FUJITSU
shaping tomorrow with you

通信ソフトウェア開発 「短期リリース」への道 ～オフショア適用にも挑戦～

富士通株式会社
兎耳山 俊吾

2015年10月22日

Copyright 2015 FUJITSU LIMITED

目次

Copyright 2015 FUJITSU LIMITED

当社ソフト開発部門の特長

FUJITSU

通信キャリア向けソフトウェア開発部門

- 特長
  - 仕様書は顧客から提示
  - 大規模開発（100人超えあり）
  - 生産物・品質に対する要求が厳格
  - 開発期間が半年～１年間
  - オフショアを積極的に利活用

『ウォーターフォール型開発』が得意

Copyright 2015 FUJITSU LIMITED

今回の開発ソフトウェアについて

FUJITSU

ＮＦＶ※1向け富士通基盤ソフト製品
ＮＦＶソフトウェアの開発要件は？

※1 NFV＝Network Functions Virtualization

① NFV標準規約がコンセプト定義に留まっている
② 顧客やベンダーに具体的な製品イメージがない
　⇒ 概念検証（PoC※2）で製品イメージ具体化が必要
③ 開発コスト削減も必要

※2 POC＝Proof Of Concept

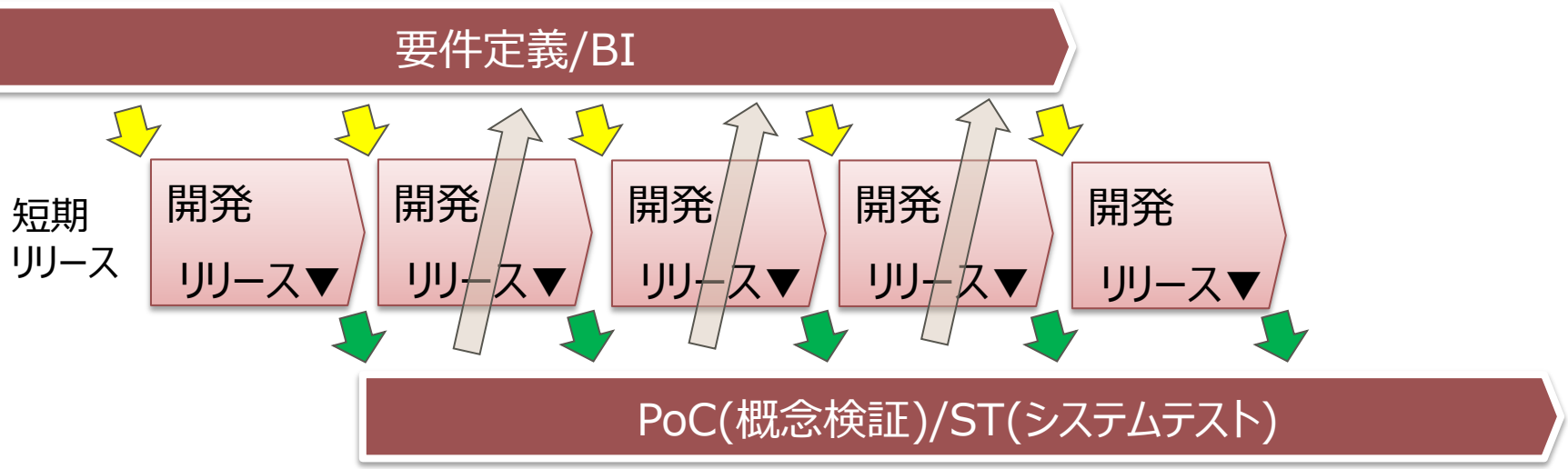
適用すべき開発プロセスは↓
- 『短期リリース』による繰り返し開発
- 『オフショア開発』によるコスト削減

