



SQiPシンポジウム2019

セッションB1-2

基盤構築プロジェクトにおける 定量的品質評価方法の提案

2019/09/12

株式会社 日立ソリューションズ
品質保証統括本部 SL品質保証部

○ 藤井 俊久 川合 大之
金倉 克行 土屋 裕
小笠原 毅 紀野 佑太郎
奥村 一郎 長岡 智子

E-mail: toshihisa.fujii.fe@hitachi-solutions.com

Contents

1. 背景と課題
2. 研究概要
3. 検証内容
4. 研究結果
5. 今後の課題・展望

1. 背景と課題

2. 研究概要

3. 検証内容

4. 研究結果

5. 今後の課題・展望

1. 1. 背景

- あるIT企業に勤めるAさんとBさんがそれぞれの近況についてやりとりしています。
- Aさんはソフトウェア開発プロジェクトのリーダー、
- Bさんは基盤構築プロジェクトのリーダーとしてそれぞれ活躍しています。



Aさん

この間S-inしたプロジェクト、結合テスト終盤にテストケース不足が発覚しちゃって…

21:12



Aさん

テストケースを追加して、そのおかげで後工程は順調で、無事S-inできたよ！

21:13

既読
21:21

ソフトウェア開発プロジェクトは羨ましいな。基盤構築プロジェクトは、テストケースの過不足を判断できる基準が無いからなあ…



Bさん



Aさん

定量的に評価してなくて、後工程で何かトラブルがあった時にどう対処しているの？

21:22

既読
21:23

結局は全てのテストケースを再チェックして、どこが悪かったのか、定性的な傾向を把握して、対処しているね。



Bさん



Aさん

それだと後手後手だよな。ソフトウェア開発のようなLOC法とかFP法でプロジェクト規模を出せれば良いな。

21:24

21:25

そうなんだよ。問題が先取りできれば嬉しいよ。



Bさん

1. 2. 基盤構築プロジェクトの課題

```

public class SimpleGame {
    ...
}

public class SimpleGame {
    ...
}

public class SimpleGame {
    public static void main(String[] args) {
        System.out.println("[1] XXの首都はどこですか?");
        System.out.println("[2] 1: AAAA");
        System.out.println("[3] 2: BBBB");
        System.out.println("[4] 3: CCCC");
        System.out.print("[5] 答えを番号で入力し、改行を押してく
ださい。=>");
        InputStreamReader inputStreamReader = new
        InputStreamReader(System.in);//[6]
        BufferedReader bufferedReader = new
        BufferedReader(inputStreamReader);//[7]
        int answer = 0;//[8]
        ...
    }
}
    
```

プロジェクト種別	ソフトウェア開発	基盤構築
プロジェクト規模	・LOC ・FP 等	?
品質指標	設計書ページ数	?
	レビュー指摘数	?
	テストケース数	?
	欠陥数	?
定量評価可否	○(可能)	×(不可能)



**【課題】プロジェクトの規模を示すメトリクスが無い
ため品質指標を定義できず、定量評価できない**

【仮説】ソフトウェア開発におけるLOCに該当するメトリクスを導出

-
1. 背景と課題
 - 2. 研究概要**
 3. 検証内容
 4. 研究結果
 5. 今後の課題・展望

基盤構築プロジェクトの規模に基づく
品質指標の有意性検証

- 設計書ページ数
- テストケース数
- 欠陥数

etc...

③ 有意性検証

① メトリクス収集

② 統計解析

- 収集したメトリクスの相関分析
- 基盤構築プロジェクト規模 算出式の導出

-
1. 背景と課題
 2. 研究概要
 - 3. 検証内容**
 4. 研究結果
 5. 今後の課題・展望

3. 1. メトリクス収集-(1)

ソフトウェア開発 プロジェクト

○ 設計書ページ数

○ 設計書レビュー指摘数

○ 設計書レビュー時間

○ テストケース数

○ 欠陥数

✗ ソースコード行数

基盤構築 プロジェクト

設計書ページ数

設計書レビュー指摘数

設計書レビュー時間

テストケース数

欠陥数



**基盤構築プロジェクト
独自のメトリクス
収集が必要**

○ : 適用可能 ✗ : 適用不可能



3. 1. メトリクス収集-(2)

▼メトリクスの収集方法

- 完了した社内の基盤構築プロジェクトが対象
- プロジェクトで作成した成果物から情報を抽出

【基本設計工程】

- ・設計書ページ数
- ・レビュー指摘数
- ・レビュー時間

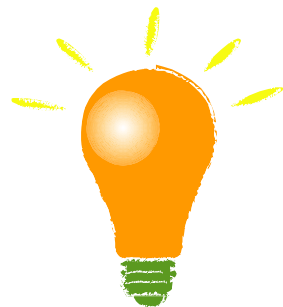
【詳細設計工程】

- ・設計書ページ数
- ・レビュー指摘数
- ・レビュー時間

【構築・テスト工程】

- ・テストケース数
- ・テストケースレビュー指摘数
- ・テストケースレビュー時間
- ・欠陥数

基盤構築プロジェクト独自のメトリクスとして……



- ✓ 導入対象の製品数
- ✓ 導入対象のサーバ台数
- ✓ 製品に設定するパラメータ数

■NEXT

メトリクス間でデータ
に相関があるか考察



▼相関分析実施時の考察

下流工程のメトリクス(テストケース数、欠陥数etc...)から
基盤構築プロジェクトの規模を推定するよりも、
上流工程のメトリクスから推定したい

上流工程のメトリクスから推定できれば以下の情報を
事前に把握できる

- ・下流工程で作成すべき成果物の量
- ・品質を担保するために必要なテストケース数や欠陥数

設計工程の上流工程である『基本設計書ページ数』
から基盤構築プロジェクトの規模を推定できることが
望ましい

3. 2. 統計解析-(2)

