

オントロジーとモデル理論に基づく ソフトウェア開発の品質向上の実践

上流工程支援と下流工程技術の両立を目指して

Noriaki Izumi Ph.D.

National Institute of Advanced Industrial Science and Technology (**AIST**) Japan

Email: n.izumi@aist.go.jp

URL: <http://staff.aist.go.jp/n.izumi/>

なぜ、システム開発プロジェクトは毎回失敗するのか？

➡ システム開発の現実

- ☑ 「失敗から学ぶ」はウソ！ ← 統計調査の結果！?
 - ⊙ 失敗した人が次に成功する確率は、初めての人が成功する確率と同じ！

➡ 間違った人材育成方針と技術ロードマップ

- ☑ 今、求人のあるJava/Web技術者は、20年前のCOBOL技術者！?
 - ⊙ アルファベットのスープとパッケージ神話による人材流出・ノウハウ消失
- ☑ マネジメント重視・技術者軽視の弊害
 - ⊙ 不具合の実態と間違った発注者・利用者へのアプローチ
 - ⊙ 「鬼の婦長さん問題」と「まじめな皿洗い現象」

➡ 技術者が主体的に成長できる方法論と環境整備とは？

- ☑ 組織・人材・スキルの改善
 - ⊙ 枯渇するエース人材と少子高齢化によるダイバーシティ推進の重要性
- ☑ ソフトウェア技術者の知情意：利用者（非専門家）の観点
 - ⊙ 可読性の高い図式と日本語文書
 - ▶ 利用者に必要な文書を整備するだけで、調達から開発、テスト、検収、運用までを可能に！！
 - ⊙ 非専門家が、コピペ・穴埋めで記述・作成できるようになる。
 - ▶ ワインバーグの指摘：辞書と文法書で、外国語文学がスラスラ執筆できる、という幻想。

➡ ソフトウェア品質を統制するためのAIST包括フレームワーク

- ☑ 下流工程技術を活用するための上流工程成果物の品質管理方法
 - ⊙ 先人の知見・経験を活用するためのオントロジーとモデル理論

自己紹介：はじめに、をかねて。

Noriaki Izumi Ph.D.

National Institute of Advanced Industrial Science and Technology (**AIST**) Japan

Email: n.izumi@aist.go.jp

URL: <http://staff.aist.go.jp/n.izumi/>

自己紹介を兼ねて

70年代に指摘され、80年に取り上げられた
今も昔も、変わらない本質!?

和泉憲明

産総研

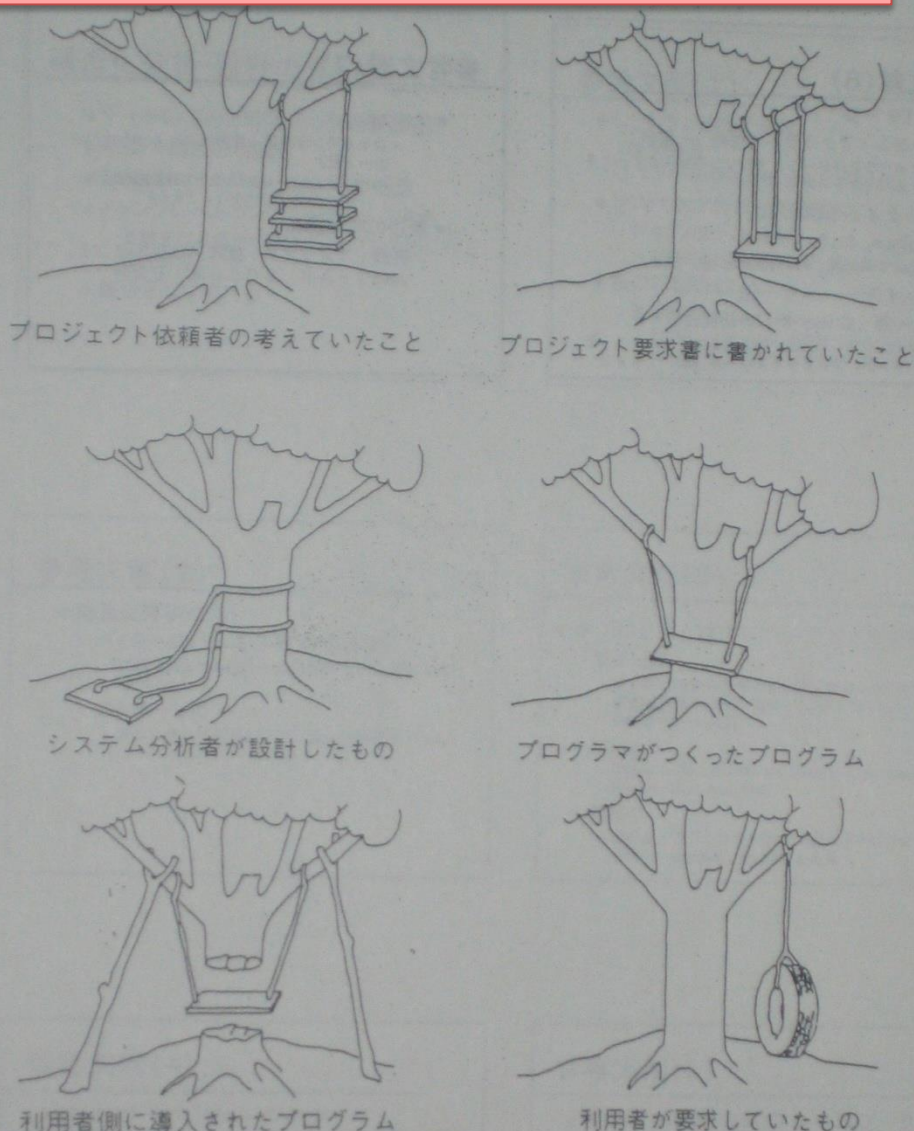
国立研究開発法人・
産業技術総合研究所
(旧・電総研)

上級主任研究員

成功するプログラ ミング

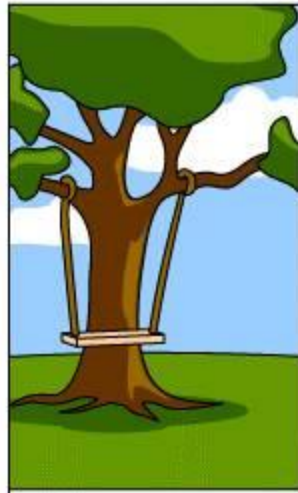
ブライアン L. ミー
ク, **永田 守男**ほか

近代科学社 ; ISBN:
4764900777 ;
(1982/01)





How the customer explained it



How the Project Leader understood it



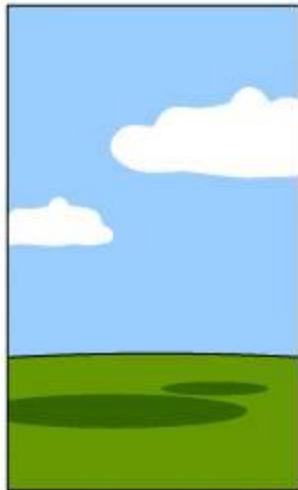
How the Analyst designed it



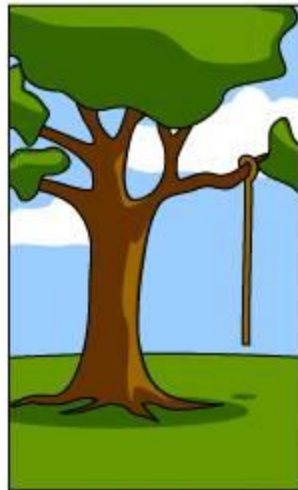
How the Programmer wrote it



How the Business Consultant described it



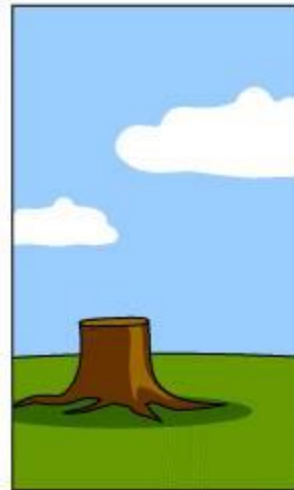
How the project was documented



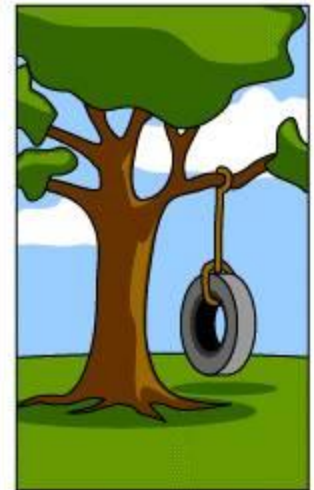
What operations installed



How the customer was billed



How it was supported



What the customer really needed

<http://weblog.cemper.com/a/200309/09-typical-project-life.php>

永田守男先生の教え：古くて新しい問題を解決する。

第3章 プログラム開発

3.2 検査とデバッグ

3.3 実行時の性能改善

第4章 特別な問題

4.2 大量のデータ

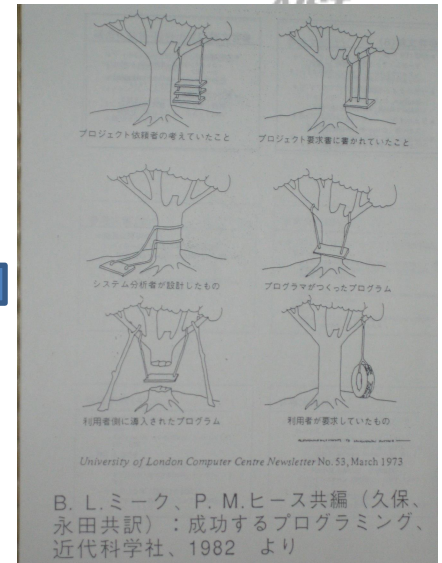
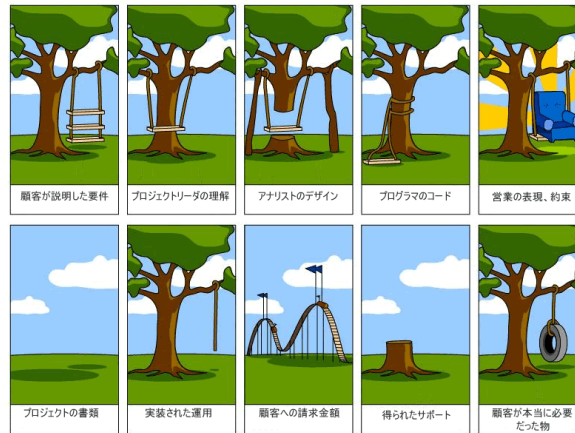
4.3 大規模プログラム

4.4 実行時間の長いプログラム

第5章 他の人々との関係

5.1 プログラムの文書作成

5.3 他人のプログラム



技術的な課題は、70年代に指摘されている？

- ❑ 「『人の活動としてのプログラミング』… プログラミングは単なる**ハードウェアの問題、ソフトウェアの問題ではない**のだ。」、ワインバーグ@71
- ❑ 「ソフトウェア技法をどう改善したら良いかという中心的疑問は、…、**人間の問題**に集中する。」、ブルックス@75

品質向上のために克服すべき課題は、どんな『人間の問題』なのか？

- ❑ 自治体の職員や地方の中小企業にはできても、他の民間企業や他自治体には無理なのか…!?

開発の現場は、なぜ、技術を活用できないのか？

事件(品質インシデント)は会議室で解決できない!?

Noriaki Izumi Ph.D.

National Institute of Advanced Industrial Science and Technology (**AIST**) Japan

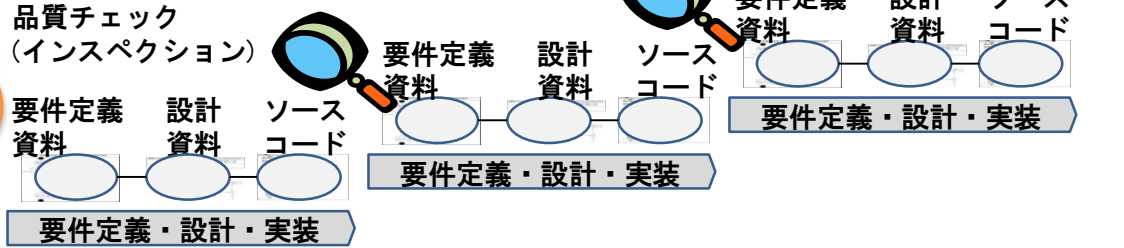
Email: n.izumi@aist.go.jp

URL: <http://staff.aist.go.jp/n.izumi/>

下流工程重視の技術改善

アジャイルソフトウェア開発は銀の弾か!?

- 多くの下流工程集約型の方法論が登場



要件定義・設計・開発・移行の適切なコスト配分は?

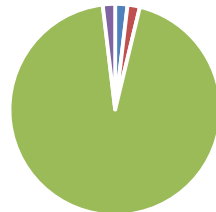
- 産総研のメトリックスでは、3 : 3 : 2 : 2
 - 成果物量と想定スキルから稼働時間を算出
- しかし、現実には、 $\Delta : \Delta : 10 : 0$ (移行は次年度?)

