

○ ● ● 表出化は誤解を期待してるよ
仮説

@kawagutiと@m_sekiによるSECIモデルの誤解による新たな
誤解の実例

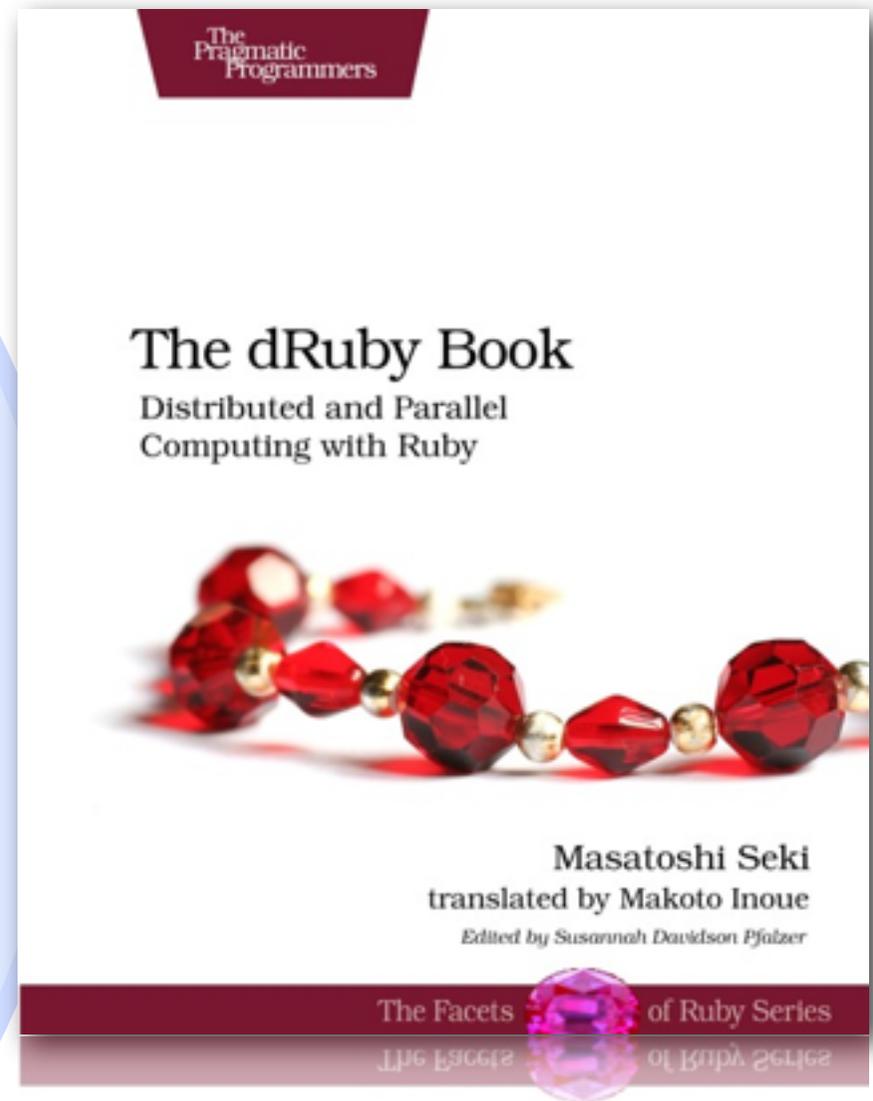


時間割

- 仮説 5分
- 議論 20分
- まとめ 5分

○ ● ● The dRuby Book

● amazonで予約now



○ ● ● 本来のアレ問題とかありそう

- 本来のTDDとか←とちぎテストの会議やった
- 本来のアジャイルとか!
- 守破離とか
 - そういえば守やってない
- みんなクヨクヨしてる気がする

○ ● ● 知識の伝播

- 知識が伝わるとはなにか
 - 100%の伝播ってなんだろう
 - そんなものあんのか？
