

実績にもとづく COBOLモダナイゼーションのススメ

2015年 4月23日

富士通株式会社

アプリケーションマネージメント事業部

モダナイゼーションサービス部

吉田禎彦

- 見える化のススメ
- 構成管理のススメ
- ドキュメント整備のススメ
- モダナイゼーションのススメ
- 品質保証のススメ
- モダナイ後の開発/保守
最新技術導入のススメ

■ システム移行のきっかけ

- ハード保守が近づいた
- 見識者がいなくなった(老兵が去った)
- ちょっとしたメンテも予定以上のコストになった

■ システム移行時の希望

- 現行は特に問題なく動いているシステム。だから、手間は掛けたくない
- コストはできるだけ抑えたい
- 業務仕様が不明確のため、現行は踏襲したい

■ システム移行の傾向

- 膨大に膨れ上がった資産(どこから手をつけたらいいの?)
- 知らない複雑な仕組み(そのままでいいから動かして)
- 理解できない業務ロジック(この例外は何?)

- いつまでも変わらない文法
- 時代に合わせてベンダー毎にエンハンスされる機能
- 蓄積されたシステム環境
(運用も開発もデバックも慣れ親しんだ担当は天国)

**そもそも、COBOL言語の開発は一度覚えれば、
これほど継続しやすいものはない**

- 将来性の不安
(もはや、これだけで一生食える訳でない)
- わかりづらいオープン技術の利用
(COBOLインターフェースが用意されているものがない)
- 読めない書けないロジック
(メンテはベテランばかり、若手の引き継ぎがない)





見える化のススメ

- 何においても、現行をしっかり把握、見極めることが重要
闇雲に突っ走れば、必ずどこかでつまづきます

- **まずは、業務を把握して、整理しましょう**

移行を考える上で、重要な業務って何ですか？
みんな重要なら優先するべきものは何ですか？
その業務、本当に継続しますか？

- **つぎに、資産を把握して、整理しましょう**

使われなくなった資産はありませんか？
似たような資産沢山あって、うんざりしませんか？
複雑にからみあっていて、がっかりしませんか？

- **そして、資産を整頓して管理しましょう**

必要最小限のドキュメントは揃ってますか？
ライブラリ管理(版数管理)を行ってますか？

業務見える化の期待

■ 各業務を評価し、今後の施策優先順位を決める

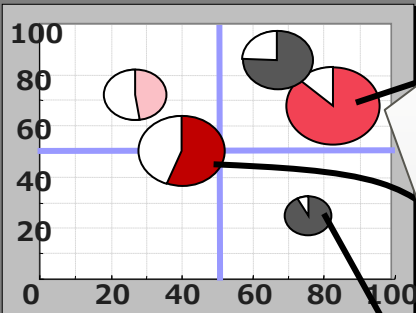
アプリケーション資産評価サービス

施策検討

次期施策

アプリケーションを多角的な観点から評価し現状把握と見える化

評価結果



- ・ 各業務を評価する
- ・ **業務に対する投資、効果等から価値を把握**
- ・ 業務単位の方方向性や優先順を見極める

評価内容を基に施策検討

各施策を最適に実施

案件毎の費用削減施策

次期システム検討

プラットフォーム移行

アプリケーション資産整理

アウトソーシングの活用

セキュリティ強化

アプリケーション構造の見える化

- アプリケーションの構造分析から機能コンポーネントを発見、地図形式で表現し、複雑度を直感的に見える化

特許
出願済

色が乱れた町並み

- 複数サブシステムのアプリケーションが入り乱れ保守性が悪い
(構造が複雑のため、改善が必要)

【凡例】

ビル（建物）：プログラム
ビルの高さ：プログラムの複雑さ
色：サブシステムの種類
ビル間の距離：プログラム間の関係性

高層ビル群

- 複雑なアプリケーションが入り乱れ保守性が悪い
(構造が複雑のため、改善が必要)

整然とした低層ビルの町並み

- サブシステムの範囲が定まっており、保守しやすい
(構造が簡素で、影響範囲を特定しやすい)

【事例】ソフトウェア地図活用例（流通業A社様）

◆ ソフトウェア地図で、システム更改に向けて既存資産の活用を判断

お客様のねらい

COBOL 3.3Ms

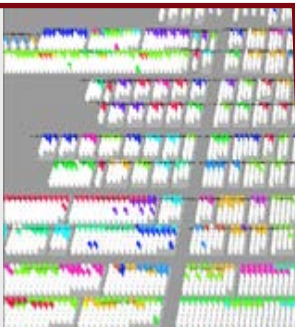
- ◆ システム更改において、既存資産を活用したい
- ◆ 既存資産の活用について判断材料が欲しい

ポイント

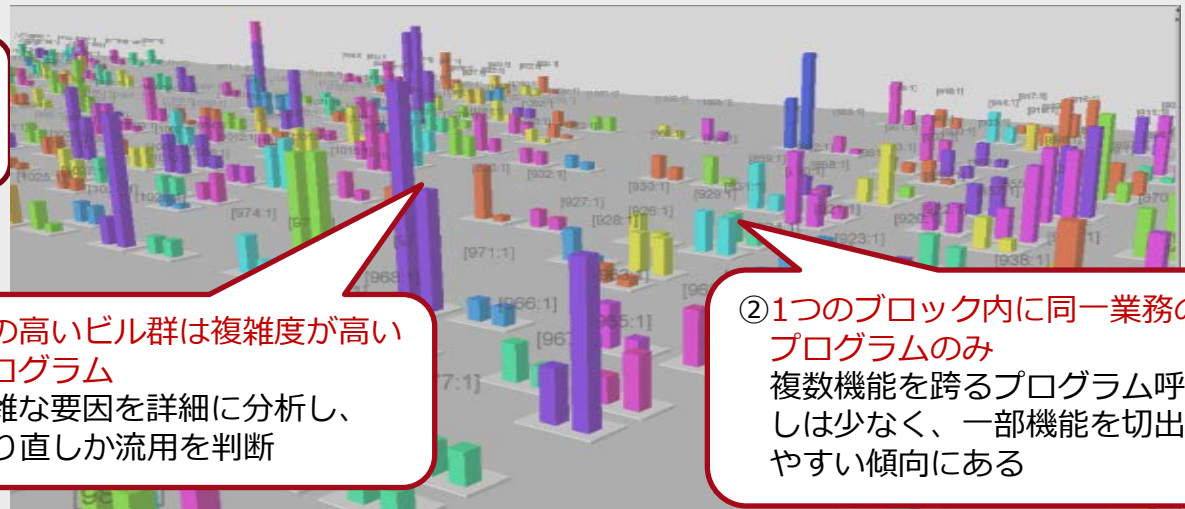
- ◆ プログラムやプログラムの管理状況を評価し、活用の判断材料とする
 - ① プログラムの稼働／未稼働を確認する
 - ② 段階移行を検討する際のプログラムの切り出し単位を明確にする
 - ③ プログラムの保守性を評価し、次期システムでの活用可否を判断する
 - ④ 移行難易度を評価する

診断結果

- ① 白色のビル群は不足プログラム
ライブラリ管理ができていない
ため、プログラムの整理が必須



- ③ 背の高いビル群は複雑度が高い
プログラム
複雑な要因を詳細に分析し、
作り直しか流用を判断



- ② 1つのブロック内に同一業務の
プログラムのみ
複数機能を跨るプログラム呼出しは少なく、一部機能を切出し
やすい傾向にある

診断結果を踏まえ、既存資産の活用（マイグレーション）を判断した
移行難易度評価、プログラム移行の試行を実施し、次工程以降の計画立案へ

アプリケーション資産の見える化

■ アプリケーションの現状を把握し、移行検討だけでなく運用保守にも活用可能

経営・企画担当

次期システム企画・検討材料として、

- ・ 有効資産規模の見積に活用
- ・ 既存資産の活用可否の判断

情シス管理担当

- ・ 複雑さに応じた要員配置、工数配分が可能
- ・ 複雑さに応じて見積を補正することで見積り精度を向上

現状の課題



稼働資産分析

稼働資産を見える化する



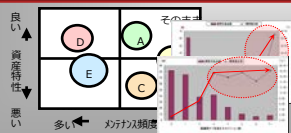
類似分析

類似（重複）処理を見える化する



資産特性分析

資産の複雑さを見える化する

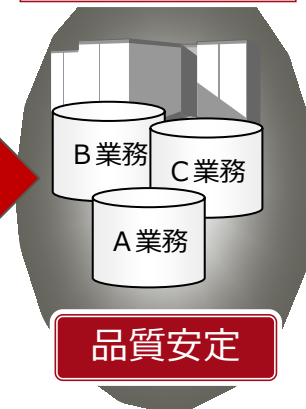


システム相関分析

処理とデータの関連を見える化する

	受発注	入庫	出庫	売掛	商品管
a マスタ		U	R	C	
b マスタ	R			C	
c マスタ	U		R		
d マスタ					R
e マスタ					R

理想とする姿



開発・保守担当

- ・ 不要資産の整理により保守作業を効率化
- ・ 影響範囲の把握により修正漏れを防止

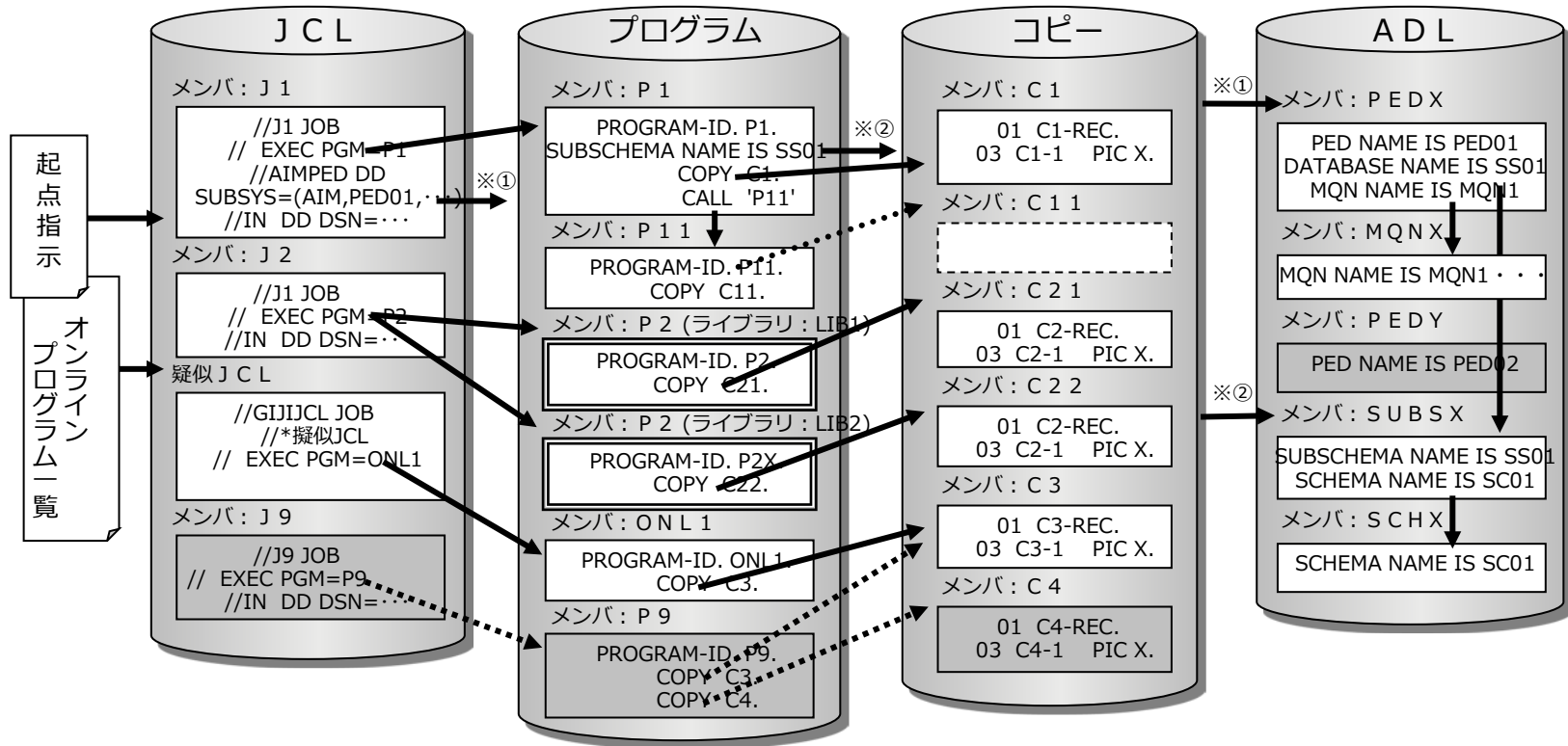
品質検証担当

- ・ 複雑さに応じた品質強化施策が可能
- ・ 複雑さに応じたテスト指標、規約策定により品質を安定

稼働資産分析

■ 既存資産の有効資産を把握し、稼働していない資産を洗い出す

- 不要なアプリケーション資産を整理する(削除する)
- 影響調査の品質・生産性向上にもつながる
- 確認作業量の軽減



類似分析

■ 類似(重複)資産を把握し、過剰な資産を洗い出す

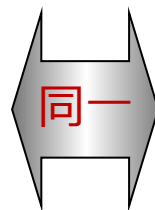
- メンバ名の一致・不一致に関わらず、COBOLソースの類似性を分析
- 同等処理より類似性を判断(表記上のブレを排除した比較)