Copyright 2015 FUJITSU LIMITED

短期リリースに向けて（1）

FUJITSU

今回開発の『短期リリース』イメージ



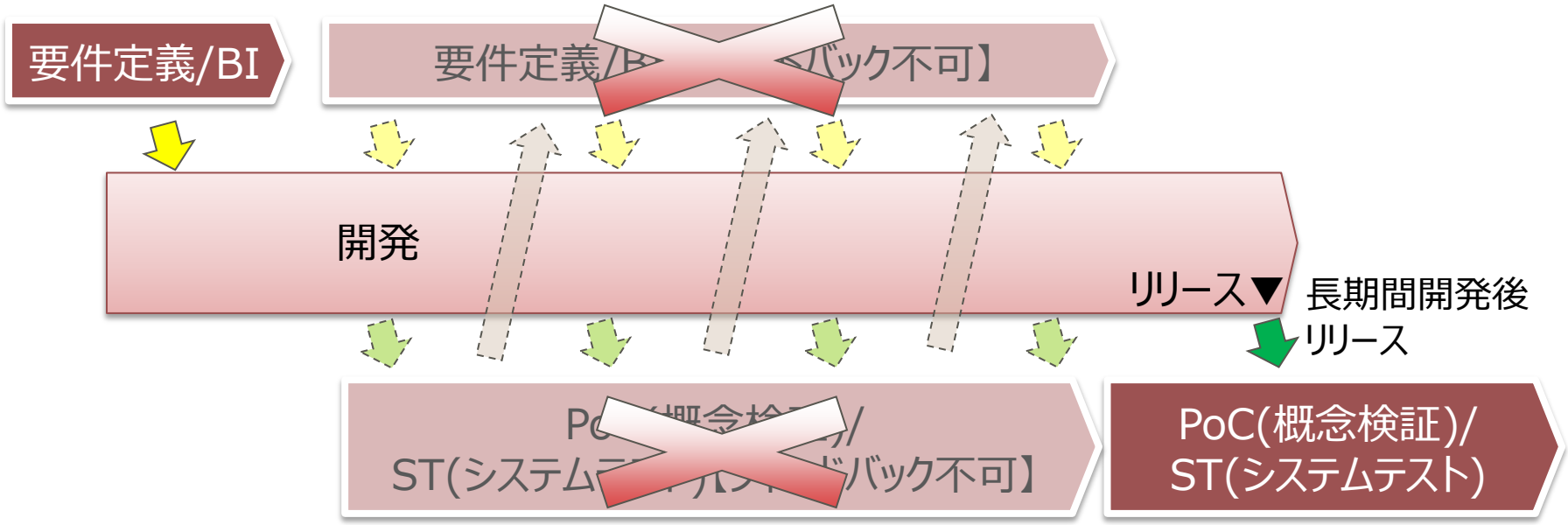
『短期リリース』＝『新しい開発プロセス適用』必要？

Copyright 2015 FUJITSU LIMITED

短期リリースに向けて（2）

FUJITSU

ウォーターフォール型開発を適用すると…



製品リリース／フィードバックは開発最後に一度だけ

Copyright 2015 FUJITSU LIMITED

短期リリースに向けて（3）

FUJITSU

『短期リリース』＝『アジャイル開発』適用では？

- 『アジャイル開発』事例を調査（社内）

  『アジャイル開発』適用プロジェクトの特長

  (1) 顧客と一体となった開発

  (2) 開発規模が小さい

  (3) 大部屋開発の適用

『オフショア』で『アジャイル開発』は
適用しづらい特長あり

『オフショア』で『短期リリース』を実現するためには
どんな開発プロセスが最適か？

Copyright 2015 FUJITSU LIMITED

短期リリースに向けて（4）

FUJITSU

オフショアで短期リリースを実現するため、開発プロセス選択を試行

作業項目	選択開発プロセス定義
工程定義	V字モデル定義 ⇔ 工程定義レス
生産物定義	工程毎定義 （↓KPT見直し後） ⇔ 文書レス（ソース命）
品質保証	工程毎品質保証 ⇔ イテレーション内品質保証
要件定義	全要件を確定 ⇔ リリース毎要件見直し
開発拠点形態	海外拠点と協業 ⇔ 国内大部屋開発

