

Agile in a nutshell

ざっくりわかるアジャイル開発



Jonathan Rasmusson

Japanese Translation: Kakutani Shintaro (2010-10-21)
Original Slides: <http://agilewarrior.wordpress.com/presentations/>

<http://agilewarrior.wordpress.com/>



ADAPTIVE PLANS

ITERATIVE DEVELOPMENT

CHANGING REQUIREMENTS



THE WATERFALL HORROR FEATURE SHOW



気をつける

アジャイルなデリバリーが

- あなたのすぐそばに迫っている -

お話しする内容

- みんなアジャイルな計画づくりを知ってる
- アジャイル開発のために必要なこと
- アジャイル開発にまつわる迷信を解く
- 代表的なアジャイル開発手法の概要
- アジャイルになるための3ステップ



だいたいわかるアジャイルな計画づくり



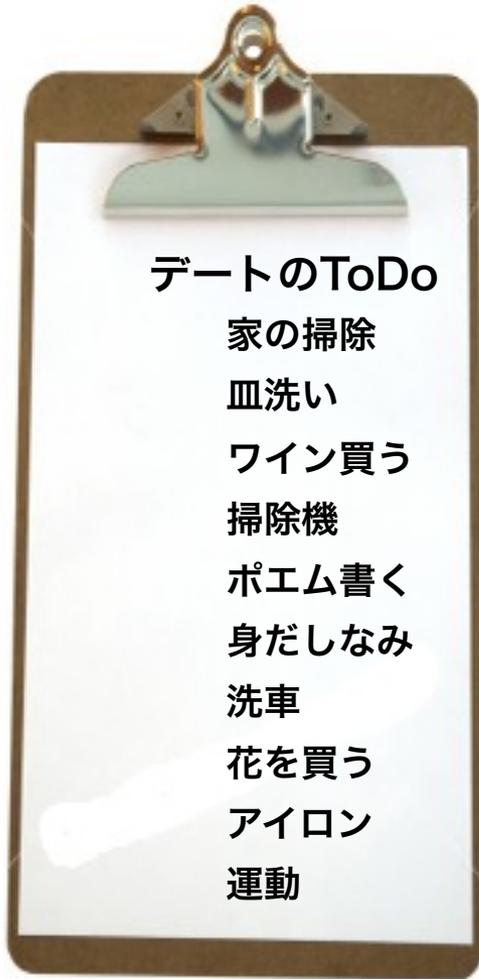
なすべきことは多く、時間は足りない



Credit: <http://www.flickr.com/photos/geneoh/>



リストをつくる

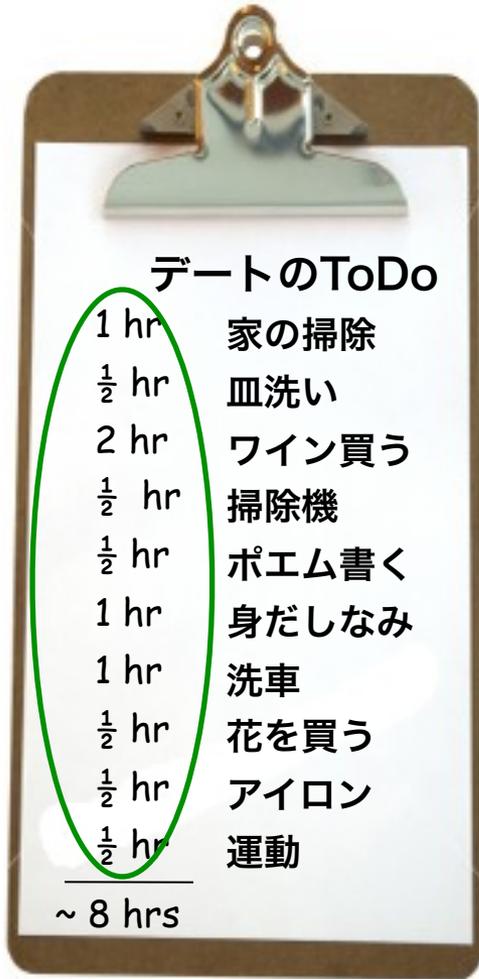


よし、デートの準備で
やらないといけないことは
これで全部かな？



気が済むまでやってよし

大きさを測る

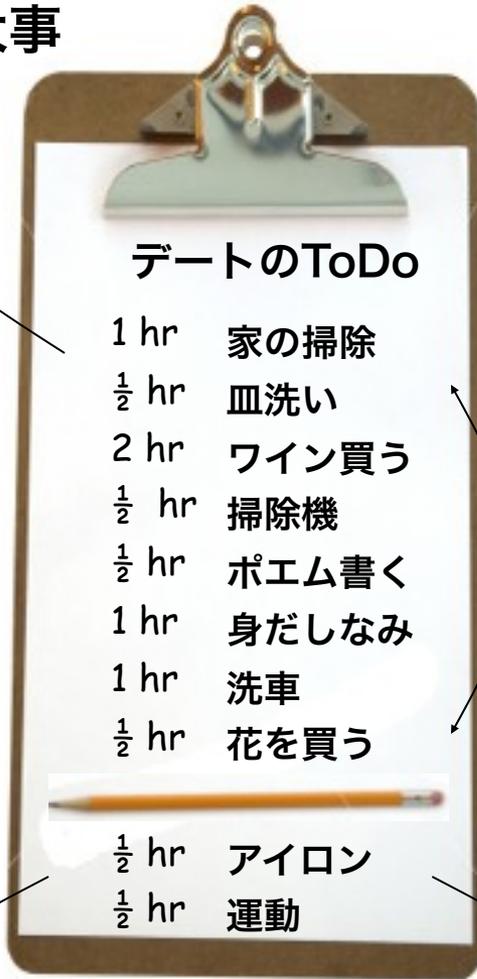


2時間以上かかるものは
なさそうだぞ!



