

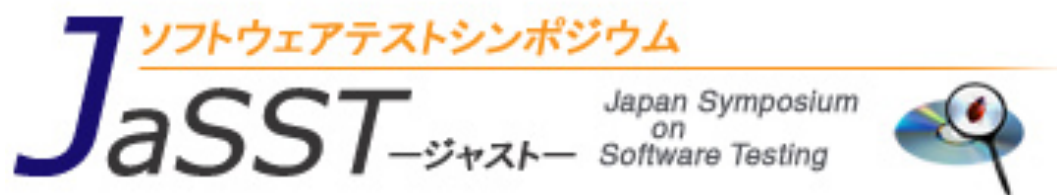
システムテストの自動化による 大規模分散検索プラットフォームの 開発工程改善

Mar/07/2014

荻野 恒太郎, パランデ アビジート, 松本 幹, 鶴飼 大志

Search Platform Group, Search Section, Big Data Department, Rakuten Inc.

<http://www.rakuten.co.jp/>



楽[®]天 [®]Rakuten



本日お持ち帰りして頂きたい事

本日お持ち帰りして頂きたい事

本発表の趣旨

自動化とCIやTDDを組み合わせ
“継続的システムテスト”を実現し
開発のアジリティを改善した

- 誰がためのシステムテスト自動化？
- アジャイルな世界でのテスターの役割は？

バックグラウンド

- ウェブサービス開発の特徴
- システムテスト自動化の目的

ウェブサービス

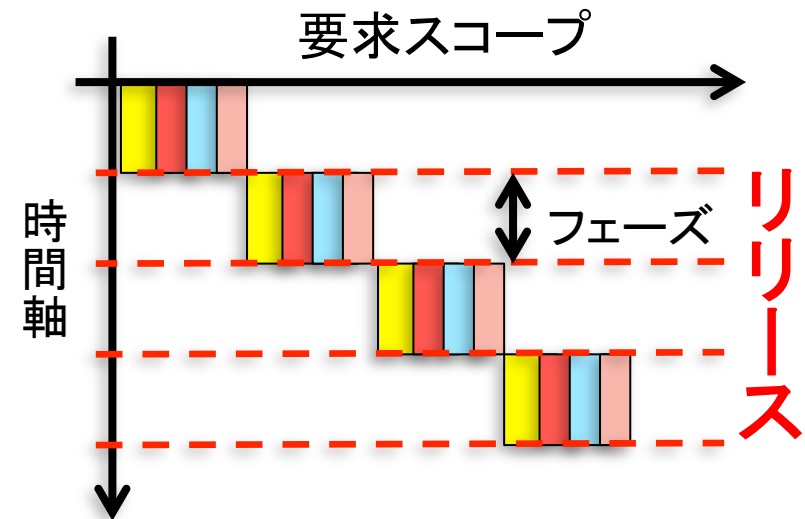
ウェブサービス開発の特徴

