

“五月雨式な”大規模組み込みソフト開発における メトリクス活用事例

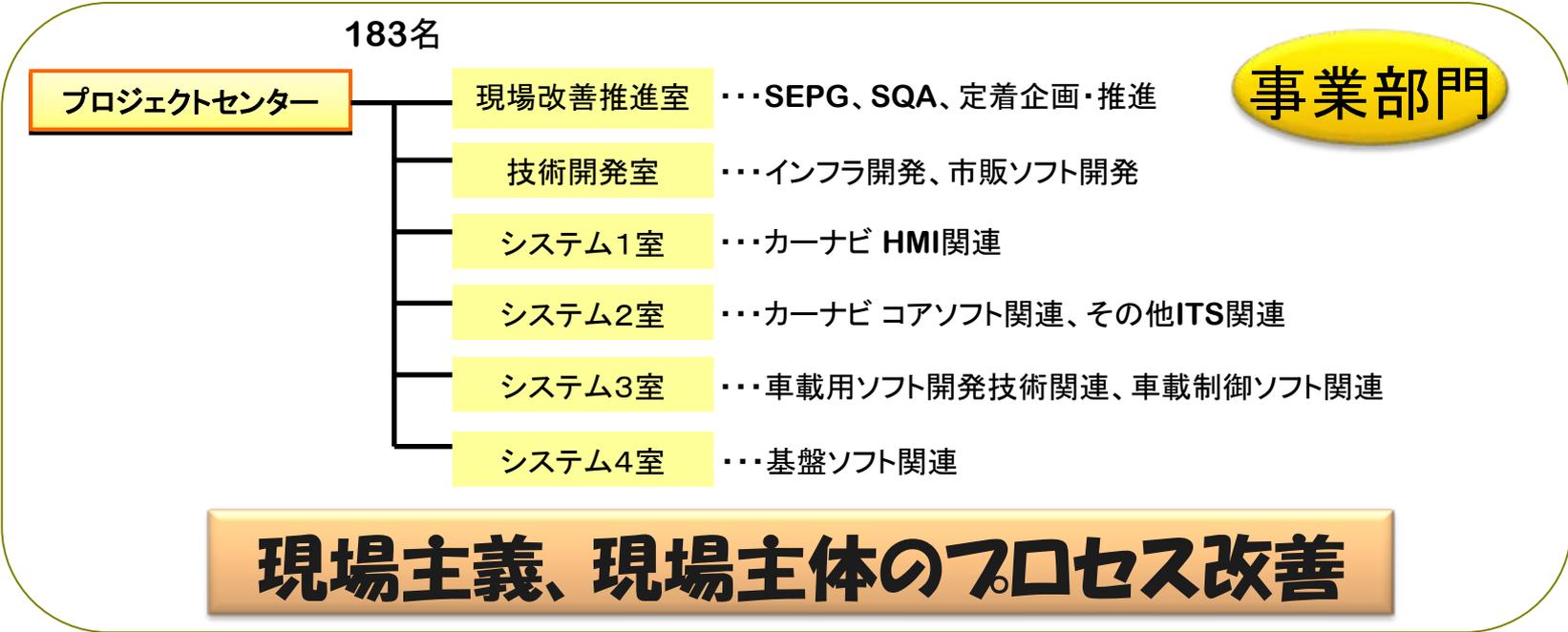
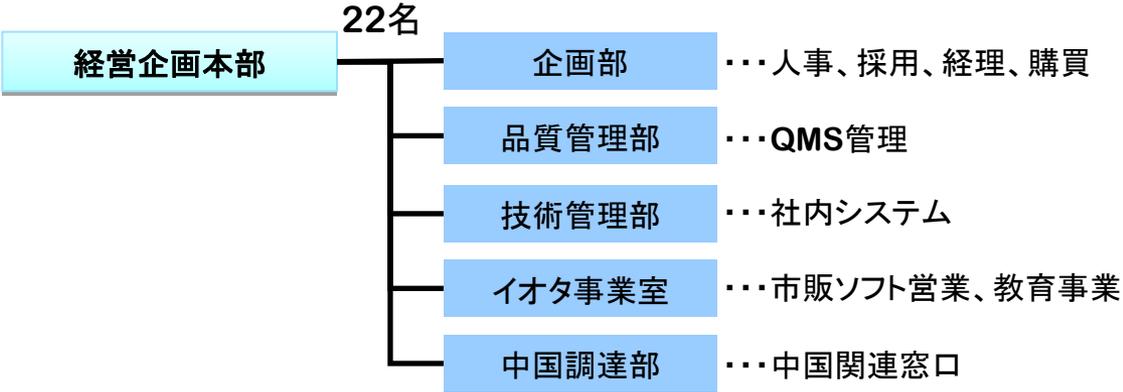
トレーニング指向アプローチによる
“考え使う”業務スタイルへの挑戦

(株)デンソークリエイト プロジェクトセンター 山路 厚

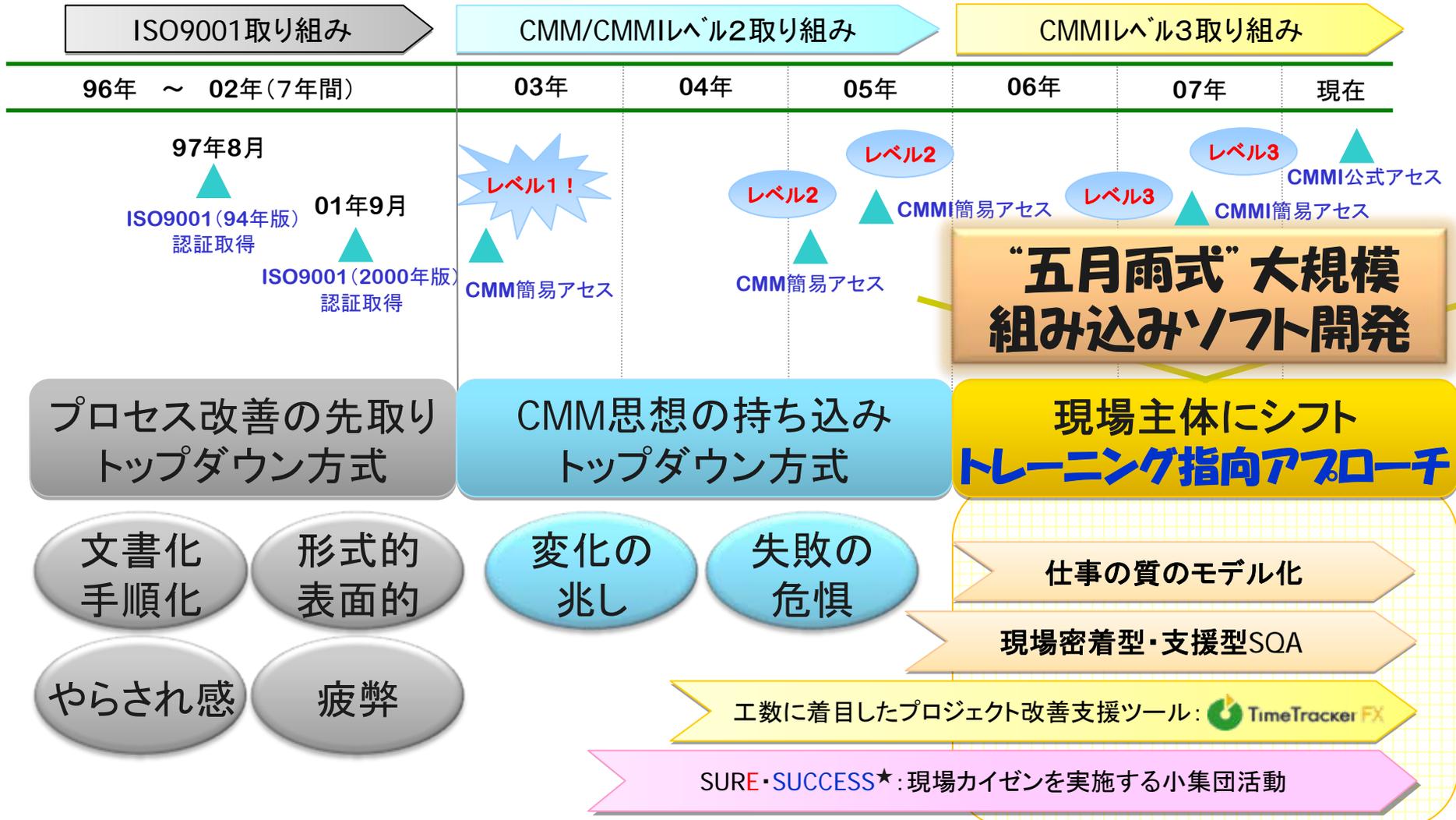
目次

1. プロセス改善の体制
2. プロセス改善の経緯
3. 五月雨式な組み込みソフト開発
4. 失敗経験
5. トレーニング指向アプローチ
6. 適用方式
7. 適用事例
8. まとめ:勉強したこと

