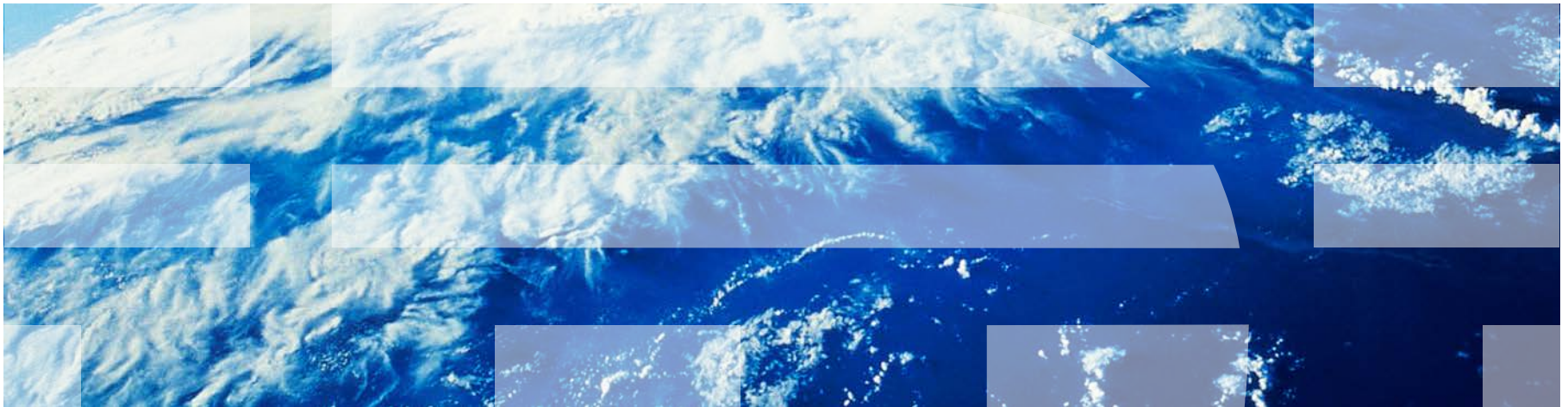
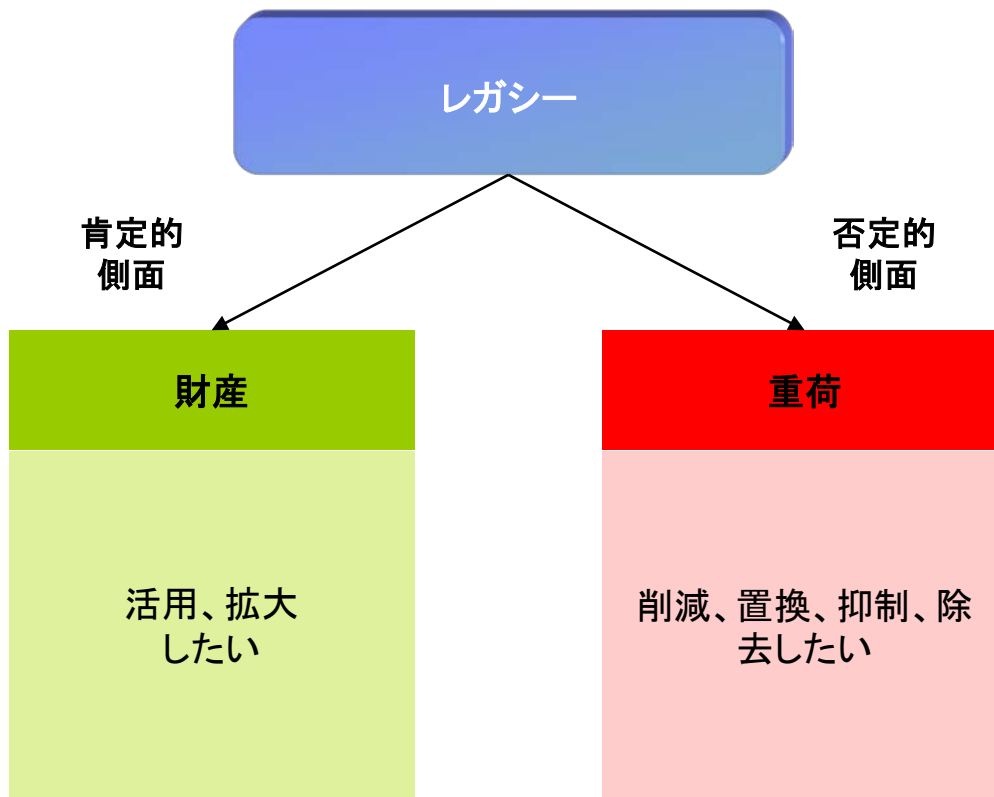