• 永続的開発

“Development and Deploy in Facebook”, Facebook, 2013,

日々サービスを成長
させるため

- 市場要求の変化
- 技術トレンドの変化

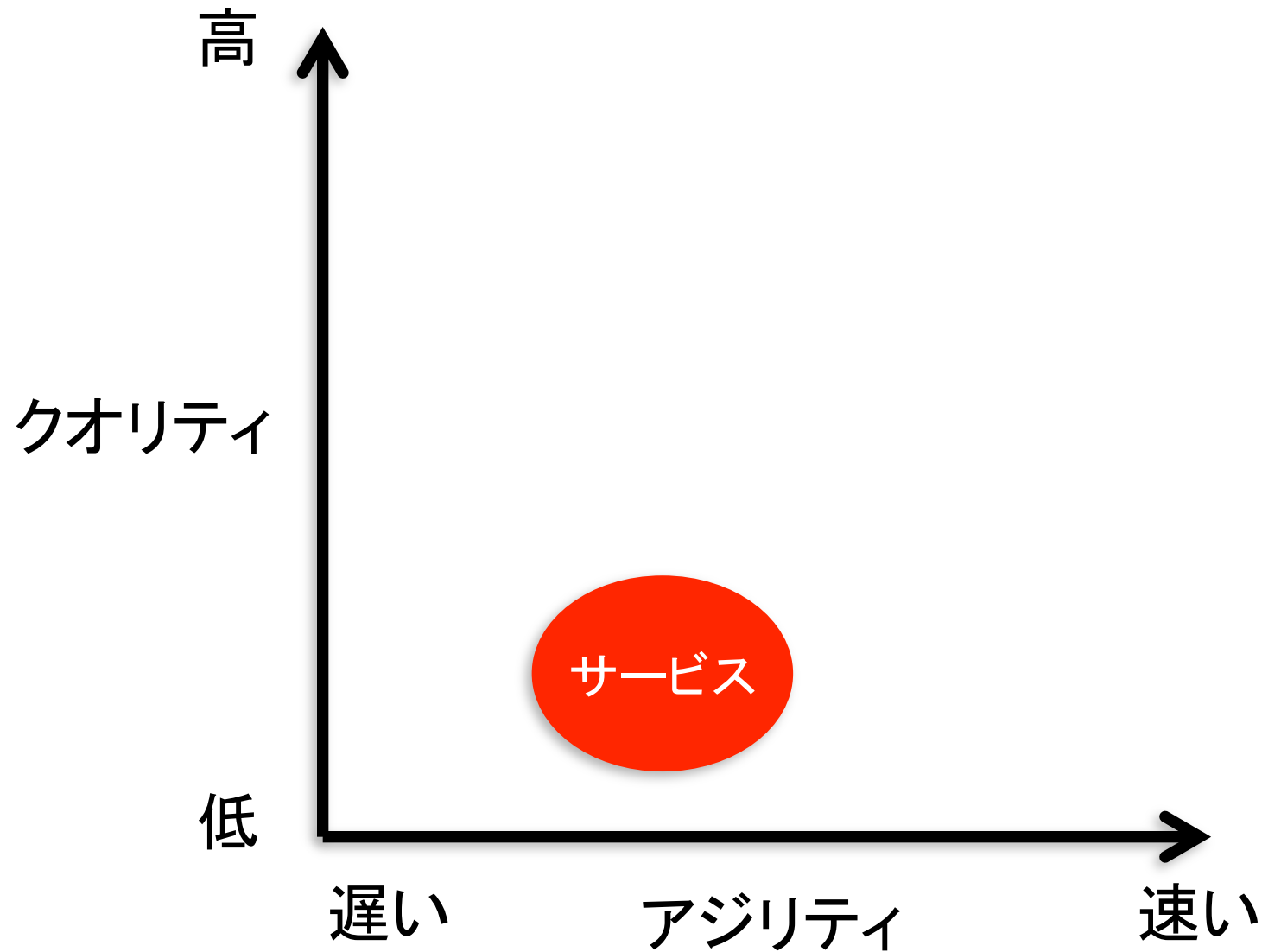


継続的なサービス改善
→ 絶え間ないリリース

バックグラウンド

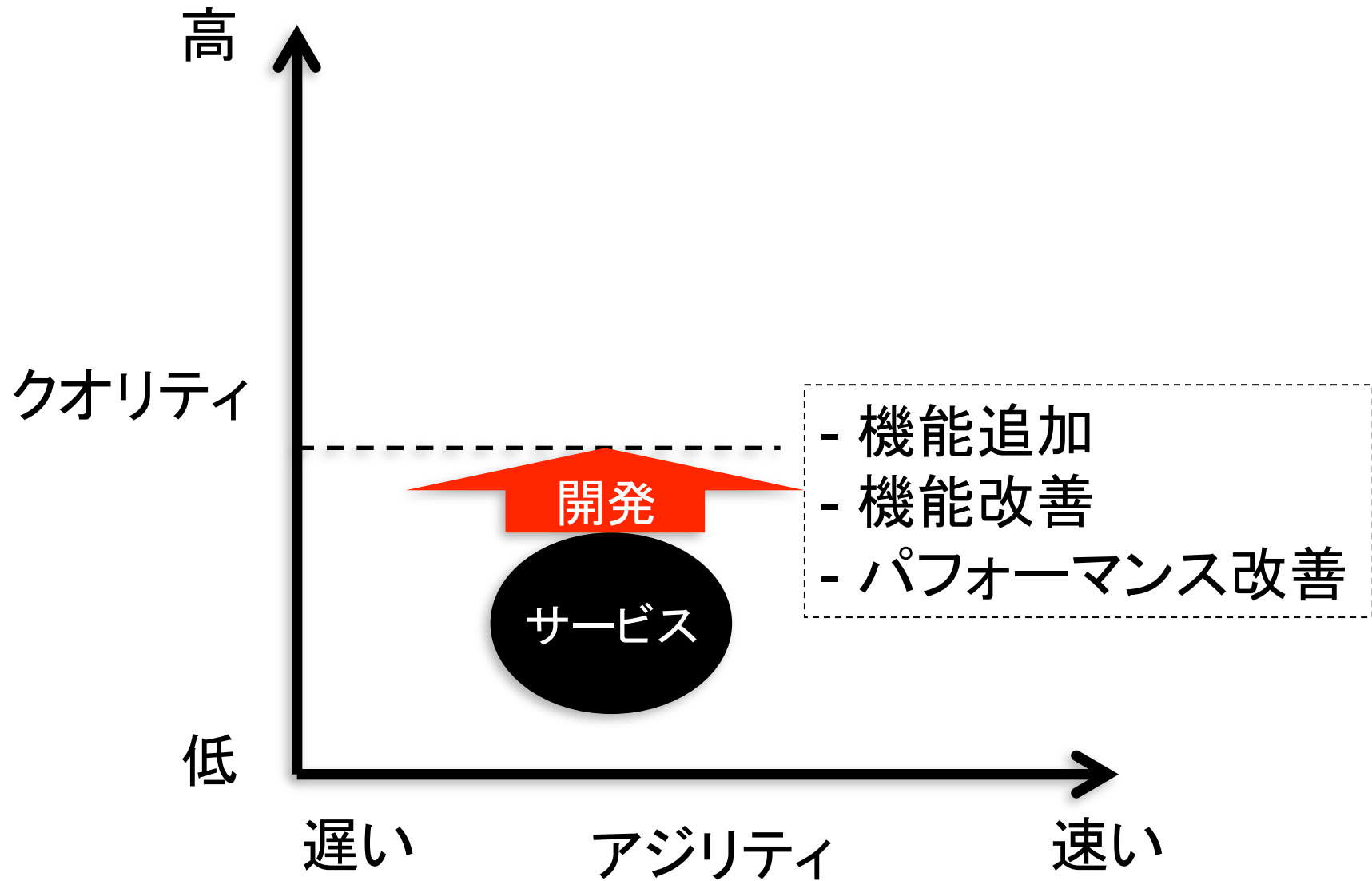
- ウェブサービス開発の特徴
- システムテスト自動化の目的

クオリティ vs アジリティ

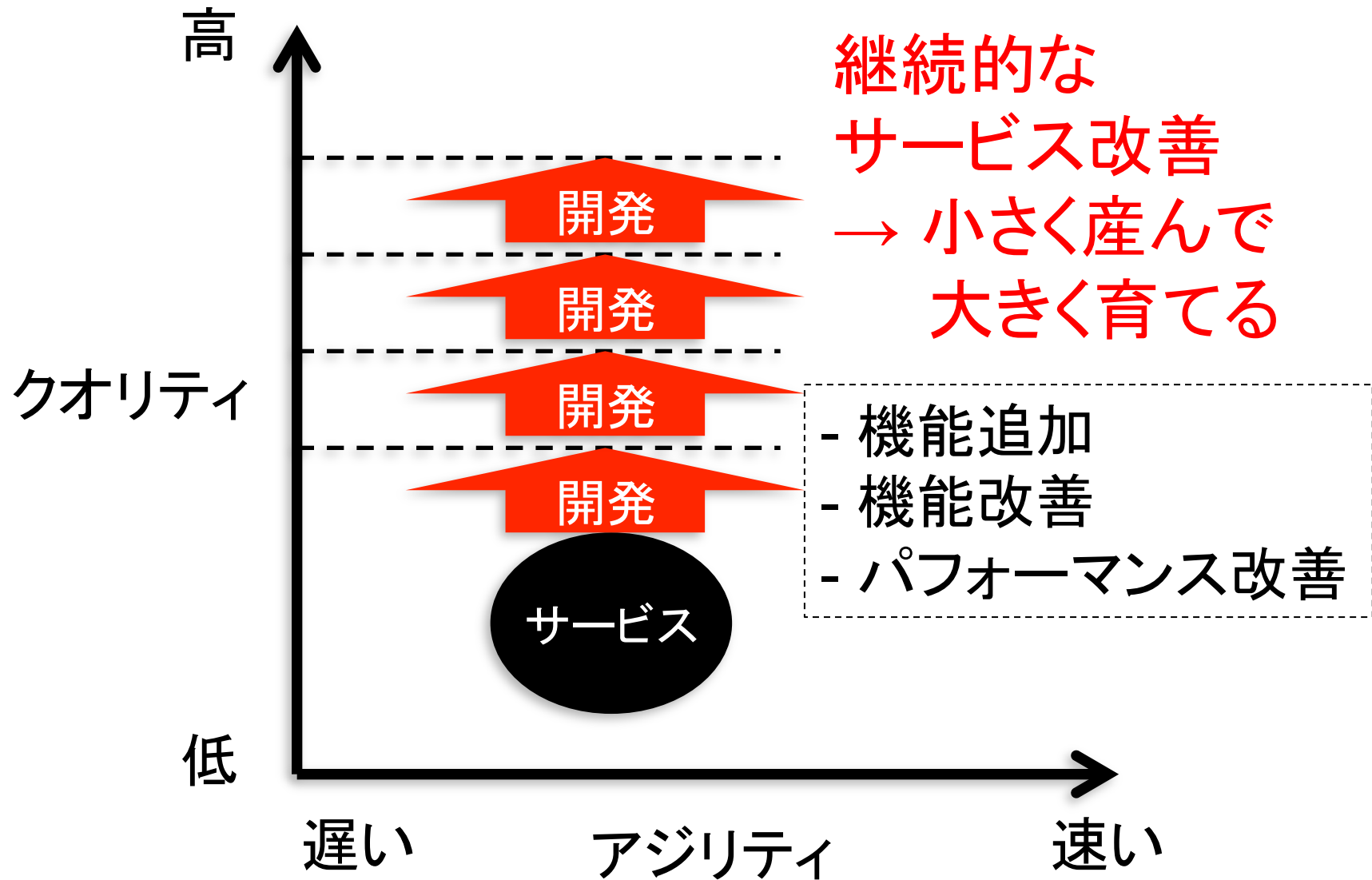


機能開発のリードタイム
バグ修正日数 など

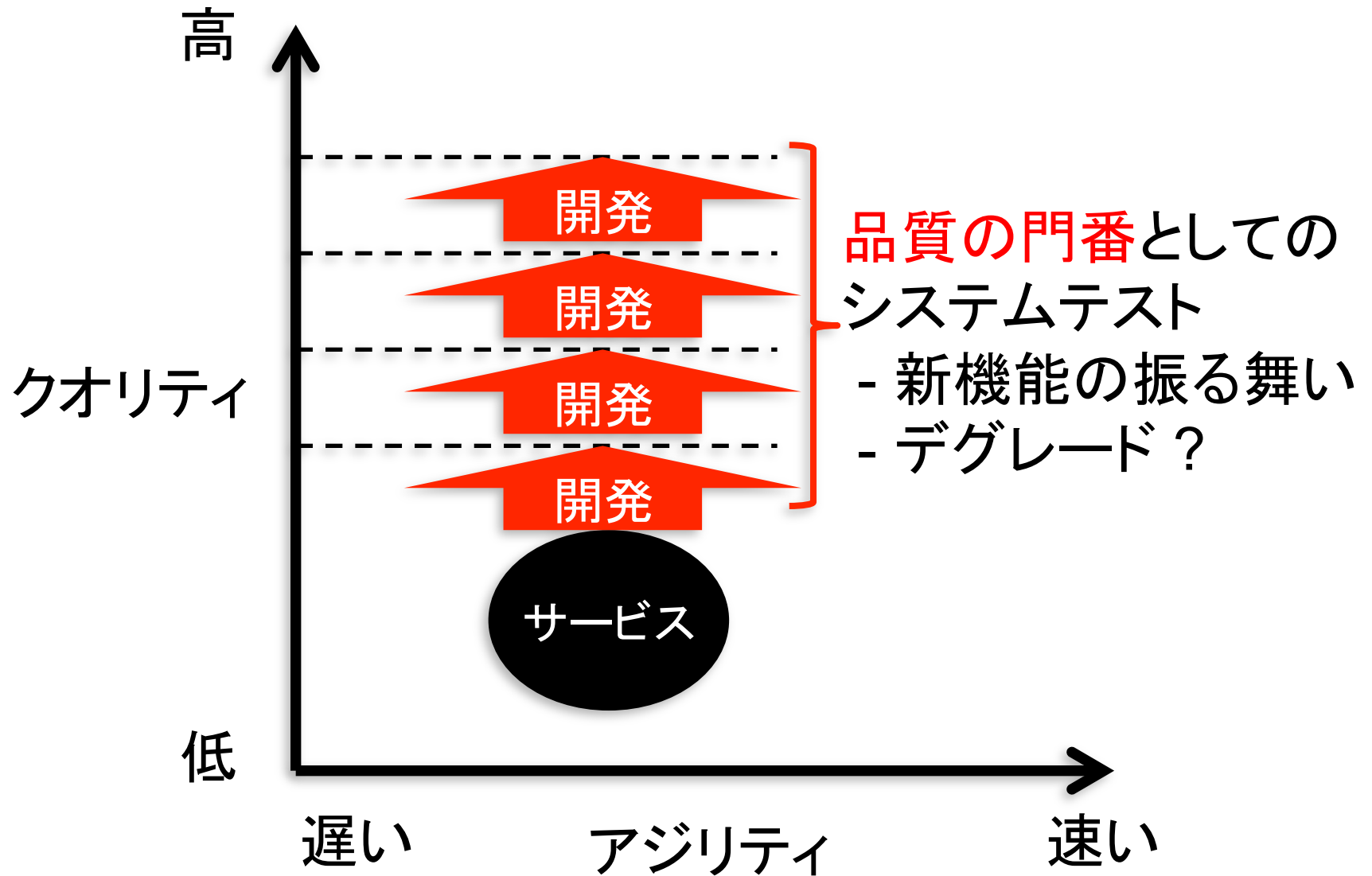
クオリティ vs アジリティ



クオリティ vs アジリティ



クオリティ vs アジリティ



機能開発のリードタイム
バグ修正日数 など

品質の門番:システムテスト

・アジャイルテスト4象限

