Speed
&Perfect
+Intelligence

現場からの障壁克服

～　XDDP を導入する際の障壁と
その克服に向けたアプローチ　～

アンリツエンジニアリング株式会社
事業本部プロトコルシステム技術部
なん　ぶ　　　たゆ　な
南部　妙水

自己紹介

- **アンリツエンジニアリング株式会社**
  - 1981年 アンリツ株式会社100%出資子会社として設立
  - アンリツ製 測定関連計測器 を中心に
    - 基板設計，FPGA開発
    - ファームウェア，ソフトウェア開発
    - 関連システム開発，エンジニアリングサービス
- **プロトコルシステム技術部**
  - 携帯電話関連，IP通信関連の測定機器ソフト（ファーム含），関連パッケージソフト

機能追加，移植，etc…
派生開発がほとんど

- **南部妙水**（なんぶ たゆな）（入社9年目，役職無しのSE）
  - 携帯基地局シミュレータMD8470Aのソフトウェア開発 （約4年）
  - 携帯電話のテスト用パッケージソフト開発（約3年）

※ Windowsアプリケーションばかり

ことのはじまり

- **入社5年目「やり方が拙いんじゃないか」と思い始める**
  - 設計文書の書き方を工夫しても…
  - テストの技法を取り入れ続けても…
  - レビューを頑張っても…
  - リスクを考慮した計画を立てても…

もっっっと
楽したい！
楽できるはず！

でも
なにをどうすれば？？？

- **SQiP参加者募集に飛びつく** (2009年)
  - 「もー，何でも良いから“違う”情報が欲しいよー」
  - 「派生開発？　聞いたことある…」
  - 「私の会社，派生開発ばっかりだよね…　行ってみよう」

当たって砕けました

- 「これぞ私の求めていた解！」なんですけど…
  - SQiP分科会で XDDPを学ぶ
  - やっぱりやり方が拙かったので
    もっと良いやり方 ＝ XDDP にしよう！

XDDPを使ってから
研究テーマを決める
はずだった

- XDDPを導入するのは困難だった
  - チーム誰ひとりとしてXDDPを始めることができなかった

これ，よくある問題なのではないか？

  - 研究テーマ変更
    「派生開発にXDDPを導入する際の障壁と
    その解消に向けたアプローチ」

- 論文はできあがったけど

目的はそれじゃないのです

挫折の内訳

- **自分の参加するプロジェクト**
  - 既存の社内標準プロセスで**「動いている」**
  - 途中でプロセスを変更することは**「ありえない」**

変更する方法も実績もない
確かに危険だと思う

- **他のプロジェクト**
  - 部門内サイトで**XDDP**紹介記事を投稿したが**「無反応」**

良さが伝わらない？
信用できない？
問題意識の違い？

挫折の内訳

- **自分の参加するプロジェクト**
  - 既存の社内標準プロセスで**「動いている」**
  - 途中でプロセスを変更することは**「ありえない」**

変更する方法も実績もない
確かに危険だと思う

最初からプロジェクトに参加する
チャンスを狙おう

- 他のプロジェクト
  - 部門内サイトで**XDDP**紹介記事を投稿したが**「無反応」**

良さが伝わらない？
信用できない？
問題意識の違い？

「やればわかる」？
卵が先か鶏が先か

自分だけでも
楽になれないかなぁ…

相手の気持ちになってみた

- 新しいこと，ルール外のことをするのはイヤ
  - 折衝が必要になる
  - 本当に成功するのか？
  - 新しいことを覚えないといけない
  - 作業が増える
- 「まぁ，いいんじゃない？」と言ってもらうには
  - 違うプロセスに見えないこと
    - 社内標準プロセスの範囲内で可能
    - 次頁で説明
  - 見慣れない文書に見えないこと
    - 「改造仕様書」って呼ぼう
    - 書くのが難しくないこと
      - サンプル用意する？
      - 作業しながら説明すればマシ？

いままでのやりかた

Speed &Perfect +Intelligence

機能設計

要求の詳細を詰めつつ
関連規格を調査しつつ
仕様を考え文書を変更する

機能仕様書(改版)

構造設計

設計書の妥当性を調査しつつ
関連箇所の仕様を調査しつつ
設計を考え文書を変更する

構造設計書(改版)