<ソース1>

```
エディタ - [FILE1.COB]
ファイル(F) 編集(E) 位置付け(L) 表示(V) ツール(T) オプション(O) ウィンドウ(W) ヘルプ(H)
▽.....▽1.....▽2.....▽3.....▽4.....▽5.....▽6.....▽
024100 集計レコード出力処理 SECTION.▽
024200 WRITE 売上集計ファイル▽
024300 IF LINENO < 55▽
024400 THEN▽
024500 CONTINUE▽
024600 ELSE▽
024700 MOVE 売上集計レコード TO SAVERECORD▽
024800 PERFORM 見出し印刷処理▽
024900 MOVE SAVERECORD TO 売上集計レコード▽
025900 END-IF.▽
025100 WRITE 売上集計帳票 AFTER 1.▽
===== 終了 =====
```

<ソース2>

```
エディタ - [FILE2.COB]
ファイル(F) 編集(E) 位置付け(L) 表示(V) ツール(T) オプション(O) ウィンドウ(W) ヘルプ(H)
▽.....▽1.....▽2.....▽3.....▽4.....▽5.....▽6.....▽
082600 集計結果出力 SECTION.▽
082700 改ページ処理.▽
082800 IF LINENO < 55▽
082900 GO TO 改ページ処理終了▽
083000 ELSE▽
083100 MOVE 売上集計レコード▽
083200 TO SAVERECORD▽
083300 PERFORM 見出し印刷処理▽
083400 MOVE SAVERECORD▽
083500 TO 売上集計レコード.▽
083600 改ページ処理終了.▽
083700 WRITE 売上集計ファイル.▽
083800 WRITE 売上集計帳票 AFTER 1.▽
===== 終了 =====
```



- ・表記上のブレを排除し、同様の処理を行っていると推測されるものを類似と判断する。
 - 空白や改行位置の相違、セクション名、段落名の相違、処理の前後関係の相違は不一致とみなさない。
 - GO TO文やCONTINUE文など、ロジック制御に関わる命令はコンペア対象としない。

<例> ファイル出力後に帳票出力するか、先に帳票出力してからファイル出力するかは問わず、
「XXファイルとXX帳票を出力するプログラムである」という観点で類似性を評価する。

- ・類似したソースをグルーピングすることで、類似状況の把握がしやすい。

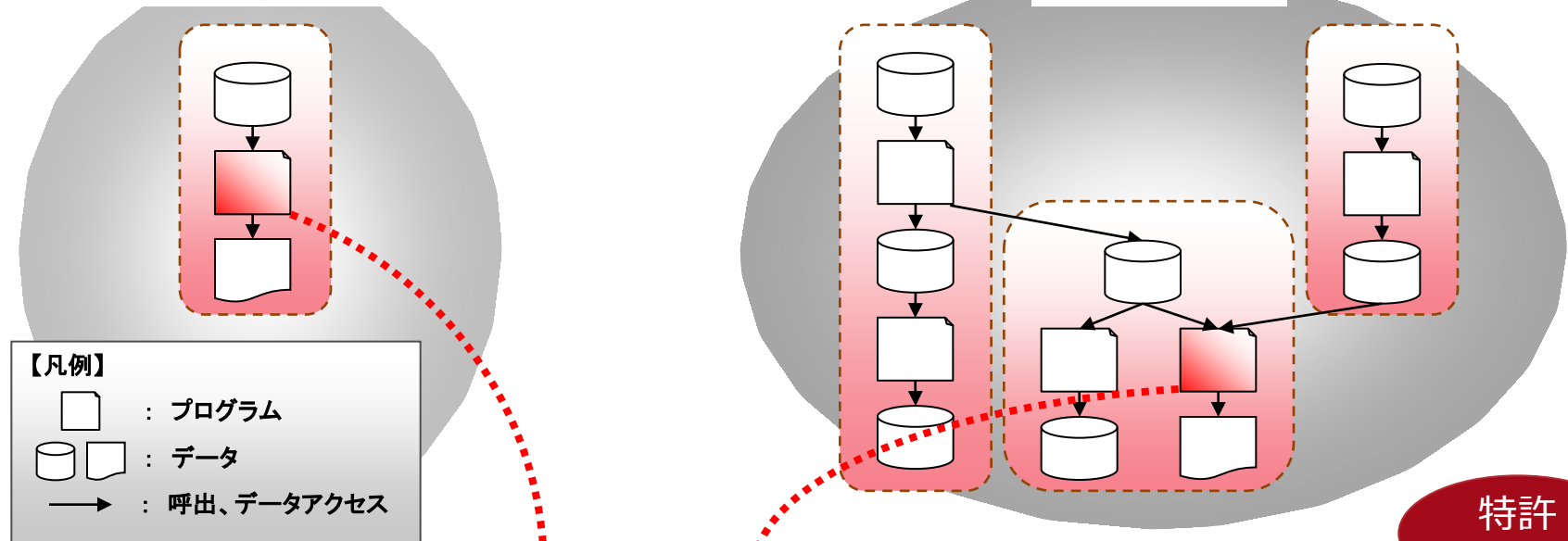
特許
出願済

資産特性分析 ～インパクトスケール®～

アプリ構造の複雑さが見える指標「インパクトスケール®」

シンプルな構造

複雑な構造



プログラムを修正・変更するときの**影響範囲**は大きく異なる
影響範囲度合いをプログラム単位に数値化で表現(特許出願済)

特許
出願済

影響範囲（調査範囲・テスト範囲）とは、

- 入出力のデータの多さ
- プログラム呼出の多さ
- データ・プログラムを介した繋がりの方数



インパクトスケール®によって
定量的に見える化可能

システム相関分析

- ジョブ、プログラム、アクセスするデータからCRUD図を作成
- データから関連するプログラムを特定し、システムへの影響を確認する

調査したい範囲に絞った検索が可能。

横軸には、プロセス（プログラム、ジョブ、サブシステム）を表示。

「ファイルプロセスマトリクス」

縦軸には、データを表示。

- ・ 一般ファイル（データセット）
- ・ NDBレコード
- ・ RDBテーブル
- ・ VSAMファイル（データセット）

このプログラムに関わる（影響する）の範囲をマトリクスから調査する。

C（生成）、R（参照）、U（更新）

「検索条件入力」

Microsoft Excel - #FPM検索システム_for_EXCEL2000.xls

MSゴシック 10 B I U

A3

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	[検索単位: プロセス], [出力単位: JCLMEM], [JCLMEM: *]																											
2	FILETYPE	FILEID	集計値																									
3		使用区分	200																									
4	F	CJDT.JFS99.CI01NCH1	1	C	C	C																						
5	F	CJDT.IKK10	14	C	C	C																						
6	F	CJDT.IKK11	14	C	C	C																						
7	F	CJDT.IKK12	14	C	C	C																						
8	F	CJDT.IKK13	11	C	C	C																						
9	F	CJDT.IKK14	11	C	C	C																						
10	F	CJDT.IKK17	8	C	C	C																						
11	F	CJDT.IKK1B	11	C	C	C																						
12	F	CJDT.IKK1C	3	C	C	C																						
13	F	CJDT.IKK1E	11	C	C	C																						
14	F																											
15	F																											
16	F																											
17	F																											
18	F																											
19	F																											
20	F																											
21	F																											
22	F	CJDT.IKK18	11	R	R	R																						
23	F	CJDT.JFS99.CI01NCH3	1	C																								
24	F	CJDT.ZTD16.TD1C8TH3	53	R	R																							
25	F	CJDT.ZTD16.TD1C8TH3																										
26	F	CJDT.ZTD16.TD1C8TH3																										
27	F	CJDT.ZTD16.TD1C8TH3																										
28	F	CJDT.ZTD16.TD1C8TH3																										
29	F	CJDT.ZTD16.TD1C8TH3																										
30	F	CJDT.ZTD16.TD1C8TH3																										
31	F	CJDT.ZTD16.TD1C8TH3																										
32	F	CJDT.ZTD16.TD1C8TH3																										
33	F	CJDT.ZTD16.TD1C8TH3																										
34	F	CJDT.ZTD16.TD1C8TH3																										
35	F	CJDT.ZTD16.TD1C8TH3																										
36	F	CJDT.ZTD16.TD1C8TH3																										
37	F	CJDT.ZTD16.TD1C8TH3																										
38	F	CJDT.ZTD16.TD1C8TH3																										
39	F	CJDT.ZTD16.TD1C8TH3																										
40	F	CJDT.ZTD16.TD1C8TH3																										
41	F	CJDT.ZTD16.TD1C8TH3																										
42	F	CJDT.ZTD16.TD1C8TH3																										
43	F	CJDT.ZTD16.TD1C8TH3																										
44	F	CJDT.ZTD16.TD1C8TH3																										
45	F	CJDT.ZTD16.TD1C8TH3																										
46	F	CJDT.ZTD16.TD1C8TH3																										
47	F	CJDT.ZTD16.TD1C8TH3																										
48	F	CJDT.ZTD16.TD1C8TH3																										
49	F	CJDT.ZTD16.TD1C8TH3																										
50	F	CJDT.ZTD16.TD1C8TH3																										

【事例】 見える化 (金融業B社様、製造業C社様) FUJITSU

◆ 類似資産を把握し、次期システムの開発規模を明確化

金融業B社様

・ 資産規模 COBOL 14Ms

類似分析より

共通化 1Ms

類似処理削減で 3Ms(20%)削減

製造業C社様

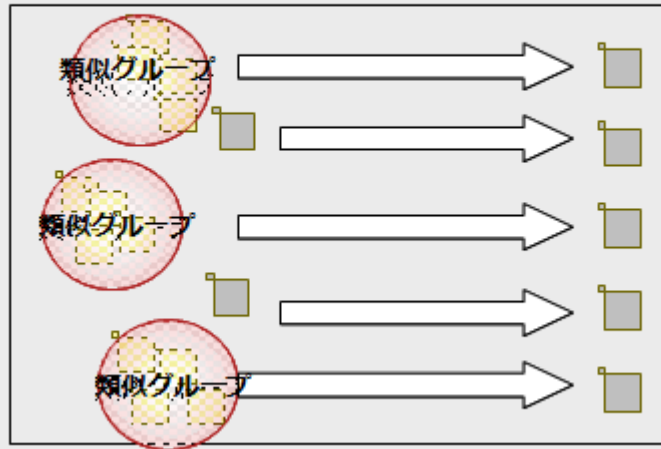
・ 資産規模 COBOL 2Ms

稼働資産分析、類似分析より
再構築対象を0.6Ms(30%)削減

適用概要 (B社様)

《《現行資産》》

《《次期システム》》

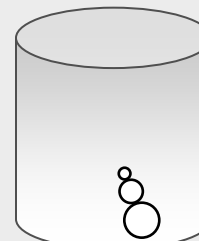


類似資産を汎用化し共通化

適用概要 (C社様)

