

# パッケージ製品の継続的開発における PDCAサイクル定着への取り組み

住友電気情報システム株式会社  
松田 行正  
2014-10-16

# 目次

1. 本発表の主旨 ～気が付けばPDCA～
2. 背景：製品Xの既知不具合の推移
3. まずは現状把握
4. 第一ステップ：仮説と検証
5. 第二ステップ：仮説と検証を継続
6. マネージャの役割：情報共有と指示
7. 第三ステップ：定着と次なる改善
8. まとめ

# 1. 本発表の主旨 ～気が付けばPDCA～

- 製品Xの保守開発(不具合修正)において、プロセスの改善が持続するようになったPDCAサイクル定着までの歩みを紹介します。
- メンバに対しては、気が付けば「これをPDCAっていうのかな」と思うようなソフトな導入の仕方が一番スムーズです。

## 2. 背景：製品Xの既知不具合の推移(1/4)

- プロジェクト着任時(2011年7月)  
製品Xの品質改善が重点課題でした。



## 2. 背景：製品Xの既知不具合の推移(2/4)

- その後3ヶ月間、さらに不具合総数は成長を続け....



ヤバイ、「頑張って」で  
なんとかなる状況ではない



## 2. 背景：製品Xの既知不具合の推移(3/4)

- 2011年10月から、プロセス改善に取り組みました



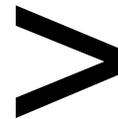
## 2. 背景：製品Xの既知不具合の推移(4/4)

増加要因

- (1) 顧客数(導入先数)増加  
使い方が多様化し、  
潜在不具合が顕在化する
- (2) 修正が新たな不具合(デグレ)  
を埋め込む

減少要因

- (1) 不具合修正して  
バージョンアップ



増加要因の(1)は製品が継続的に売れると発生し続ける  
増加要因の(2)は発生個数は(1)に比べれば微小。

### 3. まずは現状把握(1/3)

- とりあえず、現状を知ることから始めよう
- 2011年10月31日 第一回**振り返り会**  
メンバ: PM(前任 + 松田) + プログラムチーム  
→ 10個ほど「困っていること」が出てきた。

#### ★特に★

- (1) レビュー会の開始時間が遅れる…時間切れで修正が進まない  
レビューが完了しないとソースコミットできない。  
→ 時間を厳守するようにします。
- (2) レビュー会の議事録が発言録 … 何が決まったのか不明  
次回のレビュー会でもどうどう巡り  
→ 発言録からToDoリストへ整理・整頓(次ページへ)

### 3. まずは現状把握(2/3)

#### 取組の主な内容：レビュー議事録の改善

##### Before：発言録

- ・対応漏れが散見される
- ・進捗状況が把握しにくい

2011-07-01

1. ○○した方が良い(A氏)
2. ××だと思う(B氏)
- ...

誰が、何をしたらいいの？

2011-07-10

1. ▲▲は考慮していますか(C氏)
- ...



##### After：ToDoリスト化

- ・対応漏れを解消

No	指摘箇所	指摘事項	指摘者	発生日	対応担当者	対応内容	対応完了日	対応確認者
1	設計	RMS 6137 と同じ互換性パラメータ (CMPT.CreateCacheToGetStatus) を用いて Ver6.0.3.0 との互換性を確保します。	松田	2012-05-23	永野	不要	2012/7/6	松田
2	設計	rkasUtil の chkRevAuthergroup は、今回の修正箇所では getDocData() をそのまま呼び出して、結果をcache に格納する。	松田	2012-05-23	永野	対応確認しました(松田)	2012/7/6	松田
3	コード	p_folderidmap は p_folderidMap と、意味の区切り目で大文字にしてください。	松田	2012-05-23	永野	対応確認しました(松田)	2012/7/6	松田

### 3. まずは現状把握(3/3)

- 2011年12月07日 第二回 **振り返り会**  
メンバ: PM(前任+松田) + プログラムチーム

#### 1. 前回の取組みの振り返り

- (1) 時間を守る → まあまあできた
- (2) 議事録を発言録からToDoリストに → 効果あり  
→ 効果があった発言があったので、継続する宣言!!

#### 2. 数字で表現してみる

直前のリリースは約〇〇人日で不具合修正××件を設計  
→ 設計は 1.5人日/件と換算できるのでは?