ISBN:9784798119977“ 実践アジャイルテスト” p.96

ビジネス面

チームを支援する	<u>機能テスト</u> - 検索機能など - 検索ランキング * 以前結合テストと呼んでいた	探索的テスト (ET) 受け入れテスト (UAT)	製品を批評する
	単体テスト (UT) コンポーネントテスト (CT)	パフォーマンス /負荷テスト <u>「～性」テスト</u> - <u>運用性</u> - <u>可用性</u>	

技術面

品質の門番:システムテスト

・アジャイルテスト4象限

ISBN:9784798119977“ 実践アジャイルテスト” p.96

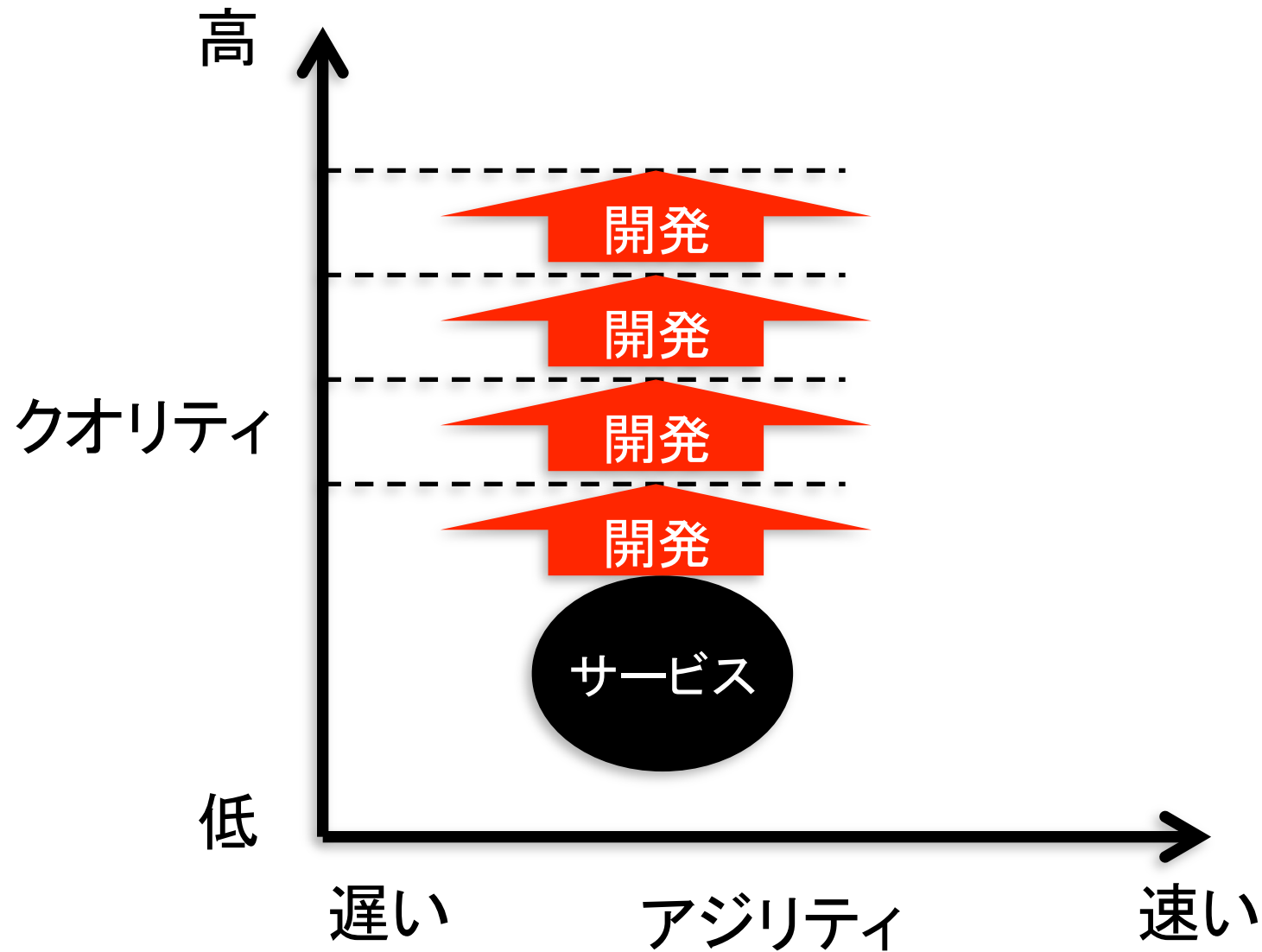
ビジネス面

チームを支援する	機能テスト * 以前結合テストと呼んでいた - 検索機能など - 検索ランキング	探索的テスト (ET) 受け入れテスト (UAT)	製品を批評する
	単体テスト (UT) コンポーネントテスト (CT)	パフォーマンス / 負荷テスト 「～性」テスト - 運用性 - 可用性	

