



軽量開発プロセスにおける Tracを用いたメトリクスの収集・蓄積・利用

株式会社NTTデータ技術開発本部

○大杉 直樹、伏田 享平、井ノ口 伸人、新井 広之
丹羽 隆、 藤貫 美佐、戸村 元久、 木谷 強

2013年9月12日

健康のために階段を使いましょう！

- 何階までなら階段を使える？
 - 2つ上の階まで？
 - 5つ上の階まで？
 - 25上の階まで！？
- いつなら階段を使える？
 - 始業15分前なら？
 - 始業10分前なら？
 - 始業3分前！？
- そもそも、健康のために上の階に上っているのではない。
- 時間までにオフィスに到着することが第一の目的。



管理のためにメトリクスを測りましょう！

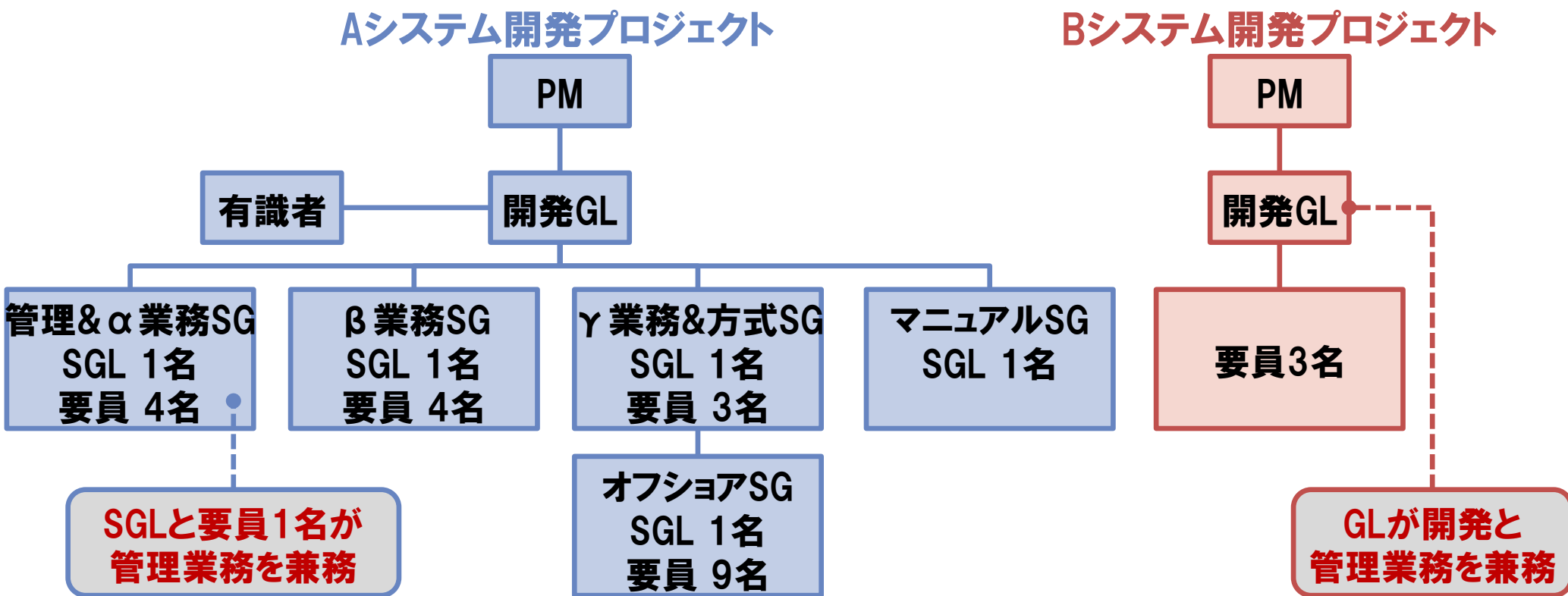
- 測定に使える時間は？
 - 1人時／週？
 - 1人日／週？
 - 5人日／週！？
- 分析に使える時間は？
 - 1人時／週？
 - 1人日／週？
 - 5人日／週！？
- そもそも、メトリクスのために開発をしているのではない。
- 納期までにシステムを完成することが第一の目的。



筆者らが参画したシステム開発では・・・

■ 管理業務(下記)は開発と兼務、大きな工数は確保できない。

- 開発プロセス、進捗管理・品質管理手順のルール整備・監視。
- 進捗管理・変更管理・構成管理ツール準備と使用ルール整備・監視。
- 開発計画策定、上位層報告、そして、メトリクス収集・蓄積・利用。



※PM: Project Manager, GL: Group Leader, SG: Sub Group, SGL: Sub Group Leader

2つのシステム開発プロジェクトの事例

- できるだけ軽いシステム開発プロセスを設計し、小さいコストでメトリクス計測・蓄積・利用できるよう努力した事例を報告する。

	Aシステム開発プロジェクト	Bシステム開発プロジェクト
開発対象	生産管理系社内システム	情報系社内システム
アーキテクチャ	Web三層 & Windowsクラサバ	Web三層
規模	中規模	小規模
要員数	ピーク時80名、 サービス開始後26名	ピーク時4名、 サービス開始後2名
言語・FWなど	Python, VB.NET, Trac, PostgreSQL	Java, JavaScript, JSP, Struts2, iBatis, MySQL
特徴	全社で標準的に使用されるシステム。稼働率とリカバリ要求などある程度の品質が要求される。	全社、G会社から使われるが研究開発色が強い。要求品質は高くないが、安く作る必要がある。

だからお前は誰なんだよ

■ 大杉 直樹

- 専門: ソフトウェア工学
(実証的ソフトウェア工学)

■ 経歴

- 2004年9月
 - ◆ 奈良先端科学技術大学院大学
博士後期課程修了
- 2004年10月～2007年3月
 - ◆ 同大学 特任助手(ポスドク)
- 2007年～現在
 - ◆ 株式会社NTTデータ技術開発本部 勤務