Copyright 2015 FUJITSU LIMITED

短期リリースに向けて（5）

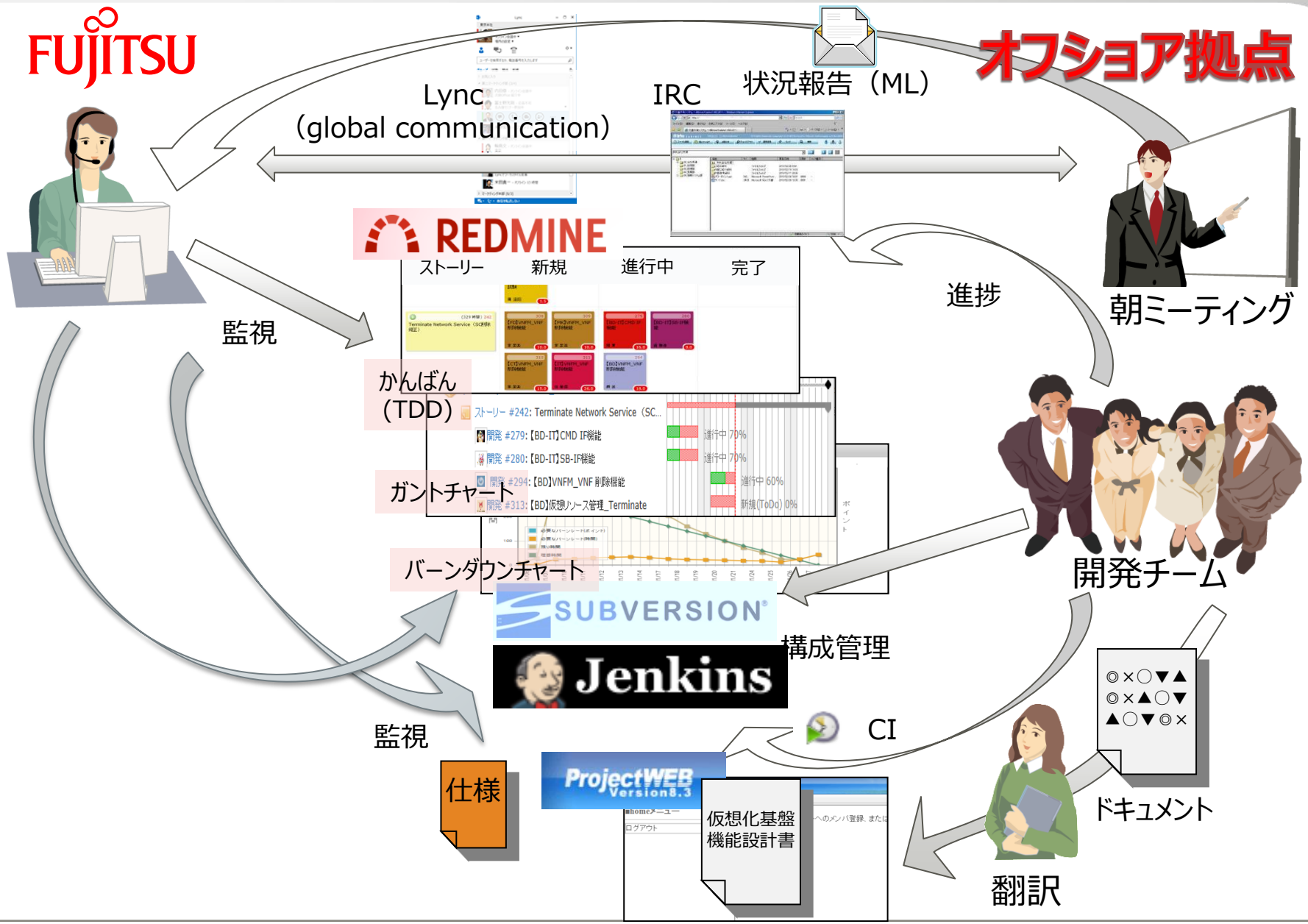
FUJITSU

今回選択した開発プロセス定義とその理由

作業項目	選択定義	開発プロセス定義選択理由
工程定義	V字モデル定義	品質／検収部門に対し、従前の開発結果提示が必要だったため
生産物定義	文書レス	短期リリースのため生産物作成量を最小限にするため
品質保証	双方取込み	品質保証はイテレーション内で2回実施、各工程毎の品質保証は実施しないため
要件定義	リリース毎 要件調整	短期リリースのリリース結果を要件定義にフィードバックするため
開発拠点形態	海外拠点と協業	海外拠点（オフショア）開発優先のため

Copyright 2015 FUJITSU LIMITED

適用したソフトウェア開発ツール



Copyright 2015 FUJITSU LIMITED

適用した開発手法

FUJITSU

- 今回の開発では以下の開発手法を適用
- Rel.1で定義した開発手法について適用結果をフィードバックし、以降のリリースにおいて開発手法を改善した

	開発手法	Release1	Release2	Release3	Release4	Release5
スクラム	朝会	○	○	○	○	○
	かんばん	○	○	○	○	○
	スプリント	1ヵ月	2週間 → ×	1ヵ月	1ヵ月	1ヵ月
	バックログ	○	○	再整備	○	○
	受入試験	–	試行(デモ)	機能試験	機能試験	機能試験
XP	ペアプロ	–	–	–	–	○
	CI	CT	CT/IT(30%)	CT/IT	CT/IT	CT/IT
	ドキュメント	削減 → ×	有	有	有	有
	工程定義	BD+FD → ×	W.F.	W.F.	W.F.	W.F.

Copyright 2015 FUJITSU LIMITED

