズ)

③分析シートへ統計に関する数式を組み込むことで統計ツールを使用しなくても**誰でも分析可能**



改善策

(e)統計分析に関する数式を埋め込んだ**統計的品質分析シート**を作成

5.改善策の概要

あるべき姿
(ニーズ)

④開発グループによらず
使用可能な汎用的な分析
シートを構築

収集データ項目

- 設計工程
設計書の出来高頁数、エラー数
- 試験工程
試験項目数、バグ数

改善策

(f)統計的品質分析シートは、他開発グループにも展開できるように使用するデータ項目は、どの開発グループでも収集しているデータ項目に限定した。また、工程も自由に変更可能にした。

6.改善策の実現方法

6.改善策の実現方法

- (a)生涯バグ数の予測モデル
- (b)工程毎のバグ摘出数の算出
- (c)サービス開始後 6 か月バグ密度を予測
- (d)試験項目数を算出するモデル式
- (e)(f)汎用的な分析シート

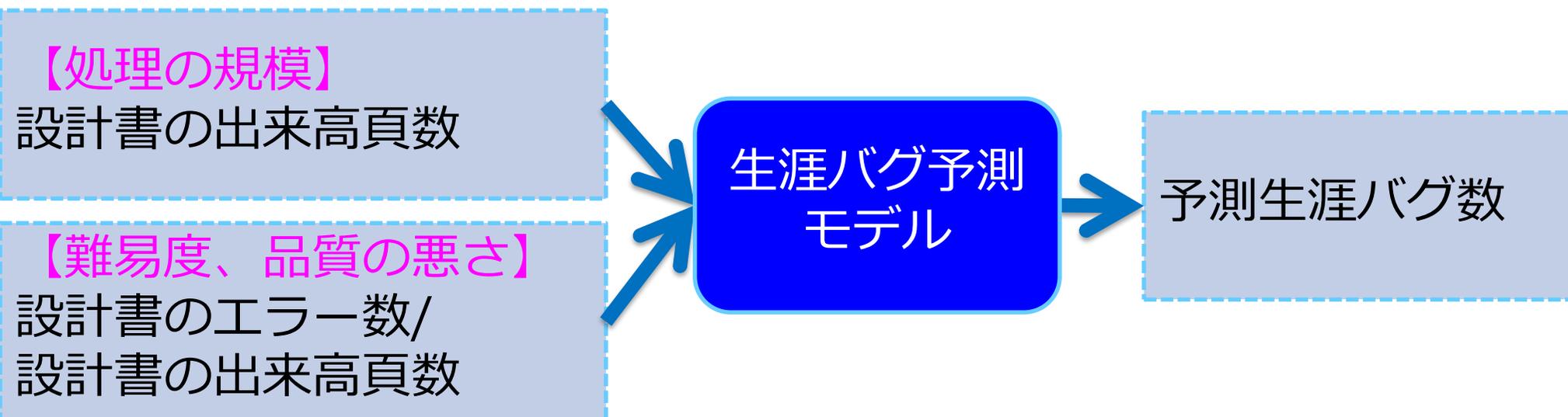
- (a)生涯バグ数の予測モデル
- (b)工程毎のバグ摘出数の算出
- (c)サービス開始後 6 か月バグ密度を予測
- (d)試験項目数を算出するモデル式
- (e)(f)汎用的な分析シート

6.改善策の実現方法

(a)生涯バグ数の予測モデル

Input

Output



6.改善策の実現方法

(a)生涯バグ数の予測モデル

Input

Output

【処理の規模】

設計書の出来高頁数

【難易度、品質の悪さ】

設計書のエラー数/
設計書の出来高頁数

バグ数

なるべく早い工程で抽出バグ数を予測したいため、処理の規模は、設計書の出来高頁数を使用

6.改善策の実現方法

(a)生涯バグ数の予測モデル

Input

Output

【処理の規模】

設計書の出来高頁数

【難易度、品質の悪さ】

設計書のエラー数/
設計書の出来高頁数

処理の規模以外に**難易度、品質の悪さ**が生涯バグ数に影響すると考えられる※

バグ数

※パラメータの選択は、収集した幾つかのデータの組み合わせの中から設計書のエラー数/設計書の出来高頁数を選択した

6.改善策の実現方法

(a)生涯バグ数の予測モデル

$$\text{予測生涯バグ数} = \underbrace{\text{処理の規模}^\beta}_{\text{処理の規模}} \times e^{\underbrace{\beta_0 + \beta_1 \cdot X_1 + \dots + \beta_n \cdot X_n}_{\text{パラメータ}}}$$

※ e : 自然対数の底 2.71828

典型的には上記のようなモデル式となる。

6.改善策の実現方法

(a)生涯バグ数の予測モデル

予測生涯バグ数

$$= \underbrace{\text{設計書の出来高頁数}}_{\beta} \times e^{\underbrace{\beta_0 + \beta_1 \cdot \text{設計書のエラー数/出来高頁数}}_{\text{難易度、品質の悪さ}}}$$