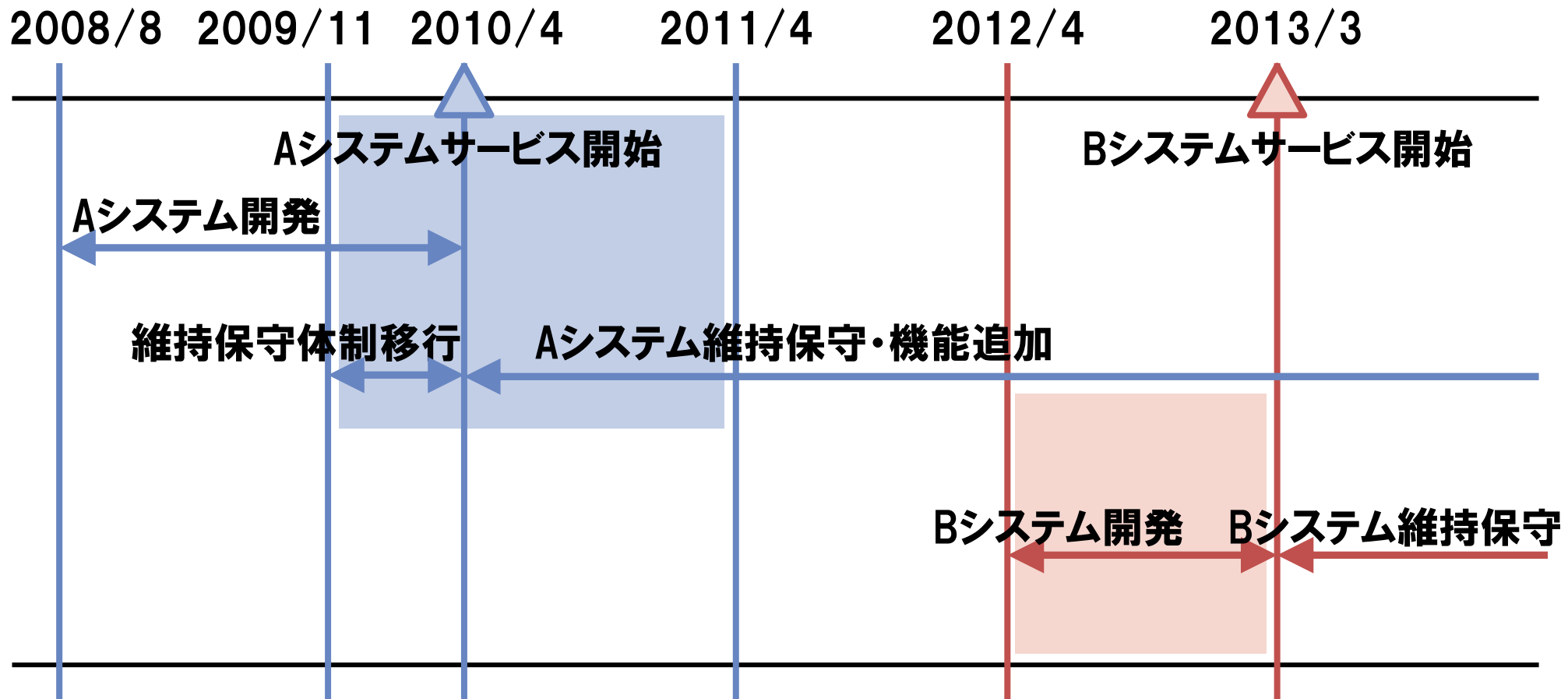


■ アクセス

- Twitter: @ohsugi

本稿での報告対象期間

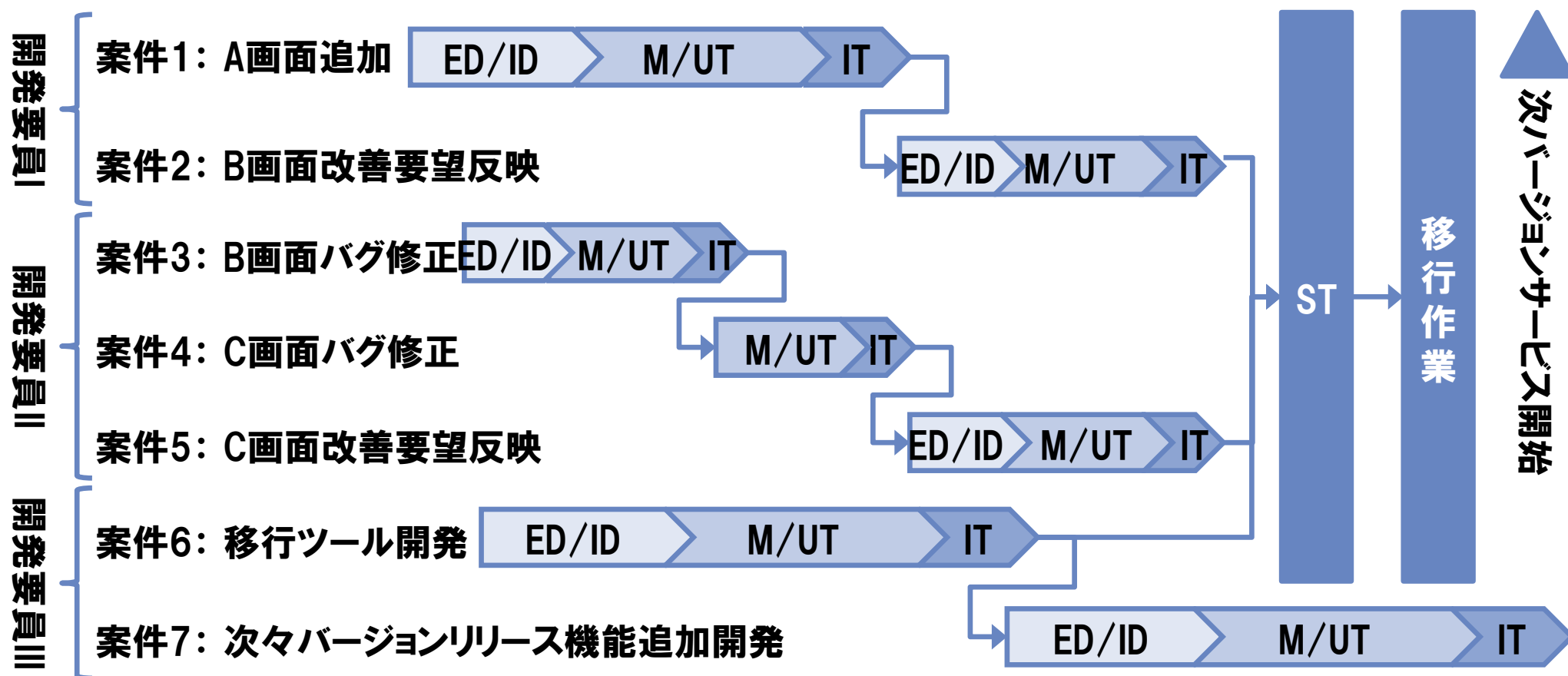
- 下記の、のべ29ヶ月間の運用事例について報告する。
 - 著者のうち3名が管理業務を実施し、状況をよく把握しているため。
 - その後もおよそ同様の方法を運用しているが、本発表の対象外。



