製品であろうとサービスであろうと開発期間の短縮は全てのビジネスの基本…プログラム開発も然り

モデリング化技術によるシステム開発期間短縮に関する考察

プログラム資産の可視化はモデリング化技術によるモデルベース開発で実現

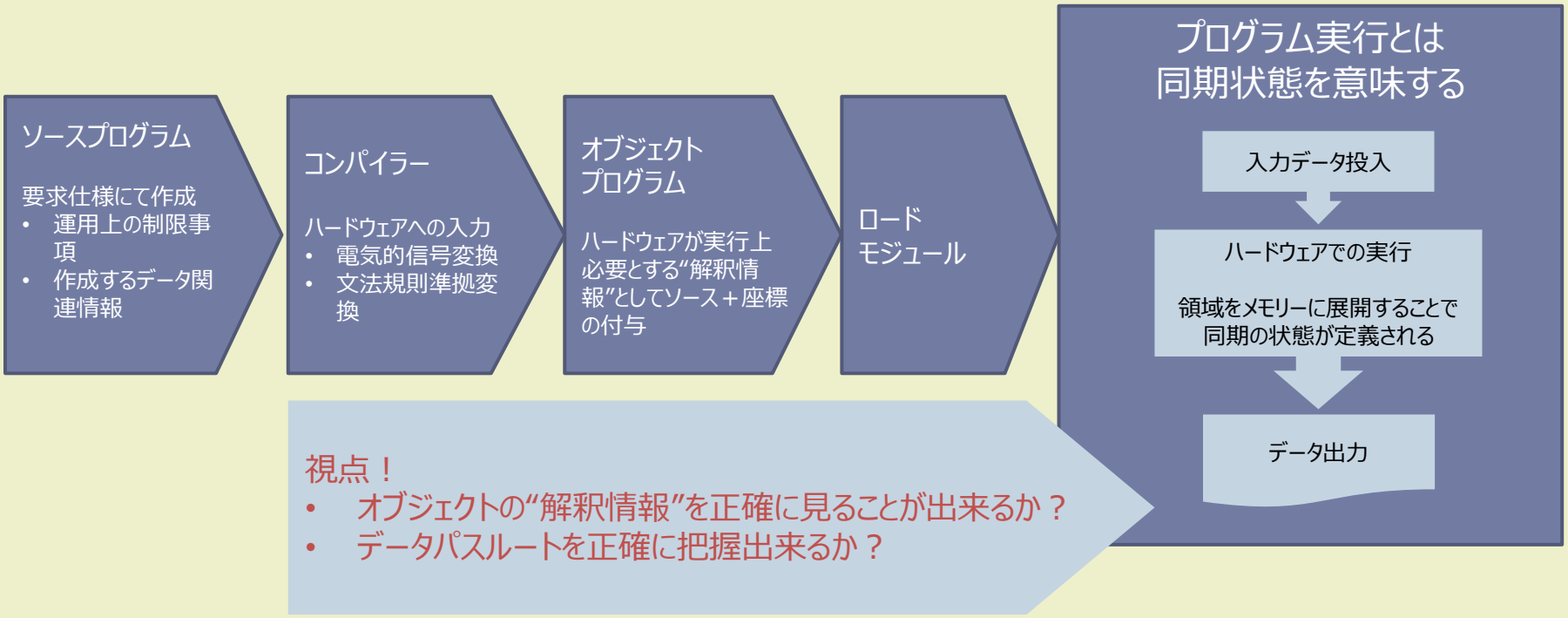
エムティインターナショナル株式会社
http://mtiinc.jimdo.com/

