

プロセスを設計する

～PFDによるプロセスの表現と設計～

Embedded Technology 2012
スペシャルセッション資料
(第2セッション)

派生開発推進協議会

梶本 和博

(株式会社 エクスモーション)

(C) copyright 派生開発推進協議会

アジェンダ

1. 開発現場の現状
2. プロセスの表現
3. プロセスの設計

1. 開発現場の現状

1.1 混乱するプロジェクト

1.2 遅延するプロジェクト

1.1 混乱するプロジェクト

- ▶ 今回のプロジェクトで作成しなければならない成果物全体が見えていない
 - ▶ 次に何を作れば良いのか？
 - ▶ どこまでやれば終われるのか？
- ▶ 今回のプロジェクトはどのように実施するか見えていない
 - ▶ 今までの習慣で実施
 - ▶ リーダーによる都度指示で実施（リーダー任せ？）

1.2 遅延するプロジェクト

- ▶ 今まで実施されてきた方法をそのまま実施している
 - ▶ 過去には上手くいったが・・・

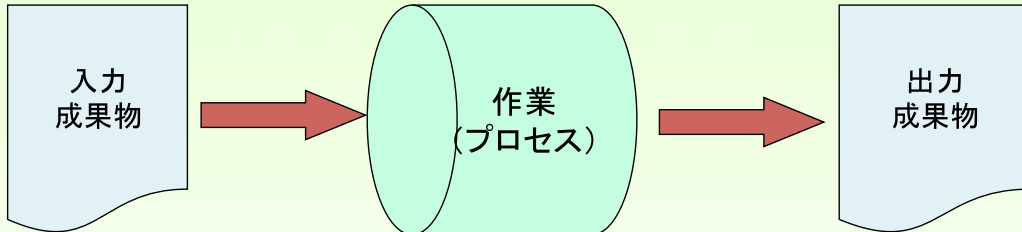
- ▶ 問題が生じると成果物が増加/肥大する
 - ▶ これが足りなかったから・・・
 - ▶ この作業を実施していなかったから・・・

2. プロセスの表現

- 2.1 プロセスとは
- 2.2 プロセスを表現する

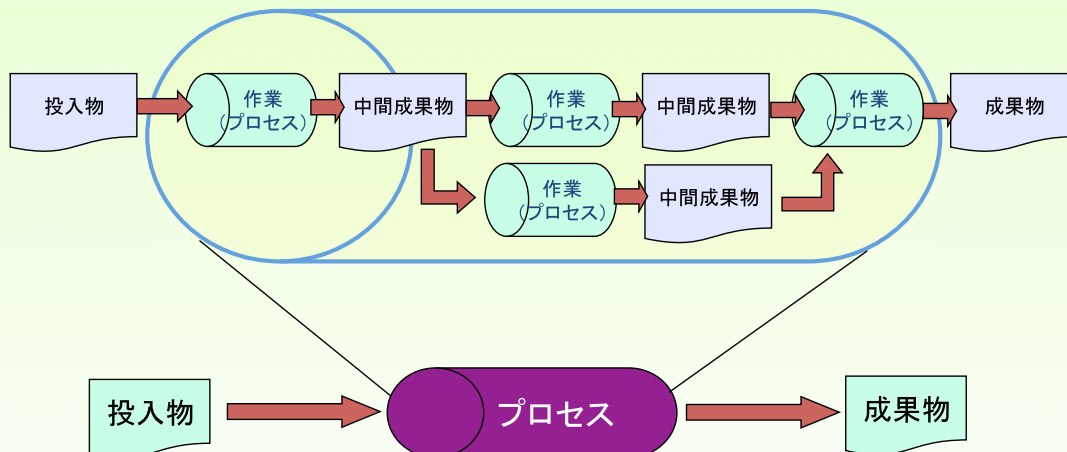
2.1 プロセスとは

- 作業は入力成果物を出力成果物に変化させる



2.1 プロセスとは

- 成果物と成果物は作業で連鎖している



2.2 プロセスを表現する

➤ プロセスの表現方法

表現方法	特徴	問題
ワークフロー	<ul style="list-style-type: none"> ✓作業の流れがわかりやすい ✓日本では広く使われている? 	<ul style="list-style-type: none"> ✓成果物と作業の関係が見えない(見えにくい) ✓テーラリングに対応しにくく、既成のものをそのまま流用する傾向
WBS	<ul style="list-style-type: none"> ✓作業の粒度を分解して表現できる ✓スケジュールに展開しやすい 	
DFD	<ul style="list-style-type: none"> ✓構造化分析の方法を流用 ✓作業とデータの関係を表現する ✓プロセスの粒度は階層で表現する ✓CMMではDFDを想定していた 	<ul style="list-style-type: none"> ✓ストア記号とフロー上のデータが混在して成果物の状態が見えにくい ✓習得が難しい
PFD	<ul style="list-style-type: none"> ✓DFDをベースにアレンジしたもの ✓成果物と作業の関係を表す ✓プロセスの粒度は階層で表現する ✓習得しやすい ✓テーラリングしやすい 	