本稿における軽量開発プロセスの定義

■ アジャイルの知見やプラクティスを取入れたウォーターフォール。

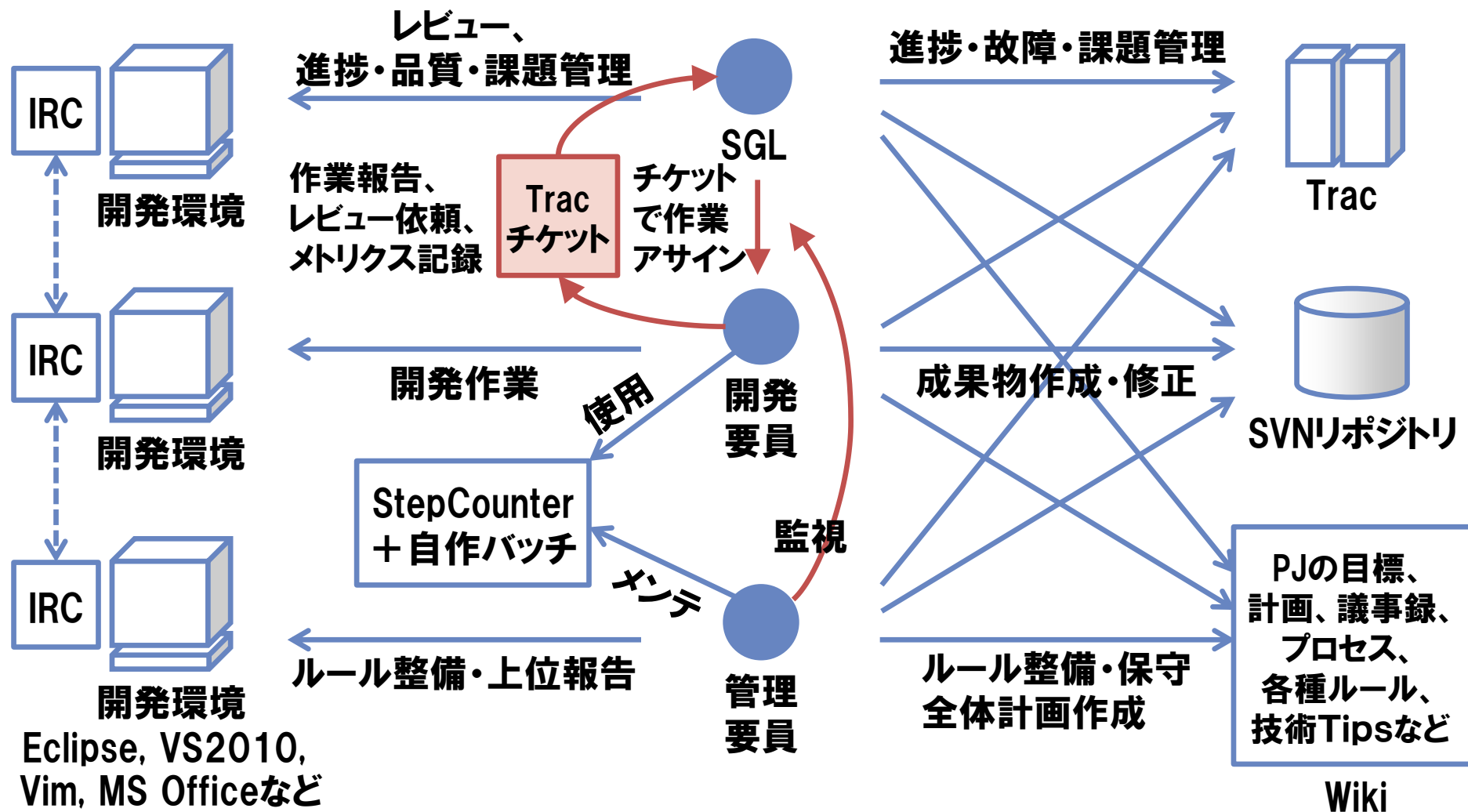
- 10営業日未満の開発単位(案件)で設計、実装、テストを繰り返す反復型開発。
- メトリクス分析や対面報告形式の集合型レビューではなく、全作業内容のピアレビューで品質保証する。



※ED/ID: 内部設計/外部設計、 M/UT: 製造/単体テスト、 IT: 結合テスト、 ST: 総合テスト

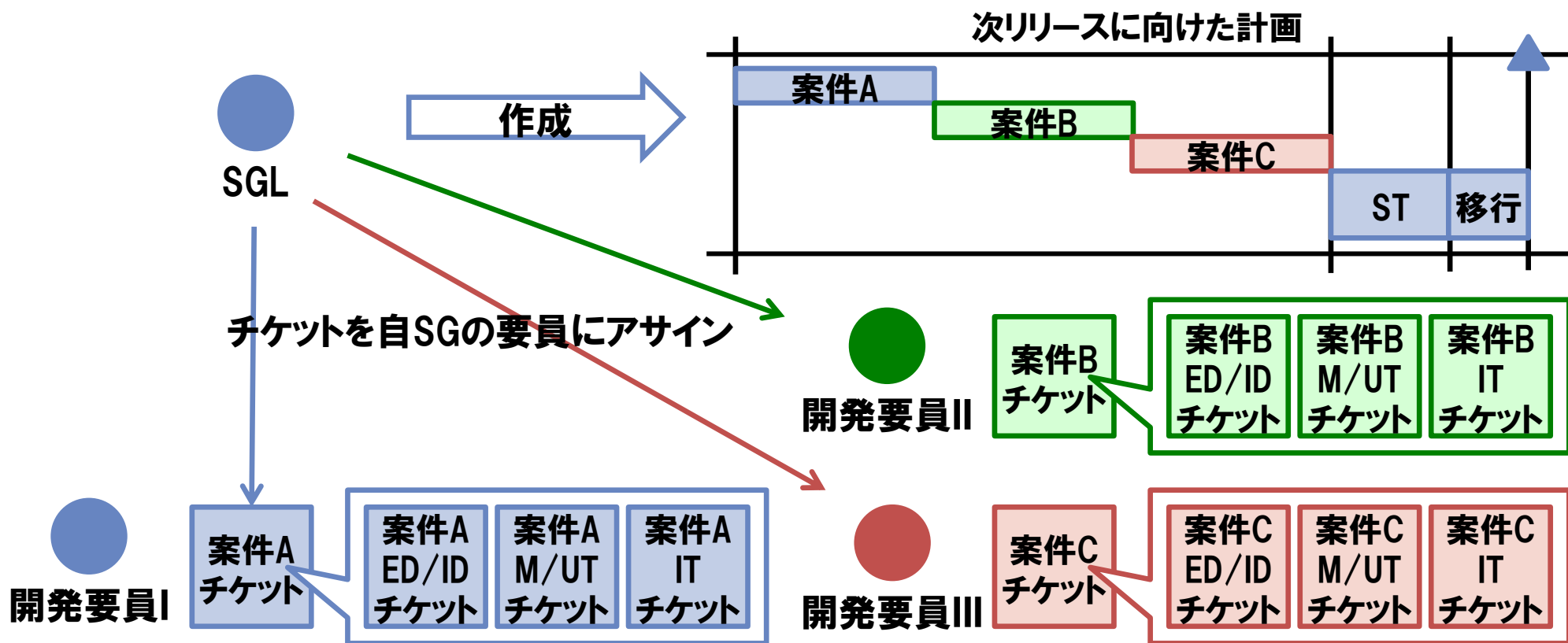
開発環境と管理のためのサーバ群

- SVNを構成管理、Wikiを情報共有に使い、Tracを進捗管理・故障管理・課題管理、メトリクス計測・蓄積・利用に使った。



Tracの運用方法：計画作成時

- SGLが次のリリースに向けた自SGの計画を作成し、各案件に対応する案件チケットを発行する。
- 各案件について、ED/ID、M/UT、ITチケットを発行し、案件チケットと共に各開発要員にアサインする。



案件チケットの例

案件名

2.0案件: データ管理画面

登録: 14ヵ月前 (2012/07/17 15:50:09)
最終更新: 7ヵ月前 (2013/02/06 19:17:29)

作業実施 情報

報告者:	oosugin	担当者:	btnodayu
マイルストーン:	ESTIMANCER(10/1版)→修正済プログラ ム資料	コンポーネント:	0060_source
バージョン:	2.0	キーワード:	
関係者:	fujinukim , oosugin , inoguchin , fushidak , btnodayu	関連チケット:	#425 , #435
開始予定日:	2012/10/16	完了予定日:	2012/10/18
開始日:	2012/12/07	完了日:	2013/02/05
実施状況:	<ul style="list-style-type: none"> ED/ID: #591 M/UT: #618 IT: #654 		

レビュー 情報

レビュー:	oosugin		
レビュー実施rev:	1829	修正後rev:	1829
レビュー実施日:	2013/02/05		
レビュー記録:	<ul style="list-style-type: none"> 2013/2/5, oosugin <ul style="list-style-type: none"> 内部リ判結果を 内部リリース判定チェックリスト_#435.xlsx に記述した。 #651 積み残り前提でOK クローズする。 		