1. プロセス改善の体制



2. プロセス改善の経緯



プロセス改善の先取り
トップダウン方式

CMM思想の持ち込み
トップダウン方式

現場主体にシフト
トレーニング指向アプローチ

文書化
手順化

形式的
表面的

変化の
兆し

失敗の
危惧

やらされ感

疲弊

仕事の質のモデル化

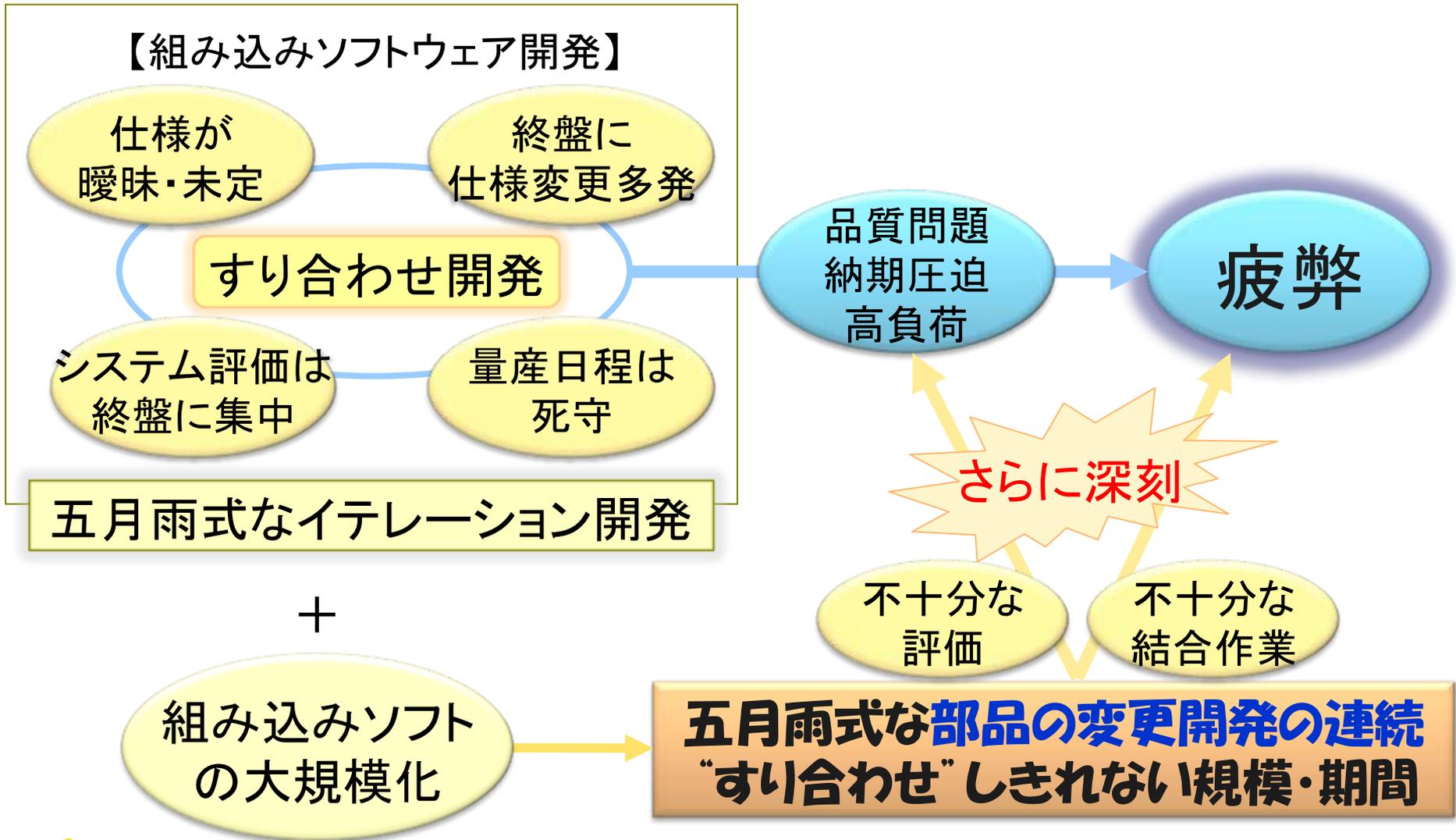
現場密着型・支援型SQA

工数に着目したプロジェクト改善支援ツール: TimeTracker FX

SURE・SUCCESS★: 現場カイゼンを実施する小集団活動

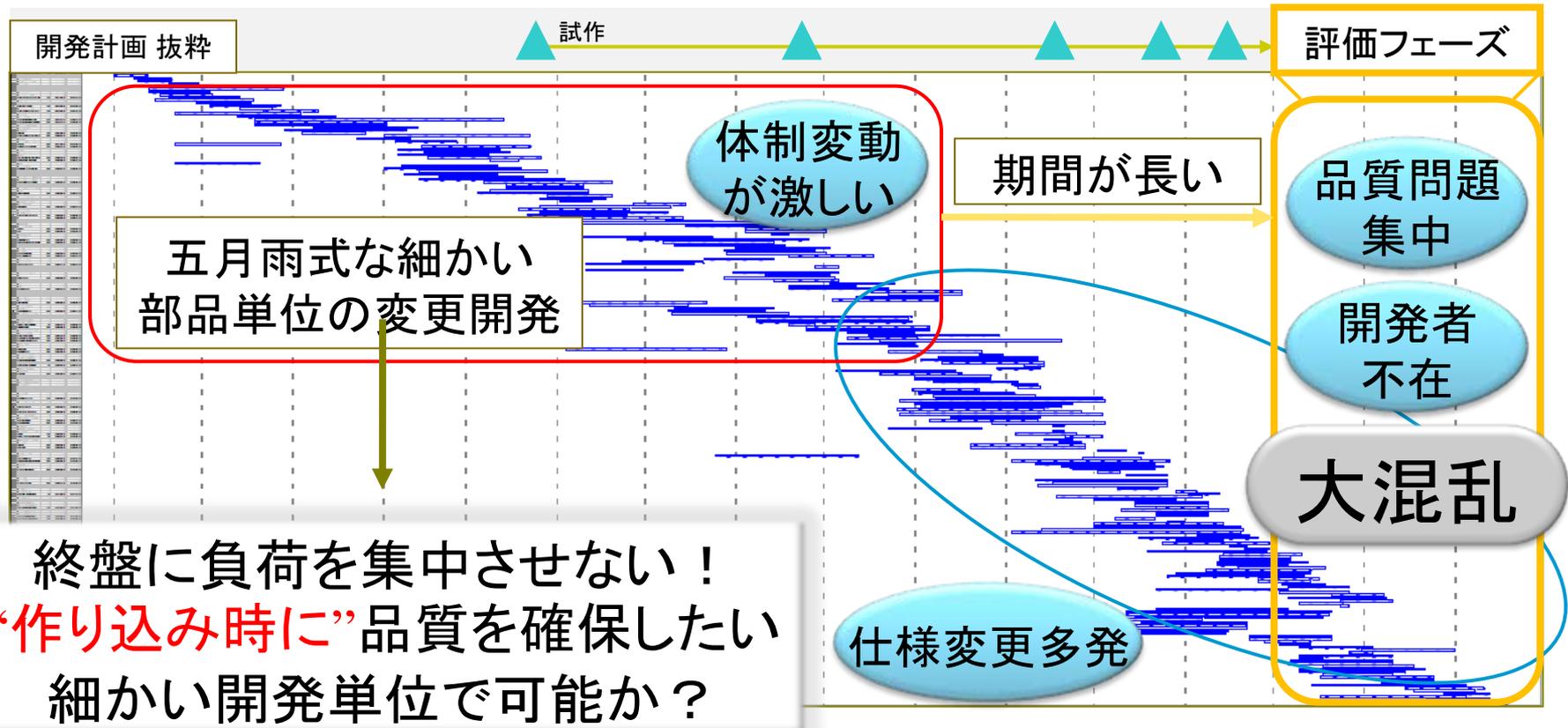
★ SURE: Step Up Review for Effectの略。社内用語
SUCCESS: Step Up Cheerful Circle for Effect, Skill, Satisfactionの略。社内用語