優先順位をつける

いちばん
大事



ちえっ! ぜんぶやってたら
間に合わなさそうだぞ

テキスト



大事
じゃない

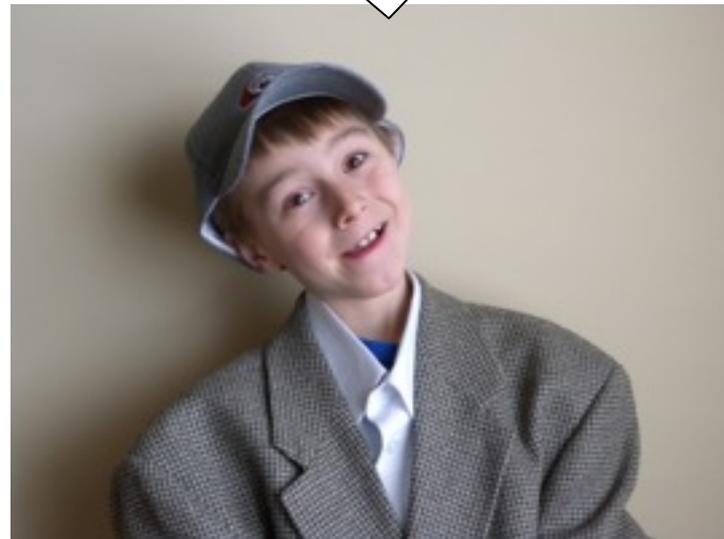
対象
範囲外



実行にうつす



ほんとは運動とかしたくないし、
アイロンも何がなんでもじゃないしね



だいたいみんながいつもやってるのと同じだったり

ちょっとした秘密

いいか



アジャイル開発
も同じことやっ
てんだぜ.....

こっちじゃなくて



ToDo リスト

タスク

当てずっぽう

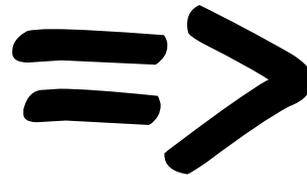
コジャレた名前でね



マスターストーリー
リスト

ユーザーストーリー

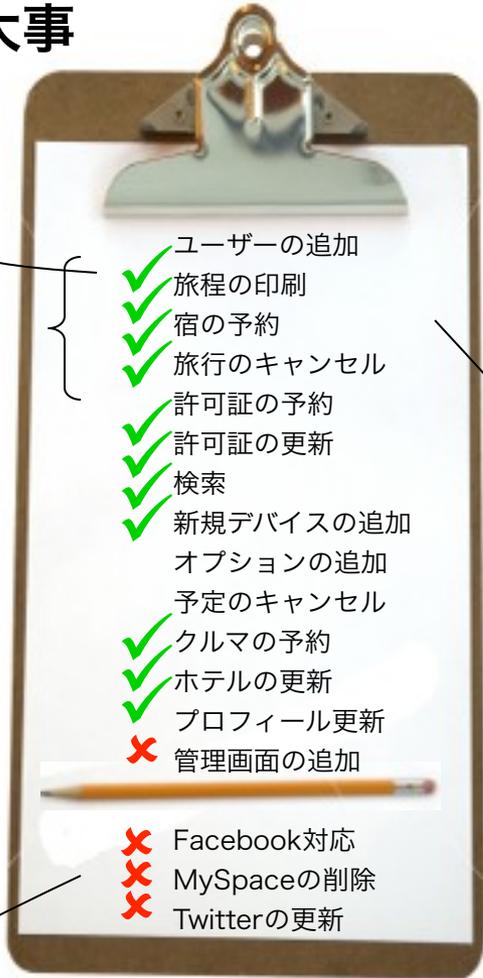
見積り



これをどうやって計画に？

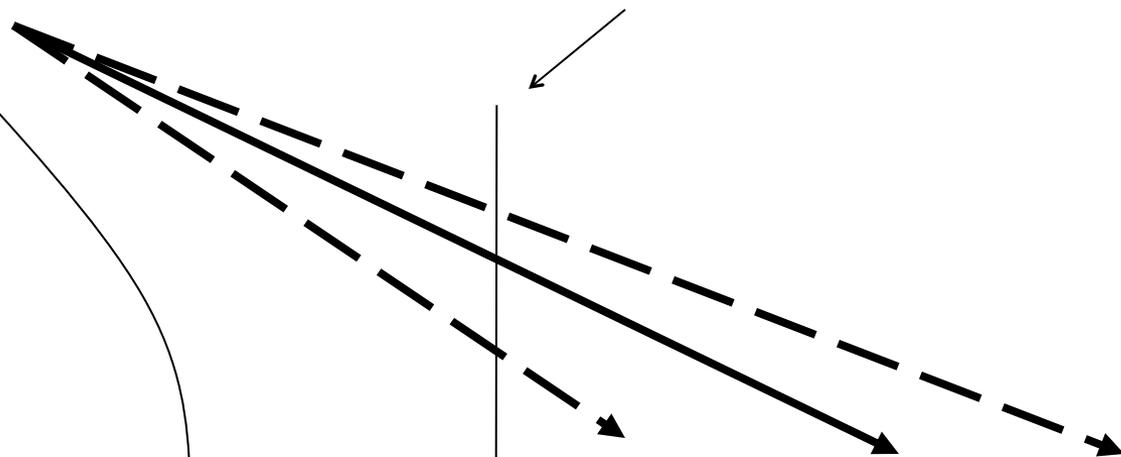
納期とか
予算とか
計画書とか

いちばん
大事



1週間

ここまでで 1/2 完了



1週間



1週間



1週間

予算切れ
時間切れ
やることなし

大事
じゃない



お話し内容

- みんなアジャイルな計画づくりを知ってる
- アジャイル開発のために必要なこと
- アジャイル開発にまつわる迷信を解く
- 代表的なアジャイル開発手法の特徴
- アジャイルになるための3ステップ



