

# チケット駆動開発によるプロジェクトの活性化

ー 見える化と権限ポリシーがプロジェクトを変えた ー

株式会社SRA

阪井 誠

<sakai@sra.co.jp>

# 目次

- 効果的なチーム
- プロジェクトの概要
- チケット駆動開発
- 導入時の工夫
- 結果と効果
- まとめと感想

# 効果的なチーム

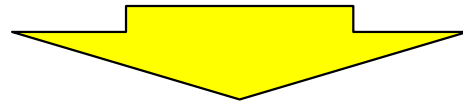
- チームで効果的に作業するには、メンバー全員が、
  - グループ内で協力的に作業し、
  - 問題の解決に参加し、
  - 必要なときに他のチームメンバーを助けることができなければならない

(W・S・ハンフリー：TSPガイドブック：リーダー編, p.60, 2007, 翔泳社.)

=>チケット駆動開発の運用ポリシーで実現した

# チケット駆動開発(TiDD)の背景

- 小規模かつ高機能なソフトウェアの開発
- オープン化に伴う環境の流動化
- ユーザ(ビジネス)要求の不安定化



- 細かく大量の作業、変化への対応が必要となり、プロセスの即時性・効率化が求められた
  - 抜けのない作業の実施
  - 予想外の事態への対応
  - 仕様変更への対応

# チケット駆動開発(TiDD)の歴史

- 2007年 まちゅ,「もうひとつのTDD」, ITpro Challenge のライトニングトーク  
<http://www.machu.jp/diary/20070907.html#p01>
- 2008年 XPJUG関西 ビジネスモデリング研究会からTiDD勉強会発足  
前田,「入門Redmine Linux/Windows対応」, 秀和システム  
岡本,「第1回Tracをオススメする, これだけの理由」, Tracで開発現場を交通整理  
<http://itpro.nikkeibp.co.jp/article/COLUMN/20080414/298973/>
- 2009年 あきぴー, XP祭り関西2009「チケットファーストでアジャイル開発！」  
<http://forza.cocolog-nifty.com/blog/2009/02/xp2009-746a.html>  
阪井,「TiDD: チケット駆動によるアジャイル開発法」, FORCE2009.  
[http://sakaba.cocolog-nifty.com/PDF/TiDD\\_FORCE09.pdf](http://sakaba.cocolog-nifty.com/PDF/TiDD_FORCE09.pdf)  
shun, 名古屋アジャイル勉強会 第13回「チケット駆動開発入門」  
<http://successdrills.com/modules/d3blog/details.php?bid=359>  
小川, 阪井,「チケット駆動開発 - BTSでExtreme Programmingを改善する-」  
<http://forza.cocolog-nifty.com/blog/2009/11/spes2009sqip200.html>
- 2010年 XPJUG, XP祭り関西2010「チケット駆動開発の体験談」ほか  
<http://www.xpjug.jp/>

# チケット駆動開発 (TiDD) とは

- 障害管理ツール(BTS/ITS)を使う
- チケットを用いて作業の管理をする
- チケットなしに(構成管理上の)更新をしない  
No Ticket, No Commit!
- ツール連携による自動化