《《現行資産》》

《《現行資産》》



類似分析
稼働資産分析



再構築対象

次期シスも
現行と同規模を
開発するとは思えない

再構築対象
規模が明確に



構成管理のススメ

■ 長年運用を続けていると以下のようなことは良くある話

- ソースがない!!
どうしても見つからない(大概は人災)
ベンダー提供であったため、ソース資産がない
- 実行モジュールと合っていないソース資産
緊急のエンハンスが相次ぎ、いつの間にか最新版同期ずれ
コピペ開発でどれと同期がっているのかわからない
- 多数の管理外資産
何故か組み込まれている実行モジュール
テスト検証のつもりが個人管理資産

■ どうにもならない資産

- 移行時には、実行モジュールからリバースできないか等の相談も多い
⇒ ソースが存在しない、同期の取れない資産は、手遅れ
- 諦めて使い続けて下さい
- 諦めて捨てて作り直して下さい

こうならないためにも、資産の管理は重要です。
さらにオープンシステムと混在したシステムへ変化し、
管理対象とする資産は益々増加しています。

なぜ、構成管理が必要なのか

- 構成管理のルールがないプロジェクトでは、ちょっとした一人の不注意や過信がプロジェクト全体に影響し、取り返しのつかないトラブルへつながりやすい



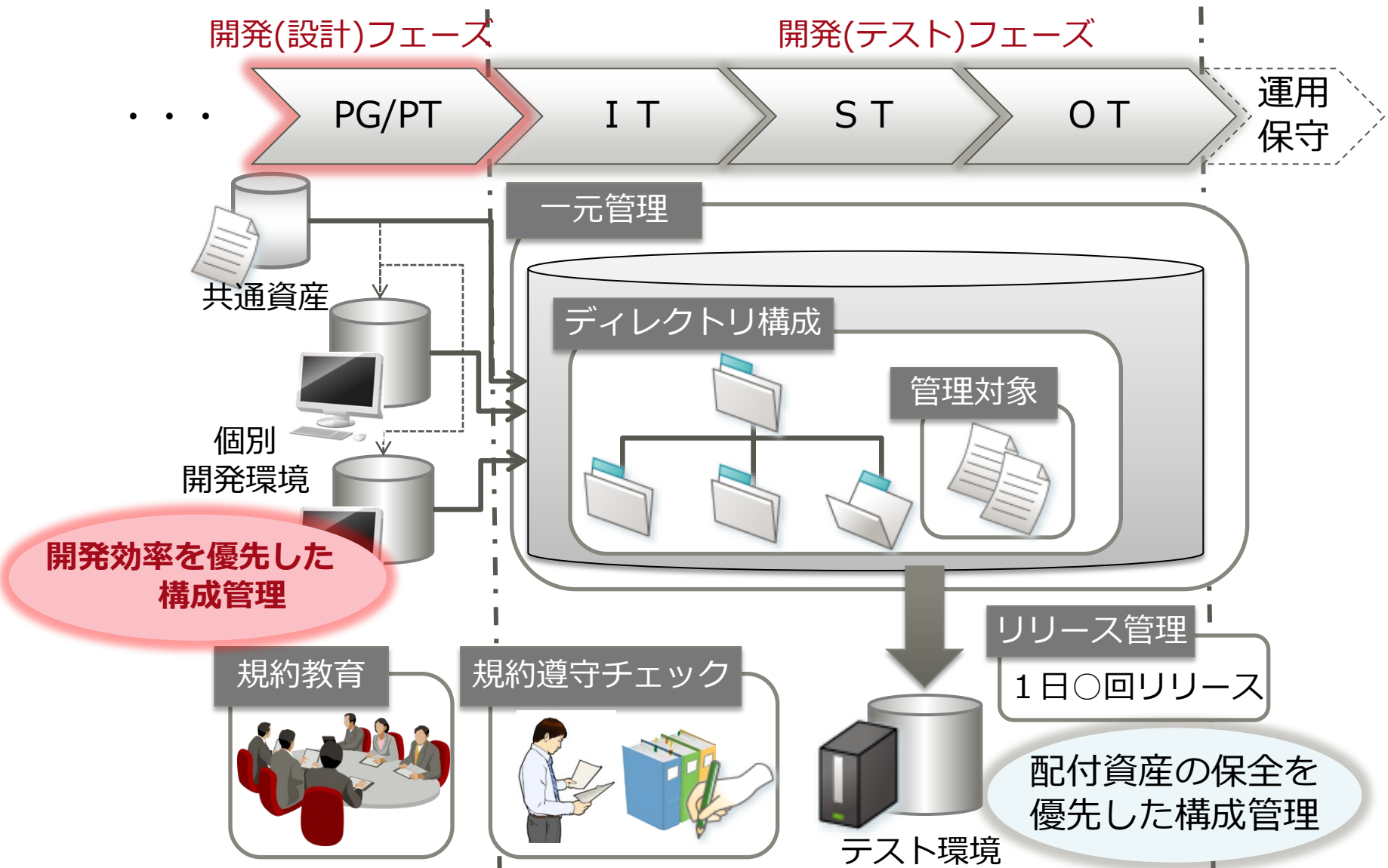
構成管理がなければ一人のミスが大きなトラブルにつながる



構成管理があればトラブルの未然防止やトラブル後の対応が容易になる

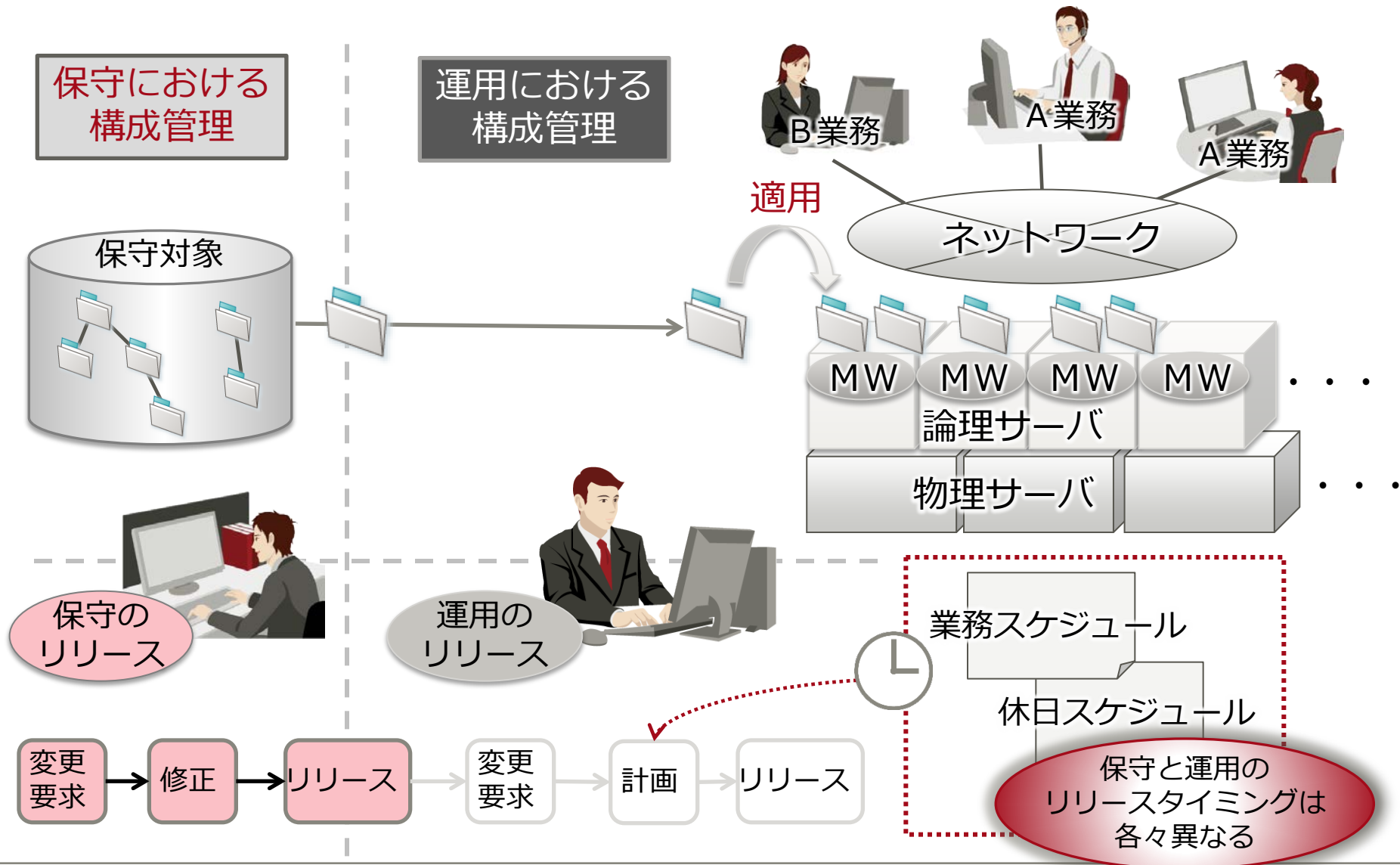
開発（設計・テスト）における構成管理

- 工程毎に資産を適切に管理し、作業進捗に合わせた変更・リリースに対応する

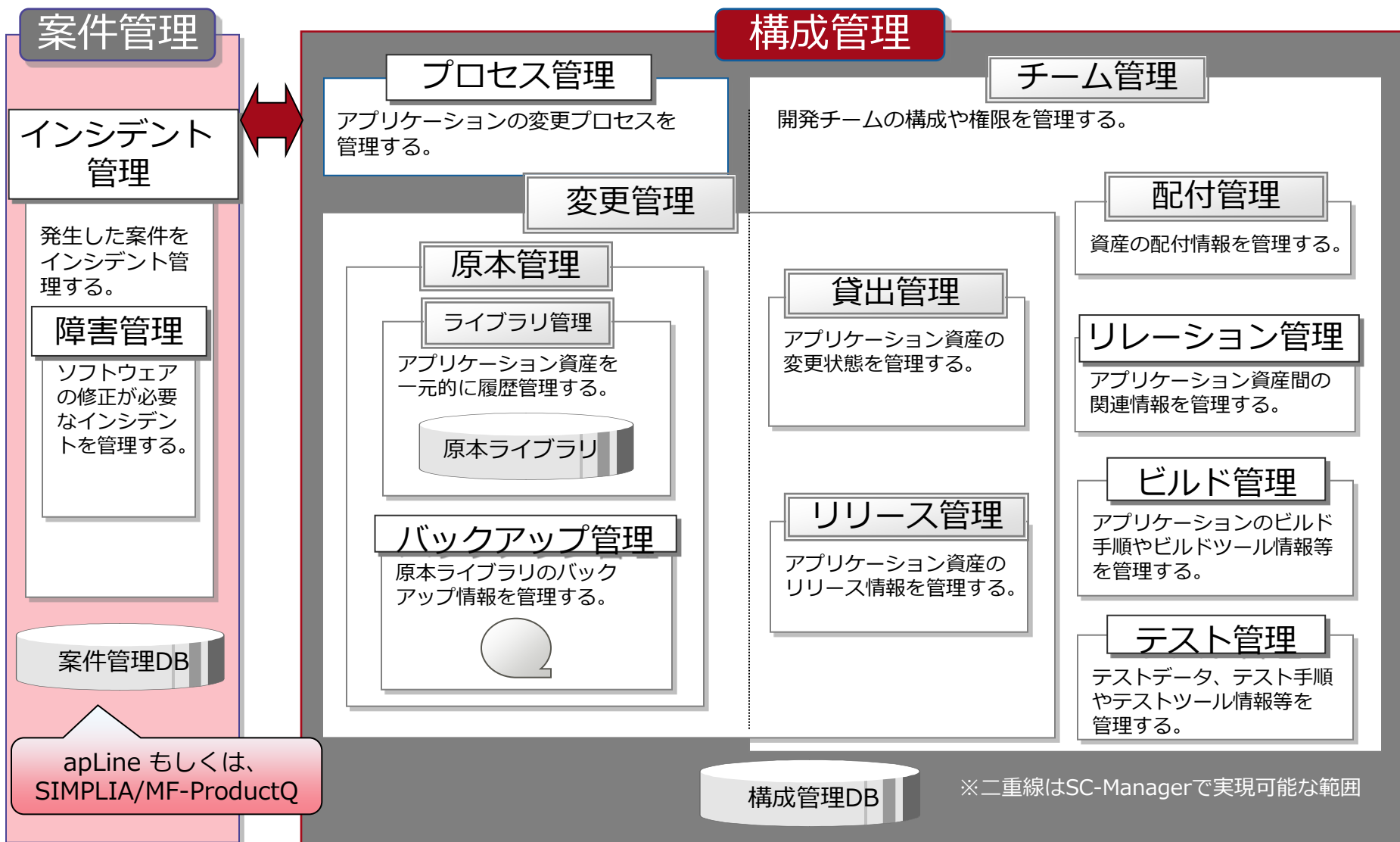


運用・保守における構成管理

■ 保守のリリースは運用の変更のトリガとなる



ソフトウェア構成管理の構成



【事例】アプリケーション運用保守プロセス適用事例(金融業C社様)