JaSST'14 Hokkaido プログラム類似性の可視化と事例の紹介

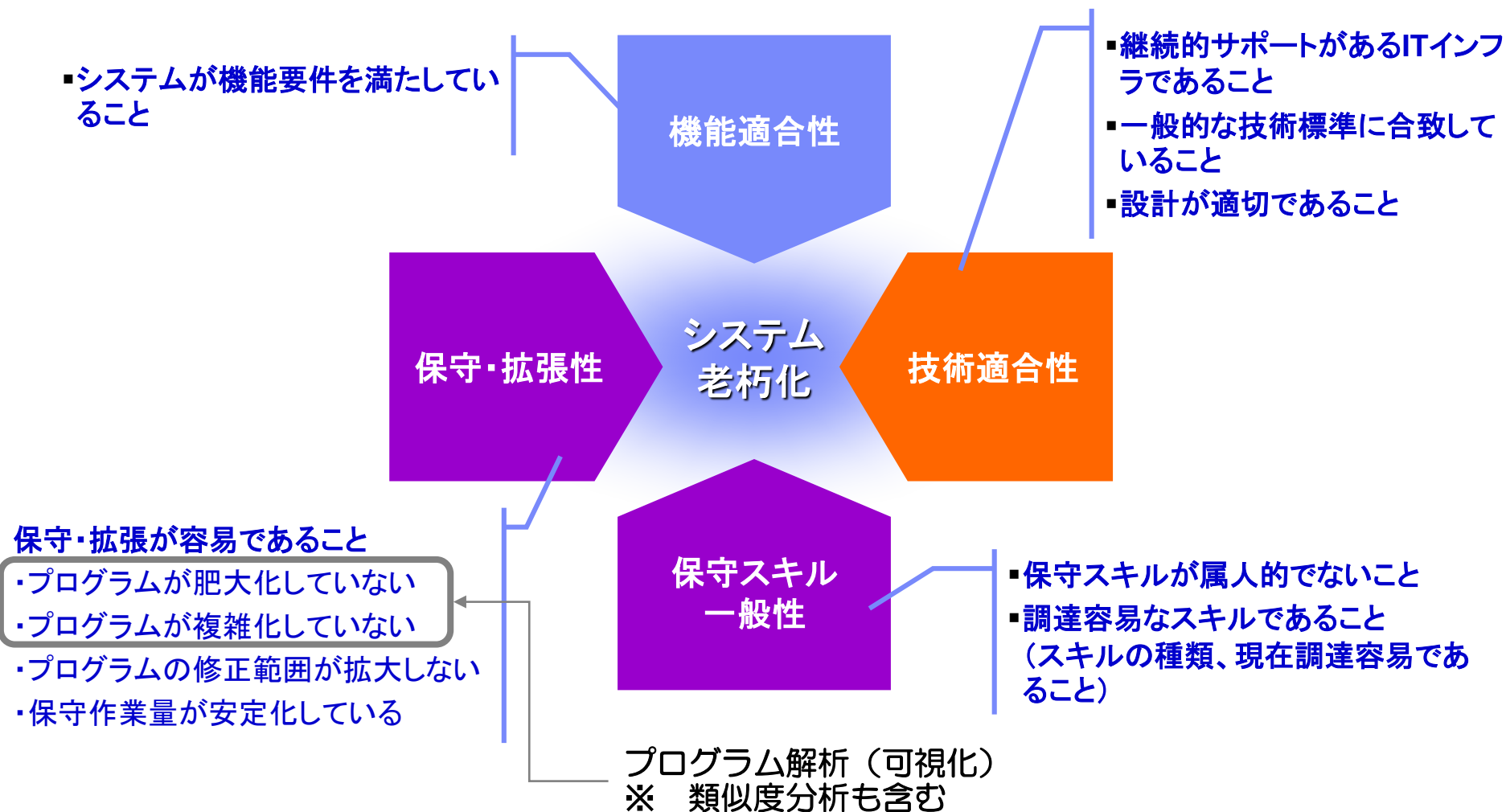
日本アイ・ビー・エム株式会社
BAM (Business Application Modernization)
繁在家 学 - Manabu Hanzaike
mhanz@jp.ibm.com