▼ 相関分析実施結果

		x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18
x1	基本設計書ページ数	1.0000																	
x2	基本設計書ビュー指摘数	0.7406	1.0000																
x3	基本設計書ビュー時間	0.8592	0.7286	1.0000															
x4	詳細設計書ページ数	0.6606	0.0129	0.0087	1.0000														
x5	詳細設計書ビュー指摘数	0.3276	0.0355	0.0078	0.7280	1.0000													
x6	詳細設計書ビュー時間	0.3061	-0.0757	-0.0073	0.8485	0.7091	1.0000												
x7	単体テストケース数	0.0952	0.0507	0.0179	-0.1703	0.0333	0.0536	1.0000											
x8	単体テストケースビュー指摘数	0.1152	0.0762	0.0741	0.0295	0.0609	0.0022	0.6269	1.0000										
x9	単体テストケースビュー時間	0.0873	0.0095	0.0036	-0.0453	-0.0012	0.0182	0.9017	0.6424	1.0000									
x10	単体テスト抽出欠陥数	0.0695	0.0376	0.0032	0.7559	0.0254	0.0130	0.2091	0.2391	-0.0649	1.0000								
x11	結合テストケース数	0.5344	0.0587	0.0519	0.2961	0.0408	0.0519	-0.1533	0.0816	-0.1377	0.3789	1.0000							
x12	結合テストケースビュー指摘数	0.4924	0.0341	0.0142	0.1163	0.0117	-0.0142	0.1744	-0.0117	-0.0569	0.0497	0.5284	1.0000						
x13	結合テストケースビュー時間	0.3784	0.0876	0.0258	0.1312	0.0421	0.1289	0.0732	0.1686	0.0866	-0.0076	0.9251	0.7279	1.0000					
x14	結合テスト抽出欠陥数	0.9445	-0.0122	0.0113	0.0966	0.0009	0.0565	-0.2188	-0.0036	-0.1424	0.9301	0.5246	0.1483	-0.0456	1.0000				
x15	導入対象の製品数	0.9346	0.0062	-0.0037	0.8823	0.0048	-0.0149	-0.2835	0.0096	0.0596	0.9074	0.5471	0.0350	0.0164	0.9613	1.0000			
x16	ホストサーバ台数	0.4876	-0.0088	-0.0015	-0.0984	0.0011	0.0045	-0.0147	-0.0044	-0.0135	0.5876	0.2706	-0.0422	-0.0131	0.6101	0.5664	1.0000		
x17	仮想マシン台数	0.2416	-0.0475	0.0325	0.2358	-0.0350	0.1625	-0.0476	0.0701	0.1225	0.5359	-0.1254	0.0665	0.0566	0.2866	0.3701	0.5462	1.0000	
x18	製品に設定するパラメータ数	0.6414	0.0682	0.0033	0.6177	0.0412	-0.0100	0.0623	0.0824	0.0400	0.7609	0.2999	0.0298	0.0044	0.7852	0.7075	0.9083	0.4798	1.0000

「x1:基本設計書ページ数」

⇔「x14:結合テスト抽出欠陥数」、「x15:導入対象の製品数」に相関あり

「x14:結合テスト抽出欠陥数」

⇔「x15:導入対象の製品数」、「x16:ホストサーバ台数」に相関あり

3. 2. 統計解析-(3)

「基本設計書ページ数」⇔「結合テスト摘出欠陥数」、「導入対象の製品数」に相関あり

「結合テスト摘出欠陥数」⇔「導入対象の製品数」、「ホストサーバ台数」に相関あり

基本設計書ページ数から
基盤構築プロジェクト規模を推定したい

《重回帰式》

$$P_{BS} = 36.5 * \sum_{i=1}^n V_{Pi} + 0.099 * \sum_{i=1}^n N_{SVi}$$

excel®の分析ツール
より推定

P_{BS} : 基本設計書ページ数 V_{Pi} : 導入対象の製品数
 N_{SVi} : ホストサーバ台数

【補正】
ホスト上に構築する仮想マシン台数も加味

《重回帰式'》

$$P_{BS} = 36.5 * \sum_{i=1}^n V_{Pi} + 0.099 * \sum_{i=1}^n N_{SVi}$$

P_{BS} : 基本設計書ページ数 V_{Pi} : 導入対象の製品数
 N_{SVi} : ホスト・仮想マシンの各サーバ台数

3. 2. 統計解析－(4)

$$\langle\langle \text{重回帰式} \rangle\rangle \quad P_{BS} = 36.5 * \sum_{i=1}^n V_{Pi} + 0.099 * \sum_{i=1}^n N_{SVi}$$

P_{BS} : 基本設計書ページ数 V_{Pi} : 導入対象の製品数 N_{SVi} : ホスト・仮想マシンの各サーバ台数

知りたい情報は基本設計書ページ数でなく基盤構築プロジェクトの規模

- ・ 弊社では、ソフトウェア開発における基本設計書の作成指標として、10ページ/1KLOCの指標あり

- ・ 「重回帰式」を10ページ/1KLOCで除算したものを基盤構築プロジェクトの規模を示す「構築スケール(Cs:Construction Scale)*1」と定義