注意、「反省会」と言うと、お通夜みたいです。  
「振り返り会」と言ってください。前向きな気分になります。

## 4. 第一ステップ: 仮説と検証(1/2)

- 毎回の削減計画を立てよう  
→ 人に納得してもらうには、数字が一番分かりやすい

(1) 仮説: 不具合修正 設計 = ▲▲人日/件

→ Aさんは20日あるから13個、  
Bさんは18日あるから12個ね。

(2) さりげなく刷り込み開始

「2ヶ月に1個、小さな改善を身につけていったら、  
1年たったら6個はステップアップしてるよね..」

(毎回何か取り組みテーマを追加する)

- ・前は「時間を守る」
- ・今回は「修正箇所仕様書の有無チェック」を追加

## 4. 第一ステップ: 仮説と検証(2/2)

### ■ 2012年02月07日 第三回 **振り返り会**

- ・仮説を検証してみる

	平均 設計時間	平均 テスト時間
仮説	12.0Hour	8.0Hour
実績	13.0Hour	6.5Hour

- (1) 検証: 設計1.5Day/日、テスト 1Day/日という仮説は確からしい  
→ 年間で〇〇人日確保すれば、××個不具合削減できると計画が立てられそう。

仮説(やってみよう) → 検証(確からしい) → 既成事実化 → 計画立案

## 5. 第二ステップ: 仮説と検証を継続(1/3)

- 振り返り会の都度、増やしてきた取り組み  
2か月に1つ、取り組みを増やしていく。

No	取り組み内容	状態
1	会議開催時間、納品時刻を守る。	継続
2	議事録はToDoリストにして完了まで管理する	継続
3	不具合修正 1個あたりの平均設計時間から計画を割り振る	継続
4	チーム内でCI( Code Inspection )をしてからレビューを受ける → 規模が大きいと判断したものに事前CIを限定する	変更
5	バイナリ以外のファイルがリリース漏れしない仕組みを運用する	継続
6	ソースコミット時に設計書も合わせておく	継続
7	不具合修正時は、不具合の画面ダンプを掲載して before/afterを明示する	継続
8	リリース日 お昼12:00にリリースを完了させる	継続

効果が出なかったものは、潔く変更、撤退する

## 5. 第二ステップ: 仮説と検証を継続(2/3)

- 「気が付けばPDCAになっていた」を目指す

P: 計画 = 仮説

D: 実行 = やってみよう

C: 検証 = 効果のほどは?