技術面

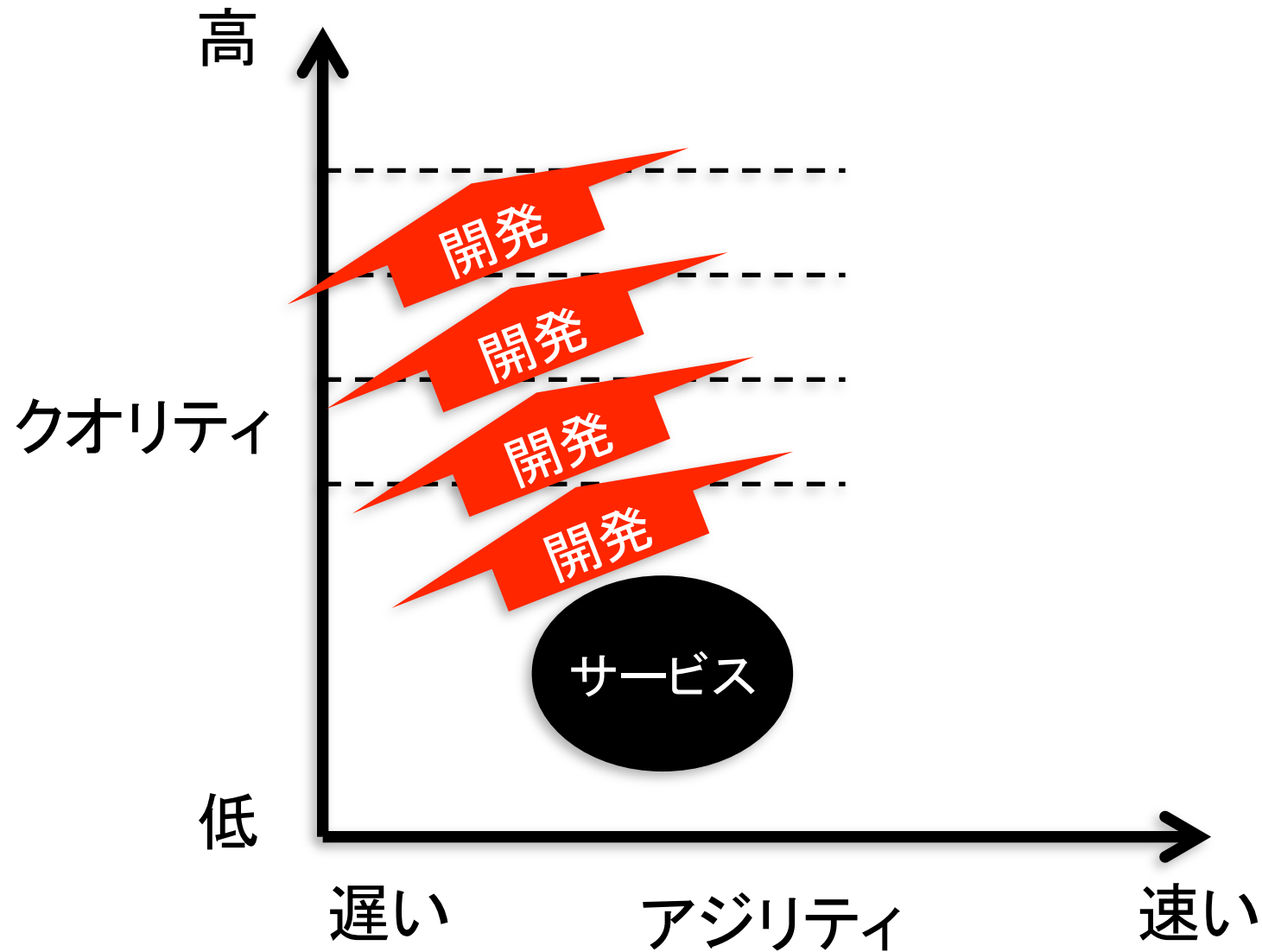
システムテスト(ST) その他

クオリティ vs アジリティ



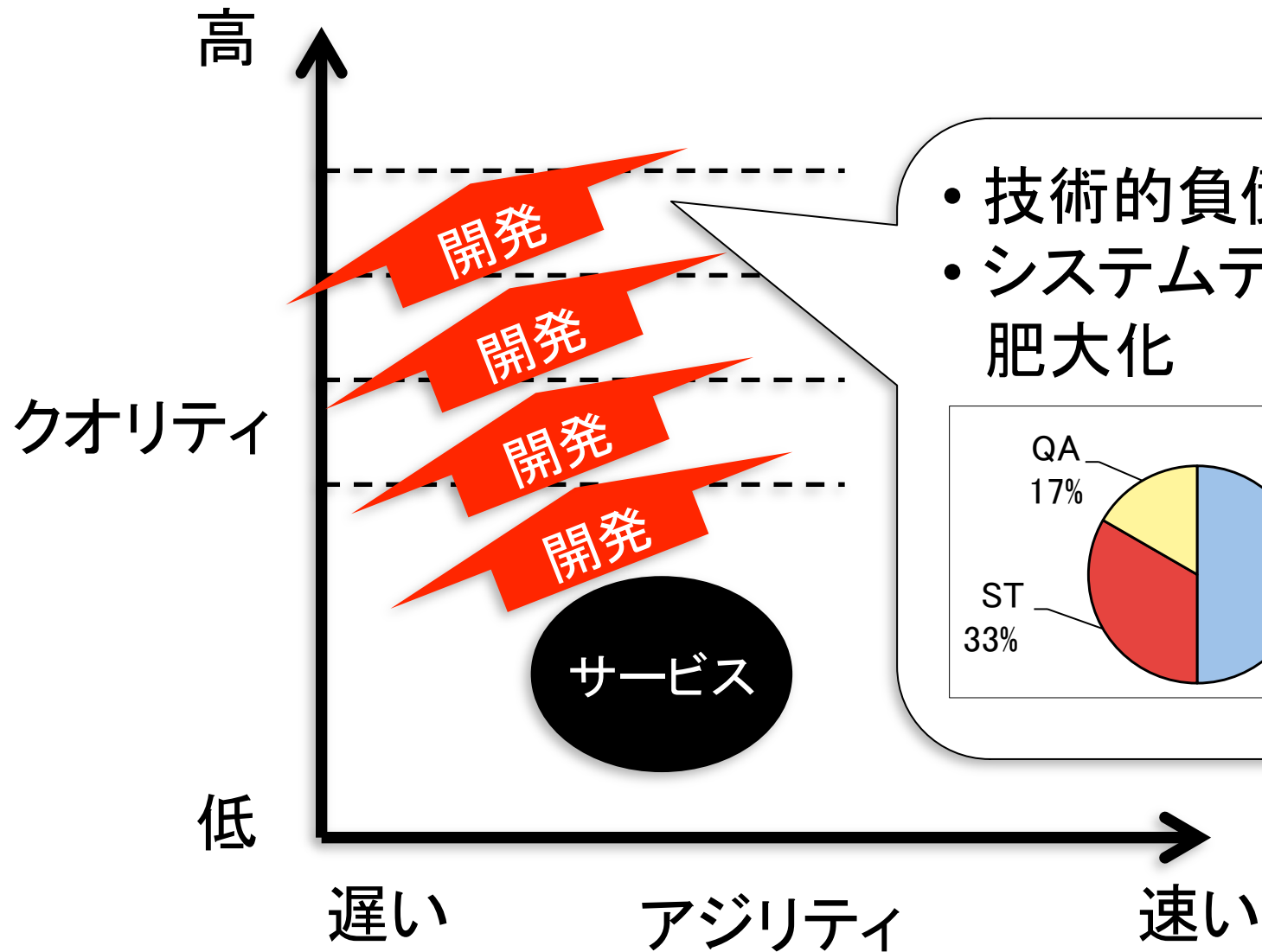
機能開発のリードタイム
バグ修正日数 など

クオリティ vs アジリティ



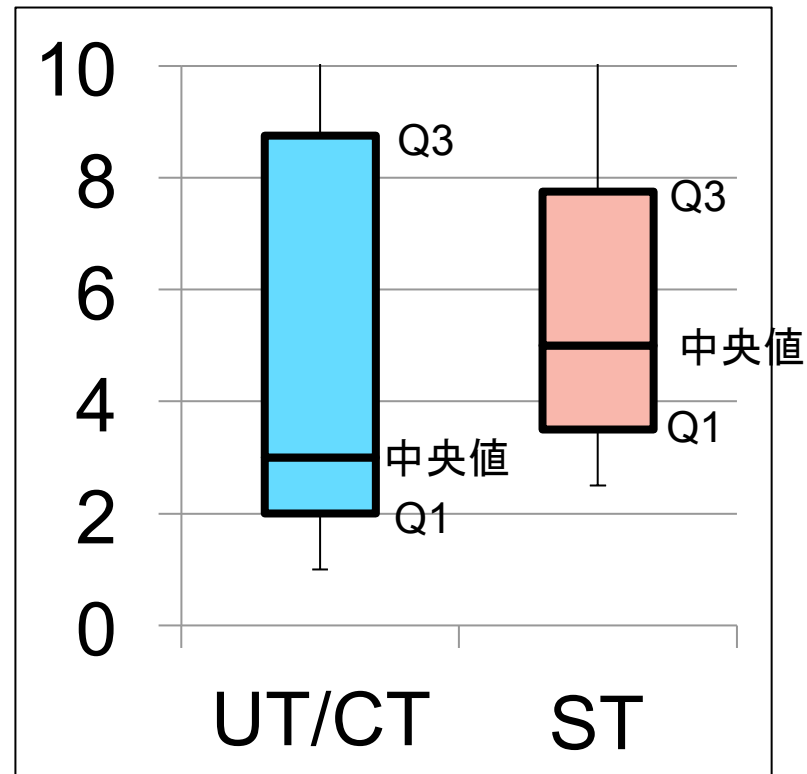
機能開発のリードタイム
バグ修正日数 など

クオリティ vs アジリティ



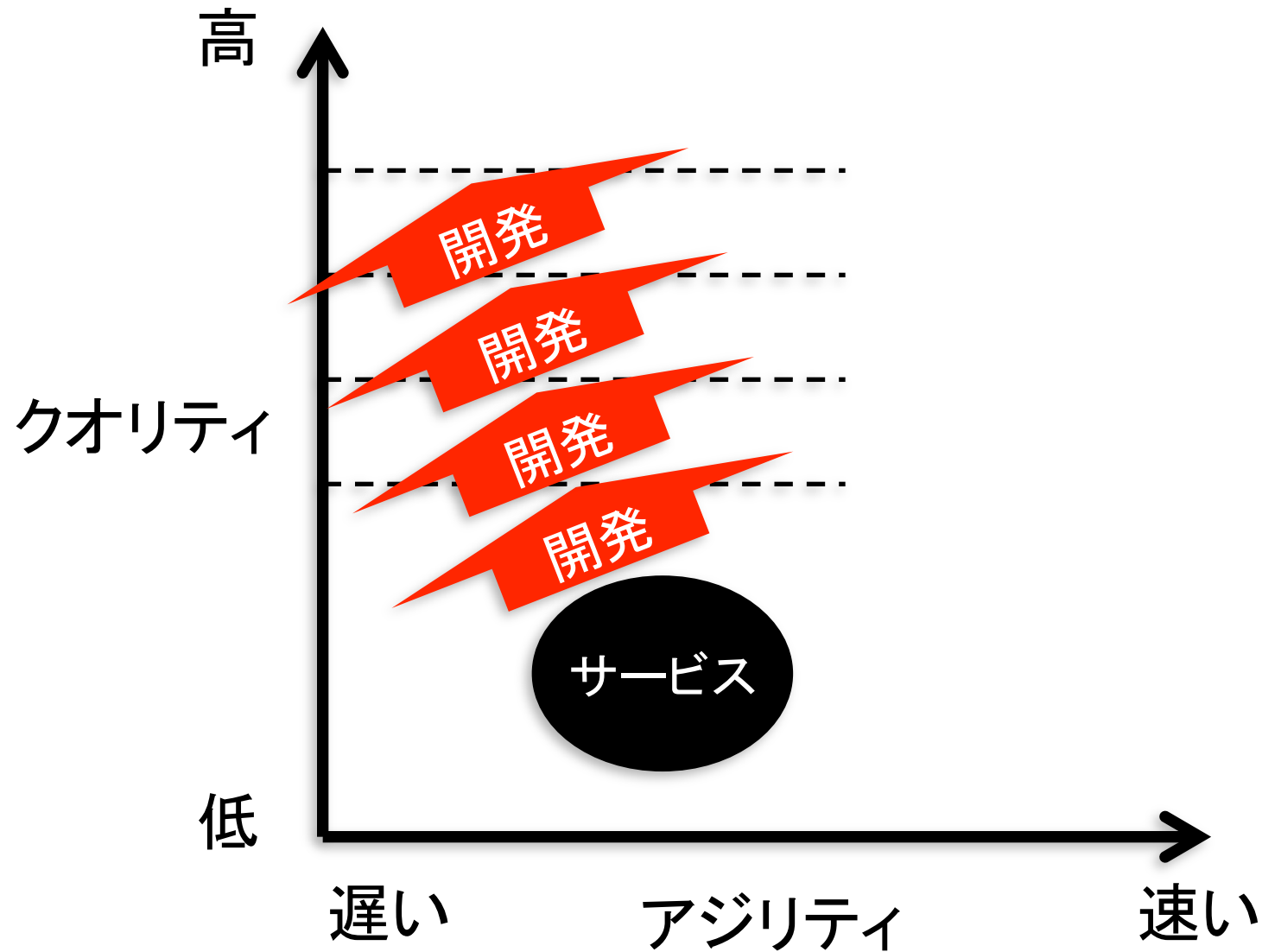
システムテストの問題: バグ修正日数

バグ修正日数



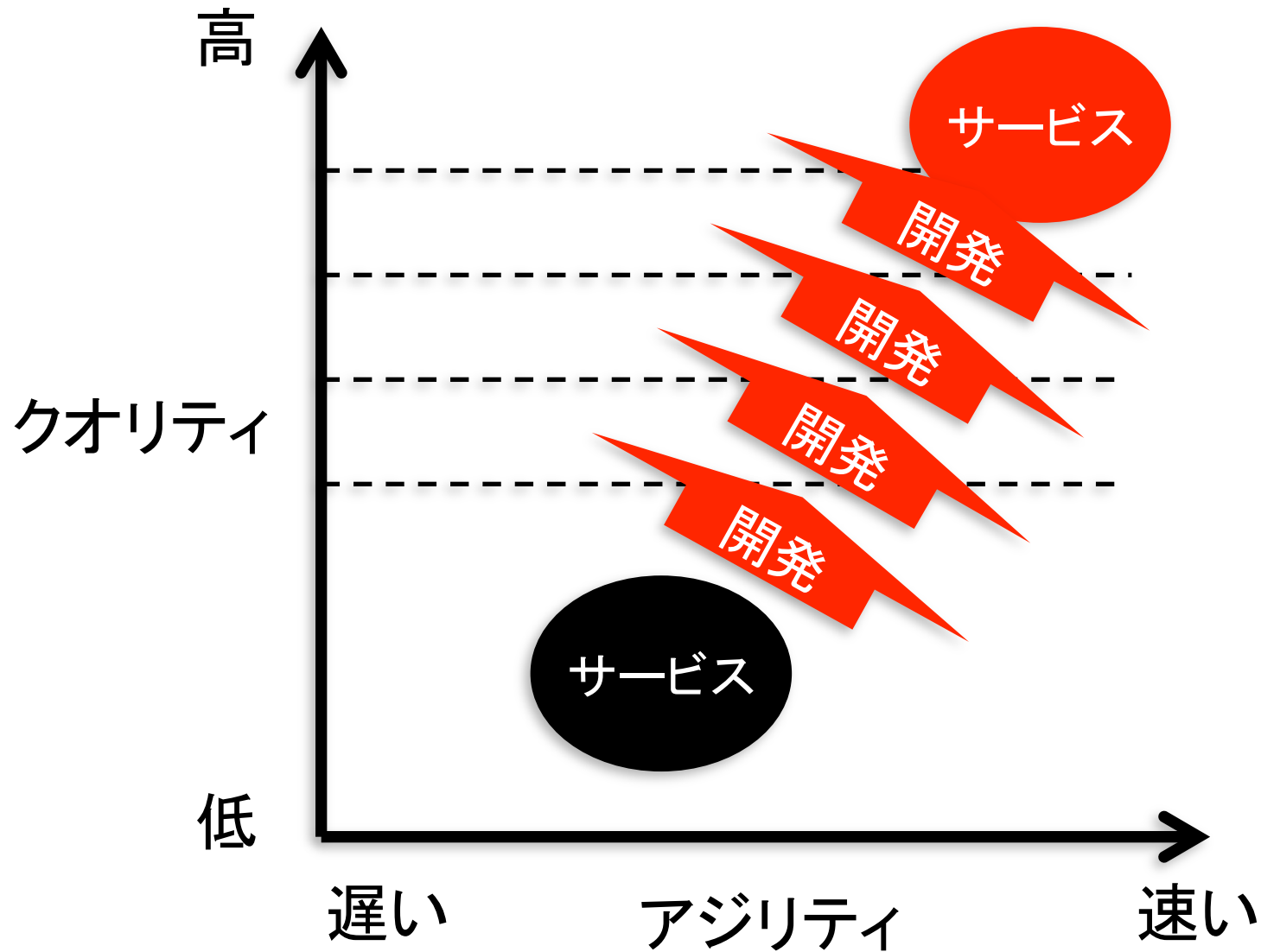
システムテストで
見つかったバグは
修正コストが高い

クオリティ vs アジリティ



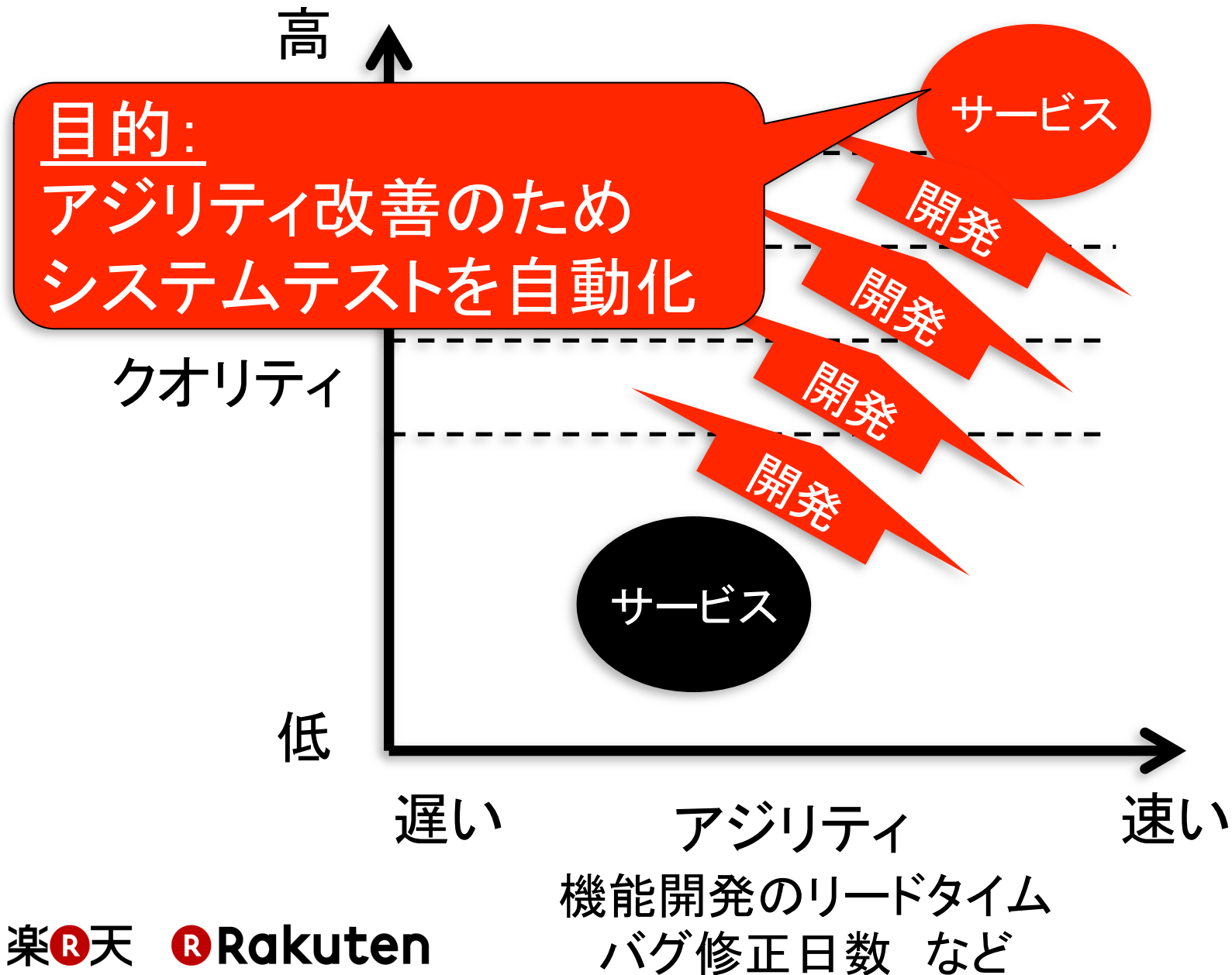
機能開発のリードタイム
バグ修正日数 など

クオリティ vs アジリティ



機能開発のリードタイム
バグ修正日数 など

システムテスト自動化の目的



継続的 システムテスト

- コンセプト
- 課題と解決方法

アジャイルなテストのアプローチにおけるシステムテスト

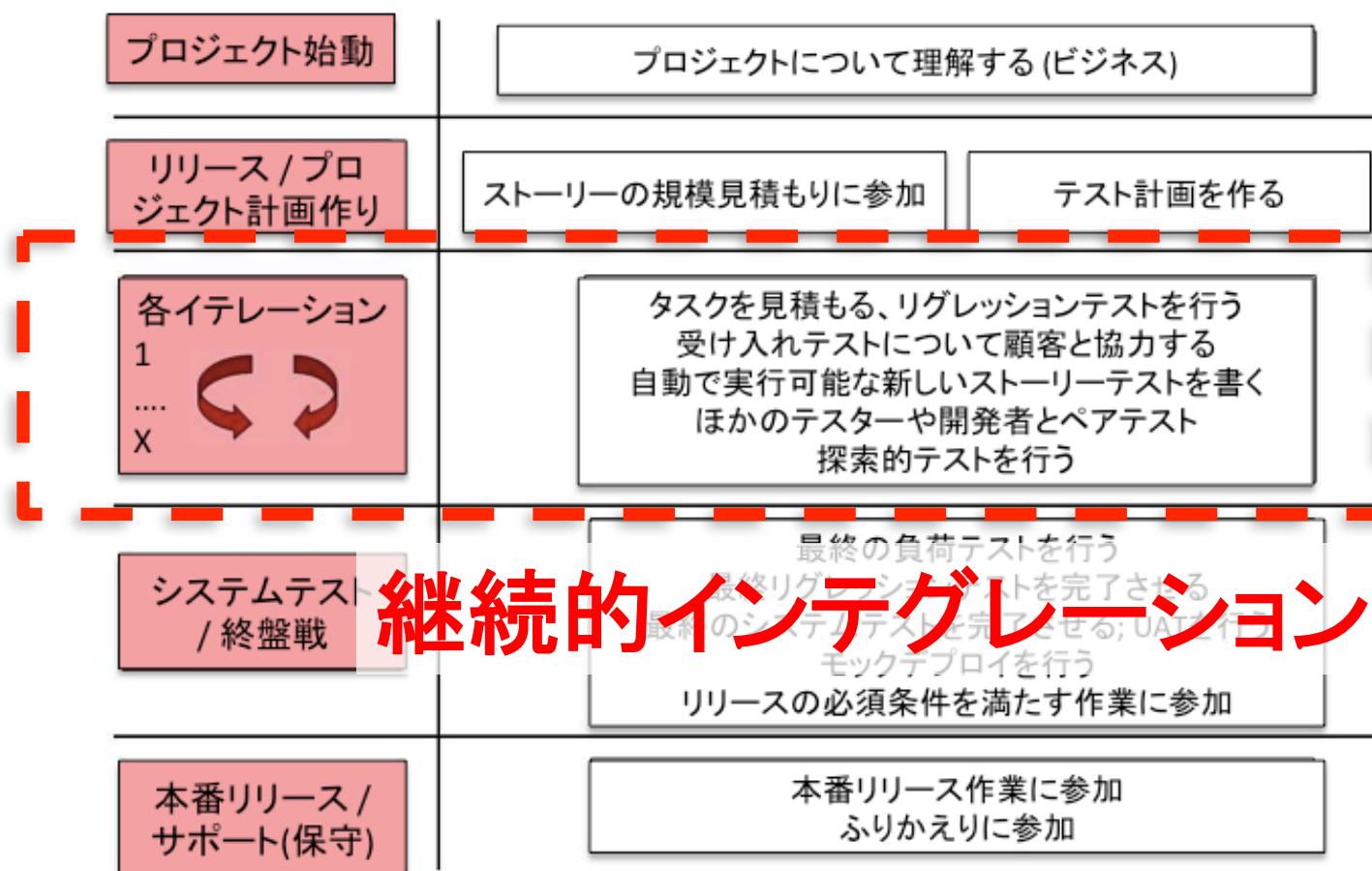
アジャイルなテストのアプローチ

プロジェクト始動	プロジェクトについて理解する (ビジネス)
リリース / プロジェクト計画作り	ストーリーの規模見積もりに参加 テスト計画を作る