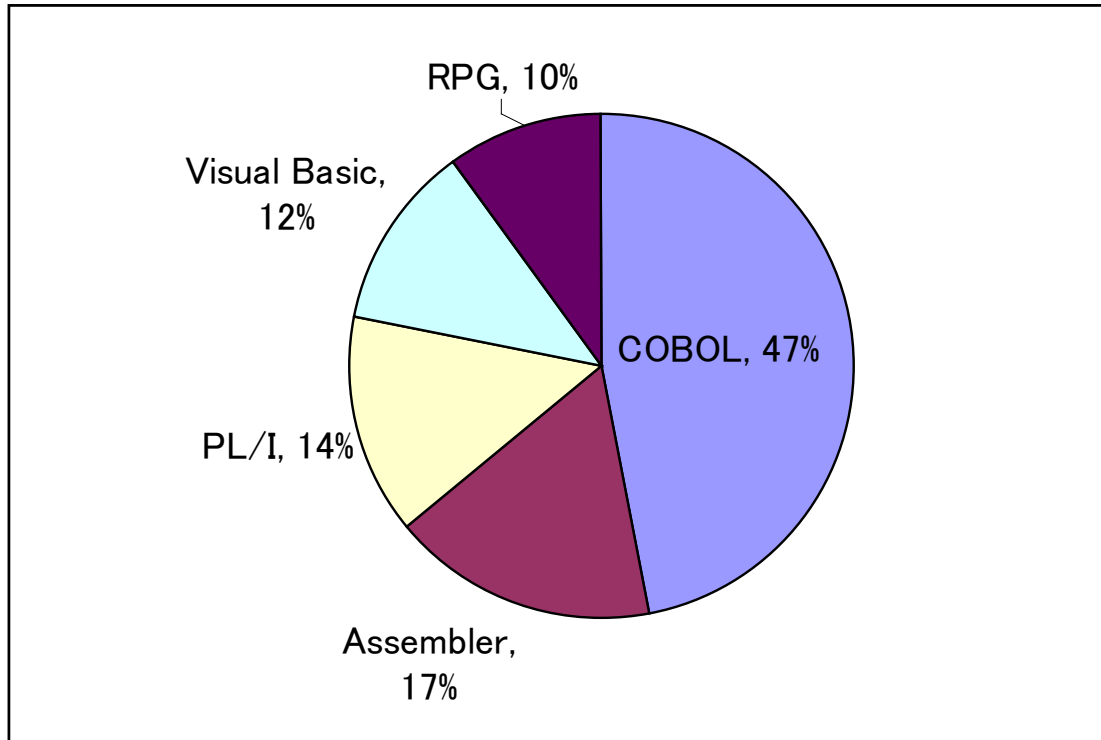
企業のレガシーには肯定的、否定的側面があります ※

※ IBM 2010 Global Technology Outlook , Future Of Legacyより引用



システム資源の類似性が
高いシステムは
品質が低いと言えるか？

- ※ システム資源 ≡ プログラム・バッチなど
 - ※ 品質といっても色々ありますが・・・
- 例： ISO9126



レガシーな言語って何 [1]

※ Khadka, Ravi, et al. "How do professionals perceive legacy systems and software modernization?." ICSE. 2014.

- 類似性分析は、コメントによる影響、変数、構造体メンバーの名前の違いによる影響等を除外して、プログラムの構造に対する類似の度合いをプログラム単位で分析するものです。分析は比較対象のプログラムの全組合せに対して実施します。
- 類似性分析の結果は現行システムの状態を表すひとつの指標であり（※1）、今後のモダナイゼーションにおける部分的な再構築、新規開発時のプログラム共通化・統合化検討、及び全体ロードマップ策定に関するインプットのひとつになります。

※1： 類似した機能をプログラムのコピーと改修によって実装することで、プログラムの可読性を確保し、保守の生産性を向上させている場合もあり、類似度と開発保守生産性が必ずしも関連付けられるものではありません。