健全な開発PJ



■ 要件定義 ■ 設計
■ 開発 ■ 移行

現実の開発PJ



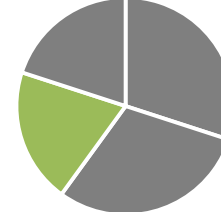
■ 要件定義 ■ 設計
■ 開発 ■ 移行

下流工程技術の発想



■ 要件定義 ■ 設計
■ 開発 ■ 移行

超開発の落とし穴



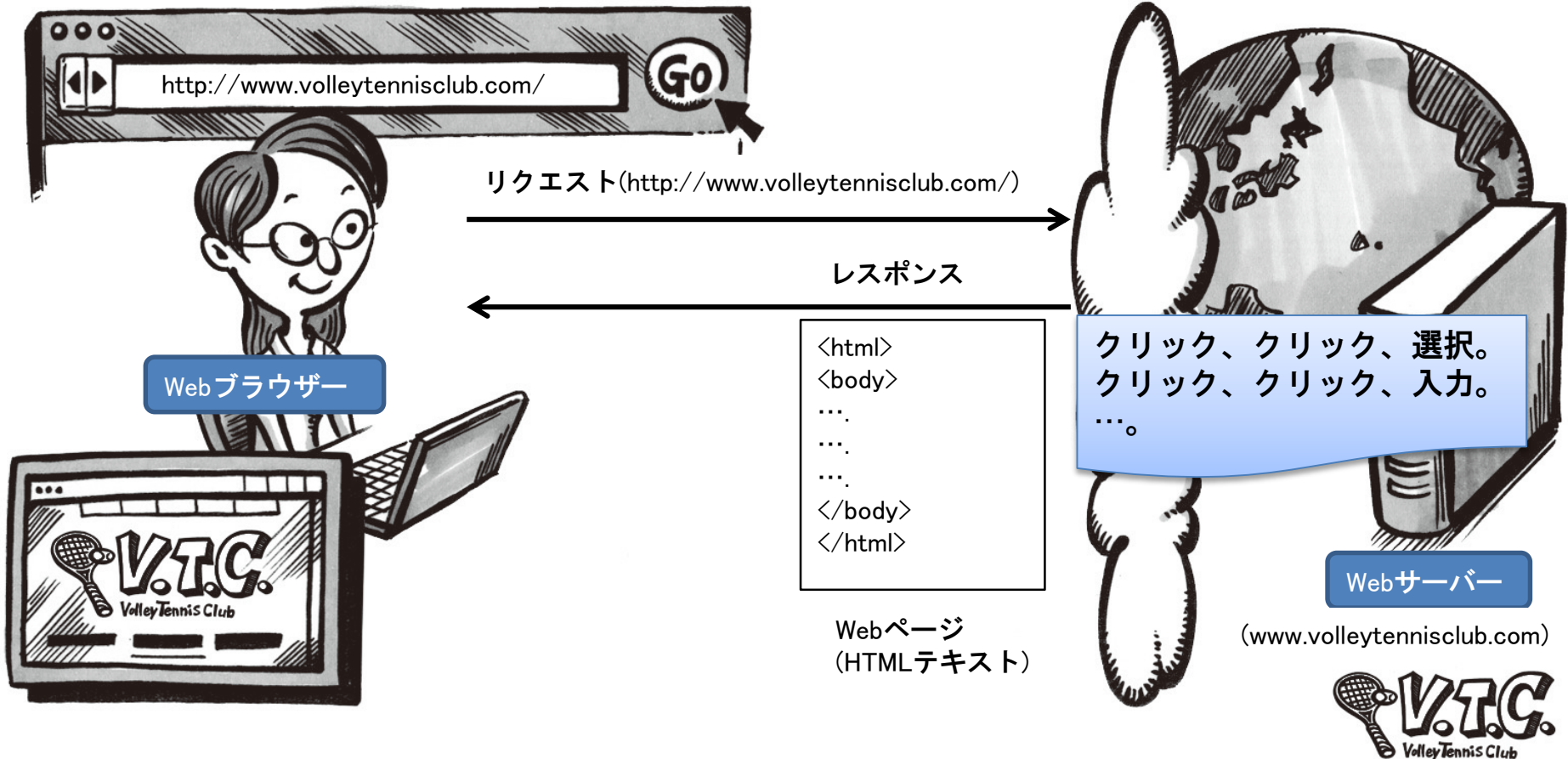
■ 要件定義 ■ 設計
■ 開発 ■ 移行

- 移行コスト (時間・費用) は、運用保守費にマージ?

画面中心開発とパッケージによる業務ノウハウの消失

画面のレイアウト(UI)ばかり
気にする傾向が…。

「画面を確認した方が早い」
⇒ 実際、画面で何を
インターネット (WWW)



例：高速開発手法によるシステムテスト成果物

- ➡ 業務知見が欠乏するとテストデータが貧弱になり…
- ➡ 最低限のデータがあると…

➡ しかし、実際は、…（実開発のインスペクション資料参照）

上流工程支援と下流工程技術は両立しないのか？

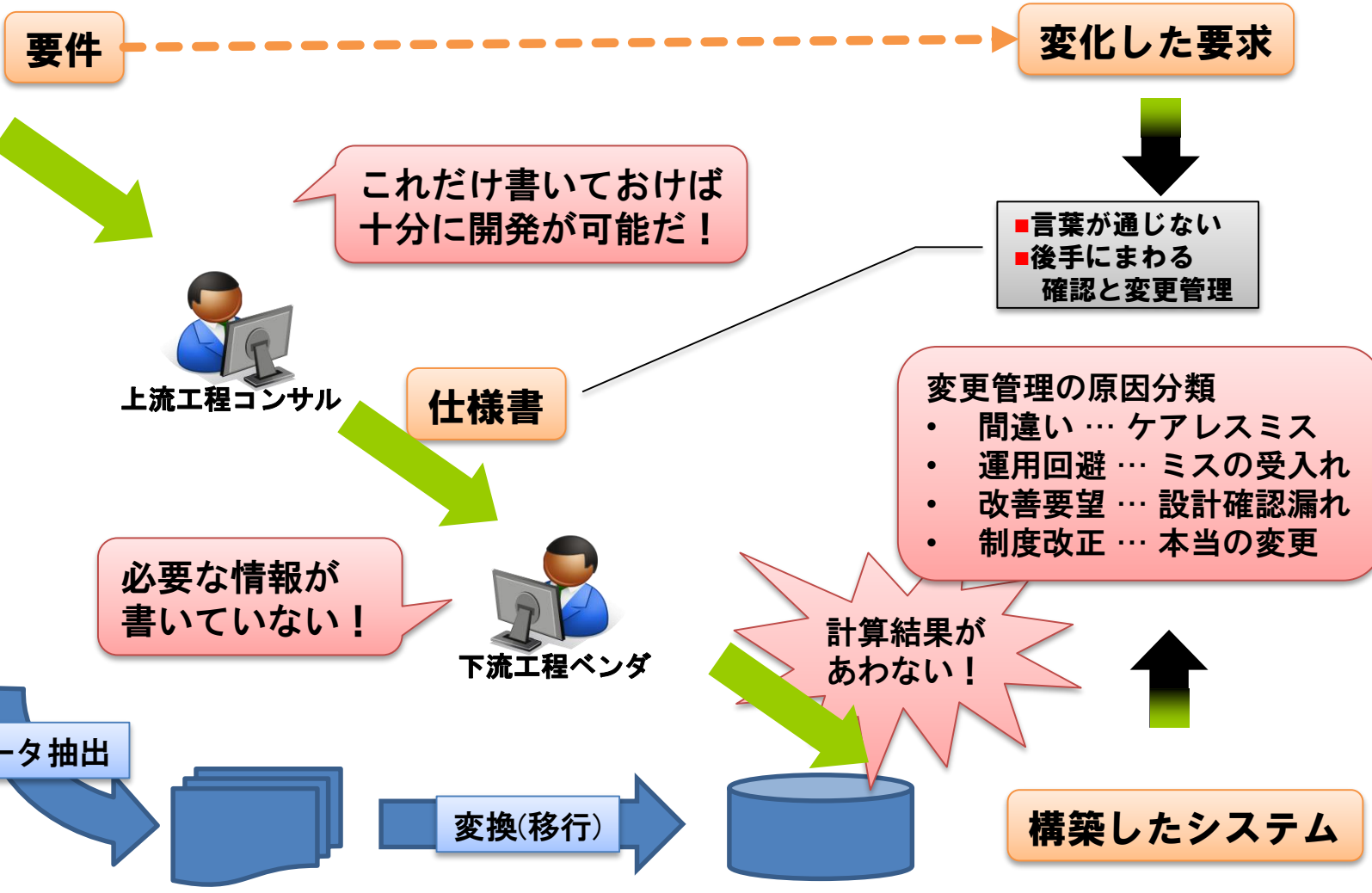
従来のシステム調達



業務



運用



下流工程手法の落とし穴：上流工程ミスの実態

例：児童虐待対象の電話番号検索

対象の電話番号検索

担当者の要求：
履歴から対象を
電話番号で検索

効率が悪い！

電話番号でのキー検索

技術者の理解：
電話番号を入力
対象を出力

画面設計しました！

出力に電話番号が必要

設計修正してモック作成完了!?

これからは、システムに
業務をあわせる時代です！

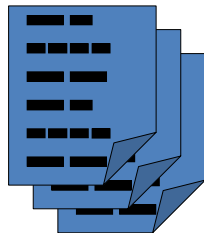
年齢や性別での絞り込み機能が必要

システムテストにて対応しました!!

乳児・幼児・男女での絞り込み機能が必要



思いが言えない
担当者



読み返せない文書

画面モックアップ

電話番号	<input type="text"/>
カナ氏名	<input type="text"/>
漢字氏名	<input type="text"/>
生年月日	<input type="text"/>
<input type="button" value="入力"/> <input type="button" value="クリア"/>	

使えないシステム



話を聞かない
SE



現場を知らない
責任者

協働のための文書: 業務・システムの最適化のために !!

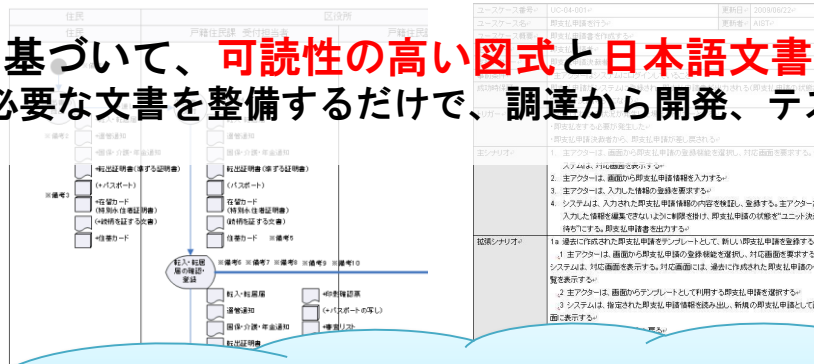
利用者と開発者の間でいかに効率的に知識を共有し作業を分担するか?

- ☑ 利用者同士の協働のための文書
 - ◎ 業務一覧・業務マニュアルなど: 利用者が作成
- ☑ 開発者同士の協働のための文書
 - ◎ 詳細設計書・データ定義書など: 開発者が作成
- ☑ **利用者**と**開発者**との協働のための文書
 - ◎ 業務フロー図・ユースケース記述など: **利用者主導**で作成 ← 発注に必要

効率がいいはず! → 伝わる内容

利用者の観点

- ☑ **モデル理論に基づいて、可読性の高い図式と日本語文書**
 - ◎ 利用者に必要な文書を整備するだけで、調達から開発、テスト、検収、運用までを可能に !!



職員可読性の高い
図式表現・日本語文書とは?

共有 + 共同編集

AIST包括フレームワークの仕組み

添削可能な上流工程の成果物とは？

Noriaki Izumi Ph.D.

National Institute of Advanced Industrial Science and Technology (**AIST**) Japan

Email: n.izumi@aist.go.jp

URL: <http://staff.aist.go.jp/n.izumi/>

AIST包括フレームワークの技術的特徴

『利用者』のための成果物の可読性

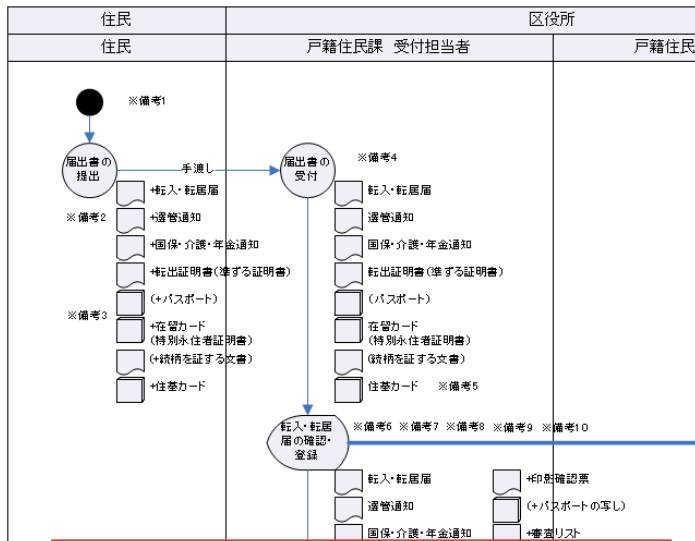
☑ 非専門の職員でも添削可能な**業務フロー図**と**ユースケース記述**

業務フロー図

決裁（調定・決済）のフローは、**特に、帳票を具体的に！**
 （一般には、〇〇情報などと丸めて標記する傾向が…。）

ユースケース記述

利用者による操作手番は、**可能な範囲で、抽象的に！**
 （一般には、画面項目などを過度に詳細標記する傾向が…）



抽出
(詳細化)