処理の規模

難易度、品質の悪さ

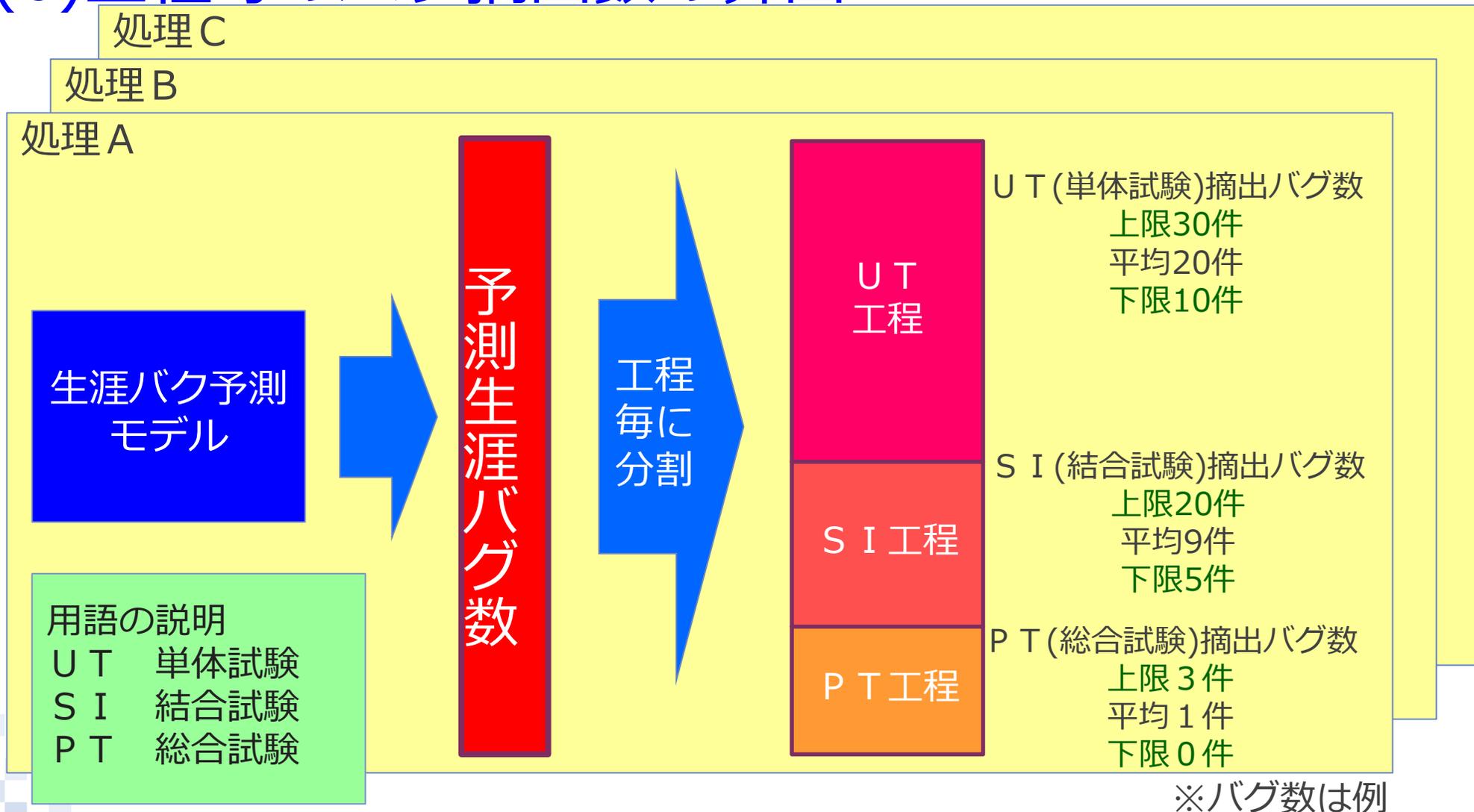
β 、 β_0 、 β_1 は定数。過去データから統計分析で算出

6.改善策の実現方法

- (a)生涯バグ数の予測モデル
- (b)工程毎のバグ摘出数の算出
- (c)サービス開示後 6 か月バグ密度を予測
- (d)試験項目数を算出するモデル式
- (e)(f)汎用的な分析シート

