PFDってなに？

AFFORDD カンファレンス2014
チュートリアル資料

梶本 和博
派生開発推進協議会
URL=http://www.xddp.jp
（株） エクスモーション
URL:http://www.exmotion.co.jp
kazuhiro.kajimoto@exmotion.co.jp

はじめに

- ＰＦＤって知ってますか、聞いたことは有りますか
- Process Flow Diagramの頭文字を取ったものです。
- ＰＤＦと間違いやすい（？）ので、気をつけてください。

自己紹介

梶本 和博（かじもと かずひろ）

＜所属＞：派生開発推進協議会
　　株式会社エクスモーション
＜業務＞：ソフトウエアプロセス改善コンサルタント
＜略歴＞：

- 1975:制御ソフト設計
  オンライン銀行ターミナル/オフスコンピュータ
- 1984:組み込みソフト設計
  ページププリンタ/スケーラブルフォント
- 1989:プリンタ商品企画
  オンデマンドパブリッシング向けプリンタ
- 1995:ソフトウエア品質/生産技術
  ＵＳＤＭ/ＸＤＤＰ/ＰＦＤ/ＳＩＯ９０００/CMM/CMMＩ
- 2010:派生開発協議会
- 2012:㈱エクスモーション

アジェンダ

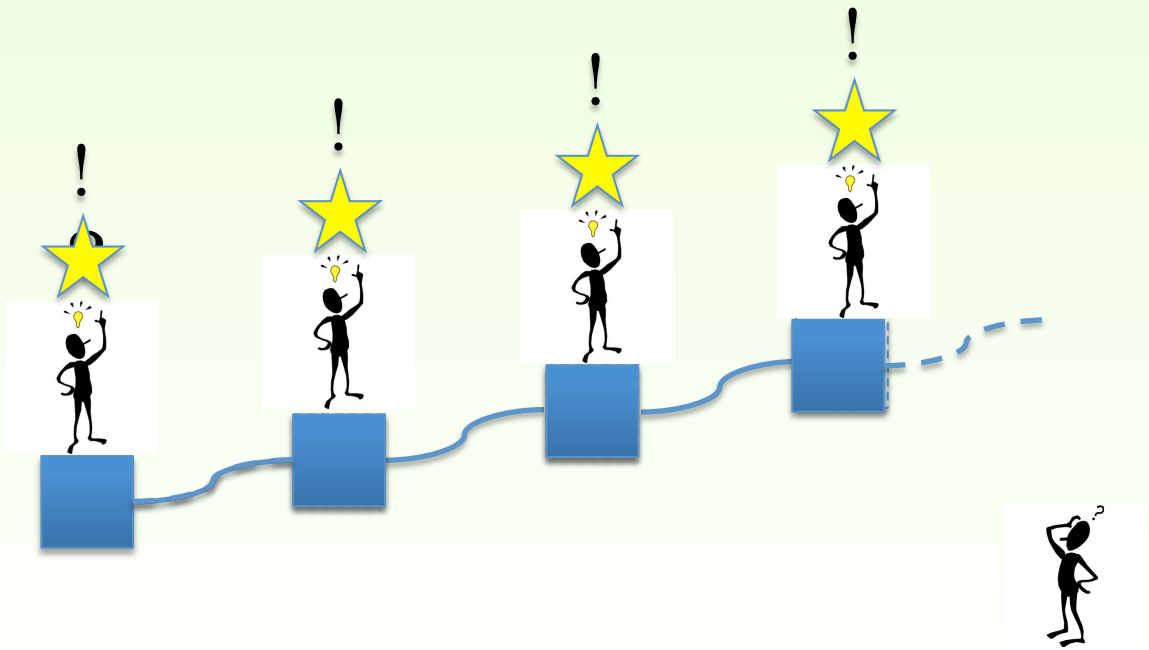
1. 開発現場の現状
2. そのように為らないために
3. ＰＦＤでプロセスを表現する
4. どんなことに使える

１．開発現場の現状

１．１混乱するプロジェクト

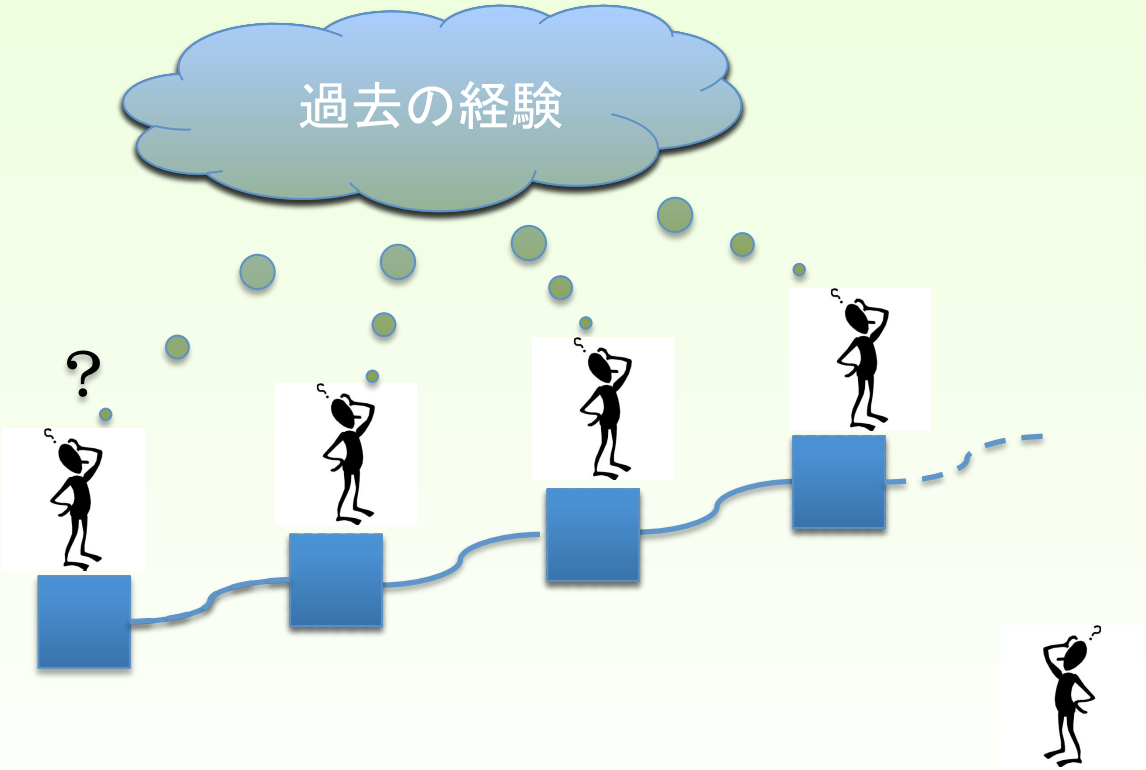
１．２遅延するプロジェクト

- こんな状態に為っていませんか！
  - 作業は、プロジェクトの進展に従って都度（場当たり的：閃き？）指示が出る
  - 今までの習慣で実施している



1.1 混乱するプロジェクト

- こんな状態に為っていませんか！
  - 作業は、プロジェクトの進展に従って都度決定
  - 過去の経験の良し悪しで結果の成否が異なる

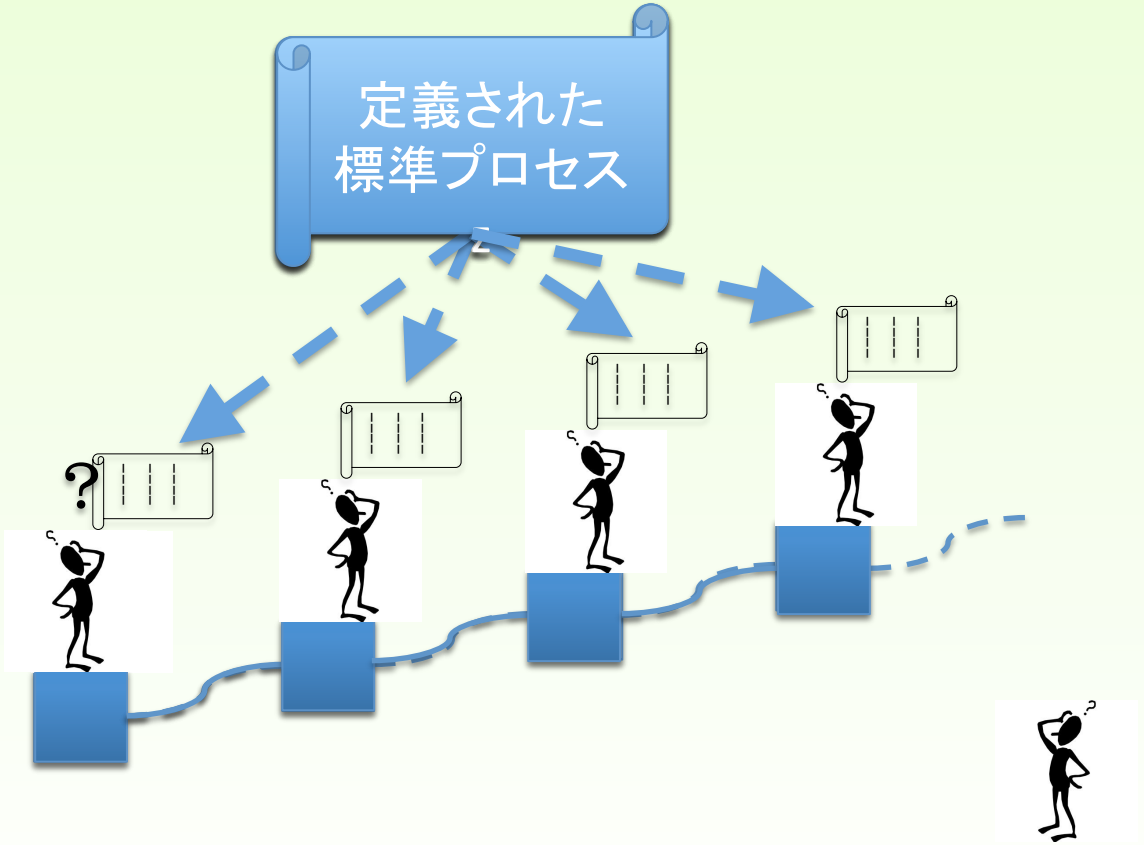


1.1 混乱するプロジェクト

- 今回のプロジェクトで作成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
  - リーダーによる都度指示で実施（リーダー任せ？）
  - 今までの習慣で実施

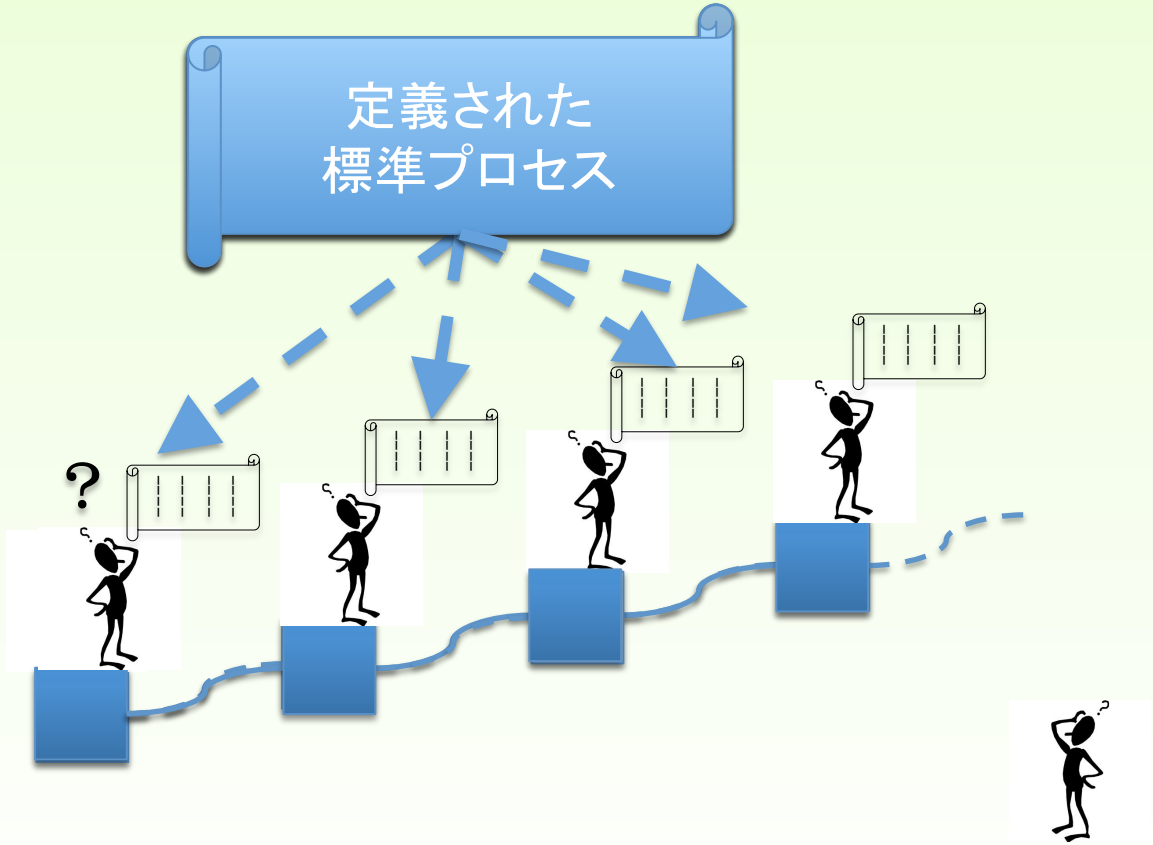
1.2 遅延するプロジェクト

- こんなことに為っていませんか！
  - 組織には標準プロセスがある
  - それに従ってプロジェクトを実施している



1.2 遅延するプロジェクト

- こんなことに為っていませんか！
  - 標準プロセスに添って実施している
  - 多くの成果物（/作業）を作らねばならない



1.2 遅延するプロジェクト

- 組織で策定され、今まで実施されてきたプロセスをそのまま実施している
  - ある時大成功を納めた方法を基に策定した方法なのに・・・
  - 策定しなければならなくて策定した方法なのに・・・
- 万能プロセスを作ってしまった
  - 策定したプロセスの粒度が粗く、不要なプロセスを含む
- 問題が生じる度に成果物が増加/肥大し作業が増える
  - 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．そのように為らないために