A: 処置 = 継続、撤退、変更

(注) 本PDCAは松田の我流アレンジが入っています

やってはいけないこと。

「PDCAを回してプロセス改善するぞ」と宣言すること

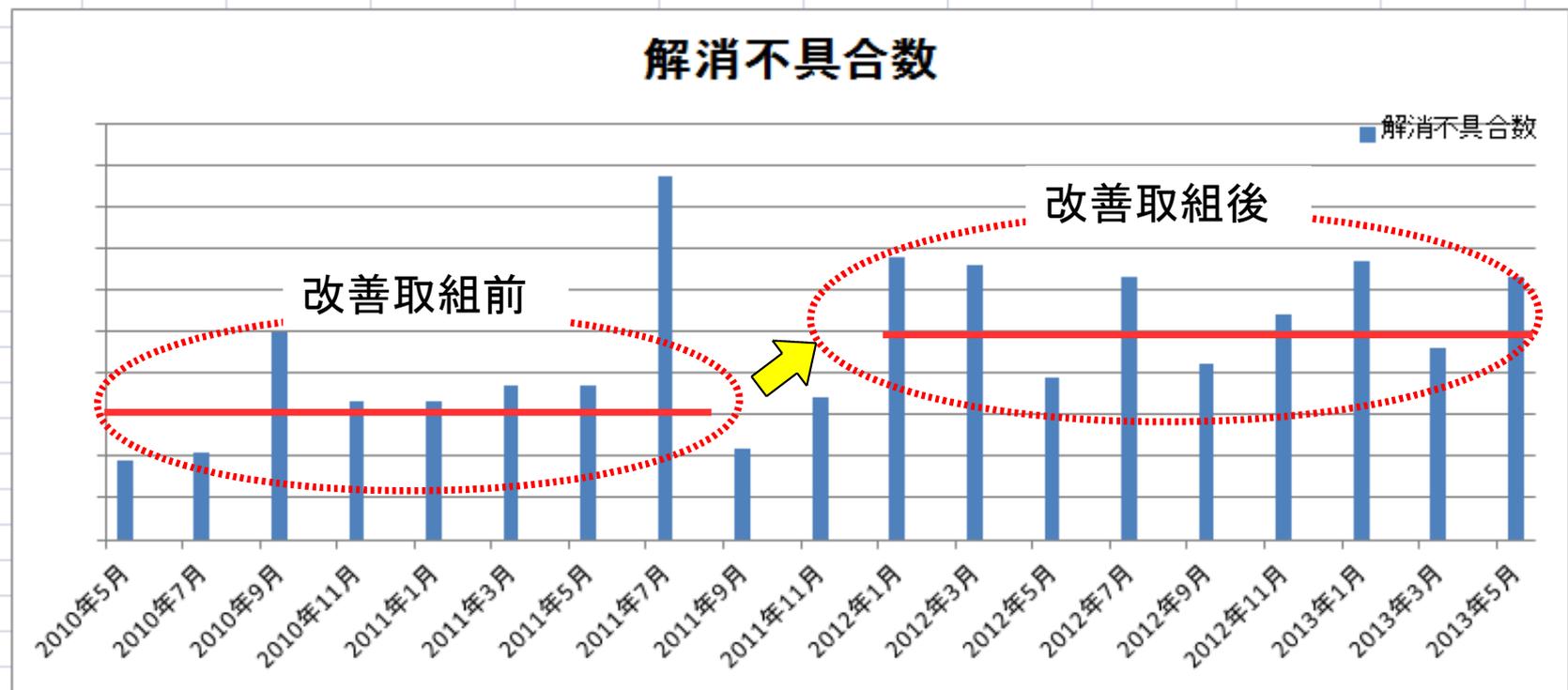
→ 変化を嫌う人は難癖付けて抵抗します。

ではどうするか

「小さなことを積み重ね、数字で納得させ、気が付いたら あたりまえの状態に定着」させてしまう

## 5. 第二ステップ: 仮説と検証を継続(3/3)

- 成果: 2ヶ月単位での不具合解消数の推移  
(注) 開発メンバーの人数構成は同じです。



## 6. マネージャの役割: 情報共有と指示(1/4)

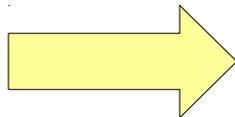
- 取り組みを掲げるだけでは改善しない。
- マネージャは期待した効果を上げられるように裏方からサポートする必要がある
  - (1) メンバが現状を理解できるように見える化を行う
  - (2) メンバに適切な指示を出す