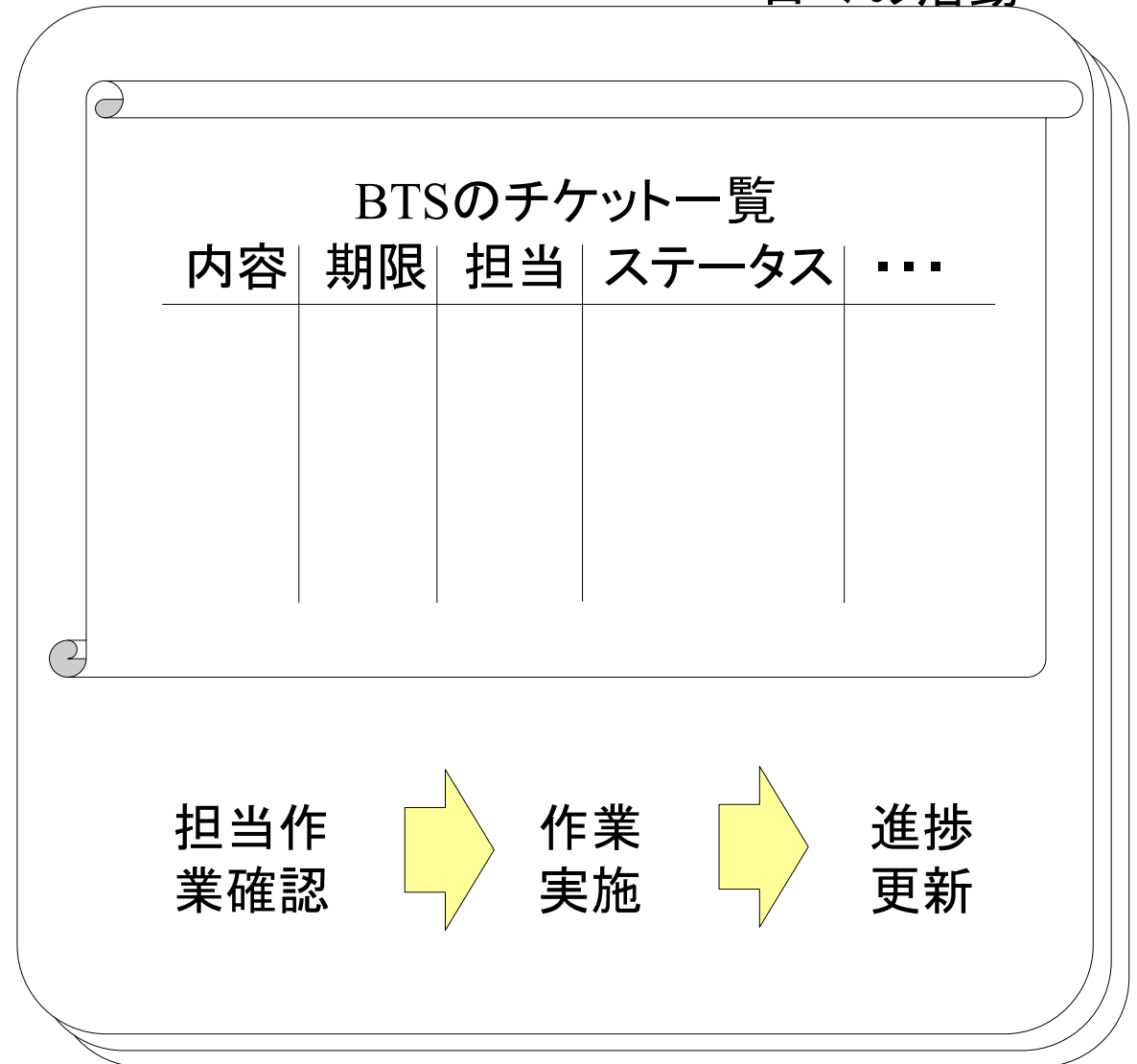
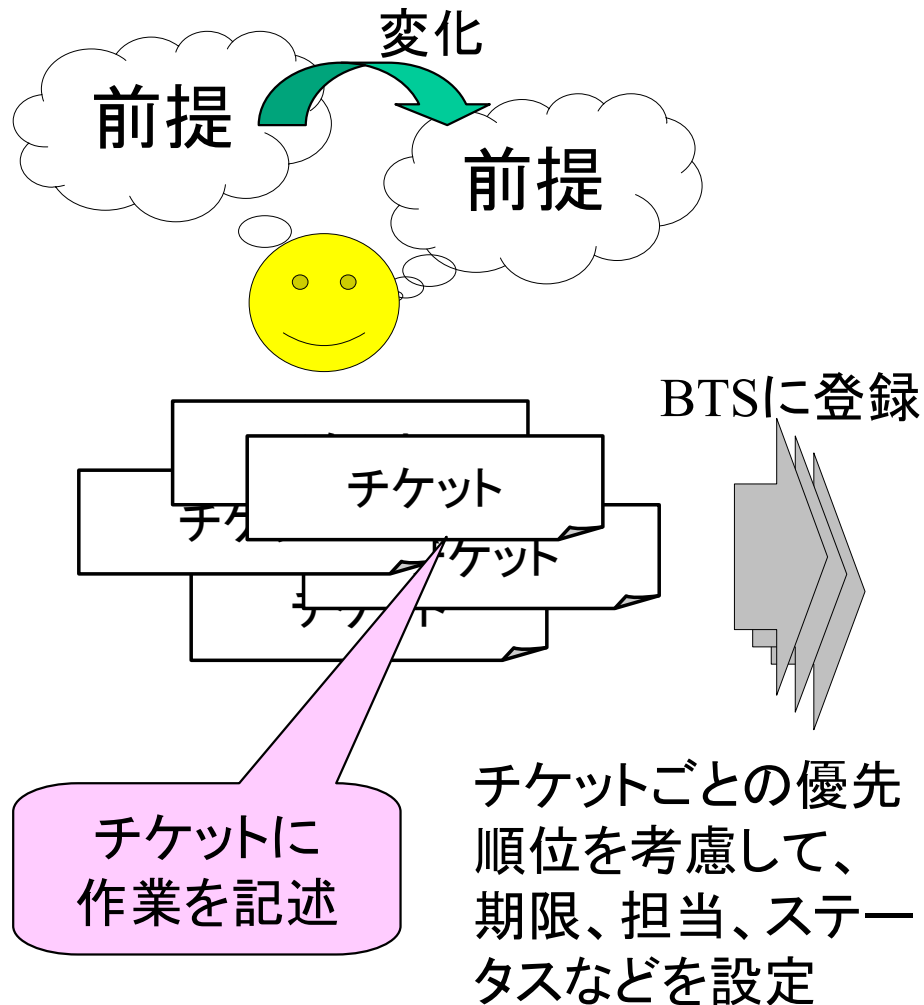
☆ 厳密な定義はなく、様々な実施方法がある

