

「非ウォーターフォール型開発WG活動報告書」 (概要資料)

平成23年3月31日

独立行政法人 情報処理推進機構
ソフトウェア・エンジニアリング・センター

- ウォーターフォール型開発形態でソフトウェア開発を経験してきた人たちにおいては、非ウォーターフォール型開発形態について十分に理解されているとは言えない。
- わが国のソフトウェア産業の仕組みに照らした非ウォーターフォール型開発の課題もいくつか指摘されている。



- 本書は、ウォーターフォール型開発形態に馴染んだ人たちを対象に、非ウォーターフォール型開発形態についての整理された情報を提供することを主目的とする。
- 非ウォーターフォール型開発形態に馴染んでいる人たちに対して、対象ソフトウェアのコンテキストに適しているかを今一度検証してもらうことに使用していただき頂きたい。
- 本書が、対象ソフトウェアのコンテキストに応じた最適な開発形態を選択するためのヒントとなって欲しい。

平成21年度の課題と平成22年度の取組み

平成21年度の課題

平成22年度の取組み

日本のソフトウェア競争力を高める
生き生きと働ける環境を作る

目指すべきゴール

日本のソフトウェアの作り方

領域
特定

アジャイル

契約 契約のあり方、調達、制度設計

価値評価 経営層やユーザ企業への理解促進

環境整備 管理手法や技術面の整備

普及 コンサルタント等の役割の整備
人財育成

調査 欧米の競争力(ビジネスドライバ、
産業構造など)の調査

非ウォーターフォール型開発の適用に適した領域
を見定め、その活用を促進していく。

・非ウォーターフォール型開発に適した契約モデルの検討。
・非ウォーターフォール型開発に適した契約モデルに
沿った契約のひな型の作成。

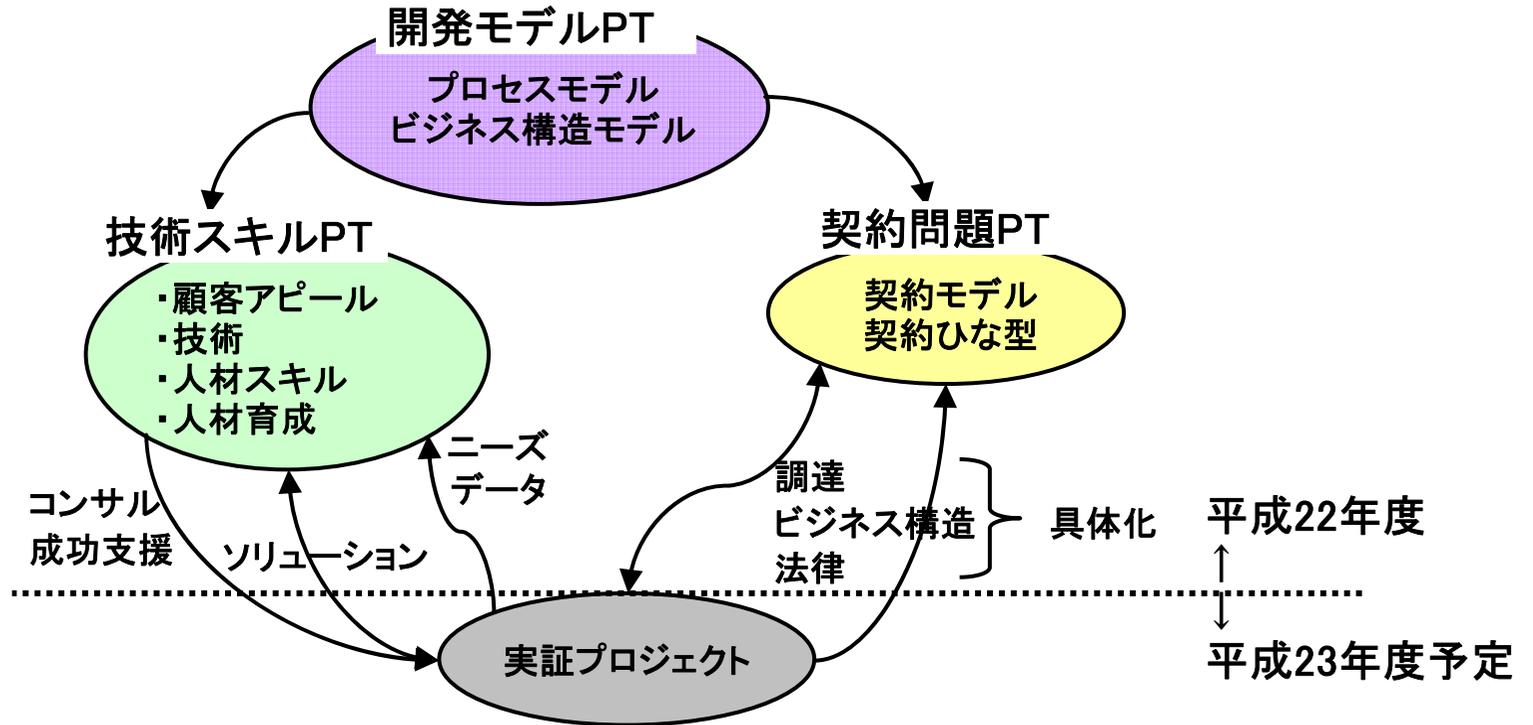
・顧客・経営層が考慮すべき点とその検討。

・顧客側と開発側に必要な包括的エンジニアリング技術、
プロジェクト運営技術・スキルの明確化。

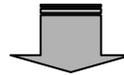
・人材育成方法の検討。
・必要な技術・スキルの獲得方法の検討。

非ウォーターフォール型
開発における
重点課題

WGにおける3つのPTとその目的



- 情報システム開発を大別すると、次の二つのタイプがあると考えられる。
 - ・ 要求が変化しないことを前提としたウォーターフォール型開発
 - ・ 要求が日々変化することを前提とした非ウォーターフォール型開発