開発プロセスの改善例（1）

FUJITSU

生産物定義（開発プロセス定義）の見直し

■初期イテレーション…生産物（ドキュメント）は最小限の作成
⇒仕様関連問題が発生
⇒仕様書を確認…関連する仕様書が作成されていない
⇒問題分析ができない（一体どうやって開発してたのか?）

■後半…作成する生産物を見直し、必要なドキュメントは作成する
⇒特に上流ドキュメントは漏れなく作成
⇒仕様関連の問題はほぼ無くなる（フィードバック効果あり）

Copyright 2015 FUJITSU LIMITED

開発プロセスの改善例（2）

FUJITSU

プロダクトバックログ調整作業の精度向上

■前半イテレーション…初めてのアジャイル（まずはチャレンジ）
⇒手さぐり状態でバックログの見積、タスク分割を実施
⇒バーンダウンチャートが右肩上がり（期間に対しタスクが増加）
⇒見積時分割タスク粒度が粗くイテレーション内で再分割を実施
⇒開発中のタスク追加（追加が必要と判断、即時追加）

■後半…KPTで調整作業見直し、フィードバックを実施
⇒タスク分割粒度細分化の徹底
⇒追加タスクの優先度により次イテレーション回しの調整
⇒イテレーション内作業量調整の精度向上＝見積精度が向上
⇒開発プロセスの継続とフィードバックで作業練度が向上

Copyright 2015 FUJITSU LIMITED

開発プロセスの改善例（3）

FUJITSU

Lyncを活用したオフショア拠点間コミュニケーションの内容改善

通知情報が
正しく理解されてない！

■初期イテレーション
⇒海外拠点担当とコミュニケーションミスが多発
⇒レガシーな情報交換(ML,IRC(InternetRelayChat))使用
⇒LyncはInstantMessenger機能と電話会議機能を使用
⇒上記手段では、詳細な情報が即時反映されにくい