3. 五月雨式な組み込みソフト開発：大規模化



3. 五月雨式な組み込みソフト開発：変更開発の連続

開発現場の状況



メトリクスを使った確実な工程完了判断
→ **品質の見通しを良くしたい！**

3. 五月雨式な組み込みソフト開発：メトリクスが使えるか

開発期間が短い

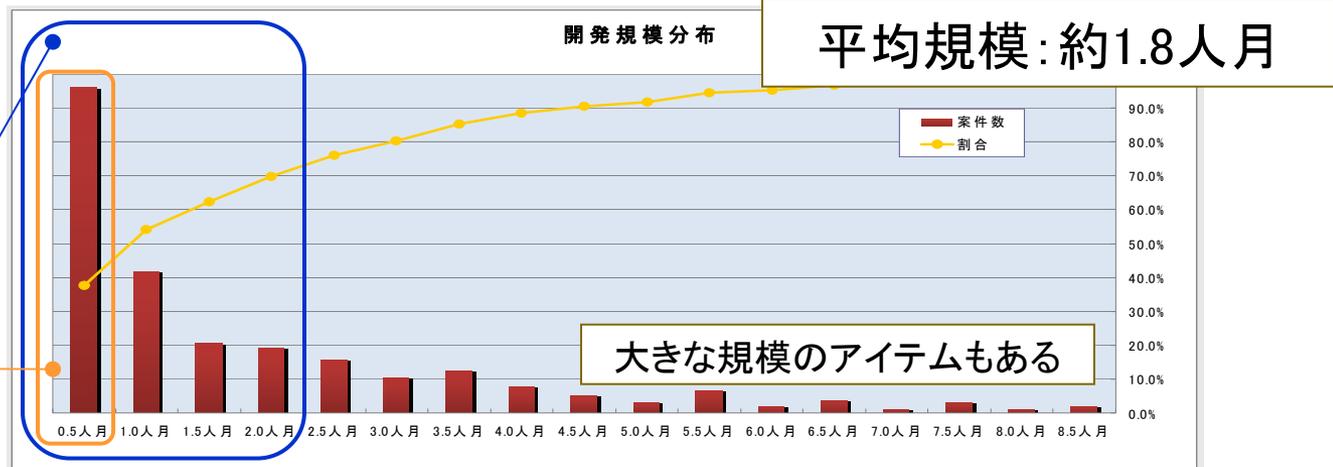
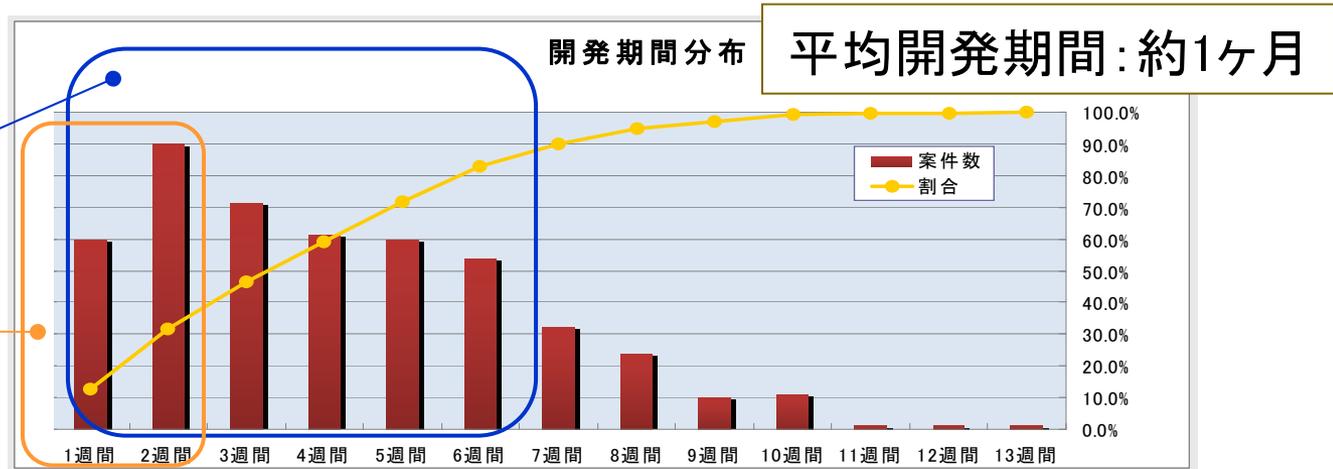
1.5ヶ月以下：約80%