- ここでは、単純なプログラムの例を用いて、類似性分析の方法を説明します。
- 下図の例のように、10命令で構成されているPGM Aと5命令で構成されるPGM Bを比較し、2命令の類似したコードが検出された場合、PGM Aを比較元としたPGM Bとの類似度は 20% となります。一方、PGM Bを比較元とした PGM Aとの類似度は40%となります。
- PGM間の類似箇所抽出は、共通した文字列（または差分）を抽出する際に一般的に利用されるアルゴリズムを利用して行っています（※1）。
- 類似性分析の結果は、類似度マトリックスと呼ばれる行列（マトリックス）を用いて表現します。類似度マトリックスの横軸と縦軸には分析対象のプログラムが同じ順番（PGMID順）で並んでおり、各行列の値は類似度（単位%）になります。対角線（左上から右下）は自分自身との類似度を比較した結果になり、常に100%になります。

類似度の比較の例

PGM A (10命令)	PGM B (5命令)
命令1 : DCL 変数 型 (数値)	命令1 : DCL 変数 型(数値)
命令2 : DCL 変数 型 (数値)	命令2 : 変数 = 変数 + 変数
命令3 : 変数 = 変数	命令3 : 変数 = 'リテラル値'
命令4 : 変数 = 'リテラル値'	命令4 : 変数 = 変数 - 変数
命令5 : IF (変数 = 変数)	命令5 : 変数 = 変数 * 変数
命令6 : THEN	
命令7 : DO	
命令8 : 変数 = 'リテラル値'	
命令9 : 変数 = 変数	
命令10 : END	

PGM AとPGM Bを比較した結果、2命令が類似しているとして検出された

類似度の結果

類似した命令数	2命令
PGM Aを比較元とした場合の類似度	20% (= 2 ÷ 10命令)
PGM Bを比較元とした場合の類似度	40% (= 2 ÷ 5命令)

類似度マトリックスの結果 [単位：%]

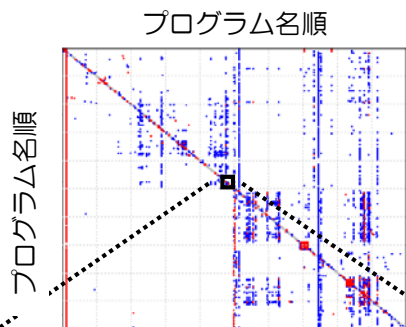
比較先プログラム \ 比較元プログラム	PGM A	PGM B
PGM A	100	40
PGM B	20	100

※1：共通した文字列（または差分）抽出として一般的なアルゴリズムである最長共通部分列（LCS：Longest Common Subsequence）を利用して類似箇所を抽出しています。

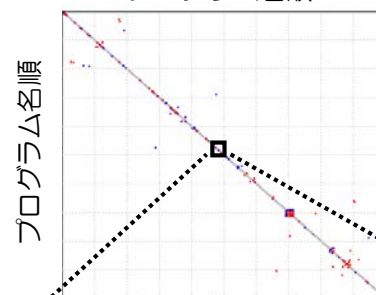
- 以下は類似性分析の結果を類似度マトリックスで表したものです。
- 類似性分析の結果を類似度マトリックスを使って説明しますが、より直感的に類似した状況を理解できるよう、マトリックスの値に対して色をつけています。色のつけ方には以下の通り、2種類があります。

- 類似度マトリックス（統合化指標）： 比較先と比較元の双方の類似度が、80%以上～90%未満の箇所を青色、90%以上の箇所を赤色で表示します。結果は対角線に対して常に左右対称に色がつきます。プログラムの統合を検討する際の指標となります。
- 類似度マトリックス（共通化指標）： 比較先、比較元のどちらかの類似度が、80%以上～90%未満の類似度の箇所を青色、90%以上の類似度の箇所を赤色で表示します。類似した命令が比較元または比較先にどの程度含まれているかを視覚化しています。プログラム間で共通化を検討する際の指標となります。

類似度マトリックス
(共通化指標)



プログラム名順



類似度マトリックス
(統合化指標)