# 6. マネージャの役割: 情報共有と指示(2/4)

- 修正中不具合の状態をマネージャが把握し、適切な指示を出す → **そのために** タスクを見える化



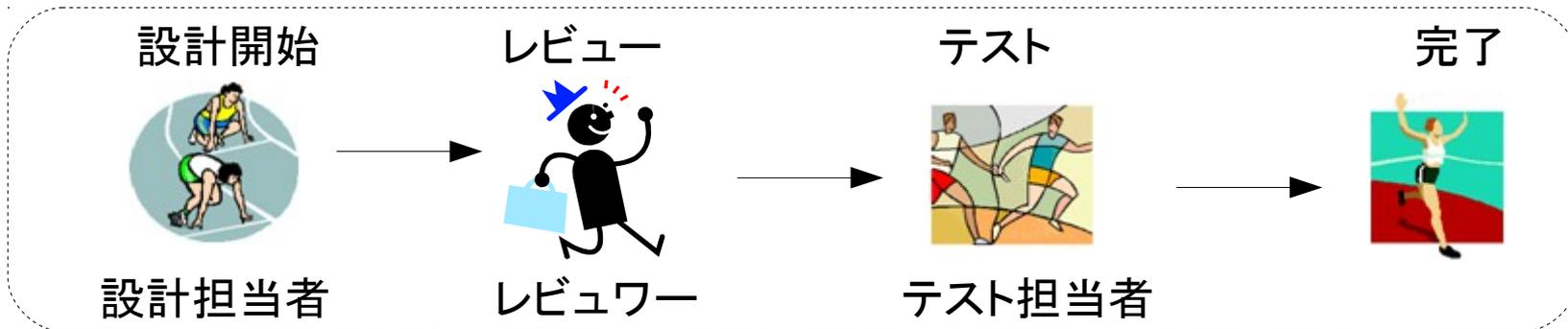
チェックに追われる日々  
のマネージャー



#	チケット	ステータス	題名	担当者	更新日	設計担当者	△	テスト担当	分類
1362		終了	6938 変更承認キーワード検索結果詳細で入力エラーを起こすと画面が崩れる		2013/06/20 11:05				不具合
1361		RMS_コミットチェック	6942 [Possible]ユーザー一括登録で削除ユーザーを登録した場合、表言語が削除ユーザーのままとなっている		2013/07/16 13:45	TCS 様	WITS	下原	不具合
1587		RMS_コミットチェック	7011 文書保存時にエラーになるとブルダウン項目の「選択して下さい」が壊れる。		2013/07/10 18:56	TCS 様	WITS	山田	不具合
1388		RMS_コミットチェック	5423 [GUI]フォームGUIの入力内容をエスケープせずに利用している		2013/07/10 19:08	TCS 様	WITS	山田	不具合
1363		RMS_コミットチェック	6936 削除されたサブ種別を更新した際のメッセージ不正		2013/07/10 19:09	TCS 様	WITS	山田	不具合

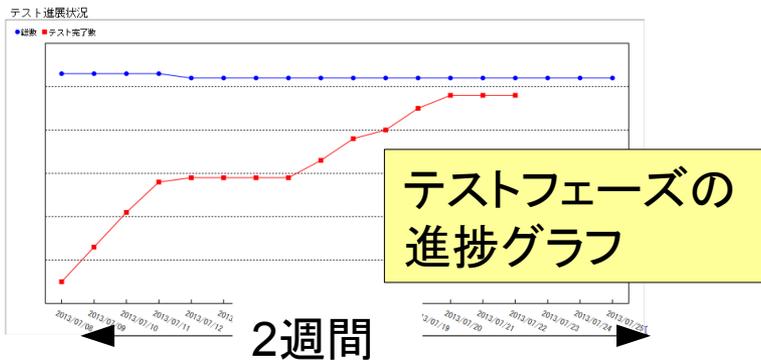
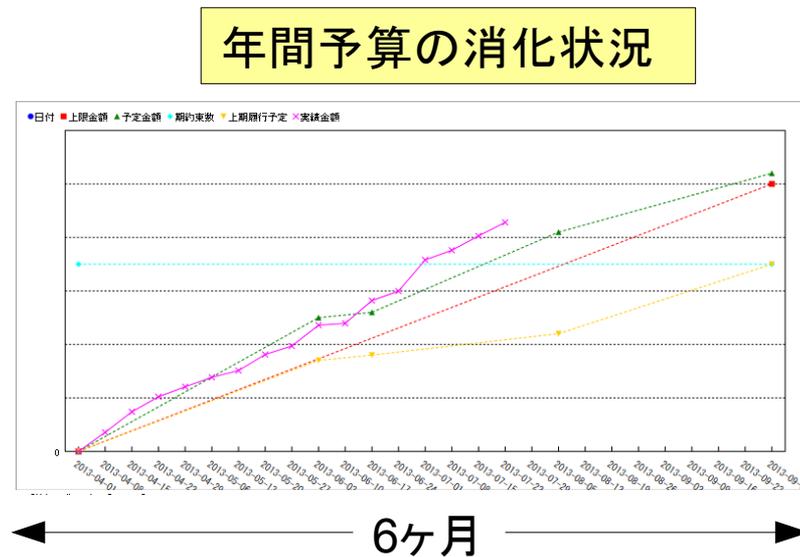
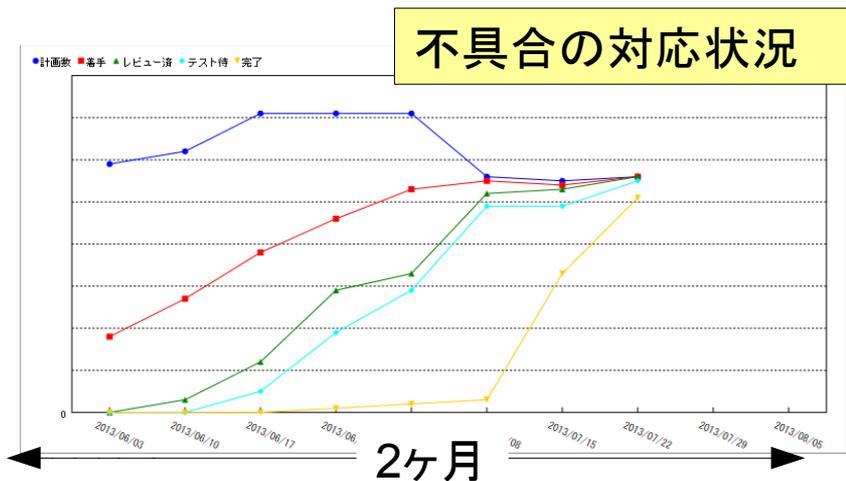
タスク管理ツールですっきり管理