- ・ 1KLOCの成果物の物量(開発量)を1Csで表現できれば、1KLOC⇔1Csの読替えが容易で、社内でも馴染みやすい

$$\langle\langle \text{重回帰式} \rangle\rangle \quad Cs = 3.65 * \sum_{i=1}^n V_{Pi} + 0.0099 * \sum_{i=1}^n N_{SVi}$$

Cs : 構築スケール V_{Pi} : 導入対象の製品数
 N_{SVi} : ホスト・仮想マシンの各サーバ台数

3. 3. 有意性検証-(1)

$$\langle\langle \text{重回帰式} \rangle\rangle \quad C_S = 3.65 * \sum_{i=1}^n V_{Pi} + 0.0099 * \sum_{i=1}^n N_{SVi}$$

C_S : 構築スケール V_{Pi} : 導入対象の製品数 N_{SVi} : ホスト・仮想マシンの各サーバ台数

「重回帰式」を用いて、各工程における品質データを定量評価した結果を複数のプロジェクトで比較。特異点がないかを検証。

メールや監視、認証といった製品を採用しているプロジェクトで、構築スケールから想定される基本設計書ページ数やテストケース数、欠陥数にばらつきが発生する点に着目。

「導入対象の製品別カテゴリ係数」を新たに定義し、 V_{Pi} および N_{SVi} に本係数をそれぞれ乗算。

3. 3. 有意性検証-(2)

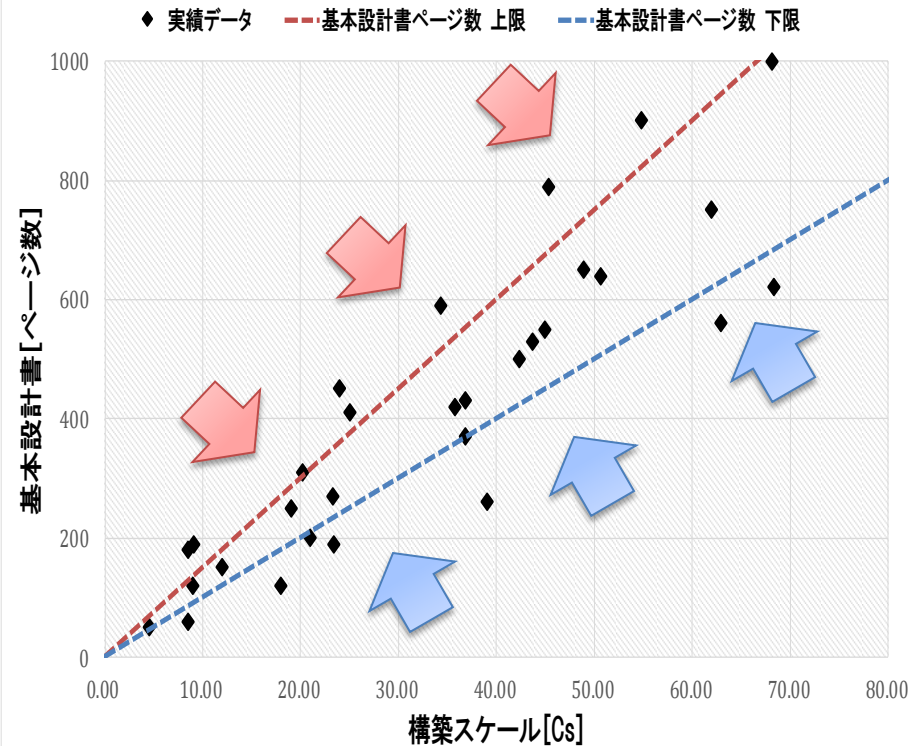
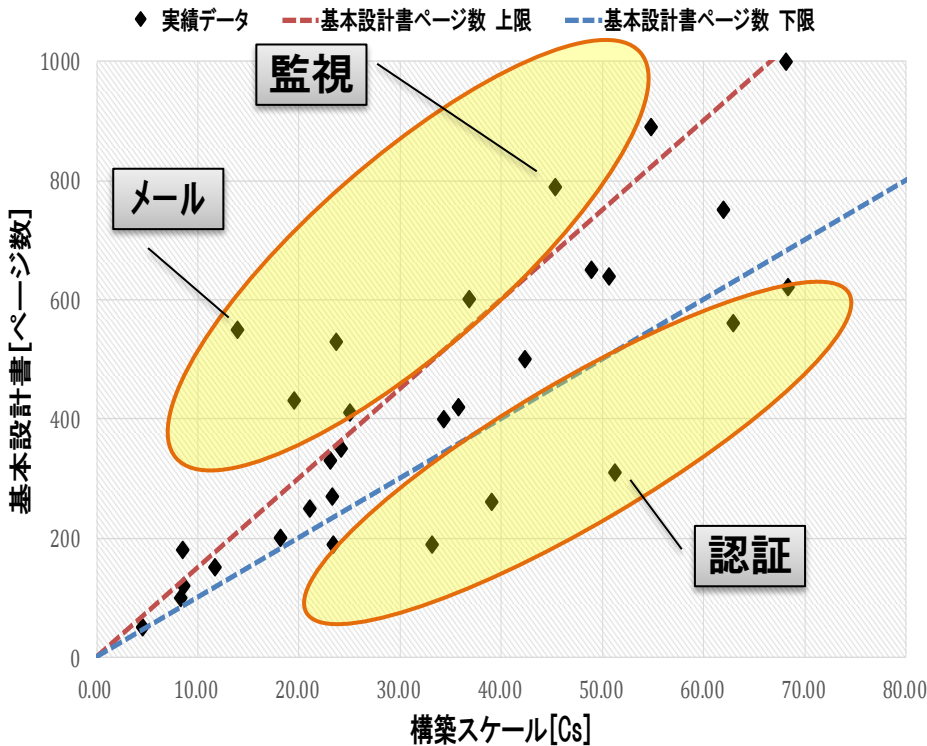
導入対象の製品別カテゴリ係数