- ◆ アプリケーション運用保守のプロセスを標準化と構成管理ツールSC-Managerの導入を実施したことで、お客様の業務負荷軽減とトラブルを防止できた。

導入経緯

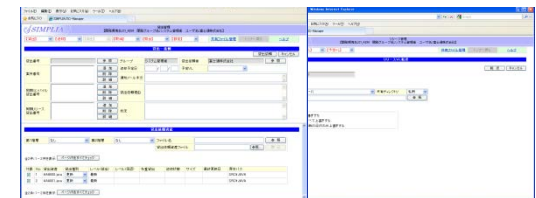
- 異種プラットフォーム（Windows,Solaris）上に存在する開発資産の一元管理がしたい。
- UNIX資産、Windows資産の整合性を取るために、開発者の作業負荷が高くなり、エラーが発生している。
- リリースルールが不明確であり、リリースミスが発生している。

環境概要

開発形態：集中型
開発対象業務：支店業務、照会業務他のサーバ構築、Solarisサーバ集約
開発要員：200名（ピーク時）
開発規模：2.5 Mstep
開発言語：C, C++、Java、COBOL

FUJITSU Software SIMPLIA

ソフトウェア開発保守支援システム



効果

事例

適用前

適用後

コンパイル時に、ライブラリ不整合によるエラーがなくなった。

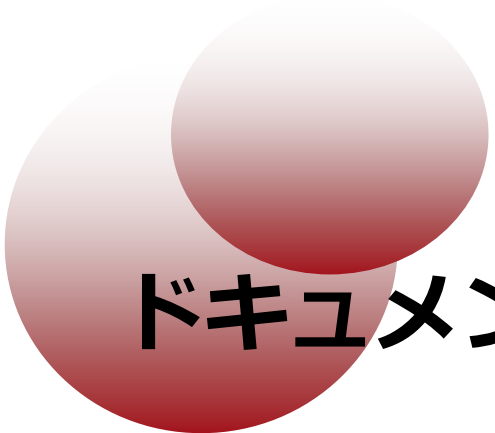
【IT工程障害発生件数】
1～2件/週（週間全3回中）

【ST工程障害発生件数】
0件/週（週間全3回中）

リリースルールが明確化、簡略化され、要員への浸透がスムーズになった。

【リリース1回の作業時間】
7H × 2名 = 14H

【リリース1回の作業時間】
ツール実行 = 0H



ドキュメント整備のススメ

■ ドキュメントの鮮度は直ぐ落ちる。ソースからのリバーズは夢のまた夢

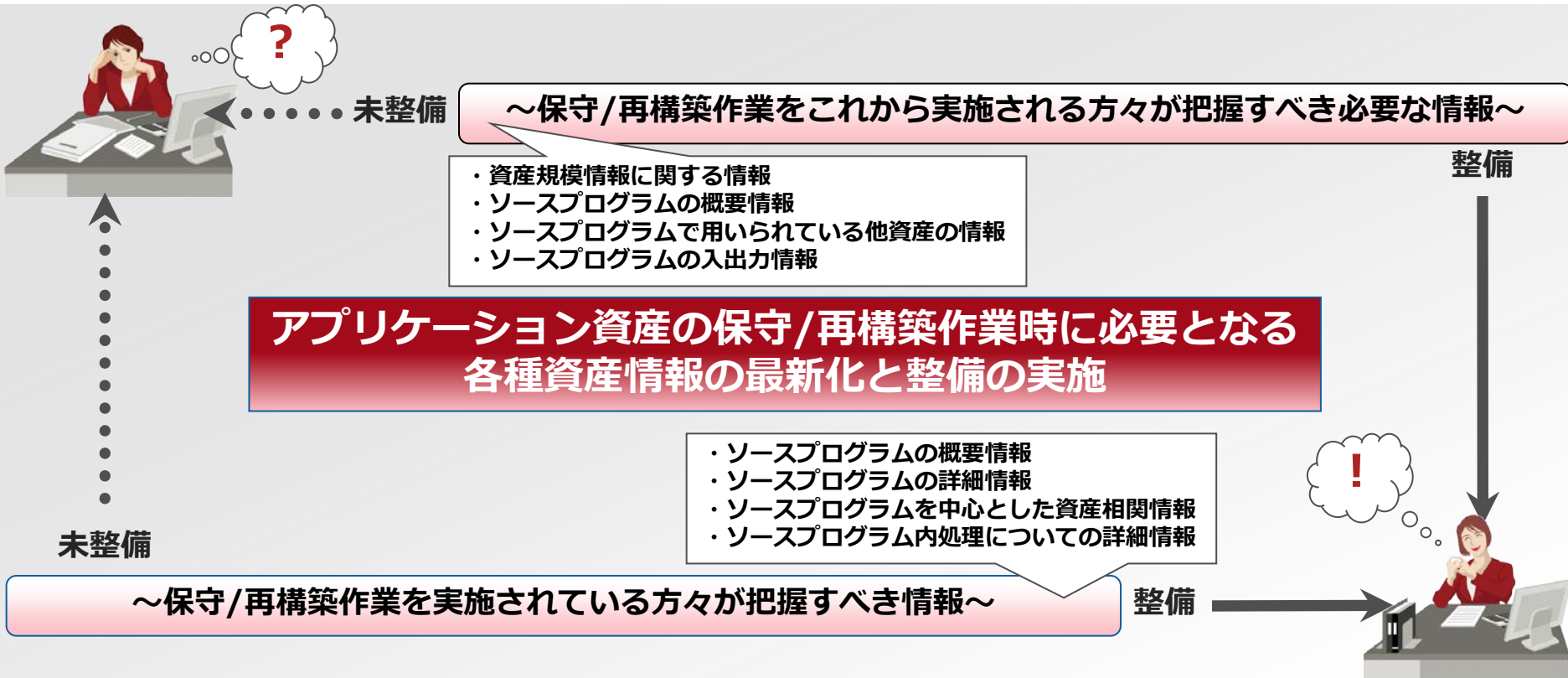
- ドキュメントが現行ソースと乖離している
 - わかっている人しか理解できない内容
 - いつの間にか変更されているプログラムソース
 - 忘れ去られた例外事項
- 引き継ぎできない資産
 - ソースだけでは業務と結びつかない
 - ひも解きできないジョブ構成
- ついていけない人材
 - 俺の背中を見よ!!(私は職人になりたくて配属された訳ではありません)
 - 取っ付きにくい環境(WEBで調べることもできません)
 - 古臭いシステムでモチベーションダウン(オープン最新技術の芝生は青く見える)

システムは意外と長生き。

安定した継続的な運用は、次の世代に引き継ぎにかかっている

ドキュメント整備のススメ

- 保守/再構築作業を効率的に行うためには、事前に資産やドキュメント、人に埋もれた情報の調査/整備を実施する必要がある
- 事前に必要な情報が整備できなかった場合、後々の作業に負荷が掛かる可能性大



★本サービスは、3種類の情報に分類されます。

資産詳細
情報

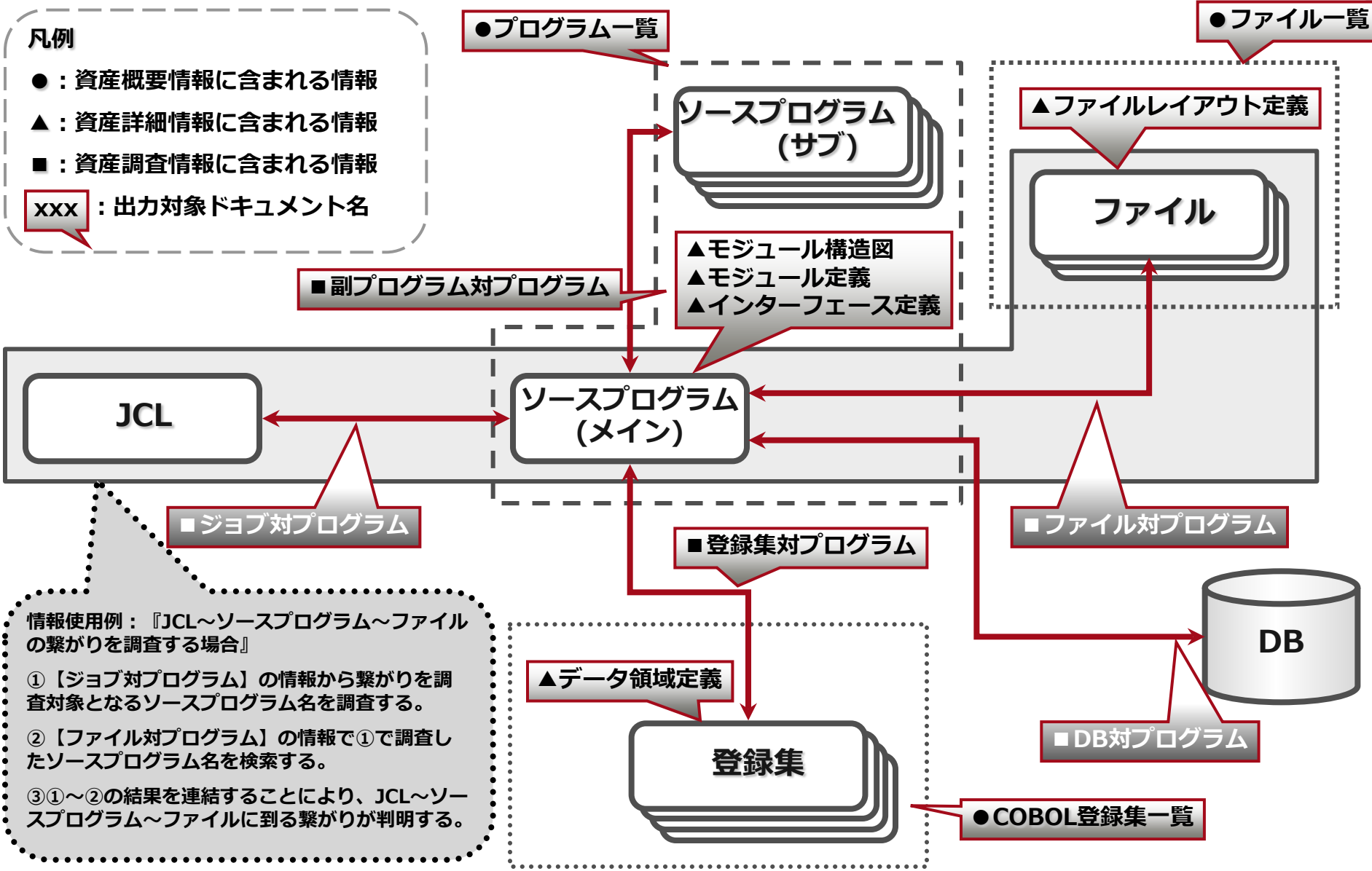
資産に関する仕様/設計情報の提供（43種類）
⇒モジュール構造図/インターフェース定義等