スコープを柔軟に



納期



予算



品質



スコープ



Flex here



同じ“doneの定義”を使う



落ち葉を集め終えたと言えるのはいつ？

計画？

設計？

報告書？

うちじゃそうじゃない！

動作するソフトウェアこそが うまくいってることを一番よく表す



どれだけ綺麗で
洗練されていたとしても…

プロジェクト計画書

テスト計画書

要求定義書

アーキテクチャダイアグラム

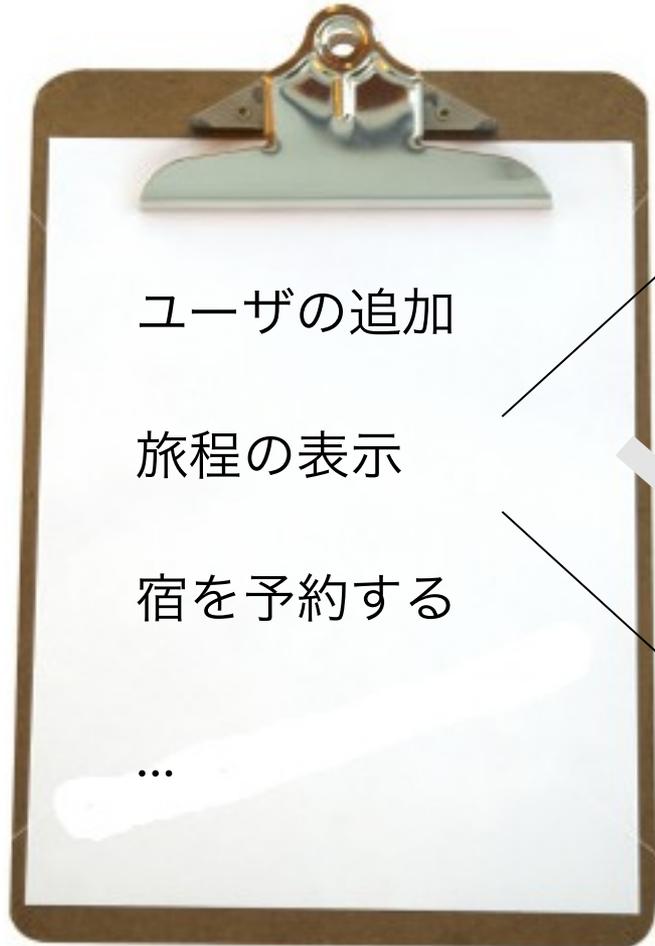
分析モデル

セキュリティ報告

開発計画書

…顧客には何ひとつ価値