モデリング化技術でシステム開発期間短縮が可能に･･･

製品であろうとサービスであろうと開発期間の短縮は全てのビジネスの基本
⇓
プログラム開発も同じ

ソウフトウェア開発期間短縮のための重要3要素

その1．ソフトウェア開発技術者はソースプログラムがコンピュータ上で稼動するまでを理解しているか？



モデリング化技術でシステム開発期間短縮が可能に･･･

ソウトウェア開発期間短縮のための重要3要素

その2．“コンパイラーは１構文１機能へ分割”という言語文法ルールに則っているか？

プログラムは全て言語文法の
ルールに基づき作成されている。

構文成立のルールとは ⇒ １構文１機能の前提

1. 予約語列
2. 演算子
3. 領域名
4. 定値名
5. ラベル名
6. 終了記号

プログラム言語構文規則

コンパイラー依存部品

1. 制御文：実行順序を決める叙述
2. 呼出文：部分プログラムを利用する為の叙述
3. 翻訳文：コンパイラーが処理する叙述

コンパイラー非依存部品：アプリケーション表現部品

1. 領域文：領域を定義する叙述
2. 関数文（代入文･定値文）：四則演算、論理演算の叙述
3. 入力文：外部情報を取得する為の叙述
4. 出力文：外部に情報を送出する為の叙述
5. 条件文：真偽値を生成する叙述
6. 注釈文：備忘を補完する叙述（コンパイラー影響なし）