２．１混乱しないためには

２．２遅延しないためには

２．３こうなれば良い

- 何をすればよいのか観えている
  - 今回のプロジェクトで作成しなければ成らない成果物が観えている
  - 今回のプロジェクトで実施しなければ成らない作業が観えている

- 要求に適した作業となっている
  - 今回作成するものは、内容の重複や使われないものは含まれていない
  - 今回作成するものは、今回の要求を達成するのに必要なもの

2.3 そのために何が出来ればよいの

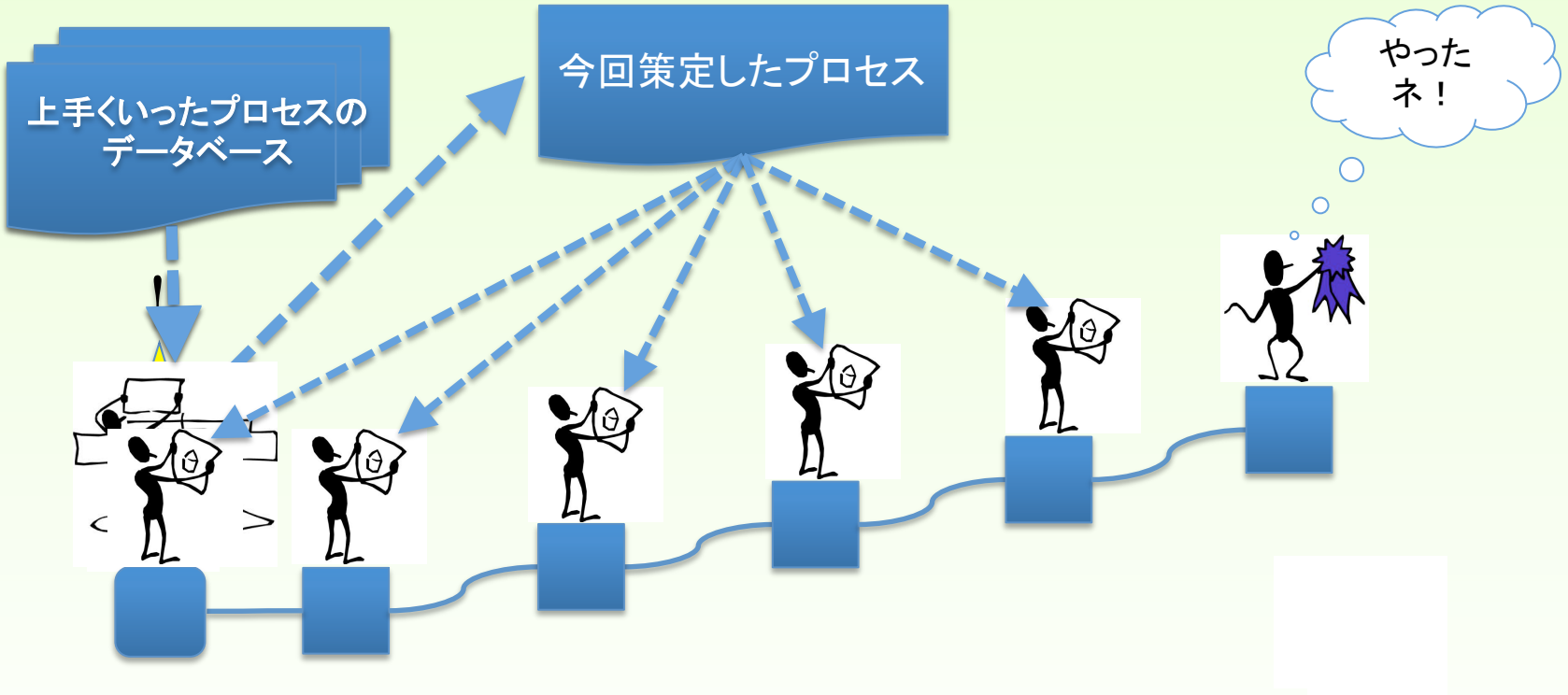
- 今回作成する成果物とそのための作業の全貌が表現されたものが有る
- それらの成果物やそれを作り出す作業は、今回の要求を実現するのに必要なものだけで構成されている
- これらをの事柄が関係者に周知され議論され承認されている

これらを実現できる1つの方法として考案された

↓

それが“PFD (Process Flow Diagram)”なんです

- このように実施出来るとGoodですね！



３．PFDでプロセスを表現する

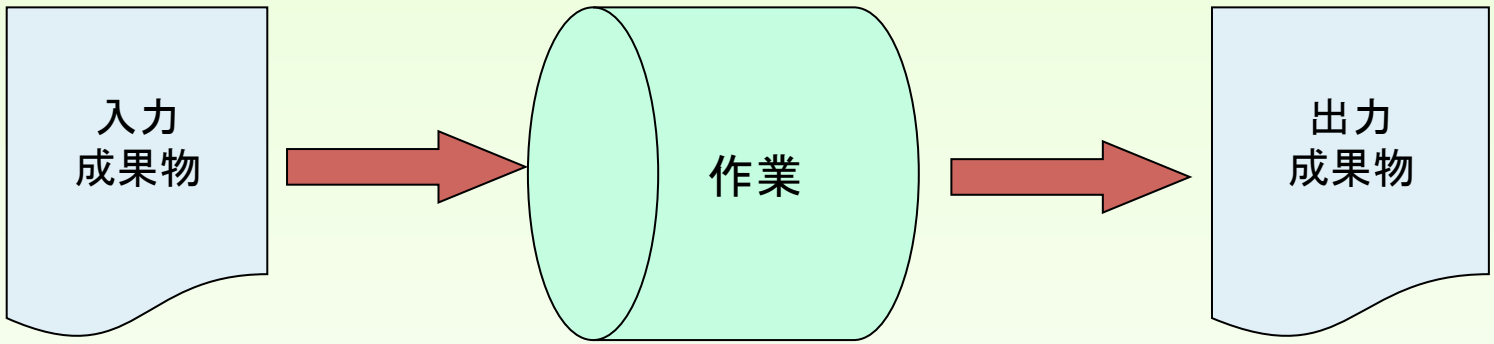
３．１　プロセスとは

３．２　プロセスを表現する

３．３　プロセスを設計する

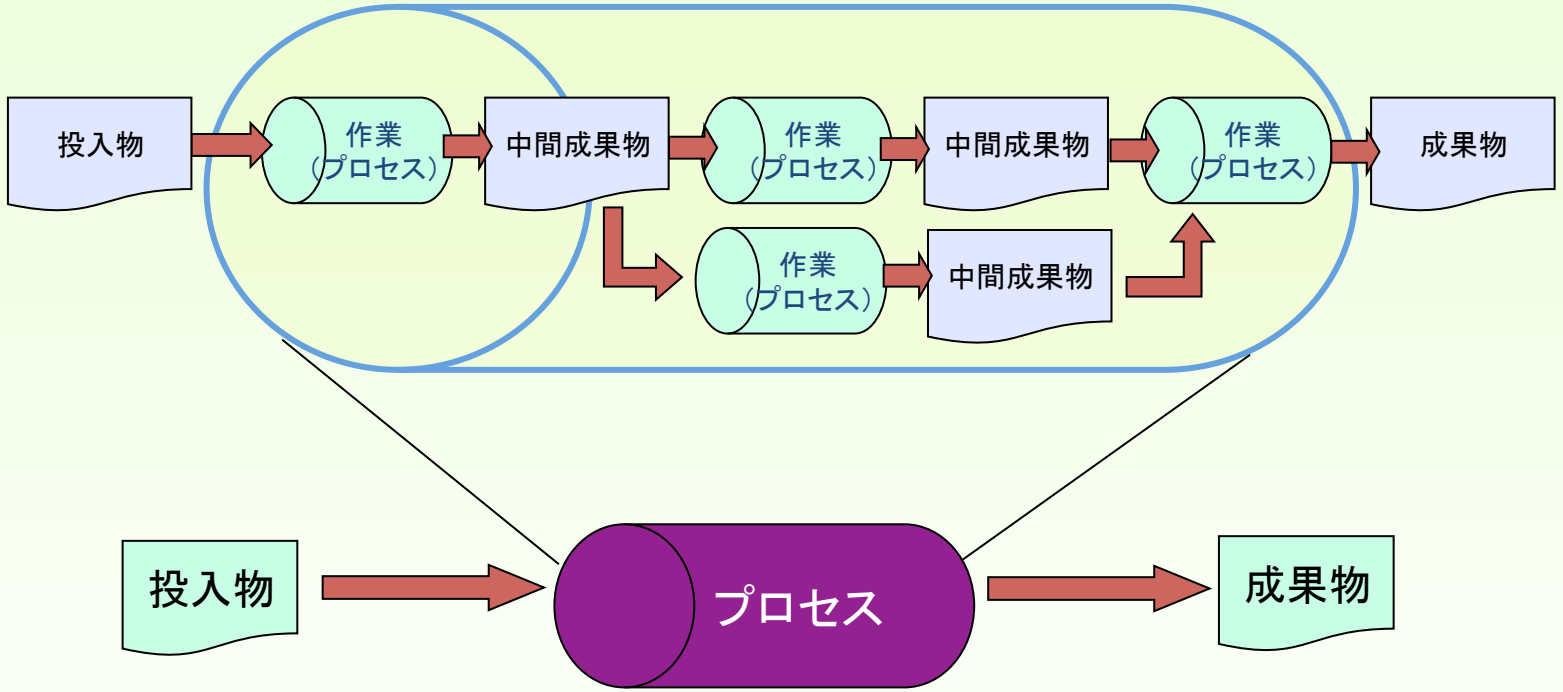
３．４　失敗しないために

- 入力成果物を出力成果物に変化させる行為/仕掛



3.1 プロセスとは

- 成果物と成果物はプロセスを介して連鎖している



プロセスを表現する方法

表現方法	特徴	問題
ワークフロー	✓ 作業の流れがわかりやすい ✓ 日本では広く使われている？	✓ 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	✓ 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	
プロセス定義書（定義書のみ）	✓ プロセスに対して詳細に手順を定義している	✓ ダイヤグラム表現が無いためプロセス間の関係が見えない
DFD	✓ 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	✓ 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	✓ 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	

3.2 プロセスを表現する

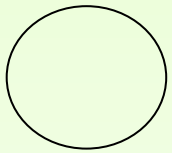
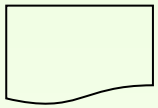
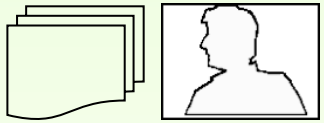
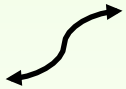
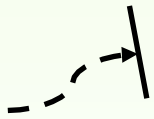
- プロセスを適切に表現する技術が必要
  - **ＤＦＤ**：世界で使われているプロセスの表現ツール
    - 構造化分析のツールを開発プロセスの「Design」に応用
    - 日本ではこの種のダイヤグラムを使わずに「プロセス定義書」だけで対応
  - ***ＰＦＤ***：DFDでは扱いにくいところを改良して提案
    - 成果物とプロセスの関係をダイアグラムで表現したもの

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

– この他の多くのルールはDFDのルールを継承

➢ ＰＦＤで使用する基本的な記号

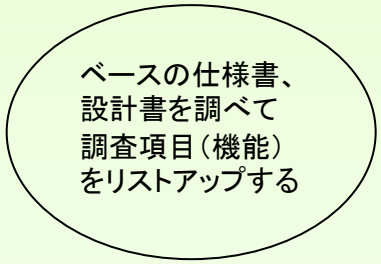
❖PFDで使用する記号は次のものが有る

	プロセス	・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	・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	・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	・プロセスの起動タイミングを表現する必要が生じたときに使用する ・できるだけ使用せずに済ますこと	

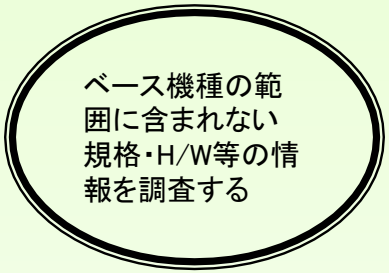
さらに詳しいことは、硬派のホームページの「PFDの書き方」を参照してください
（http://homepage3.nifty.com/koha hp/process/PFDform3.pdf）

プロセスの表現

- 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

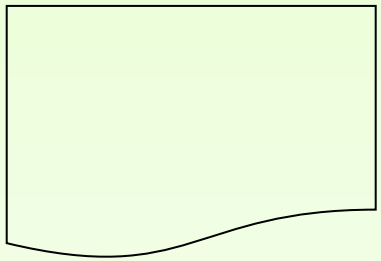


下位層を持つ
プロセス

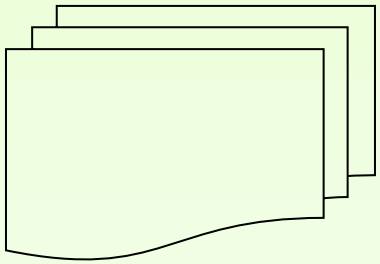
- プロセス名は「目的語一述語」で書く
- １重線のプロセスは、実際に作業を行うプロセスであり、できるだけ、作業の範囲や入力物が出力物に変換される様子がリアルにイメージできるように表現する
- 「踏査の抽出」のような表現では、具体的な行動（作業）がイメージしにくいので避ける
- プロセスにおける具体的な作業は「プロセス定義」で記述する

➢ 成果物の表現

- ❖ 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

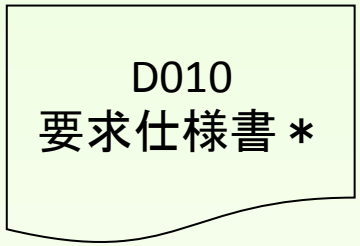


複数の分冊で構成されるもの

- ❖ タスク設計書のように、何冊も存在するものをまとめて表現する場合は「複票」で表し、具体的なイメージを誘う
- ❖ 成果物の構成などは、別に「成果物定義」で記述する