モジュール設計

設計書の妥当性を調査しつつ
実現方法を調査しつつ
設計を考え文書を変更する

モジュール設計書(改版)

実装

関係箇所を調査しつつ
実装方法を考えつつ
ソースコードを変更する

ソースコード

テスト

空いている時間に各文書にBug fixを反映

やっていることに注目(1)

Speed &Perfect +Intelligence

機能設計

要求①の詳細を詰めつつ
関連規格を**調査**②しつつ
仕様③を考え文書を変更する

機能仕様書(改版)

構造設計

設計書の妥当性を**調査**②しつつ
関連箇所の仕様を**調査**②しつつ
設計を考え文書を変更する

構造設計書(改版)

モジュール設計

設計書の妥当性を**調査**②しつつ
実現方法を**調査**②しつつ
設計を考え文書を変更する

モジュール設計書(改版)

実装

関係箇所を**調査**②しつつ
実装方法を考えつつ
ソースコードを変更する

ソースコード

テスト

空いている時間に各文書にBug fixを反映

機能設計を整理

Speed &Perfect +Intelligence

機能設計

調査

要求決め

仕様決め

調査報告書

変更要求仕様書

構造設計

~~設計書の妥当性を調査しつつ~~

~~関連箇所の仕様を調査しつつ~~

設計を考え文書を変更する

構造設計書(改版)

実装

~~関係箇所を調査しつつ~~

実装方法を考えつつ

ソースコードを変更する

ソースコード

モジュール設計

~~設計書の妥当性を調査しつつ~~

~~実現方法を調査しつつ~~

設計を考え文書を変更する

モジュール設計書(改版)

テスト

空いている時間に各文書にBug fixを反映

やっていることに注目(2)

Speed &Perfect +Intelligence

機能設計

調査

要求決め

仕様決め

調査報告書

変更要求仕様書

構造設計

~~設計書の妥当性を~~調査~~しつつ~~

~~関連箇所の仕様を~~調査~~しつつ~~

④

設計を考え文書を変更する

構造設計書(改版)

実装

~~関係箇所を~~調査~~しつつ~~

⑤

実装方法を考えつつ

ソースコードを変更する

ソースコード

モジュール設計

~~設計書の妥当性を~~調査~~しつつ~~

~~実現方法を~~調査~~しつつ~~

④

設計を考え文書を変更する

モジュール設計書(改版)

テスト

空いている時間に各文書にBug fixを反映

構造・モジュール設計を整理

Speed &Perfect +Intelligence

機能設計

調査

要求決め

仕様決め

調査報告書

変更要求仕様書

構造設計

設計

変更設計書

Traceability Matrix

実装

~~関係箇所を調査しつつ~~

~~実装方法を考えつつ~~

ソースコードを変更する

ソースコード

モジュール設計

~~設計書の妥当性を調査しつつ~~

~~実現方法を調査しつつ~~

設計を考え文書を変更する

テスト

⑥

空いている時間に各文書にBug fixを反映

XDDPと同じになった！

Speed &Perfect +Intelligence

機能設計

調査

要求決め

仕様決め

調査報告書

変更要求仕様書

構造設計

設計

変更設計書

Traceability Matrix

実装

~~関係箇所を~~調査~~しつつ~~

~~実装方法を考えつつ~~

ソースコードを変更する

ソースコード

モジュール設計

設計書

実現方

設計を

機能仕様書(改版)

構造設計書(改版)

モジュール設計書(改版)

テスト

既存文書に変更を反映

準備万端待っていたら

- **ちょうど舞い込んできた知見プロジェクト**
  - 3名で1ヶ月半，顧客要求は機能追加1件の小規模派生開発
  - 各人1オブジェクトを担当
    ＝ 自分が担当する部分ならば好きな方法を採用できる

XDDPはモジュール単位で適用できるし
自分ひとりなら「やるだけ」！

結果

XDDPで良かった！

- 調査　アテにしてたライブラリの機能が未提供
- 変更要求仕様書　細かい変更がポロポロ　→ **出戻りゼロ**
- 変更設計書　コード変更が見えるまで設計　→ **バグゼロ**
- 工数・工期は少し余った

自信満々で機会を窺っていたら

XDDPにしないと
失敗しそう...