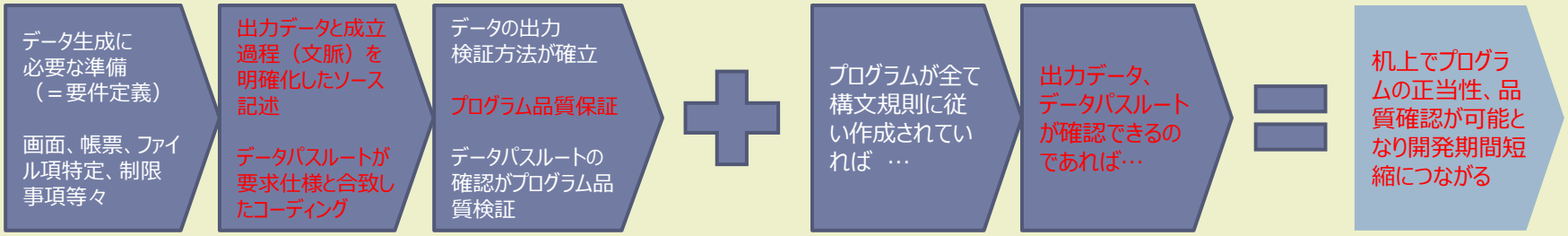
ソウトウェア開発期間短縮のための重要3要素

その3．プログラムのハードウェア実行での稼動は同期状態であることを理解できているか？

ソースプログラムの記述とハードウェア実行状態が同一様相である確認：プログラムの品質検証

プログラム作成目的とは⇒データの生成

プログラムの実行状態（同期）を再現

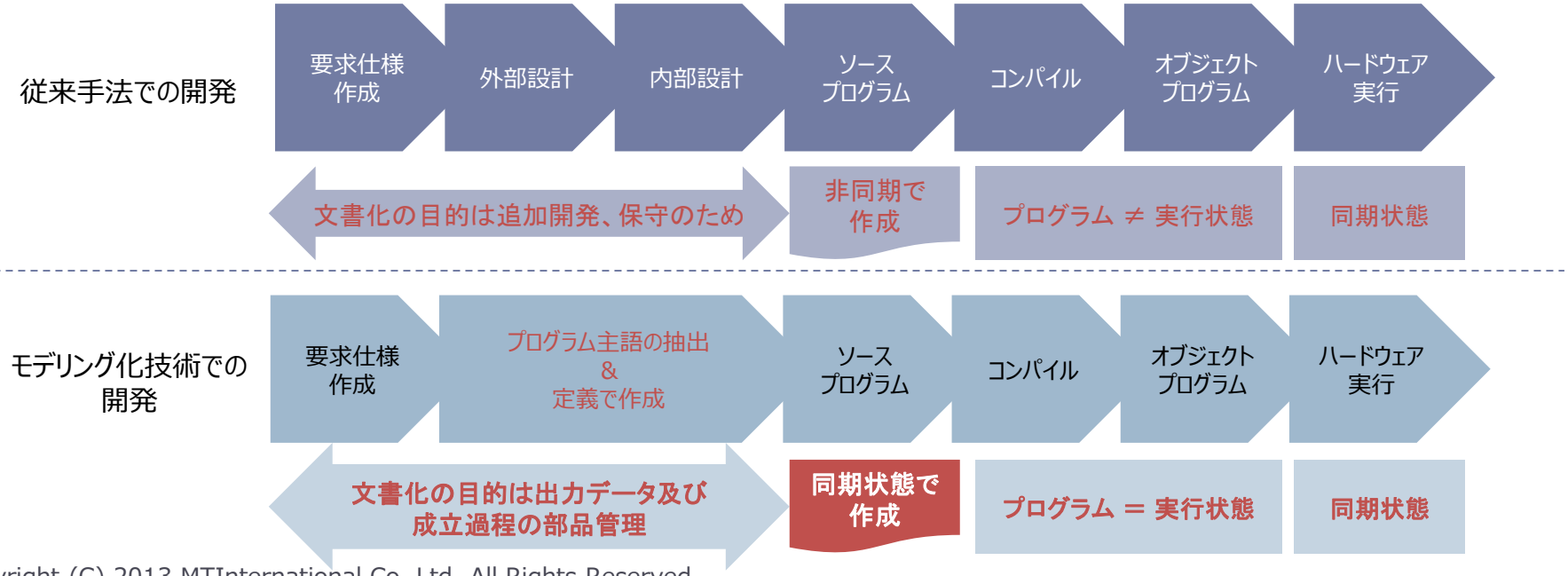


モデリング化技術でシステム開発期間短縮が可能に…

課題	従来手法では開発済みソースプログラムの正当性はハードウェアの実行結果のみでしか判断が出来ない
解決策	ハードウェアのプログラム実行状態を把握できる開発手法選択が必須 モデリング化技術による開発手法で同期状態で稼動している実態を開発済みソースプログラムで検証できる。（同期プログラミング）

課題	従来手法では開発経験が知識として各作業工程に再利用可能な状態でデータベース化されてない
解決策	全ての作業工程が数量的把握が出来るルールで実施されていること モデリング化技術による開発手法で知識の再利用を『文書化主体』から出力データ（主語）とデータ成立過程（文脈）の部品化＆データパスルート管理できる。

課題	従来手法の開発ではソースプログラムの作成完了時点での検証手段が無い。（本来はコンパイル前に検証が必須）
解決策	開発期間短縮の出来る開発手法は品質保証を出来る検証手段があること モデリング化技術による開発手法でプログラム作成は『主語と文脈部品抽出』か『追加定義作成』の上データパスルート（主語系譜）作成で第三者検証が可能になる。



システム開発のトレンドはビジネススピードへの迅速な対応

システムは作る時代から活用する時代
その時代背景を踏まえシステムの最重要課題とは…
⇓
ビジネススピード、社会制度変更へのタイムリーな改修対応のために
システム開発手法の見直しが重要要素となる

従来型開発手法	モデリング化技術での開発手法
ＳＥ技量による経験主体開発 • 開発期間短縮が難しい 開発期間＝作業工数　Ｘ　品質 • 作業工数は経験値をベースに判断 • 品質はテスト実施内容（ボリューム次第）	モデリング化技術でのモデルベース開発 • 開発期間短縮が容易 開発期間＝作業工数　Ｘ　１ 作業工数はモデル定義量で決定 • 品質は定義方法が標準でバラツキがない
作業工数定量化が難しい • 使用する言語による開発標準化基準作成に難 • ＳＥの経験年数，知識量の定量表現に難 • 再利用方法論の確立が難しい	作業工数定量化が容易 • モデリング化技術は全て公理空間定義 言語種別関係なく全て同一基準 • モデル定義とはデータとデータの文脈定義 定義する時間はほぼ一定 • 再利用は定義内容の同一化で活用 モデル定義はリポジトリー管理
品質確認の可視化が難しい • 品質担保はコンピュータ上での入出力データの値の整合性で確認 • 品質検証はコンピュータ上での稼動結果 • プログラム毎にテストケース作成 全てのテストケースは網羅できない	品質確認の可視化は確保 • 品質担保はソースプログラムのデータパスルートが要求仕様を満たしている事を机上確認 • 品質検証はソースプログラムで実施 定義のデータ＆データ文脈は知識ＤＢ • プログラム毎に全データパスルート作成 データパスルートの机上検査可能