不具合修正のリレーのボタン。今、誰がボタンを持っていますか？



# 6. マネージャの役割: 情報共有と指示(3/4)

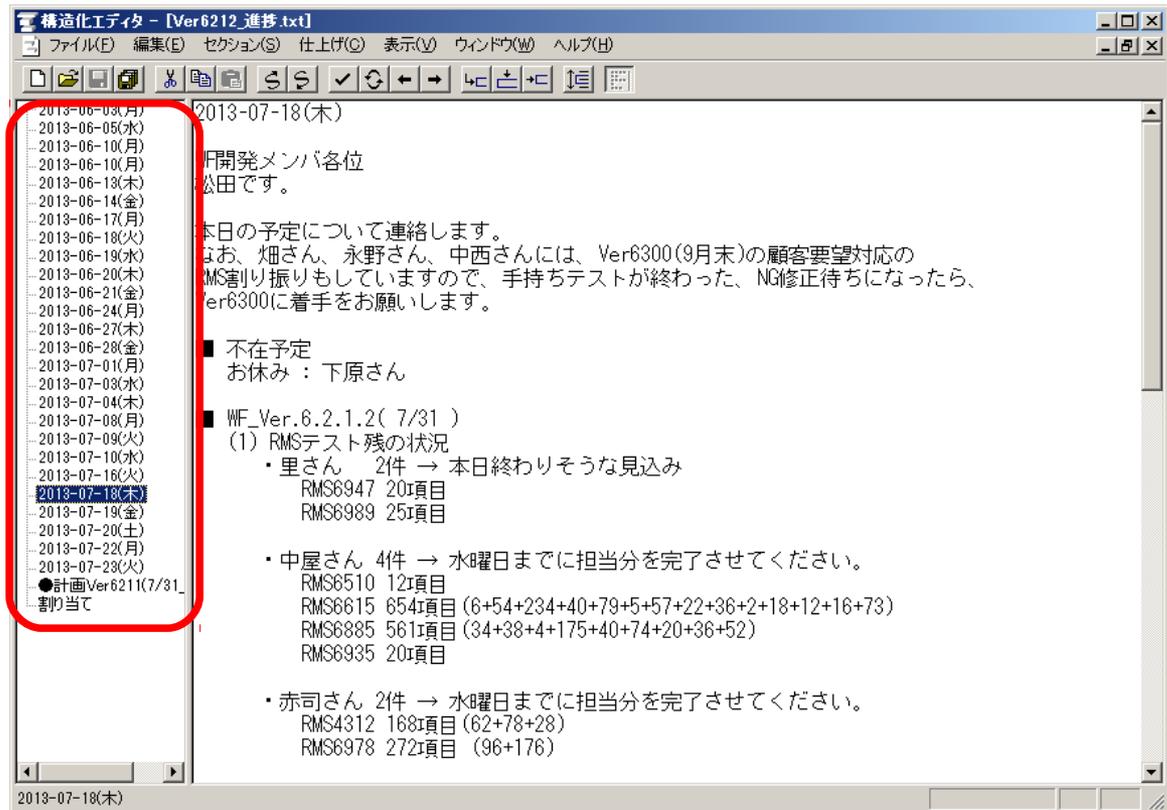
- 進捗状況の可視化：未来は過去の延長です。  
一晩で奇跡の進展は起きません。