各イテレーション 1 X	タスクを見積もる、リグレーションテストを行う 受け入れテストについて顧客と協力する 自動で実行可能な新しいストーリーテストを書く ほかのテスターや開発者とペアテスト 探索的テストを行う
システムテスト / 終盤戦	最終の負荷テストを行う 最終リグレーションテストを完了させる 最終のシステムテストを完了させる; UATを行う モックデプロイを行う リリースの必須条件を満たす作業に参加
本番リリース / サポート(保守)	本番リリース作業に参加 ふりかえりに参加

16

アジャイルなテストのアプローチにおけるシステムテスト

アジャイルなテストのアプローチ



継続的インテグレーション

16

アジャイルなテストのアプローチにおけるシステムテスト

アジャイルなテストのアプローチ

プロジェクト始動	プロジェクトについて理解する (ビジネス)
リリース / プロジェクト計画作り	ストーリーの規模見積もりに参加 テスト計画を作る
各イテレーション 1 X	タスクを見積もる、リグレッションテストを行う 受け入れテストについて顧客と協力する 自動で実行可能な新しいストーリーテストを書く ほかのテスターや開発者とペアテスト 探索的テストを行う
システムテスト / 終盤戦	最終の負荷テストを行う 最終リグレッションテストを完了させる 最終のシステムテストを完了させる; UATを行う モックデプロイを行う リリースの必須条件を満たす作業に参加
本番リリース / サポート(保守)	本番リリース作業に参加 サポート作業に参加

品質の門番

検索ランキングのデグレ