を届けていない

つまり・・・



- ✓ 分析
- ✓ テスティング
- ✓ 設計
- ✓ コーディング
- ✓ その他もろもろ

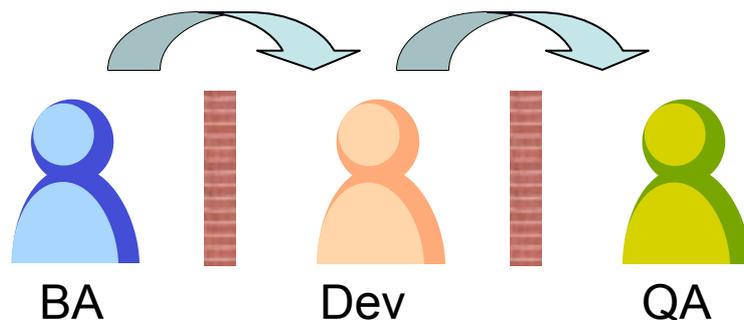
100% 完了

✓ 1週間

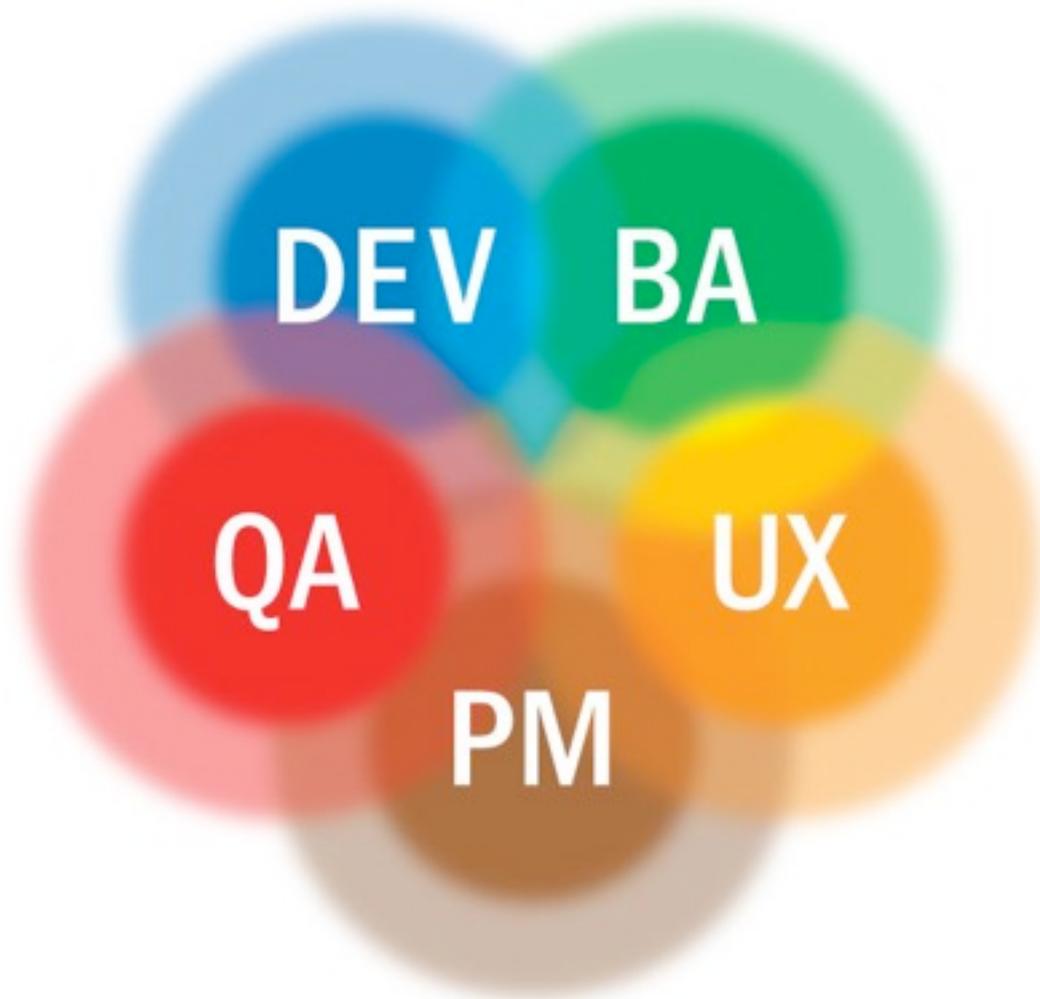
ひとつのチームとして働く



VS



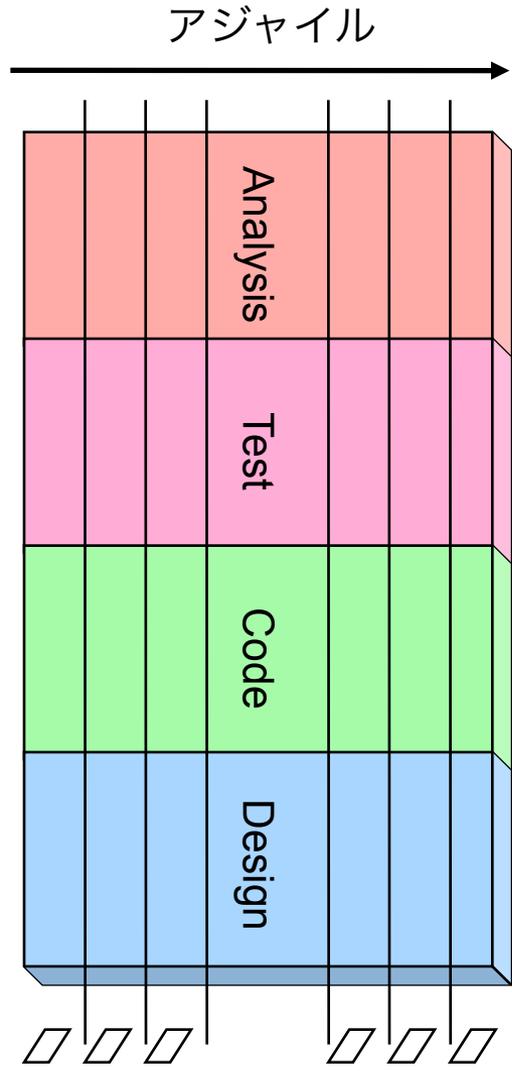
もっと役割のあいだに重なりを



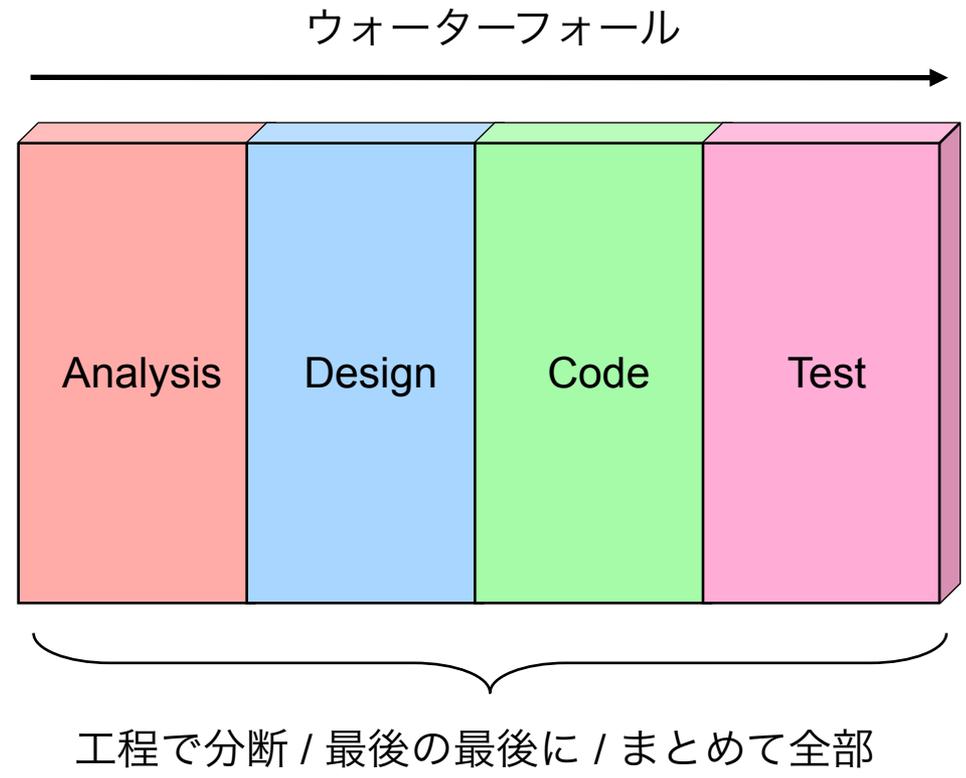