- 派生開発プロジェクトに最初から参加することに
  - 前任者不在
  - 設計文書がひとつもない
  - ドメイン有識者は超多忙

従来の方法ができない
→ これはチャンス！

提案

- **“改造仕様書”で開発しませんか**

1から仕様書を作っている時間ありませんよ

ついでに改造仕様書の書式をちょっと変えたいんですが

まずは調査から始めないと

ついに **XDDPの導入に成功**

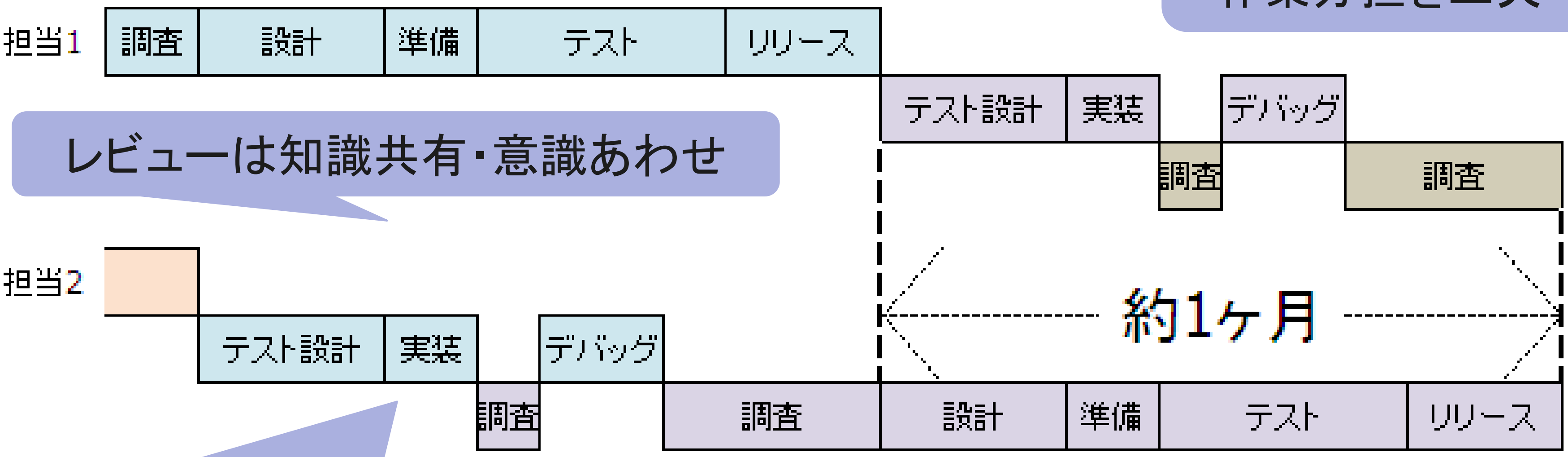
XDDPでうまくいった

- Minor version up 1回
- Major version up 1回
  - Beta版 5回 (内 4回は1ヶ月 毎にリリース)

「変更箇所に集中」で遅延を防げた

変更設計書で若手でも平気

時間が足りない！
作業分担を工夫



レビューは知識共有・意識あわせ

担当者を変えて「思い込み」をチェック

従来の方法では
間に合わなかったと思う

自慢してみた

Speed &Perfect +Intelligence

- 「事例発表会」という企画
  - 2011年度中に各人2回強制参加
  - 中堅どころを所属関係なく10人弱のグループに分け
  - 1回 2グループ で
  - 1名/グループが「事例」を発表 → ディスカッション