2.2 プロセスを表現する

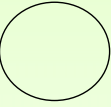

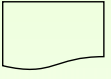
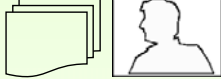



➤ プロセスを適切に表現する技術が必要

- ❖ **DFD** : 世界で使われているプロセスの表現ツール
 - 構造化分析のツールを開発プロセスの「Design」に応用
 - 日本ではこの種のダイアグラムを使わずに「プロセス定義書」だけで対応
- ❖ **PFD** : DFDでは扱いにくいところを改良して提案
 - 成果物とプロセスの関係をダイアグラムで表現したもの

ダイアグラム	成果物	プロセス	特徴
DFD	データディクショナリ	ミニスペック	「プロセスの分析」機能を「作業」に応用. 「BPR」もDFDを使用
PFD	成果物定義書	プロセス定義書	成果物の記号と下位層を持つプロセスの記号を新設

2.2 プロセスを表現する

➤ PFDで使用する基本的な記号

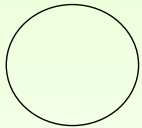
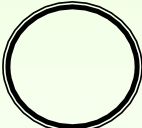
	プロセス	<ul style="list-style-type: none"> 作業を表す 階層を持つ場合には線を太くしたり二重にする 	
	成果物	<ul style="list-style-type: none"> プロセスに出入りする成果物を表す ソースファイルや、分冊の様子や、形になっていない「ノウハウ」など、適当な記号を使う 	
	フロー	<ul style="list-style-type: none"> 成果物とプロセスを繋ぐ線 両側に矢印を付けて更新の意図を表したり、線上に、成果物を構成する要素を書くこともできる 	
	トリガー	<ul style="list-style-type: none"> プロセスの起動タイミングを表現する必要があるときに使用する できるだけ使用せずに済ますこと 	

さらに詳しいことは、硬派のホームページの「PFDの書き方」を参照ください
http://homepage3.nifty.com/koha_hp/process/PFDform3.pdf

2.2 プロセスを表現する

➤ プロセスの表記

- ❖ プロセスは、1つの「○」（1重線または2重線）で表し、その中にプロセス名を記入する

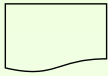
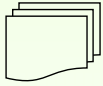
	下位層を持たない プロセス
	下位層を持つ プロセス

※プロセス名は「目的語—述語」で書く
 下層を持たないプロセス名は、**リアルにイメージできる**ように表現する

2.2 プロセスを表現する

➤ 成果物の表記

- ❖ 成果物は、「**単票**」または「**複票**」の部品で表し、その中に成果物名を記入する

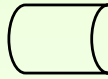

	単体で存在するもの
	複数の分冊で構成されるもの

※ 何冊も存在するものをまとめて表現する場合には「複票」を用いる
成果物の構成などは、「成果物定義」に記述する

2.2 プロセスを表現する

➤ 成果物の表記

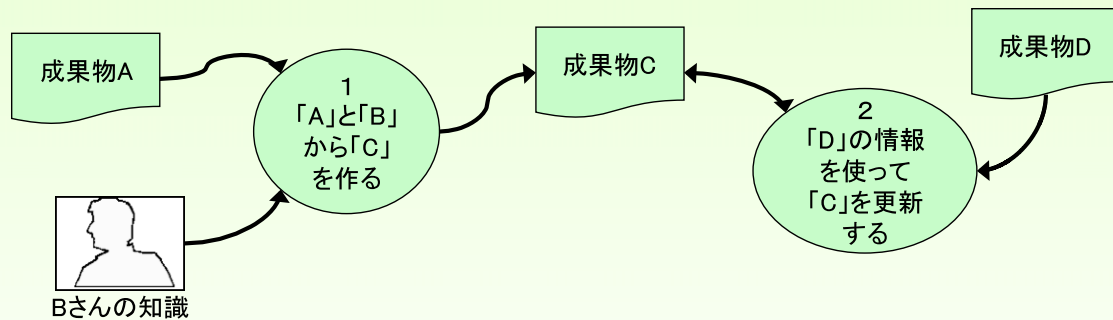
- ❖ 「**ソースファイル**」や「**人の知識/ノウハウ**」なども成果物として記述し、その中に成果物名を記入する