- 非ウォーターフォール型開発の例
 - ・ プロトタイプ/スパイラル/RAD/RUP等
 - ・ アジャイル開発
 - Evo/Scrum/DSDM/XP/FDD/Lean Software Development/Crystal Clear/EssUp等



本報告書では、アジャイル開発を非ウォーターフォール型開発の代表例として論じている。



本報告書では、アジャイル宣言の4つの価値と12の原則を満たすソフトウェア開発手法をアジャイル開発とする。

■アジャイル開発に適した領域

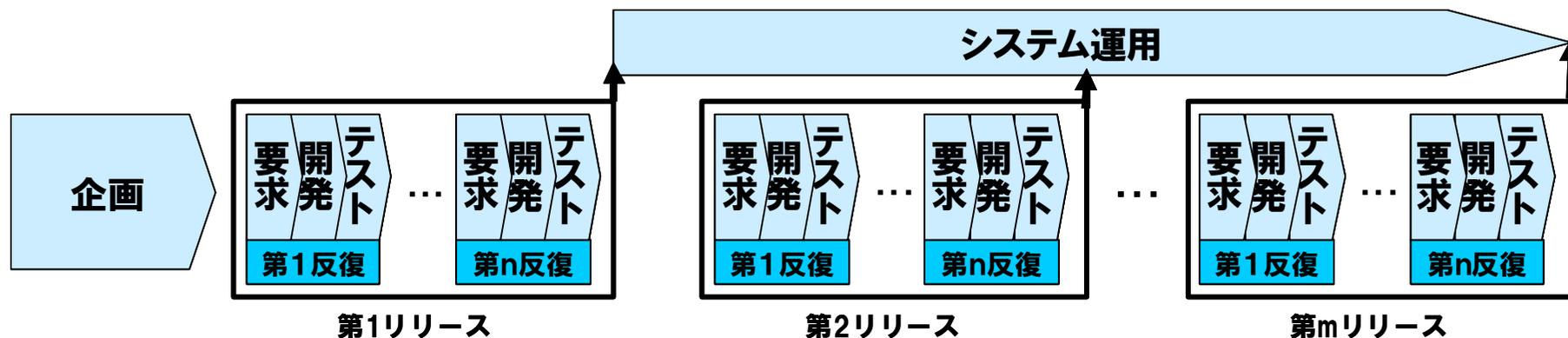
- ・ビジネス要求が変化する領域
- ・リスクの高い領域(市場リスク・技術リスク)
- ・市場競争領域(TTM-Time to Marketの短縮が優先となる領域)
- ・要求が日々変化することを前提とした非ウォーターフォール型開発

■アジャイル開発の試行領域

アジャイル開発による経験が十分には蓄積されておらず、現在、チャレンジと創意工夫が求められているのは、以下の領域である。

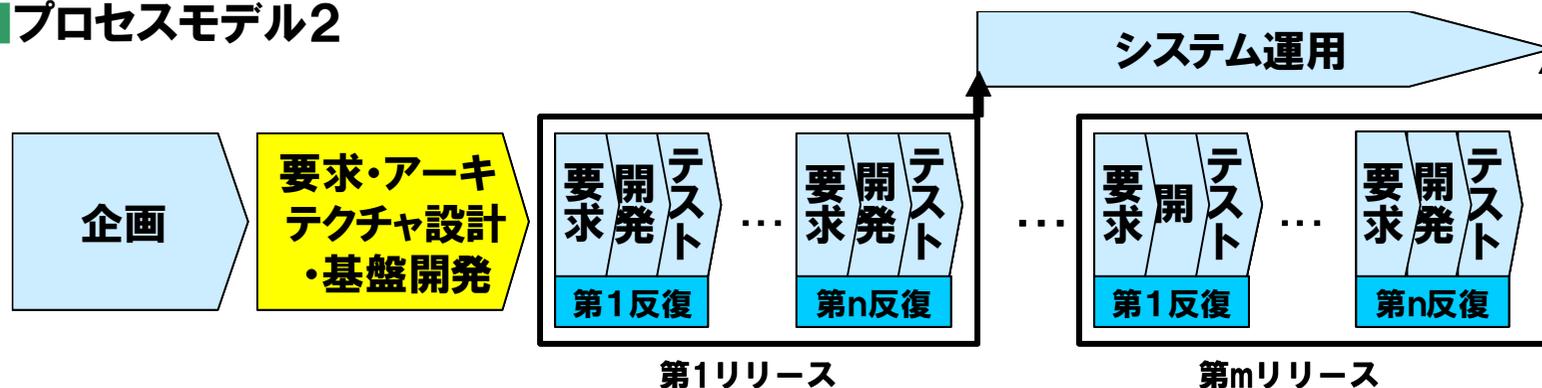
- ・大規模開発
- ・分散拠点(オフショア含む)開発
- ・組織(会社)間をまたぐ開発チームによる開発
- ・組込みシステム開発