適用前

適用後

【適用前】構築スケール×基本設計書ページ数 散布図

【適用後】構築スケール×基本設計書ページ数 散布図



《重回帰式》
《構築スケール算出式》

$$Cs = 0.6 \max(\alpha) \sum_{i=1}^n V_{Pi} + 0.01 \sum_{i=1}^n \alpha N_{SVi}$$

Cs: 構築スケール α : 導入対象の製品別カテゴリ係数
 V_{Pi} : 導入対象の製品数 N_{SVi} : ホスト・仮想マシンの各サーバ台数

-
1. 背景と課題
 2. 研究概要
 3. 検証内容
 - 4. 研究結果**
 5. 今後の課題・展望

▼構築スケール(Cs)を基にした当社内の品質指標

*メトリクス収集したプロジェクトの平均値より算出

#	工程名称	品質指標1	品質指標2
1	基本設計	作成ページ数[ページ/Cs] 10.0~15.0	レビュー指摘数[指摘数/100ページ] 11.0~20.0
2	詳細設計	作成ページ数[ページ/Cs] 30.0~50.0	レビュー指摘数[指摘数/100ページ] 11.0~20.0
3	構築/単体テスト	テストケース数[ケース数/Cs] 50.0~135.0	摘出欠陥数[欠陥数/Cs] 1.00~3.00
4	結合テスト	テストケース数[ケース数/Cs] 15.0~30.0	摘出欠陥数[欠陥数/Cs] 1.00~3.00

4. 1. 結果－(2)

▼構築スケール(Cs)を基にした定量品質評価(抜粋)

*社内24プロジェクトで試行

#	プロジェクト名称	構築スケール [Cs]	基本設計		詳細設計		構築/単体テスト		結合テスト	
			作成ページ数 [ページ/Cs]	レビュー指摘数 [指摘数/100 ページ]	作成ページ数 [ページ/Cs]	レビュー指摘数 [指摘数/100 ページ]	テストケース数 [ケース数/Cs]	抽出欠陥数 [欠陥数/Cs]	テストケース数 [ケース数/Cs]	抽出欠陥数 [欠陥数/Cs]
指標値			10.0~15.0	11.0~20.0	30.0~50.0	11.0~20.0	50.0~135.0	1.00~3.00	15.0~30.0	1.00~3.00
1	Aプロジェクト	20.26	14.9(O)	27.7(△)	24.0(▽)	12.7(O)	64.9(O)	0.89(▽)	37.9(△)	2.17(O)
2	Bプロジェクト	6.70	10.9(O)	26.0(△)	18.7(▽)	12.0(O)	83.4(O)	1.34(O)	29.6(O)	1.19(O)
3	Cプロジェクト	20.07	23.2(△)	11.2(O)	61.1(△)	16.6(O)	159.7(△)	0.95(▽)	62.5(▲)	1.69(O)
4	Dプロジェクト	5.78	33.9(▲)	46.3(▲)	94.5(▲)	11.9(O)	436.3(▲)	2.25(O)	124.2(▲)	1.90(O)
5	Eプロジェクト	23.49	11.3(O)	24.5(△)	66.4(△)	0.96(▽)	124.1(O)	0.21(▽)	115.4(▲)	0.38(▽)

《凡例》

▲: 指標値上限 2倍超過

△: 指標値上限 超過

○: 指標値内

▽: 指標値下限 未満

▼: 指標値下限 1/2未満



✓ 約50%は指標値範囲内

✓ 上限超過・下限未満を含めると約75%はCsによる定量品質評価に適合

▼構築スケール(Cs)を基にした定量品質評価での成果(抜粋)