2週間以下の
緊急対応が約30%

小規模アイテム
の集まり

2人月以下：約70%

0.5人月以下の
緊急対応が約40%



小規模なアイテム(分母が小さい)の集まり
→ 品質データのバラツキが大きくなる要因



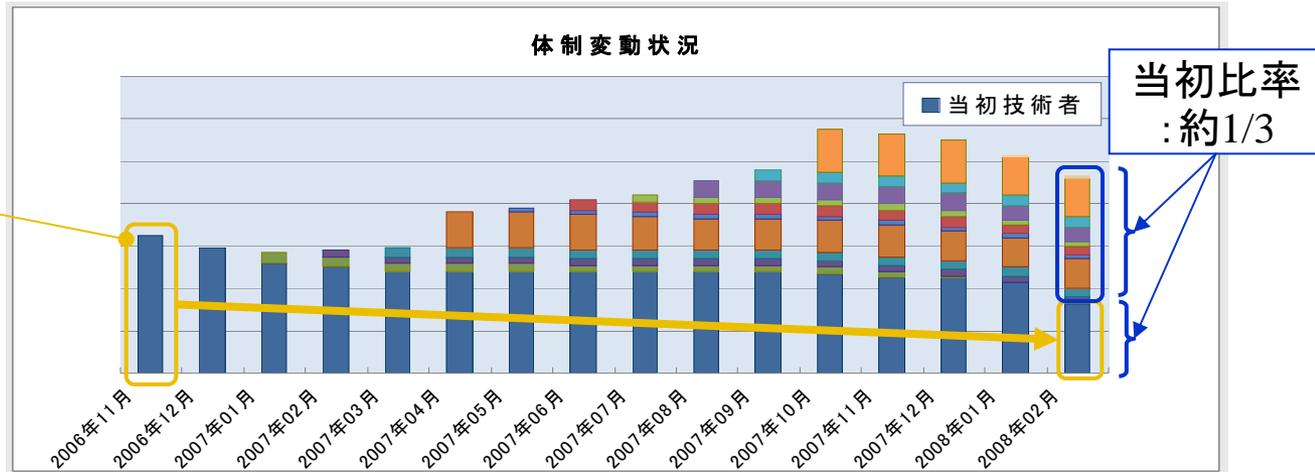
1工程あたり
2日、3日の開発期間に
なったりするんだ！

3. 五月雨式な組み込みソフト開発：メトリクスが使えるか

体制変動が激しい

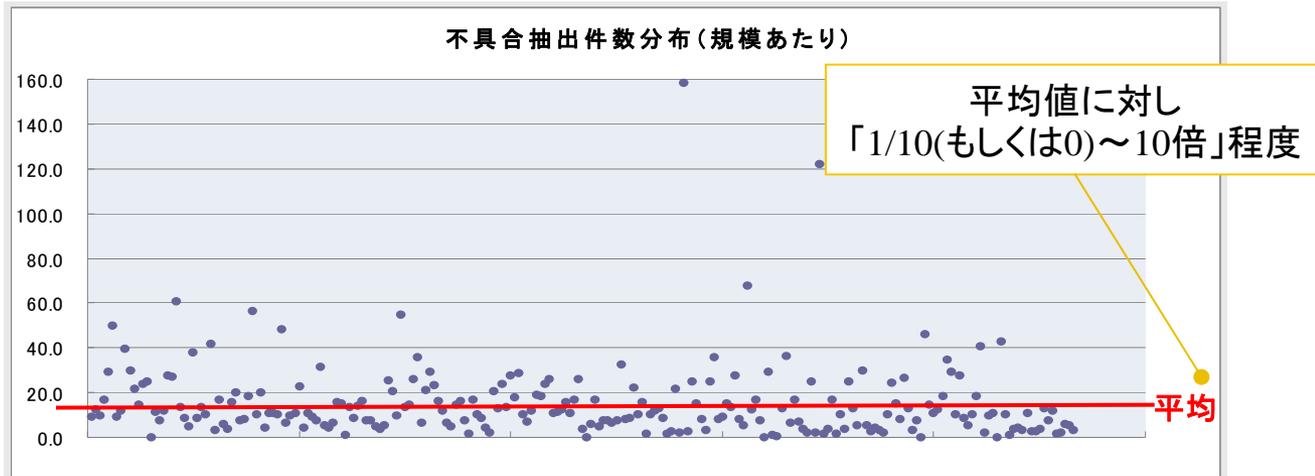
歩留まり率：約50%

人の変動は
バラツキの大きな要因



品質データの
バラツキが大きい

アイテム毎のバラツキ
100倍以上



アイテム毎で妥当な工程完了判断が難しい
マクロ値では開発現場に合わない(そのままでは使いにくい)

4. 失敗経験：陥りやすいミス



ここまでの“おさらい”

- ・五月雨式な部品の変更開発では、終盤に負荷が集中する
作り込み時に品質を確保したく、メトリクス活用による管理手法に取り組む
- ・小規模な開発アイテムの集まり → 品質データのバラツキが大きい
- ・蓄積したメトリクスでは、開発現場が使いにくい状況



どうする？



基準を整備します。
細かな基準を作ります。
メトリクスの利用基準を作ります。

精度向上に取り組めます。
さらにデータを蓄積して
メトリクスの精度を上げます。

陥りやすいミス

基準マニア

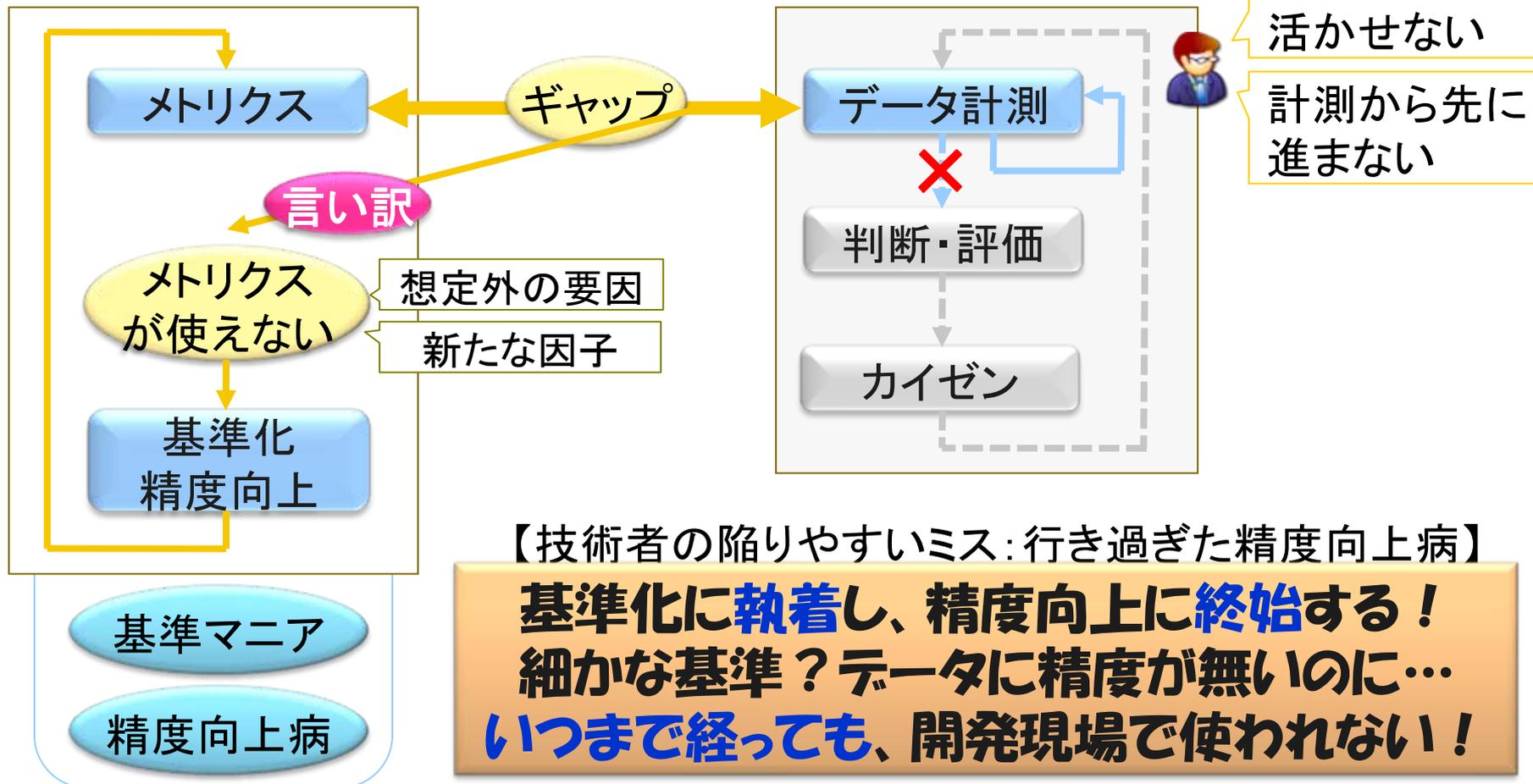
精度向上病

失敗経験

**基準作り・精度向上活動に終始し、
開発現場で活用されずに終わった！（徒労に終わる）**

4. 失敗経験：基準マニア・精度向上病

陥りやすいミス



4. 失敗経験: 気づき

①理系人間は、方程式を欲しがる！

方程式・法則・公式に従って仕事がしたい。“勝利の方程式”好き。
“この場合は、こうすれば良い”という成功方法が、“事前に”欲しい。
「銀の弾丸」も同じ理由？ 最初に決め、後は考えずに仕事がしたい。
モノに対する考え方である。

②人のやることは、方程式通りにはいかない！

しかし、ソフトウェア開発は「人のやること」なので、方程式通りにはいかない。
その場の状況に合わせ、考えながら仕事をするのが重要。
状況を見るためには、人とのコミュニケーションが不可欠。
しかしながら、ソフトウェア技術者は「人と関わりを持つこと」が苦手(嫌い)。
だから、関わりを持たないでも成功する“より完璧な”方程式・法則を求める。