■ プロセスモデル1



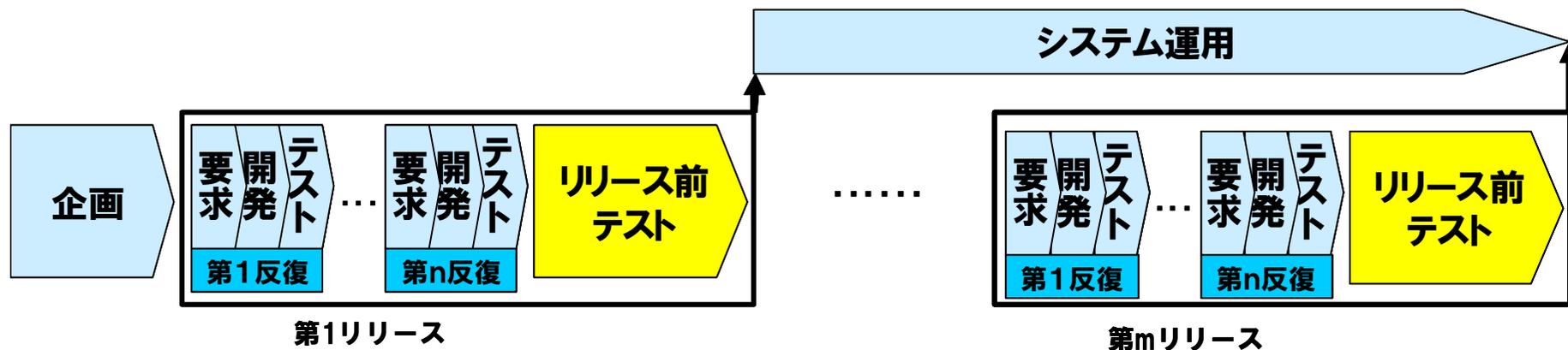
- n=1のケースもあり。

■ プロセスモデル2



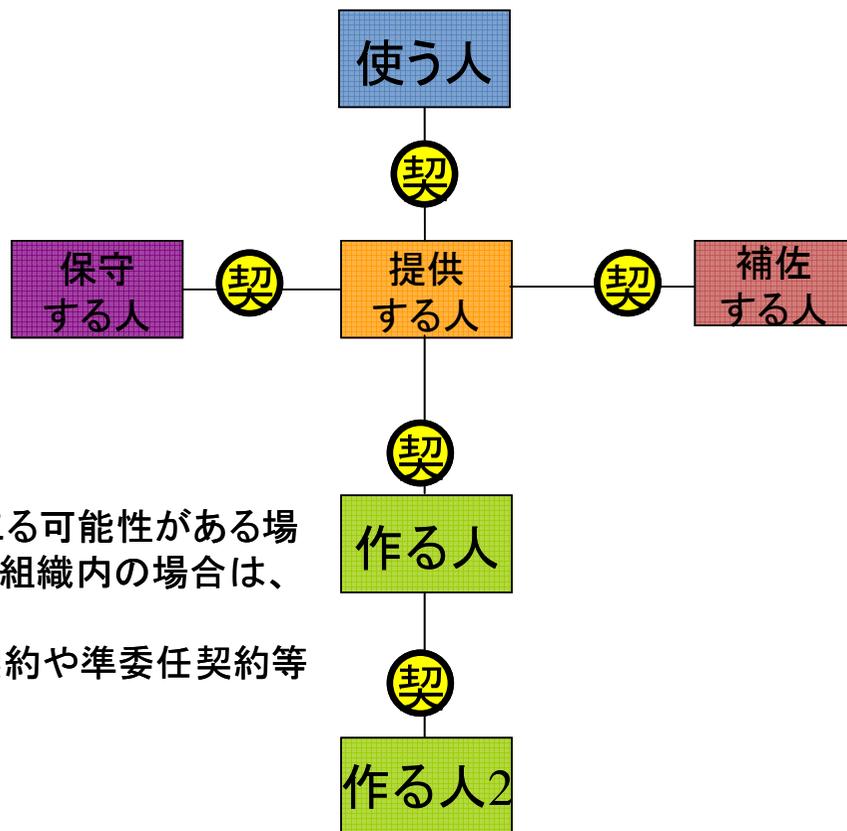
- 比較的大規模システム／新規開発で全体のシステム構造が不明確なケースなど

■ プロセスモデル3



- ・ アジャイル開発では反復ごとにリリースできる品質までテストを行うことが原則だが、各リリース工程前に行く 重点的なテストを実施することがある。
- ・ リリースは複数回繰り返される

■ビジネス構造モデルの基本パターン



契で、契約の起こる可能性がある場所を示す。同一の組織内の場合は、契約はない。
契約には、請負契約や準委任契約等の種類がある。

- ・ 使う人 ⇒ システムを実際に使用する人 (システムの利用者、エンドユーザ)
- ・ 提供する人 ⇒ 開発されたシステムやサービスを使う人に提供する人
- ・ 作る人 ⇒ システムの開発者
- ・ 作る人2 ⇒ 作る人の会社と契約している、作る人と一緒に開発する人 (作る人の会社とは別会社が多い)
- ・ 保守する人 ⇒ リリースされたシステムを保守する人 (作る人と重なることが多い)
- ・ 補佐する人 ⇒ アジャイル開発の進め方等で、開発プロジェクトを支援する人 (コンサルタントやファシリテーターが多い)