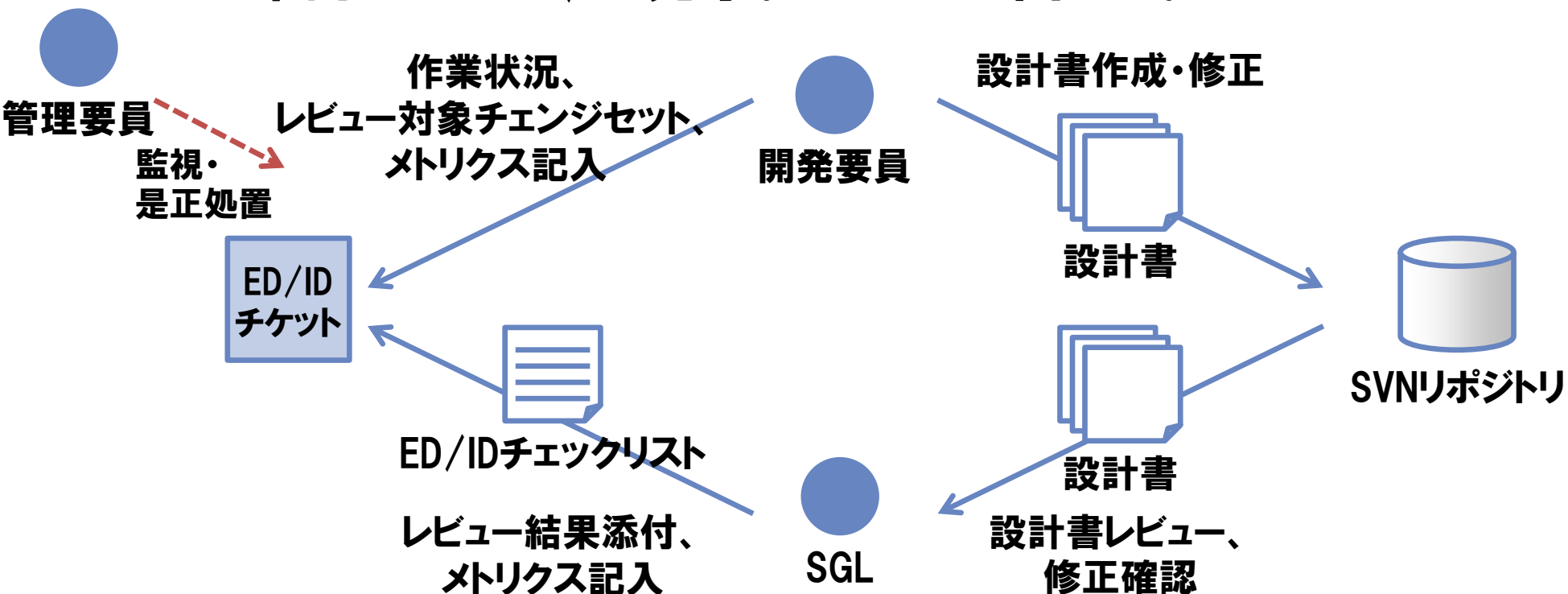
メトリクス

作業工数(人時):	0	レビュー工数(人時):	0.1
指摘件数(件):	1		
案件種別:	新規機能追加		

作業内容 の詳細

Tracの運用方法: ED/ID実施時

- 開発要員は設計書を作成・修正し、レビュー対象チェンジセットをED/IDチケットに記入してリーダに渡す。
- SGLはチェックリストに沿ってレビューを実施し、結果を記入したチェックリストを添付して返す。
- 要員は修正を実施したチェンジセットを記入して渡す。全ての指摘をSGLが確認したらED/ID完了。チケットを閉じる。



ED/IDチケットの例

チケット名

#435_ED/ID

登録: 9ヵ月前 (2012/12/07 10:15:26)

最終更新: 0秒前 (2013/09/03 17:49:50)

作業実施
情報

報告者:	btnodayu	担当者:	btnodayu
マイルストーン:	ESTIMANCER(10/1版)→ID設計書(画面設計書)	コンポーネント:	0070_specification
バージョン:	2.0	キーワード:	
関係者:	fujinukim, oosugin, inoguchin, fushidak, btnodayu	関連チケット:	#425, #429, #435, #437, #438, #439, #461, #440, #441, #442, #443, #446
開始予定日:	2012/10/16	完了予定日:	2012/11/18
開始日:	2012/12/07	完了日:	2012/12/25
実施状況:			

- 2012/12/20, btnodayu
- [1684] でデータ管理画面の設計書を作成した。

レビュー:	oosugin		
レビュー実施rev:	1684	修正後rev:	1694, 1695
レビュー実施日:	2012/12/25		
レビュー記録:			

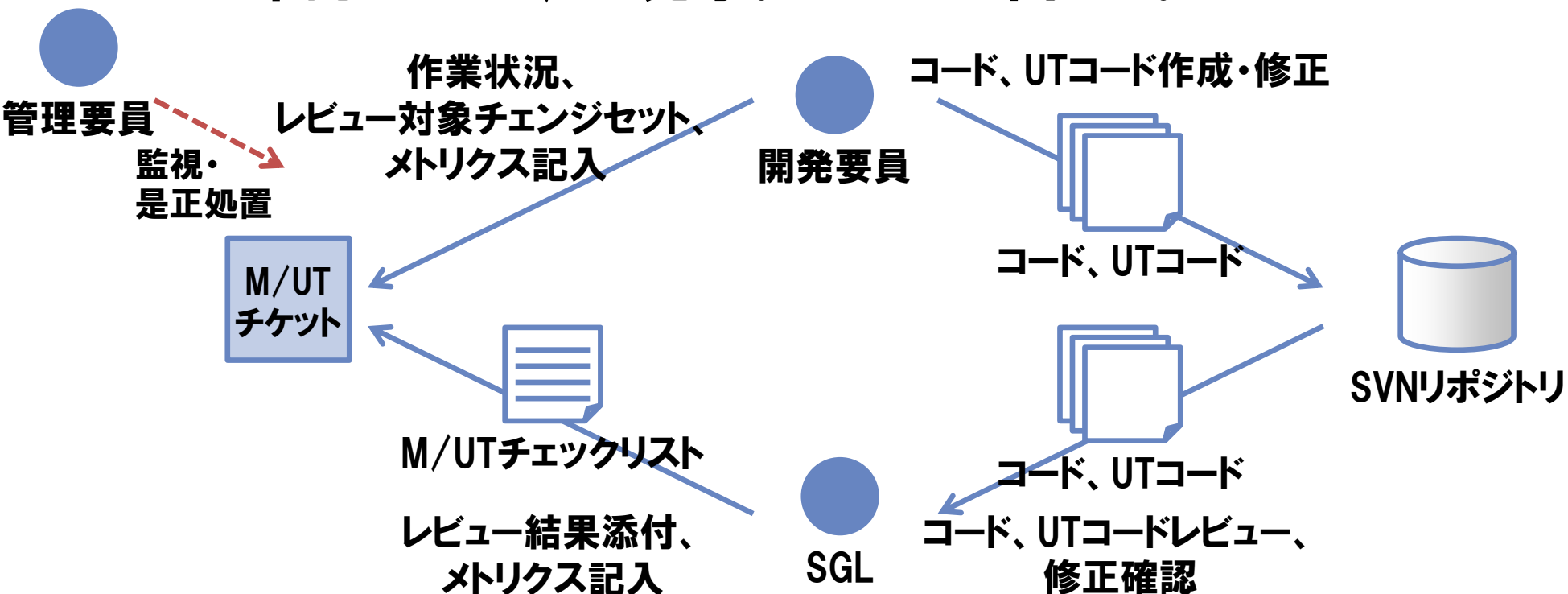
- 2012/12/24, oosugin
- ⇒EDIDチェックリスト_#591.xlsx に記述した。
- 下記前提でOK
 - M/UT で MetaBase を修正する。
 - [1694,1695] の修正を確認し、必要に応じて M/UT で確認する。