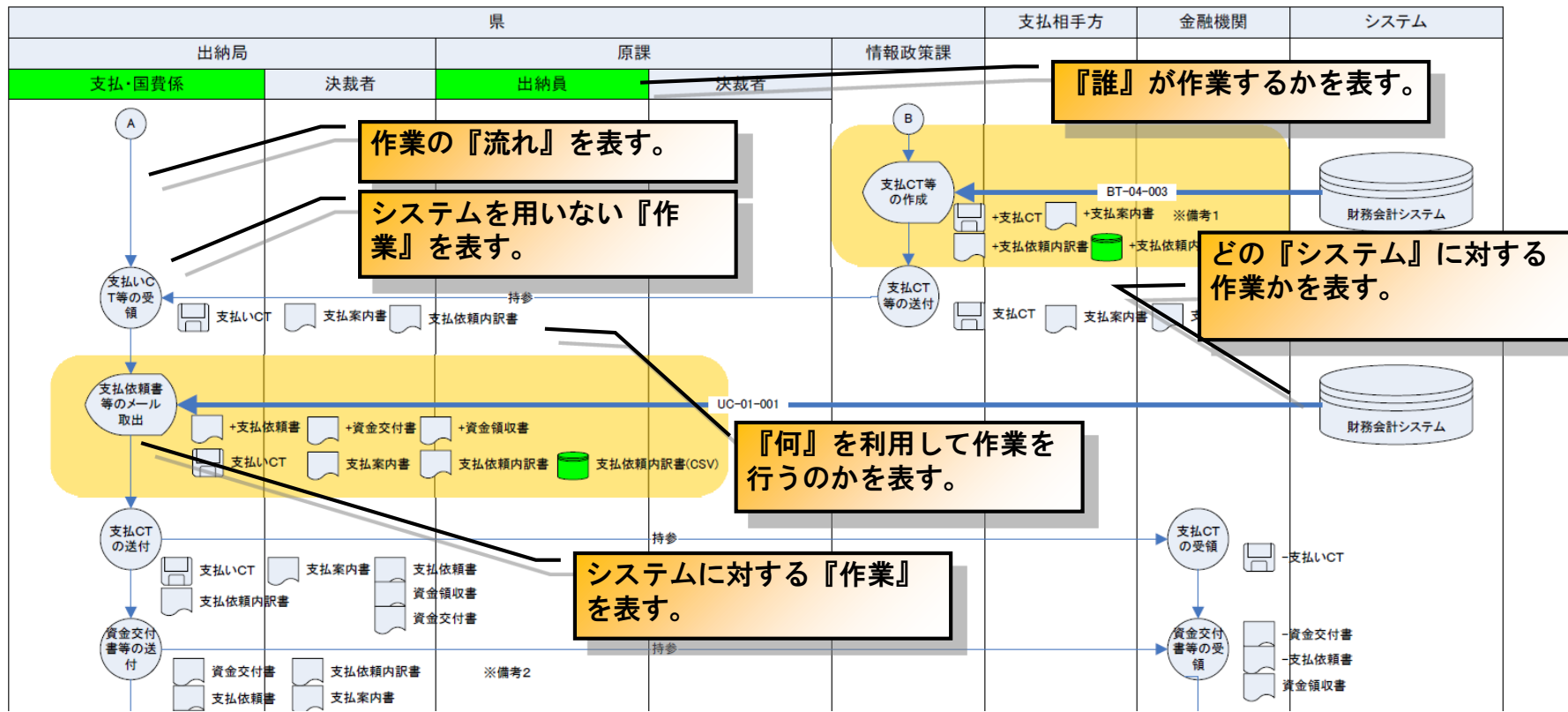
ユースケース番号	UC-04-001	更新日	2009/06/22
ユースケース名	即支払申請を行う	更新者	AIST
ユースケース概要	即支払申請書を作成する		
主アクター	即支払申請者		
支援アクター	即支払申請決裁者		
事前条件	・主アクターはシステムにログインしていること		
成功時保証	即支払申請がシステムに登録され、即支払申請書が出力される(即支払申請の状態で“ユニット決裁待ち”になる)		
トリガー	次のいずれかの状況が発生した場合をトリガーとする ・即支払をする必要が発生した ・即支払申請決裁者から、即支払申請が差し戻される		
主シナリオ	1. 主アクターは、画面から即支払申請の登録機能を選択し、対応画面を要求する。システムは、対応画面を表示する。 2. 主アクターは、画面から即支払申請情報を入力する。 3. 主アクターは、入力した情報の登録を要求する。 4. システムは、入力された即支払申請情報の内容を検証し、登録する。主アクターが入力した情報を編集できないように制限を掛け、即支払申請の状態を“ユニット決裁待ち”にする。即支払申請書を出力する。		
拡張シナリオ	1a 過去に作成された即支払申請をテンプレートとして、新しい即支払申請を登録する。 ① 主アクターは、画面から即支払申請の登録機能を選択し、対応画面を要求する。システムは、対応画面を表示する。対応画面には、過去に作成された即支払申請の一覧を表示する。 ② 主アクターは、画面からテンプレートとして利用する即支払申請を選択する。 ③ システムは、指定された即支払申請情報を読み出し、新規の即支払申請として画面に表示する。		

業務のコンテキスト

機能のコンテキスト

業務フロー図の特徴：利用者主導開発のための文書

- 『エンドユーザ（システム開発の非専門家）による可読性（添削と共有）』により『最新の開発方法論（Java, Web, UML, アジャイル）』へのトレーサビリティを確立！
- ◆ 実際の業務を正確に捉え、業務に依存する課題や効率化の範囲を正しく把握できる。
 - ◆ 利用者と受注者の共通の会話を成立 ⇒ 利用者が添削可能な成果物のひとつ。
 - ◆ 添付書類などを具体化した稟議決裁フローにより、帳票の迷子・みなし子を見つけ出し、システム化範囲の漏れ抜けや、実現機能の不足を洗い出すことが可能。



システムユースケース記述の意義

画面遷移（プログラムの仕様）相当の部分を日本語で記述

☑ 職員は、業務内容（日本語による記述）に集中できる。

◎ 職員は画面遷移レベルで仕様を記述・確認。さらに、システムの受け入れ・検収にも！

◎ 改修は、ユースケース記述を「赤入れ」するだけ。

日本語記述の効果

☑ 職員による業務の改善や共通化の検討が可能に。

☑ Cf: 画面による業務共通化は、「違いへのこだわり」が強くなるだけ！（障害要因に!?)

ユースケース数と開発規模の相関

☑ ユースケース単位で開発や改修の規模を概算で算出可能になり、開発ベンダと対等な立場に！

ユースケース番号	UC-01-001	更新日	
ユースケース名	メールを取り出す	更新者	
ユースケース概要	個人宛メールBOX および所属共通メールBOXに格納されている帳票等を照会する。必要に応じて取り出し、帳票として出力する。		
主アクター	原課担当者		
支援アクター	(特になし)		
事前条件	・ 主アクターはシステムにログインしている。		
成功時保証	メールBOXに格納されている帳票類が出力される。		
トリガー	メールBOXに格納されている帳票類を取り出す必要が生じた。		
主シナリオ	<ol style="list-style-type: none"> 1. 主アクターは、メール取り出しの対応画面を要求する。 2. システムは、対応画面を表示し、個人宛メールBOXおよび所属共通メールBOXに格納されている帳票類を一覧表示する。 3. 主アクターは、メールBOXから取り出したい帳票類を選択し、取り出しを要求する。 4. システムは、指定された帳票類を取り出し、内容を画面に表示する。 5. 主アクターは、表示された帳票類の出力を要求する。 6. システムは、要求された帳票類を出力し、処理を終了する。 		
拡張シナリオ			
関連ビジネスルール	・ BR-01-001:メール取り出しに関するルール		
関連資料			
備考			

上流工程成果物を正確に読み解く

添削可能な上流工程の成果物とは？

Noriaki Izumi Ph.D.

National Institute of Advanced Industrial Science and Technology (**AIST**) Japan

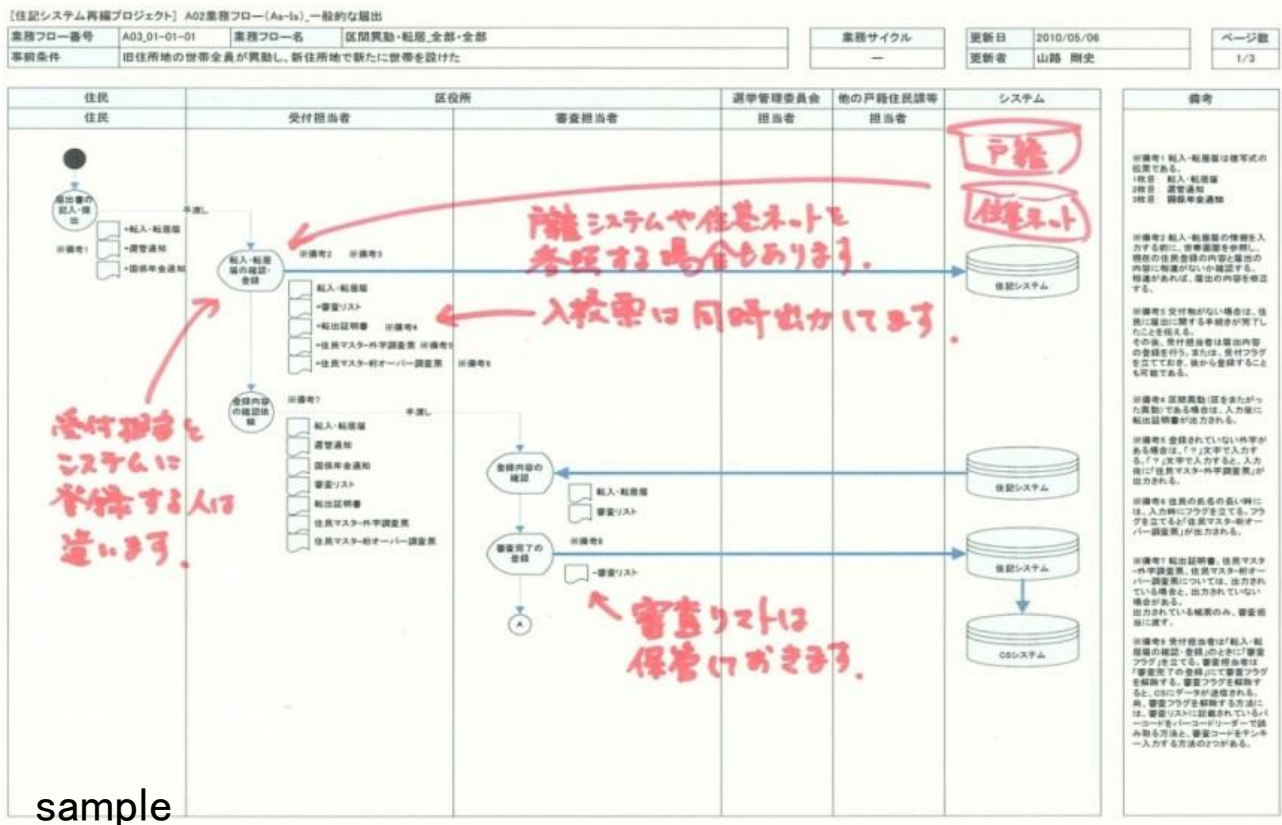
Email: n.izumi@aist.go.jp

URL: <http://staff.aist.go.jp/n.izumi/>

非専門職員による利用者主導開発の事前検証:

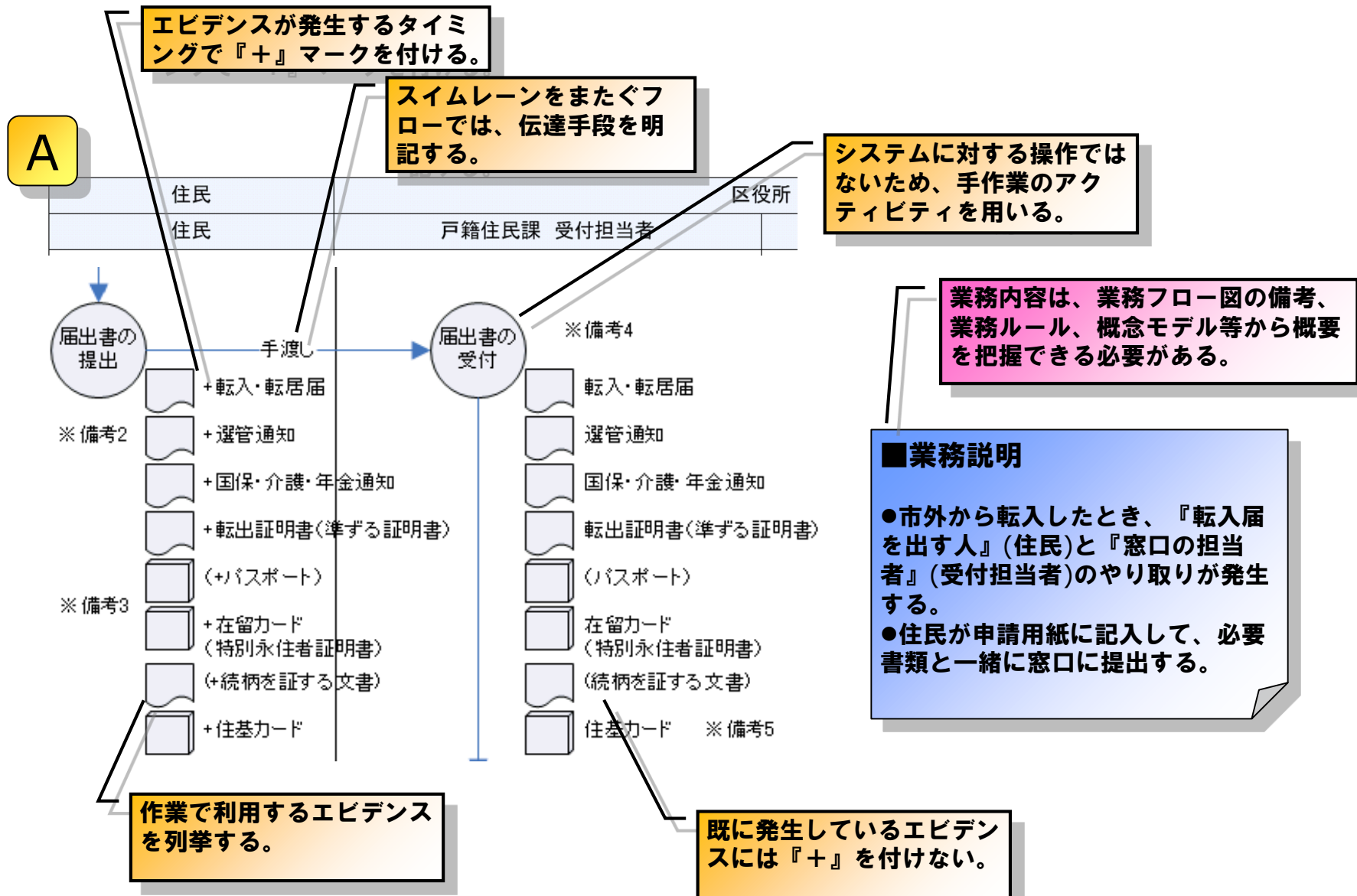
利用者が業務フロー図・ユースケース記述に注力することの効果

- ▶ **パイロット開発：児童相談所の障がい児施設補助関連事業役所**
 - ☑ 他都市の業務モデルが、別組織で再利用(容易に添削)できた！
 - ◎ ケースワーカが担当のため、ヒアリングなども困難で、かつ、ITの素養もない。
 - ☑ 業務モデル（仕様書）の流用効果は、予算に換算すると…！！



