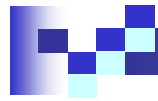


# プロマネがTDDと上手に付き合う方法 ～ 技術は管理できない～

オブジェクト倶楽部 2006 冬  
永和システムマネジメント  
岡島 幸男

いきなり宣伝です



プロジェクトを成功させる  
現場リーダーの  
**技術**  
岡島 幸男 / 著

**「才能」ではなく「技術」だ!!**

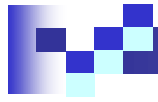
アクションかんぱん、キャラクターマップ、3Dフレームワーク、  
議事録ドリブンな会議、朝会、ペアCC、捨ててしまえ、  
美しさを優先する、ベターな複数案、KPT、タイムライン……

数々のソフトウェア開発プロジェクトで現場リーダーを努めた著者が、  
すぐに使えるテクニックを多数紹介

SoftBank Creative

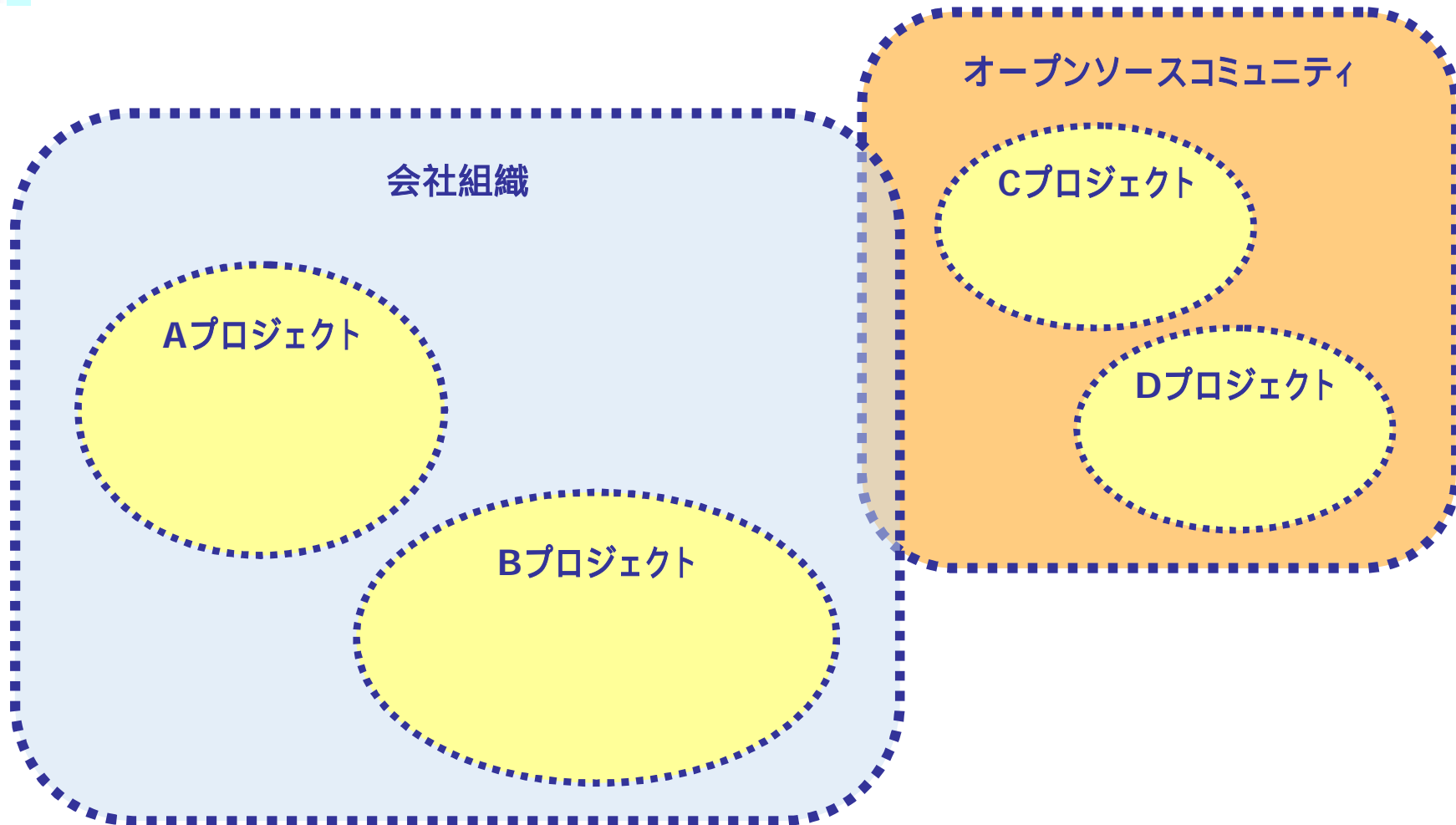
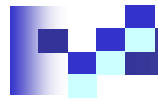
テーマは今思えば、「プロジェクトと人間」