運用性、可用性のデグレ

システムレベルの検証が必要な項目①: 検索ランキング

Fearless change 特集検索 ショップ 全商品一覧

あす楽 | 海外販売 | ランキング | オークション | ブックス

無料アプリ 初めて電子書籍購入で 楽天スーパーポイント500ポイントプレゼント

トップ > 「Fearless change」の検索結果

並び替え 1件~45件 (全 273件)

在庫の有無

関連キーワード フィアレス プラチナム・エディション テイラー・スウィフト フィアレス
Fearless Platinum Edition

	Fearless Change [川口恭伸] 【送料無料】 <input checked="" type="button" value="楽天ブックス"/> 比較 ★お気に入り	2,625 円 送料込 ↑ 最安ショップを見る 翌日配送
	アジャイルに効くアイデアを組織に広めるための48のパターン / 原タイトル: Fearless Change [本/雑...] 【送料無料選択可!】 <input checked="" type="button" value="CD&DVD NEOWING"/> 比較 ★お気に入り	2,625 円 送料別 ↑ 最安ショップを見る
	【送料無料】 Fearless Change アジャイルに効くアイデアを組織に広めるための48のパターン / 川口... <input checked="" type="button" value="LAWSONほっとステーション"/>	2,625 円 送料込 ↑ 最安ショップを見る