90%以上の類似度は赤色で表示します。

80%以上～90%未満の類似度は青色で表示します。

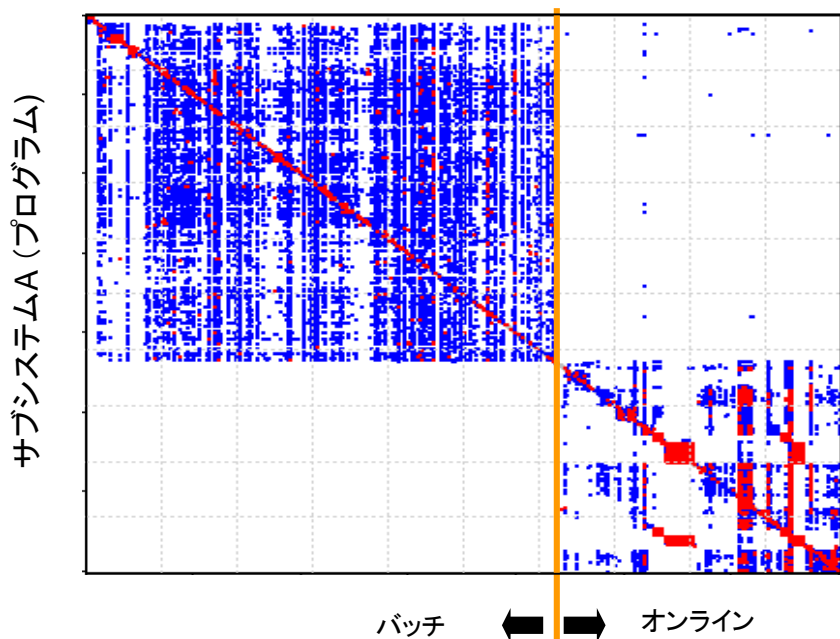
比較元プログラム		比較先プログラム									
	PGMA	PGMB	PGMC	PGMD	PGME	PGMF	PGMG	PGMH	PGMI	PGMJ	
PGMA	100	94	87	80	86	45	42	41	31	74	
PGMB	94	100	87	80	86	45	42	41	31	74	
PGMC	83	83	100	80	88	46	41	40	31	75	
PGMD	33	33	34	100	59	33	17	16	18	34	
PGME	46	48	51	80	100	42	25	24	26	48	
PGMF	46	46	56	82	78	100	37	35	37	46	
PGMG	77	77	78	77	83	66	100	52	43	77	
PGMH	72	72	74	71	77	59	50	100	50	68	
PGMI	58	58	62	82	88	67	44	53	100	59	
PGMJ	71	71	75	80	83	43	40	37	30	100	

比較元プログラム		比較先プログラム									
	PGMA	PGMB	PGMC	PGMD	PGME	PGMF	PGMG	PGMH	PGMI	PGMJ	
PGMA	100	94	87	80	86	45	42	41	31	74	
PGMB	94	100	87	80	86	45	42	41	31	74	
PGMC	83	83	100	80	88	46	41	40	31	75	
PGMD	33	33	34	100	59	33	17	16	18	34	
PGME	48	48	51	80	100	42	25	24	26	48	
PGMF	46	46	50	82	78	100	37	35	37	46	
PGMG	77	77	78	77	83	66	100	52	43	77	
PGMH	72	72	74	71	77	59	50	100	50	68	
PGMI	58	58	62	82	88	67	44	53	100	59	
PGMJ	71	71	75	80	83	43	40	37	30	100	

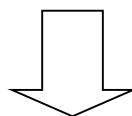
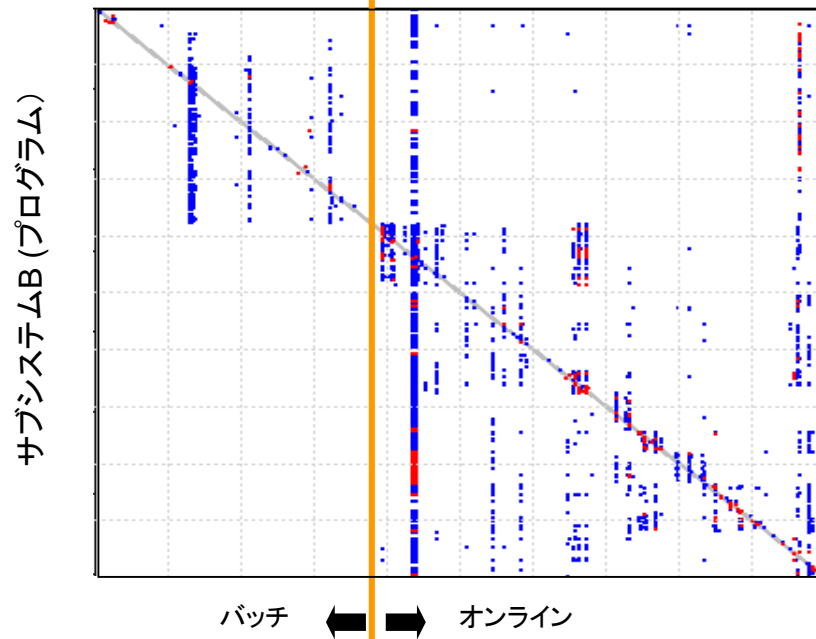
比較先、比較元双方が80%以上～90%未満の場合は青色、90%以上は赤色で表示します。比較対象の片側だけが80%以上の場合には青色または赤色による表示はされません。