作業工数(人時):	8	レビュー工数(人時):	22.5
レビュー頁数(頁):	5	指摘件数(件):	10

作業内容

Tracの運用方法: M/UT実施時

- 開発要員はコードとUTコードを作成・修正し、レビュー対象チェンジセットをM/UTチケットに記入してリーダに渡す。
- SGLはチェックリストに沿ってレビューを実施し、結果を記入したチェックリストを添付して返す。
- 要員は修正を実施したチェンジセットを記入して渡す。全ての指摘をSGLが確認したらM/UT完了。チケットを閉じる。



M/UTチケットの例

チケット名

#435_M/UT

登録: 9ヶ月前 (2012/12/20 20:23:25)

最終更新: 0秒前 (2013/09/03 17:58:53)

作業実施 情報

報告者:	btnodayu	担当者:	btnodayu
マイルストーン:	ESTIMANCER<10/1版>→修正済プログラ ム資料	コンポーネント:	0060_source
バージョン:		キーワード:	
関係者:	fujinukim, oosugin, inoguchin, fushidak, btnodayu	関連チケット:	#425, #429, #435, #437, #438, #439, #461, #440, #441, #442, #443, #446, #591
開始予定日:	2012/10/16	完了予定日:	2012/11/18
開始日:	2012/12/21	完了日:	2013/02/18
実施状況:	<ul style="list-style-type: none"> ➡ [1801,1802] でコードを修正した。 ➡ [1803,1804,1806] で Simple ライブラリをベンダブランチに格納し、マージした。 ● UTはエクスポート関連は確認、インポートはrequestでファイルが渡せなかったため未コミット、単体、関連のチェックは未作成、カバレッジは未確認 		

レビュー 情報

レビュー:	inoguchin		
レビュー実施rev:	1801, 1802, 1803, 1804, 1806	修正後rev:	1825
レビュー実施日:	2013/02/04		
レビュー記録:	<ul style="list-style-type: none"> ➡ 2013/02/04, inoguchin ● [1811,1813,1814,1815,1825] のとおりコードを修正した。 		

メトリクス

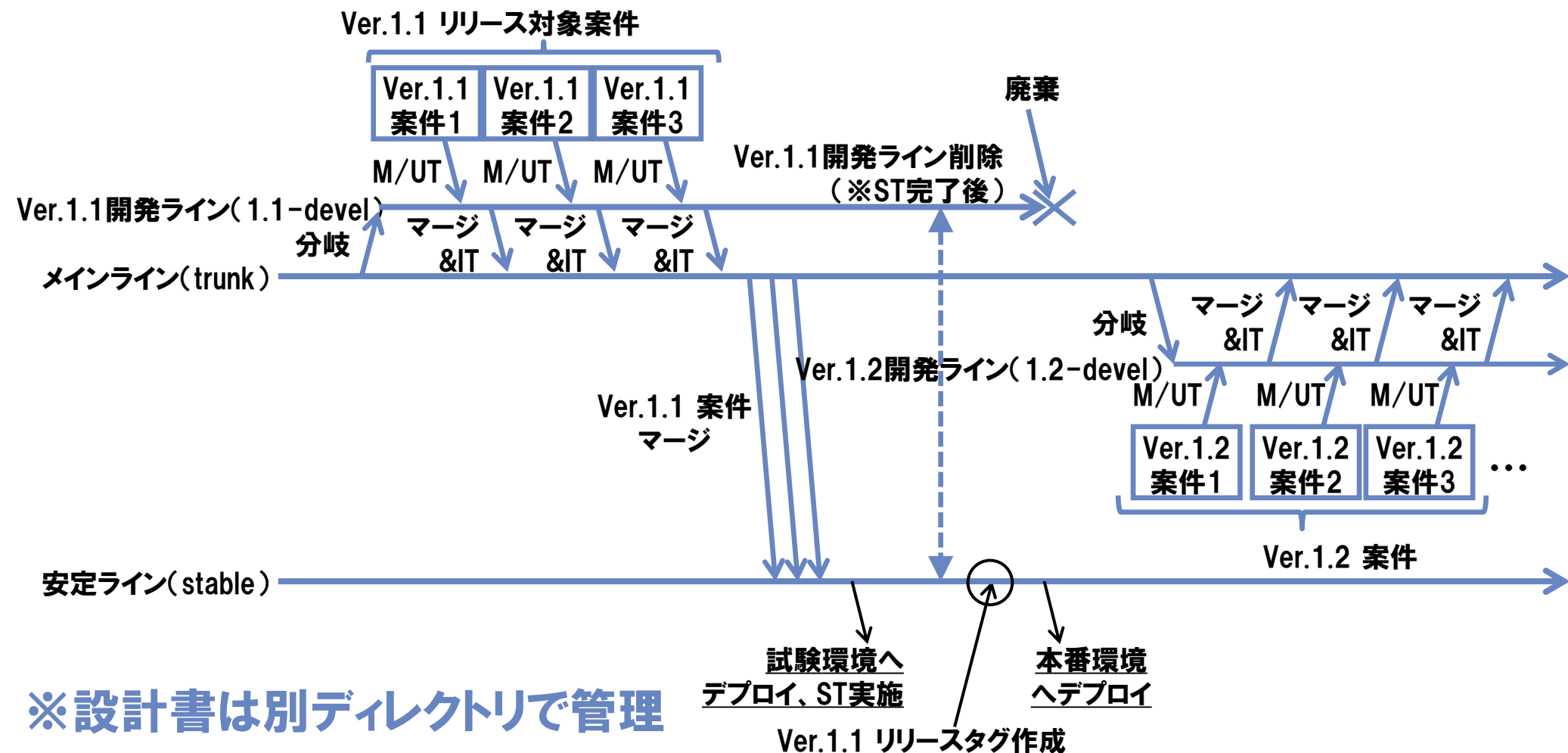
作業工数(人時):	36	レビュー工数(人時):	12
製造規模(Step):	2012	テスト規模(Step):	780