■非ウォーターフォール型開発のメリット

- ・優先度の高い機能が順次提供され、提供された機能を検証、投入することで、変化が激しく、優先度も変化するビジネス環境に対応できるサービスやシステムを手に入れられる
- ・機能単位等でリリースすることにより、顧客・経営層は、開発途中での完成状況を知ることができ、顧客・経営層は、ビジネス上のアクションをすばやく採ることも可能となる。
- ・本番稼働後の経営環境や戦略の変更、ユーザが期待する品質の変化などに対応して、保守しやすく、品質向上もさせやすいため、改善の可能性と利便性が極めて高い。

顧客・経営層によるアジャイル開発手法の採用判断の例



D社

ライバル社は、まもなくサービスを開始すると発表した。
わが社も、早期にサービスを提供しないと、シェアを奪われてしまう。



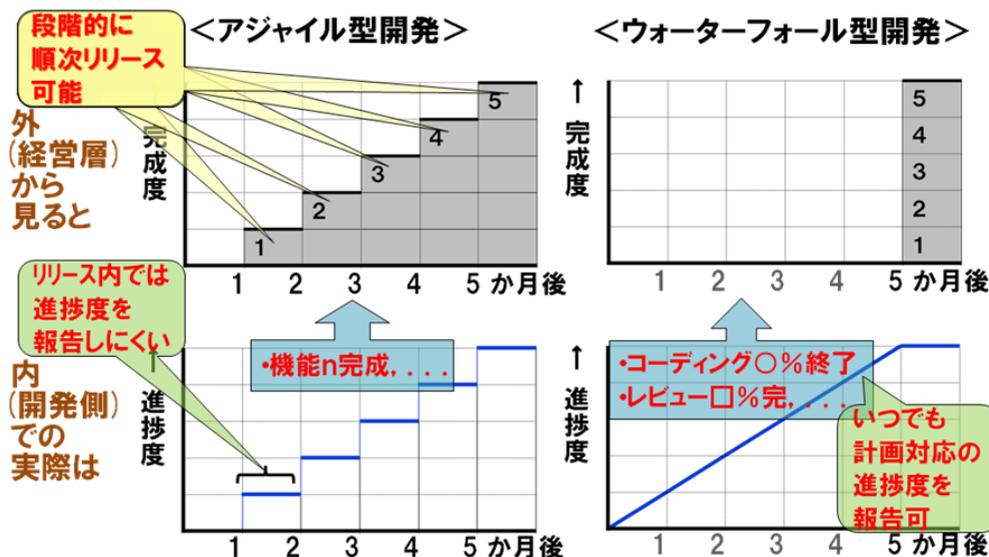
S社

	サービス	システム/ソフトウェア	
		仕様	実装
優先順位 ↑	メニュー1	機能1	コード群1
	メニュー2	機能2	コード群2
	メニュー3	機能3	コード群3
	メニュー4	機能4	コード群4
	メニュー5	機能5	コード群5

アジャイル型開発の採用

- 顧客・経営層は、アジャイル開発の採用を決断した時点で、顧客がチームの一員として参画し、タイムリーな意思決定を行ったり、品質や進捗状況の把握等に関し、主体的に開発に関わらざるをえないということに十分な理解と覚悟を持つことが必要となる。
- アジャイル開発においては、反復の都度、コードを書き変えていくスタイルが採られる。品質に重大な悪影響が及ぶかどうかの観点での、プロダクト品質の見える化が求められる。
- アジャイル開発の特徴に応じた「見える化」項目を用いて開発プロジェクトとの円滑なコミュニケーションを図り、アジャイル開発採用の本来の目的が損なわれないように努める必要がある。

アジャイル型とウォーターフォール型における開発進捗度の見え方の特徴



■アジャイル型開発におけるスキルとは何か

- ・プロジェクトに参加するメンバの役割に応じた、もしくは他の役割を支援するための能力。
- ・「～～が出来る、～～が作れる」と表現し、一般的な教育、育成によって成長が可能。
- ・プロジェクトのアウトプットに直接的に作用しやすい。

■アジャイル型開発型における価値観とは何か

- ・アジャイル型開発に求められる大事なものを理解し、それに沿って活動を心構え。
- ・「～～を優先する、～～を大事にする」と表現し、教育、育成による成長というよりも、気付き、自立心により磨かれていくものである。
- ・プロジェクトのアウトプットには間接的に作用し、チームビルド、コミュニケーション、自己実現といった要素には大きな影響をもたらす。



・アジャイル型開発では、スキルよりも価値観を重要視する。



・アジャイル型開発では、作業の実施内容がウォーターフォール開発と異なる点がある。

アジャイル型/ウォーターフォール型開発に必要とされる作業例(1/2)