問題意識



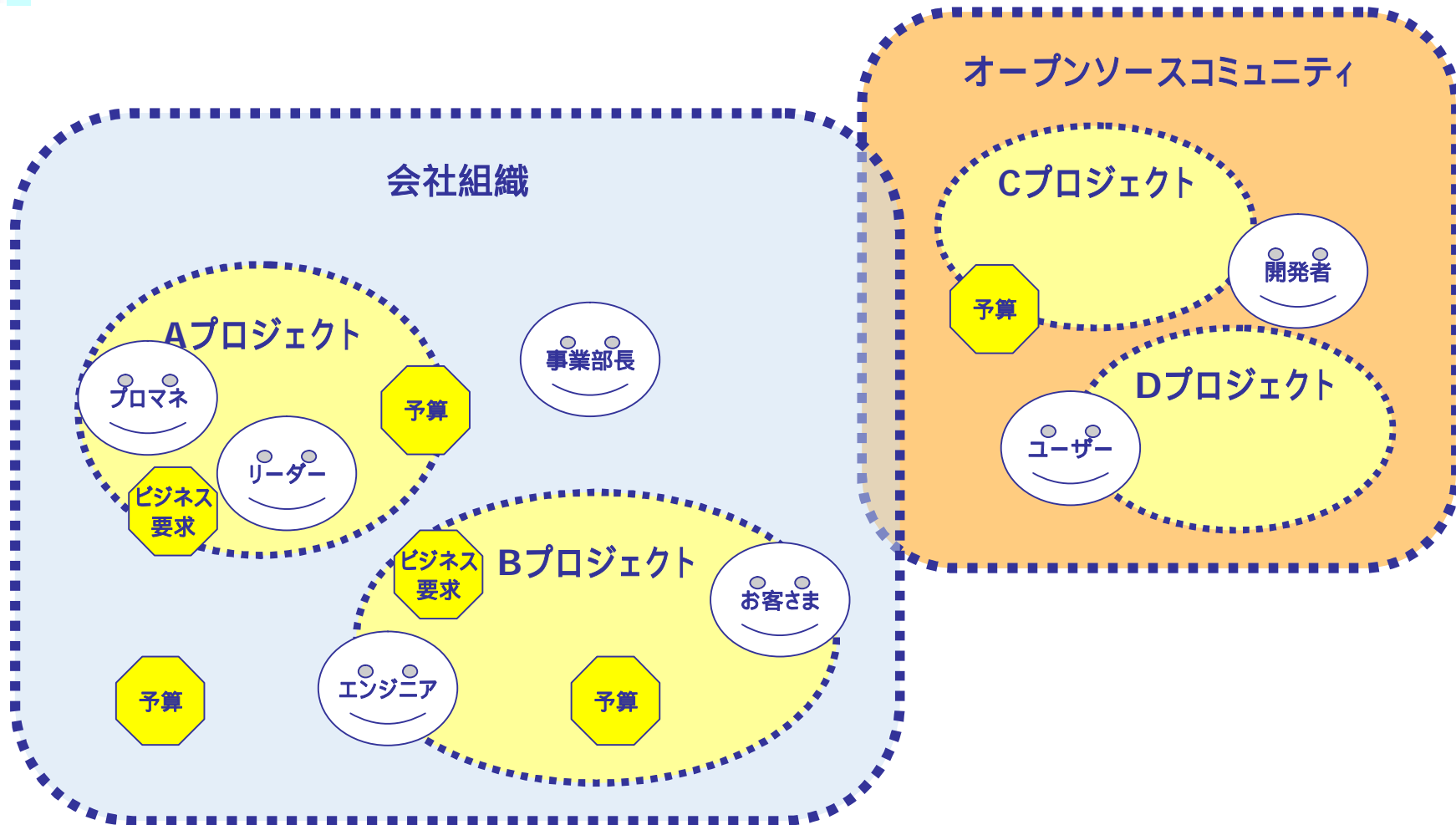
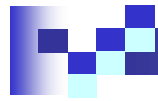
今日のテーマは「プロジェクトと技術」