6.改善策の実現方法

(b)工程毎のバグ摘出数の算出



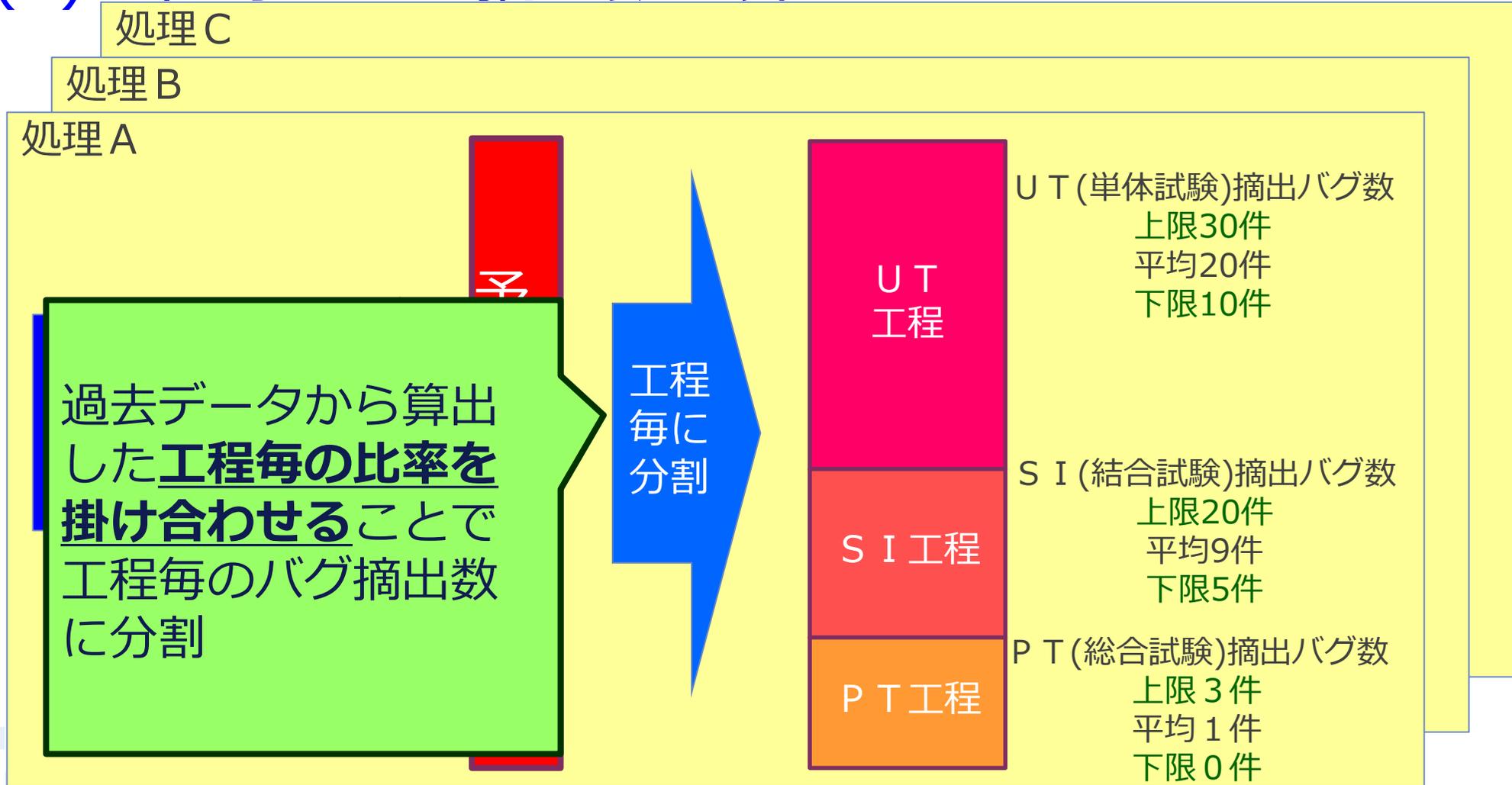
6.改善策の実現方法

(b)工程毎のバグ摘出数の算出



6.改善策の実現方法

(b)工程毎のバグ摘出数の算出



※バグ数は例

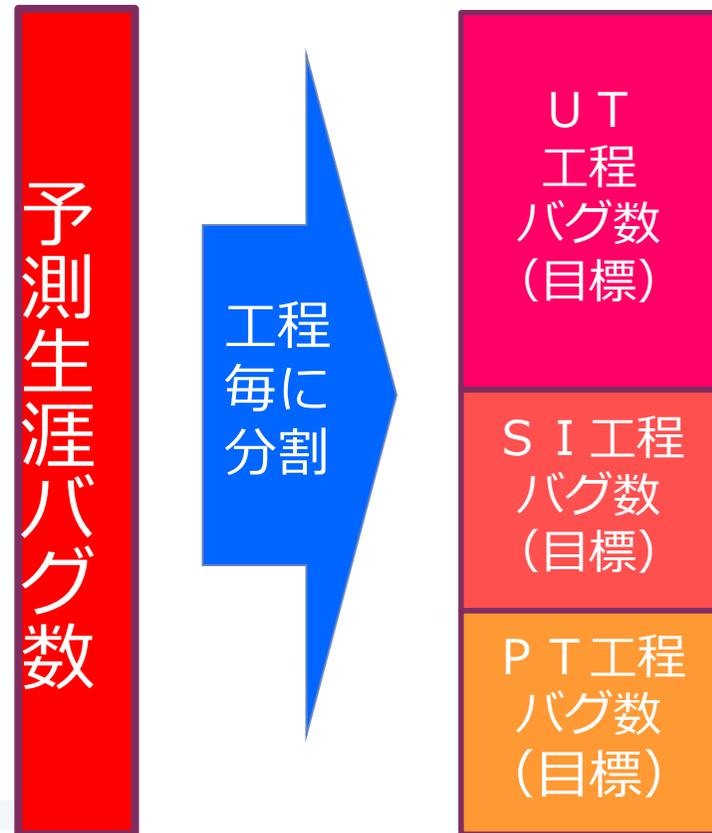
6.改善策の実現方法

- (a)生涯バグ数の予測モデル
- (b)工程毎のバグ摘出数の算出
- (c)サービス開始後 6 か月バグ密度を予測
- (d)試験項目数を算出するモデル式
- (e)(f)汎用的な分析シート