	ソースファイル
	人の知識/ノウハウ

2.2 プロセスを表現する

➤ 成果物とプロセスをフローでつなぐ

- ❖ 「フロー」によって成果物とプロセスをつなぐ

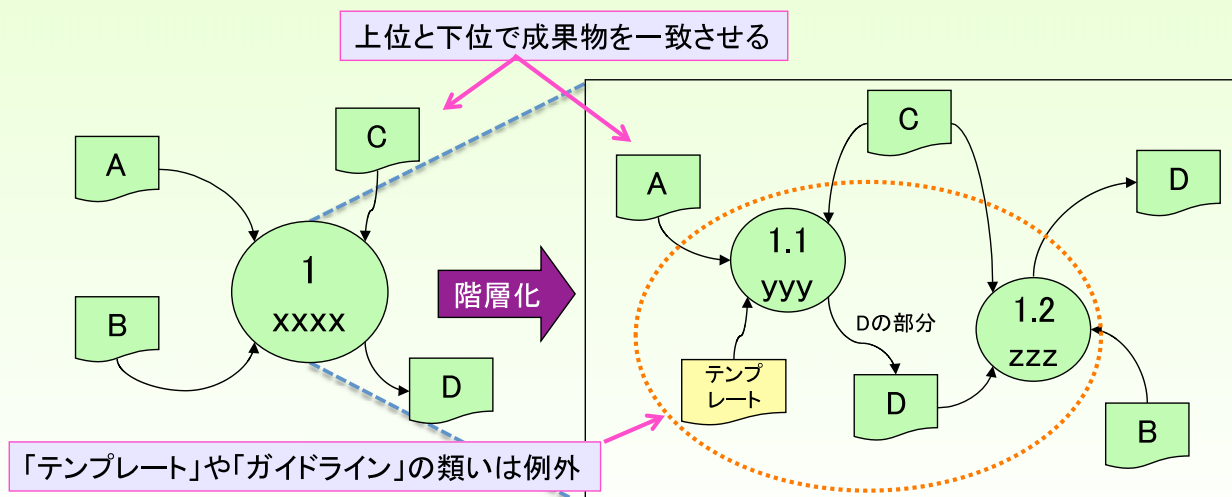


※ 生成や参照の詳しいことは、「プロセス定義」に記述する

2.2 プロセスを表現する

➤ プロセスの階層化

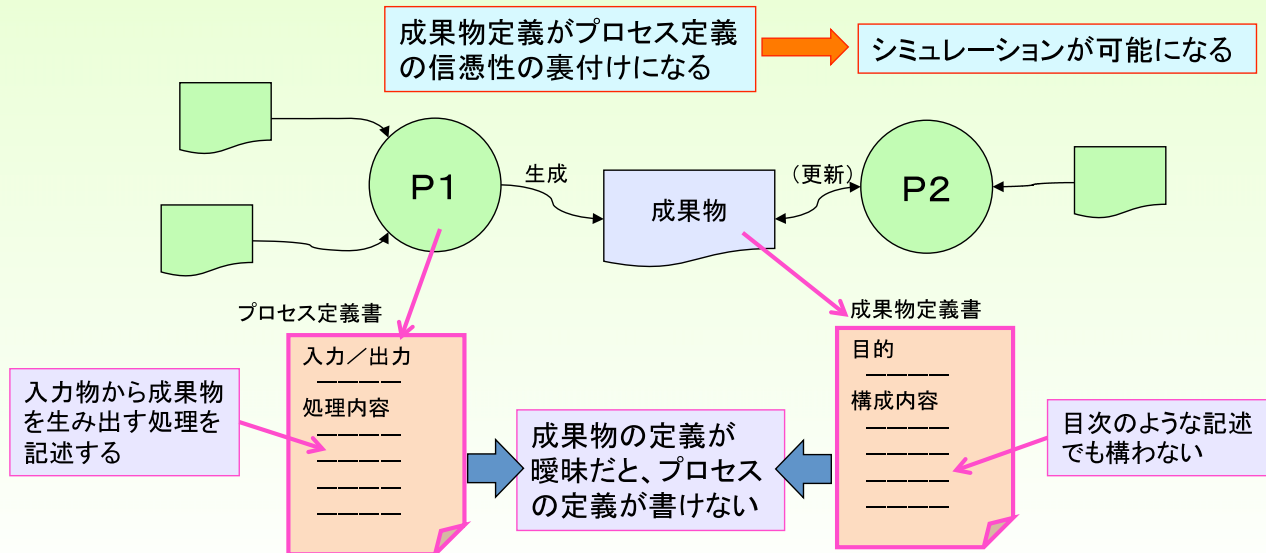
- ❖ 1枚のシートに必要な以上に多くのプロセスを配置しない
「7±2」個を目安として階層化する