- プログラム言語 PL/I

サブシステムA (プログラム)



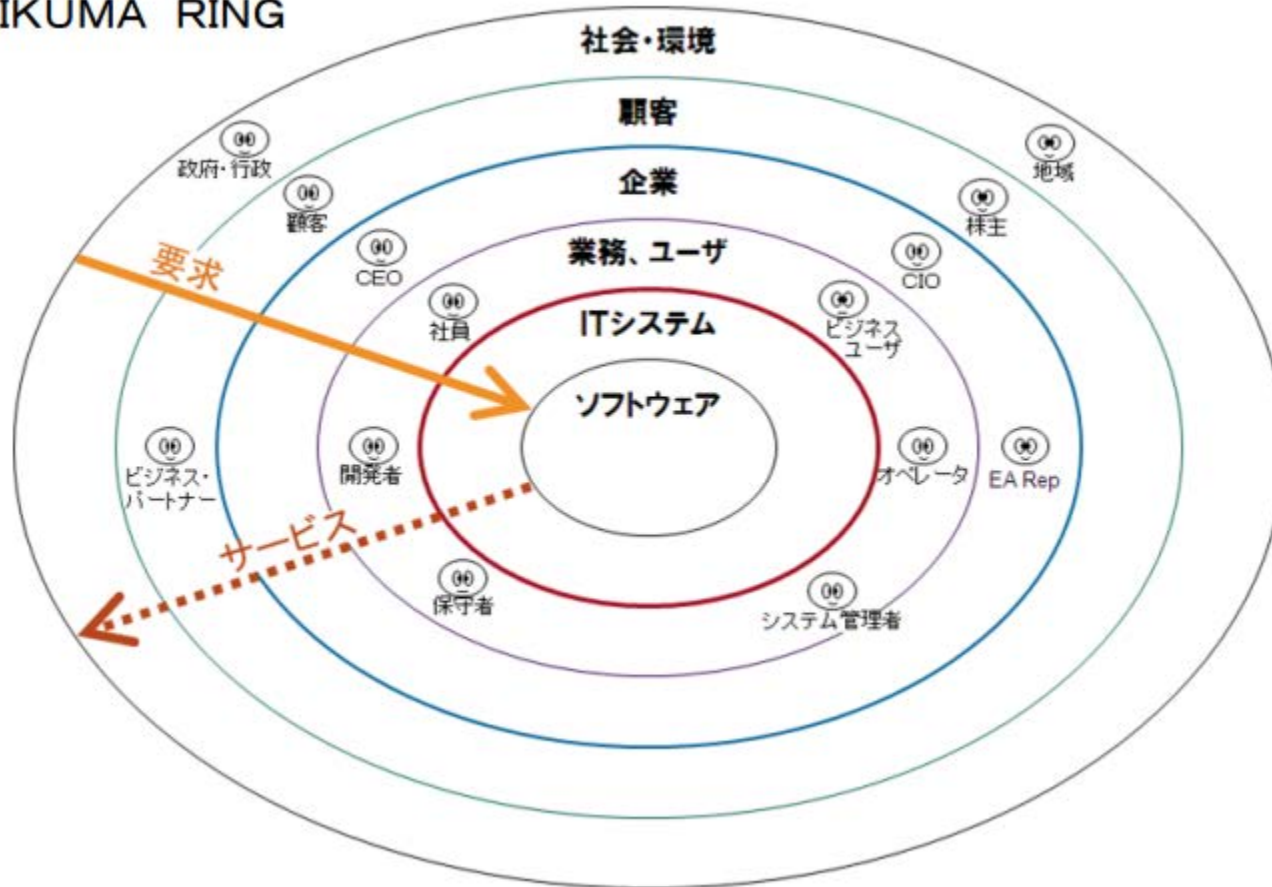
サブシステムB (プログラム)