課題

- ・ 作成段階で結合テストのテストケース不足を把握

対策

- ・ テストケースを追加し、結合テストで抽出すべき欠陥を発見

成果

- ・ 後工程で結合テストで抽出すべき欠陥無し！無事S-in！！



課題

- ・ 基本設計の進捗約50%段階で、作成ページ数が少ない傾向を把握

対策

- ・ 設計すべき仕様として、不足・漏れが無い点検

成果

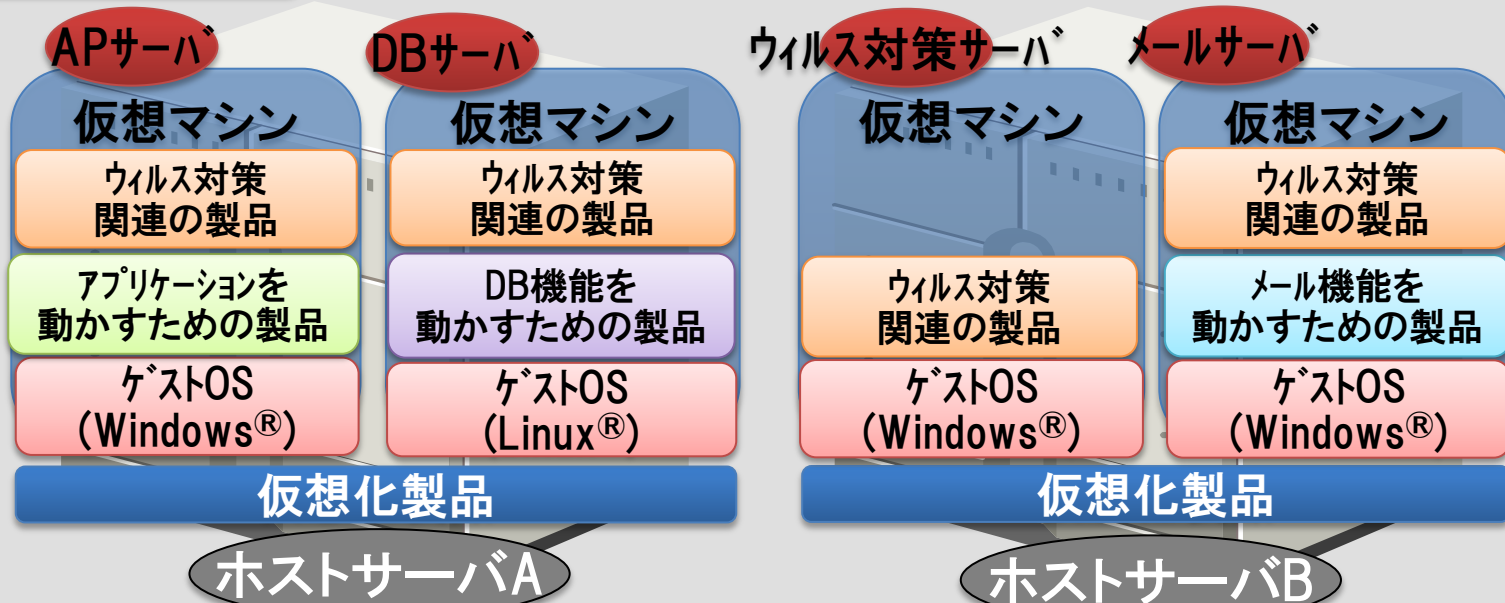
- ・ お客様も考慮が不足していた設計仕様を追加！
- ・ 受入テストも大きな問題無く完了！！

4. 2. 【演習】基盤構築プロジェクト規模の算出－(1)

Tips

▼構築スケールを実際に算出してみましょう

Xシステム



《構築スケール算出式》
$$Cs = 0.6 \max(\alpha) \sum_{i=1}^n V_{Pi} + 0.01 \sum_{i=1}^n \alpha N_{SVi}$$

Cs : 構築スケール α : 導入対象の製品別カテゴリ係数

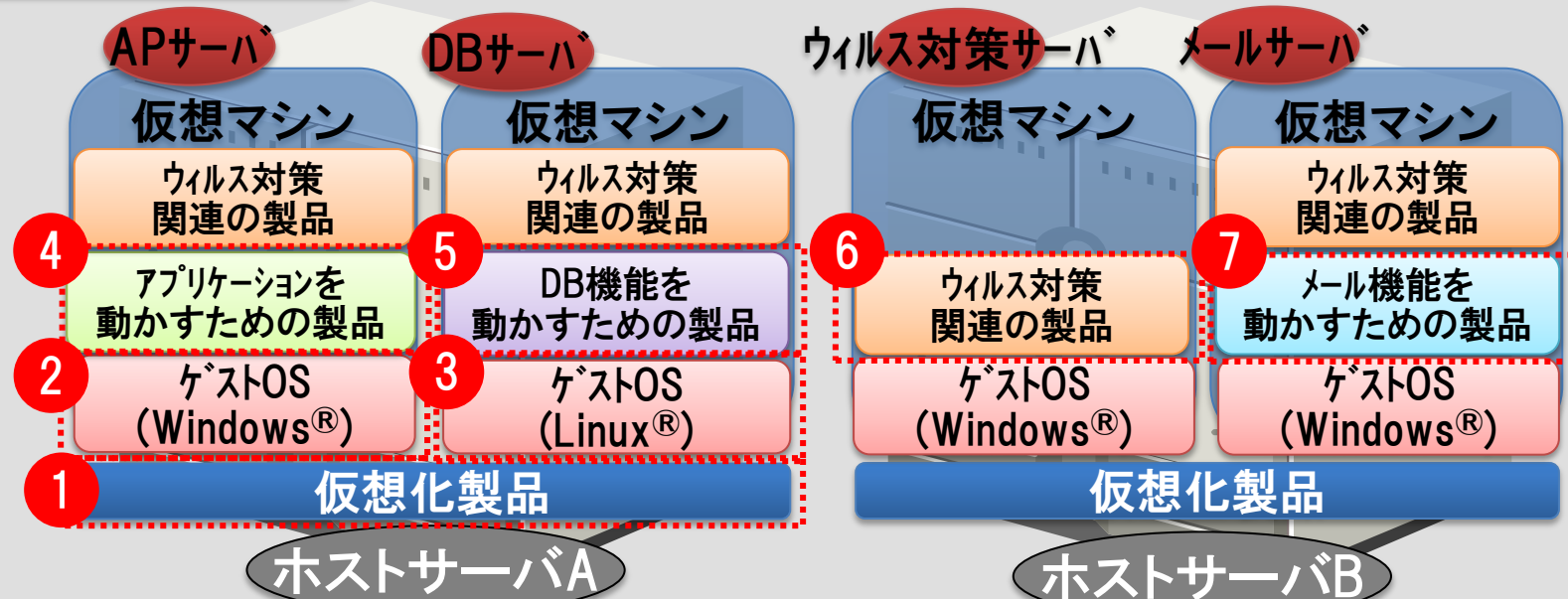
V_{Pi} : 導入対象の製品数 N_{SVi} : ホスト・仮想マシンの各サーバ台数