アジャイル型開発	ウォーターフォール型開発
<p>プロダクトオーナー システムのための要件を定義し伝達する役割、優先順位を設定</p>	<p>ユーザー 事業、顧客側</p>
<p>スクラムマスター ファシリテーティング <u>※調整役であって意思決定をする役割ではない</u> <u>アジャイルプロセスのルールを施行する</u></p>	<p>プロジェクトマネージャ プロジェクトリーダー、SE 人モノ金の判断をする 仕様、スケジュール、要員の管理 報告、変更の管理</p>
<p>開発者メンバー 要求定義を行う コードを書きながらプロダクトオーナーやテスターと協力してコードが開発されていることを確認する コードのための単体テストを書く 受け入れテストをサポートするテストを書き、テストの自動化を行う 毎日コードを共有リポジトリにチェックインする</p>	<p>開発者、SE、PG 要求定義を行う コードを書く 単体テストを書く テストの自動化を行う コードを管理する</p>
<p>開発者メンバー(テスターを含む) 機能の理解を確認し、要望されている機能と紐付けられているか確認する。 コードが書かれている間に、受け入れテストのテストケースを書く。 受け入れテストの際にコードをテストする <u>毎日、共有リポジトリにテストケースをチェックインする</u> 受け入れテストやコンポーネントテストを、継続的なテスト環境に統合するためにテストの自動化を開発する</p>	<p>テスター 機能理解、要望機能との紐付け確認 テストケースを書く テストをする テストコードを管理する</p>

アジャイル型/ウォーターフォール型開発に必要とされる作業例(2/2)

アジャイル型開発	ウォーターフォール型開発
<p>アーキテクト(プロジェクト外) 多くのアジャイルチームはアーキテクトという言葉が役職に就く人々を含んでいない。 アーキテクチャはアジャイルチームにとって非常に重要である。チームの活動を通して「アーキテクチャは出現する」と言われているが、システムレベルでは、アーキテクチャはシステムの全体の構造を決定する責任を持つシステムアーキテクト、ビジネスアナリストによって調整されるものである。 アジャイルチームはチームの外側に存在する複数のアーキテクトと話す窓口を持つことになる。</p>	<p>アーキテクト 設計初期に登場 システム全体の構造を決定する責任あり プロジェクトチームは、複数のアーキテクトと話す窓口を持つことになる。</p>
<p>インフラ(プロジェクト外) マシン、ネットワーク環境を提供する</p>	<p>インフラ マシン、ネットワーク環境を提供する</p>
<p>QA・品質保証 (プロジェクト外) <u>多くの場合、品質保証のための主な責任は、開発者やテスターに移行されることになる。</u> <u>QA担当者は全体の品質を監督するという、本来意図されていた役割(レビューと品質の監視)に戻る。</u> <u>QA部門にいたりリソースの多くをプロジェクトチームと一緒に仕事できるように派遣してリアルタイムに品質を確認させることになる。</u> <u>システムレベルのテスト開発(負荷試験等)に関わるようになる。</u></p>	<p>QA・品質保証 品質保証部門として社内検査を行う 出荷時の最終チェック的な役割 非機能試験は行わない、形式チェック</p>

プロジェクトの進め方(プロセス)において必要とされる作業例

アジャイル型開発		ウォーターフォール型開発
仕様検討 大まかなやりたい事のヒアリング		仕様検討 契約の元となる確定条件の整理
イテレーション計画策定 やりたい事の優先度を決め、どれくらいの期間で、どの程度の機能 (シナリオ)を作るかを定める		プロジェクト計画策定 ウォーターフォール型のストレートな計画策定 基本、やりたい事は全部やる、全部設計してみる
反 復	イテレーション (設計、開発、テスト) 小さな単位にしてドキュメント、コードをセットで作る テストコードも作る、毎日テストをする	基本設計 ドキュメントベースの設計 この時点で再度レビューをして再見積もりもあり 契約によっては、ここから先が別会社ということもあり
	イテレーション (提供・デモ、評価) 2週間程度の周期で、動くコードを見せる 実際に触ってもらう その場でフィードバックをもらう	詳細設計、開発、テスト ドキュメント、もしくはビルダーベースの設計 開発 開発が終わってから、まとめてテスト
	納品 ある程度の単位、決まった期日になったら、本番環境にデプロイする	品質監査、納品 プロジェクトとは別部門による品質監査、形式チェック 納品

役割とスキルの対応例

		役割						
		テクニカル コーチ	アジャイル コーチ	プロダクト オーナー	ストーリー オーナー	ビジネス アーキテクト	開発チーム 代表	開発メンバ
プロジェクト運営		◎	◎	◎	○	○	◎	○
運営チーム	スクラム一般	◎	◎	◎	◎	○	◎	◎
	スクラム PJ固有	◎	◎	◎	◎	○	◎	◎
業務知識	実務	—	—	○	◎	○	○	◎
	ドメインモデル	—	—	○	○	◎	◎	◎
	モデリング	○	◎	△	○	◎	○	◎
技術	Java	◎	◎	—	—	—	○	◎
	OOプログラミング	◎	◎	—	—	—	○	◎
	OO設計	◎	◎	—	—	—	○	◎
	UML	◎	◎	—	—	◎	○	◎
	リファクタリング	◎	◎	—	—	—	○	◎
	TDD	◎	◎	—	—	—	○	◎
	単体・結合テスト	◎	◎	—	—	—	○	◎
	システムテスト	◎	△	—	—	—	○	◎
	HTML/CSS	◎	△	—	○	—	○	◎
	JavaScript	◎	△	—	△	—	○	◎
	WebUIテスト	◎	△	—	○	—	○	◎
	SQL	◎	△	—	—	—	○	◎
	DB物理設計	◎	△	—	—	—	○	◎
	DBチューニング	◎	△	—	—	—	○	◎
開発環境	Eclipse	◎	◎	—	—	—	○	◎
	Subversion	◎	◎	—	—	—	○	◎
	Trac	◎	◎	—	—	—	○	◎
	Hudson	◎	◎	—	—	—	○	◎
	フレームワーク	◎	△	—	—	—	○	◎
	ステージング環境	◎	△	—	—	—	○	◎