2.2 プロセスを表現する

➤ 成果物とプロセスの定義

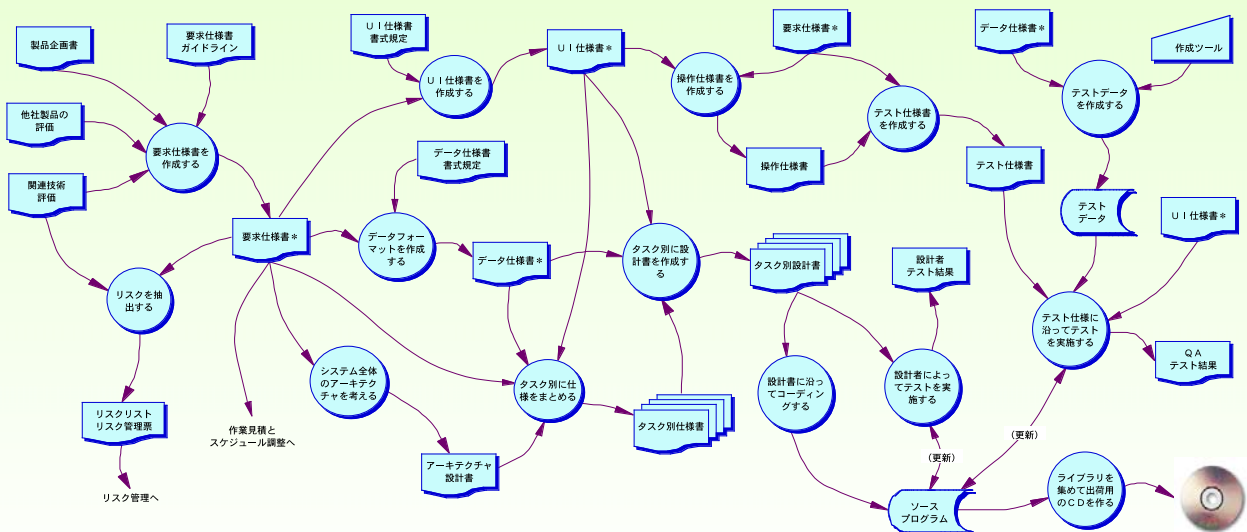
❖ PFDの実行性は、成果物とプロセスの「定義書」で裏付けられる



2.2 プロセスを表現する

➤ PFDのサンプル

❖ 「一般的な、新規開発の最上位層 (PFD-0) のパターン」



2.2 プロセスを表現する

➤ 成果物定義のサンプル

❖ レイアウトは以下のフォームを推奨

「成果物定義書」は、組織内で再利用しやすいので、きちんと書いておくと良い

成果物定義書			作成日
文書番号	成果物名・ファイル名		2003/1/29
D10	変更依頼書		
成果物の目的	変更の依頼内容を記述したもの		
編集方針	依頼者が希望する変更したい仕様の内容や追加機能の概要を箇条書きにしたもの 事前に何度か打ち合わせをしているが「USDM」の形式にはなっていない		
成果物の構成と簡単内容			
	1	機能追加の要求 = [追加して欲しい機能名 + 当該機能の動作概略 + 1 [特に実現して欲しい動作および仕様]n]	
	2	変更の要求 = [変更対象の機能名 / * 変更がある仕様がある機能名。ベースの機能仕様書の構成順に並べる * / + 1 [変更して欲しい項目 / * 機能仕様書に書かれている仕様に対する変更 * / + 削除して欲しい項目 / * 当該機能から削除して欲しい仕様 * / + 追加して欲しい項目]n] / * 当該機能に付加して欲しい仕様 * /	