# プロジェクトは文化である



プロジェクトには、全てそれ固有の「文化」が根付いている。

# 人・モノ・金は環境である



「プロジェクト文化」は、様々な環境要因によって特徴化される。

# 代表的なプロジェクト文化

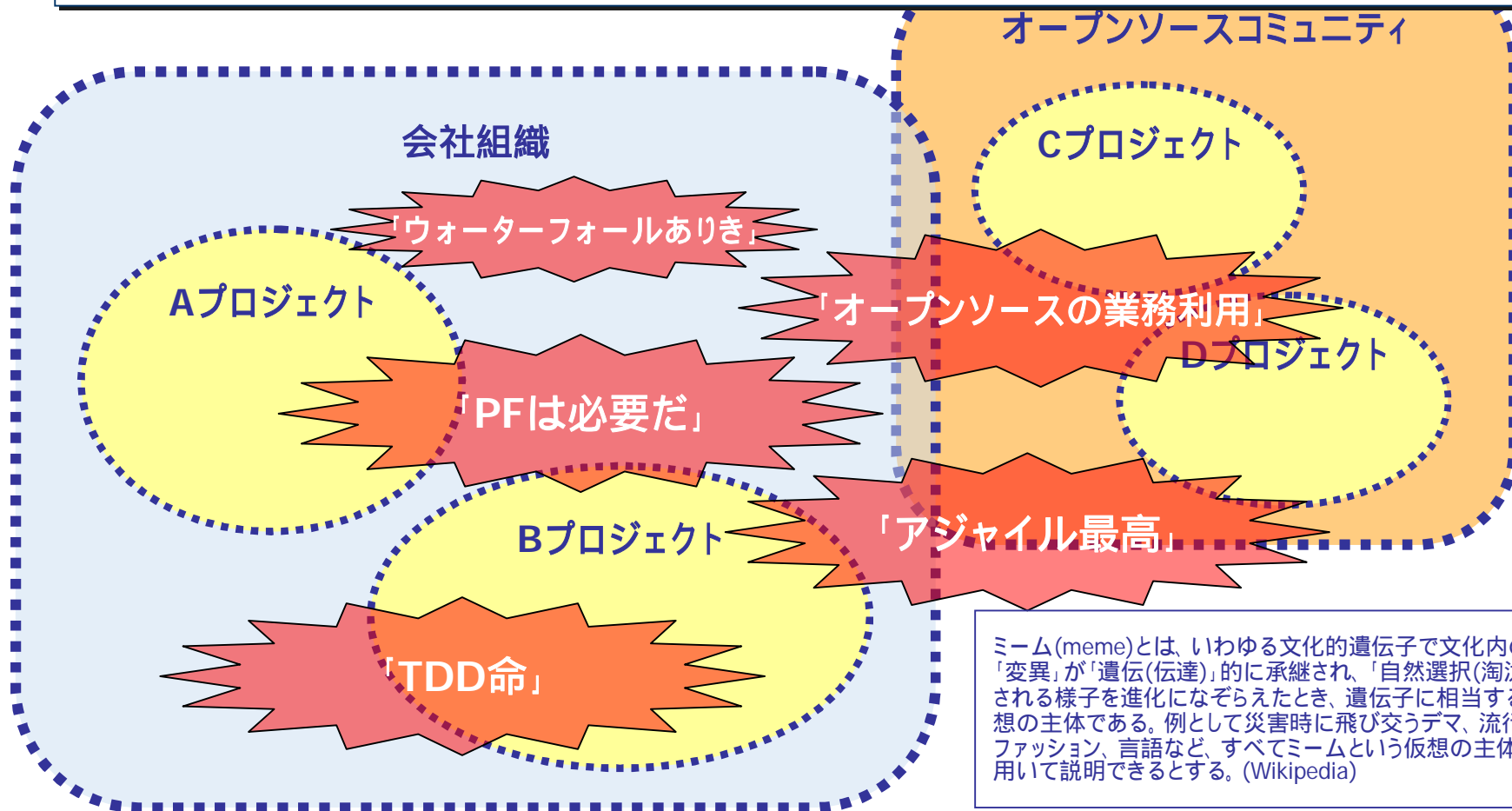
私の属する文化

	SI業界文化	IT業界文化
スポンサー(エンドユーザー)の目的	経費削減 (手作業の自動化)	ビジネスに対する投資 (新しい「作業」を生み出す)
見積可能性	(比較的)高い	低い
マネジメント	事前計画型	漸次改善型
スポンサーにもたらされる価値	削減された経費	生み出されたビジネスの価値
開発者(社)にもたらされる価値	受注金額 - かけた経費	ビジネス価値の一部
成果物に重視される要素	品質・確実さ	スピード・目新しさ
開発者(社)に求められるもの	安定したチーム	少数精鋭の個人
開発者(社)の関心ごと	「いかにコストをかけずに開発を すますか」(製造業的)	「いかにスポンサーのビジネスに貢 献するか」(サービス業的)
スポンサーの関心ごと	「いかにコストをかけずに開発を すますか」(製造業的)	「いかに優秀なエンジニアを確保す るか」(サービス業的)

同じ技術を使っているが、プロジェクトの性質はかなり異なる。