対象別育成カリキュラム例

	開発チーム	スクラムマスター	顧客／プロダクトオーナー	先行チーム	リーダー	PM	経営者層／購買担当など
アジャイル概要	○	○	○	○	○	○	○
アジャイル基礎知識	○	○	○	○	○	○	
アジャイル疑似体験	○	○	○				
業務知識	○		*	○			
開発環境	○			*			
基本アーキテクチャ	○			*			
業務分析／モデリング	△		△				
開発技術	△			*			
ファシリテーション概要	○	○	○	○	○	○	
ファシリテーション演習		○			○	○	

○: 立上げ前後の必須教育の領域 △: 事前に準備が困難でOJTが必要な領域 *: 内容を組織内で個別に検討する必要がある領域

■平成21年度に実施された契約方式調査による契約事例から導かれた契約の問題点

ウォーターフォール型開発における請負契約適用時の問題点

問題点(リスク)	発注側	受注側
要件が未確定	—	成果物の要件が決まっていないため、契約金額の範囲内で業務が完了できない場合がある。
成果物が不明確	—	契約時に成果物を規定できないため、納入すべき成果物としての妥当性を主張できない。
性能と品質が不明確	工期延長に伴い、システムの稼動時期が遅れるリスクがある。	契約時に合意した内容がないため、発注側の要請により、工期が際限なく延び、契約金額の範囲内で業務が完了できない場合がある。
開発工数の見積りが難しい	過大見積りにより、発生工数に比べて過大な契約額となる場合がある。	過小見積りにより、トラブルが発生しなかったとしても適正な対価が得られない場合がある。
工期が不明確	<ul style="list-style-type: none"> ・契約期間が決まらないため、発注側メンバのアサインができない。 ・システム稼動時期が想定できないリスクがある。 	契約期間が決まらないため、開発メンバのアサインができない。
ユーザとベンダの連携が必要	<ul style="list-style-type: none"> ・発注側と受注側の責任分解点を明確にできないため、開発したシステムに不具合が発生したときの責任の所在が不明確になるリスクがある。 ・発注側メンバから受注側メンバへの指示が職業安定法や労働者派遣法に抵触する可能性がある。 	発注側と受注側の責任分解点を明確にできないため、開発したシステムに不具合が発生したときの責任の所在が不明確になるリスクがある。
ユーザとベンダのコミュニケーションが必要	発注側メンバから受注側メンバへの指示が職業安定法や労働者派遣法に抵触する可能性がある。	—