 - フィードバックによる確認とエラー訂正

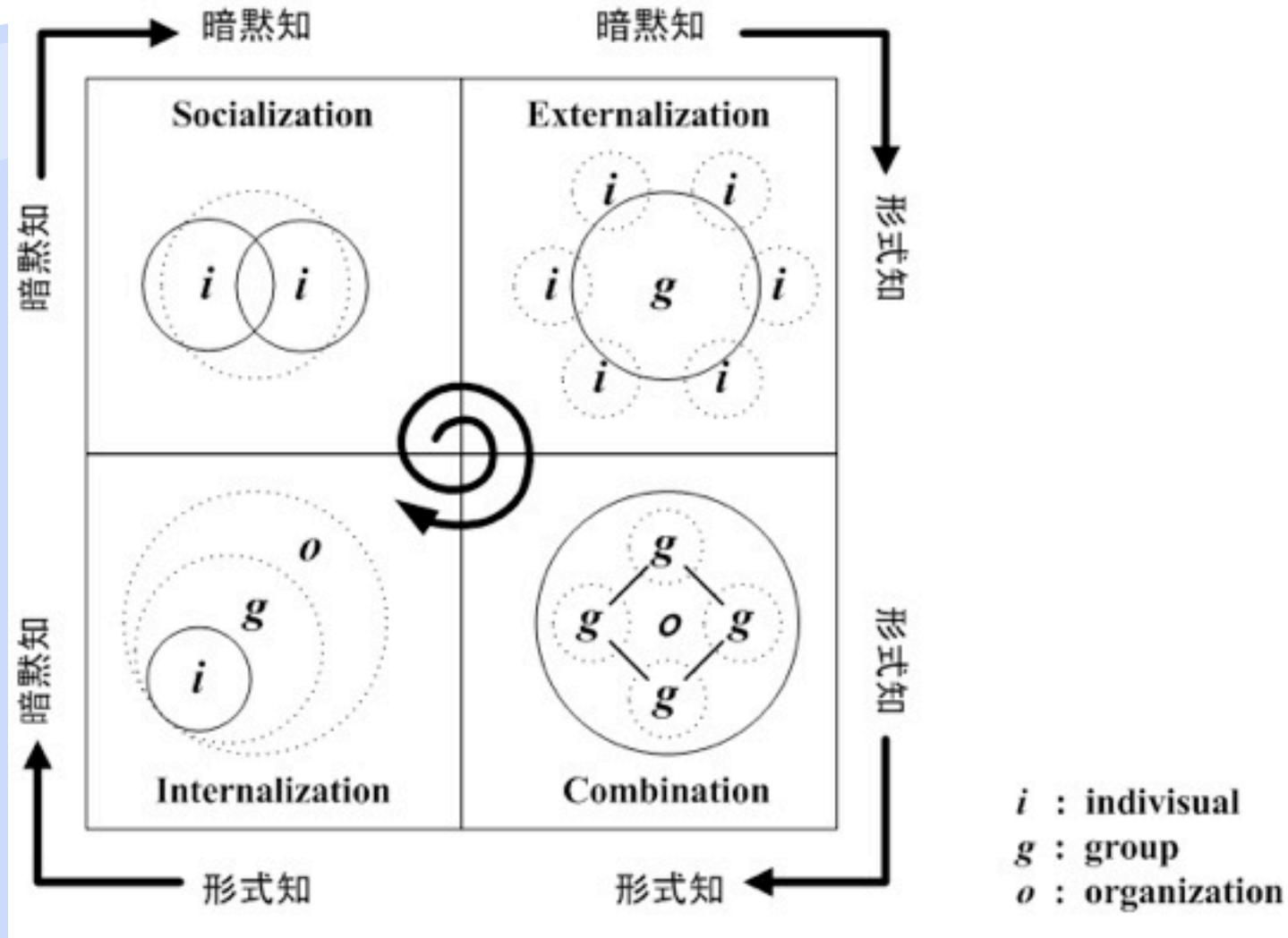
○ ● ● 誤解できるしあわせ

- 誤解したおかげでいろいろ生み出せた
- 完全なクローンが望めなくてもいいじゃん
- というよりむしろその伝播エラーこそが新しい知を生むんじゃないか

○ ● ● ということ

- 前向きで気が楽だ

SECIモデルを誤解するぞ



[Nonaka 98] Nonaka, I. and N. Konno (1998). "The Concept of 'ba': Building a Foundation for Knowledge Creation," California Management Review , 40-3, pp.40-54, 1998.