➢ 成果物の複製表示

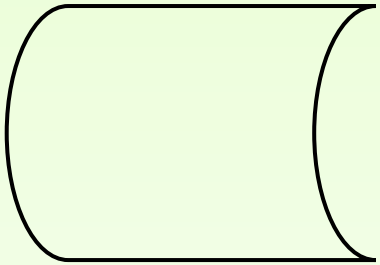
- ❖ １つのPFDの場面の中で、ある成果物が複数のプロセスに関係している場合、出来るだけフローをのばして接続し記述づる
- ❖ ただし、関係箇所の距離が離れている場合や、他のフローと交差してPFDの可読性が悪くなる場合は、成果物名または番号の横に「＊」をつける「複製表示」の方法で対応する



- ❖「複製表示」は、１つのPFD上に１つの成果物が他にも表示されていることを示す
- ❖PFDの層が異なれば複製表示の必要はない

➢ 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

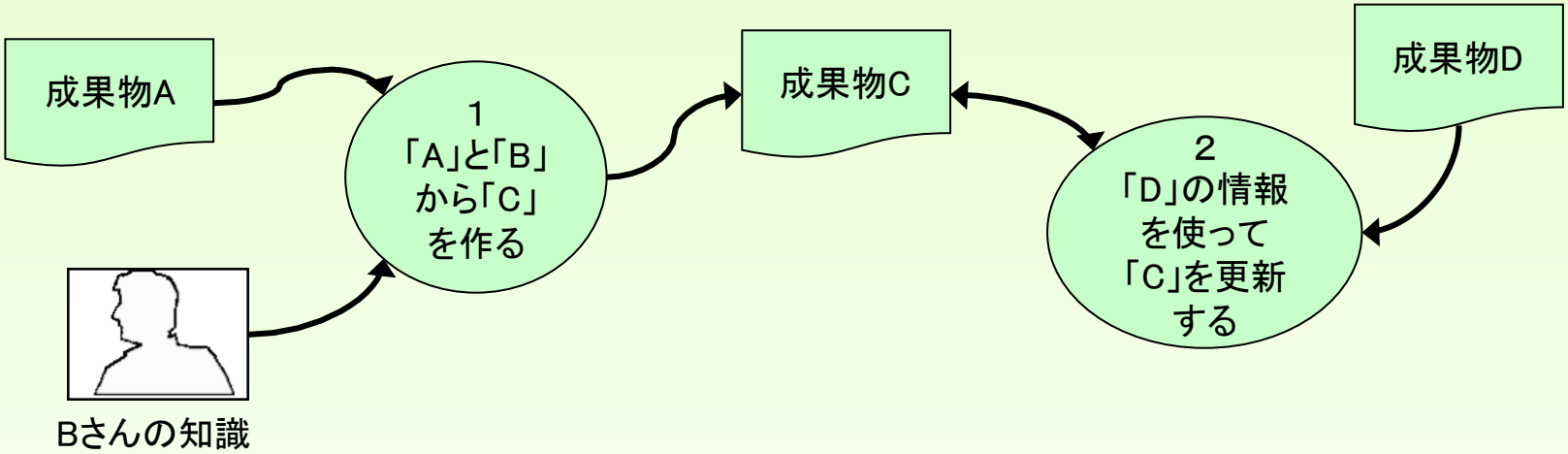


ソースファイル

人の知識/ノウハウ

➢ 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



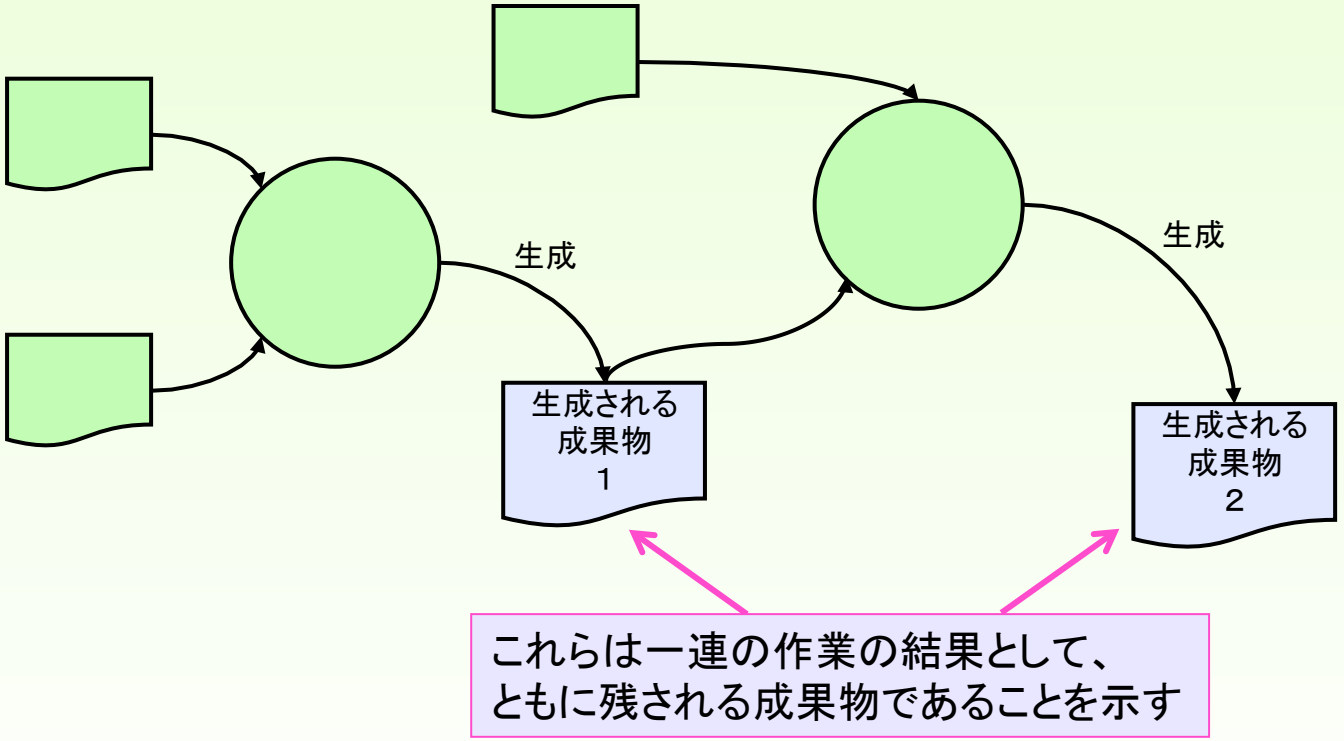
- ❖成果物「C」が、どの成果物から付く足れているか
- ❖成果物「C」が、どのプロセスで活用されているか

} が一目でわかる

- ❖生成や参照の詳しいことは、「プロセス定義」に記述する

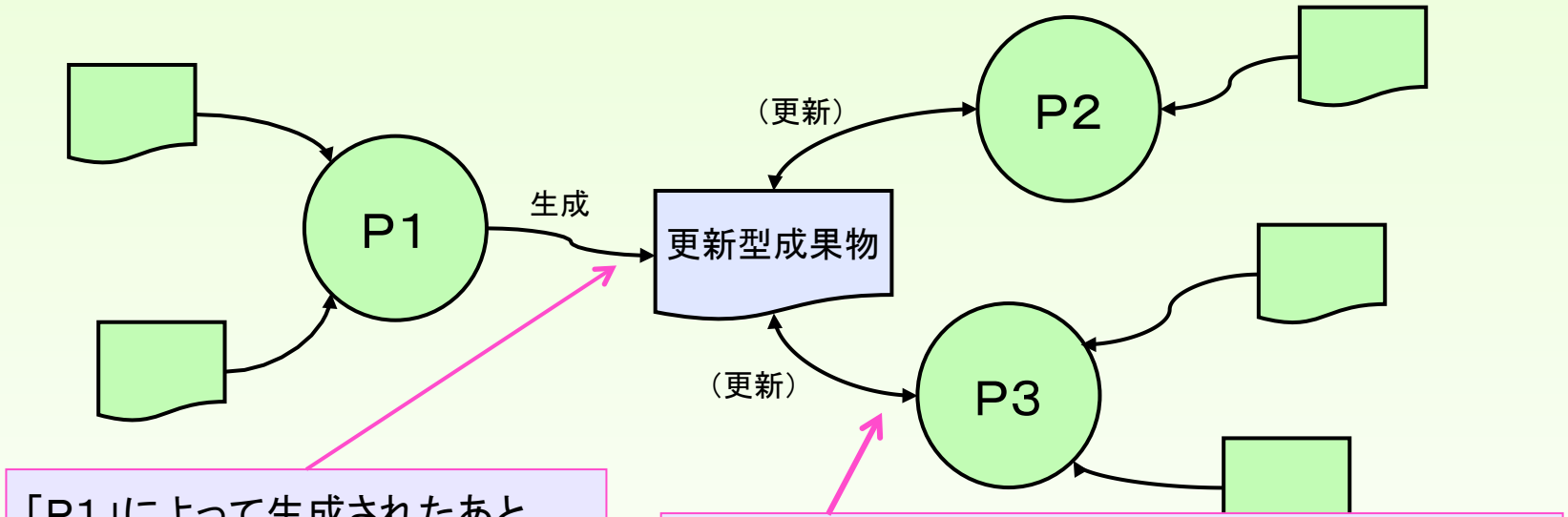
➢ 生成型と更新型（１）

- ❖生成型成果物とは、あるプロセスから新規に生み出される成果物
  - ❖生成された成果物は、通常は以降のプロセスの入力に使われる



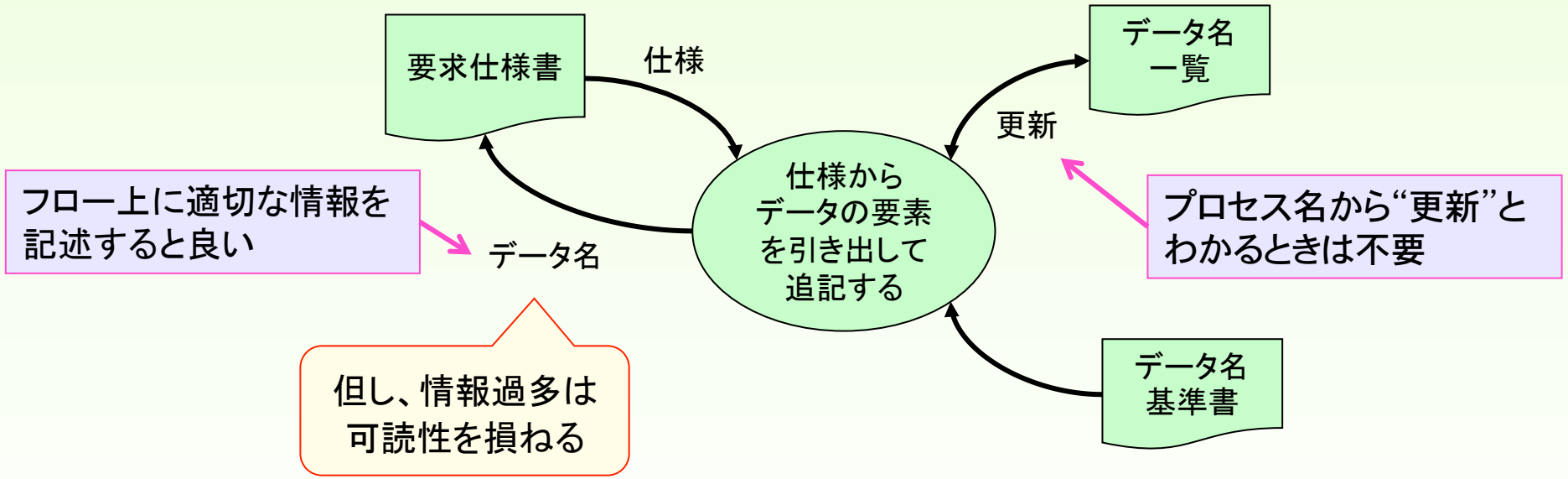
➢ 生成型と更新型（２）

❖更新型成果物とは、一度「生成」された成果物が、その後に他のプロセス から内容が「追記」されたり、「更新」されたりする成果物



➢ 往復フローと２本の単方向フローの使い分け

- ❖成果物があるプロセスによって更新される場合、通常は「更新型」のフローを使って表現する
- ❖成果物から読み出す情報や書き出す内容を意図的にフロー上で見せたい場合は、「更新型」のフローを使わずに入力と出力の二本のフロー を使う