分析能力，発表能力等を鍛える目的

- SQiPで行った研究について発表

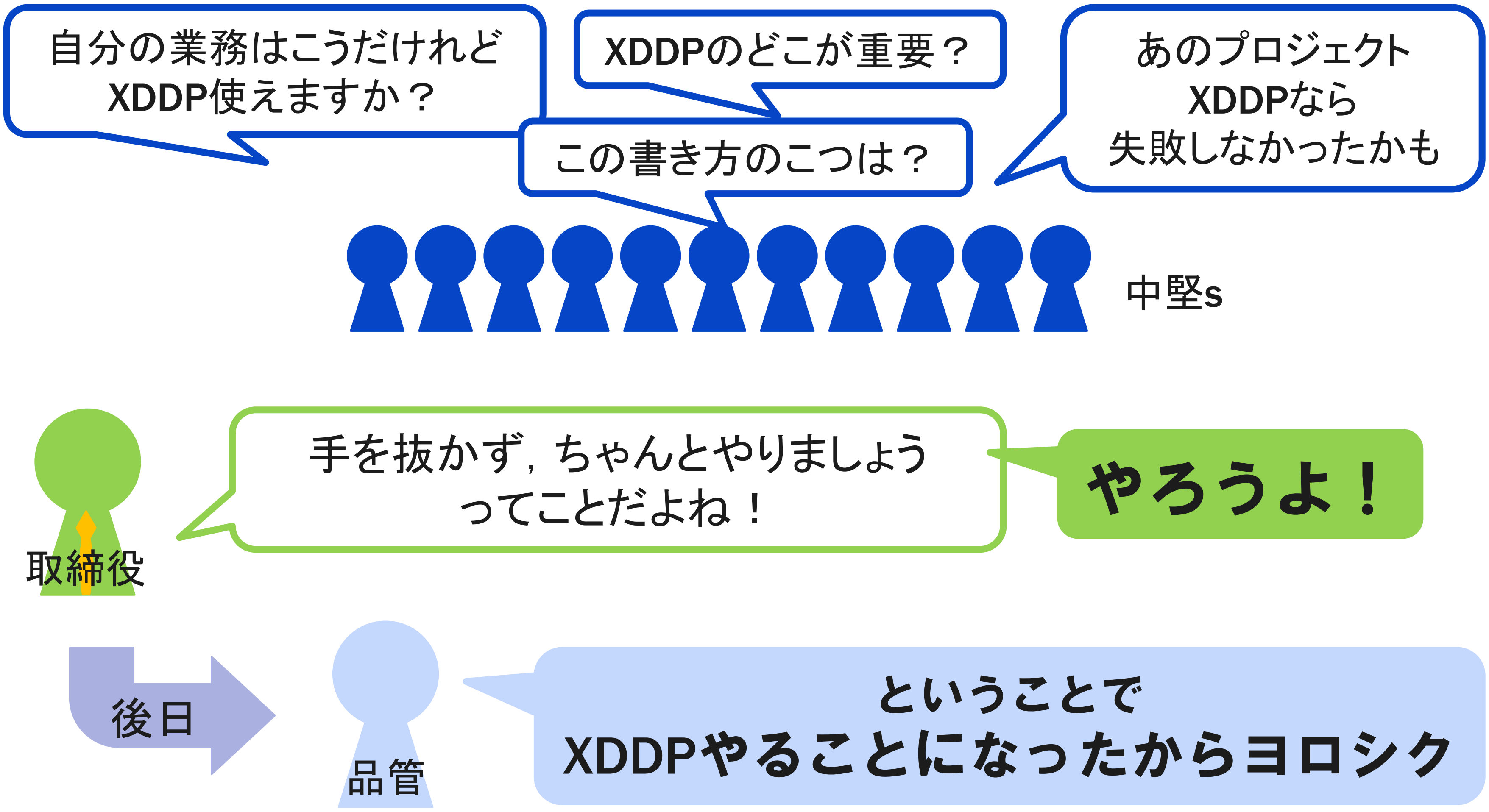
するためにXDDPの説明を

4ヶ月連続リリースできたと宣伝

したら

XDDPがウケた！

XDDPがウケた


自分の業務はこうだけれど
XDDP使えますか？
XDDPのどこが重要？
この書き方のこつは？
あのプロジェクト
XDDPなら
失敗しなかったかも
中堅s
手を抜かず，ちゃんとやりましょう
ってことだよね！
やろうよ！
取締役
後日
品管
ということで
XDDPやることになったからヨロシク

そして今

Speed &Perfect +Intelligence

- **まだ全社に「派生開発」「XDDP」を紹介している段階**
  - 社内報 (Web) に XDDPの紹介記事を連載
  - SQiP に継続的に人を出す方針が決まる

流行ってる感じではない

新しいものへの抵抗？

問題意識が低い？

周りの様子見？

お手伝いは喜んで♪
無理強いはなし

「自分だけでも楽になる」
で問題なし

全員でなくても
モジュール単位で可能

「実績が物を言う」のなら
XDDPを続けてれば，どんどん「やりたい人」が増えるハズ！

ご清聴ありがとうございました。