システム規模で正規化した障害頻度は？

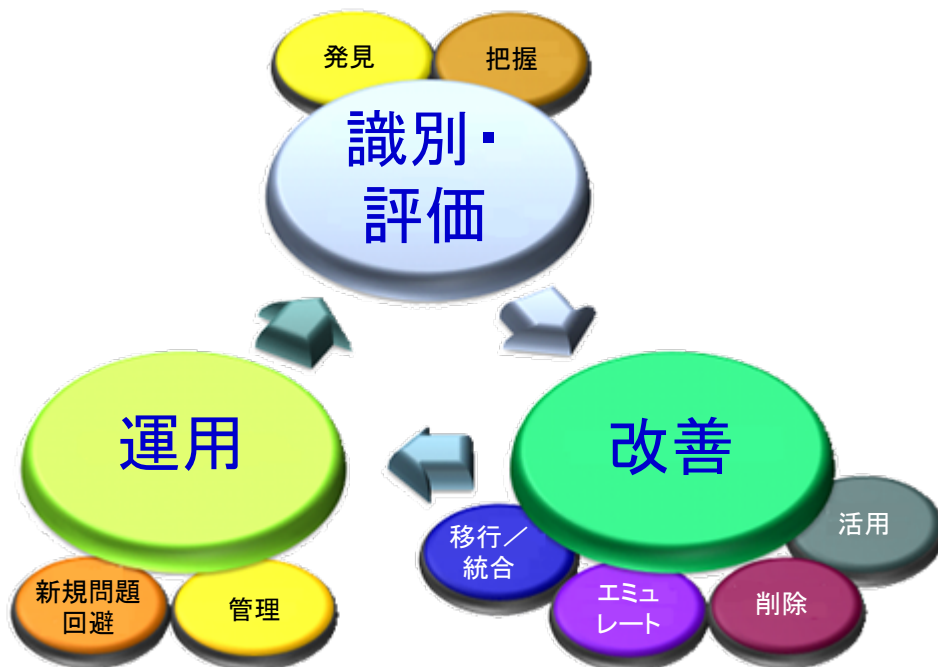
- ITシステム・ソフトウェアは様々なステークホルダと環境に取り囲まれている

KIKUMA RING



※ ITアーキテクトサミット2009
日本IBM 榎原彰技術理事 講演資料より 一部改変
(「要求の変化に対応する情報システム構築技術の適用に関する調査」IPA-SEC.)

レガシー・コントロール・ループの中で継続して実施が必要なアクション



レガシー・コントロール・ループがもたらす効果

- ビジネスを変え、市場の変化に対応可能
- 環境の変化に応じてビジネス・オペレーションを管理可能
- ビジネスに不可欠なアプリケーションの運用を強化
- 他との差別化

識別・評価

レガシーの発見・評価

- 論理的、物理的両面からレガシーを見つけ、そのビジネス的価値を評価します。

レガシーの把握

- レガシーを有効に管理するために、資産を把握します。

レガシーの移行、統合

- レガシーを新しいソフトウェアやインフラストラクチャーに移行します。
- ビジネス・モデルやビジネス・プロセスも合わせて検討します。
- クラウドやSOAを活用します。
- レガシー・コードをラップします。(例: GUI部分の新規追加)

冗長性の除去

- ビジネス的価値の高いレガシーを強化します。
- ビジネス的価値の低いレガシーを停止します。

レガシーの活用

- レガシーの機能を拡張します。
- 他アプリケーションのデータと連携させます。
- データのアプリケーション依存をなくします。

レガシーのエミュレート

- 新しいハードウェアで古いコードをエミュレートします。
- 他のビジネス・プロセスにより同等の機能を実現します。

レガシー管理

- ビジネスに必要不可欠なレガシーを維持し続けます。
- レガシーを管理する資源(人的資源やハードウェア等)を強化します。

新たな問題の回避

- レガシー運用計画
- レガシー管理カウンスル(「レガシー検証」への投資)

改善: 利用または削減

運用

※ IBM 2010 Global Technology Outlook ,
Future Of Legacyより引用

End Of File