6.改善策の実現方法

(c)サービス開始後 6 か月バグ密度を予測

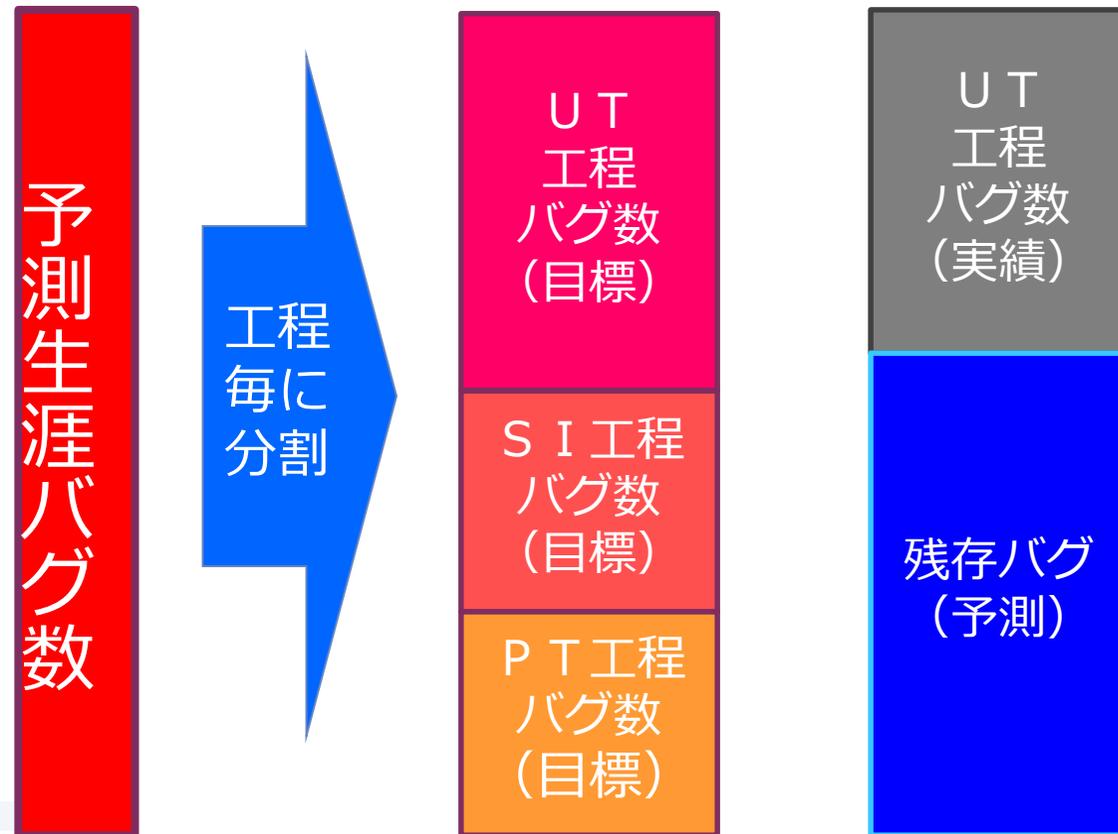
設計工程完了時



6.改善策の実現方法

(c)サービス開始後 6 か月バグ密度を予測

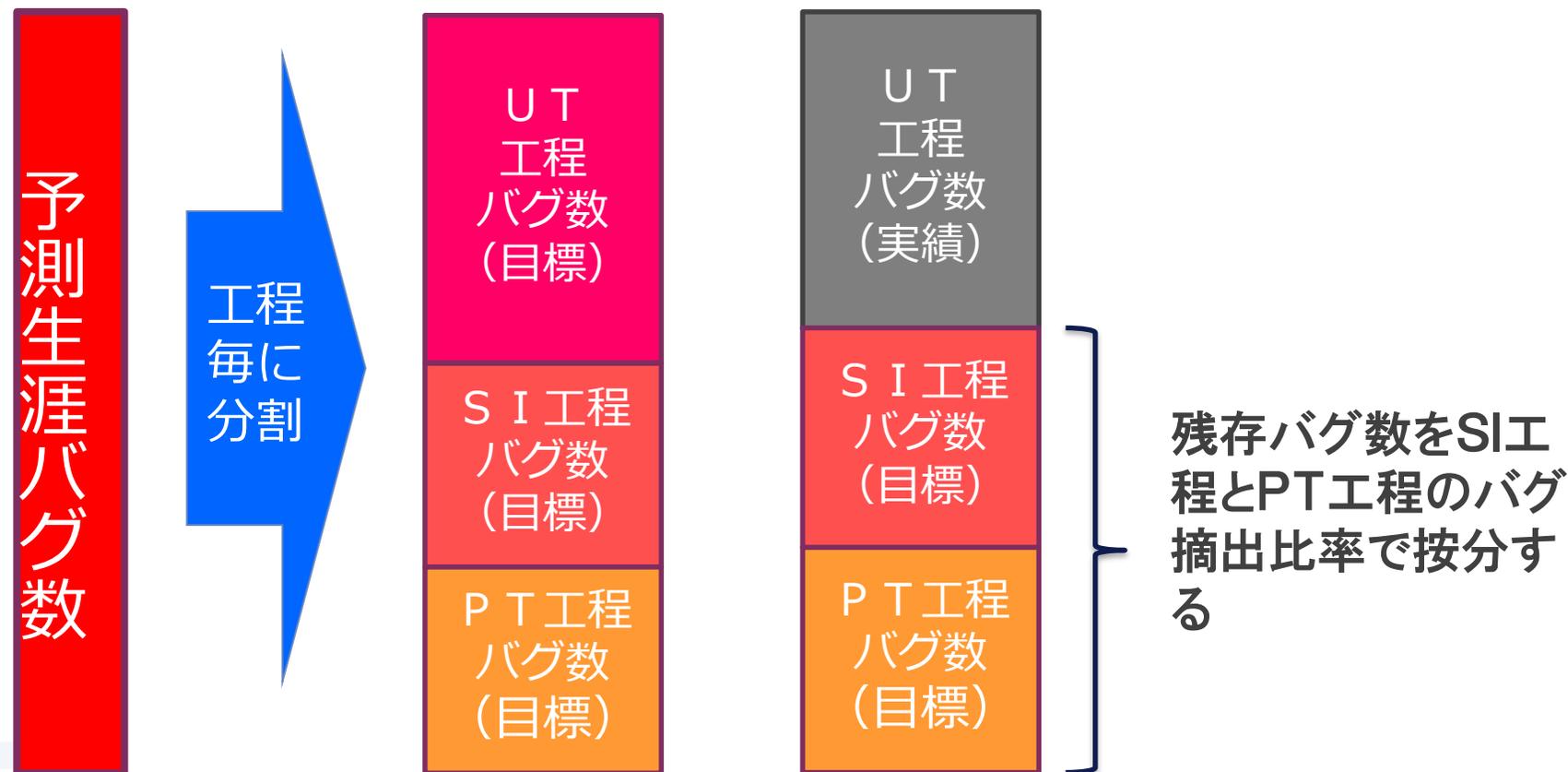
設計工程完了時 U T 工程完了時



6.改善策の実現方法

(c)サービス開始後 6 か月バグ密度を予測

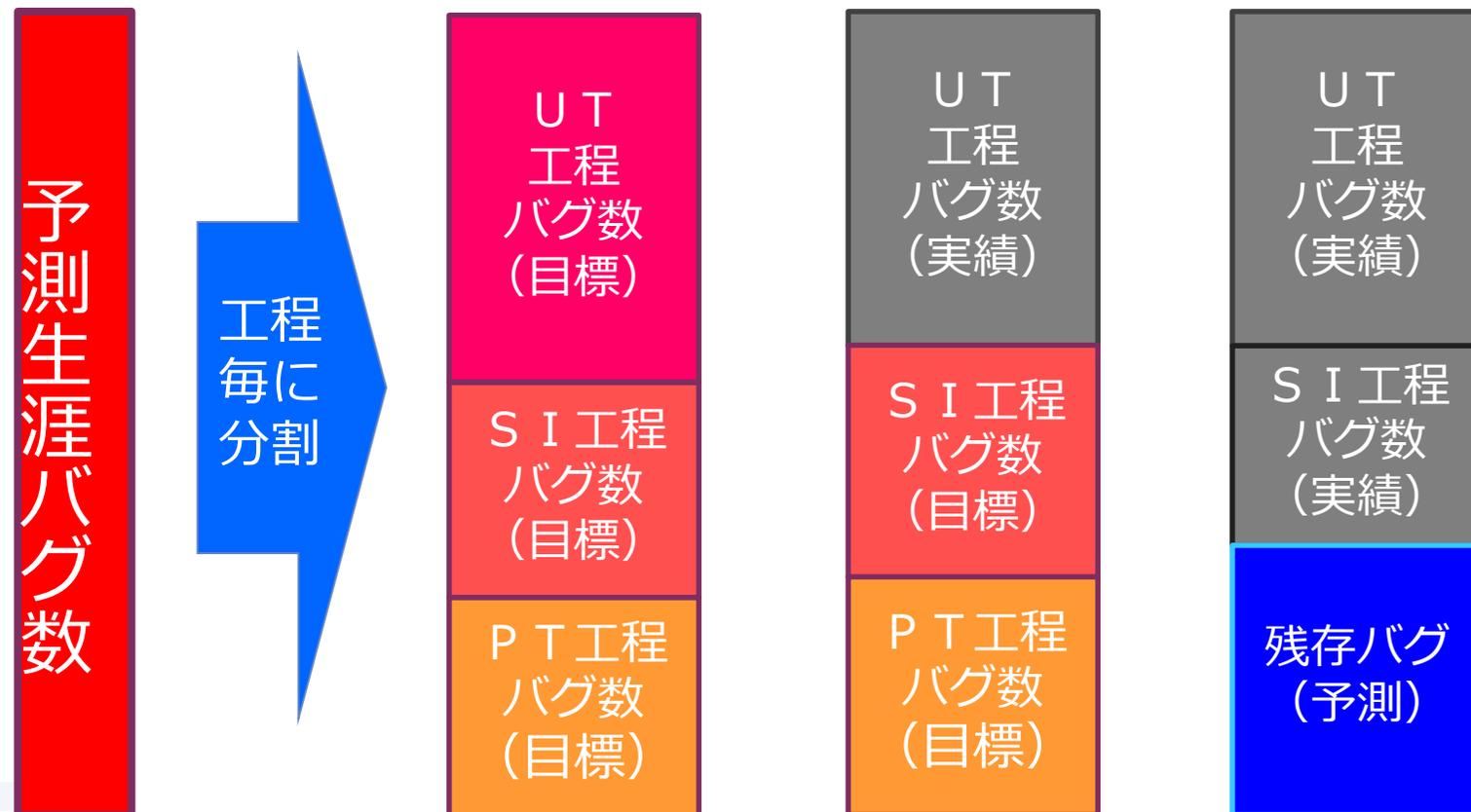
設計工程完了時 U T 工程完了時



6.改善策の実現方法

(c)サービス開始後 6 か月バグ密度を予測

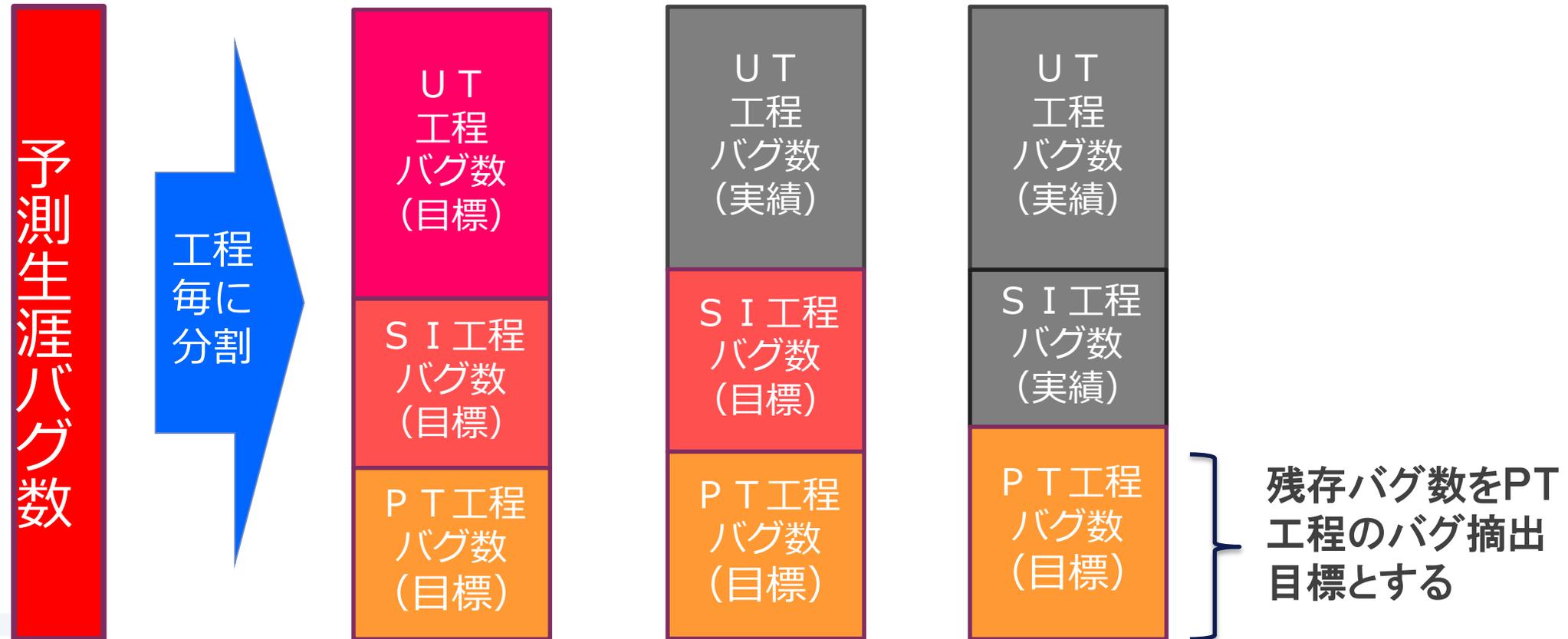
設計工程完了時 U T 工程完了時 S I 工程完了時



6.改善策の実現方法

(c)サービス開始後6か月バグ密度を予測

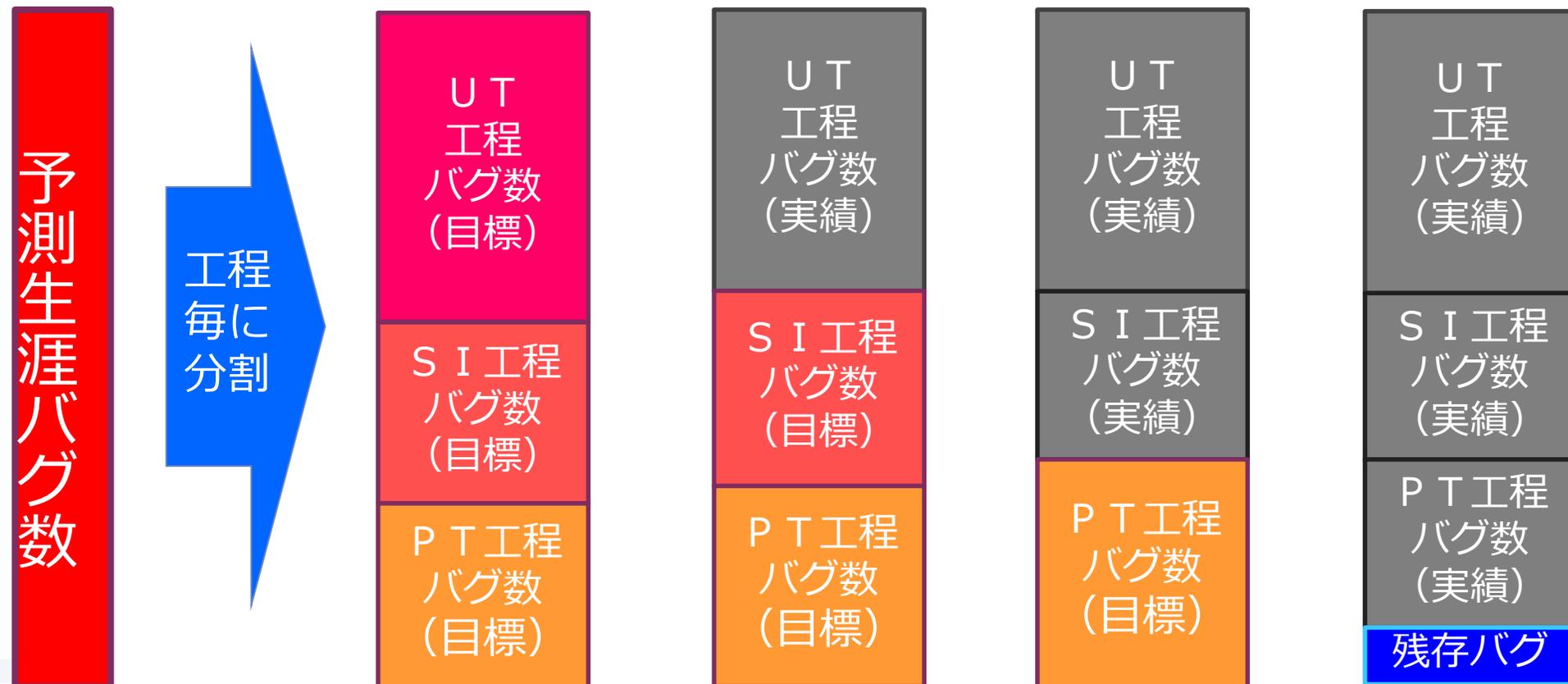
設計工程完了時 UT工程完了時 SI工程完了時



6.改善策の実現方法

(c)サービス開始後6か月バグ密度を予測

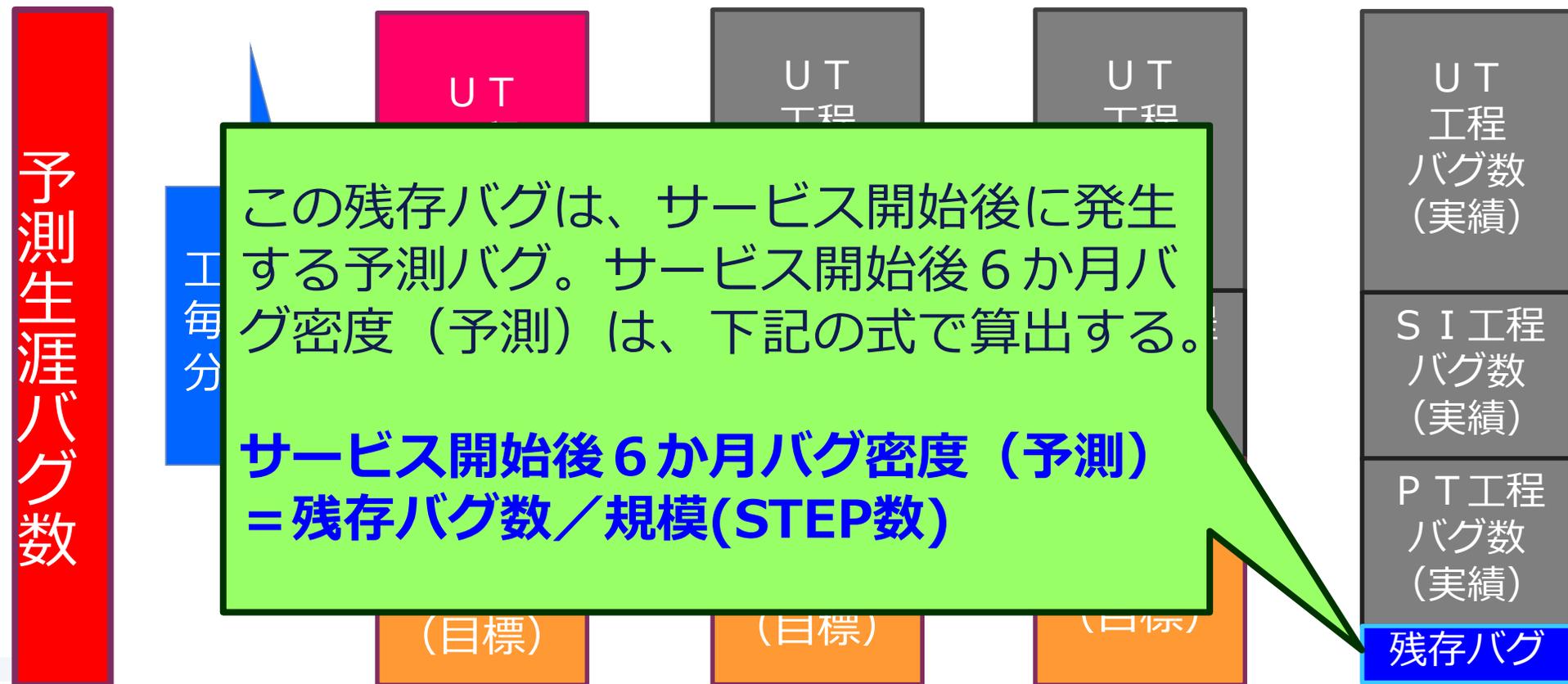
設計工程完了時 UT工程完了時 SI工程完了時 PT工程完了時



6.改善策の実現方法