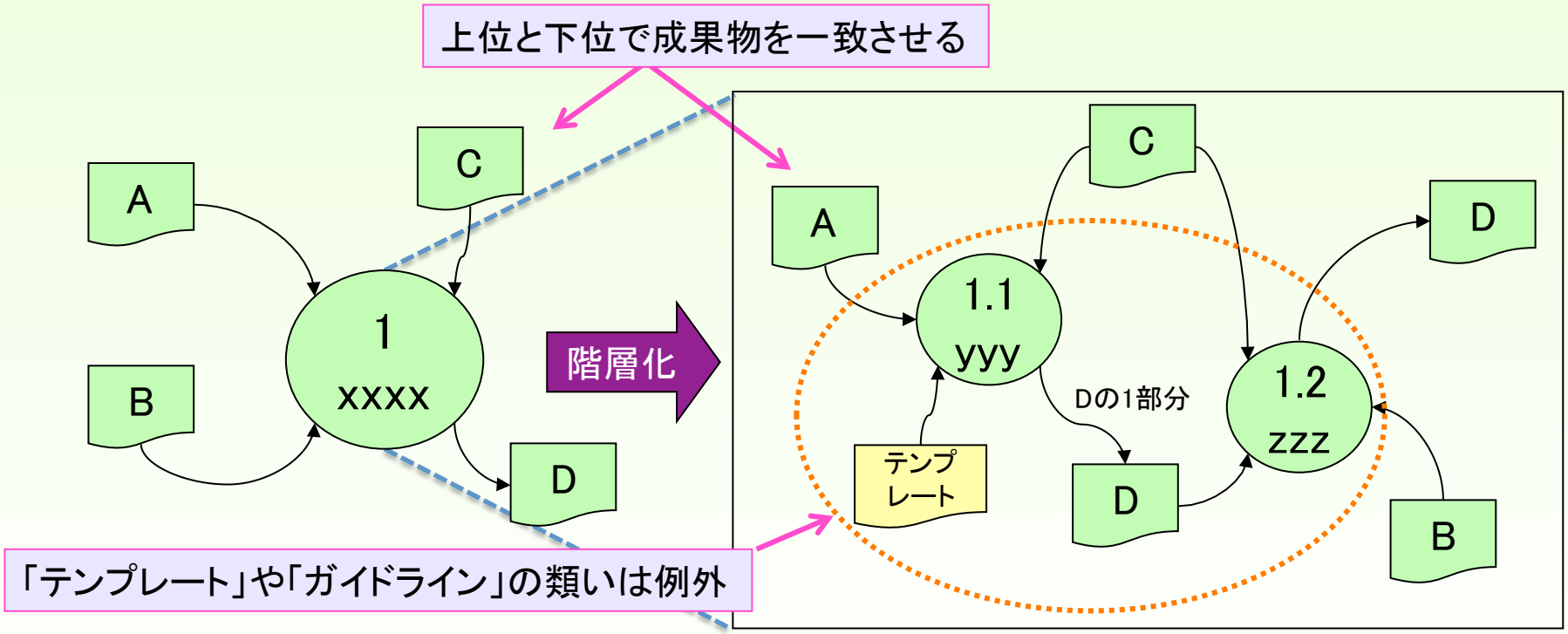


➢ プロセスの階層化　（１）

- ❖１枚のシートに必要以上に多くのプロセスを配置しない
  - ❖「７±２」個を目処として、できるだけ階層化する
  - ❖下位層を持つプロセスは二重線で書く
- ❖最上位のPFDにおいて全面的に階層化を活用することで、最上位のPFD は「組織標準」の候補として使うことができる
  - ❖ただし、そのまま「組織標準」にしない
- ❖階層の上下間では、「親子間のバランス」を確保する
  - ❖上位のプロセスに接する成果物は、下位層のPFDの生成関係と一致する

➢ プロセスの階層化　（２）

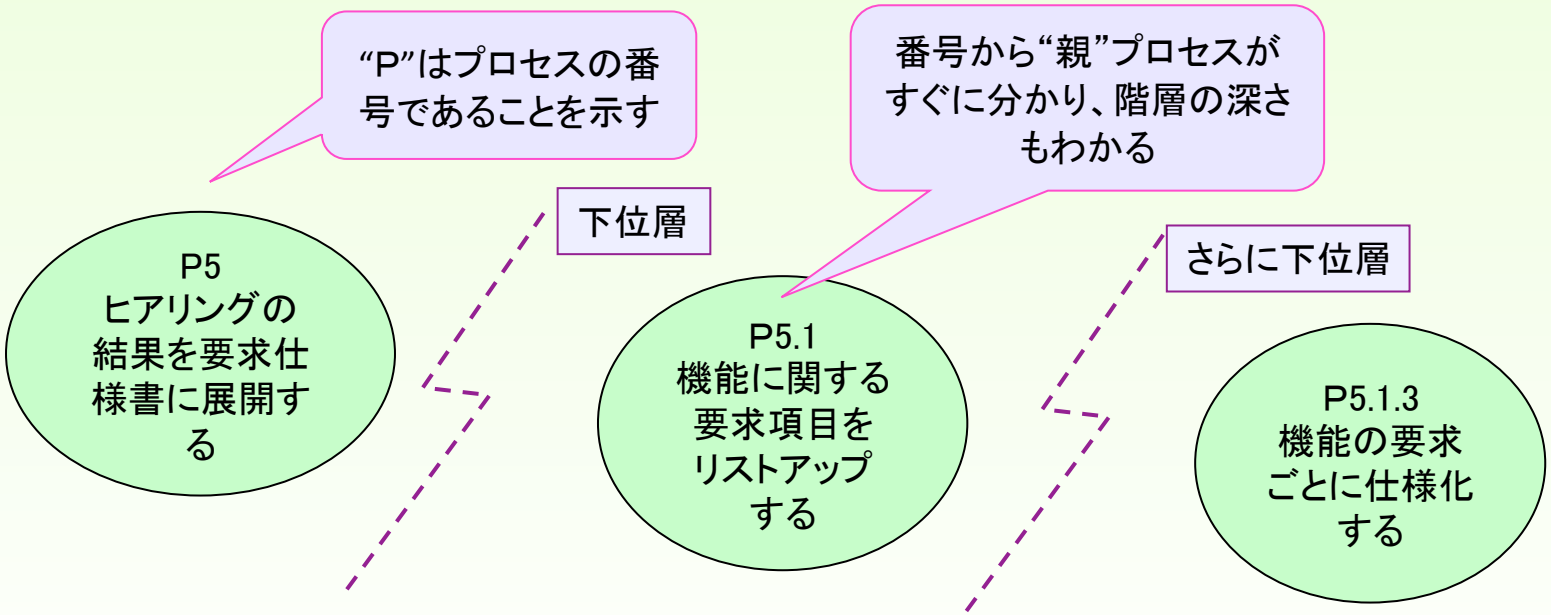
- ❖ 階層の上下間では、「親子間のバランス」を確保する
  - ❖プロセスの「番号」によって階層の様子（深さ）が見える
  - ❖成果物が一致しないと、どちらが正しいのかわからなくなる



➢ プロセスに番号を付ける

❖ プロセスには「番号」を付ける

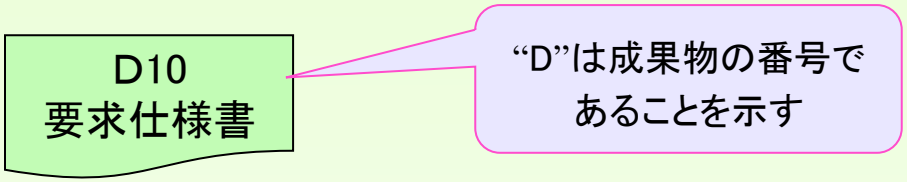
- ❖プロセス番号は単なる識別子であって、実行の順序を示すものではない
- ❖番号が飛んでもかまわない
- ❖番号は「階層」を表現する



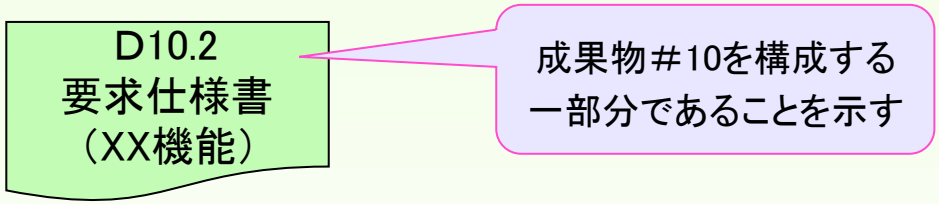
➢ 成果物にも番号を付ける

❖ 成果物にも「番号」を付けるとよい

❖生成する成果物はすべて最上位層に現れる



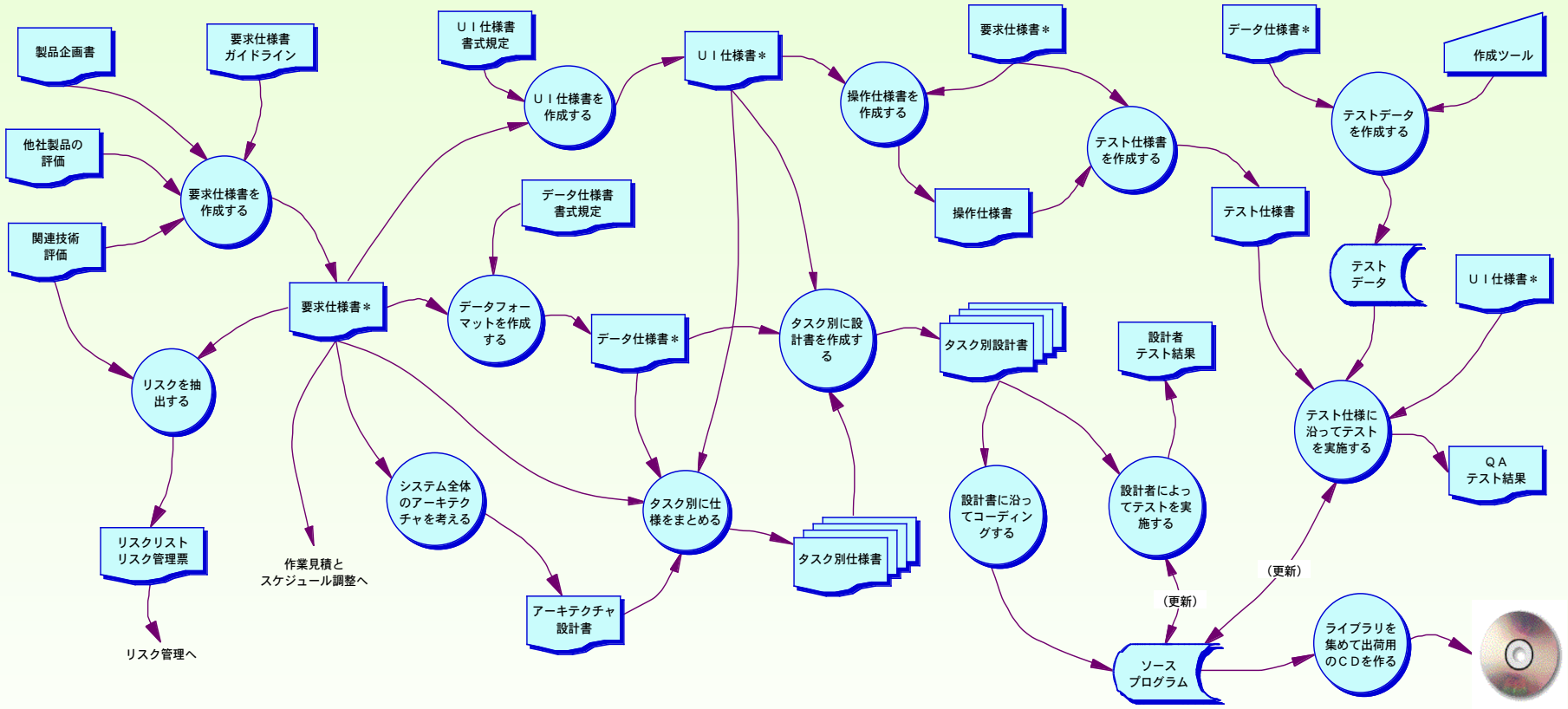
❖成果物の番号もPFDの「階層」関係の中で使用するときに役に立つ



❖ただし成果物定義書において当該成果物の構成がきちんと定義されている
ことが前提となる

3.2 プロセスを表現する

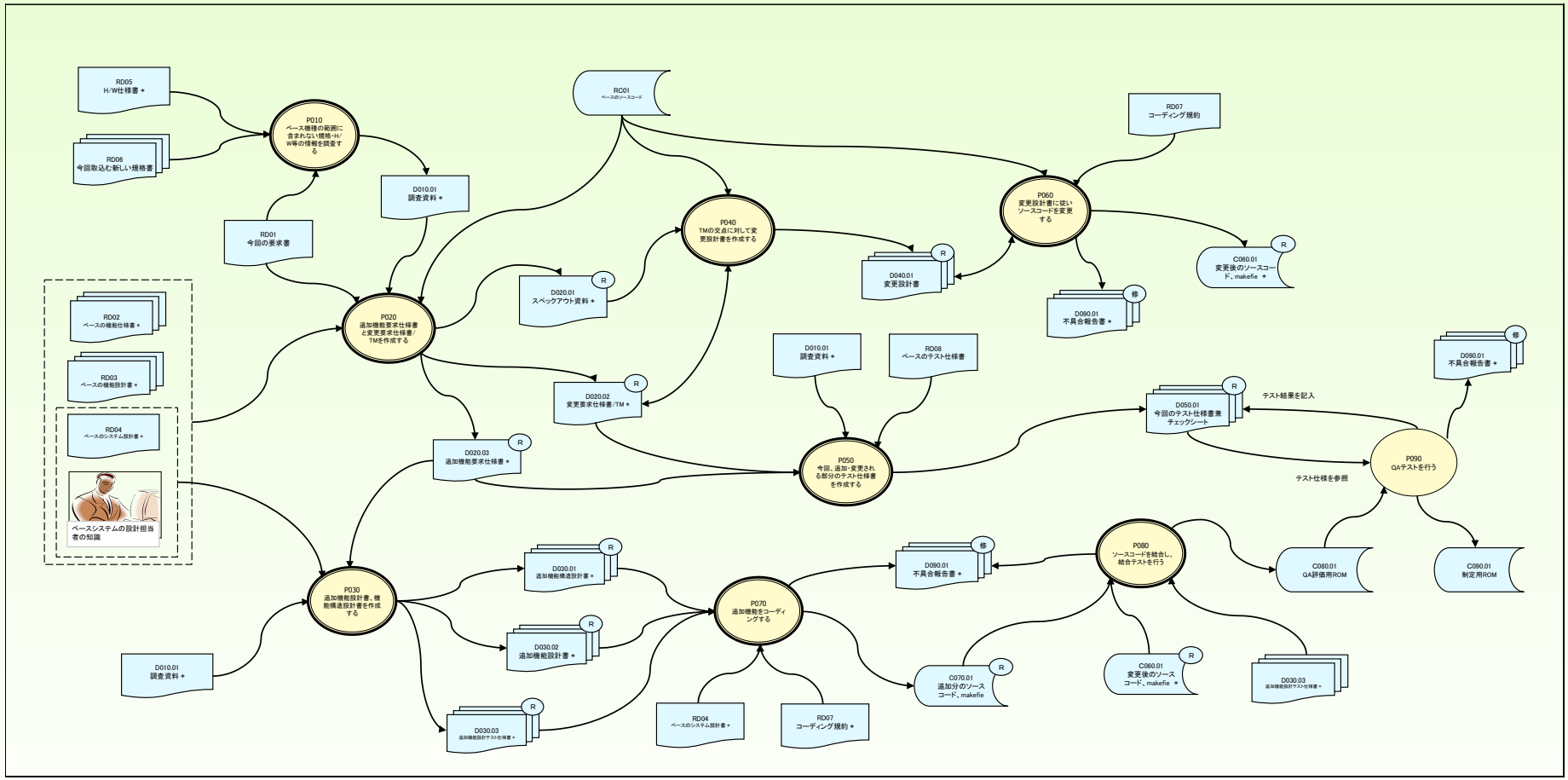
- ＰＦＤによるプロセスの表現(サンプル１)
  - 一般的な新規開発の最上位層(PFD-0)のパターン