《業務フロー図のサンプル記述》

[業務説明] 窓口で市外転入の申請を受け付ける



《業務フロー図のサンプル記述》

[業務説明] 転入情報をシステムへ仮登録する

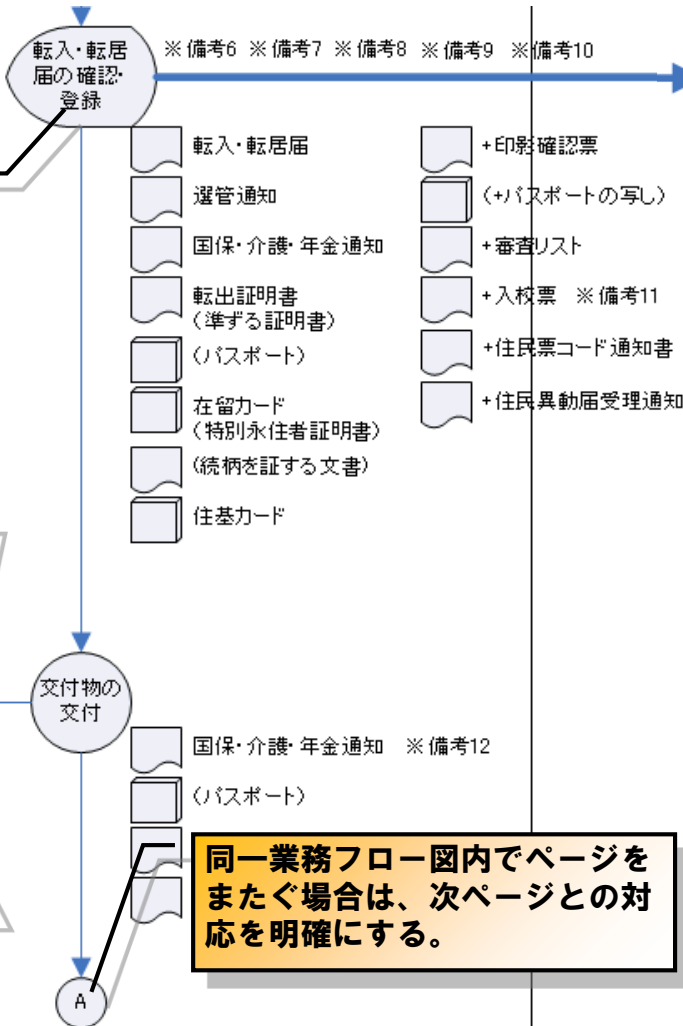
住民	区役所
住民	戸籍住民課 受付担当者

B

システムに対する操作は、システム操作のアクティビティを用いる。

帳票等は、業務フロー上、生成されると必ず消滅する。

アクティビティが消滅するタイミングで『-』マークを付ける。



登録先である住記システムを明記する。

矢印の向きはデータの流れの向きと考える。(今回は登録なのでシステムを向いた矢印となる) 該当作業の主となる内容で方向を決める。

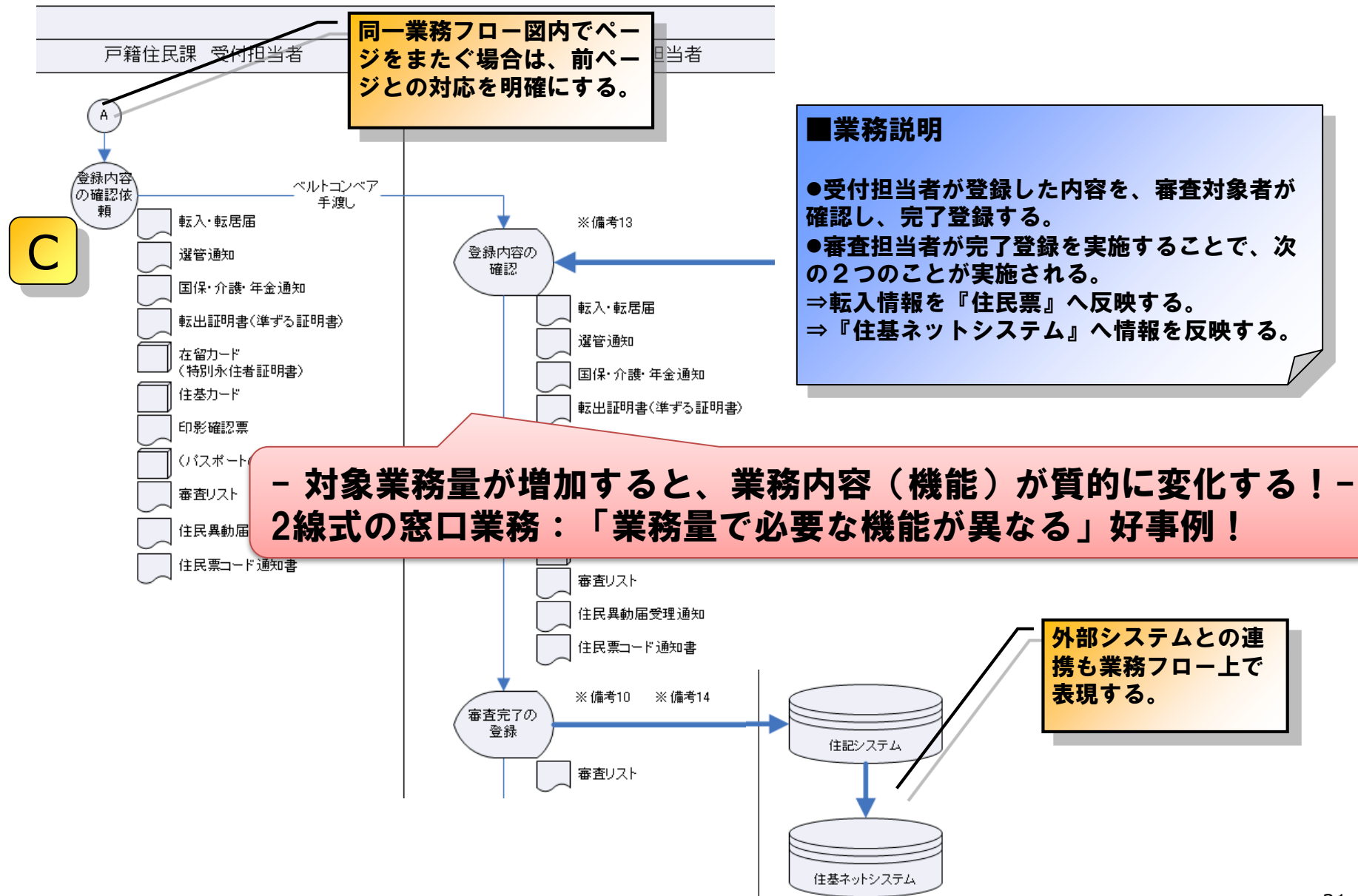
■業務説明

- 窓口担当者が受け取った内容をシステムに登録し、その結果として作成された書類(入校票等)と一緒に届出者へ一部提出物を返却する。
- このタイミングで登録されたものは、仮登録状態であり、住民票へはまだ反映されない。
- 国保・介護・年金通知は、以下の場合このタイミングで戻すことになる。
⇒国保加入者又は介護資格者のうち、すぐに異動内容の反映が必要な場合

同一業務フロー内でページをまたぐ場合は、次ページとの対応を明確にする。

《業務フロー図のサンプル記述》

[業務説明] 仮登録された転入情報を確認し確定させる



品質の良い成果物作成のポイント

品質の良い成果物作成のための「教育・セミナーの資料」の目次

- ▶ 成果物の『量』に対して、どのような『質』の管理（対策）を行えるか。
 - ☑ 受注者が作成した大量の成果物を、**一定のチェックの方法により、均質化を図るかがポイント。**
 - ◎ 記述の抽象度を上げて均質化することは、内容品質を失うことになるので、注意する。
- ▶ 成果物品質の向上のための**10の観点**
 - ☑ **成果物の書式に関する観点**
 - ◎ 1. 業務フロー図での条件分岐を明確にする。
 - ◎ 2. 業務フローをつなげる。
 - ◎ 3. 業務フロー図間のつながりを意識する。
 - ☑ **成果物の記述内容に関する観点**
 - ◎ 4. システム運用関連業務の漏れを無くす。
 - ◎ 5. 対象業務で扱われる帳票を明確にする。
 - ◎ 6. 対象業務（検討対象範囲）を網羅する。
 - ◎ 7. 作業担当者による記述粒度の違いを無くす。
 - ◎ 8. 外部システムとの連携を明確に定義する。
 - ☑ **成果物の関連資料に関する観点**
 - ◎ 9. 議事録を正しく作成する。
 - ◎ 10. 業務フロー図の備考から業務ルールを抽出する（抜き出す）。

方法論 = プロジェクト管理 × モデル理論

Noriaki Izumi Ph.D.

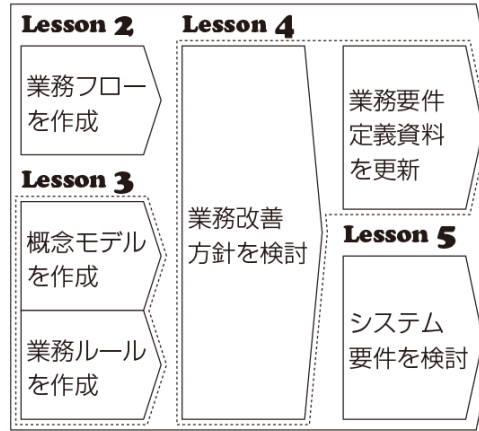
National Institute of Advanced Industrial Science and Technology (**AIST**) Japan