資産概要
情報

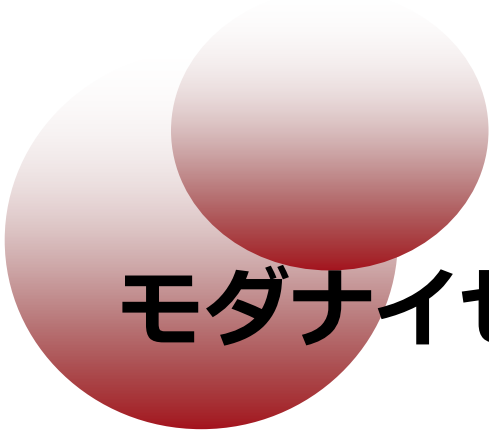
資産に関する一覧/規模情報の提供（31種類）
⇒資産一覧/ファイル一覧等

資産調査
情報

資産間の相互関係情報の提供（25種類）
⇒登録集対プログラム/ファイル対プログラム等



登録集対プログラム			
No	登録集ファイル名	登録集種別	プログラム名 プログラム定義ファイル名(行番号)
1	SALCPY01	登録集	D¥C0BDDC0例題¥COB¥SALC0B01.COB(31)
2	SALFMT00	画面定義体	D¥C0BDDC0例題¥COB¥SALC0B01.COB(31)
3	SALFMT10	画面定義体	D¥C0BDDC0例題¥COB¥SALC0B20.COB(31)
4	SALFMT20	画面定義体	D¥C0BDDC0例題¥COB¥SALC0B20.COB(31)
5	SALFMT21	画面定義体	D¥C0BDDC0例題¥COB¥SALC0B21.COB(32)
6	SALFMT00	帳票定義体	D¥C0BDDC0例題¥COB¥SALC0B00.COB(25)
7	SALFMT10	帳票定義体	D¥C0BDDC0例題¥COB¥SALC0B10.COB(34)
8	SALFMT20	帳票定義体	D¥C0BDDC0例題¥COB¥SALC0B20.COB(34)
9	SALFMT21	帳票定義体	D¥C0BDDC0例題¥COB¥SALC0B21.COB(34)
11	SALFMT40	帳票定義体	D¥C0BDDC0例題¥COB¥SALC0B40.COB(28)



モダナイゼーションのススメ

モダンイゼーションのススメ

■ レガシー機能を今風に

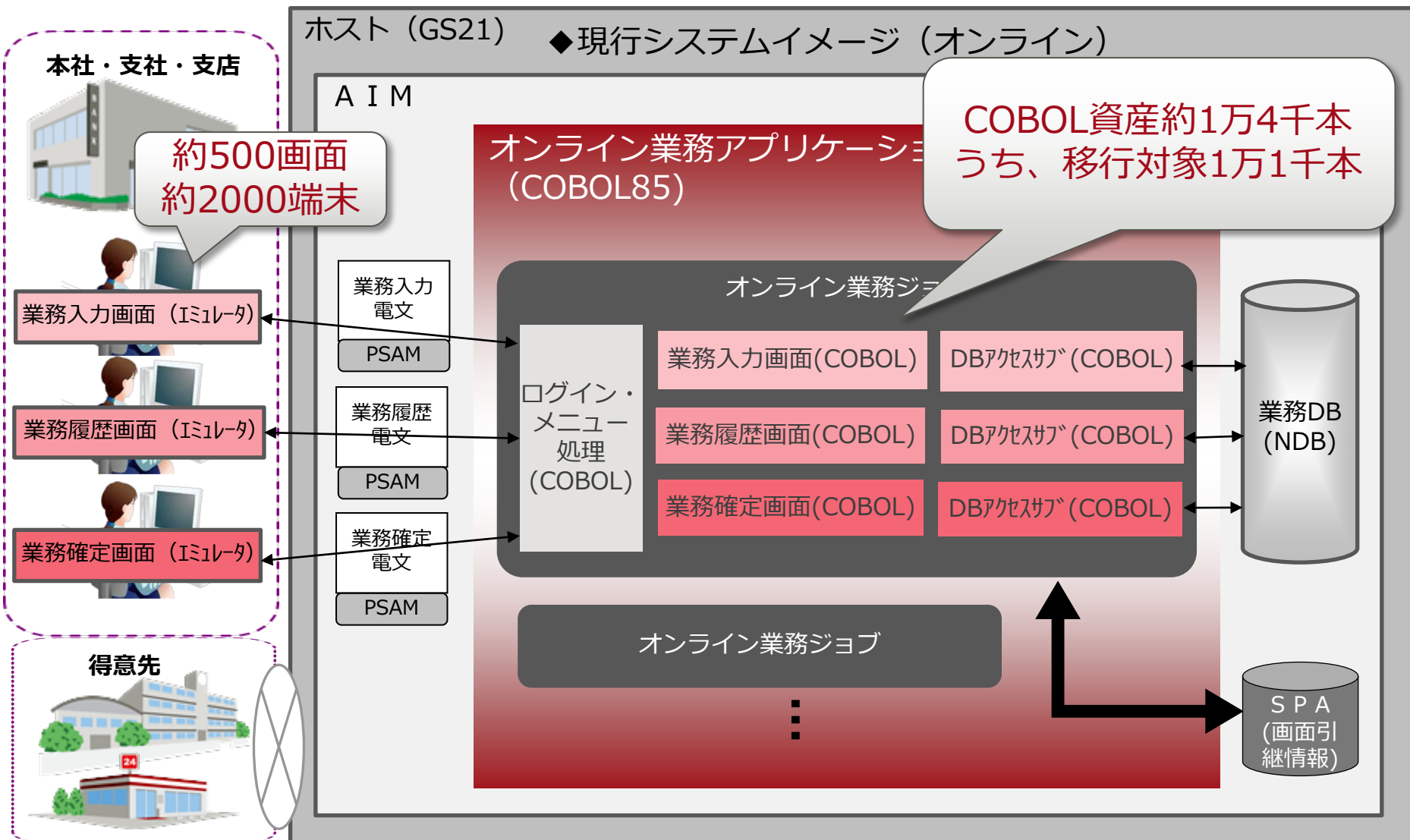
システム



■ 根本のロジックは現役続行!! システムとして今後も継続できる

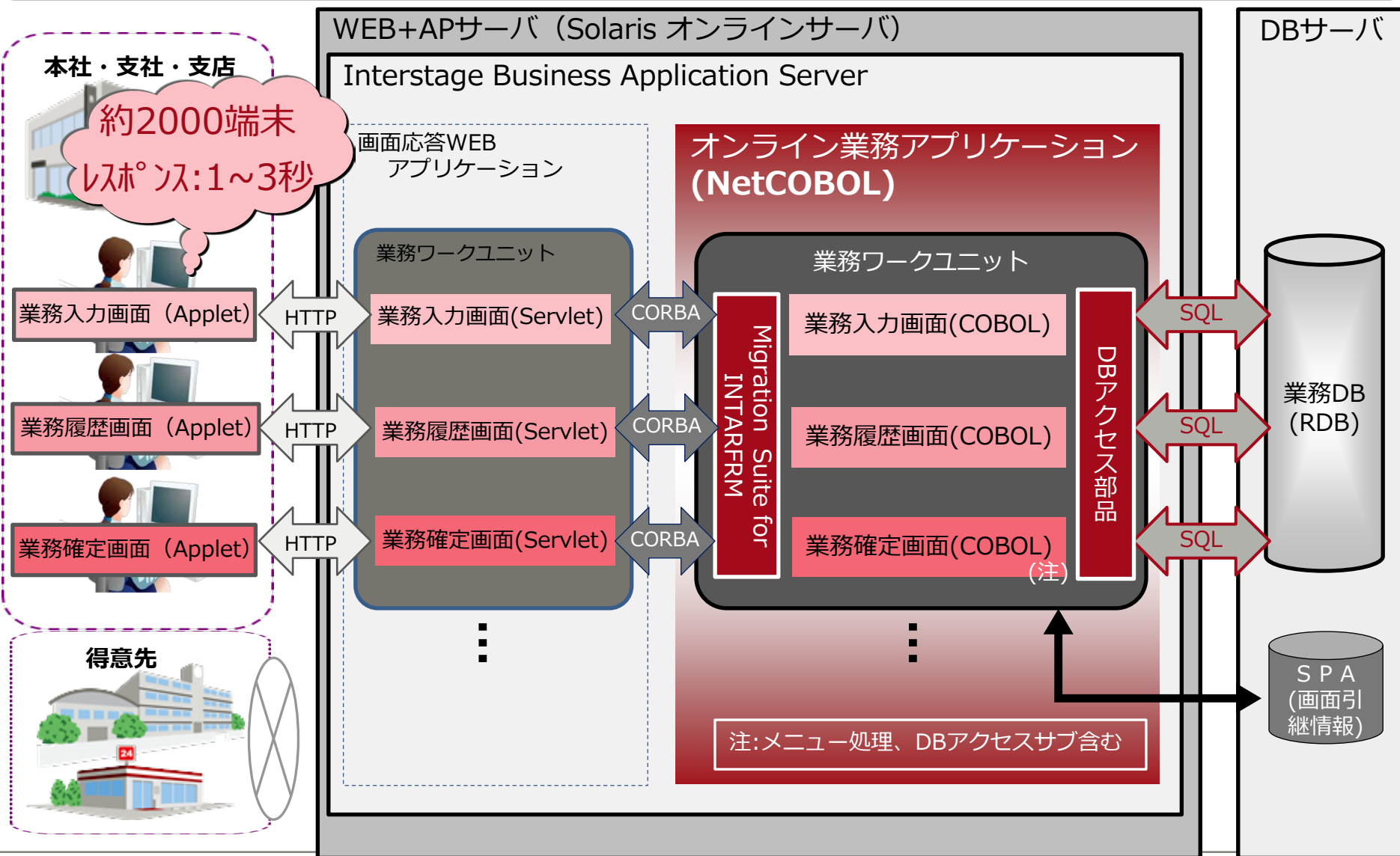
【事例】大規模事例 流通D社様

■ 大規模資産かつ多端末 現行同等のレスポンス要求



【事例】大規模事例 流通D社様

◆ ミドルウェア連携部品で業務ロジックの変更なし。要求通りのレスポンスを確保



【事例】複雑なDB例 製造B社様

- 複雑なDB構造に対するアクセスに従った業務ロジック
複雑なロジックとなり、解析困難なためRDB化も遅々として進まない

ホスト(GS21)