4. 2. 【演習】基盤構築プロジェクト規模の算出ー(2)

Tips

▼構築スケールを実際に算出してみましょう

Xシステム



$$Cs = 0.6 \max(\alpha) \sum_{i=1}^n V_{Pi} + 0.01 \sum_{i=1}^n \alpha N_{SVi}$$

第1項に着目

α : 導入対象の製品別カテゴリ係数
 V_{Pi} : 導入対象の製品数

- ✓ V_{Pi} は重複排除
- ✓ 導入対象の製品数 V_{Pi} は①～⑦の7つ
- ✓ ①～⑦の製品別カテゴリ係数は次ページ



4. 2. 【演習】基盤構築プロジェクト規模の算出－(3)

Tips

▼構築スケールを実際に算出してみましょう

製品別カテゴリ係数 一覧

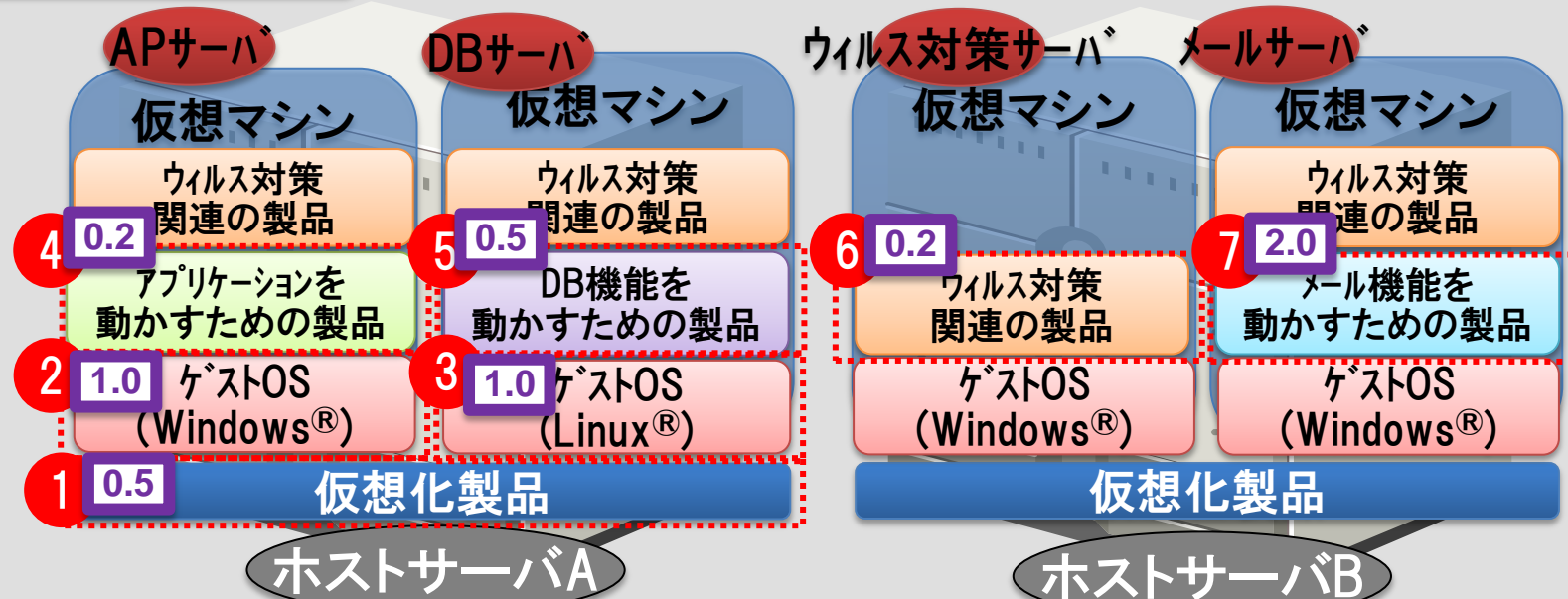
#	品質指標値 製品カテゴリ	係数	#	品質指標値 製品カテゴリ	係数
1	OS(Linux [®] 系)	1.0	19	リモートアクセス	0.2
2	OS(Windows [®] 系)	1.0	20	バックアップ	0.2
3	DHCP	0.2	21	ファイル転送	0.2
4	LDAP	0.2	22	ファイル共有	0.2
5	クラスタ	0.2	23	ポータル	0.2
6	APサーバ	0.2	24	メール	2.0
7	Webサーバ	0.2	25	Proxy	0.2
8	DB	0.5	26	ネットワーク機器管理	0.2
9	KVS	0.2	27	インベントリ管理	0.2
10	Active Directory [®]	3.0	28	サーバ	0.2
11	認証管理	0.2	29	ストレージ	0.2
12	RADIUS	0.2	30	仮想デスクトップ	0.2
13	セキュリティ関係	0.2	31	NTP	0.2
14	仮想化	0.5	32	DNS	0.2
15	ジョブ管理	0.2	33	FW	0.2
16	監視	3.0	34	ライブラリ装置	0.2
17	監査証跡	0.2	35	負荷分散	0.2
18	メッセージング	0.2	36	その他	0.2