③人に着目した(焦点をあてた)アプローチ！

そこで、ソフトウェア開発は人に着目したアプローチが“特に”重要。
「状況に合わせて考える」という要素を入れた方式。人の能力を引き出す。

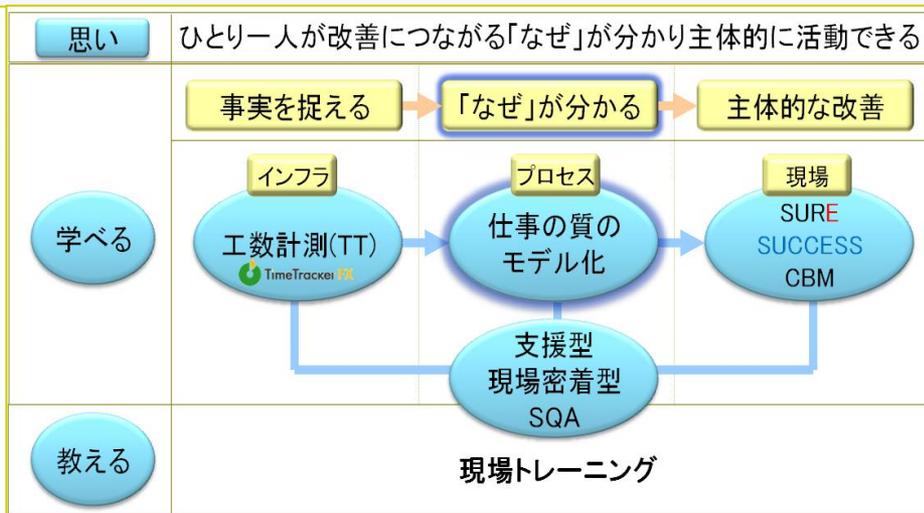
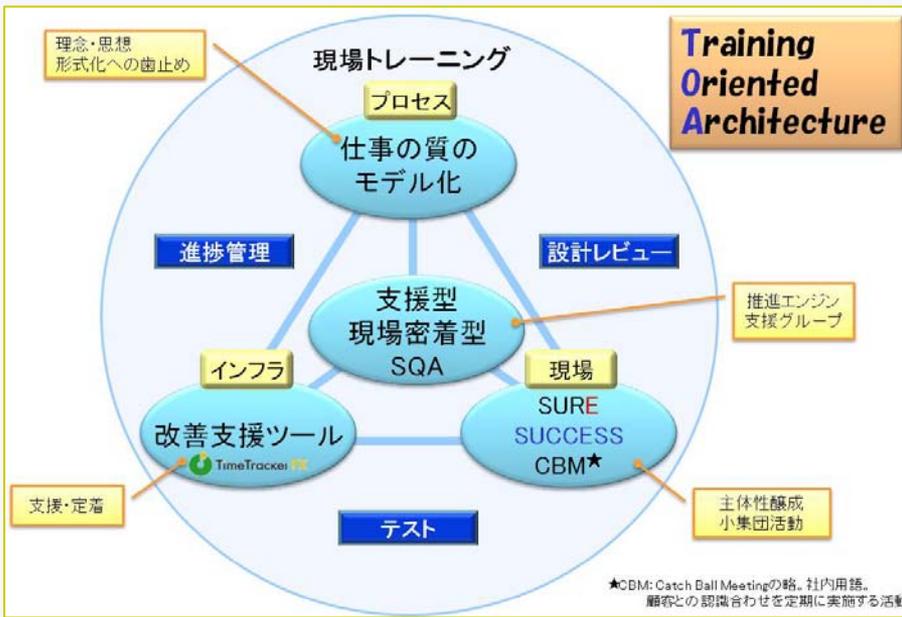
「トレーニング指向アプローチ」 → 考えてメトリクスを使う

5. トレーニング指向アプローチ: 仕事の質のモデル化

SPI Japan2007: 「仕事の質のモデル化」は“自ら考える”業務スタイルに導く仕組み

仕組み

人が育つ



仕事の質のモデル化は、「なぜ」が分かるに手当した仕組み

着眼点

「仕事の質のモデル化」を適用することで
メトリクスを使う業務スタイルへ!

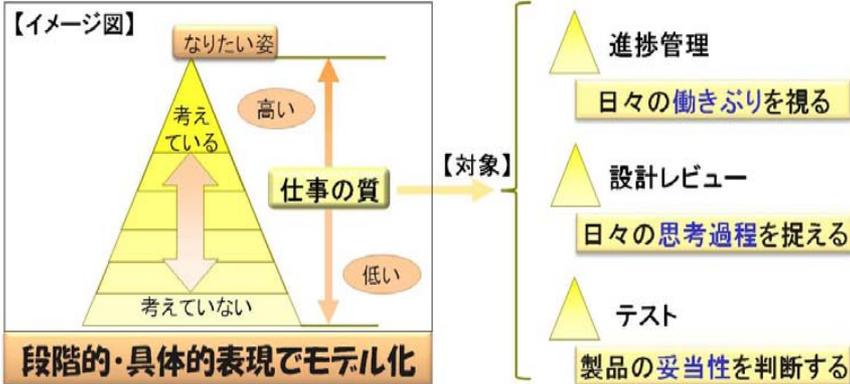
5. トレーニング指向アプローチ: メトリクスを使う

SPI Japan2007: “やり方”を**考える**。考えて“判断・評価”する。

どんなモデル?

“仕事ぶり”を良くする → **自ら考える業務スタイルで仕事を行う**

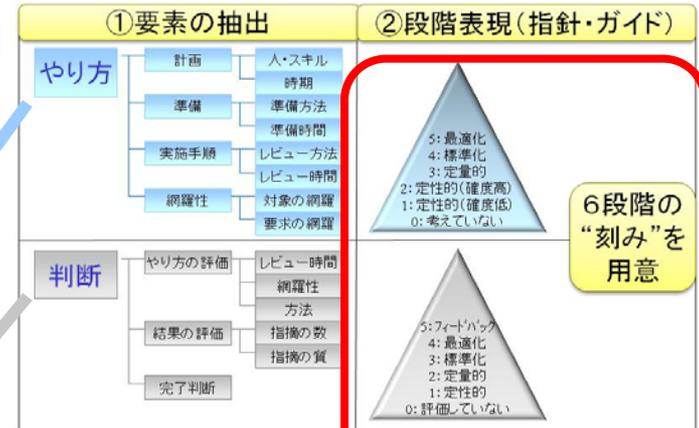
モデル化の概念
根拠のレベルを“仕事の質”とする
仕事の妥当性を如何に考え判断したか



モデル化の方式

- 【モデル化の方式】
- ①仕事の質に影響を与える要素を抽出
「やり方」「判断」という切り口で抽出
 - ②「なりたい姿」を段階的・具体的な形式で表現

【例: 設計レビュー】



成長モデルを活用

やり方

- 5:最適化
- 4:標準化
- 3:定量的
- 2:定性的(確度高)
- 1:定性的(確度低)
- 0:考えていない

仕事の特徴(規模・人等)を考慮してメトリクスを使う

判断

- 5:フィードバック
- 4:最適化
- 3:標準化
- 2:定量的
- 1:定性的
- 0:評価していない

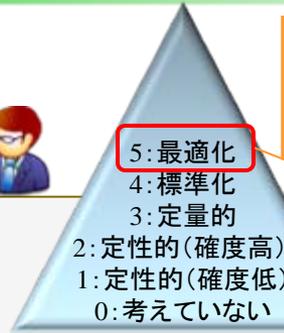
仕事の特徴(規模・人等)を考慮して判断・評価する
仕事のやり方にフィードバックする