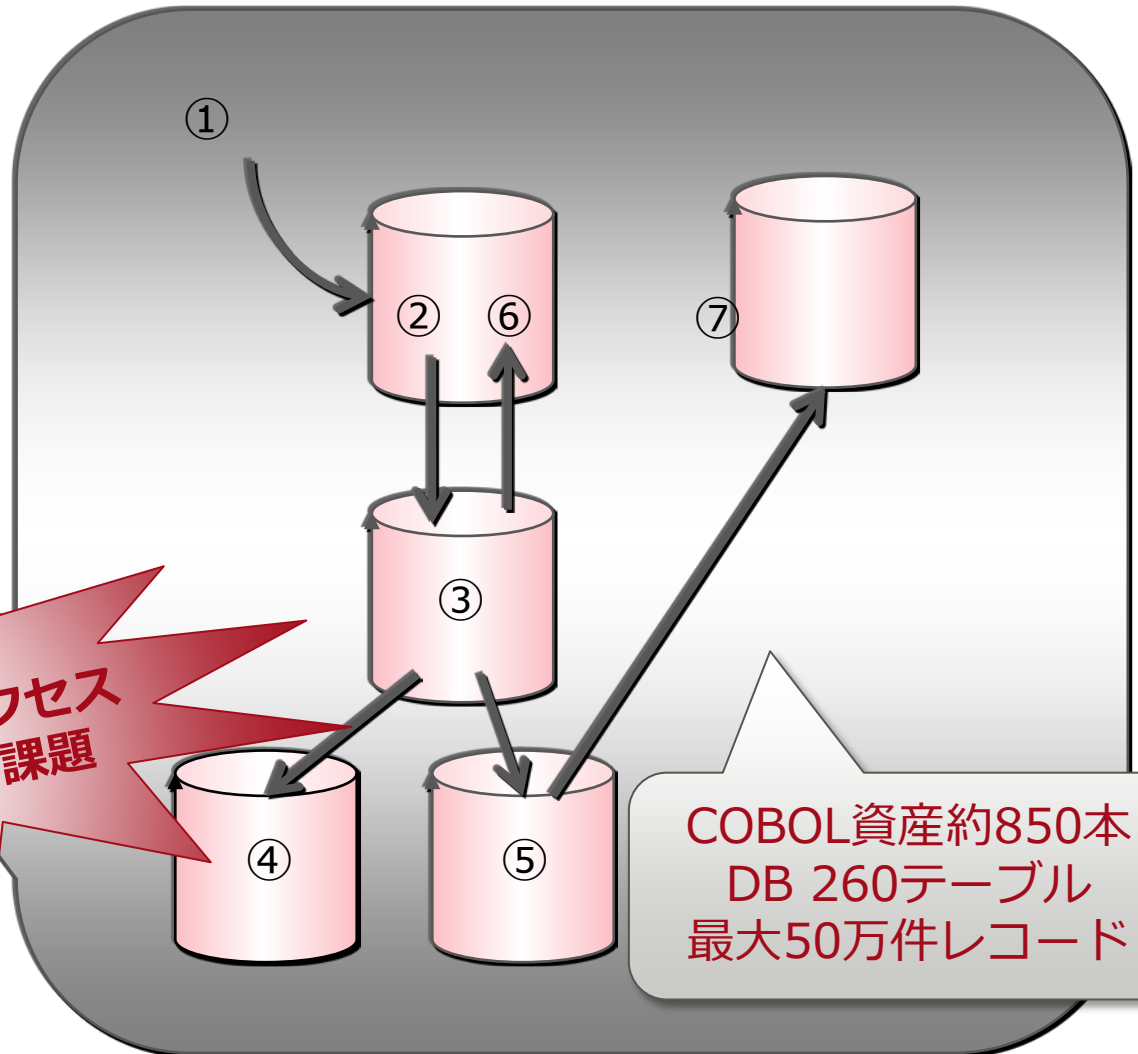
COBOL85

業務ロジック

DBアクセスロジック

GET ANY
GET NEXT
:

複雑なDB構造へのアクセス
処理の担保と性能が課題



COBOL資産約850本
DB 260テーブル
最大50万件レコード

【事例】複雑なDB例 製造B社様

- ◆ 現行DBに合わせてRDB化。業務ロジックの変更を最小限にアクセス部分を部品化で対応

Windows2008SV

NetCOBOL

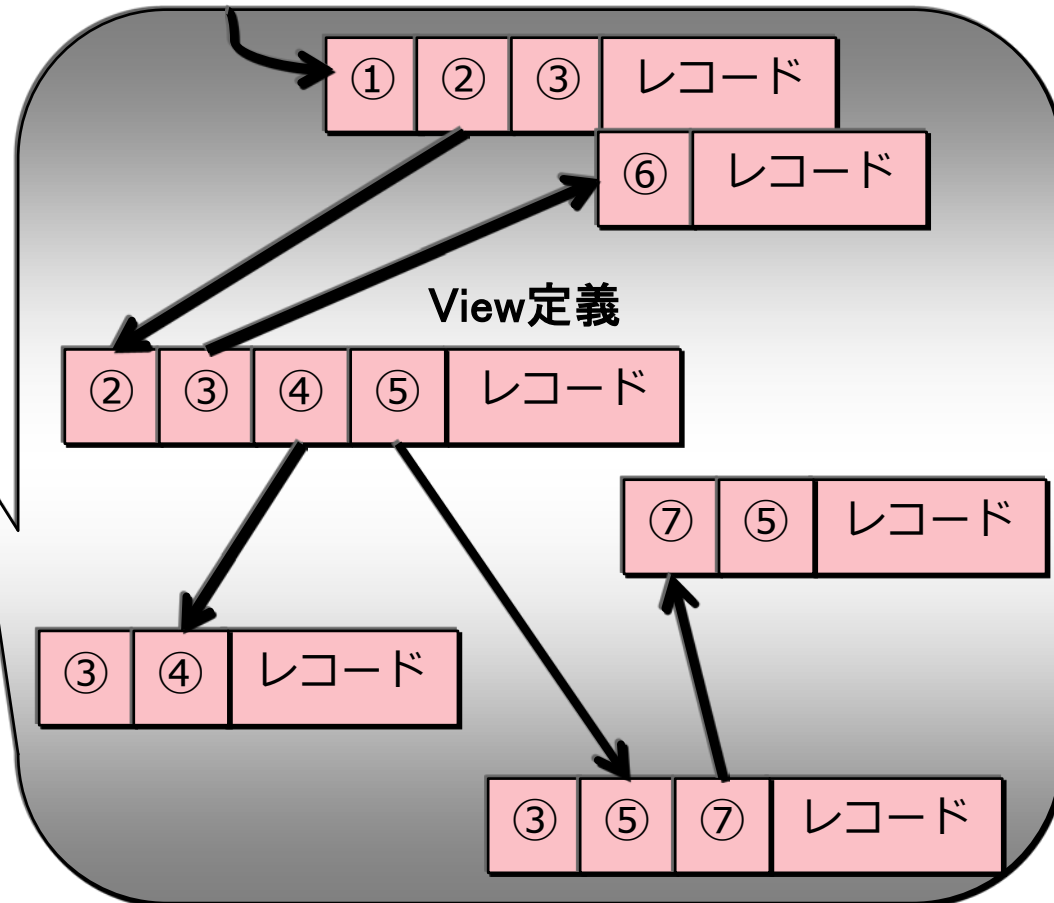
業務ロジックはそのままに

DBアクセスロジック

```
CALL 'DBSR'  
  USING GETANY  
CALL 'DBSR'  
  USING GETNEXT  
:
```

DB
部品

**DBアクセス部品で機能代替
COBOLロジックの変更はなし!
ホストと同処理時間**

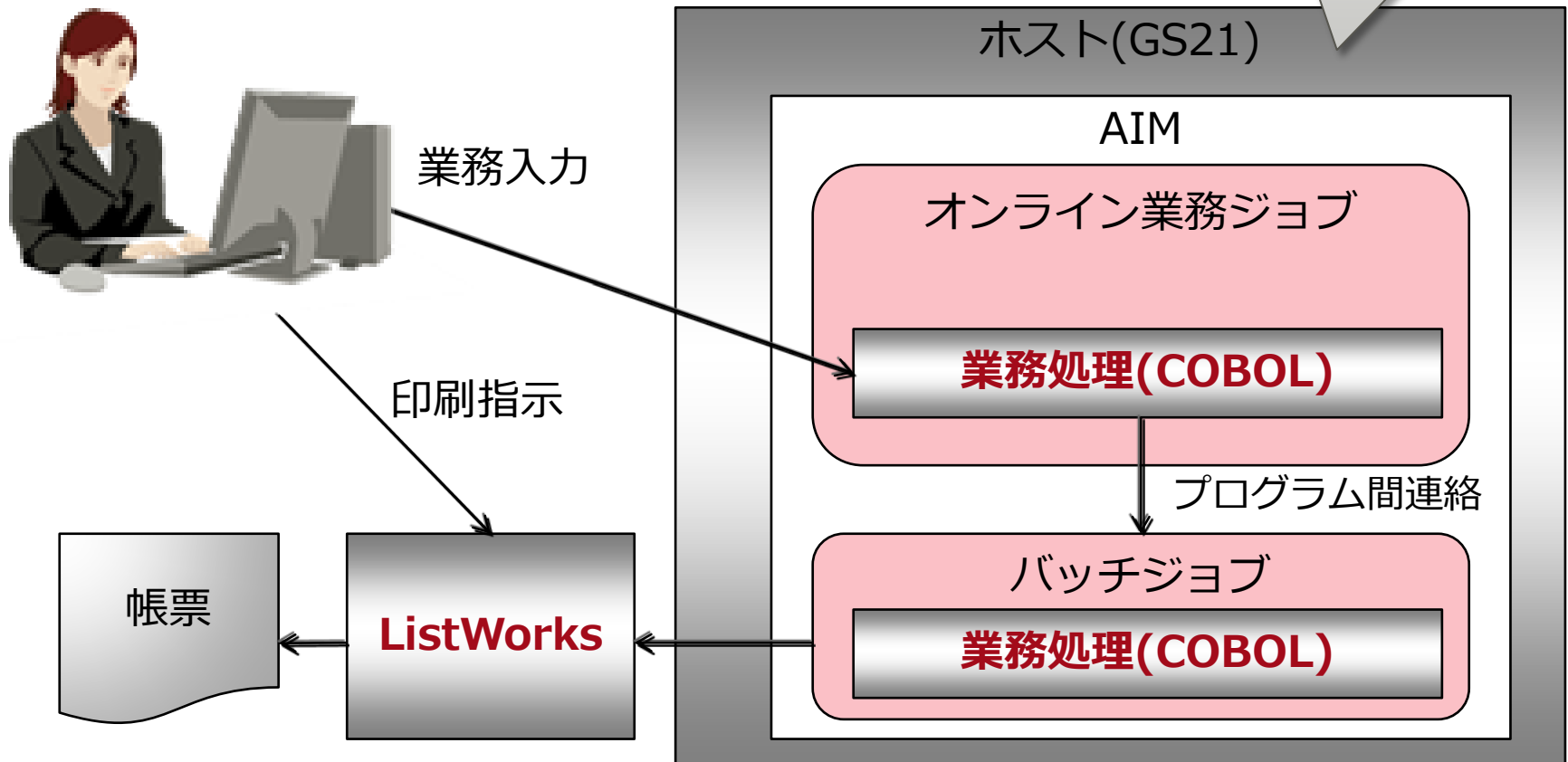


- 性能問題は常にある。チューニングは必須
現行ロジックの変更も計画にいらておくこと

【事例】 帳票出力事例 製造業F社様

- 従来はオペレータによる印刷指示で帳票出力
マイグレーション時にPDF化の要望

COBOL資産約1300本
うち、帳票 210本



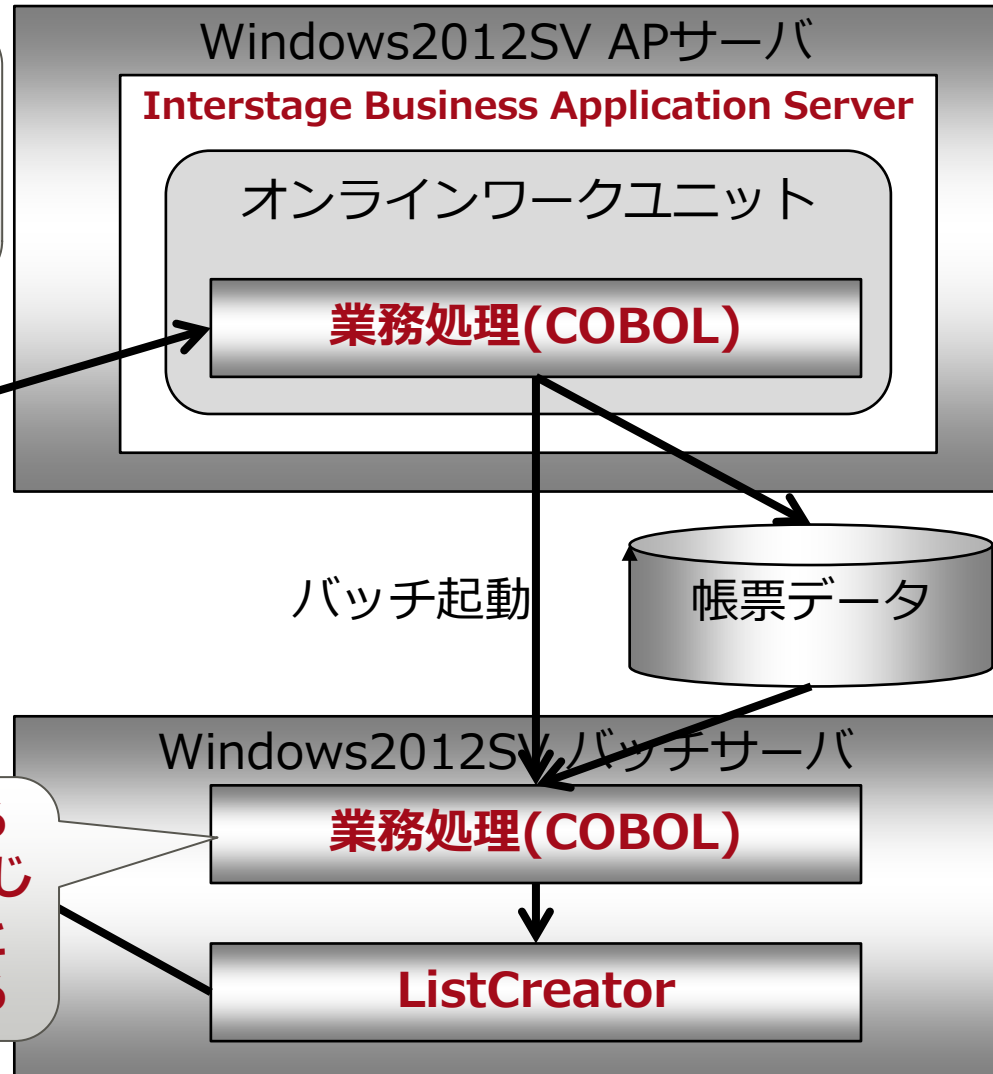
【事例】 帳票出力事例 製造業F社様

- ◆ 帳票をPDF化し操作端末画面に出力することで、帳票確認を容易にし作業効率化、ペーパーレスに貢献

- ・ 帳票をPDF化
- ・ 業務アプリは出力方式合わせ出力先の変更を実施
- ・ PDF帳票を操作端末へ表示



業務入力

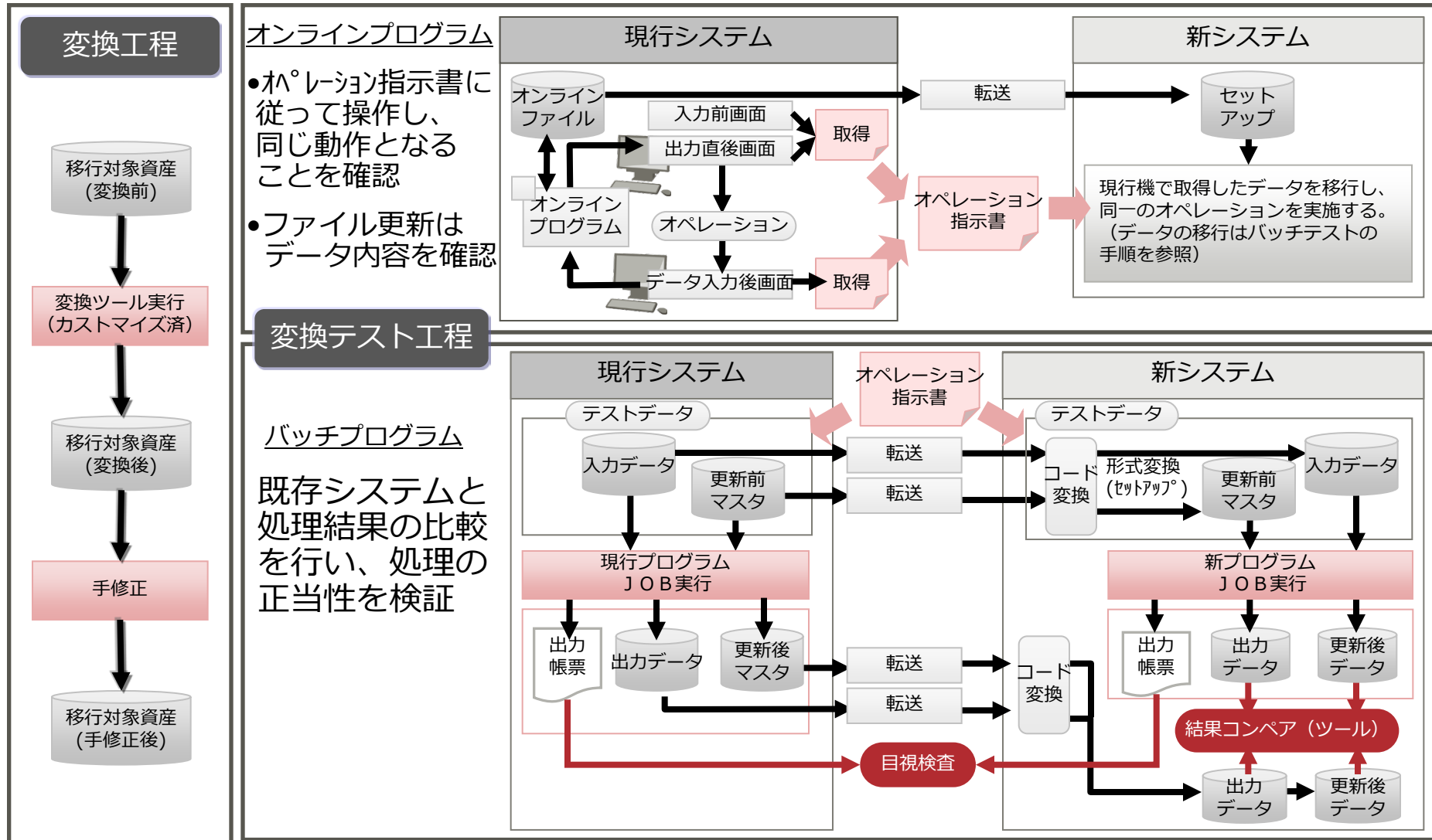


帳票用紙の違い、出力できるフォントの違いより現行と同じ帳票を出力するには、手間と時間がかかることを考慮する



品質保証のススメ

- 業務ロジックの変更を最小限におさえ、
現新比較より修正箇所(非互換箇所)を保証する



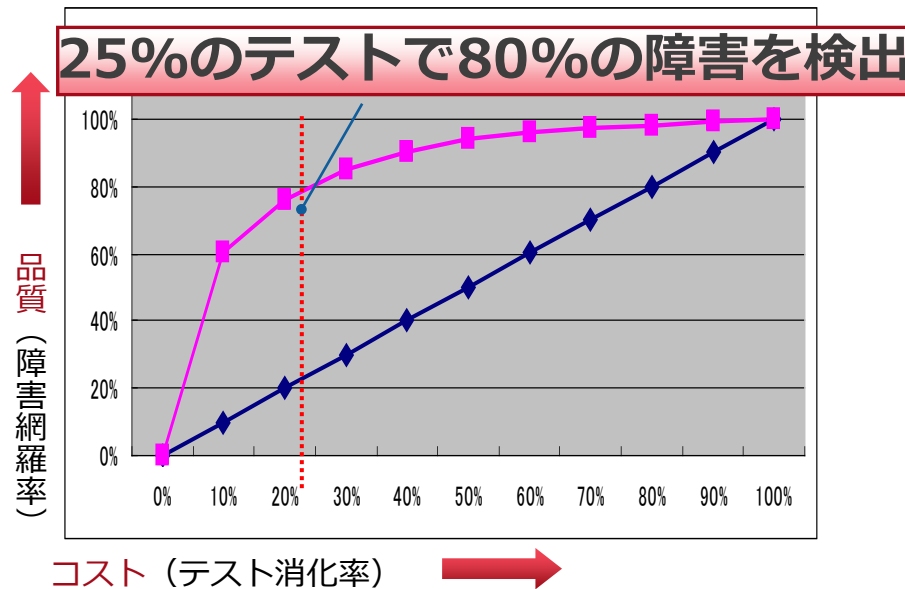
テスト工数削減の試み

◆ 品質を担保した上で、テストの絶対量を削減する

マイグレーション検証の特性（弊社事例の分析）

資産分析による集中テスト対象資産の選定

【マイグレーションプロジェクトにおける障害発生状況】



- 膨大な資産の検証作業を
均一に行うことはコスト増につながる
- 検証作業を重み付け。
品質リスクの高い資産を集中的に検証。
- 集中テスト（25%）、1パターンテスト（80%）
による品質とコストの両立。

非互換項目

	JCL	必須確認項目	STEP	PGM/PROC	JCL家換項目										COBOL家換項目									
					J	J	J	J	J	J	J	J	J	J	C	C	C	C	C	C	C	C	C	C
22	MSB14500		STEP000	IEFBRT4																				
22	MSB14500		STEP005	UTCOPY2																				
22	MSB14500		STEP010	MF901																				
22	MSB14500		STEP030	MGIM01																				
22	MSB14500		STEP040	MGPA73																				
22	MSB14500		STEP050	SORT																				
22	MSB14500		STEP060	SORT																				
22	MSB14500		STEP070	SORT																				
22	MSB14500		STEP080	SORT																				
22	MSB14500		STEP090	SORT																				
22	MSB14500		STEP100	MGPA75																				
22	MSB14500		STEP170	UTP80																				
23	MSB17100		STEP010	MF901																				
23	MSB17100		STEP020	IEFBRT4																				
23	MSB17100		STEP021	UTCOPY2																				
23	MSB17100	IMS DB	STEP030	MF1450																				
23	MSB17100		STEP060	SORT																				
23	MSB17100		STEP070	MGPI10																				
23	MSB17100	IMS DB	STEP080	MGIM05																				
23	MSB17100		STEP110	UTCOPY2																				
23	MSB17100		STEP111A	WKPTT2A																				
23	MSB17100	概要	STEP111B	WKPTT2B																				

【非互換項目マトリクス】

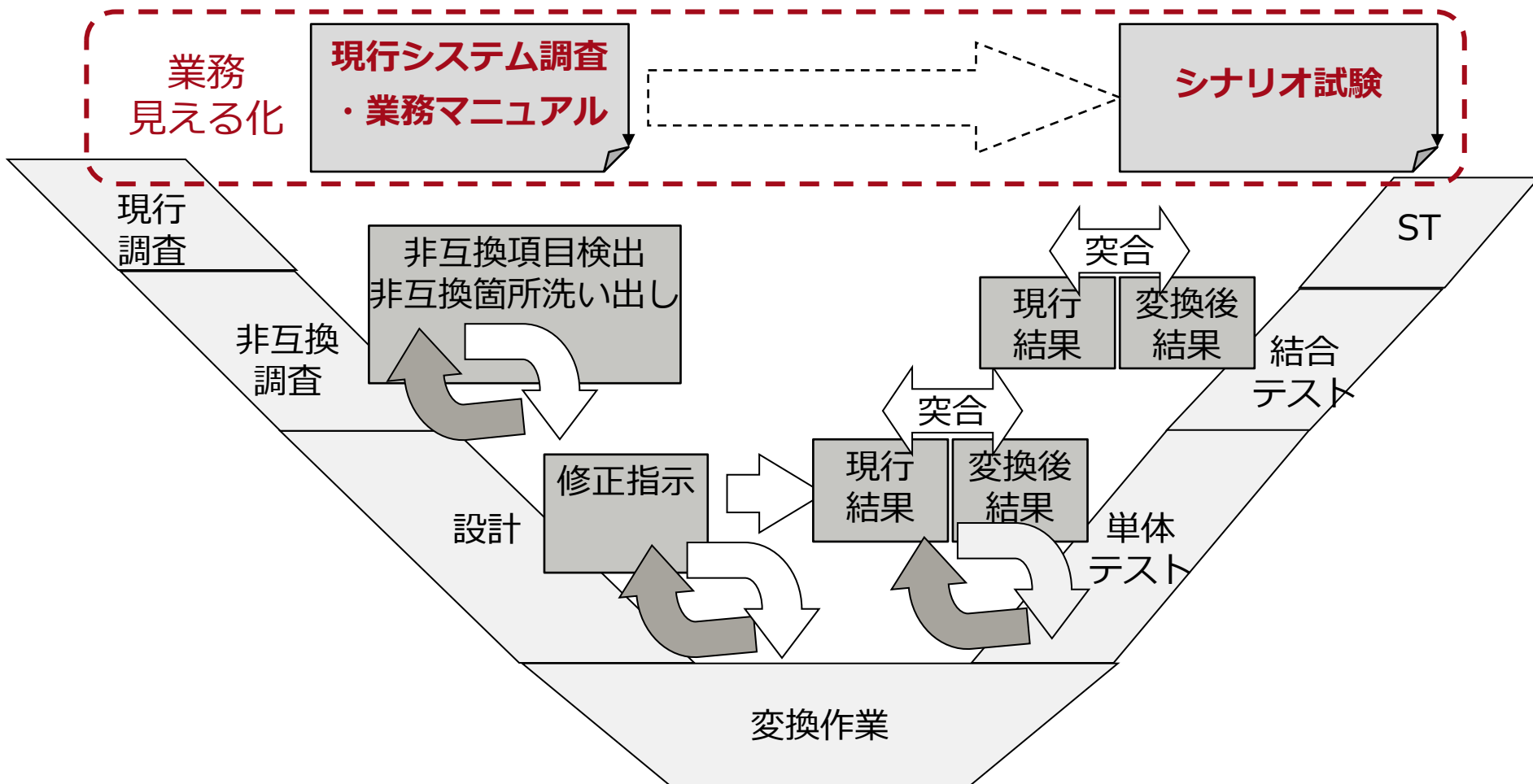
- 集中テストのポイントは非互換の網羅性。
- 資産分析（非互換項目マトリクス）により
資産を分類・グループ化。
- 資産グループ（非互換パターン）毎に
集中テスト対象資産を選定。