作業内容 の詳細

詳細

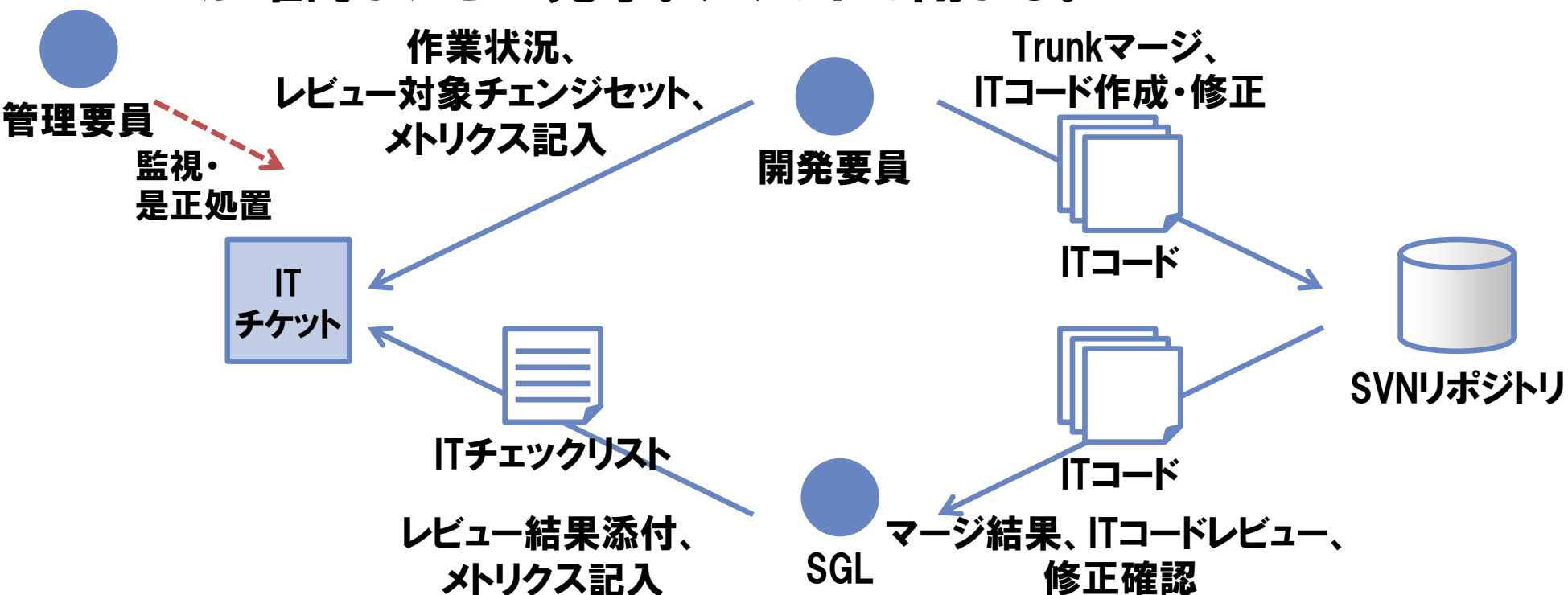
ソースコードのライブラリ管理

- M/UTは開発ライン、ITはメインライン、STは安定ラインの資材で実施。
- 案件毎にコミットしメインラインへマージ、リリース毎に安定ラインへマージ。
- 衝突は複数人が同一資材を同時編集しないようスケジューリングして回避。



Tracの運用方法: IT実施時

- 開発要員はコードとUTコードをメインラインへマージし、ITコードを作成・修正。チェンジセットをITチケットに記入してリーダに渡す。
- SGLはチェックリストに沿ってレビューを実施し、結果を記入したチェックリストを添付して返す。
- 要員は修正を実施したチェンジセットを記入して渡す。全ての指摘をSGLが確認したらIT完了。チケットを閉じる。



ITチケットの例

チケット名

#435_IT

登録: 7ヵ月前 (2013/02/04 17:25:40)

最終更新: 0秒前 (2013/09/03 18:29:10)

作業実施
情報

報告者:	inoguchin	担当者:	btnodayu
マイルストーン:	ESTIMANGER(10/1版) → 修正済プログラ ム資料	コンポーネント:	0060_source
バージョン:	2.0	キーワード:	
関係者:		関連チケット:	#435
開始予定日:	2013/02/04	完了予定日:	2013/02/05
開始日:	2013/02/04	完了日:	2013/02/04
実施状況:			

- 2013/2/4, inoguchin
- [1827] で #618 #435_M/UTの修正をメインラインにマージした。
 - [1829] で IT コードを修正・実行する
 - IT結果を確認、必要に応じて修正した。
 - ITコードに仕様未反映 → trunkでITコードを修正
 - M/UTからのすり抜け → develでM/UT再実施

レビュー
情報

レビュー:	oosugin		
レビュー実施rev:	1829	修正後rev:	1829
レビュー実施日:	2013/02/04		
レビュー記録:			