3.2 プロセスを表現する

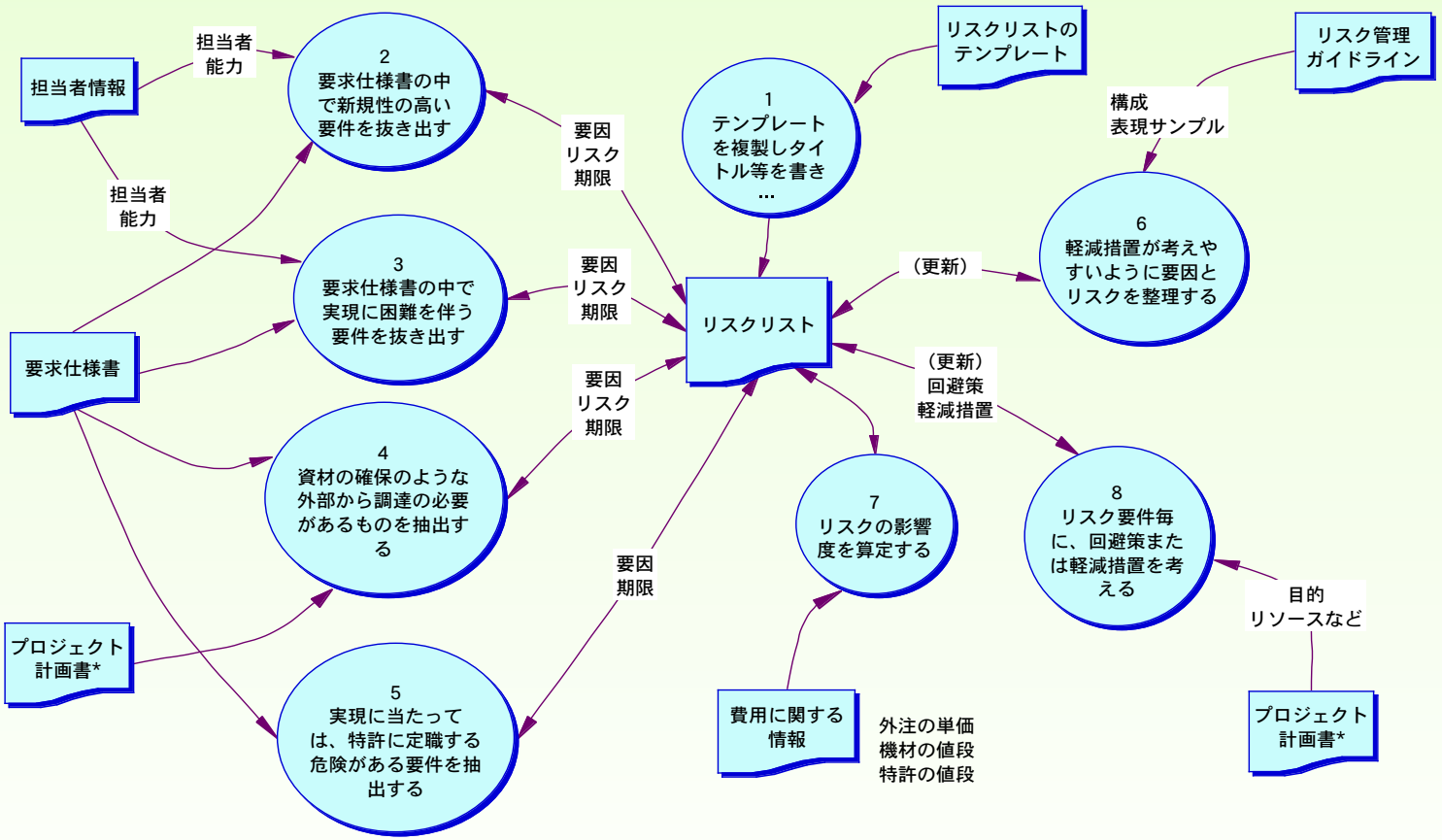
➢ ＰＦＤによるプロセスの表現（サンプル２）

❖ある派生開発の最上位層（PFD-0）のパターン



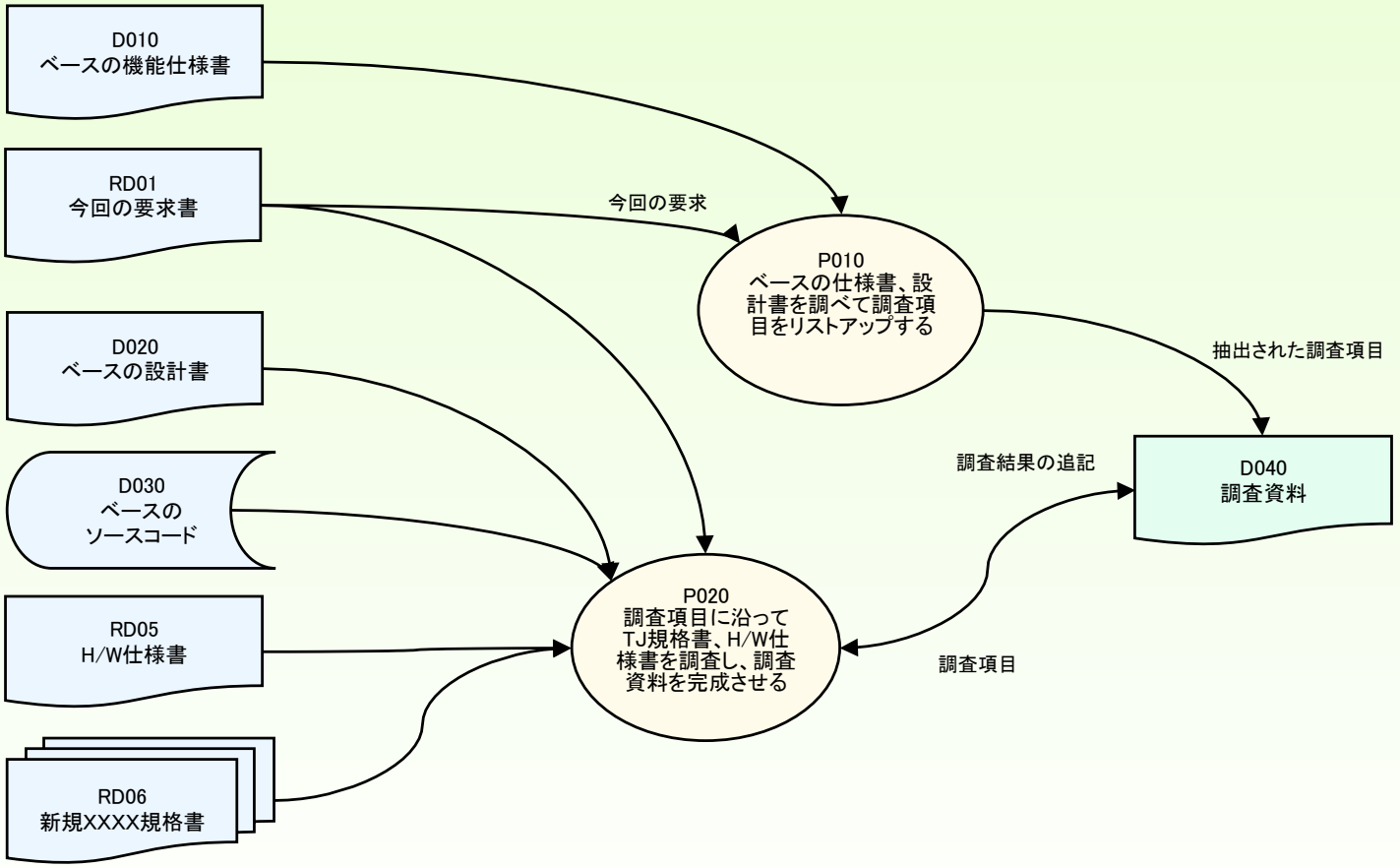
➢ ＰＦＤによるプロセスの表現(サンプル３)

❖リスク項目を引き出すプロセスのPFD(PFD-0)パターン



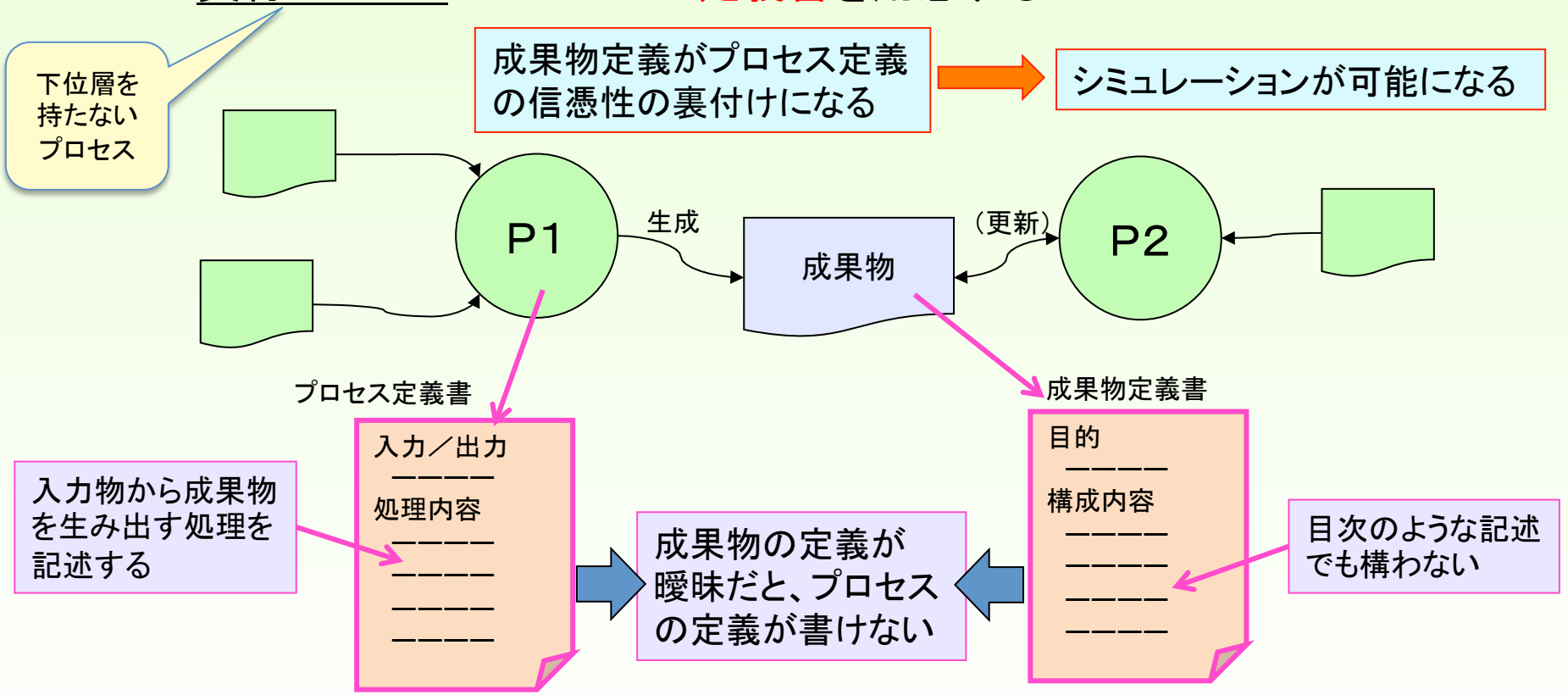
➢ ＰＦＤによるプロセスの表現(サンプル４)

❖ ある新しい規格の機能を製品に取込む調査資料を作成するプロセスのPFD(PFD-0)パターン



➢ 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
  - ❖すべての成果物は最上位のPFDに現れ成果物定義書を用意するる
  - ❖実行プロセスにはプロセス定義書を用意する



3.2 プロセスを表現する

「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

❖ レイアウトは以下のフォームを推奨する

❖ そこに何を書くかを記述しないとプロセス定義を検証できなくなる

成果物定義書

文書番号	成果物名 ・ ファイル名	作成日
成果物の目的		
編集方針		
成果物の構成と簡単内容		

➢ 成果物の構成の定義方法

- ❖成果物定義の中の「構成」をきちんと定義しないと、プロセス定義が行き詰まる
- ❖定義方法は以下のいずれでもよい
  - ❖データの構造体や成果物の目次のように表現する方法
    - ※ 慣れない時は、この方法でも良い
  - ❖構造化分析のデータディクショナリの記述ルールを使う方法

定義	f = ～	「f は右辺の～のように定義される」という意味
結合	＋	構成要素の結合をあらわす
繰り返し	a{ }b a{ } { }b	{ }内の項目が繰繰り返される 繰り返し回数は、a回以上、b回以下 回数表現を省略したときは「0」回以上
選択	[a | b]	[ ]内の項目から1つを選択する
オプション	(a)	( )内の項目は省略してもよい
コメント	／＊　　＊／	／＊と＊／の間にコメントを記述する

➢ 成果物定義書のサンプル

❖派生開発における変更依頼書の成果物定義書

成果物定義書			
文書番号		成果物名 ・ ファイル名	作成日
		変更依頼書	2014/6/6
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} ／＊当該機能に付かして欲しい仕様＊／	

構成を表す番号。成果物の「枝番号」はこの番号を使用する

➢ プロセス定義書

❖レイアウトは以下のフォーマットを推奨

プロセス定義書

プロセス番号		プロセス名		作成日
	入力情報			
	出力情報			
	作業内容			作業担当者
	プロセス実施条件:			
	必要スキル:			
	担当者:			
	作業内容:			
	プロセス終了判定:			