Email: n.izumi@aist.go.jp

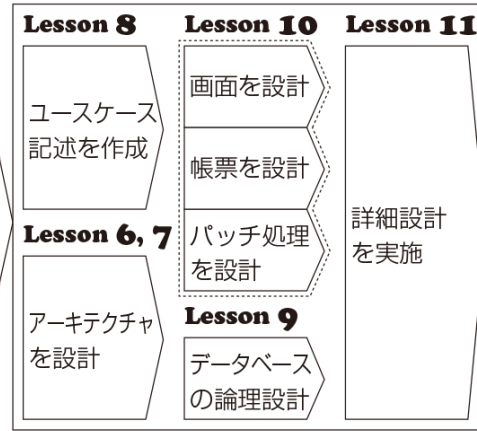
URL: <http://staff.aist.go.jp/n.izumi/>

プロジェクトノウハウを形式知にするための知識基盤

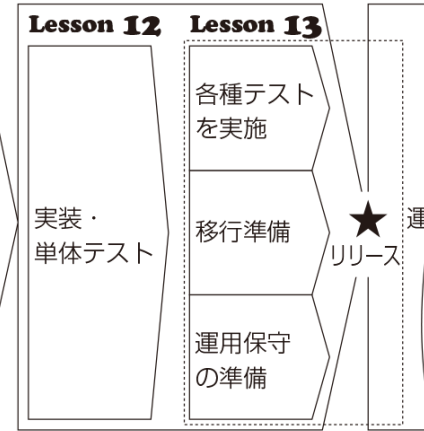
Part 2 要件定義プロセス



Part 3 設計プロセス



Part 4 開発プロセス～実稼働

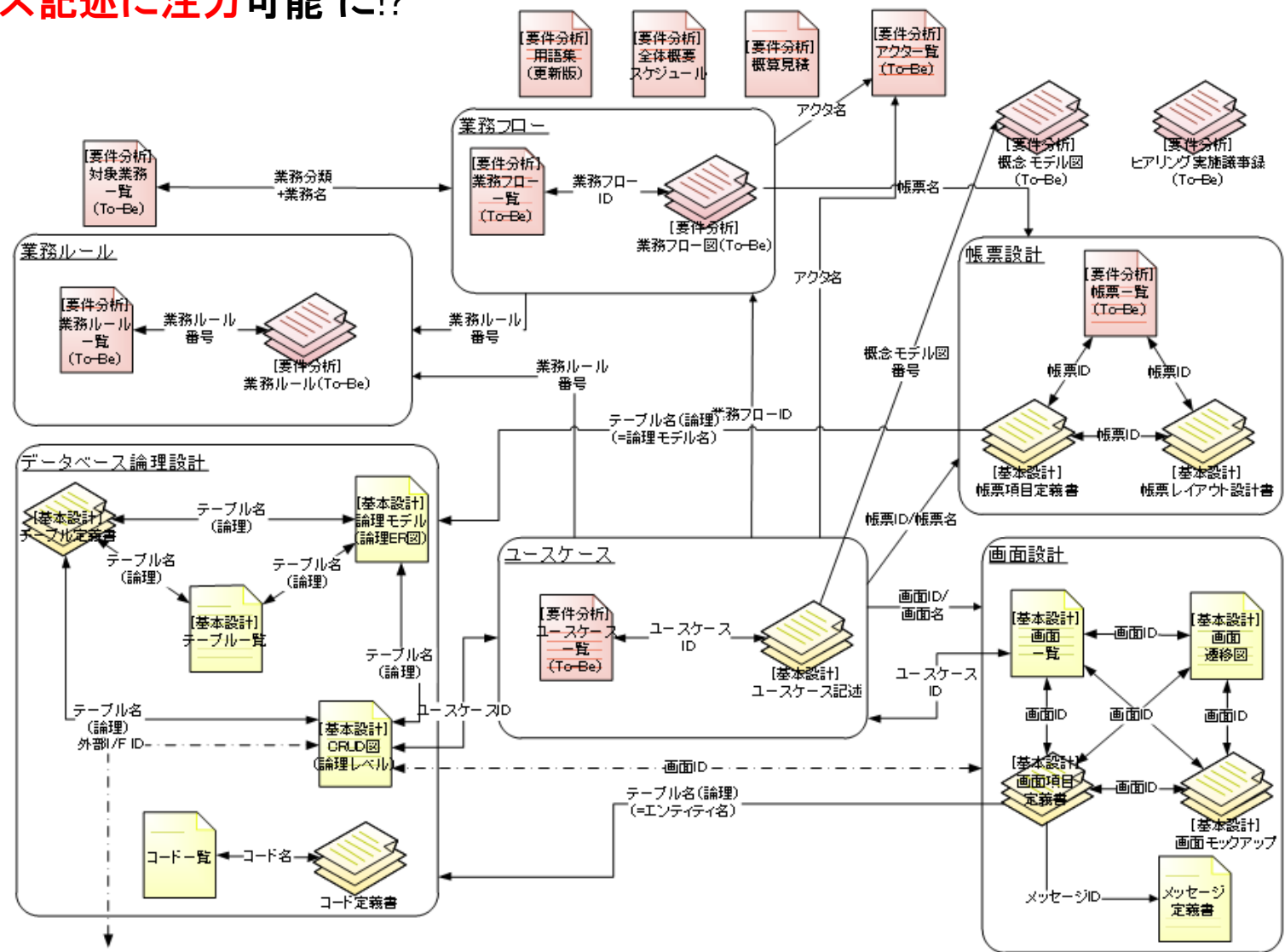


業務 要件定義書	システム要件定義書	基本設計書	詳細設計書	実装成果物	テスト成果物	移行 設計書
<ul style="list-style-type: none"> ・業務フロー図 ・概念モデル ・業務ルール など 	<ul style="list-style-type: none"> ・機能要件定義 ・非機能要件定義 など 	<ul style="list-style-type: none"> ・アーキテクチャ設計書 ・プロトタイプ ・各種規約・手順書 ・ユースケース記述 ・画面遷移図 ・画面モックアップ ・画面項目定義書 ・帳票レイアウト 	<ul style="list-style-type: none"> ・機能一覧 ・実装要素一覧 ・機能開発計画 ・シーケンス図 ・クラス定義書 ・物理 ER 図 ・テーブル定義書 など 	<ul style="list-style-type: none"> ・ソースコード ・単体テストコード ・各種設定ファイル など 	<ul style="list-style-type: none"> ・テスト計画書 ・テスト仕様書 ・テスト結果報告書 ・テストデータ ・テストツール など 	<ul style="list-style-type: none"> ・移行計画書 ・移行手順書 ・移行ツール ・移行データ など
						運用保守 設計書
						<ul style="list-style-type: none"> ・運用保守設計書 ・運用保守手順書 など

どれだけの成果物をいつ（どのタイミングで）
どの程度（正確に）作成・確認すると良いのか！?
⇒ 具体的にどのようなプロジェクト課題に直面するのか？

成果物のモデル理論に基づくトレーサビリティと 成果物間の関連に基づく現場での役割分担

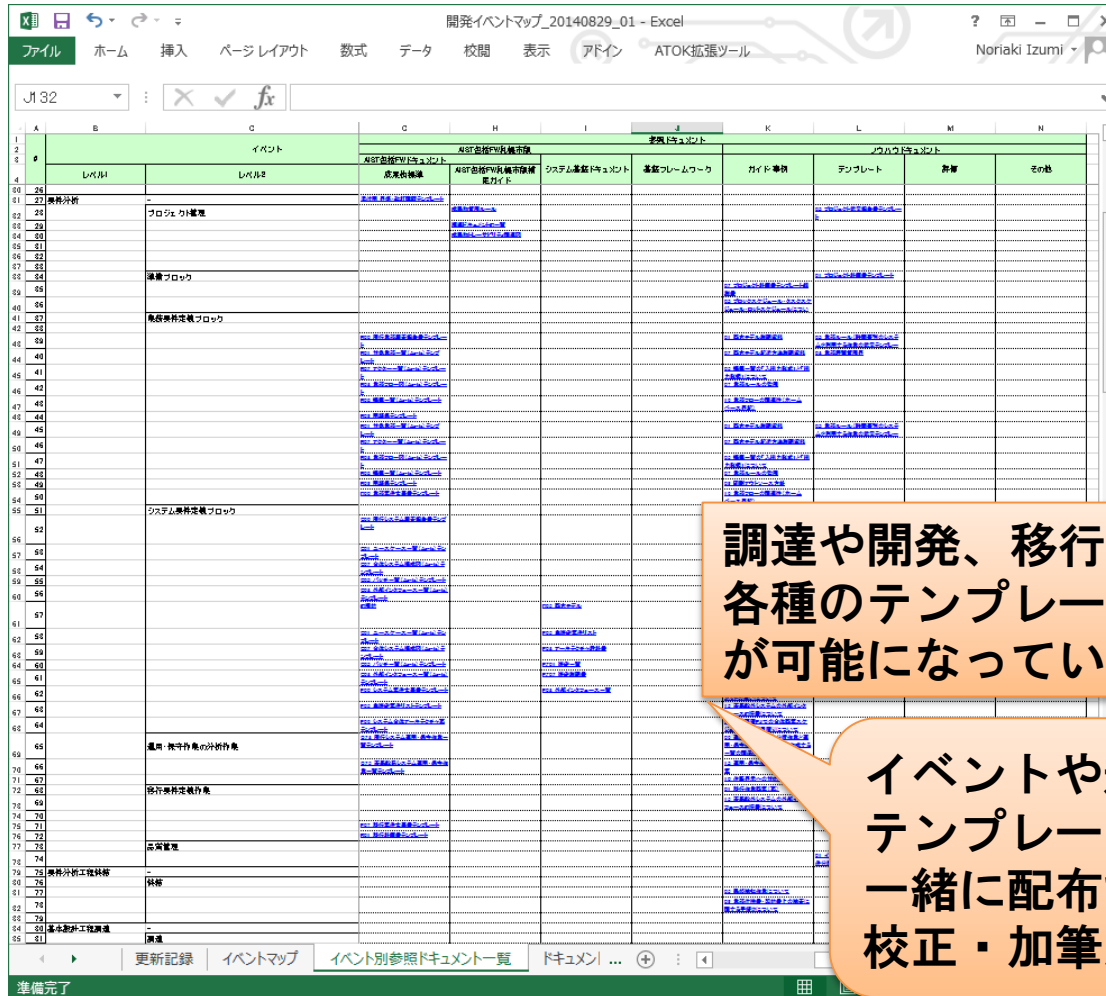
- 成果物間の厳密な関連定義により、利用者（非専門家）は、**業務フロー図・ユースケース記述に注力可能**に!?



プロジェクトノウハウを形式化するために:例1 詳細イベントと参照ドキュメントをマトリックスで管理

各種定義書と利用可能なテンプレートやガイド

工程ごとの詳細なイベント



イベント	参照ドキュメント							
	要件定義書	設計仕様書	開発ガイド	システム構築ドキュメント	基盤フレームワーク	ガイド事例	テンプレート	その他
開発計画	要件定義書	設計仕様書	開発ガイド					
設計	要件定義書	設計仕様書	開発ガイド					
開発	要件定義書	設計仕様書	開発ガイド	システム構築ドキュメント	基盤フレームワーク	ガイド事例	テンプレート	その他
検証	要件定義書	設計仕様書	開発ガイド					
移行	要件定義書	設計仕様書	開発ガイド					
運用	要件定義書	設計仕様書	開発ガイド					

調達や開発、移行などのノウハウが、各種のテンプレートとして蓄積・共有が可能になっている。

イベントや規約類を関連させ、テンプレートやリファレンスと一緒に配布することで、校正・加筆がルーチンになる！

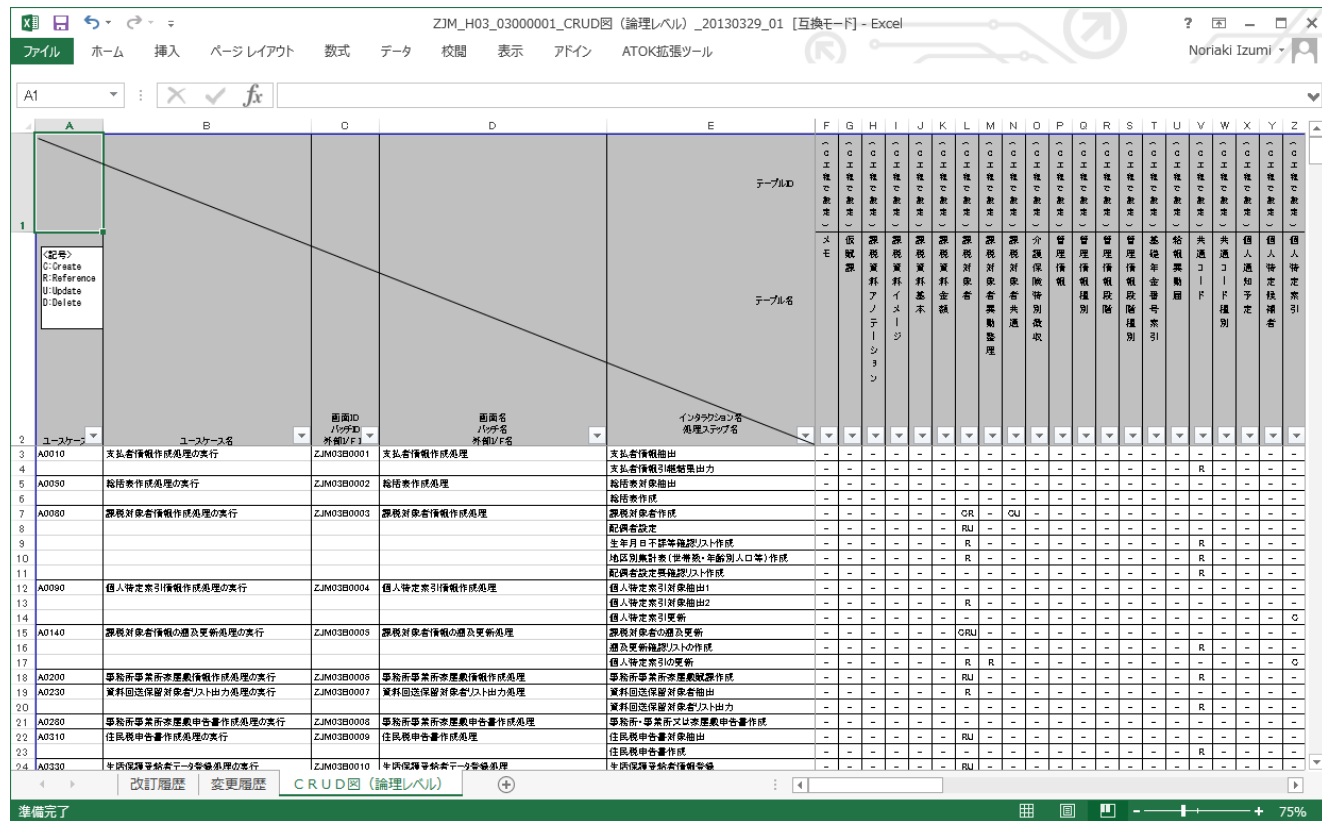
プロジェクトノウハウを形式化するために: 例2

成果物関係のチェックに関するツールとノウハウ

ユースケースと帳票類の関連を、機能群とデータ構造の関連により、抜け漏れと設計妥当性のチェック。

帳票からDBテーブル項目までのデータ階層

記述レベル階層で整理した機能



ユースケースID	ユースケース名	画面ID /フローID 参照先ID	画面名 /ボタン 参照先名	インタクション名 処理ステップ名	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A010	支払者情報作成処理の発行	ZJM03B0001	支払者情報作成処理	支払者情報抽出	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A0030	給付者情報作成処理の発行	ZJM03B0002	給付者情報作成処理	給付者情報抽出	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A0080	課税対象者情報作成処理の発行	ZJM03B0003	課税対象者情報作成処理	課税対象者作成	-	-	-	-	-	CR	CU	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A0090	個人特定索引情報作成処理の発行	ZJM03B0004	個人特定索引情報作成処理	個人特定索引情報抽出1	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A0140	課税対象者情報の値及更新処理の発行	ZJM03B0005	課税対象者情報の値及更新処理	課税対象者の値及更新	-	-	-	-	-	CRU	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A0200	事務所事業所差戻金情報作成処理の発行	ZJM03B0006	事務所事業所差戻金情報作成処理	事務所事業所差戻金抽出	-	-	-	-	-	RU	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A0230	資料回送保留対象者/以出力処理の発行	ZJM03B0007	資料回送保留対象者/以出力処理	資料回送保留対象者/以出力抽出	-	-	-	-	-	R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A0260	事務所事業所差戻金申告書作成処理の発行	ZJM03B0008	事務所事業所差戻金申告書作成処理	事務所事業所差戻金申告書作成	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A0310	住民税申告書作成処理の発行	ZJM03B0009	住民税申告書作成処理	住民税申告書抽出	-	-	-	-	-	RU	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A0330	年納保障手帳発行→事務所差戻金処理の発行	ZJM03B0010	年納保障手帳発行→事務所差戻金処理	年納保障手帳差戻金抽出	-	-	-	-	-	RU	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

オントロジー技術を援用した プロジェクト推進ノウハウの個別化

各種のプロジェクト標準に『魂』を込めるための
テンプレート・リファレンスの具体化

Noriaki Izumi Ph.D.