(c)サービス開始後6か月バグ密度を予測

設計工程完了時 UT工程完了時 SI工程完了時 PT工程完了時



6.改善策の実現方法

- (a)生涯バグ数の予測モデル
- (b)工程毎のバグ摘出数の算出
- (c)サービス開始後 6 か月バグ密度を予測
- (d)試験項目数を算出するモデル式**
- (e)(f)汎用的な分析シート

6.改善策の実現方法

(d)試験項目数を算出するモデル式

Input

Output



6.改善策の実現方法

(d)試験項目数を算出するモデル式

$$\text{必要な試験項目数} = \beta_0 + \beta_1 \cdot \text{抽出したいバグ数}$$

β_0 、 β_1 は定数。過去データから統計分析で算出

6.改善策の実現方法

(d)試験項目数を算出するモデル式

$$\text{必要な試験項目数} = \beta_0 + \beta_1 \cdot \text{抽出したいバグ数}$$

β_0 、 β_1 は定数。過去データが

「(b) 工程毎のバグ抽出数の算出」で算出したバグ数を入力する。

6.改善策の実現方法

- (a)生涯バグ数の予測モデル
- (b)工程毎のバグ摘出数の算出
- (c)サービス開始後 6 か月バグ密度を予測
- (d)試験項目数を算出するモデル式
- (e)(f)汎用的な分析シート

6.改善策の実現方法

(e) (f)汎用的な分析シート