➢ プロセス定義のサンプル

❖派生開発における調査資料悪性のプロセス定義書

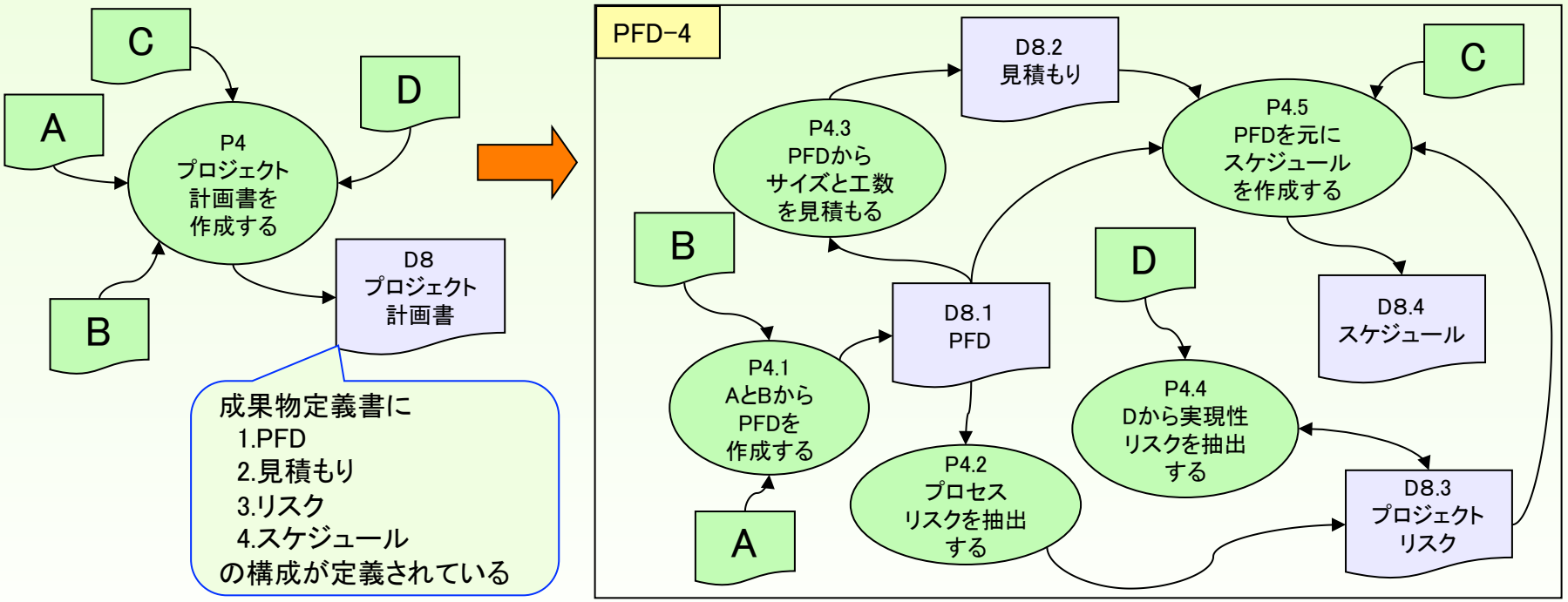
プロセス定義書

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		2014/6/6
	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		
	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		

関数仕様書と同じイメージ

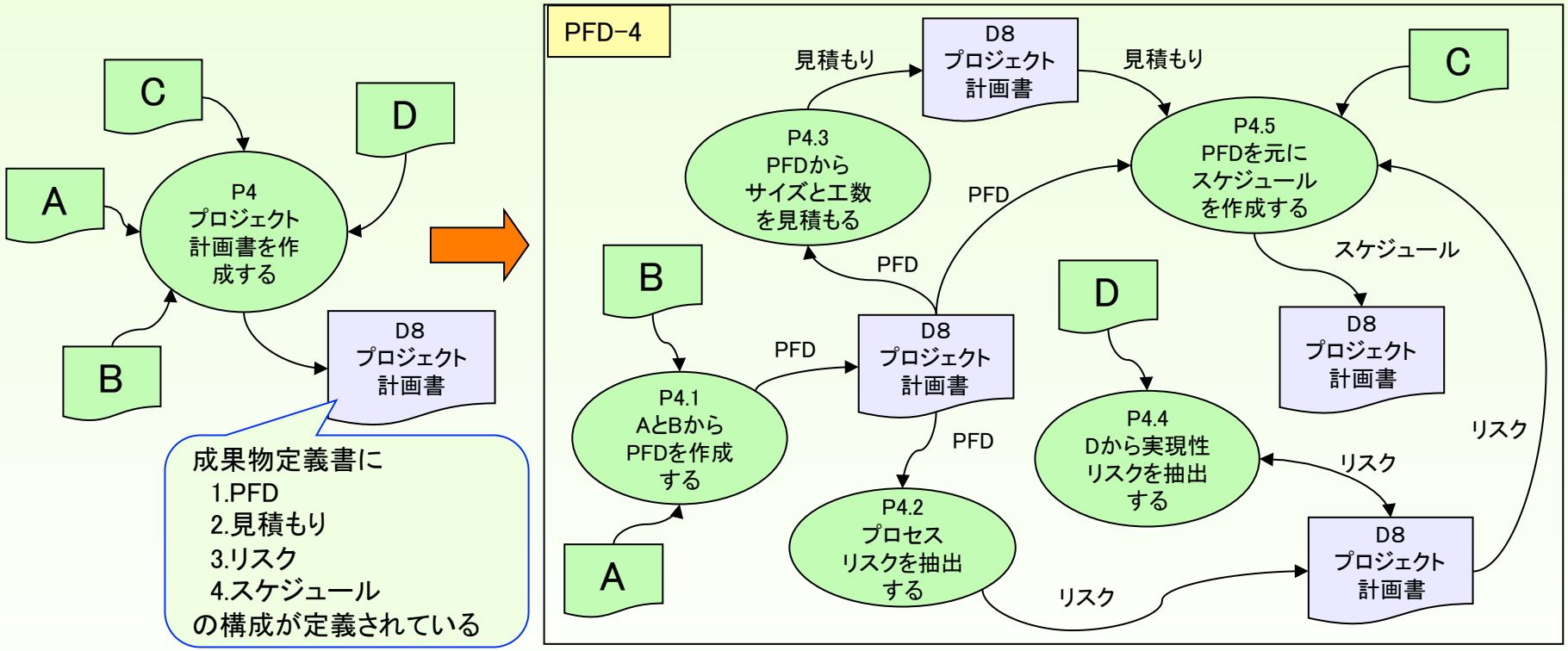
➢ 成果物定義で親子間のバランスを取る方法(1)

- ❖上位層のプロセスと下位層のPFD間の「親子間のバランス」の取り方として、成果物の「枝番号」を使う方法がある
  - ❖下例では、「D8.1」～「D8.4」を生成することで「D8」を生成したことと同じ意味になり、「プロジェクト計画書に統合する」というプロセスは不要になる



➢ 成果物定義で親子間のバランスを取る方法(2)

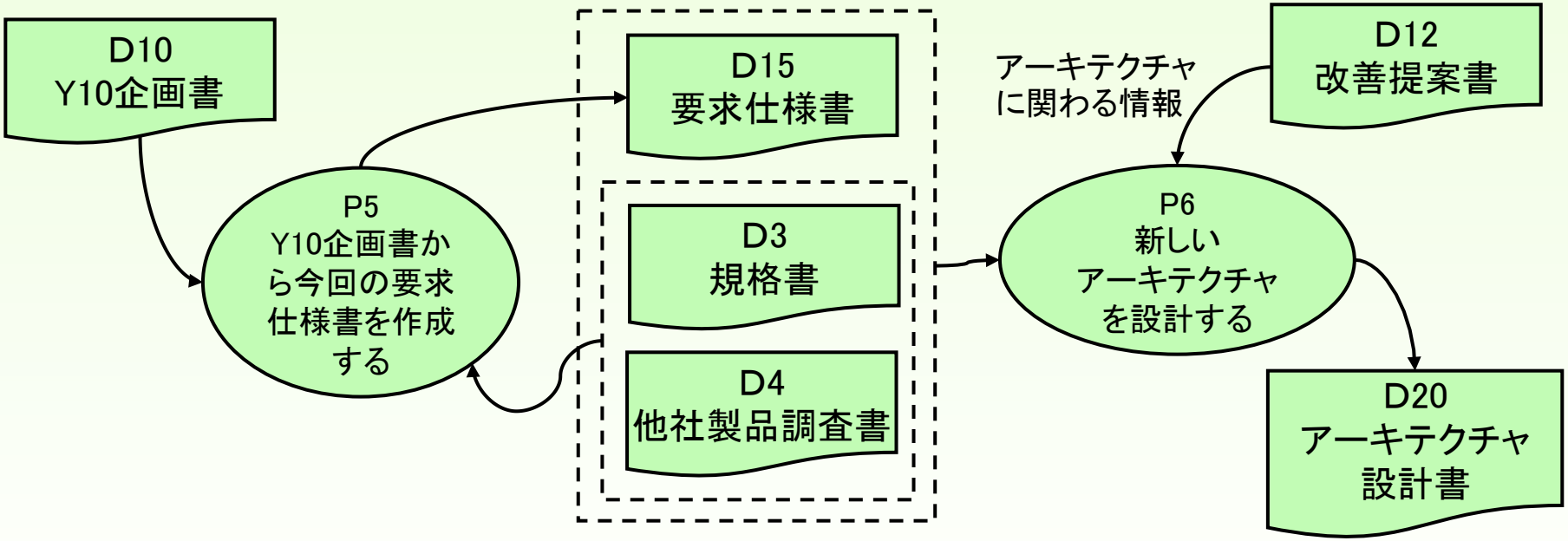
- ❖成果物の「枝番号」の代わりに、「フロー情報」で「親子間のバランス」を取ることもできる
  - ❖この場合でも成果物定義で成果物の構成を定義しておく
  - ❖構成の定義がなければ、フロー情報の裏付けを失う



➢ 成果物のグループ表示

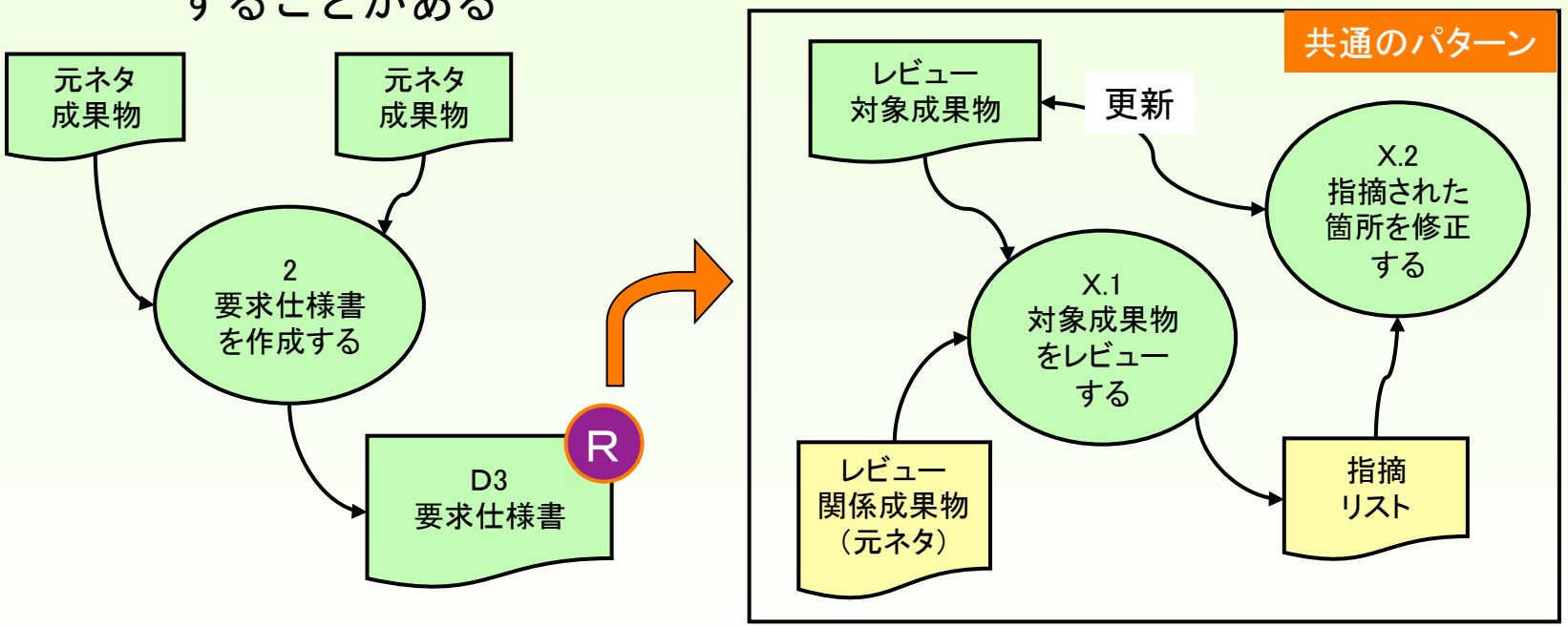
❖一つのプロセスに対してたくさんの成果物が関与し、成果物とプロセスを結ぶフローが多くなってPFDの可読性を損ねるときは、成果物を点線の枠で囲む「グループ表示」の方法を使う

❖成果物の配置の工夫とグループ表示を使うことで整理できる



➢ 定型パターンの扱い

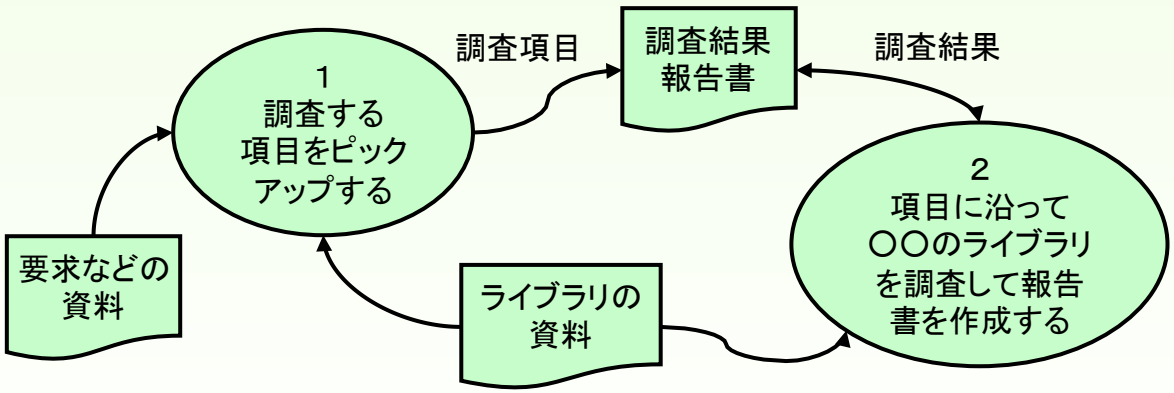
- ❖レビューのようなプロセスをPFD上の全ての成果物に対して表現すると、PFDの可読性が悪くなるので特殊な記号を使用する
  - ❖組織の中でパターンを共有し、PFD上では「記号」で表現するとよい
  - ❖ただし、レビュープロセスでもそのプロセスを強調したり、レビューでの成果物 の活用を表現するために意図的にレビュープロセスを表現することがある