# 技術トレンドはミームである

ミームは「概念の遺伝子」で、人の心を媒介にして進化し、互いに融合したり競合したりする。



技術は人が広めるのではなく、「人を利用して広まる」。という逆転した視点が大事。

# ミームとの付き合い方

- ミームの進化や、生き残りの可否は、その文化の環境要因によって決まる
  - 「プロジェクトにはリーダーが必要」は、ほとんどの文化に適応する強いミームである
- たとえプロマネであっても、作用できない環境要因がある
  - お客さまは変えられない
  - 一度決まった予算はなかなか変えられない
- 文化に技術が根付くには、その技術を含むミームそれ自身が環境に適応し、競合するミームに勝たなくてはならない
  - 例えば「TDDは必要」というミームと、「TDDは時間の無駄である」というミームは競合する
  - どちらが勝ち残るかは、そのプロジェクト文化固有の環境により決まる
- ただし、プロマネは、その文化の中で特に強い影響力を持つある種の環境要因である
  - だから、ある特定のミームに肩入れしたり、進化を誘導することは可能
- プロマネにできることは、
  - 操作可能な環境要因をチューニングすること
    - 予算・工数を多少多めに見積もる
    - メンバーを教育する・入れ替える
  - ミームの持つ技術を正しく評価し、より広まりやすくすること
  - 自分が良いと信じる(感染された)ミームを見つけること