shaping tomorrow with you

品質担保の考え方

- 業務、アプリケーションの見える化、ドキュメントの整備より把握した現行業務のシナリオを策定し、業務としての品質を担保する
- 最終的には現行把握の正確さがテスト効率と品質を向上させる





モダナイ後の開発/保守 最新技術導入のススメ

最新技術の導入

オープンスタンダードな開発環境

■ NetCOBOL Studio (Windows版 NetCOBOLで提供)

- ・ Eclipseを採用した開発環境
言語共通の操作性と高い生産性でCOBOLアプリケーション開発
- ・ V11新機能でさらなる開発効率化 ※ Windows(64bit)

若い人も抵抗なく使えるよ。

Interstage Studio (別製品)

Eclipseを採用した統合開発環境

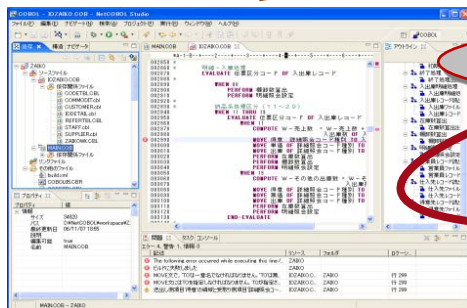
NetCOBOLをプラグインすることにより、Interstageを利用したJava、COBOLによるオンライン、バッチシステム開発が可能

※COBOL開発には別途NetCOBOLのインストールが必要

■ Visual Studio (.NET版 NetCOBOLで提供)

VB、C#などの言語共通の操作性と言語間連携で、.NETアプリケーションおよびMicrosoft AzureアプリケーションをCOBOLで開発

※ Microsoft Azure向け開発はソフトウェアサービス (NetCOBOL COBOLアプリケーション実行基盤) のアドオンが必要



プログラミング

ソース編集
画面・帳票作成

ビルド

翻訳、リンク

テスト

デバッグ、
各種チェック機能

保守・管理

ドキュメント生成
テストデータ作成など

※製品、対応OSによって利用できる機能が異なります。

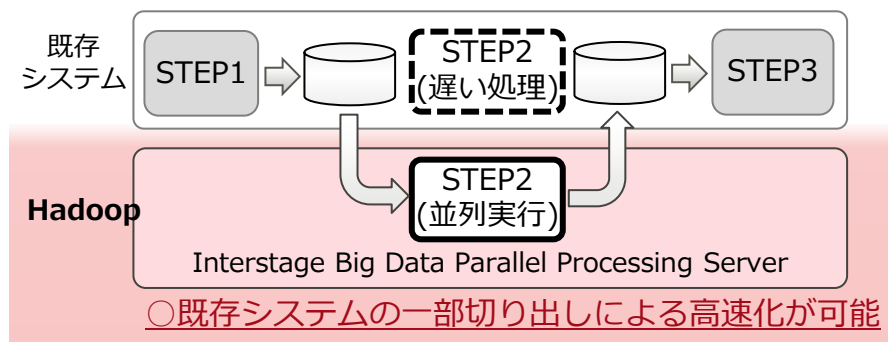
Hadoopと連携しCOBOLバッチアプリの処理時間を大幅短縮

- Apache Hadoop*¹との連携による並列分散処理をサポート



既存COBOL資産がそのまま
Hadoop上で分散実行により高速化

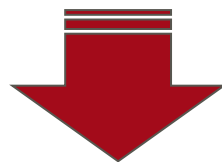
- Interstage Big Data Parallel Processing Server*²により、
既存システムとのシームレスな連携、さらなるバッチ処理の高速化が可能



* 1 : Apache Software Foundation(ASF)が開発したビッグデータの効率的な分散・並列処理を行うオープンソースソフトウェアです。

* 2 : Apache Hadoopベースの当社並列分散処理製品です。

- COBOLのロジックは継続性に優れている
(周辺の機能を整えることで、ロジックは容易に活用できる)
- 長年使っているだけに、とにかく資産の管理は徹底的に
(これをサボって、良い結果が得られた例はありません)
- 新しい技術もしっかり対応
(新しい技術との融合で、業務の息を吹き返せ)



ALL富士通でお客様のCOBOL資産を
全力でサポートします。

フォローアップセミナー参加のススメ

- レガシーシステムのモダナイゼーションは簡単ではなく、お客様に固有の問題や、根気を要するドロドロとした作業も少なくありません。
- 本日は時間の都合から駆け足の説明になってしまい、また、ご質問をお受けする時間をご用意できませんでした。
そこで、本日ご紹介した事例を含めて、もう少し詳しくご説明するためのフォローアップセミナーを企画いたしました。
奮ってご参加ください。

開催日: 2015年6月中旬

場所: 富士通 トラステッド・クラウド・スクエア

参加費: 無料

お申込み: 下記URLよりご案内いたします。

<http://software.fujitsu.com/jp/middleware/events/>



お問い合わせ先（TransMigrationサービス）

既存アプリケーション資産をお客様のニーズと最新の技術動向を踏まえて最適なプラットフォームへ移行し、既存システムのモダナイゼーションを支援します。

サービスの詳細は以下サイトをご覧ください。

TransMigrationサービスホームページ：

<http://jp.fujitsu.com/solutions/sdas/migration/>

ホーム > ソリューション & サービス > 富士通の総合システム開発体系「SDAS」 > TransMigrationサービス

富士通の総合システム開発体系「SDAS」

▼ TransMigrationサービス

- モダナイゼーションの考え方
- POCサービス
- 資産マイグレーションサービス

TransMigration(トランスマイグレーション)サービス

トランスマイグレーション

TransMigrationサービス

既存資産を最大限に活用
継続的なシステム最適化を実現します

TransMigrationサービスは、既存アプリケーション資産をお客様のニーズと最新の技術動向を踏まえて最適なプラットフォームへ移行し、既存システムのモダナイゼーションを支援します。

- 当社の技術・ノウハウを結集し、最適なプラットフォームへの移行を支援します。
- 短期間で高品質なマイグレーションと基幹業務に耐える高い運用性を実現します。
- 多彩なマイグレーションパターンによりお客様のシステム最適化ニーズにお応えします。

[モダナイゼーションの考え方](#) >>

サービスメニュー：診断フェーズ（次期システム方針をご検討中のお客様向け）

POC(Proof of Concept)サービス

プロジェクト開始前に、お客様の既存アプリケーション資産を使用して、移行方式の妥当性検証や課題の洗い出しを行います。

サービスメニュー：構築フェーズ

資産マイグレーションサービス

既存アプリケーション資産を有効活用して、業務仕様を変えずに、異なるプラットフォームに適合するようにアプリケーションを交換します。

関連リンク

お問い合わせ

- 本商品に関するお問い合わせ

導入をご検討中のお客様

- 導入事例

関連製品・サービス

- INTARFRM
- NetCOBOL
- Interstage Job Workload Server

NetCOBOLの最新情報をご提供しています

NetCOBOLホームページ : <http://software.fujitsu.com/jp/cobol/>

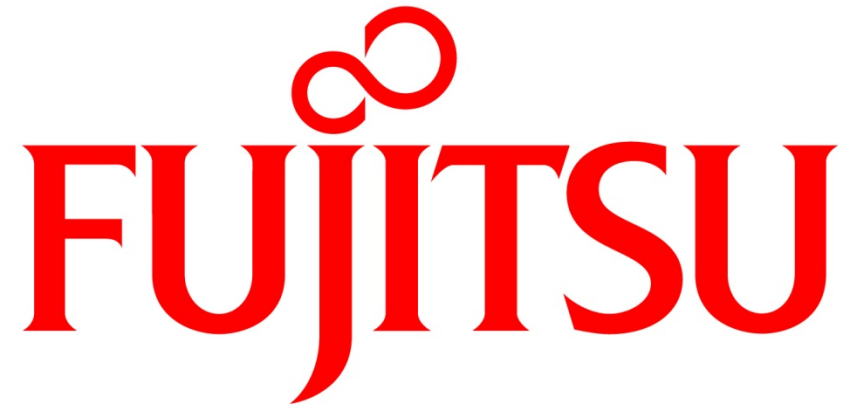


The screenshot shows the NetCOBOL homepage. At the top, there's a navigation bar with the Fujitsu logo, a language selector (Japan), a region selector (国・地域を変更), and a search bar (富士通サイト内検索). Below this is a breadcrumb trail: ホーム > ソフトウェア > NetCOBOL. The main content area features a large banner for 'FUJITSU Software NetCOBOL オープンプラットフォーム COBOL 開発環境'. To the left is a sidebar with a 'ソフトウェア' menu containing links to 'NetCOBOL', 'コンセプト', '製品ラインナップ', '特長', '機能', '動作環境', '価格', 'カタログ・資料', '関連製品', '今までに発表した製品', 'サービス&サポート', '購入方法', '導入事例', '技術情報', and 'トレーニング'. The main text describes NetCOBOL as an open platform COBOL development environment for the latest technology and environment, supporting business operations. On the right, there's a 'お問い合わせはこちら' (Contact Us Here) section with a '入力フォーム' (Input Form) link and contact information: 0120-933-200, available 9:00-17:30 (excluding holidays and company-designated rest days). Below this is a 'メールマガジン' (Email Magazine) section for 'FUJITSU's Middleware News' and a '連載' (Serialized) section for 'Interstage' featuring 'ビッグデータ活用を支える 富士通のソフトウェア' (Software supporting big data utilization by Fujitsu).

- コンセプト
- 製品情報
- FAQ（ご購入前、ご購入後）
- オンラインマニュアル
- 体験版貸し出し
- 導入事例
- 技術資料

富士通コンタクトライン : 0120-933-200

- Microsoft、Windows、Windows Server、Windows 8、Windows 7、Windows Vista、Microsoft AzureおよびSQL Azureは米国Microsoft Corporationの米国およびその他の国における登録商標です。
- Red Hat は米国およびその他の国でRed Hat, Inc. の登録商標若しくは商標です。Linuxは Linus Torvalds氏の商標です。
- Oracle とJavaは、Oracle Corporationおよびその子会社、関連会社の米国およびその他の国における登録商標です。
- Apache、Apache Hadoop は、The Apache Software Foundation の登録商標または商標です。
- NetCOBOL 、Interstage、Systemwalkerは富士通株式会社の登録商標です。
- 記載されている会社名、製品名等は各社の商標または登録商標である場合があります。
- 本資料では、Oracle SolarisはSolarisと表記しています。
- 本資料に記載されているシステム名、製品名などには必ずしも商標表示（TM・®）を付記しておりません。



shaping tomorrow with you