## 6. マネージャの役割: 情報共有と指示(4/4)

- リソースの組換：毎日、タスクの負荷平準化を行う

ほぼ毎日  
朝会代わりの  
メールを送る



★ マネージャは朝令暮改を気にせず、  
「その時点での最適化」を維持してください。

## 7. 第三ステップ: 定着と次なる改善

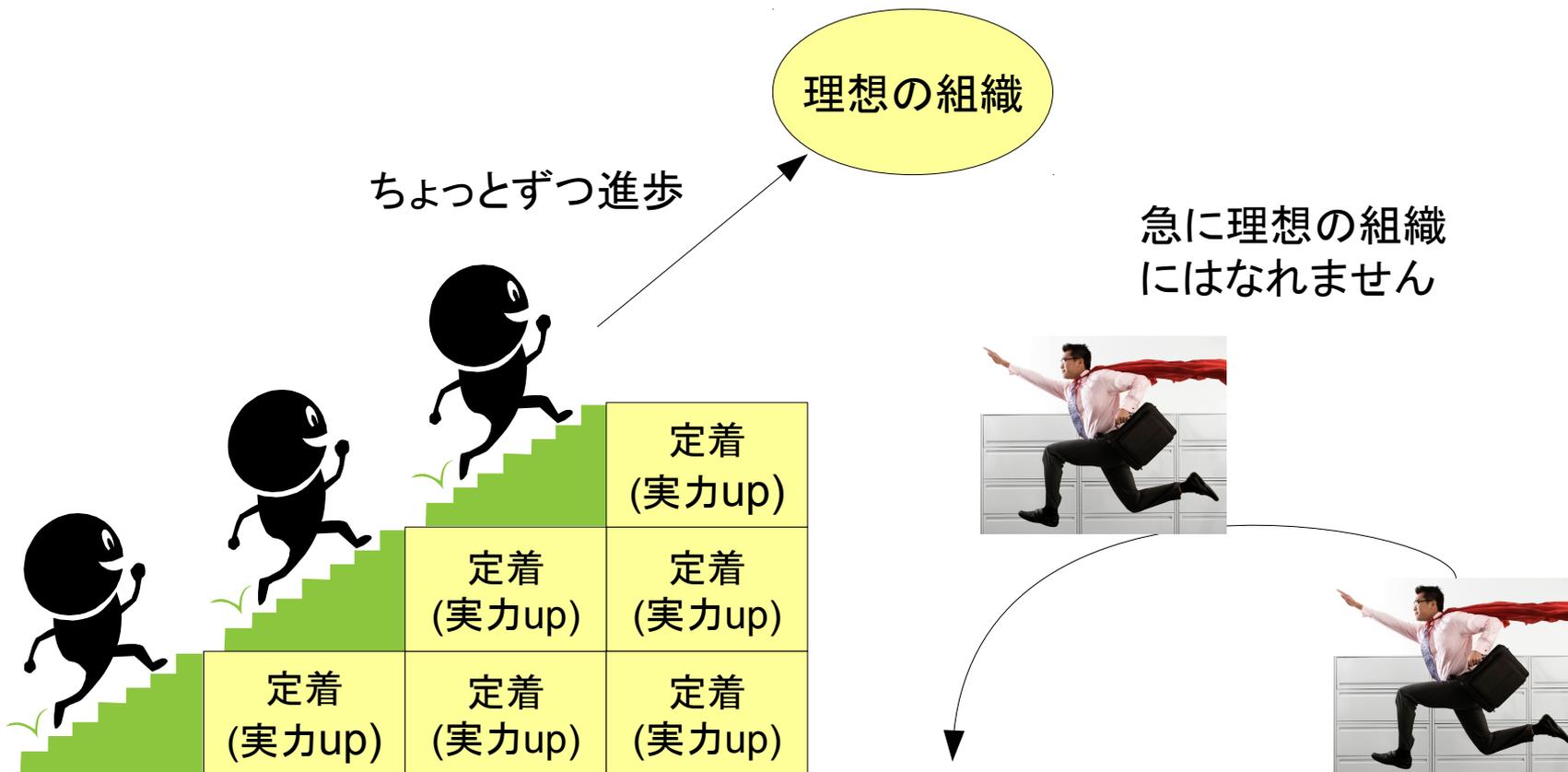
- 定着してきたら、更に**定着**させる
  - 振り返り会で、数値を**連呼し続ける**。
  - 数値があたりまえになれば、定着した証拠。

インシデント 1個あたりの平均時間		
	設計	テスト
	目標値→ 12H(1.5日) 8H(1日)	
2012_02_04月	11.7	6.3
2012_04_06月	12.4	5.1
2012_06_07月	11.2	5.1
2012_08_10月	16.7	7.3
2012-10_11月	12.8	7.8
2012-12_01月	9.9	4.7
2013-02_03月	13.4	7.2
2013-04_05月	12.3	5.1

ほぼ安定して  
設計=12H  
テスト=6H

## 7. 第三ステップ: 定着と次なる改善

- ちょっとずつ取り組み、気が付いたら定着していた  
→これこそ「**気が付いたらPDCA**」



## 8. まとめ ～PDCAを導入するには～

1. 目的を決め、「そのためには」を5回繰り返して、手の届く目標にブレークダウンしましょう。
2. 小さな定着を積み重ねて、気が付いたら実力upしていたと思えるようにしよう。
3. 達成の成果は、できるだけ数字で示すようする。
4. 組織には「PDCA」とは言わず「気が付いたらPDCAになっていた」で充分。
5. ただしリーダーは「PDCA」を意識して仕込みを入れる。