メトリクス

作業工数(人時):	8	レビュー工数(人時):	3
テスト規模(Step):	126		
バグ数(件):	0	すり抜けUTバグ数(件):	0

作業内容
の詳細

詳細

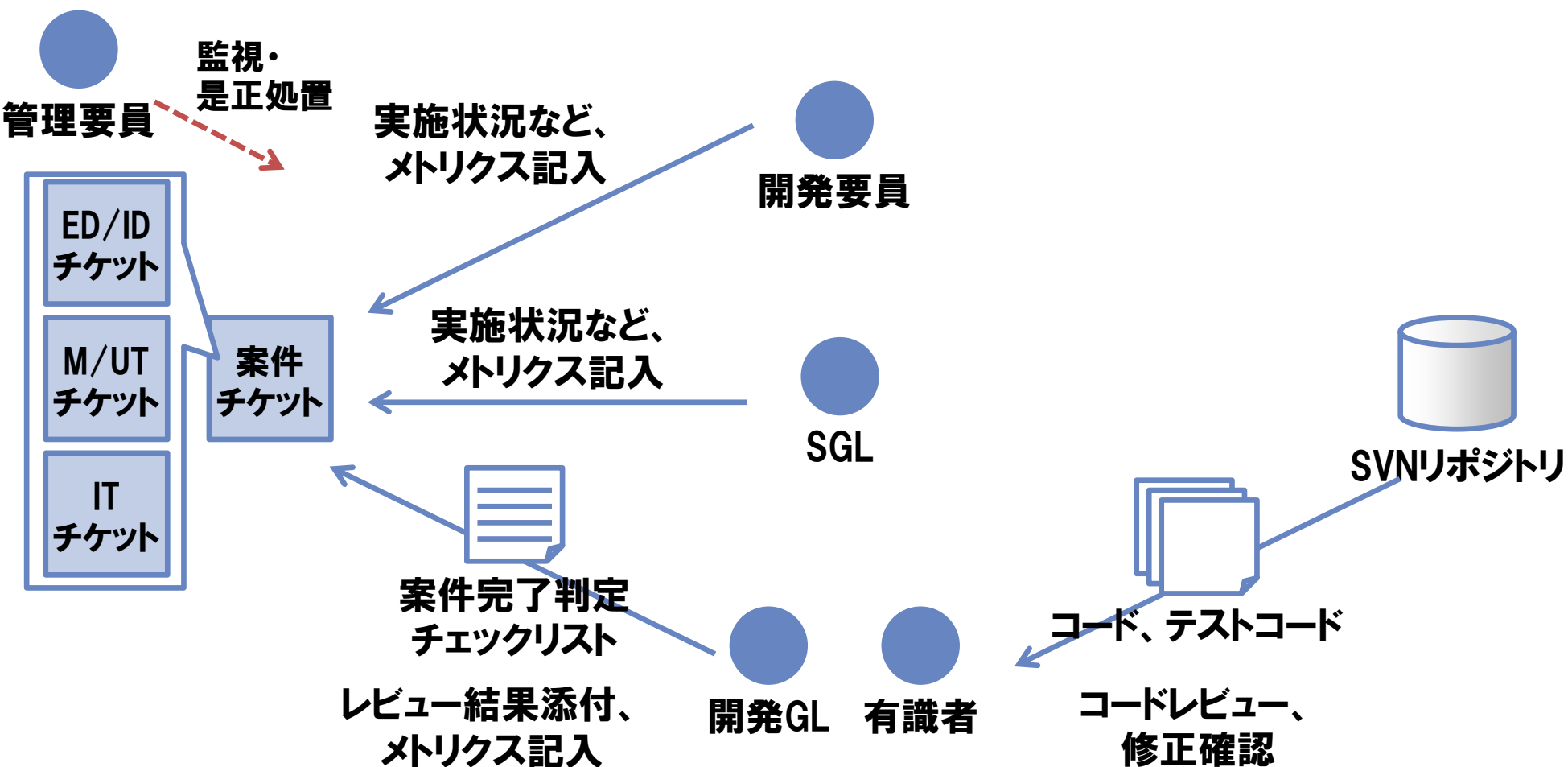
作業内容とレビュー対象・観点

■ 各自席でチェックリストに沿ってレビューを実施。全観点OKになれば通過。

工程	作業内容	レビュー対象	レビュー観点
設計 (ED/ID)	<ul style="list-style-type: none"> 仕様検討 設計書作成修正 	<ul style="list-style-type: none"> 設計書のチェンジセット 	<ul style="list-style-type: none"> 要件の充足 影響範囲の考慮
製造 (M)	<ul style="list-style-type: none"> ソースコード作成・修正 	<ul style="list-style-type: none"> ソースコードのチェンジセット 	<ul style="list-style-type: none"> 設計との整合性 アーキテクチャや規約 性能や信頼性の考慮
単体テスト (UT)	<ul style="list-style-type: none"> リクエストレスポンスでのブラックボックステスト C1網羅のためのホワイトボックステスト 	<ul style="list-style-type: none"> テスト項目表、証跡 UTコードのチェンジセット UTコード走行結果 	<ul style="list-style-type: none"> 修正個所に対するC1網羅 テスト観点の充足 無駄なテストコードがないか 影響範囲の回帰テスト
結合テスト (IT)	<ul style="list-style-type: none"> 画面遷移確認 画面間データ連動確認 	<ul style="list-style-type: none"> テスト項目表・証跡 ITコードのチェンジセット ITコード走行結果 	<ul style="list-style-type: none"> 設計書の修正個所網羅 テスト観点の充足 無駄なテストコードがないか
案件完了 判定	<ul style="list-style-type: none"> 全作業結果確認 (主要部分のみ) 	<ul style="list-style-type: none"> 全対象物 (主要部分のみ) 	<ul style="list-style-type: none"> ITまでの全観点 (主要部分のみに絞って確認)
総合テスト (ST)	<ul style="list-style-type: none"> 業務、移行、運用、環境、外部接続 マニュアルの連動 非機能要件の確認 	<ul style="list-style-type: none"> システム全体 	<ul style="list-style-type: none"> 業務、移行、運用、環境、外部接続が正しく連動するか マニュアルと動作の整合性 非機能要件の充足

Tracの運用方法：案件完了後

- 開発要員は実施状況などを書き込み、案件チケットをSGLに渡す。
- SGLは有識者、および開発GLに案件完了判定を依頼する。
- 案件完了が承認されれば、案件チケットをクローズする。



収集した基本メトリクス(1)

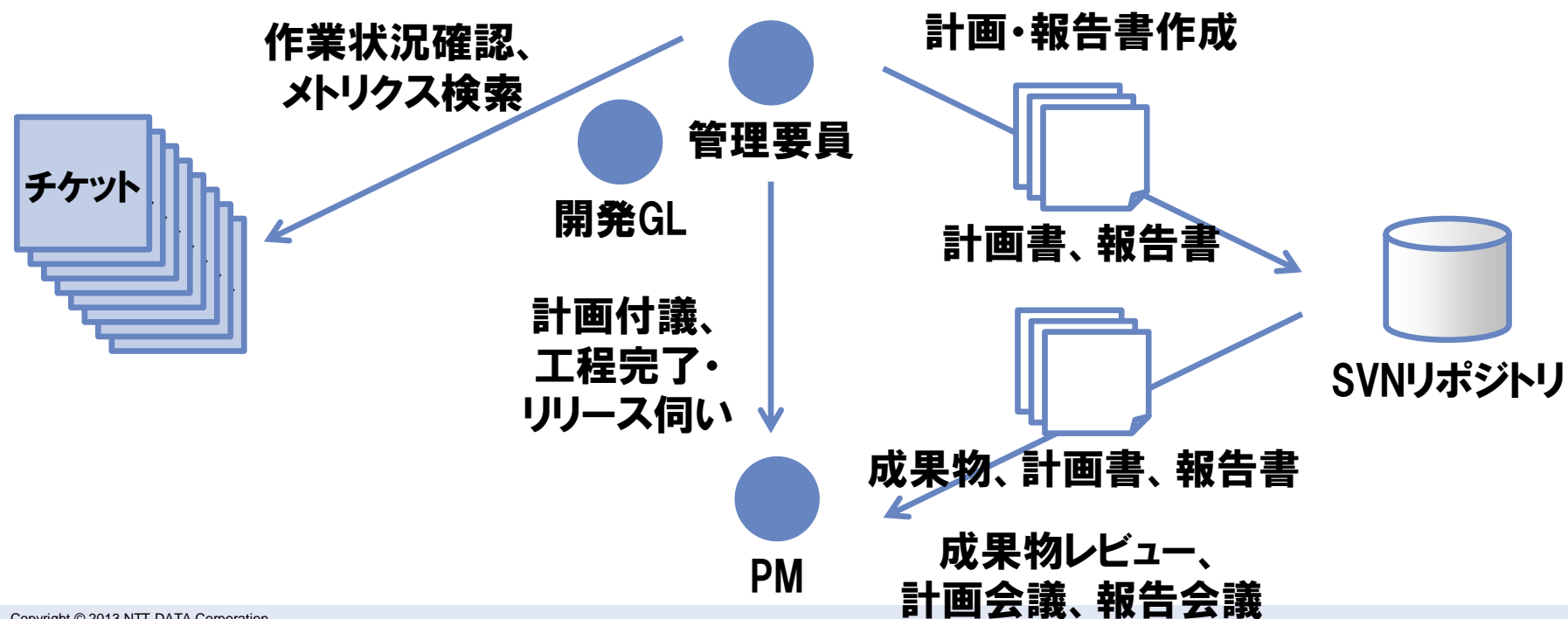
メトリクス名	チケット	記入担当者	記入内容
作業工数 (人時)	全チケット	開発要員	作業、レビュー指摘事項への対応工数。
レビュー工数 (人時)	全チケット	SGL	作業のレビュー、レビュー指摘修正確認の工数。
レビュー頁数 (頁)	ED/ID	SGL	レビューした設計書頁数。レビュー対象がMS Excel形式の場合、レビューしたシート数。
指摘件数 (件)	ED/ID	SGL	レビュー指摘件数。レビュー結果はMS Excel形式のレビュー結果票で1行に指摘1件を記入するため、レビュー結果票の行数(=指摘数)。
開発規模 (Step)	M/UT	開発要員	M/UT開始前～完了までの追加・変更行数の合計、削除行数は数えない。 リビジョン間差分のLOCを計測して記入。
テスト項目数(件) または テスト規模(Step)	M/UT, IT, ST	開発要員	テストの量。Aシステム開発ではテスト項目数、Bシステム開発ではテストコード(UTはJUnit、ITとSTはSelenium)の追加変更行数、削除行は数えない。

収集した基本メトリクス(2)