構成を表す番号。成果物の「枝番号」はこの番号を使用する

2.2 プロセスを表現する

➤ プロセス定義のサンプル

❖ レイアウトは以下のフォームを推奨

プロセス定義書

プロセス番号	プロセス名		作成日
P1.1	ベースの設計書を調べて調査項目(機能)をリストアップする		
入力情報	D10 D3	変更依頼書 ベースのタスク設計書	
出力情報	D11	調査資料	
作業内容			作業担当者
プロセス実施条件:	変更依頼書(D10)が届いた時点 ベースのタスク設計書などは事前に確保しておく		関数仕様書 と同じイメージ
必要スキル:			
担当者:			
作業内容:	1	調査する機能をリストアップする 今回の変更依頼[D10]の内容から、関係する機能およびサブ機能を調査資料[D11]にリストアップする	
	2	タスク設計書の該当箇所を追記する タスク設計書[D3]から該当箇所を探し、参考になる箇所を調査資料[D11]に追記する この時、資料として不十分なことが判明したときは調査資料[D11]の調査範囲に追記する	
	3	調査に必要な項目を見積もって追記する 調査範囲に記述された項目から、調査に必要な工数を見積もって調査資料[D11]に追記する	
プロセス終了判定:	変更依頼の内容に対して、関連資料が把握できたり、調査が必要な項目が拾い出された状態		

3. プロセスの設計

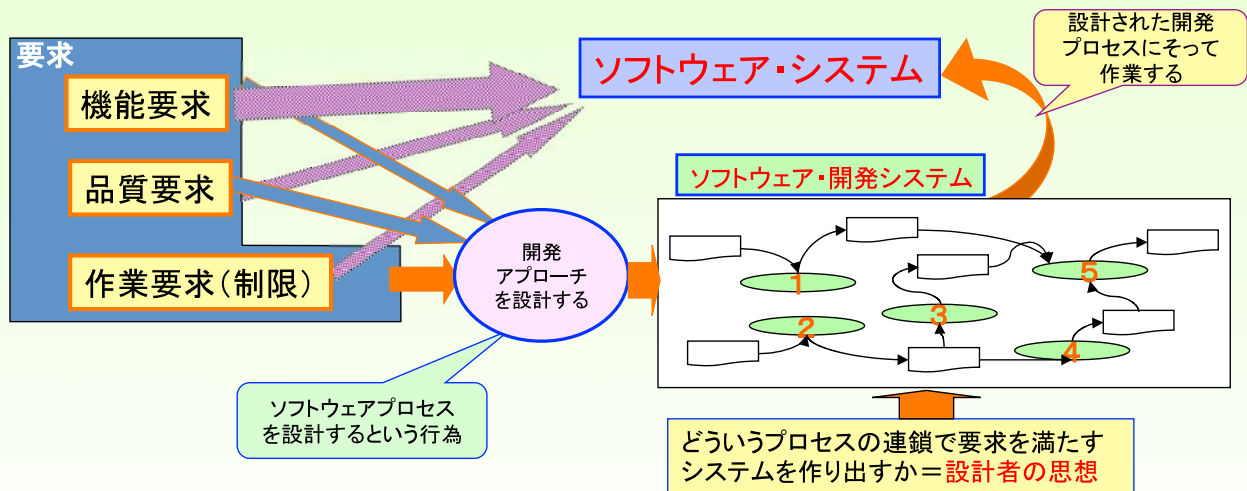
- 3.1 派生開発におけるプロセス設計の必要性
- 3.2 プロセスを定義するから設計するへ
- 3.3 関係を表すだけ
- 3.4 ゴールから考える
- 3.5 プロセスをシミュレーションする
- 3.6 良いプロセスであるか評価する

3.1 派生開発におけるプロセス設計の必要性

- 派生開発こそ合理的なプロセスの設計が必要
 - ❖ 要求や規模などの要因が**多様**なため、合理的なプロセスが不可欠
 - ❖ **期間が短く**やり直しが効かない
- 短納期で結果の確認が短期間でできる
 - ❖ 設計したプロセスの善し悪しが短期間に検証できる

3.2 プロセスを定義するから設計するへ

- 今回の要求を満たすソフトウェア・システムを作り出すためのプロセス（開発アプローチ）を「設計する」



3.3 関係を表すだけ

- PFDは、成果物とプロセスの“関係”を表すだけで、“順序”を表現しない

- ❖ その成果物（の内容）が

- ① どのプロセスから生み出され
- ② どのプロセスで活用されるかを表現するだけ

- ✓ 一つの成果物は、一つのプロセスから作り出されるとは限らない
 - 最初に1つのプロセスで生成された後は、いくつかのプロセスによって内容が追加されたり、更新されることがある
 - その様子を正確に表現することが大事
- ✓ 一つの成果物は、そのあと複数のプロセスで使われることがある
 - それぞれのプロセスで、この成果物のどの部分を使うのかを明示する