No.	基本情報					UT工程						
	案件名 (任意)	処理方式 (任意)	新規/改造 (任意)	ID (必須)	処理名(ユースケース名) (必須)	バグ摘出目標件数					バグ実績件数	評価
						処置限界 下限値	許容限界 下限値	許容限界 目標値	許容限界 上限値	処置限界 上限値		
1	○○				○○を照会する	1	3	8	20	53	0	▼
2	○○				□□を照会する	1	1	4	9	23	0	▼
3	○○				××を登録する	1	1	3	8	20	0	▼
4	○○				☆☆を入力する	0	0	1	2	5	0	○
5	○○				○○データアクセス処理	0	0	1	3	7	0	○
6	○○				□□共通処理	1	2	4	10	24	0	▼
7	○○				△△設計書	0	0	1	3	7	0	○
8	○○				××メッセージ定義書	0	0	0	1	3	0	○
9	△△△△△				□□を照会する	0	1	1	3	8	0	▼
10	△△△△△				○○を照会する	0	0	0	1	3	0	○
11	△△△△△				□□を照会する	0	1	1	5	12	0	▼
12	△△△△△				××を登録する	0	0	0	1	8	0	○
13	△△△△△				☆☆を入力する	0	0	0	5	12	0	▼
14	△△△△△				○○データアクセス処理	0	1	1	4	10	1	○