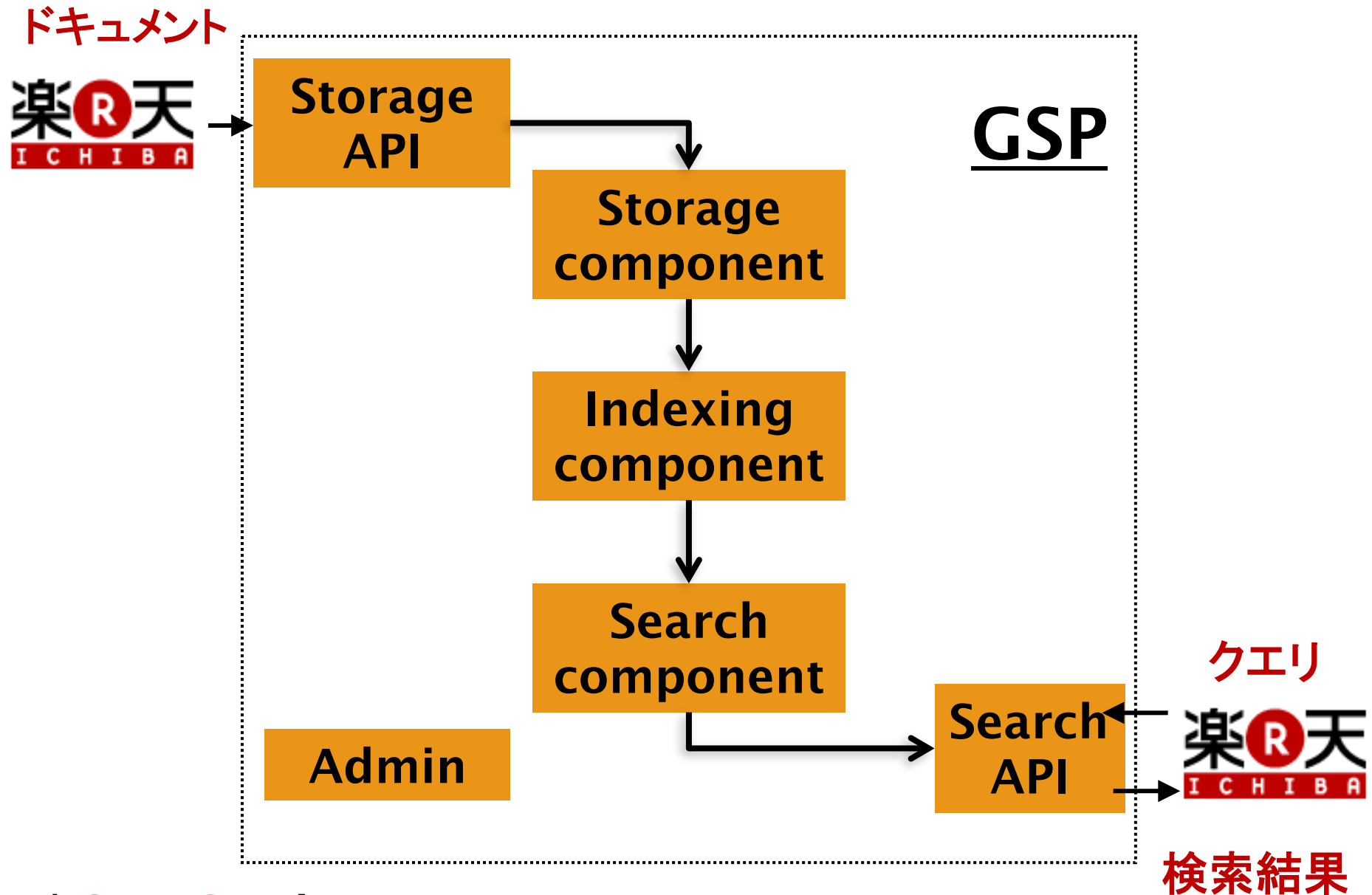
システムレベルの検証が必要な項目①: 検索ランキング

検索キーワード

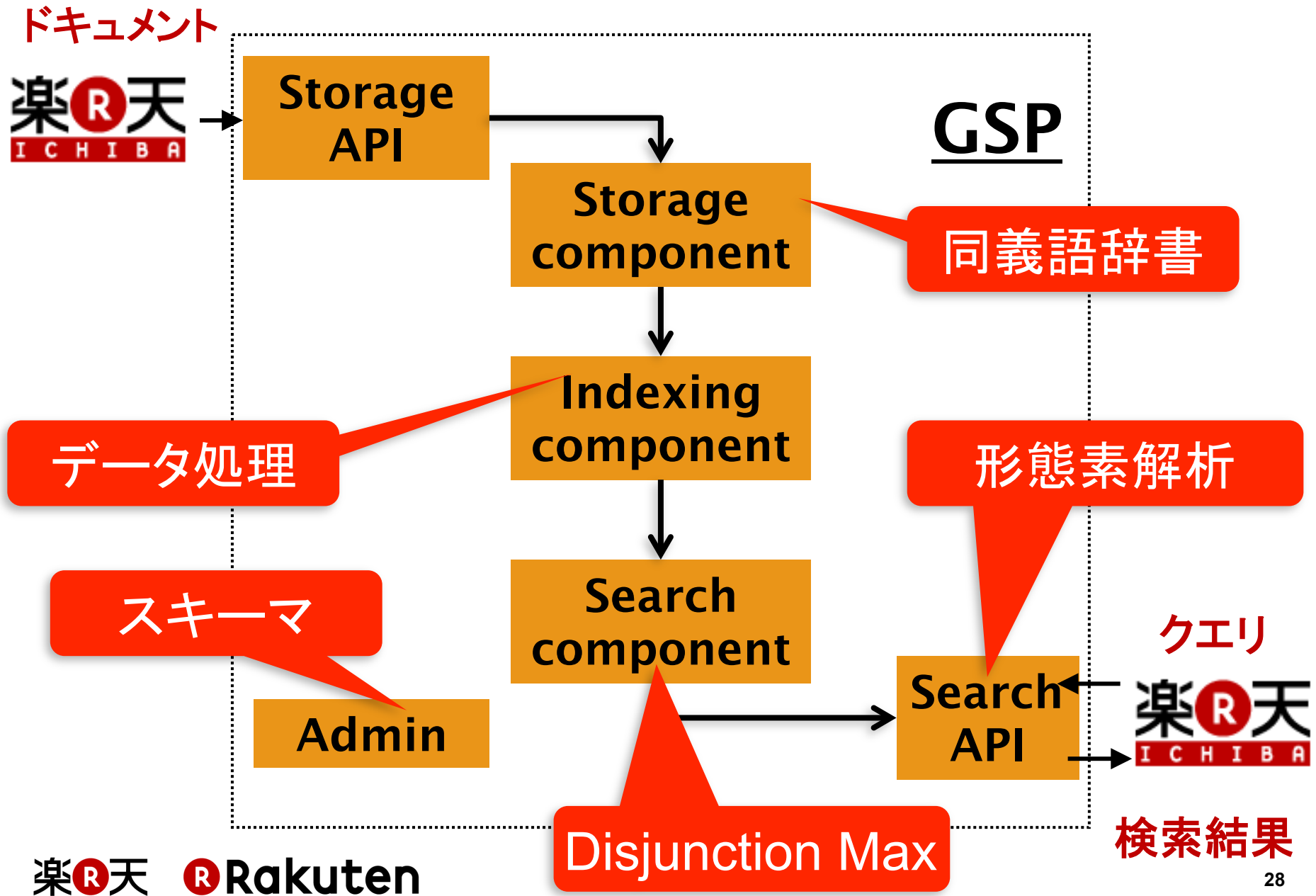


検索キーワードへの関連順で並べる

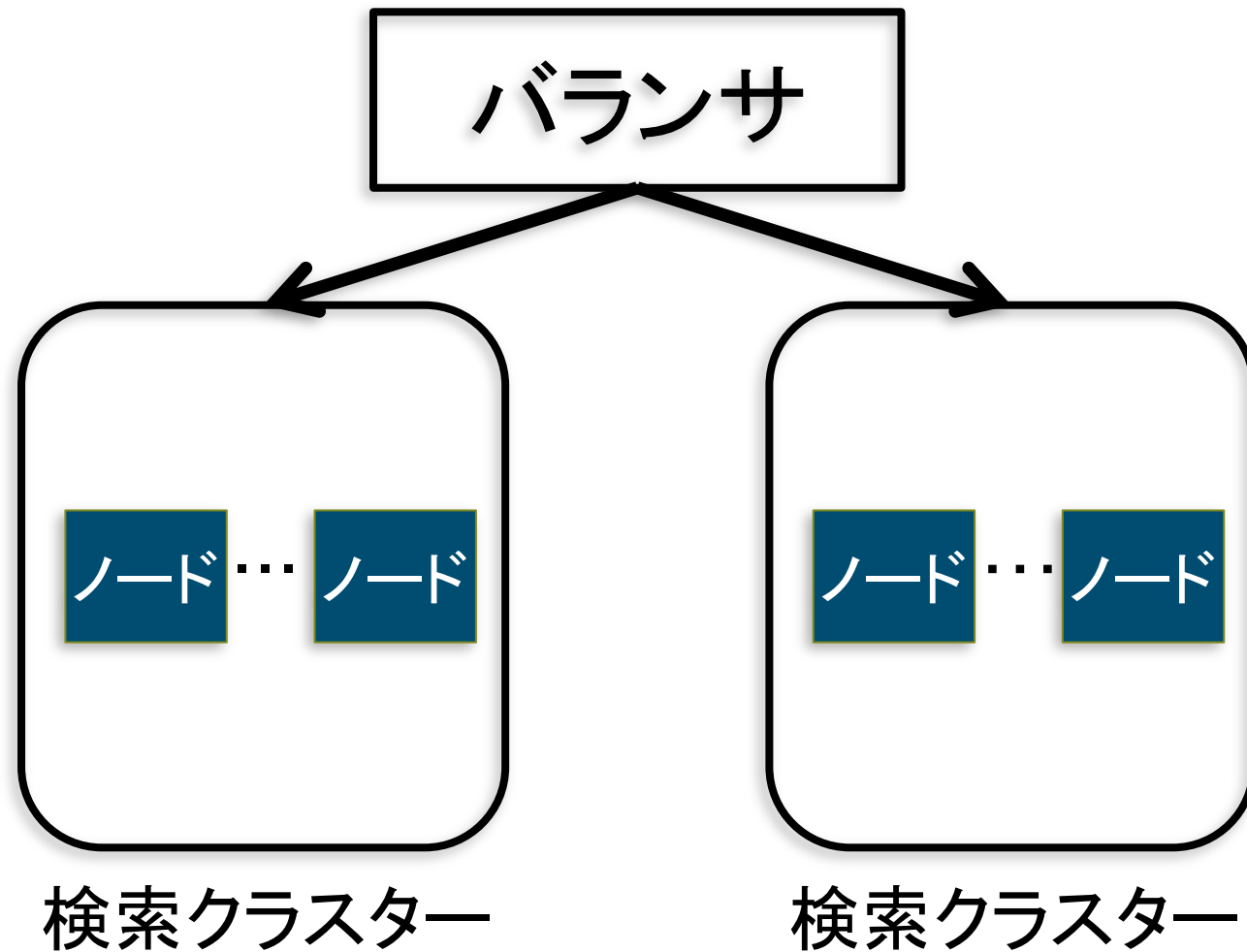