4. 2. 【演習】基盤構築プロジェクト規模の算出 - (4)

Tips

▼構築スケールを実際に算出してみましょう

Xシステム



$$Cs = 0.6 \max(\alpha) \sum_{i=1}^n V_{Pi} + 0.01 \sum_{i=1}^n \alpha N_{SVi}$$

第1項に着目

α : 導入対象の製品別カテゴリ係数
 V_{Pi} : 導入対象の製品数

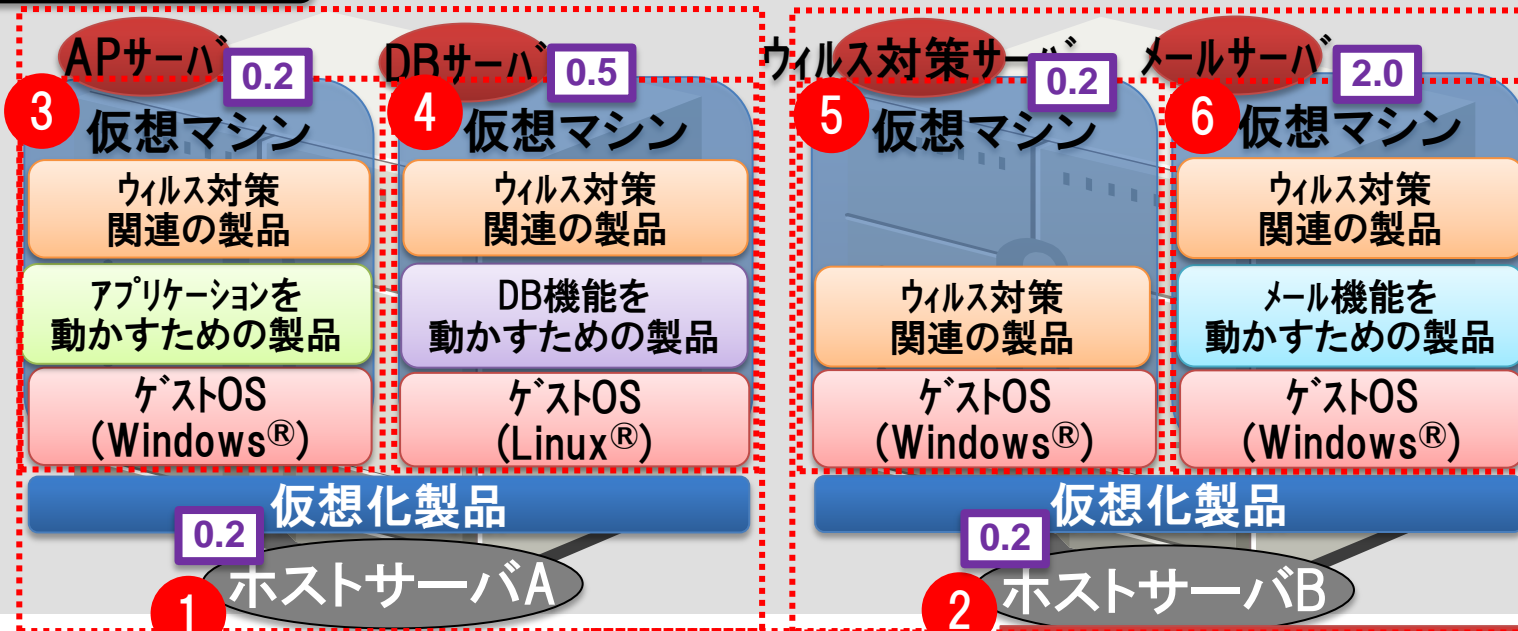
- ✓ V_{Pi} は重複排除
- ✓ 導入対象の製品数 V_{Pi} は①～⑦の7つ
- ✓ ①～⑦の製品別カテゴリ係数の最大値は⑦の2.0
- ✓ 第1項: $0.6 * 2.0 * 7 = 8.4$

4. 2. 【演習】基盤構築プロジェクト規模の算出－(5)

Tips

▼構築スケールを実際に算出してみましょう

Xシステム



$$Cs = 0.6 \max(\alpha) \sum_{i=1}^n V_{Pi} + 0.01 \sum_{i=1}^n \alpha N_{SVi}$$

α : 導入対象の製品別カテゴリ係数
 N_{SVi} : ホスト・仮想マシンの各サーバ台数

第2項に着目

- ✓ ホスト・仮想マシンの各台数 N_{SVi} は ①～⑥の6台
- ✓ ①～⑥のホスト・仮想マシンの主たる役割に応じた製品別カテゴリ係数*1を乗じた後、合算
- ✓ 第2項: $0.01 * (0.2 * 1 + 0.2 * 1 + \dots) = 0.033$