■後半…Lyncのデスクトップ共有機能を使用開始
⇒拠点の担当者間で直接資料の共有が可能
⇒打合せ時、資料の変更内容を即時反映
⇒オフショア拠点と詳細なコミュニケーションが実現可

Copyright 2015 FUJITSU LIMITED

開発振り返り（1）

FUJITSU

- 成功点
  - 短期間リリース　一ヶ月毎リリースを厳守！
    ⇒特に適用したソフトウェア開発ツールが有効
  - 開発変化に柔軟に対応
    ⇒顧客動向PoC検証結果のフィードバックが有効
    ⇒次イテレーションへの仕様追加件数＝３９件
- 気づいた点
  - 開発効率はWFとほぼ同等
    ⇒WFプロセスを適用したため
  - 従前の品質評価が可能
    ⇒WFプロセスを適用したため

Copyright 2015 FUJITSU LIMITED

開発振り返り（2）

FUJITSU

課題

- 開発メンバー内「アジャイル開発すれば、楽できるはずなのに…」という思い違いが発生
  ⇒実際はWF作業が多く、開発プロセスに疑問を持つ者も
  ⇒開発メンバーへの開発プロセス定義共有が不徹底
  ⇒メンバー全員の開発プロセスの正しい理解が重要
- 自動試験がうまく機能しなかった（試験項目の約１５％で適用）
  ⇒自動試験システムは環境構築／手順作成に手間がかかる
  ⇒自動試験導入メリットの意識向上が必要
- 品質指標の見直しが大変…
  ⇒今開発はイテレーション（一ヶ月）毎の見直し作業発生
  ⇒イテレーション毎の品質指標見直し妥当性を再検討

Copyright 2015 FUJITSU LIMITED

今後の開発にあたり（1）

FUJITSU

- 開発プロセスの改善
  - 自動試験適用範囲の拡大
    ⇒対象は結合試験／リグレッション試験（今回未対応）
    ⇒開発プロセス内手作業の排除（品質向上と工数削減）
  - リリース期間短縮に向けた施策（1month⇒2week）
    ⇒リリース計画時のタスク見積精度の向上
    ⇒開発プロセスと生産物量の見直し（できるだけ削減）
    ⇒下記顧客納品物との調整
- 契約プロセスの連携改善
  - 生産物定義に対する顧客納品物の理解度向上
    ⇒上記開発プロセスに適合した生産物定義での承認
    ⇒契約期間の定義（開発リリース毎／複数リリース後）

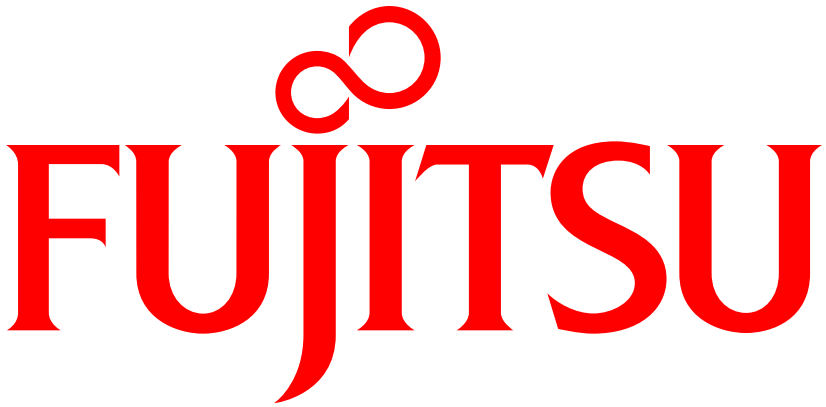
Copyright 2015 FUJITSU LIMITED

今後の開発にあたり（2）

FUJITSU

- 品質保証プロセスの連携改善
  - 工程完了判定および開発完了判定条件の定義精査
    ⇒完了判定条件（品質指標／バグ密度、等）定義に対して
    ⇒開発量ではなくタスク単位の品質指標の作成

Copyright 2015 FUJITSU LIMITED



shaping tomorrow with you