モデリング化技術でシステム開発の可視化を実現

可視化とは公理の世界で
証明されなければならない。

モデリング化技術はＬＹＥＥ(株）根来文生氏の
最新版ＬＹＥＥ論文の実装技術

プログラム開発の現状

✓ テストの繰り返し（過去の経験）にて品質確保

プログラム記述
の文法チェック

命令順序列で
オブジェクト化

データ投入で要
件チェック

入力と出力の
整合性

使用マシン上でチェック
ワークロード大

要求仕様
作成

外部設計

内部設計

プログラミング

コンパイル

単体テスト

統合テスト

システム
テスト

モデリング化技術活用
公理空間で可視化

✓ 大幅なコスト削減
✓ 開発済プログラム品質担保
✓ 要求仕様の明確化

➢ 主語とは領域宣言のデータ項目に
代入があり実体化されたデータ項目
➢ 脈絡図とは主語の成立過程の様相

開発済み
プログラム

可視化（公理）コンパイラー：コンパイラー＆マニュアル文法準拠で作成

公理空間で
可視化

プログラム構文(ステップ）を1構文1機能分割があり実体化されたデータ項目
命令＆データの順序列化

脈絡図の
作成

プログラムの実行様相
プログラム記述の妥当性チェック：文脈作成で確認
命令＆データの実行様相の全経路確認

主語系譜の
作成

要求仕様の可視化
プログラムと要件の妥当性チェック：データ遷移
作成するデータ内容，役割，構文使用箇所特定

システム調査技術の比較　現状 vs モデリング化技術

現状のシステム調査方法

ソースコード資産 調査分析	● 対象資産の明確化
プログラムパラメータ 主体分析	● 予約語ベースに仕様調査
対象資産文書確認	● 更新された要求仕様書の存在？
調査要望に対する 回答書作成	✓ 現状資産の可視化は可能か？

VS

モデリング化技術での調査方法

ソースコード資産 調査分析	● 対象資産の明確化
プログラムを１構文１機能 に分割・分析	● プログラムステップは公理空間で分析 ● 分割構文＝単元文に座標付与
プログラム正統性チェック	● 非実行箇所抽出：終了文探索
脈絡図作成 プログラムの実行様相 把握	● 命令＆データの実行様相表現（領域名 宣言データにデータ代入で主語化） ● 主語はデータ成立文脈を持つ ● 実行様相全経路抽出：テスト範囲特定
主語系譜作成 要求仕様の可視化	✓ 可視化を公理の世界で証明 ● 主語系譜でプログラム品質担保 ● 要求仕様書通りのプログラム作成を検証

現状のプログラム開発手法

要求仕様作成	
外部設計書作成	
内部設計書作成	
プログラミング	
コンパイル	
単体テスト	
	✓ テスト回数で品質確保 ✓ 品質担保に科学的根拠は？

VS

モデリング化技術での開発手法

要求仕様作成	● 画面、帳票、ファイル
プログラミング	● 公理空間で作成 ● 予約語は使用言語文法準拠 ● ウィルス排他生成
脈絡図・主語系譜作成 検証作業	✓ 机上でプログラム品質公理検証 ✓ 期間短縮、コスト削減
コンパイル	
プログラムテスト	✓ 実行性能確認 ✓ 操作性確認

私たちエムティインターナショナルは、
西暦２０００年問題の頃からの長いマイグレーション経験のもとにプログラムを第三者視点で品質検証の出来る「モデリング化技術」を完成させました。
最近のシステム開発トレンドは「内製化・コストの大幅な削減・プラットフォームフリー」で、「レガシー資産で稼動しているプログラム」をいかにして最新技術に融合するかが重要課題になっております。
モデリング化技術でプログラム断・捨・離を実施、ビジネスにタイムリーに応えられるシステム構築を支援させて頂きます。

プログラム資産の可視化はモデリング化技術によるモデルベース開発で実現
エムティインターナショナル株式会社
http://mtiinc.jimdo.com/