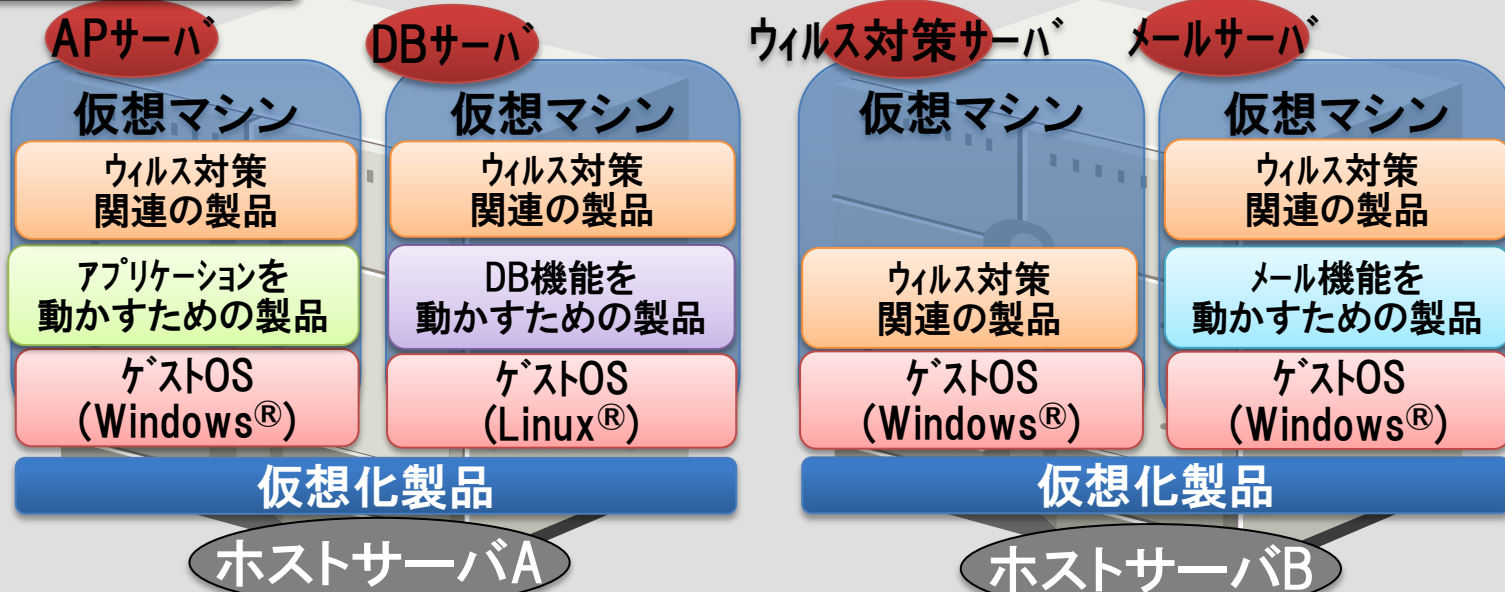
*1 製品別カテゴリ係数 一覧は付録を参照

4. 2. 【演習】基盤構築プロジェクト規模の算出－(6)

Tips

▼構築スケールを実際に算出してみましょう

Xシステム



$$Cs = 0.6 \max(\alpha) \sum_{i=1}^n V_{Pi} + 0.01 \sum_{i=1}^n \alpha N_{SVi} = 8.4 + 0.033 \times 255 = 8.433$$



■作成すべき成果物量は...

『基本設計書ページ数85～127』、『単体テストケース数422～1139』 etc...

-
1. 背景と課題
 2. 研究概要
 3. 検証内容
 4. 研究結果
 - 5. 今後の課題・展望**

2018年度下期完了時点で下記課題あり
継続して構築スケール算出式や品質指標を見直していく

- 「新規システム構築」と「システム刷新/一部機能改修」
の差異点考慮
- クラウドサービスを利用したプロジェクトの考慮
- 新規取り扱い製品の考慮
- 基盤構築プロジェクトに係る作業コスト見積への応用
- ネットワーク構築プロジェクトへの応用

END

基盤構築プロジェクトにおける 定量的品質評価方法の提案

2019/09/12

株式会社 日立ソリューションズ
品質保証統括本部 SL品質保証部

○ 藤井 俊久
金倉 克行
小笠原 毅
奥村 一郎

川合 大之
土屋 裕
紀野 佑太郎
長岡 智子

E-mail: toshihisa.fujii.fe@hitachi-solutions.com

▼製品別カテゴリ係数 一覧

#	品質指標値 製品カテゴリ	係数	#	品質指標値 製品カテゴリ	係数
1	OS(Linux [®] 系)	1.0	19	リモートアクセス	0.2
2	OS(Windows [®] 系)	1.0	20	バックアップ	0.2
3	DHCP	0.2	21	ファイル転送	0.2
4	LDAP	0.2	22	ファイル共有	0.2
5	クラスタ	0.2	23	ポータル	0.2
6	APサーバ	0.2	24	メール	2.0
7	Webサーバ	0.2	25	Proxy	0.2
8	DB	0.5	26	ネットワーク機器管理	0.2
9	KVS	0.2	27	インベントリ管理	0.2
10	Active Directory [®]	3.0	28	サーバ	0.2
11	認証管理	0.2	29	ストレージ	0.2
12	RADIUS	0.2	30	仮想デスクトップ	0.2
13	セキュリティ関係	0.2	31	NTP	0.2
14	仮想化	0.5	32	DNS	0.2
15	ジョブ管理	0.2	33	FW	0.2
16	監視	3.0	34	ライブラリ装置	0.2
17	監査証跡	0.2	35	負荷分散	0.2
18	メッセージング	0.2	36	その他	0.2