6. 適用方式: 取り入れ方

開発の中で使う方式



① 計画段階で仕事の特徴を自ら考え、メトリクスから“計画指標”を設定する



【開発現場】



使う



② レビュー時に計画指標・実績値を使い判断・評価する実績を振り返りメトリクスにフィードバックする

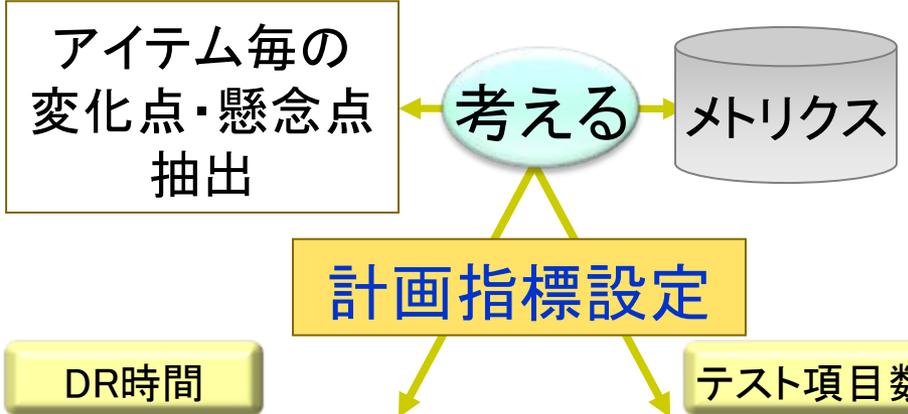
陥りやすいミス



現場の中で使う！ 使うことで現場もデータも育つ方式！

6. 適用方式: ①計画指標の設定の具体例

計画の抜粋	2007年09月23日				2007年09月30日				2007年10月07日				2007年10月14日			
	日	月	火	水	日	月	火	水	日	月	火	水	日	月	火	水
アイテム開発	アイテム開発															
仕様分析ワークフロー	仕様分析ワークフロー															
の実験ワークフロー	の実験ワークフロー															
詳細設計/製作ワークフロー	詳細設計/製作ワークフロー															
テスト設計ワークフロー	テスト設計ワークフロー															
テスト実施ワークフロー	テスト実施ワークフロー															
不具合対応ワークフロー	不具合対応ワークフロー															
仕様分析ワークフロー	仕様分析ワークフロー															
の実験ワークフロー	の実験ワークフロー															
詳細設計/製作ワークフロー	詳細設計/製作ワークフロー															
テスト設計ワークフロー	テスト設計ワークフロー															
テスト実施ワークフロー	テスト実施ワークフロー															



アイテム名	詳細設計						ソースコード						テスト設計									
	機能不具合検出件数		成果物不具合検出数		DR時間		行数		機能不具合検出件数		成果物不具合検出数		DR時間		行数		機能不具合検出件数		成果物不具合検出数		DR時間	
	計画	実績	計画	実績	計画	実績	計画	実績	計画	実績	計画	実績	計画	実績	計画	実績	計画	実績	計画	実績	計画	実績
	1.624365	2	1.608	2	2.0	3.3	200	209	2.803738	2	3.48	4	4.2	9.8	500	867	0.098614	0	3.48	6	1.3	
	0.132539	0	0.4704	3	0.4	4.5	7	23	3.273546	3	4.14	8	5.6	7.5	200	331	0.041089	0	3.4	5	0.9	
	0.484018	0	0.4876	1	1.8	2.3	106	196	1.545989	3	1.534	2	2.1	2.3	260	374	0.059168	0	7	8	3.1	
	0.155039	0	0.204															0	9.555	10	2.8	
	1.579244	1	2.38															0	3.24	4	5.3	
	1.243523	1	1.243523															0	5.38125	8	1.9	
	0.378682	2	3															0	4.72	8	1.2	
	5	1	1															0	15.7	17	12.6	
	2	2	1		1.5	3.5	28.08434	56	2	2	2	1	4.3	6.6	629.1904	331	0	0	13.8	9	8.5	
	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	10	9	0.972556	2	8.4	12.4	273	307	0	0	0	0	0.0	0.0	0	0	0	0	2.801358	5	1.2	

工程別に設定(全21項目)
計画段階で“根拠”を考え、値とともに記録

1W~2Wの緊急対応が多い

担当者一人ひとりが“やれること”

一人ひとりの力を引き出せる

現場カイゼンにつながる

小規模アイテムの集まり(全460件)に適用する!

6. 適用方式: ②判断・評価の具体例

成果種別	成果量単位	実績					工程完了承認				
		機能不具合件数	成果物不具合件数	機能不具合成果物不具合合計	非不具合件数	総レビュー時間	総成果量	レビュー品質 (成果量単位/人)	承認日	判断理由	
単体テスト仕様書仕様	見出し							0.11	2008/1/30	成果量の乖離について、当初は■は対象外としていたが、顧客との調整の結果、対応することになったため、計画より増加した。問題なし。 工数品質、機能不具合の成果品質については問題なし。 成果物不具合の成果品質の乖離については、△△の処理など、既存のテスト仕様書を参考にてきため、計画ほど不具合が発生しなかった。問題なし。 工数品質、成果品質ともに問題なし	
	心配点	1	1	2	1	0.50h	3	6.00	0.33	2008/2/1	工数品質、成果物不具合の成果品質については問題なし。 機能不具合の成果品質については、過去実績を元に発生する不具合を見込んだが、今回のアイテムでは画面構成が他の機能と類似していたため、参考のできる部分が多く、不具合が発生しなかった。 計画時では、□□中表示機能が変化した場合のタイマー更新処理を考慮に入れていなかったため、計画時より成果量が増加した。 工数品質については、担当者の経験値を考慮し確認時間を多めに計画したが、比較的容易な変更であり、短い時間で問題なくことを確認できたため、レビュー時間が減少した。品質的には問題なし。
詳細設計書	ページ								0.30	2008/2/9	成果量については、既存の単体テスト仕様書を流用することができたことにより増加した。 工数品質の乖離については、□□中イベント受信した際の確認項目の妥当性確認時間を要したため、計画よりレビュー時間が増加した。 成果品質については、問題なし。 工数品質、成果品質ともに問題なし
ソースコード	行	1	1	2	6	1.83h	113	50.18	0.21	2008/2/9	計画では何件の不具合が発生すると見込んでいたが、基本的には他の△△や□□を参考に設計することができ、複雑な処理もなかったため、不具合が発生しなかった。問題なし。
単体テスト仕様書	項目								0.30	2008/2/5	
テストデータ	行								0.30	2008/2/5	
単体テスト報告書	項目	0	0	0	0	0.05h	127	2540.00	0.00	2008/2/12	

自ら考えた指標
真摯に考え使う

行き過ぎない
マニアに陥らない

“なぜ”が分かる
カイゼンにつながる

計画指標と実績値を基に
レビューで工程完了判断

対象規模	計画(予想)		実績	
対象成果物	5.27			
	83.00	部品	90.00	部品
	15.75	部品/規模	17.08	部品/規模
	285.00	関数	429.00	関数
	54.08	行/規模	81.40	行/規模
			200.42	部品
			38.03	部品/規模
			622.28	行
			118.08	行/規模

■計画の根拠と妥当性判断
分析の結果、必要と想定される成果量にて計画。

■計画と実績の差異分析
ほぼ計画通りのため、問題なし。

対象	計画		実績	
	レビュー時間	時間	時間	時間
機能不具合検出数	2.40	24.40	1.62	55.67
成果物不具合検出数	2.00	2.00	0.00	0.00
レビュー時間	2.41	3.00	3.33	1.80
機能不具合検出数	3.00	34.53	4.83	103.25
成果物不具合検出数	9.32	4.00	6.99	12.73
レビュー時間	8.25	7.00	9.00	9.53
機能不具合検出数	16.32	11.66	6.45	9.72
成果物不具合検出数	0.45	6.00	0.82	0.54
レビュー時間	1.14	1.14	3.00	4.89
機能不具合検出数	10.00	1.90	12.00	1.19
成果物不具合検出数	1.90	2.26	2.26	1.19

■計画の根拠と妥当性判断
仕様分析の結果および担当者の経験から計画。

■計画と実績の差異分析
レビュー時間については、計画では2人分を考慮した時間、実績はレビューそのものの時間であるため、ほぼ計画通り。
機能不具合検出数については、過去実績を元に発生する不具合を見込んだが、□□、△△、■は機能確定が他の機能と類似していたため、参考のできる部分が多く、不具合が発生しなかった。問題なく作成されている。

統合時に実施される計画となっている。

「エビデンス作成マニュアル」に従い、Diff-Grepによる確認が実施されている	OK
最終プログラムに対し、QACがかけられ、顧客依頼項目が漏れなく除去されている	OK

自ら考えた計画指標と実績を対比し、工程完了判断!