「調査する」プロセスの表現に注意

- 「・・・を調査する」というプロセスは一般に以下のように表現される
  - しかしながらこの表現では、「調査する」プロセスの工数が見積れない

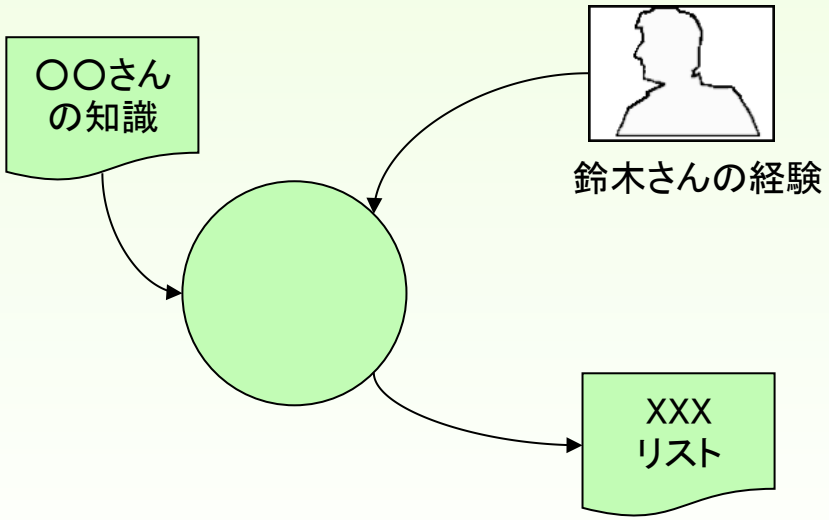
  - 調査項目をピックアップするプロセスと調査するプロセスを分けることで、見積りの問題をクリアする



- 調査項目数の見積りに基づいて「2」のプロセスの工数を見積ることができる
- 「1」のプロセスの実績値によって項目数と項目の内容が見えるので、「2」のプロセスの工数を調整できる。
- 「1」のプロセスの工数の見積りは小さいので、「1」で誤差がでても大きな問題にならない。

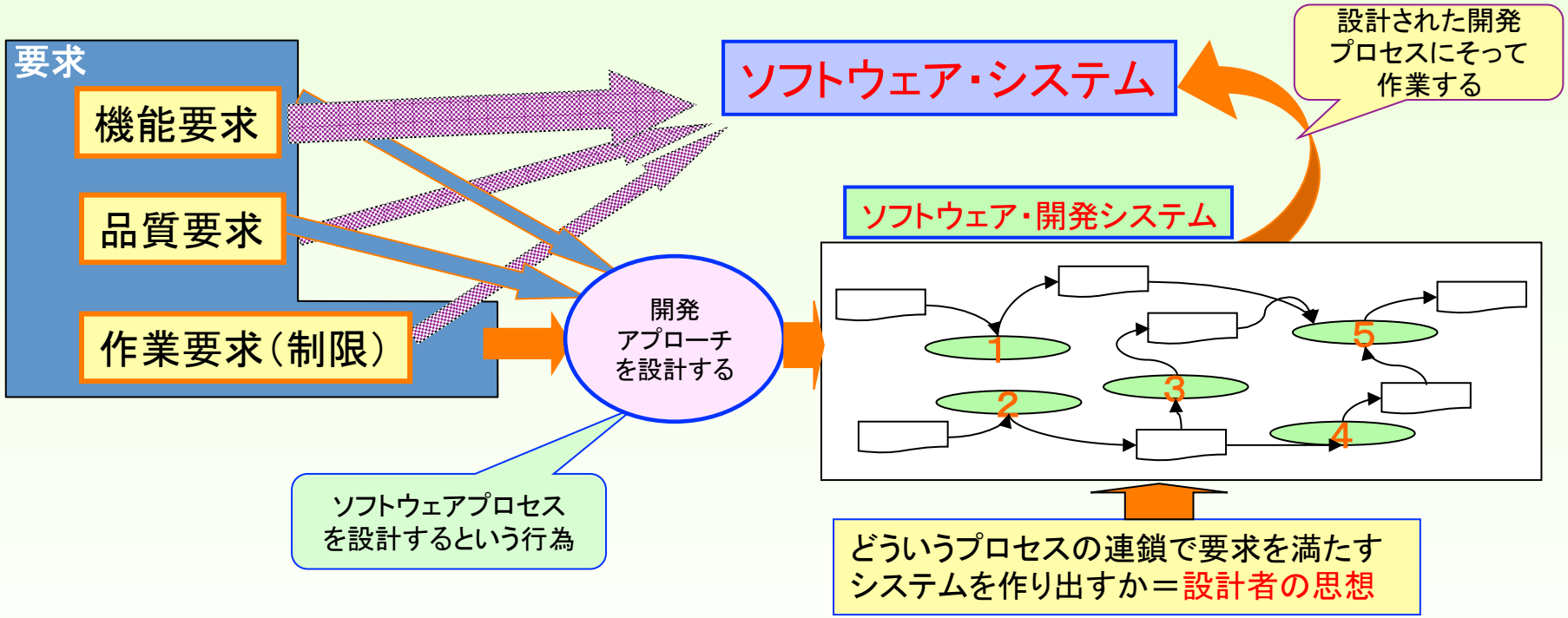
➢ 無形成果物の表現

- ❖入力源として、必ずしも「形」になった成果物が存在するとは限らない
  - ❖上流のプロセスでは、経験豊かな人の「ノウハウ」や「知識」を活用するプロセスも存在するので、上手に表現すること
    - ❖成果物の図を使っても、特別な図を使ってもかまわない
  - ❖ただし、「鈴木さんの経験」も「成果物定義」が必要で、そこでどのような経験内容が活用されるのかを記述する



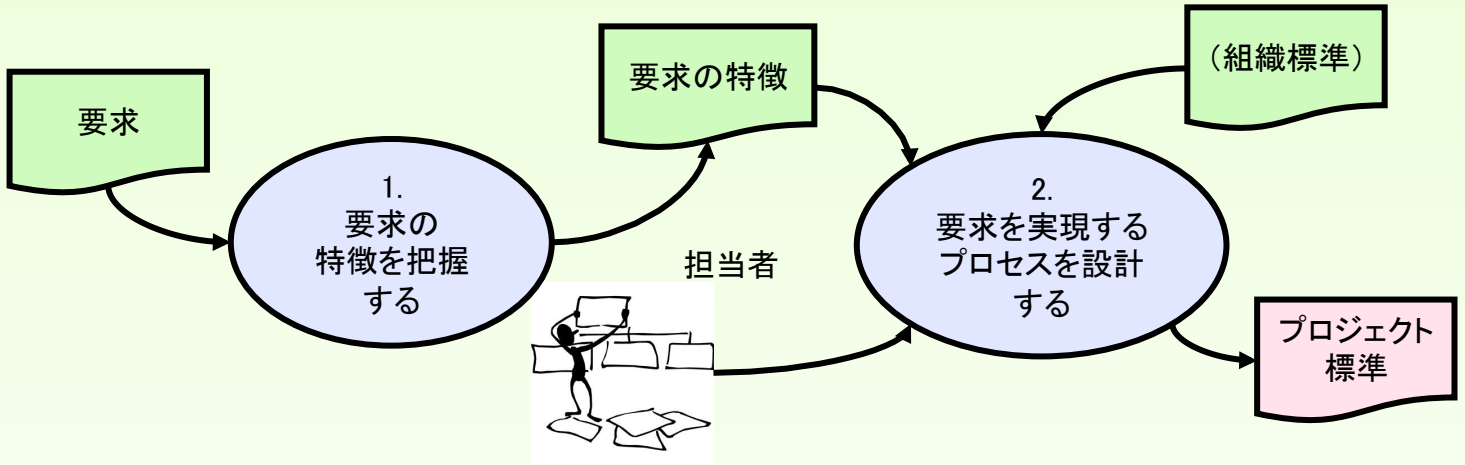
3.3 プロセスを設計する

- プロセスは定義するのではなく設計する
- 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PFDを使用して「設計する」



➢ 自分のプロセスを設計する

❖変化する市場（顧客）の要求を実現するためのプロセスは、実際にその作業を担当する人が「設計」することが重要



多くの組織の間違いは、プロセスを実施する人が自分が実施するプロセスを設計していないこと

➢ PFDを使ってプロセスを設計する

❖自分の担当する範囲のプロセスの表現を繰り返す

プロジェクトリーダー	✓ プロジェクト全体をPFDの「レベル−0」として書く ＝ 責任範囲 ✓ 必要なら、全体を俯瞰するために「CD（概観図）」を書く ✓ 下位層の担当者に実現するためのプロセスの設計を求める ✓ 各担当者のPFDを並べて整合性をとる
担当者	✓ 上位プロセスを意識しながら自分が担当する層（下位も）のプロセスを書く ✓ 出力する成果物を、次の人のプロセスの入力条件に適合させる

- 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

3.3 プロセスを設計する

- 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　「これとこれでできること」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

3.3 プロセスを設計する

- 派生開発におけるプロセス設計の必要性
- 今ソフト開発はそのほとんどが派生開発
  - 要求が多種・多様
  - 開発期間が短くやり直しが効かない
  - 設計したプロセスの善し悪しが短期間に検証できる

➢ プロセスを設計してみよう

- ❖演習課題
  - ❖来年の年賀状を作成するプロセスを設計する

  - ❖入力物
    - ❖年賀状管理住所録
    - ❖今年頂いた年賀状
    - ❖昨年頂いた喪中はがき
    - ❖挨拶語句事例集
    - ❖年賀用イラスト集
    - ❖写真集（家族・風景など）

  - ❖成果物
    - ❖来年の年賀はがき

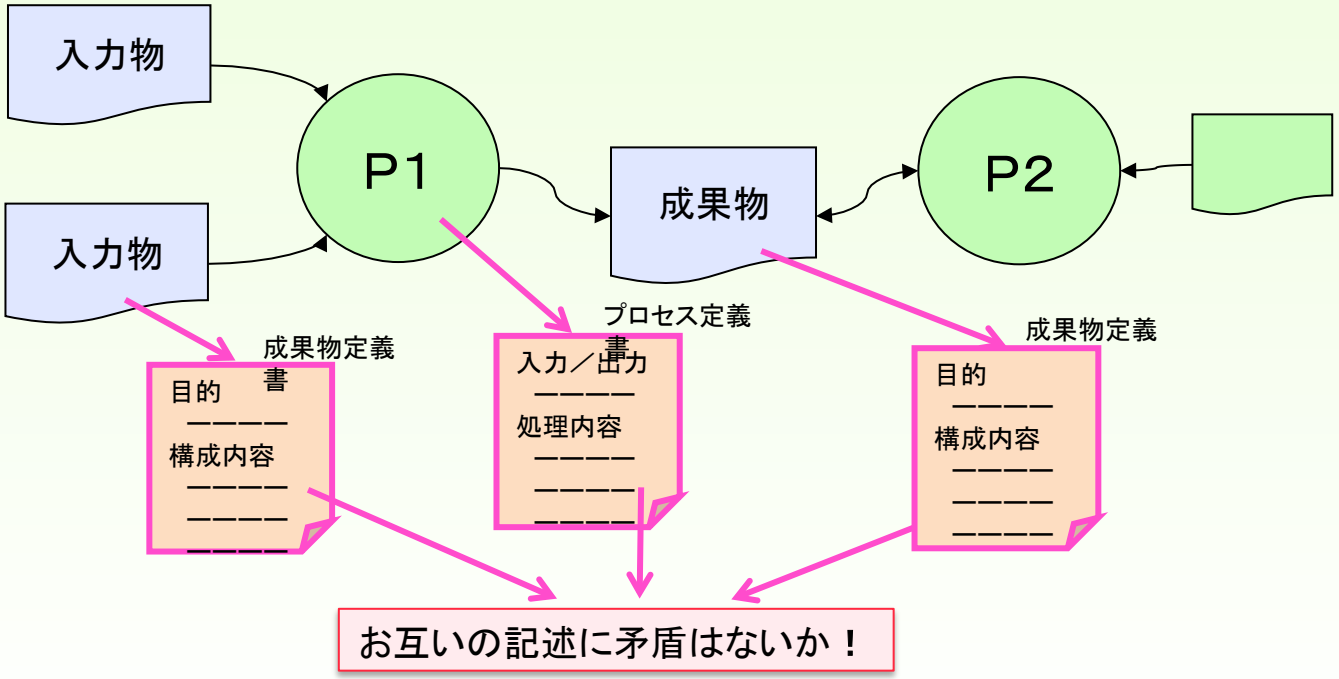
➢ プロセスを自在に設計する

変化する要求に対してプロセスを固定することは危険

- ❖変化する要求に対してプロセスを固定することは従来経験したことの再発に繋がる
- ❖そうならないためには、多様な要求を満たすことの出来るプロセスを設計する必要が有る→プロセスを自在に設計するスキルが必要
  - ❖ 成果物とプロセスの合理的な連鎖の設計(Design)
  - ❖ 成果物の構成の設計
  - ❖ それら成果物を生み出すプロセスの処理(アルゴリズム)の設計
- ❖実際にそのプロセスを実行する担当者が設計することが重要

➢ プロセスをシミュレーションする（１）

- ❖PFDでプロセスを表現（設計）できれば完了というわけではない
  - ❖新規性の高いプロセスはないか
  - ❖実際に予定したとおりに作業が捗るか
  - ❖入力物が遅れたときにどう対応するか
  - ❖その構成内容・アルゴリズムから支障なく生成できるか
  - ❖他に参照すべき資料（データ）が漏れていないか