メトリクス名	チケット	記入担当者	記入内容
バグ数 (件)	IT, ST	開発要員	ITやSTで検出されたバグの数。 UTバグは取り切ってからM/UT完了する想定のため計測しない。 コーディングとUTコード記述を並行実施するため、計測にかかるコストも大きすぎる。
すり抜けバグ数 (件)	IT, ST	開発要員	ITやSTで検出されたものの、完了済の前工程の観点で検出すべきだったバグの数。 開発要員が判断に迷った場合は、レビュー(SGL)と管理業務担当者が判断する。
開発期間 (日)	案件	開発要員	案件開始～完了までの日数。開始日と終了日から算出。
案件完了判定 指摘件数 (件)	案件	SGL	IT完了後に実施する案件完了判定での指摘件数を記入。

Tracの運用方法：メトリクス利用時

- 管理要員はTracのカスタムクエリによって蓄積したメトリクスを抽出し、MS Excelファイルで集計する。
- 各案件の実施状況やメトリクスに基づき、計画書や報告書を作成し、PMに計画付議、または工程完了やリリースを伺う会議を実施する。
- 外部からの依頼などに応じて、適当なタイミングで集計・報告もする。



カスタムクエリによるチケット検索結果の例

カスタムクエリ (94 件が該当)

▼ フィルタ

ステータス ☒ accepted ☒ assigned ☒ closed ☒ new ☒ reopened ☒ verified

分類

And Or

▶ カラム

結果のグループ化: ☐ 降順

各検索結果の下に表示: ☐ 詳細 ☐ 実施状況 ☐ レビュー記録

1ページに表示する件数

更新

<input type="checkbox"/>	チケット	概要	分類	作業工数(人時)	レビュー工数(人時)	テスト規模(Step)	バグ数(件)	すり抜けUTバグ
<input type="checkbox"/>	#262	#217_IT	IT	4.5	2	78	0	0
<input type="checkbox"/>	#271	#226_IT	IT	6	2.5	38	0	0
<input type="checkbox"/>	#277	#218_IT	IT	2.5	1	54	0	0
<input type="checkbox"/>	#286	#266_IT	IT	1	1	10	0	0
<input type="checkbox"/>	#335	#315_IT	IT	1	0.5	1	1	1
<input type="checkbox"/>	#330	#301_IT	IT	1	0.5	0	0	0
<input type="checkbox"/>	#349	#317_IT	IT	1	0.5	0	0	0
<input type="checkbox"/>	#420	#397_IT	IT	0.5	0.5	0	0	0
<input type="checkbox"/>	#451	#401_IT	IT	1.1	1	0	0	0
<input type="checkbox"/>	#469	#416_IT	IT	0.5	0.5	1	0	0

報告書におけるメトリクス集計結果の例

項番	メトリクス	今リリース					これまでの累積				
		標本数	合計	80%ile	中央値	20%ile	標本数	合計	80%ile	中央値	20%ile
1	工期(日)	31	811	53.00	18.00	3.00	39	842	51.40	13.00	1.80
2	総工数(人時)	32	1100.1	121.11	31.80	8.88	40	1113.7	118.77	28.11	8.88
3	生産能率(KS/ 人月)	25	—	234.11	71.70	8.71	31	—	231.09	71.69	12.78
4	レビュー工数 (人時)	18	108.18	11.17	4.00	0.88	22	116.88	11.89	4.00	0.88
5	レビュー対象頁 数(頁)	18	137	13.00	7.00	1.70	22	171	13.00	7.00	1.10
6	指摘件数(件)	18	188	21.90	10.00	1.70	22	196	20.40	7.00	1.00
7	レビュー密度 (分/頁)	18	—	884.00	27.14	11.84	22	—	881.36	27.04	7.88
8	エラー摘出密 度(件/100頁)	18	—	256.86	156.07	42.50	22	—	330.57	137.50	15.36
9	製造規模(Ste p)	25	10441	2528.40	516.00	18.48	31	11506	2704.00	414.00	28.88
10	テスト規模(人月)	25	1077.17	1077.17	1077.17	0.77	31	1333.23	1333.23	1333.23	0.77

利用した導出メトリクス

メトリクス名	評価対象	導出方法	主な用途
生産性 (KStep/人月)	案件	開発規模 ÷ 案件に要した全工数	計画時の見積り、スコープ調整や再計画で見積り根拠として利用。
1日当たり稼働時間	案件	案件に要した全工数 ÷ 開発期間	要員毎の負荷調査、稼働時間が低い要員や案件の検出と問題改善。
レビュー密度 (分/頁)	ED/ID	ED/IDレビュー工数 ÷ レビュー頁数	レビュー不十分な可能性がある案件の検出。仕様が複雑など、レビューに時間がかかる要注意案件の抽出。
エラー密度 (件/百頁)	ED/ID	ED/ID指摘件数 ÷ レビュー頁数	レビュー不十分でエラーが少なすぎるまたは、多すぎる案件の検出。
テスト密度 (件/KStep) または テストコード比率 (%)	UT、 IT、 ST	テスト項目数 ÷ 開発規模 または テスト規模 ÷ 開発規模	テスト不十分な可能性がある案件の検出。仕様が複雑など、テスト量が非常に多い要注意案件の検出。
バグ密度(件/KStep)	IT、 ST	バグ数 ÷ 開発規模	テスト不足でバグ抽出が過小、またはバグが多すぎる問題案件の検出。
すり抜けバグ密度 (件/KStep)	IT、 ST	すり抜けバグ数 ÷ 開発規模	前工程のテスト不備とレビューチェック漏れ検出。再発防止に繋げる。

集計したメトリクスの用途

■ 各バージョンの開発計画策定

- 生産性(KStep/人月)で見積り。完了済案件の生産性と、開発予定案件の見積り規模を積算して各案件の見積り工数を算出した。
- 大規模機能追加や仕様が複雑な部分の修正は、複数案件に分割して見積り。

■ スcope調整・再計画における再見積り

- その時点での完了済案件の生産性や、1日当たり稼働時間で再見積り。
- 他業務と兼務の要員など、実質上の稼働が1日3時間程度のケースもあった。

■ 完了案件に対する週次定量分析

- 過去案件のデータと相互比較してメトリクスが大きく乖離した案件は、案件完了判定とは別にGLが問題の有無を再度ピアレビューで確認。

■ 工程完了やリリースを伺う上位マネジメント層への報告

- メトリクスが他大きく乖離した案件について、管理担当者が成果物をレビューしたり、各要員にヒアリングしたりして、成果物やプロセス上の問題を再確認。

考察：利用したメトリクスの有効性

■ 規模、工数、LOC、開発期間：有効

- 計測と解釈が容易で、収集や要員の学習コストも小さい。
- 他のメトリクスと組み合わせて分析できる。
 - ◆ 1日当たり開発量 ⇒ 見積りに利用
 - ◆ 案件やモジュール毎の生産性 ⇒ 相対値が悪いものの抽出に利用

■ バグ密度：課題あり

- IT、STバグ数を計測したが、多くの案件ではIT、STバグは検出されなかった。
- バグが検出された否かのみに着目すればよく、密度を算出した意義は薄い。
 - ◆ 機能やモジュールを単位とし、期間を決めてバグ数や密度を算出 ⇒ 当該部分の品質や、当該部分担当者・担当チームのスキル優劣の判断材料に利用できる？

■ テスト量やテスト密度：大きい体制ではある程度有効

- GLが全案件ピアレビューできないため、危険案件のフィルタリングに使える。
- GLが全案件ピアレビューする小さい体制では、詳細把握しているため意義薄。
 - ◆ C0/C1網羅率、コードクローン分析など、ピアレビューでは検出できない課題を検出できるメトリクスが有効？



NTT data

変える力を、ともに生み出す。