チームとして責任を果たす



分析、設計、テスト、コーディングはどれも 継続的な営み



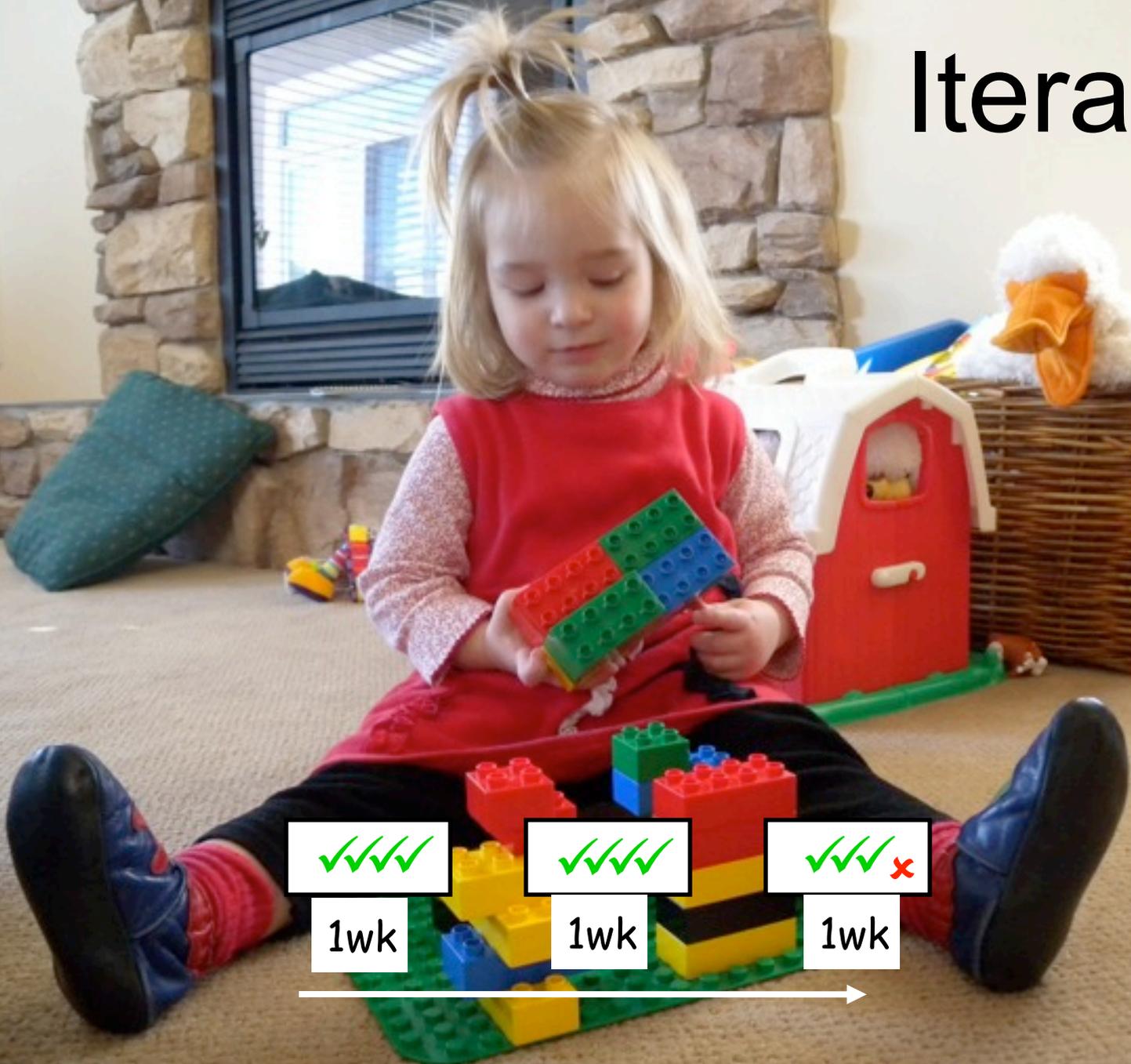
VS



聞いたことがあるかも



Iterative



✓✓✓✓

1wk

✓✓✓✓

1wk

✓✓✓✓x

1wk



Iterative



タイムボックスで届ける



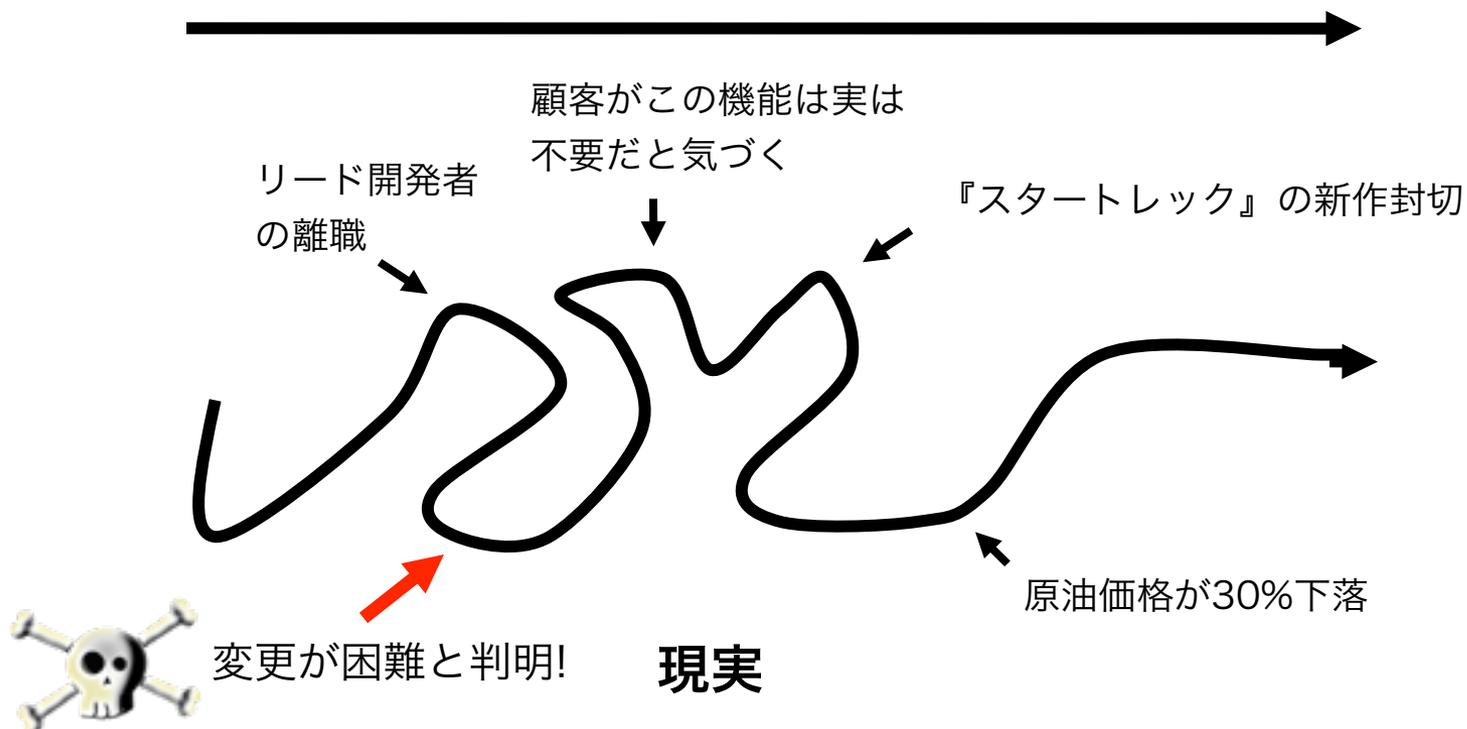
Credit Yogi: <http://www.flickr.com/photos/yogi/1147960/>