7. 適用事例: プロジェクト概要と効果の測定方法

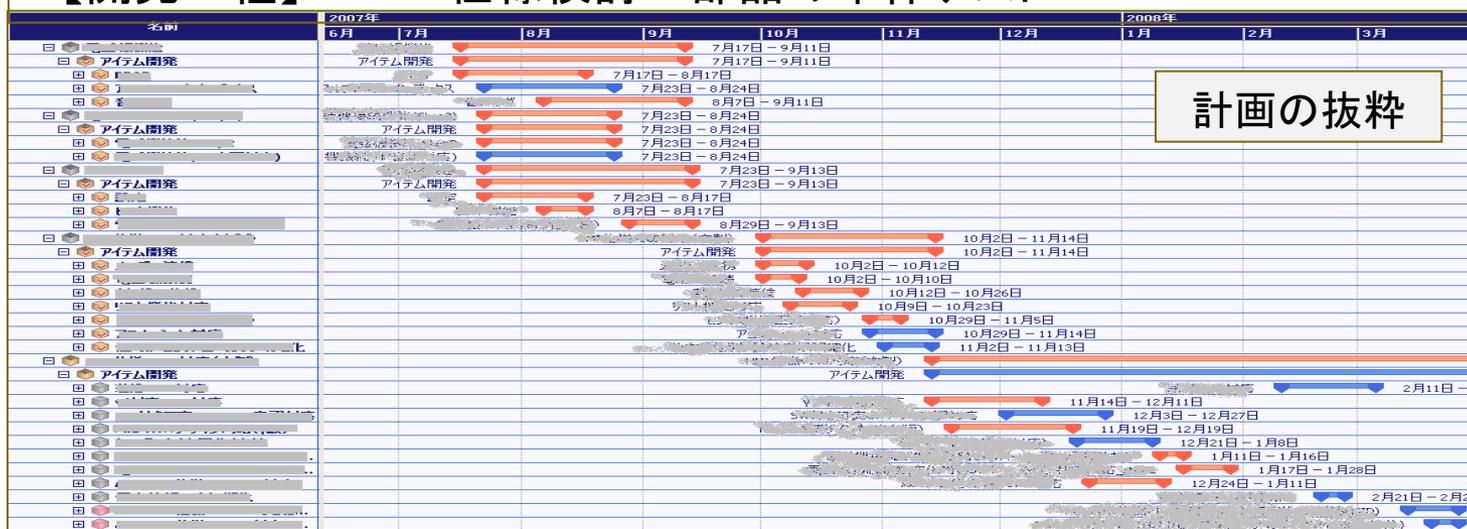
五月雨式な組み込みソフトウェア部品の変更開発

【期間】 2006/11～2008/3(約17ヶ月)

【規模】 約1000人月

【開発アイテム数】 460アイテム → 全てに適用した

【開発工程】 仕様検討～部品の単体テスト

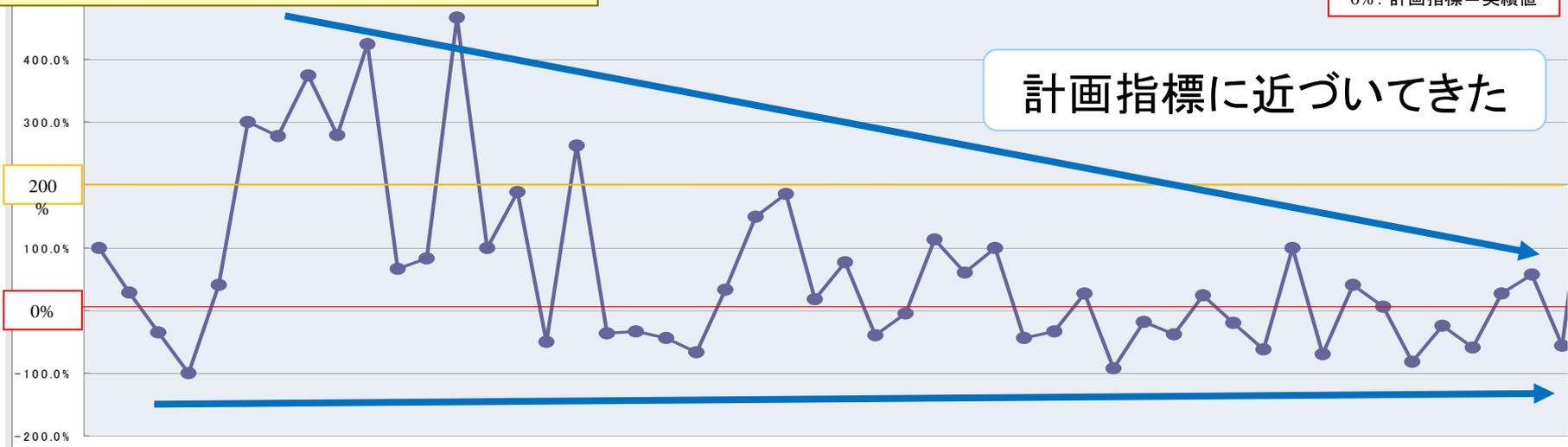


効果の測定方法

- ① 計画指標と実績値との差 → チームとしての成長度合い
- ② 流出不具合率・作り込み欠陥率・手戻り工数率
→ 作り込み品質の向上度合い

7. 適用事例：内製チームの“成長度合い”

抽出欠陥：計画指標と実績値との差



不具合流出率



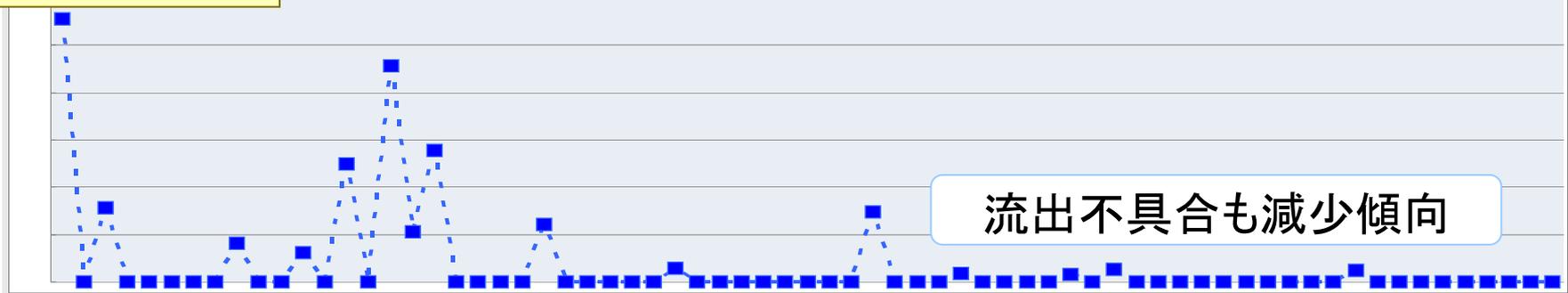
時間経過とともに、計画指標に近づいてきた
流出不具合があり、まだまだ**成長途中**である

7. 適用事例：国内協力会社チームの“成長度合い”

抽出欠陥：計画指標と実績値との差



不具合流出率



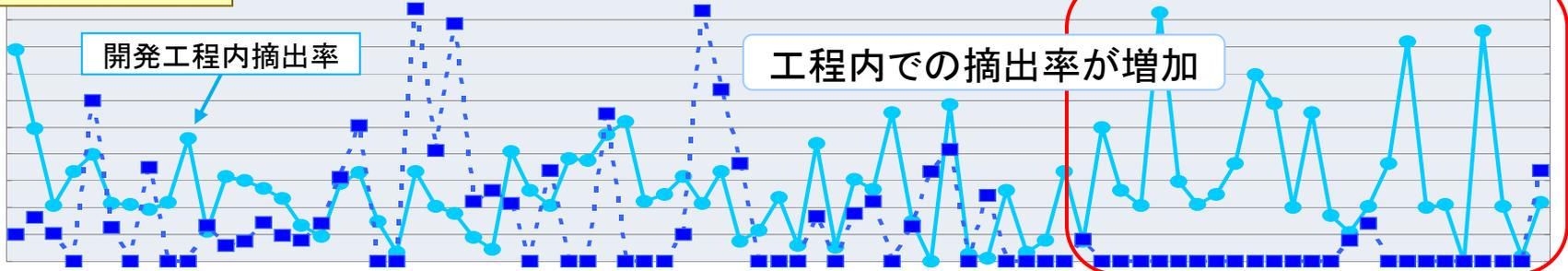
時間経過とともに、計画指標に近づいてきた
 計画指標より実績値が下回る傾向にある
 流出不具合率が低下 → **成熟してきた**