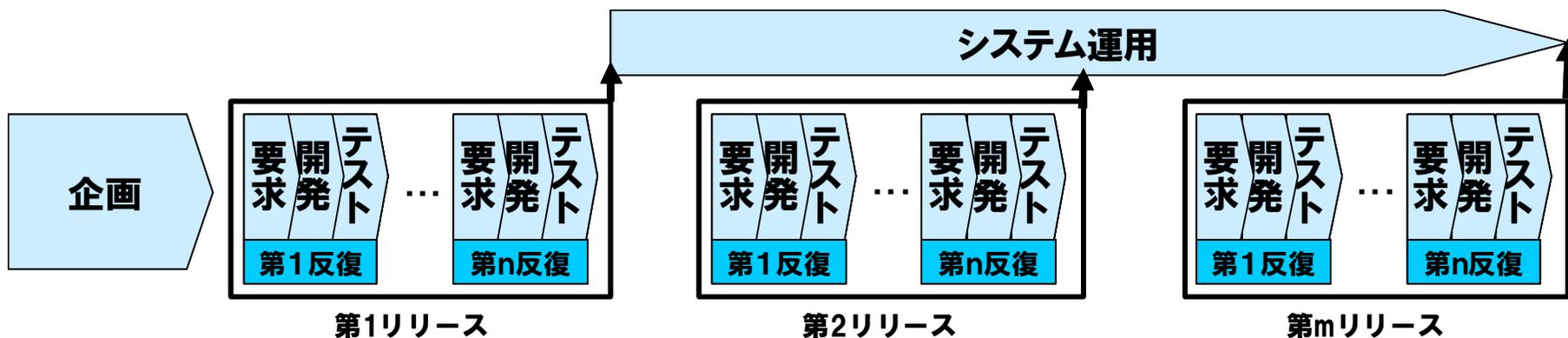
■平成21年度に実施された契約方式調査による契約事例から導かれた契約の問題点

ウォーターフォール型開発における準委任契約適用時の問題点

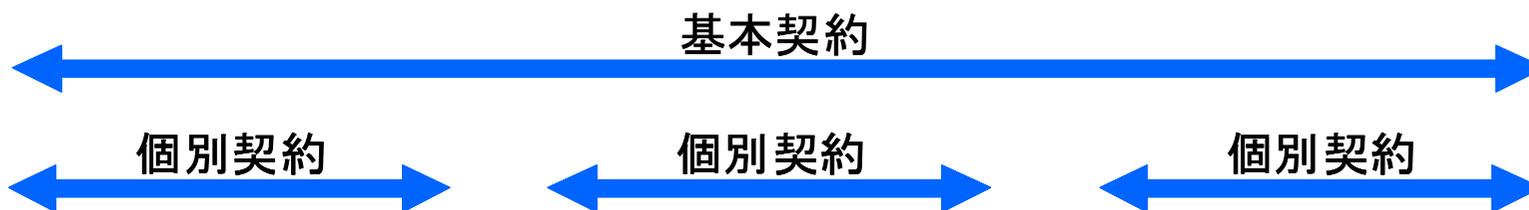
問題点(リスク)	発注側	受注側
要件が未確定	<ul style="list-style-type: none"> 要件の確定に時間がかかると、その分委託費用が増加する可能性がある。 成果物に不具合が発生した場合に、製造上の瑕疵に相当する内容であっても、受注者側に責任を取らせることができない。 	—
成果物が不明確	<ul style="list-style-type: none"> 要件の確定に時間がかかると、その分委託費用が増加する可能性がある。 成果物に不具合が発生した場合に、製造上の瑕疵に相当する内容であっても、受注者側に責任を取らせることができない。 	—
性能と品質が不明確	工期の延長は、委託費用の増加につながるため、予算超過となる場合がある。	—
開発工数の見積りが難しい	—	—
工期が不明確	委託費の額が想定できないため、予算超過になる場合がある。	—
ユーザとベンダの連携が必要	発注側メンバから受注側メンバへの指示が職業安定法や労働者派遣法に抵触する可能性がある。	—
ユーザとベンダのコミュニケーションが必要	発注側メンバから受注側メンバへの指示が職業安定法や労働者派遣法に抵触する可能性がある。	—