順序は

- スケジュールで表せばよい
- 状況によって途中で変化させる

3.4 ゴールから考える

- ゴールから考えることで、**無駄のない開発アプローチ**を設計する
 - ❖ 「その結果を得るには、どのような（入力）成果物があればよいのか」
 - ❖ 「ダイクストラの導出法」
- 前から考えると「これとこれのできることをベースにプロセスが作られる」
 - ❖ 結果として、**無駄なプロセス**が組み込まれる
 - ❖ いったん組み込まれたプロセスは「必要性」を主張する

無駄なプロセスに見えない

3.5 プロセスをシミュレーションする

- PFDでプロセスを表現できれば完成というわけではない
 - ❖ PFDは最適なプロセス(と成果物)を**設計**するためのツール
- ① 「**設計**」したプロセスを**シミュレーション**で「**安定**」させる
 - 最終成果物を確実に作り出すことができるか
 - 全く使われないもしくは1度しか使われない成果物は無いか
 - 機能的合理性と経済的合理性を確認する
- ② 身に染み込んだ**従来の習慣**を**シミュレーション**で**薄める**
 - これによって、本番では「新しいプロセス」を迷いなく実施できる

これで市場の変化に対応できる

※ 自分が設計したプロセスに基づいて見積もり、さらにそこからスケジュールを立てる

3.6 良いプロセスであるか評価する

合理的であること

❖ 要求を実現するために「**機能的合理性**」と「**経済的合理性**」を確保する

機能的合理性	成果物の内容が狙い通りに生成される仕組みがある	品質
	入力物が生成されるタイミングに食い違いがない	
	プロセスが適切なタイミングで実施できる	
経済的合理性	生成物が過大ではない（見積りと組み合わせる）	生産性
	無駄なプロセスを経由していない	
	生成プロセスの工数が回収できる	

- 2つの合理性を追求しなければ・・・
 - 不足を補うプロセスが吟味されないまま実施され、工数の逸失に繋がる
 - 無駄なプロセスによって工数を失いプロセスを省くことになる

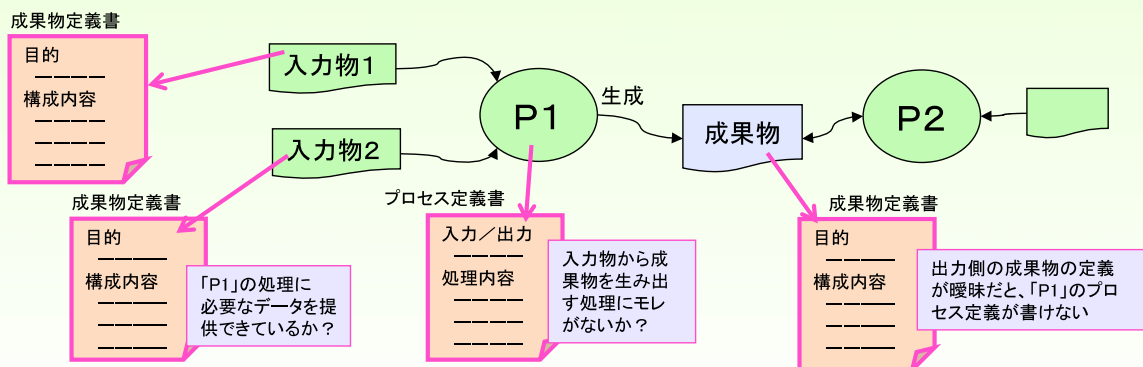
バグや納期遅延に繋がる！

3.6 良いプロセスであるか評価する

機能的合理性が確保されているか？

- ❖ このプロセスの連鎖で、**目的の成果物**が作りだせることを確認する
- ✓ ダイアグラム上で不足している成果物はないか？
 - ✓ この入力物が遅れる要素は？
 - ✓ そのときにこのプロセスはどうする？
 - ✓ 必要に応じて成果物定義書で**構成**を確認する

シミュレーションされていること！



3.6 良いプロセスであるか評価する

経済的合理性が確保されているか？

- ❖ このプロセスの連鎖で、**無理のない工数**で成果物が作れるか？
 - ✓ サイズ見積もりなどを考慮した後で評価する
 - ✓ 機能的には生成関係は成立していても、必要以上にボリュームが大きい？
 - ✓ そのプロセスの生産性が悪く、多くの工数を必要としないか？