ミームは完全には管理できない。より有用で強いミームに進化するのをサポートせよ。



# 「TDDは必要」ミームとの付き合い方

- ミームの本質を理解し、強みを分析せよ
  - TDDはエンジニアにとっては「設計行為」であり、良い設計を促す
  - 繰り返し実施可能なテストは、品質を担保し、無駄なドキュメントを省く効果もある。
- ミームの感染源(あるいは感染先)である現場のエンジニアに相談せよ
  - 「テストは当たり前！」
  - 「テストがないと安心できない」
  - 「でも、切羽詰ると省略しちゃうこともある」
  - 「納品用にテストを書くのはむなしい」
- 自分のプロジェクトに取り込んだときの影響をよく考えよ
  - テストを書く分、見かけ上の工数は増えるのでは？(契約前の見積時には想定していない工数だよ)
  - ドキュメント(紙のテスト仕様書や設計書)って、納品対象では？
- 競合するミーム(「TDDは時間の無駄」)のことも意識し、弱らせる手をうて
  - テストを書くのにかかった時間を実績としてきっちり記録する
  - 工数が増える分は、「保守フェーズへの投資」とみなす戦略をとる
  - 成果・効果を計測し、評価する
  - 見積と実績を比較・分析し、次のフェーズ、次のプロジェクト用の基礎データとする

# 現時点での気づきと今後の課題

# 計画時点でTDDを前提とする必要がある

- プロジェクト計画時には、それを前提に余裕を持った計画を行う必要がある。
  - 具体的には、見積工数の最大1.2倍で考える。例えば、10人月(2名X5ヶ月)と見積もったなら、12人月分の人間を用意する。(期間は変わらないので、約2.5名X5人月)
  - この分の工数は、今後の保守コスト(保守担当の心理的な面も考えて)を考えればペイできるはず
  - また、推敲フェーズ時の予算は多めにみておく。(残業が増える可能性高し)

## とあるプロジェクトの実績データの一部

一日  時間計算

ユースケース	担当	基本設計書	PG&UnitTest	手動テスト	詳細設計書	UI自動テスト	合計(時)	合計(日)	見積(日)
		12	56	8	3	11	90	9.0	9
		6	8	5	2	2	23	2.3	5
		6	103	7	2	31	149	14.9	4
		6	43	7	2	3	61	6.1	7
		6	34	7	2	3	52	5.2	4
		10	45	6	2	6	69	6.9	9
		6	10	4	2	0	22	2.2	5
		52	299	44	15	56	466	調整前	
	ハマリ調整	-2	-103	-7	-2	-31	321	調整後	
	平均作業率	16%	61%	12%	4%	8%	100%		