© Copyright 2009, Rasmusson Software Consulting

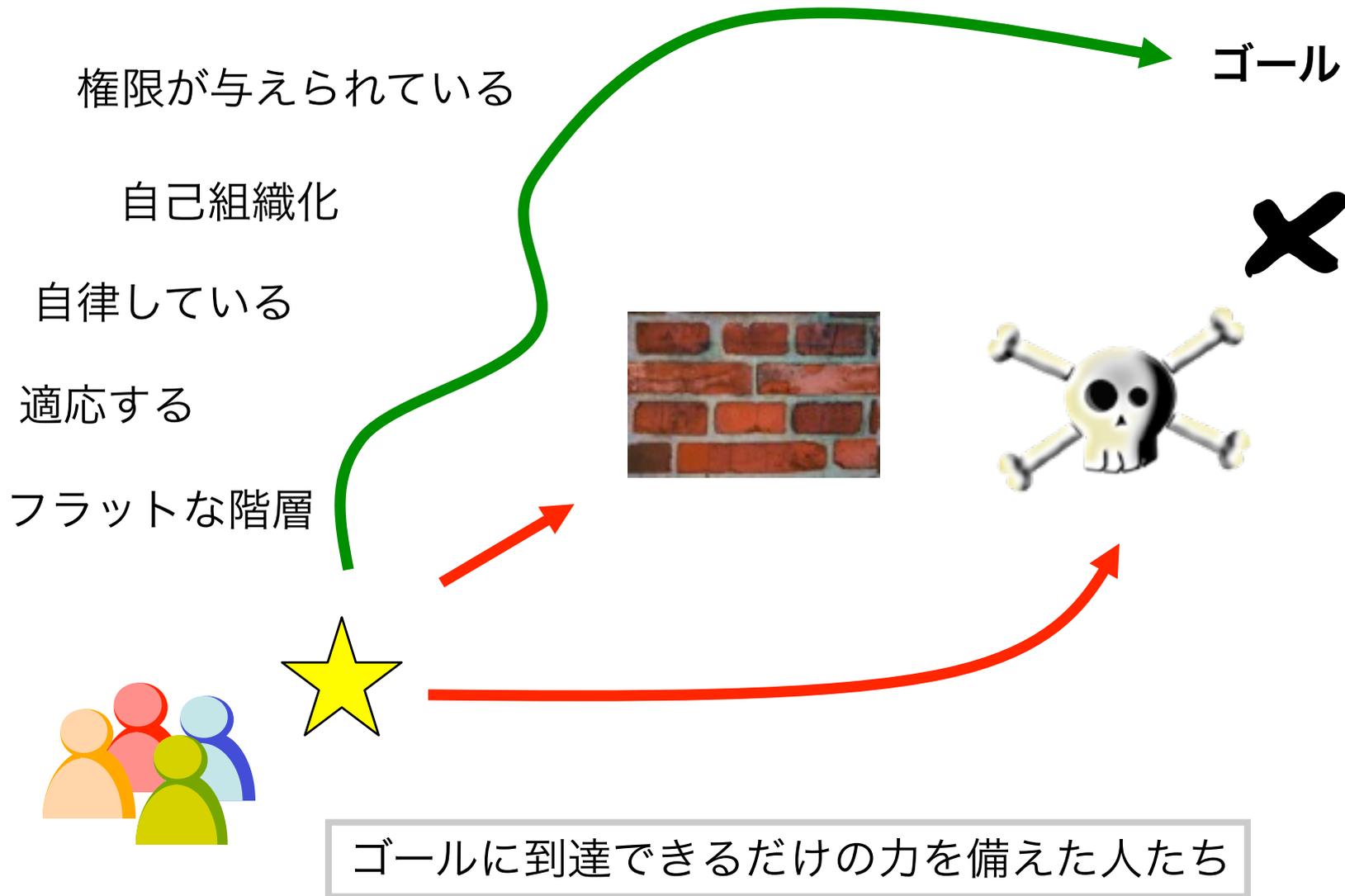


適応する計画づくり

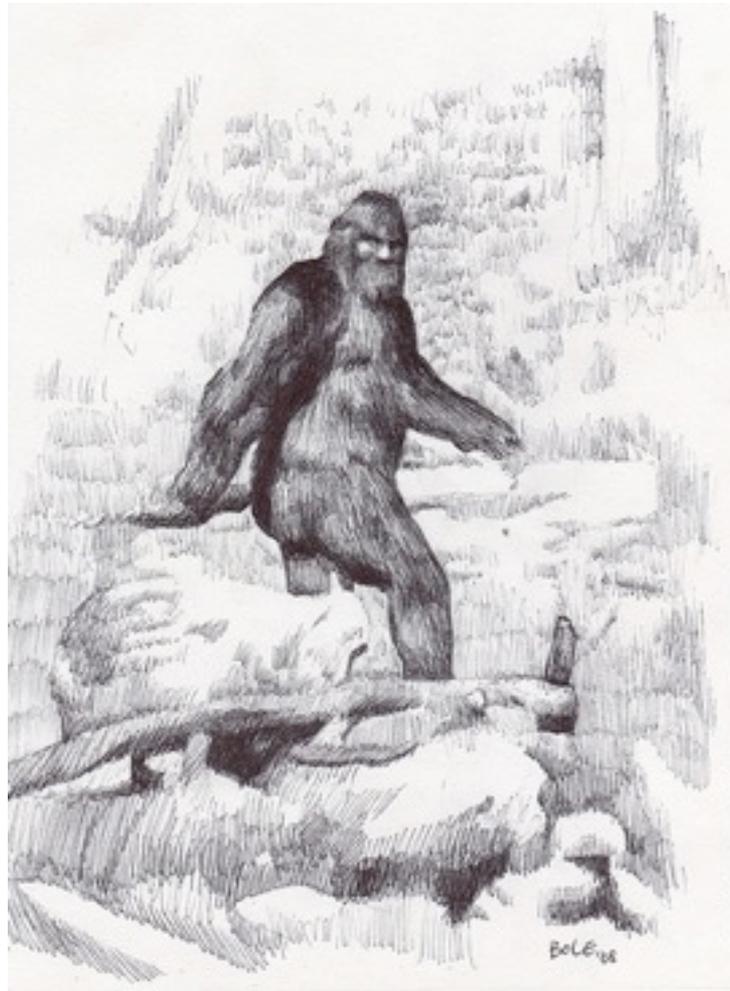
当初の計画



アジャイルチームの特徴



アジャイル開発にまつわる迷信



アジャイル開発とは何でないか



アジャイル開発は銀の弾丸

- ものすごい失敗をしでかします
- あらゆる問題はただそのまま残るだけ
 - アジャイル開発だと問題がすぐにわかってしまうというだけの話



アジャイルチームは ドキュメントを書かない

- 正確には、アジャイルチームは unnecessary なドキュメントは書かない
- アジャイルチームは文書よりも対面のコミュニケーションを好む
- ドキュメントを特別視しない
 - 優先順位をつけ、見積もり、納品する



アジャイル開発は計画しない

- アジャイルチームはどんどん計画する
 - 四半期ごと(リリース計画)
 - 隔週(イテレーション計画)
 - 毎日(スタンドアップミーティング)
- 使うツールは違う
 - バーンダウンチャート 対 ガントチャート, PERT図
- 目に見えやすい計画づくり
 - 問題があれば早い段階でステークホルダにもわかる

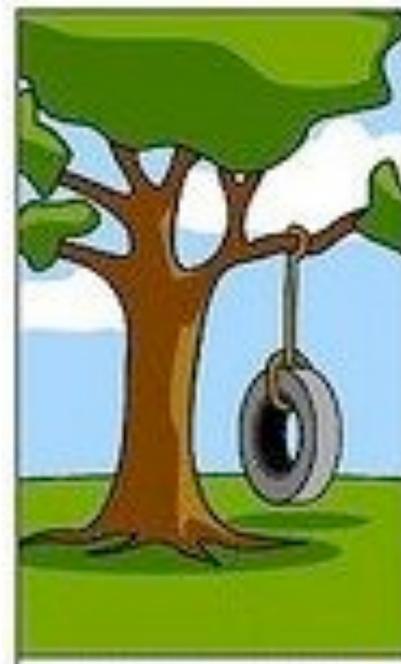


アジャイル開発はアーキテクチャを否定する



アジャイル開発はやりすぎるアーキテクチャに懐疑的なのだ

こんな設計に時間を費してはだめだ...