■ 基本契約/個別契約モデル契約

基本契約/個別契約モデル契約のアジャイル開発プロセスモデル

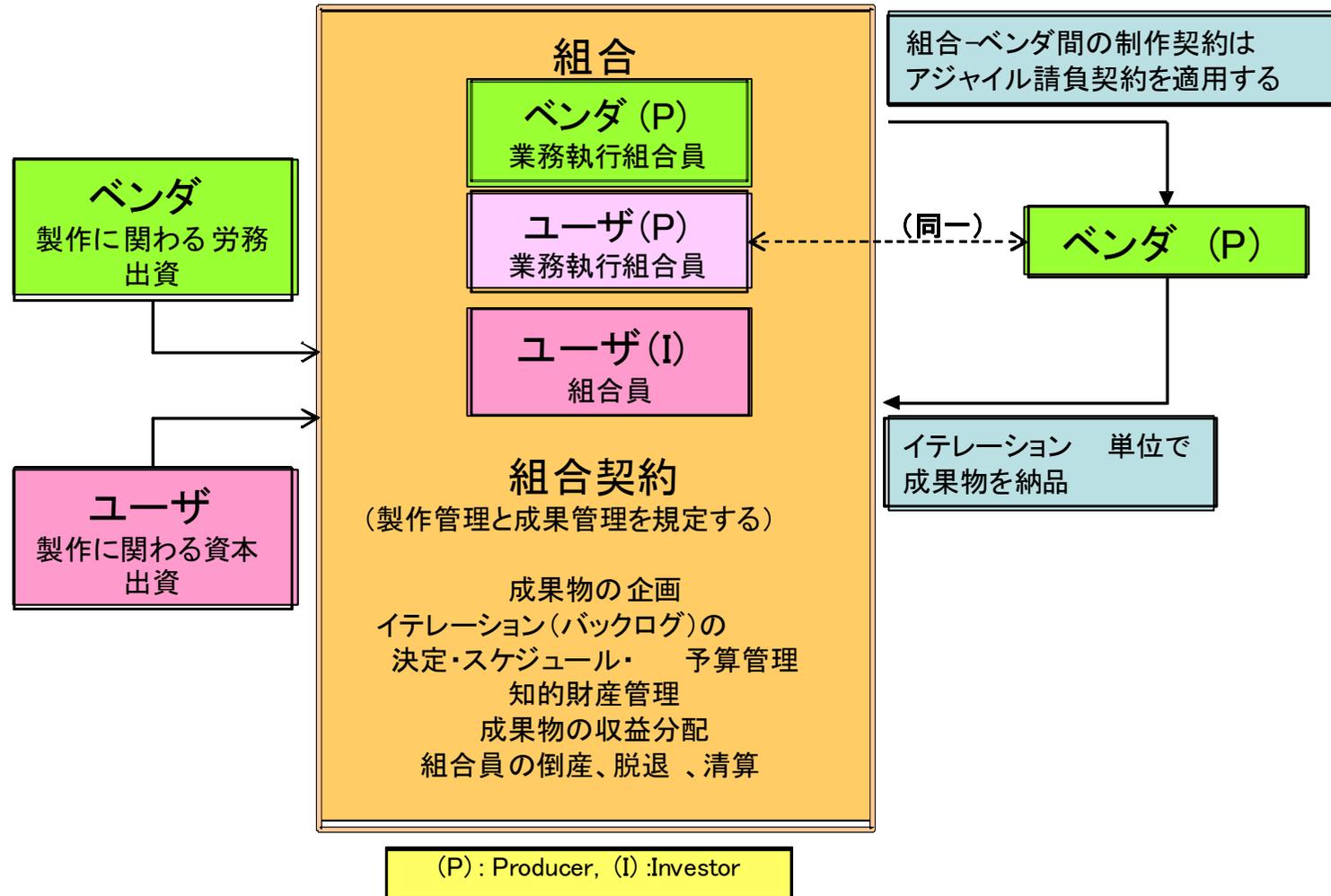


- n=1のケースもあり。



■ 組合モデル契約

組合モデル契約書(共同事業モデル)案のイメージ



- 本報告書では、非ウォーターフォール開発WG発足の背景、目的を述べ、そのコンテキストの中で「非ウォーターフォール型開発」および「アジャイル開発」を定義した。
- 非ウォーターフォール開発WGでは、現在の日本の産業構造の問題点を認識した上で、日本のソフトウェア競争力を高めるために、および、エンジニアがいきいきと働ける環境を作るために、非ウォーターフォール開発手法を位置づけたいと考えている。
- 本報告書では、アジャイル開発はすべての領域で最も優れた手法である、という立場をとっていないことに注意して欲しい。アジャイル開発には得意領域があり、その領域が現代の不確実性の高いビジネス環境の中で日々に拡大しているという認識である。
- さらに、経営層への理解浸透が、次の一步への大きな要素であるという認識から、企業情報システム開発における考え方のパラダイムシフトの意義を考察し、アジャイルへの懸念事項とともにまとめた。

参考：非ウォーターフォール型開発に関する検討委員

■非ウォーターフォール型開発に関する検討は、次の委員から構成された。

非ウォーターフォール型開発WG		
	氏名	所属
主査	松本 吉弘	財団法人京都高度技術研究所 顧問
委員	稲村 直穂子	株式会社ディー・エヌ・エー システム統括本部 本部長
	大槻 繁	株式会社一 コンサルティンググループ 副社長
	合田 治彦	富士通株式会社 システム生産技術本部 本部長代理
	田澤 久	楽天株式会社 開発ユニット 開発環境整備課 課長
	戸村 元久	株式会社NTTデータ 技術開発本部 プロジェクトマネジメント・イノベーションセンタ センタ長
	羽生田 栄一	株式会社豆蔵 取締役
	平鍋 健児	株式会社永和システムマネジメント 副社長、 株式会社チェンジビジョン 代表取締役
	広瀬 敏久	日本電気株式会社 主席技術主幹
	古川 正伸	株式会社東京証券取引所 品質管理部 課長
	前川 徹	サイバー大学 IT総合学部 教授
	馬嶋 宏	株式会社日立製作所 情報システム事業部 方式設計センタ
	松島 桂樹	武蔵大学 経済学部 教授
	南 悦郎	新日鉄ソリューションズ株式会社 技術本部システム研究開発センター 所長
和田 憲明	富士通株式会社 システム生産技術本部 SI生産革新統括部 SDEM推進部	
エキスパート	伊久美 功一	IPA/SEC専門委員／元IPA/SEC研究員
オブザーバ	鴨田 浩明	経済産業省

参考：非ウォーターフォール型開発に関する検討委員

■非ウォーターフォール型開発に関する検討は、次の委員から構成された。

開発モデルPT		
	氏名	所属
リーダー	平鍋 健児	株式会社永和システムマネジメント 副社長、 株式会社チェンジビジョン 代表取締役
委員	合田 治彦	富士通株式会社 システム生産技術本部 本部長代理
	戸村 元久	株式会社NTTデータ 技術開発本部 プロジェクトマネジメント・イノベーションセンタ センタ長
	広瀬 敏久	日本電気株式会社 主席技術主幹
	古川 正伸	株式会社東京証券取引所 品質管理部 課長
	前川 徹	サイバー大学 IT総合学部 教授
	松島 桂樹	武蔵大学 経済学部 教授
	南 悦郎	新日鉄ソリューションズ株式会社 技術本部システム研究開発センター 所長
	和田 憲明	富士通株式会社 システム生産技術本部 SI生産革新統括部 SDEM推進部
エキスパート	伊久美 功一	IPA/SEC専門委員／元IPA/SEC研究員

■非ウォーターフォール型開発に関する検討は、次の委員から構成された。

技術・スキルPT		
	氏名	所属
リーダー	松島 桂樹	武蔵大学 経済学部 教授
委員	天野 勝	株式会社永和システムマネジメント コンサルティングセンター センター長
	川端 光義	アジャイルウェア代表
	児玉 公信	株式会社情報システム総研 取締役副社長／モデラー
	田澤 久	楽天株式会社 開発ユニット 開発環境整備課 課長
	馬嶋 宏	株式会社日立製作所 情報システム事業部 方式設計センタ
	和田 憲明	富士通株式会社 システム生産技術本部 SI生産革新統括部 SDEM推進部

参考：非ウォーターフォール型開発に関する検討委員

■非ウォーターフォール型開発に関する検討は、次の委員から構成された。

契約問題PT		
	氏名	所属
リーダー	前川 徹	サイバー大学 IT総合学部 教授
委員	飯塚 顕治	新日鉄ソリューションズ株式会社 法務・知財財産部 法務グループシニアマネージャー
	高橋 雅宏	アジャイルプロセス協議会 見積・契約ワーキンググループ
	板東 直樹	アップデートテクノロジー株式会社 代表取締役社長
	平野 高志	ブレークモア法律事務所 弁護士
	古川 正伸	株式会社東京証券取引所 品質管理部 課長
エキスパート	梅本 大祐	IPA/SEC専門委員／ブレークモア法律事務所 弁護士
オブザーバ	鴨田 浩明	経済産業省
	葛山 弘揮	経済産業省
	下田 裕和	経済産業省
	柳橋 祥人	経済産業省
	新保 康夫	日本コンピューター・システム株式会社 事業推進本部
	鈴木 律郎	社団法人情報サービス産業協会 企画調査部
	茂木 智美	社団法人情報サービス産業協会 企画調査部／審査業務部