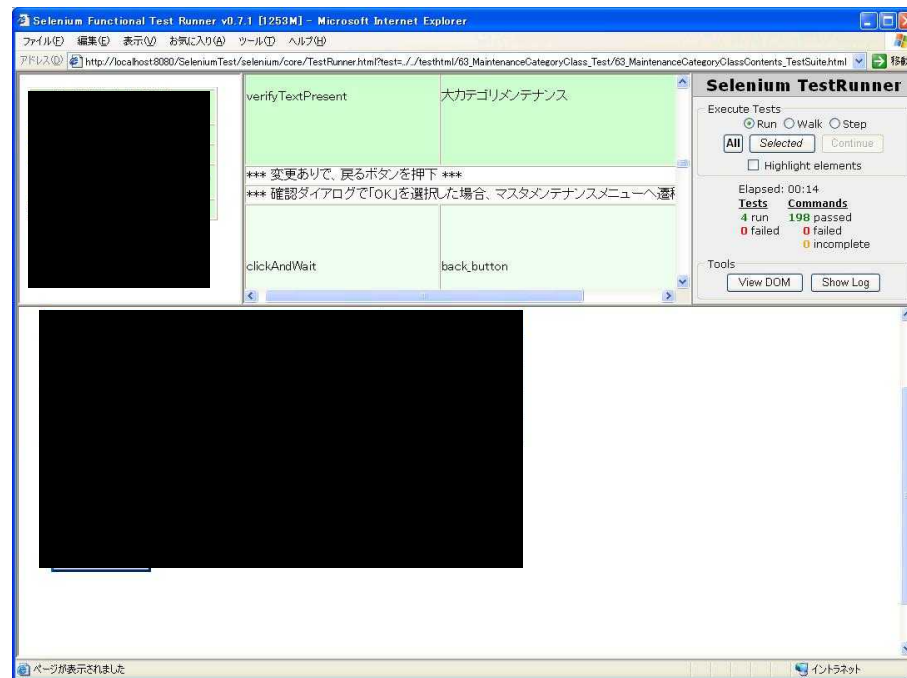
ハマリ時間は除外する

この時間が、見た目上  
増える工数

# テストの費用(工数)は、保守フェーズへの投資と考える

- テストの費用(工数)は、保守フェーズへの投資と考えればうまくいきそうだ。
  - カバレッジの高いテスト(UI自動テストも含む)は、保守担当にとっては、とてもありがたい。
  - 加えて、きちんとメンテナンスされたドキュメントもありがたい。テスト資産と設計ドキュメント資産のバランスを取る。
  - ドキュメントの記述粒度は、「保守担当者が読んだだけで理解できるかどうか」を判断基準とする。

## Selenium



ちなみに、「Seleniumを使う」ミームは、  
現在勢力を強め、進化している最中。

- 最初は、単純な入力チェックに利用
- 次第に、ロジックのチェックにも利用
- ユーザーエクステンションの活用が流行  
さらに、チーム内に感染している

# 記録された実績は、いろいろと役に立つ

- 実績を記録するのはとても役に立つ
  - コラボレーション感のある管理ツール、および「壁」を活用し、「管理されている」感を軽減
  - 実績データがあれば、見積に対する信頼感がとても強くなる
  - プロジェクトを通じて実績値を蓄積することで、次の類似プロジェクトでは、かなり精緻に見積もれるはず

ちなみに、「dotProjectを使う」チームは、  
現在勢力を弱めている

- 仕様変更、追加などによる頻繁にスケジュールの調整を行った
- 割り込みタスクが増え、実績の記録が難しくなった
  - プロマネが指示することによりなんとか生きながらえている

## dotProject

The screenshot shows the dotProject web application interface. The browser window title is 'dotProject - Mozilla Firefox'. The address bar shows the URL 'http://mos/dotproject/index.php?m=tasks&a=view&task\_id=157'. The page content includes a sidebar with navigation links like 'Sagee' and 'ツール'. The main content area displays task details for 'プロジェクト: 設計・実装・テスト' and a table of sub-tasks.

タスク名	タスクカラー	割り当てたユーザー名	開始日	持続	完了日	最終更新
100%			13/11/2006 09:00 am	8時間	13/11/2006 06:00 pm	-
80%			13/11/2006 10:00 am	57時間	21/11/2006 07:00 pm	-
100%			13/11/2006 02:00 pm	92時間	17/11/2006 08:00 pm	-
100%			14/11/2006 09:00 am	24時間	16/11/2006 06:00 pm	-
100%			14/11/2006 09:00 am	51時間	22/11/2006 06:00 pm	-
70%			20/11/2006 10:00 am	16時間	21/11/2006 07:00 pm	-

## (参考)「ペースメーカー」を壁に貼る

- 実績をベースに、工程(基本設計・実装・テスト他)ごとに必要な時間が割り出せる
  - これを開発メンバーに「ペースメーカー」として公開することで、過剰な前倒しを防げる

日数	概要設計	PG	移動単体	詳細設計	テレワーク
10日	1.5	6	1	0.5	1
7日	1.05	4.2	0.7	0.35	0.17
6日	0.9	3.6	0.6	0.3	0.6
5日	0.75	3	0.5	0.25	0.5
3日	0.5	2	0.25	0.15	0.25

## よい文化を継承する

- プロジェクトという文化は、開発メンバーが去った後でも残っている
- 成果物(TDDの優れたテスト資産)は、「遺伝子」として、プロジェクト文化に残る
- 保守メンバーには、このテスト資産を通じてそのプロジェクトの文化が継承される
- そして、このプロジェクト文化は、さらに上位の文化にミームを通じて継承される
  - プロジェクトから他のプロジェクト、あるいは会社組織へ