- 運用方式: 完全チケット方式 / 補完チケット方式\* / タスクカード方式
- 権限ポリシー: ワークフロー型 / オープン型\*

小川, 阪井: Redmineによるタスクマネジメント実践技法, 翔泳社, 2010.

# TiDD(Ticket Driven Development)

日々の活動



\* チケットはスクラムのバックログ、XPのタスクカードと同じ

# チケット駆動開発(TiDD)の効果

- ワークフローの管理
- 見える化
  - プロジェクト状況把握
  - 担当や重要度など様々な視点での作業管理
  - コミュニケーション・納得性向上
- ソースのトレーサビリティ向上
- ツール連携による自動化
- アジリティの向上



# プロジェクトの概要

- 文教パッケージのカスタマイズ(最大8人x1年)
  - 仕様の決定遅れ・変更 =>短納期
  - スキルは高いが経験者が少ない(リーダーは途中交代)
  - 体制と年齢の逆転(参加時期、BP)による遠慮
  - 初めての組み合わせ(サブシステム、ミドルのバージョン)
  - 義務感と不安、重苦しい雰囲気、閉塞感、、、
  - 守りに入るので、コミュニケーションが悪い

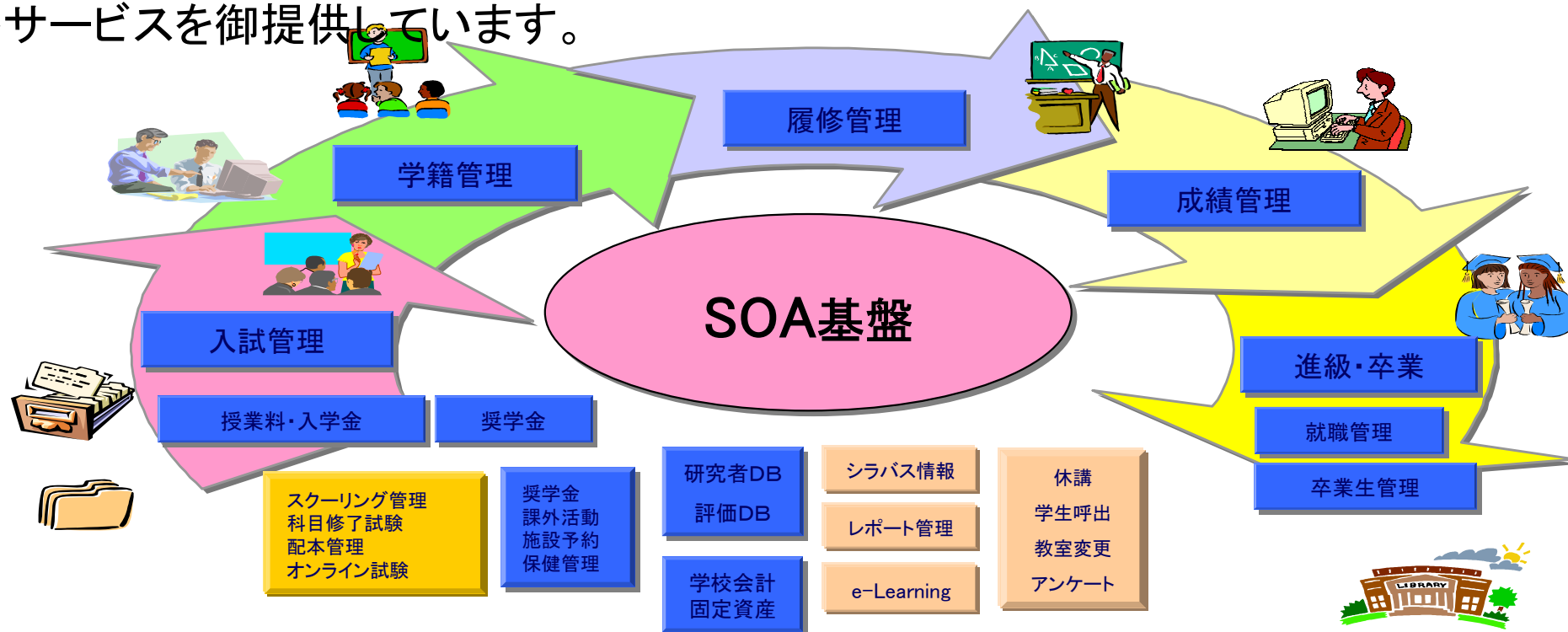
システムテストの時期になると、計画外の環境構築やリリース準備作業が明らかになった(環境に関連するバグも、、、)