National Institute of Advanced Industrial Science and Technology (**AIST**) Japan

Email: n.izumi@aist.go.jp

URL: <http://staff.aist.go.jp/n.izumi/>

オントロジーとは？

Boyack, K.W., Klavans, R., & Börner, K. (2005, in press). Mapping the backbone of science. *Scientometrics*.

概念定義（属性・属性値）＋概念階層

「『機械(コンピュータ)』のための大量データの解釈定義」

「『機械学習システム』のためのカンニングペーパー」

☑ 『機械学習システム』が学習完了するためには、最低でも数千のサンプルが必要

✗ 「『和泉憲明@産総研』と『和泉憲明@静岡大学』が同一人物である」は判定不可能

「同じモノの定義(オントロジー)」を与えて
データ空間を圧縮する。

「何と何が同じか」の定義（オントロジーの基本は「同じ」の定義）

海外旅行にオントロジーがあると…

The diagram illustrates an ontology for an entry card. It features several nodes and their relationships:

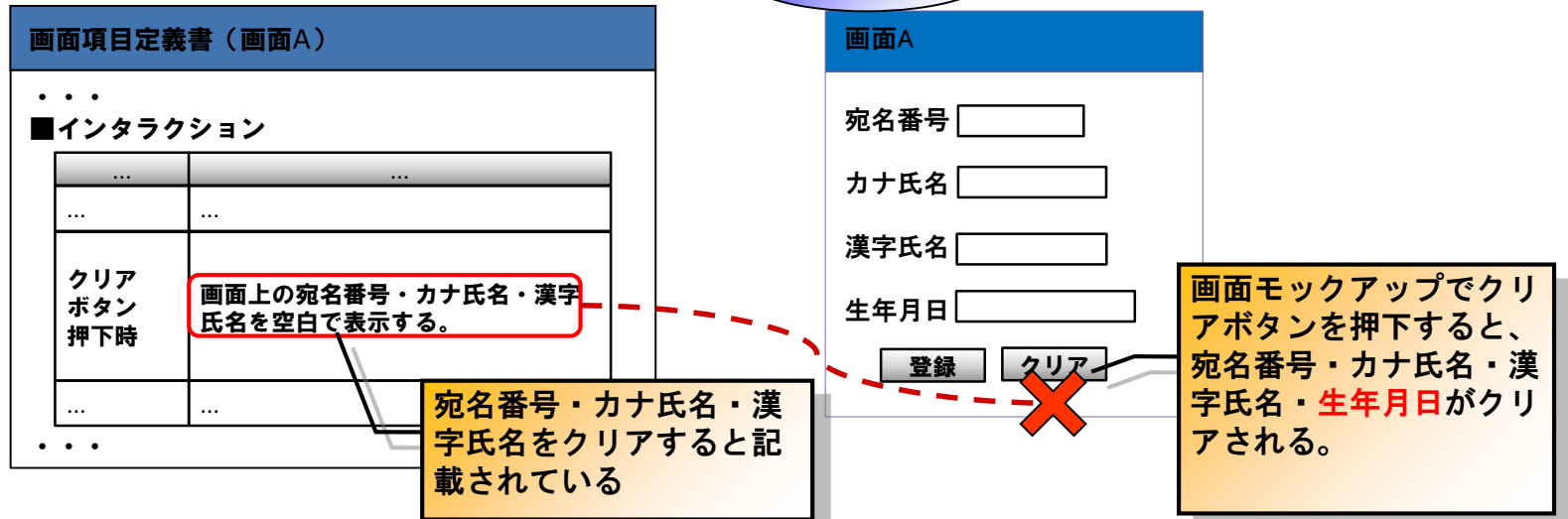
- 入国者 (Entrant)** is connected to **氏名 (Name)**, **パスポート (Passport)**, and **ビザ (Visa)**.
- 氏名 (Name)** is further divided into **氏 (Surname)** and **名 (Given Name)**, which together form **名前 (Full Name)**.
- 到着便 (Arrival Flight)** is connected to **便名 (Flight Name)**.
- パスポート (Passport)** is connected to **発行地 (Issuance Location)** (都市 - City) and **有効期限 (Validity Period)**.
- 番号 (Number)** is connected to **有効期限 (Validity Period)** and **日付 (Date)**.
- ビザ (Visa)** is connected to **番号 (Number)** and **有効期限 (Validity Period)**.
- 属性 (Attribute)** is connected to **ビザ (Visa)**.
- クラス (Class)** is connected to **氏名 (Name)**.

The background shows an 'ENTRY CARD FOR FOREIGN TRAVELLERS' with fields for Family Name, Given Names, Date of Birth, Nationality, and Passport No. Below it is a Thai Immigration arrival card for ITAGAKI TOSHIO, with fields for Family Name, First Name, Middle Name, Nationality, Passport No., Date of Birth, and Address in Thailand. A pink cloud contains the text '機械処理の精度とその設計に好影響' (Positive influence on the accuracy and design of mechanical processing).

AIST包括フレームワークの個別化事例

成果物標準を機能させるために

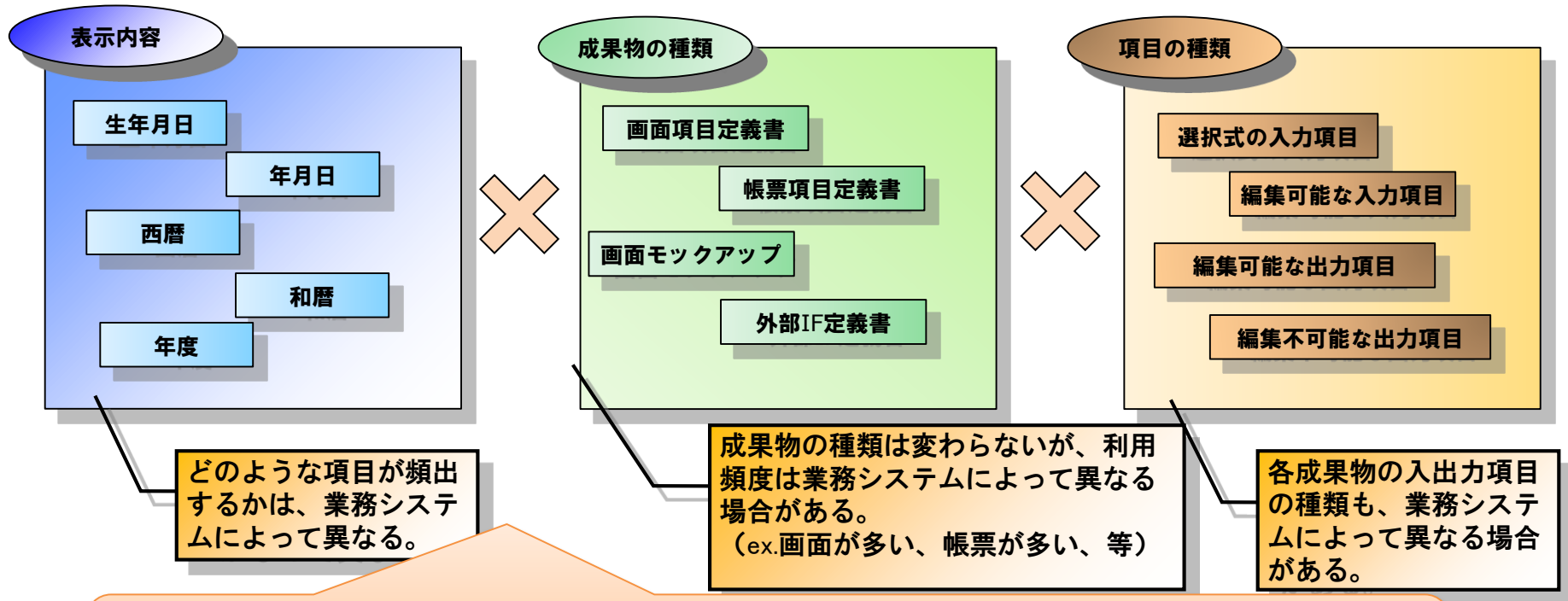
- ▶ 成果物量の増大に起因した相互参照（内容整合性）に関する品質問題
 - ❑ 大量の成果物・成果物間の相互参照（内部整合性）に関する誤りは、**単純な存在チェックでは発見しづらい。**
 - ❑ にもかかわらず、現場で運用されている標準・ルールの多くは「**項目定義は正確に**」「**整合性を確認すること**」のみ！
- ▶ 品質インシデントの実例
 - ❑ 画面モックアップにてクリアボタン押下時にクリアされる項目が、画面項目定義書に記載のクリアされる項目から漏れている。



- ▶ **ワインバーグによる1971年の指摘：**
 - ❑ 「『人の活動としてのプログラミング』… プログラミングは単なる**ハードウェアの問題、ソフトウェアの問題ではないのだ。**」
 - ❑ 「**ものの書き方を学ぶための一つの方法は書いてみることだが、それ以外にもう一つ、どういう種類の書きものについても、読んでみるという方法がある。**」
- ⇒ **機能しない多くの開発標準は、「辞書と文法書を与えればすばらしい文学が書ける」という思い込み !?**

成果物品質のための作業統制を阻害する課題

- ▶ 課題要因の実例：年月日入力形式が不一致となっている。
 - ❑ 画面モックアップを確認すると、様々な年月日に関わる入力項目が存在するが、**成果物標準としてのルールが不明確**である。
 - ❑ **成果物作成が統制されていない**。⇒ **ヒアリングや確認も統制されていない**。



ルールが遵守されていない ⇒ 事前の指示が徹底されていない！
 ⇒ (どのみち)ヒアリングや成果物の作成・修正すると考え、不要なコストを浪費している！?
 ⇒ 『鬼の婦長さん』登場！！

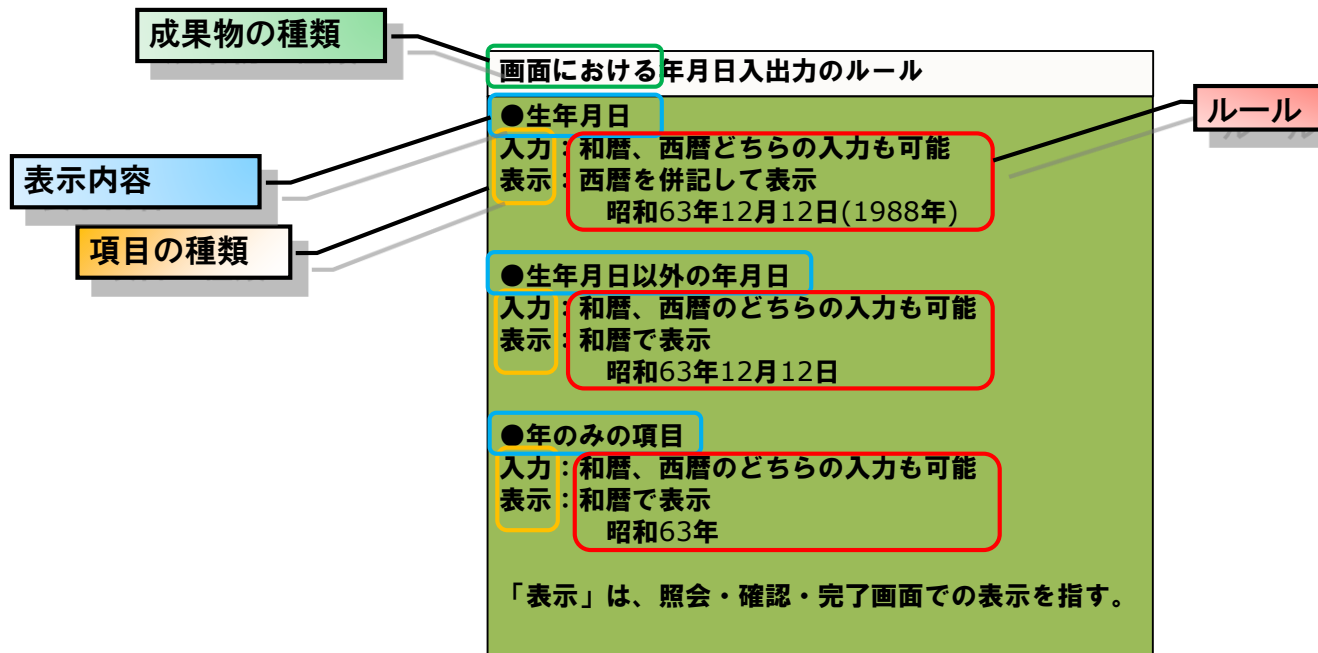
対策：プロジェクトルールによる成果物品質の統制

➡ 現実問題として…

- ☑ 通常、利用者・発注者は、画面モックアップの確認のみ。
- ☑ 画面項目定義は、基本、開発ベンダのための文書であり、全項目に関するヒアリングやチェックは困難！

➡ 対象業務領域のパターンとしての参照成果物の共有

- ☑ 入出力項目について整理し、テンプレート化・ルール化する。



➡ <<補足>>

- ☑ 具体例として画面を取り上げたが、帳票についても同様なルールを適用している。

業務領域に特化したプロジェクト固有ルール の 確立

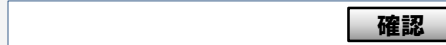
具体例：画面設計ルールで統制を行った例

- 画面上の部品の配置は、ユーザインタフェース規約で定義をしているが、業務システムの特性に応じて、オリジナルのルールを逸脱しない範囲で業務固有の取り決めに追加すると、品質統制と作業量削減の両方に効果がある。
- 以下は、利用する業務（画面）の特性に応じて、業務（画面）と画面のボタン配置をパターン化した例である。

例：画面上のボタン配置の標準化

給与支払報告書（新規）画面、
申告書等（新規）画面、
...