➢ プロセスをシミュレーションする（２）

- ❖「設計」したプロセスをシミュレーションで「安定」させる
  - ❖ 成果物が変化すればプロセスも変化する
  - ❖「追加」したプロセス、「変化」させたプロセスの周りをを入念に
  - ❖ 機能的合理性と経済的合理性を確認する

※ 新規性が高いものほど必要 → 本番で混乱しない

これで
市場の変化に対応できる

- ❖ 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
  - ❖「PFD」を書いただけでは、習慣は改まっていない
  - ❖シミュレーションによって、本番で「新しいプロセス」を迷いなく実施できる

➢ プロセスをシミュレーションする（３）

- ❖ 途中で発生する「事態」に対処しやすい
  - ❖ 発生した事態は、事前に考えたことか
  - ❖ その事態は、他のどのプロセスに影響するか
  - ❖ 当初の結果を得るために、プロセスをどう変化させるか

シミュレーションした分だけうまく行く

- ❖ 初めてのことも失敗しない
  - ❖全く初めてのプロセス
  - ❖今回変化させたプロセス

⇒ 本番前に体に覚え込ませる

初めての状態で本番に入ってしまえばプロジェクトは失敗する

➢ プロセスを評価する（１）

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！

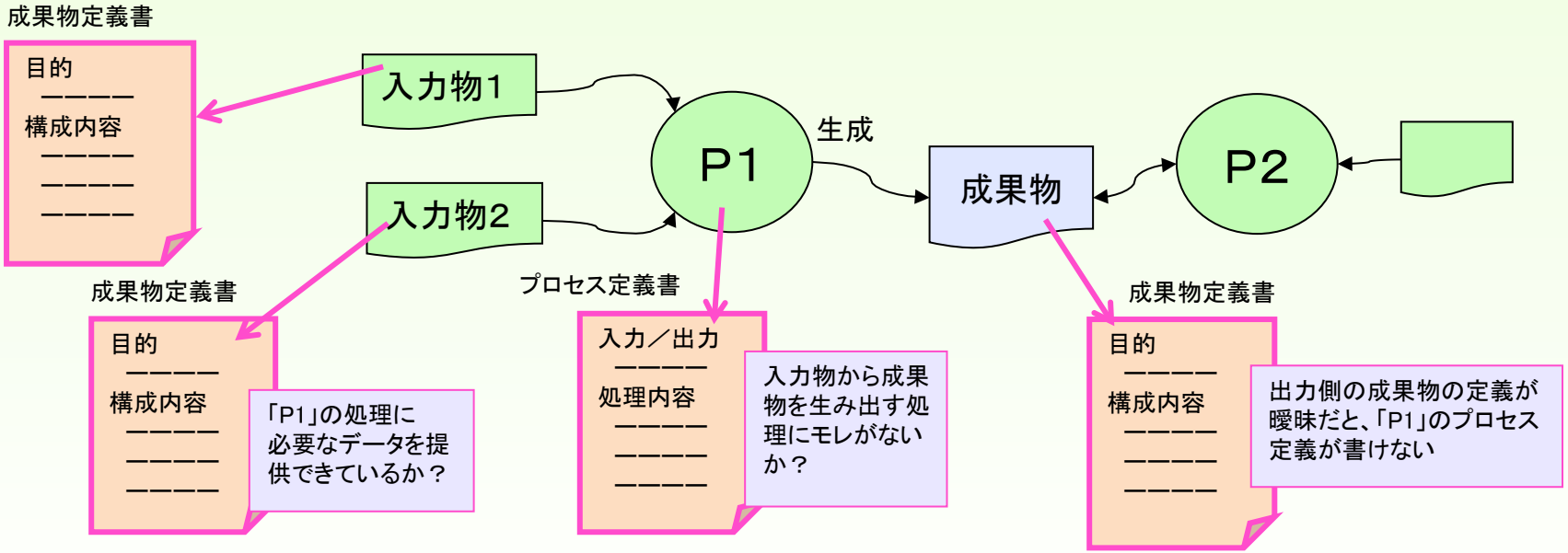
➢ プロセスを評価する（２）

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、目的の成果物が作れ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

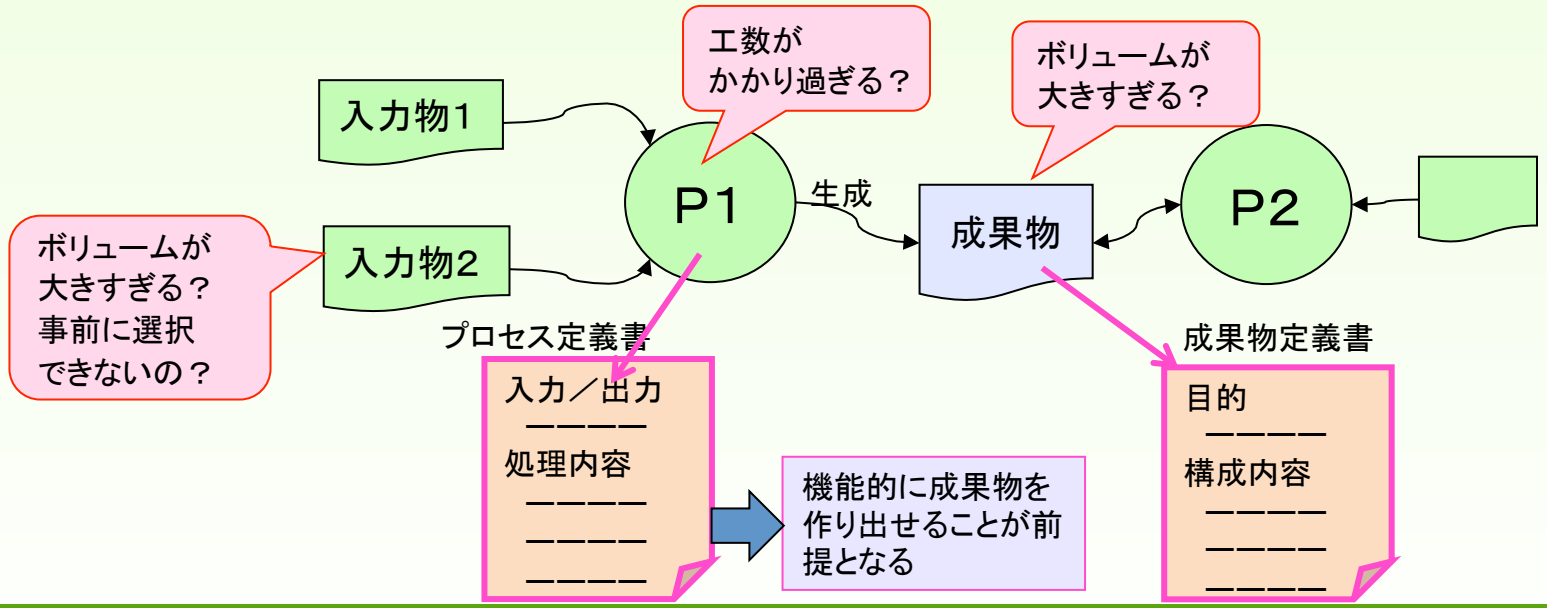
シミュレーションされていること！



➢ プロセスを評価する（３）

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？
  - ✓成果物の内容に重複が必要以上に存在していないか
  - ✓成果物の中野データで１度も使われないデータは存在していないか



４．どんなことに使える

４．１　プロジェクトの検証

４．２　原因分析

４．３　組織標準

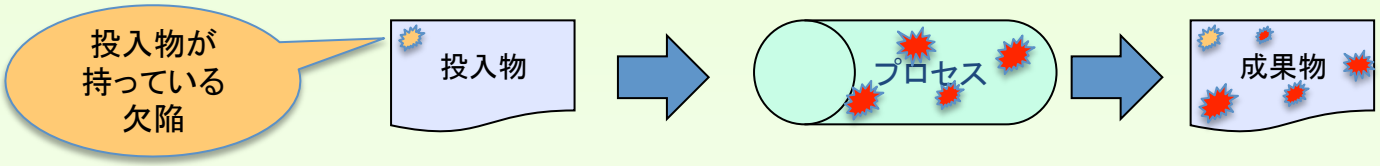
４．４　プロセスの検証

４．５　その他

- 実施されたプロジェクトの検証（プロセスと成果物の検証）
  - プロジェクトの実施にあたって、策定され,検証され,承認されたPFDが有ることで
  - 実施されたプロセスは正しく実施されたのか、作成された成果物は正しく作成されたのかを、PFDに照らして検証することが出来る
    - 実施されたプロセスは、承認されたプロセスに従って実施されたか（→プロセスの品質）
    - 作成された成果物は、承認されたプロセスに従って作成されたか（→成果物の品質）

4.2 原因分析

- 問題が発生したときの原因分析に
  - 今回はなぜ成功しなかった（うまく行かなかった）のかを知る手がかりを見つける
  - プログラムの“バグ”や、スケジュールの遅延等のエラーは、プロセスの欠陥が主な原因である



成果物の欠陥は、プロセスの欠陥で作られる

  - プロセスの欠陥の３つのタイプ
    1. 予定されたプロセスが不適切に実施された
    2. 必要なプロセスが省かれた
    3. 必要なプロセスがあったのに気付かなかった（ために省かれた）

「失敗」を繰り返すチーム・組織は③ の原因に気付いていない

その場合、③ の原因は「失敗」から学べない可能性が残る

4.3 組織標準

- 組織の標準プロセスを作る
  - プロジェクトごと（開発案件ごと）にプロセスを最初から作ると次のような課題が生じる
    - 担当者によりプロセスが安定するまで時間がかかる可能性
    - 全体の工数におけるプロセス設計の工数負担
  - 実績のあるプロセスをベースに「組織標準」を作る

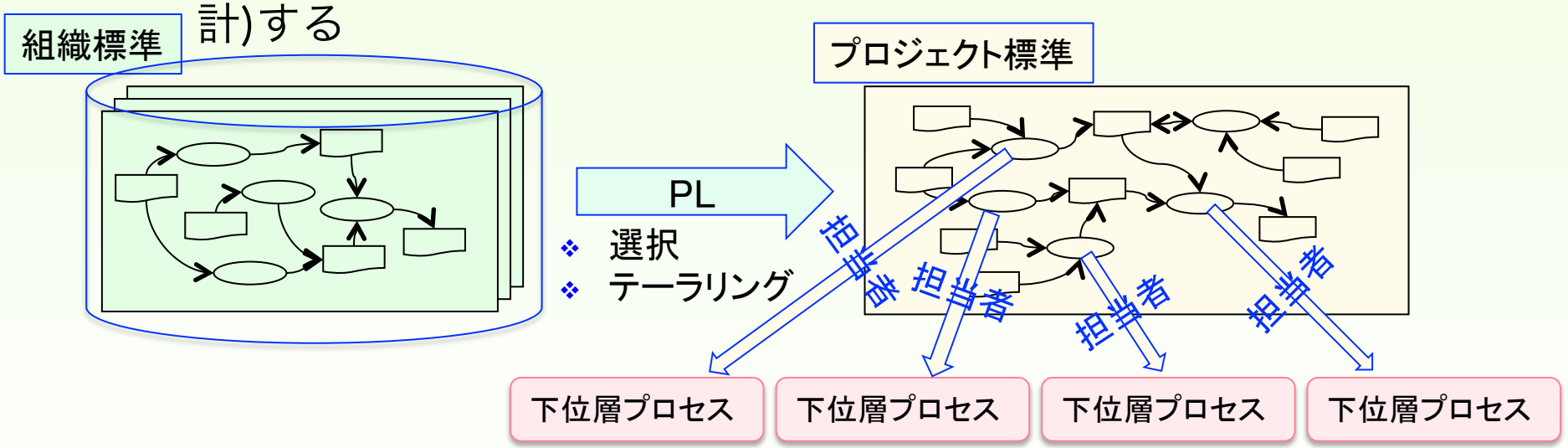
規模や案件タイプに応じて複数パターン作成する

  - 実施したプロジェクトで上手くいったプロセス(プロジェクトが成功していなくても上手く出来た部分)を抽出しプロジェクト固有の部分を削除する→基本要素の抽出
  - 抽出した基本要素をベストプラクティスとして登録する
  - 登録された基本要素から組織標準を策定する
  - 策定された組織標準を実施するプロジェクトのプロセスの参考にする

- 組織標準から実施するプロセスを策定する
  - プロジェクトごと（開発案件ごと）に、組織標準から適切なプロセスを選択しプロジェクトに適合させる
    - 開発案件に合った(近い)プロセスを組織標準から選択する
    - 選択したプロセスを開発案件に適合するように変更(テーラリング)する（→プロジェクト標準）

プロジェクトの要求に合わせて不足する成果物とプロセスを設計付加する

    - 策定されたプロジェクト標準の下位プロセスを担当者が策定(設計)する

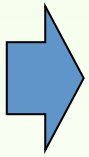


4.4 プロセスの検証

- 組織が持っている（実施している）プロセスの検証
  - プロセス改善の手がかりをつかむ
    - 現在組織が実施しているプロセスをPFDを用いて記述
    - 記述されたプロセスを評価

2つの合理性（機能的・経済的）について評価し改善の手がかりを得る

- スケジュールへの展開
  - プロセス定義書からスケジュールへ展開
    - ２つの合理性の検証が終了
    - シミュレーションも終了



プロセスの実施順序が観えてくる

    - プロセス定義書の「作業内容」をスケジュールの作業項目としてスケジュールを策定

4.5 その他

- その他色々
  - ソフトウエアの開発だけでなく
  - 身の回りのいろんなことに
    （初めて行うことを上手く実施するために）

・
・
・
・

参考文献

- 清水吉男：ＰＦＤ(Process Flow Diagram)の書き方
  “ http://homepage3.nifty.com/koha_hp/process/PFDform3.pdf ”
- 清水吉男：ＡＦＦＯＲＤＤ勉強会PFD1.1.pfd

御清聴ありがとうございました

『PFDってなに？』

梶本 和博

派生開発推進協議会

株式会社エクスモーション