15	△△△△△				□□共通処理	0	0	0	3	7	0	○
16	△△△△△				△△設計書	0	0	0	3	7	0	○
17	△△△△△				××メッセージ定義書	0	0	0	2	6	0	○
18	△△△△△				□□を照会する	0	1	1	6	14	1	○
19	△△△△△				○○を照会する	0	1	1	3	8	0	▼
20	△△△△△				□□を照会する	0	0	1	2	4	0	○
21	□□□□□				××を登録する	0	1	2	5	12	1	○
22	□□□□□				☆☆を入力する	0	1	3	7	17	11	▲
23	□□□□□				○○データアクセス処理	0	0	1	2	4	4	▲
24	□□□□□				□□共通処理	0	0	0	1	3	0	○
25	□□□□□				△△設計書	0	0	1	3	7	0	○
26	□□□□□				××メッセージ定義書	0	0	1	2	5	0	○
27	□□□□□				□□を照会する	0	0	1	2	4	0	○
28	□□□□□				○○を照会する	0	0	0	1	3	0	○
29	□□□□□				□□を照会する	0	1	2	4	10	0	▼
30	□□□□□				××を登録する	0	0	0	1	3	0	○
31	□□□□□				☆☆を入力する	0	0	0	1	3	0	○
32	□□□□□				○○データアクセス処理	0	0	0	1	3	0	○