パターン1



課税対象者（完了）画面、
支払者情報（完了）画面、
...

パターン2



対象者検索画面、
課税資料検索画面、
...

パターン3



課税対象者（確認）画面、
扶養・専従者（確認）画面、
...

パターン4



- ✓ 画面項目だけでなく、ボタンの振る舞いまで統制可能となるので、
- ✓ 発注側のヒアリングや添削の負担だけでなく、
- ✓ 開発側の作業まで効率化！



- ✓ ノウハウを業務固有のルールに
 - ・ 共通化と開發生産性向上
 - ・ 規模圧縮と大量生産

上記と同様の例として、住所コードの出力形式や、画面入力項目が規定値・上限値を超えた場合の振る舞い、ユーザの権限有無に応じた画面の振る舞いルール、等があげられる。

オントロジー導入の効果

形式知化による成果物運用のスキル向上の例

内部レビュー後

納税証明書（一般）の証明書発行要求時に、市・道民税証明（所得証明）・納税証明・課税証明請求書を元に、請求者情報を特定し、賦課情報や収納情報を入手して画面でそれらの情報を確認する。

確認結果により、証明書の発行の可否を判断する。

賦課収納状況の確認を行う際に、同姓同名該当確認や同一法人該当確認・支店該当確認の実施を行うことがある。

一般的な成果物標準に基づく指摘後

納税証明書（一般）の証明書発行要求時に、市・道民税証明（所得証明）・納税証明・課税証明請求書を元に、請求者情報を特定し、賦課情報や収納情報を入手して画面でそれらの情報を確認する。

職員が、画面に表示されるメッセージを確認し、証明書の発行の可否を判断する。

なお、賦課収納状況の確認を行う前に、同姓同名該当確認や同一法人該当確認・支店該当確認により、納税義務者を職員にて確認する。

ガイド・規約の改善（プロジェクトルール化）に基づく赤入れ後

納税証明書（一般）の証明書を発行する際に、職員は証明書の発行可否を判断するために、市・道民税証明（所得証明）・納税証明・課税証明請求書を元に、同姓同名・同一法人・同一法人の支店候補より請求者を特定し、税目・年度・税額・納付額未到来額・未納額とともに、未納税目有の表示を確認する。

ガイド・規約の発注者（職員）主導による改善

➡ 当初、策定されていた規約

5W1Hを正確に記載し、もれなく、曖昧なく、記述すること。

➡ 最終的に現場で運用されている規約 ?

下記の記載があるかを確認すること。

- ①どんな契機（タイミング・条件・目的 等）で
－「いつ」「何のために」
- ②何を入力・対象とし
－「何を」
- ③どのような操作・処理をして
－「どうやって」
- ④結果としてどうなるのか
－「どうする（どうなる）」

※「誰が」が主語として記述される

※「何を」の部分では、入力帳票なのかDB参照なのか等も判別できるような記述とする。

※「どうやって」の部分は、入力帳票のどの部分を対象としているのか、DBのどの情報を対象としているのか、まで記述する。（どの部分、どの情報は概念モデルから）

※動作の種類を『標準化』するほうがよい。

（業務フロー図のアクティビティの表記を統一しているのと同様のことを実施する）

※意識してほしいこと。その動作が、登録することなのか、複数から選択することなのか、変換することなのか、など、どのような動作であるのかが明確になるようにする

おわりに

Noriaki Izumi Ph.D.

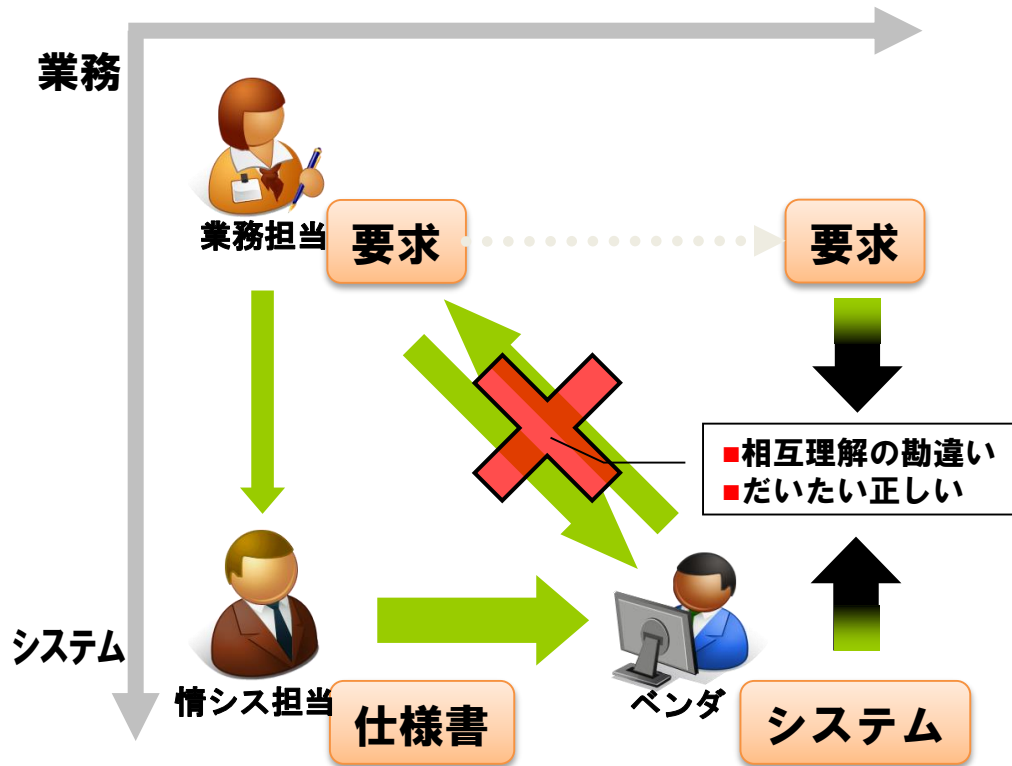
National Institute of Advanced Industrial Science and Technology (**AIST**) Japan

Email: n.izumi@aist.go.jp

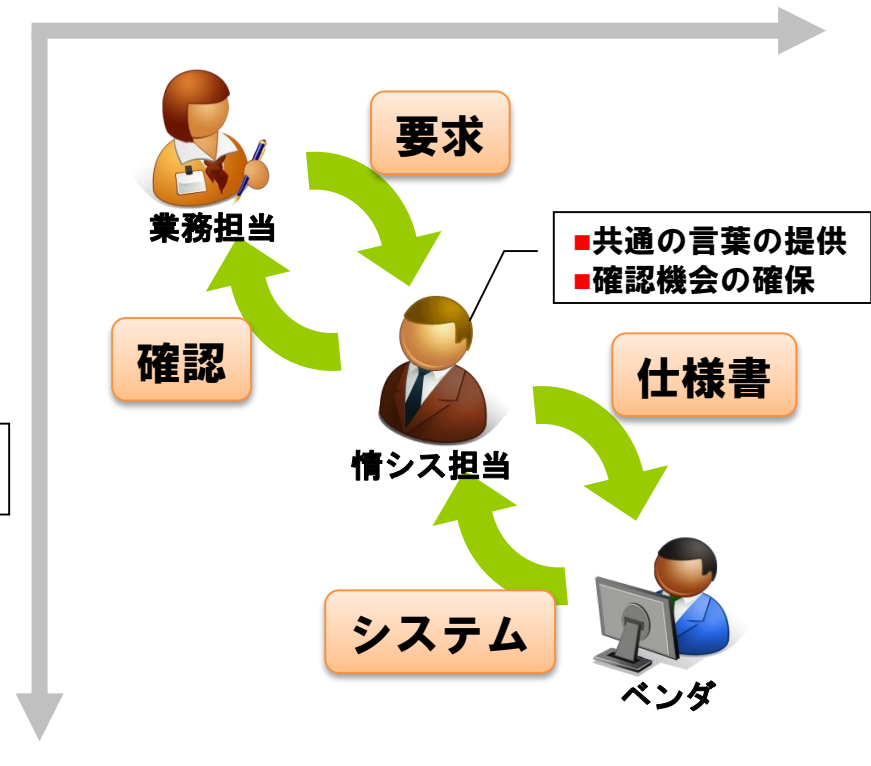
URL: <http://staff.aist.go.jp/n.izumi/>

上流工程支援と下流工程技術を両立させるために!?

従来システム調達



利用者主導によるシステム導入



<発生する様々な問題>

- 所管課の要求を満たさないシステムの完成
- 正しく開発したのに現新比較で異なる結果
- 瑕疵と変更が区別できずベンダーロックに

<従来の問題に対する改善>

- 成果物の品質向上による委託管理能力の改善
- 上流工程品質の向上による下流工程の効率化
- 成果物トレーサビリティによる最終品質の確保

まとめ:オントロジーとモデル理論に基づく 下流工程技術のための上流工程品質の向上アプローチ

- ➡ **実践フレームワーク（AIST包括FW）の提供：**
 - ☑ **情報システム開発に関する品質向上アプローチ**
 - ⊙ 開発工程の生産性向上が、システム開発全体に好影響をもたらすための上流工程品質
 - ⊙ 各種のツール・パッケージと業務システムとの意味内容的なギャップを埋めるためのオントロジー
- ➡ **理論的アプローチ：**
 - ☑ **古くて新しい問題への対応**
 - ⊙ 概念獲得とチームワーク、知的生産性向上に関する方法論（オントロジーとKM）
 - ☑ **技術者の作業視点を重視**
 - ⊙ 70年代からの技術教養を重視 … 辞書と文法書だけではフランス文学は執筆できない。
- ➡ **構成：**
 - ☑ **利用者・発注者のための情報システム開発における合意形成**
 - ⊙ 生産性向上のための標準開発手順を、**ドメイン固有知識**で強化
 - ⊙ モデル理論に基づいたトレーサビリティによる成果物**記述法と品質チェックリスト**など
 - ⊙ 非専門職員のための**教育研修とノウハウ文書群**
- ➡ **効果：**
 - ☑ **開発コスト削減と品質向上、現場力向上を同時に達成**
 - ⊙ **利用者による仕様確認が可能となり、開発ツールの生産性が最大化**されるため、発注者・受注者の双方に好影響が確立。
 - ▶ **ドメイン依存のテンプレートと参照モデルを用いた『亜種開発』**が可能となり、品質の統制も容易になる！
 - ▶ 横浜市役所や札幌市役所で、開発費用の抑制効果を実証！（ただし、再委託構造を優先して、成果物間の関連を無視した役割分担をすると、…）
 - ☑ **共通部分を、IT総合戦略室の「実践フレームワーク」として配布（予定）**
 - ⊙ 今後、多様な方法論のポテンシャルを活かすための枠組みとして！

付録1：品質改善とスキル向上の理論的背景

－ 質疑応答用のスライド －

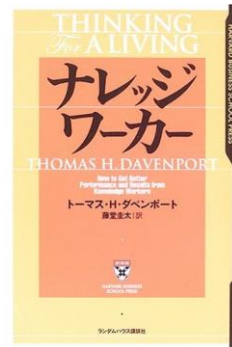
Noriaki Izumi Ph.D.