7. 適用事例：海外協力会社チームの“成長度合い”

抽出欠陥：計画指標と実績値との差



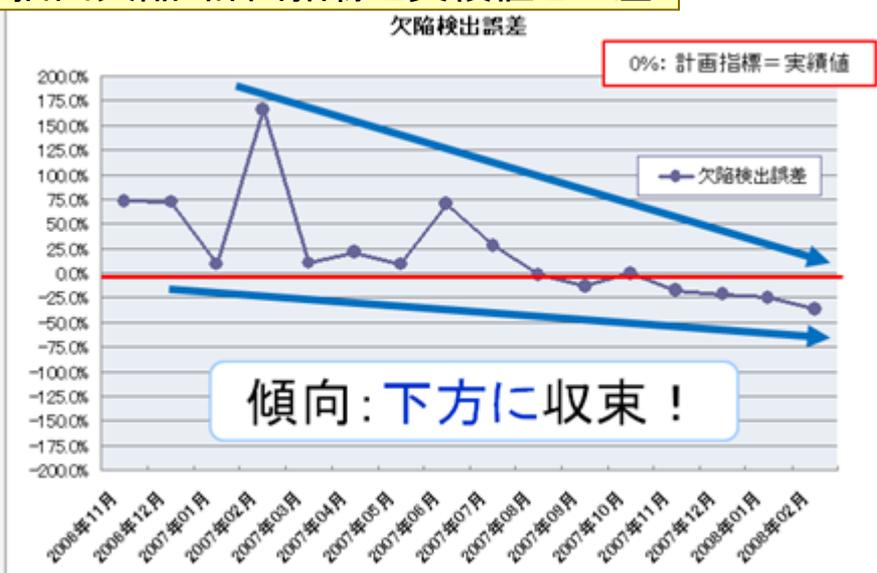
不具合流出率



当初から差が小さい → データ操作の疑いあり
流出不具合率は減少、摘出不具合率は増加
→ 実力が分かり、品質活動・管理活動で未然に防止

7. 適用事例: チーム全体としての評価(1/2)

抽出欠陥: 計画指標と実績値との差



手戻り工数率



時間経過とともに、計画指標に近づいてきた
 計画指標を下回る傾向が確認できた
 手戻りが削減され、チームが成長してきた

気づき

【計画より下回る傾向】

計画段階で「課題・懸念点」を考えるから、意識され手が打たれる。
 結果として、計画より“作り込まれる”欠陥が減っていく!

7. 適用事例: チーム全体としての評価 (2/2)

作り込み品質が向上するメカニズム

【考える業務スタイル】

事前に考える
考えて判断・評価
(現場で使う)

今回の方式

意識

レビューの質
が上がる

相乗効果

作り込まれる
欠陥が減る

流出不具合
が減る



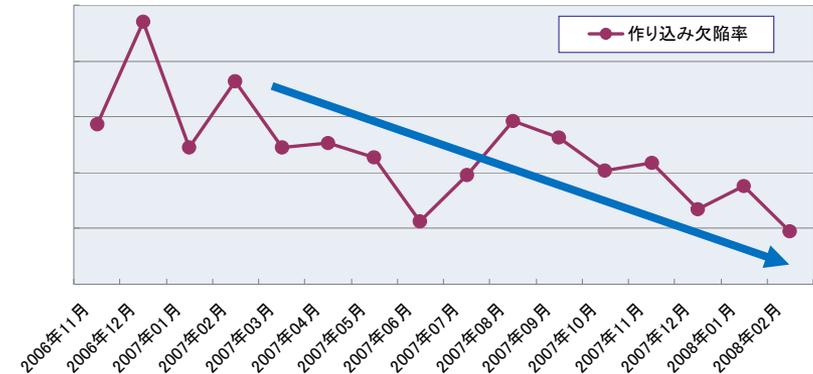
計画より下回る傾向

レビューの質



補足: “仕事の質のモデル”を基に内部アセスメントにて診断した結果より

規模あたりの作り込まれる欠陥率(月別)

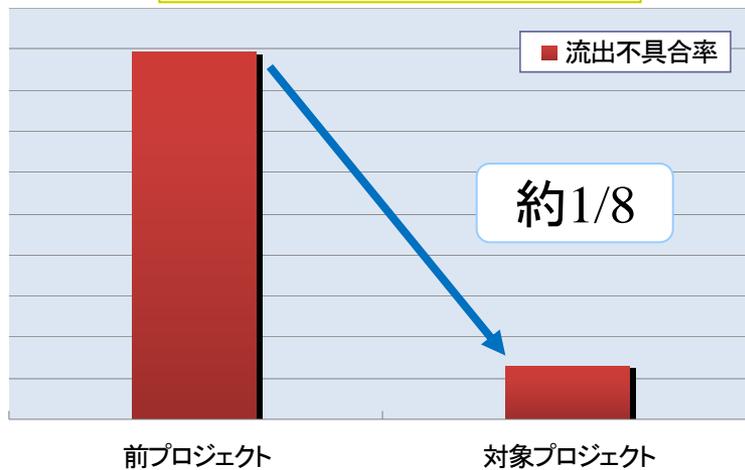


レビューの質が上がリ、作り込まれる欠陥が減ってきた

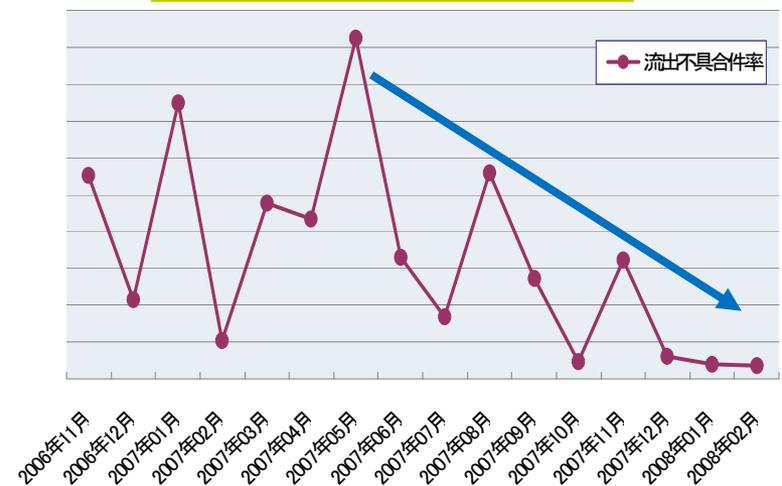
7. 適用事例：効果の確認

品質向上：流出不具合が約1/8に低減
開発が進むにつれ、流出不具合率が低下

規模あたりの流出不具合率(全体)



規模あたりの流出不具合率(月別)



作り込みの品質向上に効果あり！
自ら“考え使う”ことが、日々のトレーニングとなる！

8. まとめ: 勉強したこと

- ①基準マニア・精度向上病・・・銀の弾丸**
理系人間は、“**事前に**”方程式・法則・公式を欲しがる
人のやることは方程式通りにはいかない！解はいろいろ！
- ②事前に考える・・・意識が自然に“手を打つ”**
事前に考えると、事実と**真摯**に向き合える
事後に考えても、**言い訳**になりやすい
- ③データの質・・・使うことが“質を研ぐ”**
開発現場が“**考え使う**”ことにより、データの質が高まる
“なぜ”が分かり、“**自ら成すべきこと**”が見えてくる
管理データが**改善のためのデータ**に変わってくる

ソフトは人ない。人に着目した(焦点をあてた)アプローチ！

END:ありがとうございました

END:ありがとうございました