... 求められてるのがこうならね

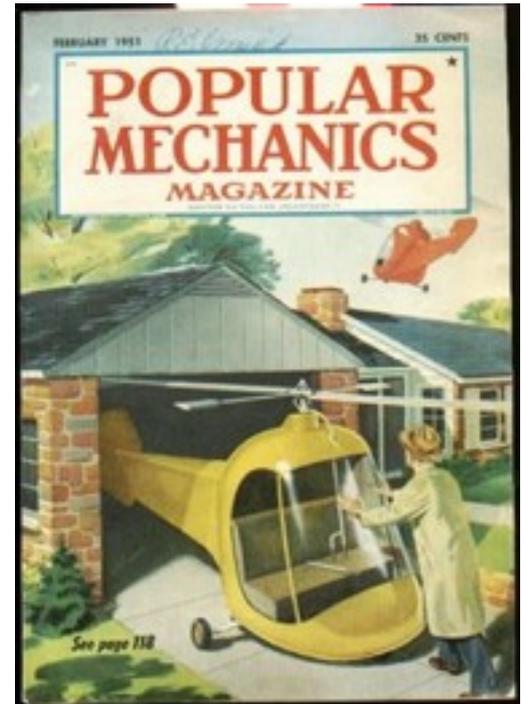
アジャイル開発はスケールしない

- 他の手法と同じぐらいはスケールするよ
 - あえて言うまでも無いかも
- スケールアップするかじゃなくて、
 - どうやったらスケールダウンできるかを考えるんだ



アジャイルチームは モデリングをしない

- アジャイルチームも
モデリングするよ
– 過信しないだけ
- モデルはなるべく早く
動作するソフトウェアに
するのが良い



アジャイル開発には規律がない

- 簡単な習慣しか採用しなかった人々のせいでアジャイル開発の評判が悪い
 - 難しい習慣はやってないじゃん
- アジャイル開発はとても規律が厳しい
 - テストを書かねばならない
 - 設計しなければならぬ
 - 書いたコードはどんどん結合せねばならない
 - どんどん動作するソフトウェアをリリースせねばならない
 - イテレーションごとに何かしら価値を届けねばならない
- チームがアジャイルになればなるほど、規律ただしくなっていく



アジャイル開発は予測がつかない

- アジャイル開発はソフトウェアが本来備えている複雑性と不確実性を認めている
- アジャイル開発では予測していたものに実測データをすぐに反映させていく
- 当初の計画と現実とのギャップにいつまでも悩まない
- 物事は変化するということを受け入れているので、よりよい情報が得られたら少しずつ計画を見直し続ける

どうしても予測可能であって欲しいというのなら、
選ぶ職業を間違えたと言わざるをえない

Credit: <http://www.flickr.com/photos/shadphotos/>



アジャイル開発には才能が必要

スキルを備えた人たち



プロジェクトの成功



価値をうみだす斬新なソフトウェアにはスキルを備えた人たちが必要

お話する内容

- みんなアジャイルな計画づくりを知ってる
- アジャイル開発のために必要なこと
- アジャイル開発にまつわる迷信を解く
- 代表的なアジャイル開発手法の特徴
- 開発をアジャイルにするための3ステップ



アジャイル開発には 色んなフレーバーが

Crystal

Scrum



Lean

DSDM

FDD

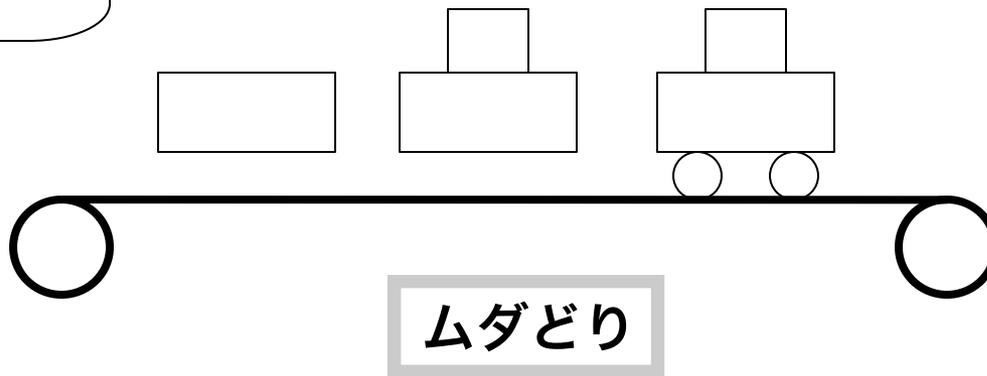
Extreme Programming (XP)



Lean

プリウス
ひとつください

トヨタ式の極めて無駄のない製造工程



Lean

+ pluses

- とても良いハイレベルなコンセプトとプラクティス群
- 全体最適によってシステムと組織を改善する

- minuses

- ITに限った話ではない
- 実践するのが難しい

Leanから得るものは数多い - 学ぶだけの価値がある



Scrum

- プロジェクトマネージャのための包装紙。テクノロジーやビジネスとは直交する
- ITプロジェクト以外でも使えるかも



Scrum

+ pluses

- わかりやすい
- はじめやすい
- 選びやすい
- マネージャに話を通じる
- 怖くない
- 今やいちばん有名

- minuses

- エンジニアリングをスルーしている
- 実践が難しい部分を飛ばしてやりやすいところだけになっている

はじめやすい - いちばん怖くないかも



Extreme Programming (XP)

- アジャイル開発に必要とされるエンジニアリングのプラクティスを広めた
- 強調しているのは:
 - テスティングの前倒し
 - 自動化
 - 創発的設計
 - 継続的インテグレーション



Extreme Programming

+ pluses

- しっかりとしたソフトウェアエンジニアリング
- 開発者たちに根強い人気
- 最初に広まったアジャイル手法

- minuses

- 開発者にフォーカスしすぎ
- 他と協調していきづらい
- 熱狂的な信奉者の印象が強い

本質的なエンジニアリングの習慣づけ



選択肢

XP

徹底する

Scrum

+

XP

engineering practices

Scrum

+

XP

+

Lean

engineering practices

spirit

* engineering practices

- unit testing, refactoring

- continuous integration, Test-Driven Design (TDD)



おすすめ

はじめはここ

XP

* engineering practices

言葉づかい
をえらぶ

Scrum

or

XP

学ぶ

Lean



やり方はひとつじゃない

Extreme Programming (XP)