National Institute of Advanced Industrial Science and Technology (**AIST**) Japan

Email: n.izumi@aist.go.jp

URL: <http://staff.aist.go.jp/n.izumi/>

AIST包括フレームワーク: 上流支援と下流技術の融合



- ▶ ホワイトカラーの**知的生産性向上に関するノウハウ**
 - ☑ ルーチン化・パターン化がカギ
 - ☑ 知識集約型作業を、単純作業の繰り返しと、少量の試行錯誤に！
- ▶ 業務システム最適化計画の枠組みを**ソフトウェア研究の成果で強化**
 - ☑ 標準記法を数理モデルにより拡張、参照モデル開発に基づいて標準策定、**プロトタイプ開発にてメトリックス計測と傾向分析**（メトリックス分析）、**オントロジーにより品質向上のためのノウハウをガイド**として策定
 - ☑ 標準記法と一般的ノウハウは、**IT総合戦略室の実践フレームワーク**として採用

プロセス標準

成果物標準

業務共通化と開發生産性向上のノウハウ



開発規模圧縮と大量生産のノウハウ

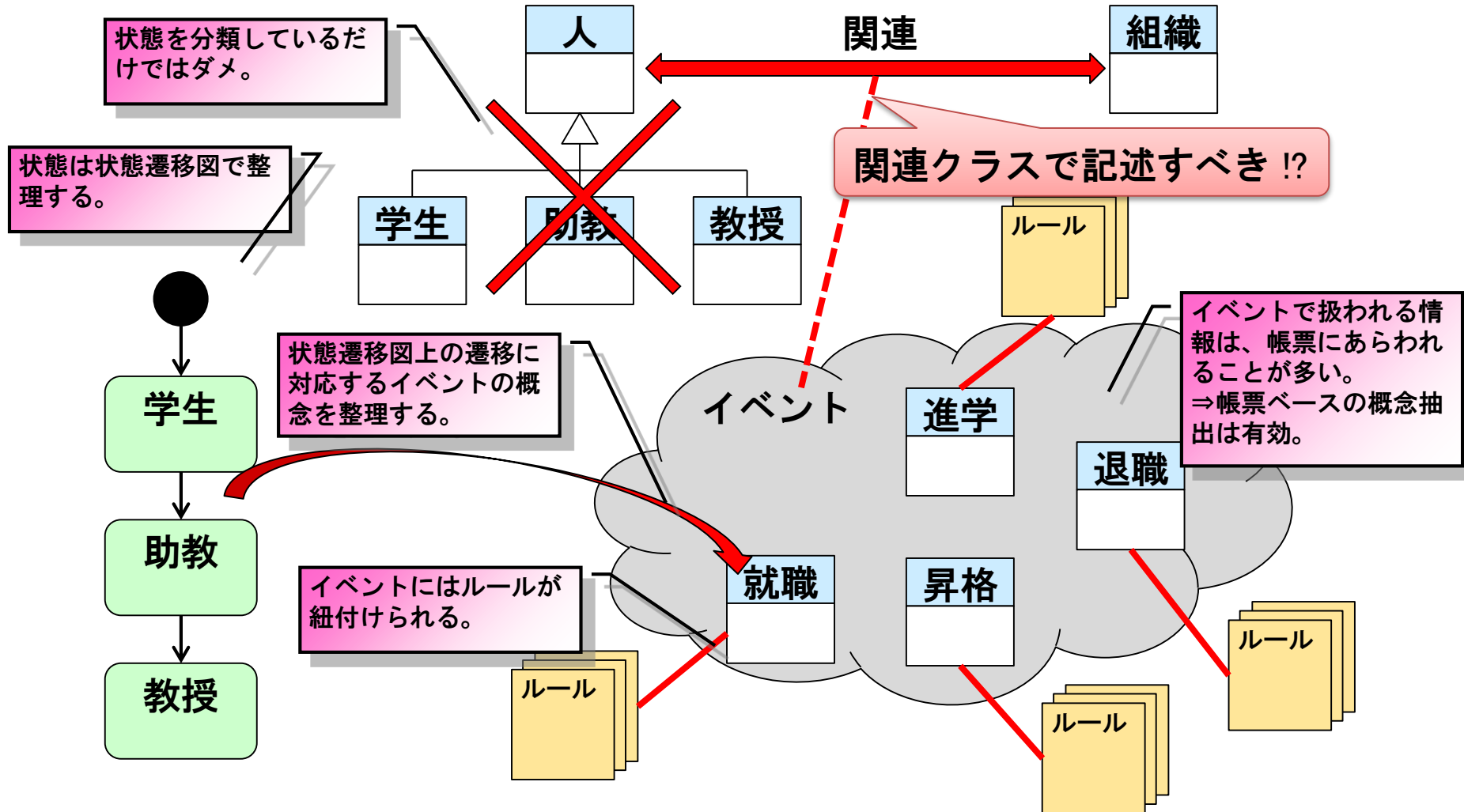
基盤技術

業務非依存のノウハウを
プロジェクト**固有のルール**
に作り込むスキル

盲目的な標準化・効率化でなく
オントロジー技術によって
用語などで**成果物を統制**し、
経験知・ノウハウが分野固有
のルールになり、**実働可能に！**

品質課題：図式仕様でさえ正確に読み書き出来ない!?

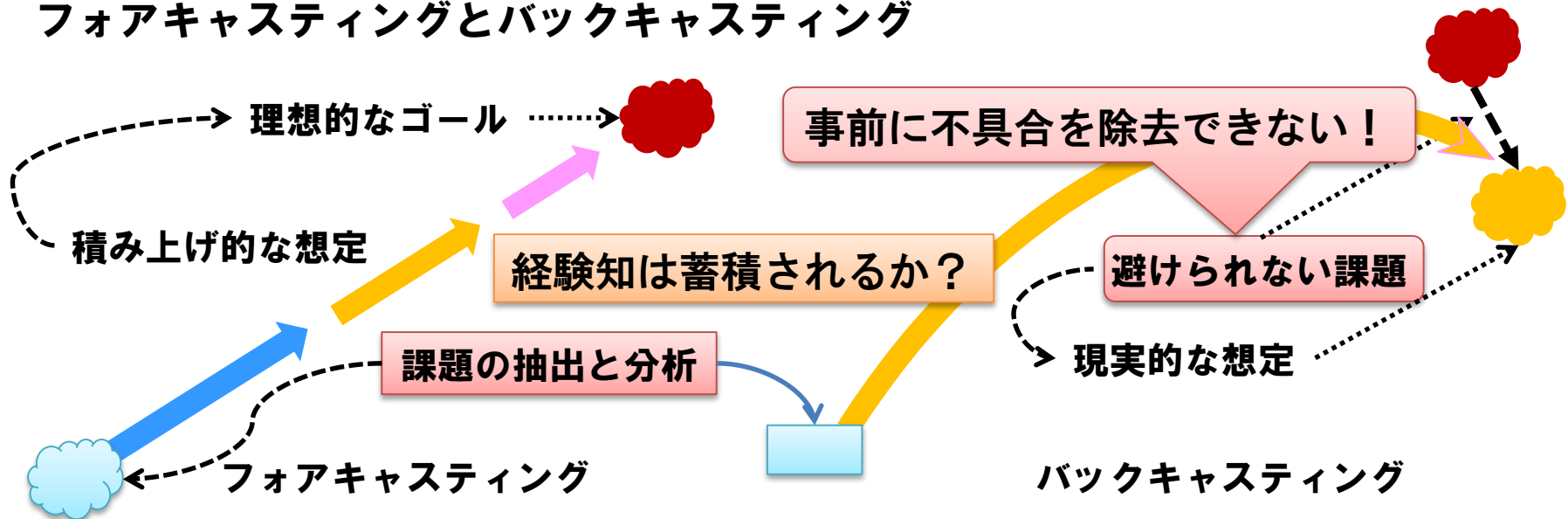
- 概念モデル：ドメインを理解していることを確認できる1つの指標であるが、なかなかうまく活用できていない。



システム開発におけるプロジェクト管理のスキル

- なぜ、プロジェクトは理想的に進行しないのか？
 - ◆ 全く同じプロジェクトは存在しないため、**過去の知見(技術・経験)が役に立たない??**
 - 多くの失敗プロジェクトは、**理想的な状況を願望として計画**してしまっている。
 - ◆ 運良く作業内容が理解できても、次に、**具体的な効率化の方策**がわからない!?
 - **品質改善やインスペクションが発注者任せ**になってしまう。

■ フォアキャストとバックキャスト



- 開発プロジェクトにおける本当の課題：何に関する『現状分析』が重要なのか？
 - ◆ 知見(技術・経験)が生かされるかどうかは、プロジェクトの**環境との相対関係**で決まる。
 - **ブルックス@75**
 - ◎ 「ソフトウェア技法をどう改善したら良いかという中心的疑問は…、人間の問題に集中する。」

付録2:超高速開発手法への適用

- 具体的な下流工程を想定すると? -

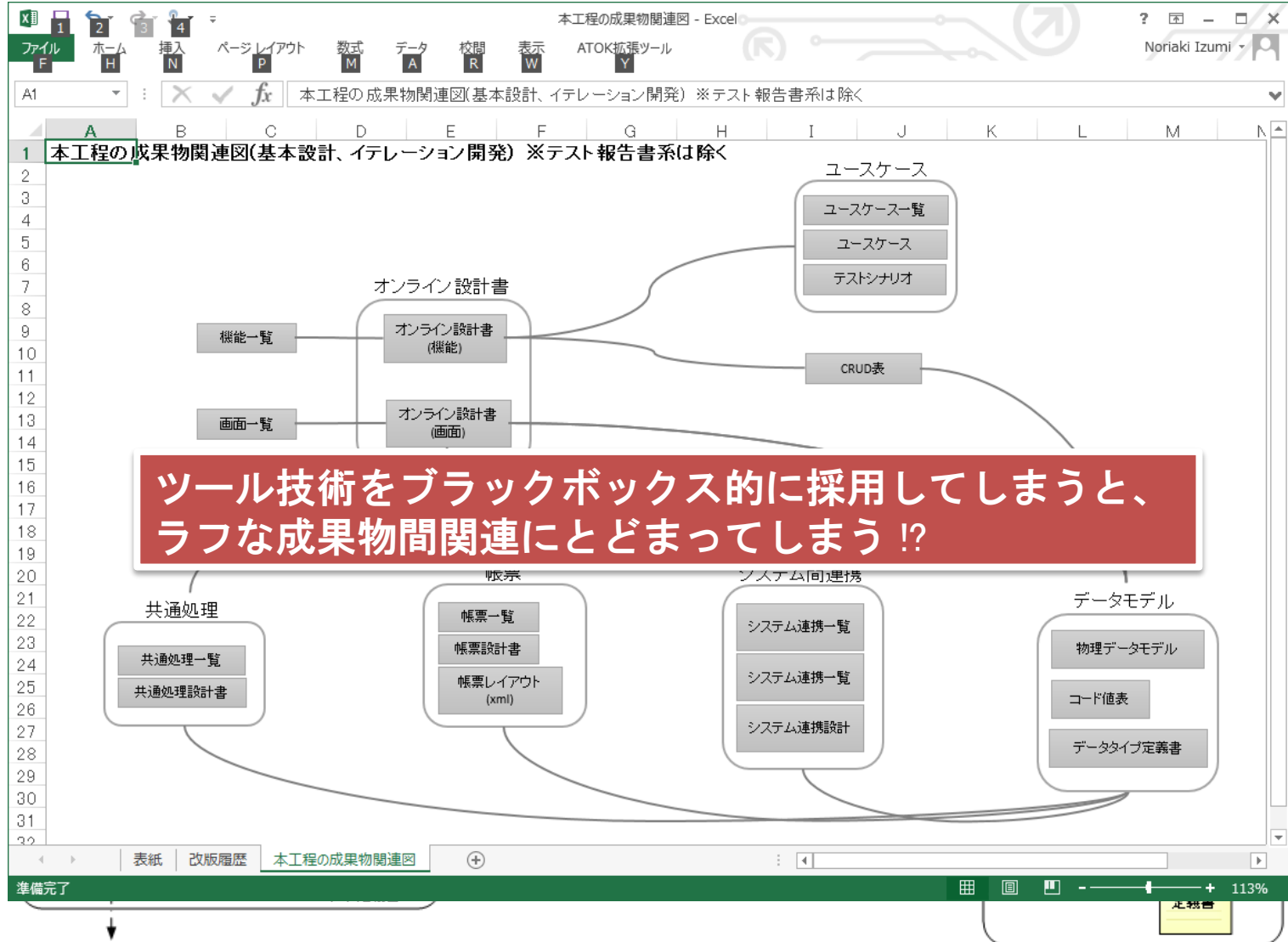
Noriaki Izumi Ph.D.

National Institute of Advanced Industrial Science and Technology (**AIST**) Japan

Email: n.izumi@aist.go.jp

URL: <http://staff.aist.go.jp/n.izumi/>

(超高速開発手法をツールとして採用した場合の) 成果物関連図



高速開発手法とスクラッチ開発とのプロセス比較

Excel: 工程と成果物の関連図 - Excel

作業工程

作業工程	設計作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業
設計工程	基本設計(外部設計) (確定時点)	※開発作業	設計(内部設計)-開発	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業
開発工程	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業
実行工程	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業	開発作業

成果物間関連がラフになると、
 ノウハウ共有は、ツールの操作スキルに隠蔽される! ?

準備完了

開発標準と標準プロセスの強化

➡ 開発規約

- ☑ *概要説明、*標準システム構成、*開発方針、*特定ツールコーディング規約、*テスト関連規約、*帳票関連規約、*自動テストツール規約

➡ 共通部品・共通機能

- ☑ *スタイルシート、*バリデーション、*メッセージ、*日付・年齢計算、*排他計算、*その他（遅延金計算など、業務的な難易度が高くて流用可能性の高いモノ）
- ☑ 各部品を構成する機能（メンテナンス画面、ログのダウンロード・移動、祝日計算、…）

➡ テストツール関連

- ☑ インストール手引、ツールガイド、サンプルテストケース、ハンズオンテキスト、…

➡ ドメイン依存の概念構造が重要!! ⇒ テストに直結 !?

大規模システム開発のノウハウを移植

品質管理

- ☑ *redmineの
について0
た開発テ
04.p
務
- ☑ ユー

工程の作業の進め方
システム連携を捉え
の進め方について
の整理02.xlsx, *業

システムテスト以降の
連携テストや
データ移行・受入テスト
の周辺を強化 !?

開発標準

- ☑ *共通テンプレ
- ☑ 成果物チェックリスト
 - ◎ *シナリオテスト（動くもの）チェックリスト01.xlsx, *シナリオテスト用チェックリスト01.xlsx, *成果物
チェックリスト01.xlsx
- ☑ 成果物テンプレート
 - ◎ 01.プロジェクト管理作業/, 03.総合/, 運用保守作業の設計作業/, 05.移行方針設計作業/
◎ 06.移行作業/

書01.xlsx



システムが正しいことを
確かめるための枠組み

チェックポイント確認表
、移行リハーサル

システムID].xlt, 移

- ▶ 01.移行計画/,システム移行手順書[システムID].dot 移行計画書[システムID].dot
- ▶ 02.移行リハーサル/、移行リハーサルチェックポイント確認表
(移行
実施
- ◎ 07研
- ◎ 09.全体システム
ID].xlt, 全体システムテスト報告書[シス
- ◎ 全体システムテスト仕様書[システム
- ☑ 成果物テンプレート定義書
 - ◎ *共通テンプレート定義書03.docx