http://www.jaist.ac.jp/ks/labs/umemoto/ai_km.html

○ ● ● チームで動いているとき

- 知はみんなの血肉になっている
 - 暗黙知
 - 内面化
 - 共同化
- 情報は濃密に交換されててエラー訂正頻繁

○ ● ● チームの外へ言いふらす

- 短期間で全てを伝えられない
 - コンセプトにして外に出す
 - ある視点/モデルで切り取って誇張する
 - 表出化

○ ● ● 誰かが受信

- 受信側のもってるコンセプトと混ぜて受ける
 - 結合化
- エラー訂正はないし100%伝わることは期待してない

○ ● ● 表出化と結合化

- 花粉をまく
- 受粉する
- 積極的に伝播エラーを誘うことで似ているように違うものを生み出す
 - よりよくなるとも限らない
- 挿し芽みたいなクローンじゃない

○ ● ● 伝播エラーの実例

- TDDを聞いてたらテストに駆動された開発をするようになってたわー
- 「テストに駆動された開発」と聞いて手動テストの領域にもそれを持ち込んだらすごくうまくいったわー

○ ● ● 伝播エラーの実例

- ストーリーカード、かんばん
 - BTSのチケットと相似に見えちゃったので勘違いしてチケットでプロジェクトを微分してマネジメントしてたわー
 - その後よく似たコンセプトの「チケット駆動」を聞いて似てると思ったらちがってたわー

○ ● ● 伝播エラーの実例

● ERB

○ Railsさんそんな風に使わないで...

○ ● ● 伝播エラーの実例

- 2008年の関さんのデブサミ講演で触発されて、かわぐちがアジャイルのプラクティスを始めるきっかけに！

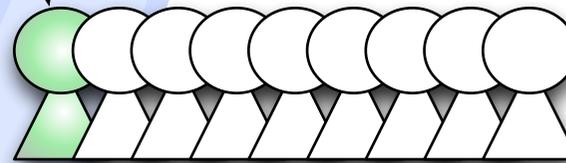
デブサミ2008 「7年目のアジャイル」

(Joel Spolsky の裏番組)



@m_seki

「XPの新しいところは"計画ゲーム"だけだと思う」
「Wikiでテストケースを管理していて、自動で動く」
「毎回走らせなくてもよくなったテストを外す」
「テストを管理する人は去年のデブサミで出会った」



@kawaguti

○ ● ● 伝播エラーの実例

- 受け取ったデータは
 - 「計画ゲーム」「Wikiでのテスト自動化管理」
- やったのは
 - デプロイ自動化、次に、スクラム (全然違)

デブサミ初参加。最初のセッションの衝撃

計画ゲームはみんなで計画するのか、すごいアイデア
XPはペアプロとか、チームがついてきてくれないと
できないけど、テスト自動化の部分は、自分の技術が
あれば、結構みんなが楽になる部分。
工夫すれば今の状況でもできるかも！
っていうか、7年前くらいからやってる人がいるのに
なにやってんだ、俺。

デプロイ自動化へ続く。

○ ● ● SECIモデルを誤解し誤解しよう

- だれかのコンセプトに刺激されて別の何かを始める例を示した
- ということかっこいいが原理主義的にはただの勘違いじゃん
- 勘違いしてるかどうかより自分たちのインスタンスがうまいかどうかを楽しもう

○ ● ● さて結合化の時間です

- 今日の話を誤解しよう