⇒ チケット駆動開発導入を決意！

# 総合文教ソリューション UniVision

UniVisionは入試管理、学籍管理、教務管理といった基幹事務機能を中心に学生サービスに関する情報全般を統合的に管理する統合文教ソリューションです。

また、2009年度より、新たに法人系システム(学校会計、固定資産)もサービス開始しております。システムコンサルティングから構築、運用支援にいたるトータルサポートサービスを御提供しています。

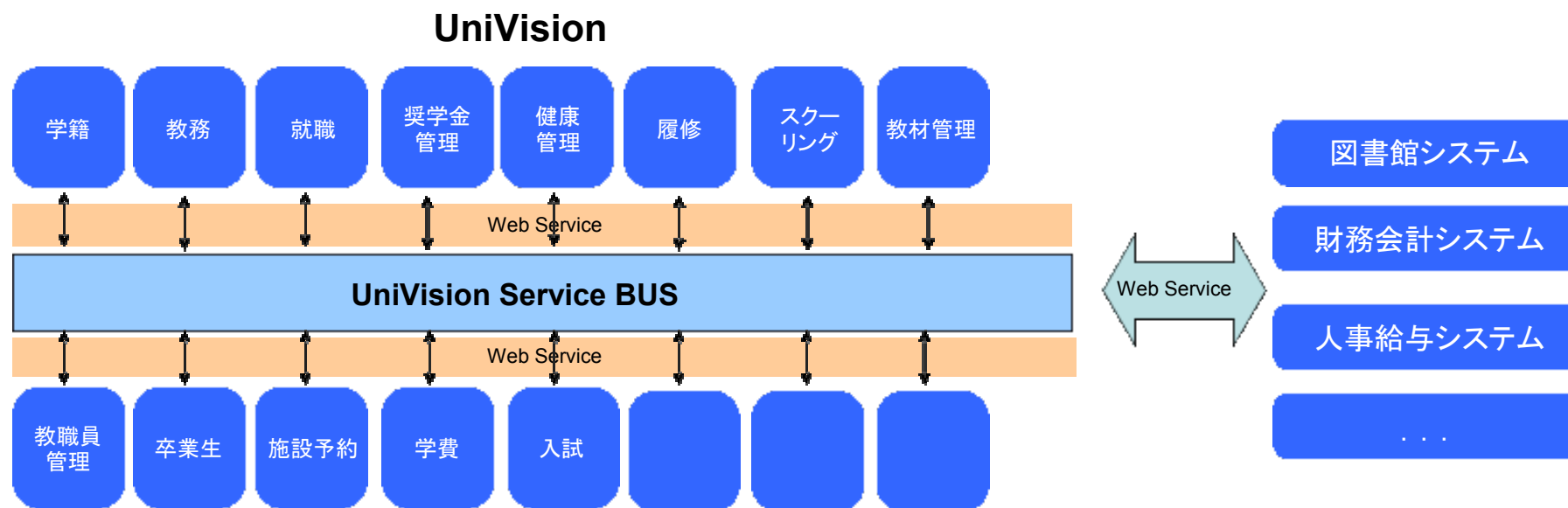


SOAによるシームレスな業務処理

【UniVision全体概念図】

# 総合文教ソリューションUniVisionの特徴

- SOAアーキテクチャパッケージ
  - 複数のサブシステムを組み合わせて構成できます
  - 個別要件にも柔軟に対応します
- オープンシステムにも対応
  - お客様の規模、ご予算に合わせて、ハードウェアやミドルウェアの構成が選べます