イメージです

6.改善策の実現方法

(e) (f)汎用的な分析シート

No.	基本情報					UT工程						バグ実績件数	評価
	案件名 (任意)	処理方式 (任意)	新規/改造 (任意)	ID (必須)	処理名(ユースケース名) (必須)	バグ摘出目標件数							
						処置限界 下限値	許容限界 下限値	目標値	許容限界 上限値	処置限界 上限値			
1	○○				○○を照会する	1	3	8	20	53	0	▼	
2	○○				□□を照会する	1	1	4	9	23	0	▼	
3	○○				××を登録する	1	1	3	8	20	0	▼	
4	○○				☆☆を入力する	0	0	1	2	5	0	○	
5	○○				○○データアクセス処理	0	0	1	3	7	0	○	
6	○○				□□共通処理	1	2	4	10	24	0	▼	
						0	0	1	3	7	0	○	
						0	0	0	1	3	0	○	
						0	1	1	3	8	0	▼	
						0	0	0	1	3	0	○	
						0	1	2	5	12	0	▼	
						0	0	1	3	8	0	○	
						0	1	2	5	12	0	▼	
						0	1	2	4	10	1	○	
						0	0	1	3	7	0	○	
						0	0	1	3	7	0	○	
						0	0	1	2	6	6	○	
						0	1	2	6	14	1	○	
						0	1	1	3	8	0	▼	
						0	0	1	2	4	0	○	
						0	1	2	5	12	1	○	
						0	1	3	7	17	11	▲	
						0	0	1	2	4	4	▲	
						0	0	0	1	3	0	○	
						0	0	1	3	7	0	○	
						0	0	1	2	5	0	○	
						0	0	1	2	4	0	○	
						0	0	0	1	3	0	○	
						0	1	2	4	10	0	▼	
29	□□□□□				□□を照会する	0	1	2	4	10	0	○	
30	□□□□□				××を登録する	0	0	0	1	3	0	○	
31	□□□□□				☆☆を入力する	0	0	0	1	3	0	○	
32	□□□□□				○○データアクセス処理	0	0	0	1	3	0	○	

過去の実績データを投入すると自動で重回帰分析を行い、バグ摘出目標件数が表示される。

6.改善策の実現方法

(e) (f)汎用的な分析シート

No.	基本情報					UT工程					バグ実績件数	評価
	案件名 (任意)	処理方式 (任意)	新規/改造 (任意)	ID (必須)	処理名(ユースケース名) (必須)	バグ摘出目標件数						
						処置限界 下限値	許容限界 下限値	目標値	許容限界 上限値	処置限界 上限値		
1	○○				○○を照会する	1	3	8	20	53	0	▼
2	○○				□□を照会する	1	1	4	9	23	0	▼
3	○○				××を登録する	1	1	3	8	20	0	▼
4	○○				☆☆を入力する	0	0	1	2	5	0	○
5	○○				○○データアクセス処理	0	0	1	3	7	0	○
6	○○										0	▼
7	○○										0	○
8	○○										0	○
9	△△△										0	▼
10	△△△										0	○
11	△△△										0	▼
12	△△△										0	○
13	△△△										0	○
14	△△△										1	▼
15	△△△										0	○
16	△△△										0	○
17	△△△										0	○
18	△△△										1	○
19	△△△										0	▼
20	△△△										0	○
21	□□□										1	○
22	□□□										11	▲
23	□□□										4	▲
24	□□□										0	○
25	□□□										0	○
26	□□□□										0	○
27	□□□□				□□を照会する	0	0	1	2	4	0	○
28	□□□□				○○を照会する	0	0	0	1	3	0	○
29	□□□□				□□を照会する	0	1	2	4	10	0	▼
30	□□□□				××を登録する	0	0	0	1	3	0	○
31	□□□□				☆☆を入力する	0	0	0	1	3	0	○
32	□□□□				○○データアクセス処理	0	0	0	1	3	0	○

使用するデータ項目は、どの組織でも収集しているデータ項目に限定した。
 また、工程も自由に変更可能にした。

7.運用について

統計的品質分析シートへの運用

①対象とする開発案件

ソフトウェア開発&規模●●step以上

②使用時期

設計工程から

③他組織の適用条件

以下のデータを収集している組織であれば、基本的に分析シートは適用可能。

- ・設計工程 設計書の出来高頁数、エラー数
- ・試験工程 試験項目数、バグ数

8.改善による効果

8.改善による効果

従来

①品質水準値の算出方法が開発グループ毎に異なる

②サービス開始後6か月バグ密度を予測する統計的モデルがない

改善による
効果

従来の品質水準値と統計的品質分析シートを併用しているため、課題は残っている。

統計的品質分析シートにより、サービス開始後6か月バグ密度の予測が可能となった。

8.改善による効果

従来

③統計ツールの利用は、統計手法の知識が必要となるため、統計の専門家が分析を実施

④分析方法は、開発グループ毎に異なる

改善による
効果

統計的品質分析シートは、統計分析を自動で行われるため、誰でも分析可能となった。

現在、5開発グループに適用しており、統計的品質分析手法について標準化が行えた。その結果、統計的分析に関する開発G、SEPG双方の負荷が軽減された。

8. 今後に向けて

9. 今後に向けて

今回、策定した統計的品質分析シートについては、今後、以下の事項を実施していく。

①品質データのデータベースと連携して統計的品質分析シートに自動インポートする機能を追加する。

②従来のようなバグ抽出密度（バグ数/規模）ではなく、処理ごとにバグを件数で予測するモデルのため、適用に当たっては、啓蒙活動をしていく必要がある。



NTT DATA

Trusted Global Innovator