Crystal

Scrum

Lean



DSDM

FDD

自分たちに合うことをやるう

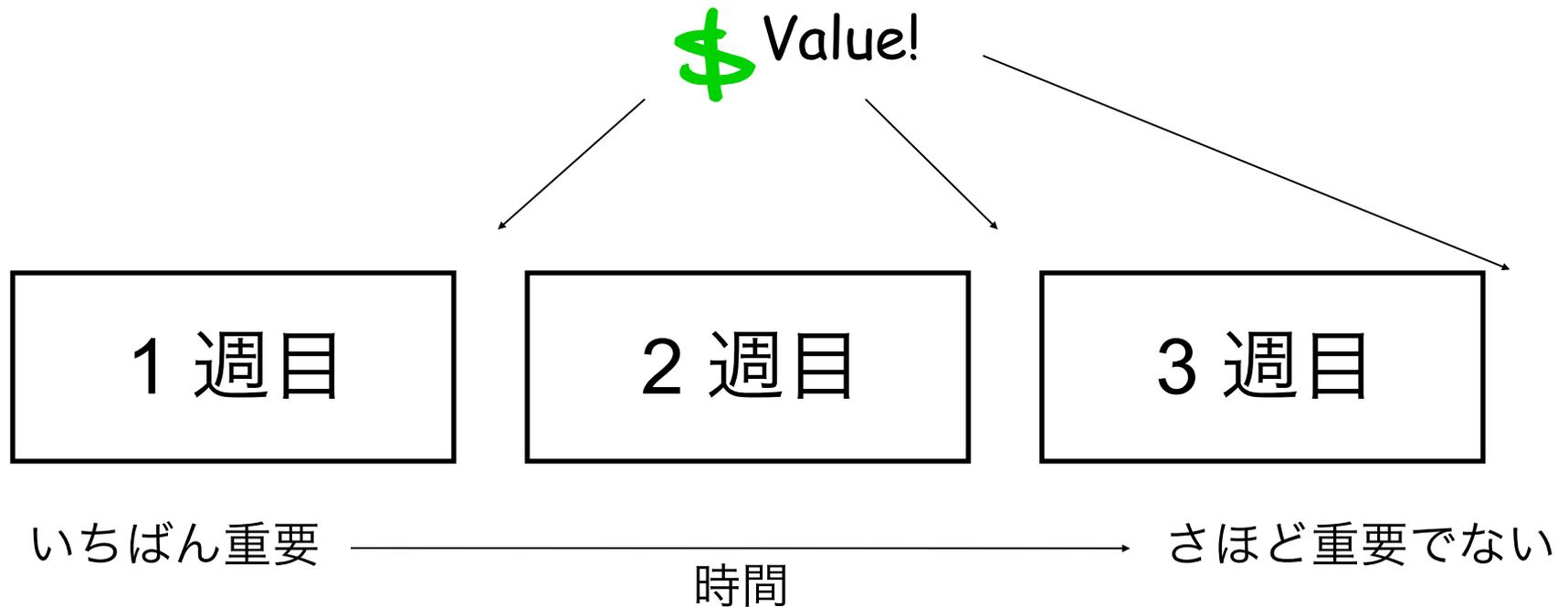
独自につくろう!



開発をアジャイルにする ための3ステップ



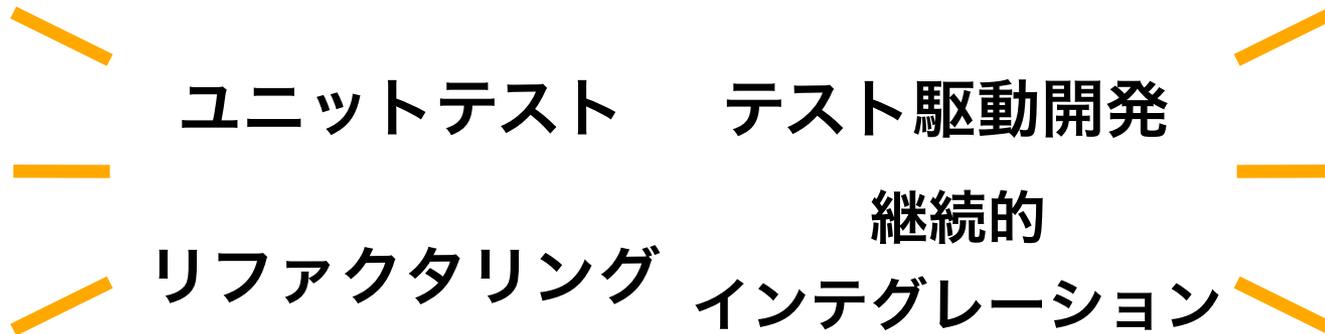
価値あるものを毎週とどける



4つのプラクティスから

今日から!

- アジャイル開発を支えるのは「技術的卓越」
- 必須のプラクティスはこれ：



- これらをきちんとやれたら
 - 他の事もどんどんやりやすくなっていく



3つのシンプルな真実を認める

1. プロジェクトの開始時点にすべての要求を集めることはできない
2. 集めた要求は必ずとっていいほど変わる
3. 要求はいつだって、与えられた時間と資金よりも多い



さいごに



さいごに

- 魔法なんてない - みんなアジャイルに考えてる
- 変化するものだと思おう
- やり方はひとつじゃない



**いつの日か誰もがアジャイルな
やり方になると思う？**



No

みんながみんな、食べすぎをやめて運動するようにはならないのと理由は同じ

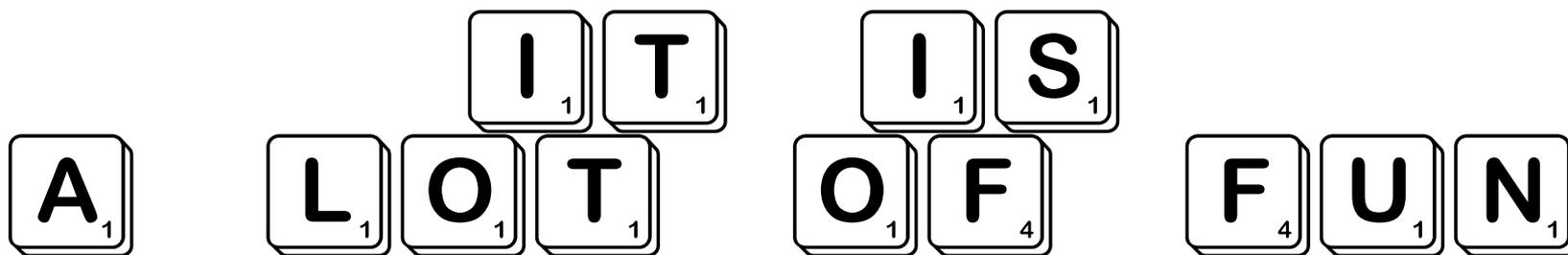


アジャイル開発はしんどい
アジャイル開発は大変な仕事
アジャイル開発には規律が求められる

誰しものがこんなことをしたいわけじゃない!



でもこれは仕事のやり方としては道理にかなってる



うまくいくよ — それもかなり



つづきは

<http://agilewarrior.wordpress.com/>



jonathan@agilewarrior.com