# オープンなフレームワークなので、、、

## 長所

- 同じようなコードが減る
- 個別の業務要件に対する固有の開発だけでよい

## 短所

- お約束が多い複雑な作業で、気が抜けない
- 工夫できる余地が少ない
- 少人数で大規模なシステムが作れてしまう
- 実行環境の構成は日々変化

複雑かつ**大変！** …… チケット駆動開発に向いている

# TiDDの実施方式

- 運用方式: 補完チケット方式
  - WBSと併用(と言っても更新作業は全てチケットあり)
- 権限ポリシー: オープン型
  - だれでもチケットが作成できる
- システムテスト以降
- システム全体
- メンバー全員
- trac(単独)
  - ... SRA共通開発環境(trac,subversion,mailmanの仮想環境)

# はじめの一步

- 宣言と実行

「バグだけではなく、ソースを触るときや、WBSにない作業をするときは、チケットを登録してください！」

- 環境の準備

- レポート(チケットの一覧)の作成

- bugのみ、taskのみ、その他、など
- 抽出条件(WHERE句)や表示項目(前に"\_"があると表示されない)は自由に変更できる

- 権限の追加

- tracの権限の設定は堅いので、チケットを修正できるようにmemberにTICKET\_ADMINの権限を与えた(リスクを考慮して設定してください)

# 結果

- チケットの数
  - システムテスト: 31 (不具合)
  - 本番環境構築: 42 (作業)
    - データの準備、環境準備、BUG関連で増えた作業、細かな仕様変更など、手順書にない1回だけの作業
- 作業漏れ減少！納期までに作業が完了！  
(知らないこと、気付かないことはできませんでした、、、orz)

それ以外にも、メンバーに変化が、、、

# 目が輝いた！

サブリーダ(クラス)なのに遠慮をしていたメンバーが、生き生きしだした

- 「チケットを切ってもいいですか？」  
⇒ 義務的な作業からの解放
- 「チケットを切っておかないと忘れてしまう！」  
⇒ すくに使いこなしていた
- 「ちゃんとクローズしてね」  
⇒ 他の人に指導をしていた！
- 「残っているチケットが多くてわかりにくいから整理しますね」  
⇒ 今後のことも考えている



# 注意すべき点

- チケットの登録忘れ
    - チケットがルーチンワークになっていないと忘れがち
  - チケットの実施忘れ
    - 他のチケットがないときにチェックを忘れる
- ⇒ 粒度の大きいチケットの後で発生
- ⇒ 粒度が小さいとリズムが生まれて、発生しにくい  
(リーダが管理すれば防げますが、自主性も大切)
- 上司への連絡 (^\_^;

# TiDDの効果

マネジメント方針がプロジェクト内で共有された

- 変化を抱擁できる
  - 当初想定しなかった作業を管理できた
  - 納期優先、優先度を考えて効率的に作業する
- コミュニケーション
  - 見える化され全員が状況を把握
  - 他の人のチケットも自由に登録
    - それまではサブシステム間の連絡はリーダーが担当
  - お互いに相談・協力するようになった

# まとめと感想

- システムテスト以降にTiDDを導入
- オープン権限ポリシーで実施した
- SRA共通開発環境のtracを利用した
- 計画されていなかった作業を管理できた
- コミュニケーションが向上し、プロジェクトが活性化
  
- 感想
  - TiDDは意外と簡単！(備忘録のつもりで)
  - マネージメント方針は運用ポリシーで伝わる
  - RedmineのSubtasking等を使ってもう少し上流でも実施してみたい

# 参考文献

- W・S・ハンフリー：TSPガイドブック：リーダー編，p.60，翔泳社，2007.
- 小川，阪井：Redmineによるタスクマネジメント実践技法，翔泳社，2010.