検索ランキングに関連するコンポーネント



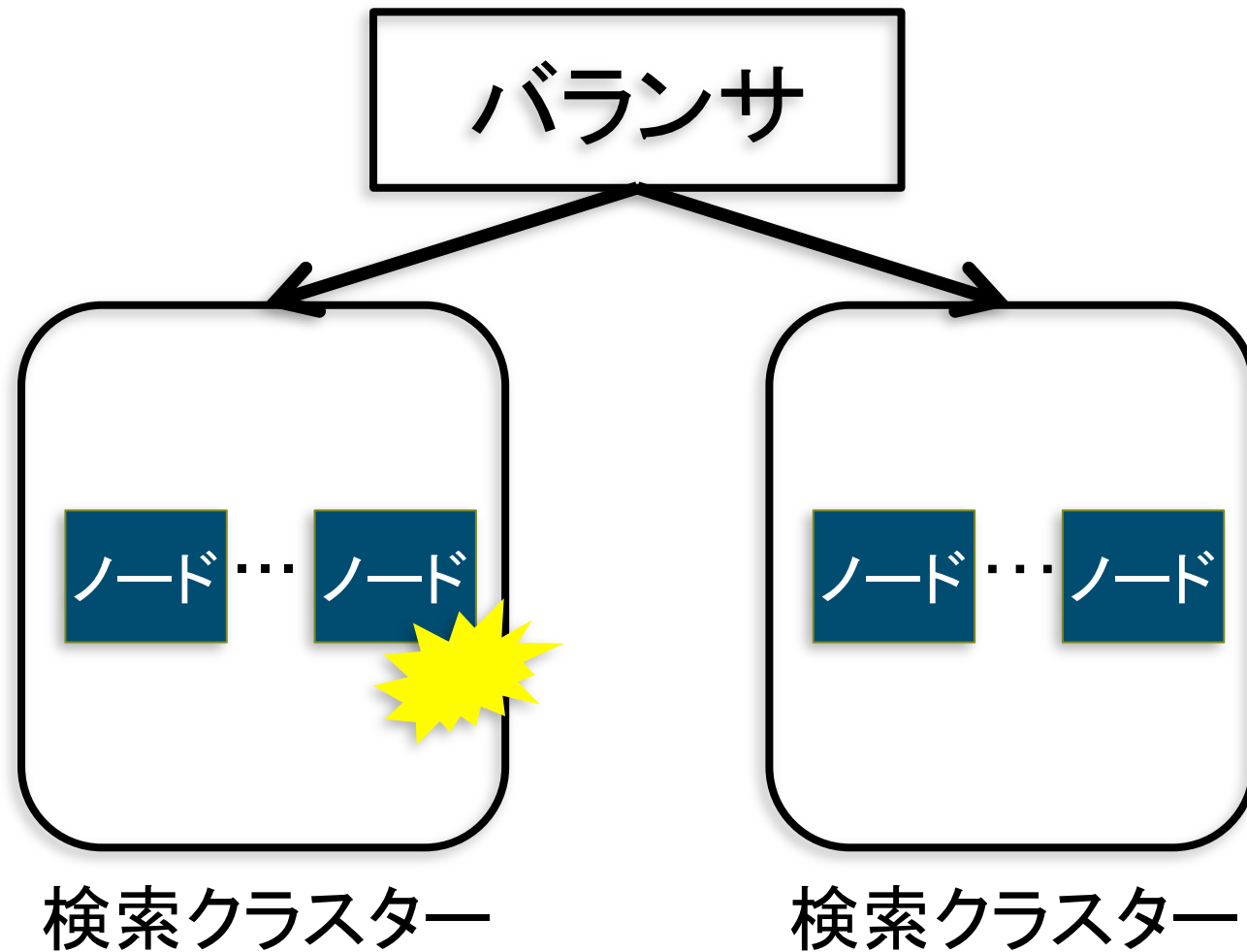
検索ランキングに関連するコンポーネント



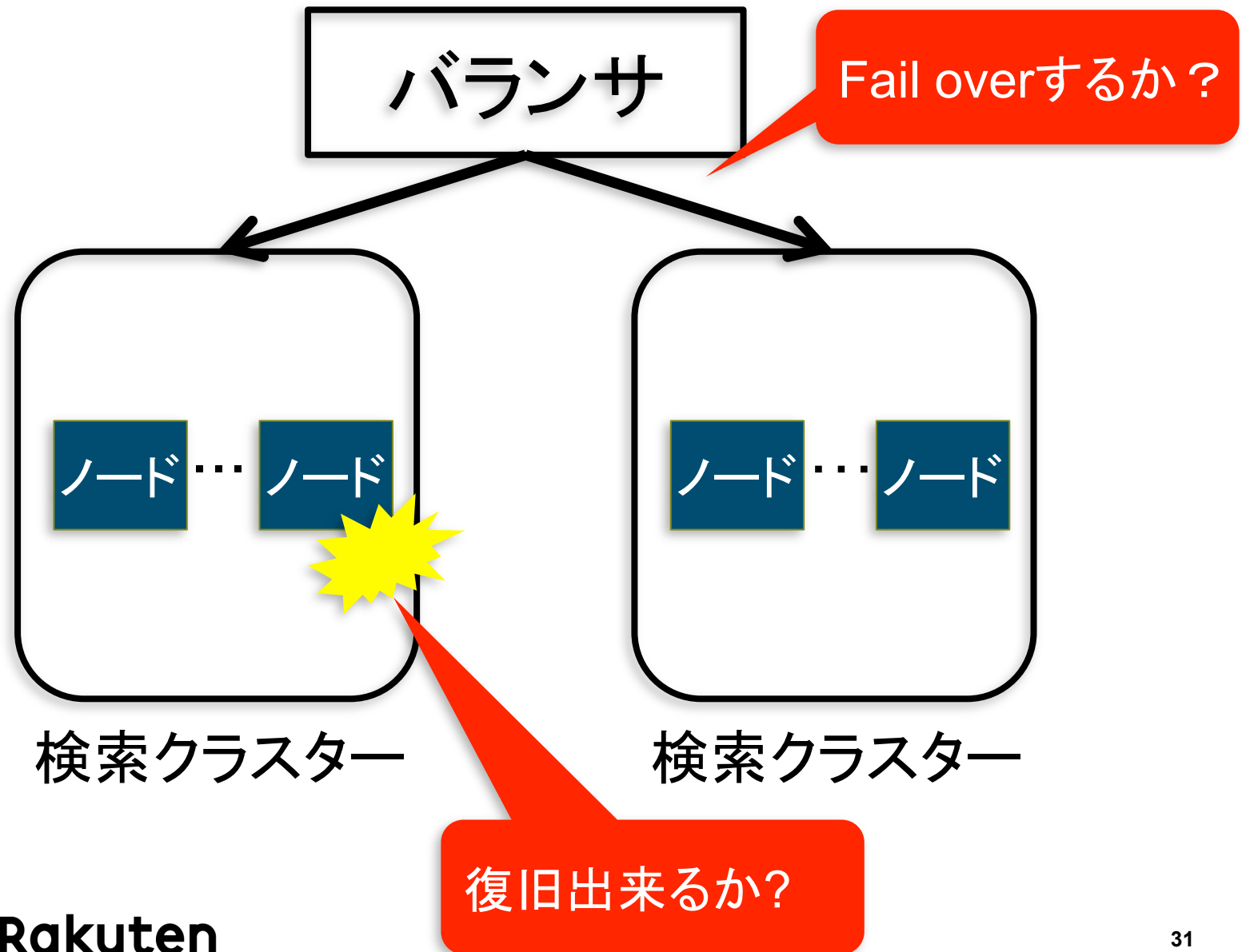
システムレベルの検証が必要な項目②: 可用性・運用性



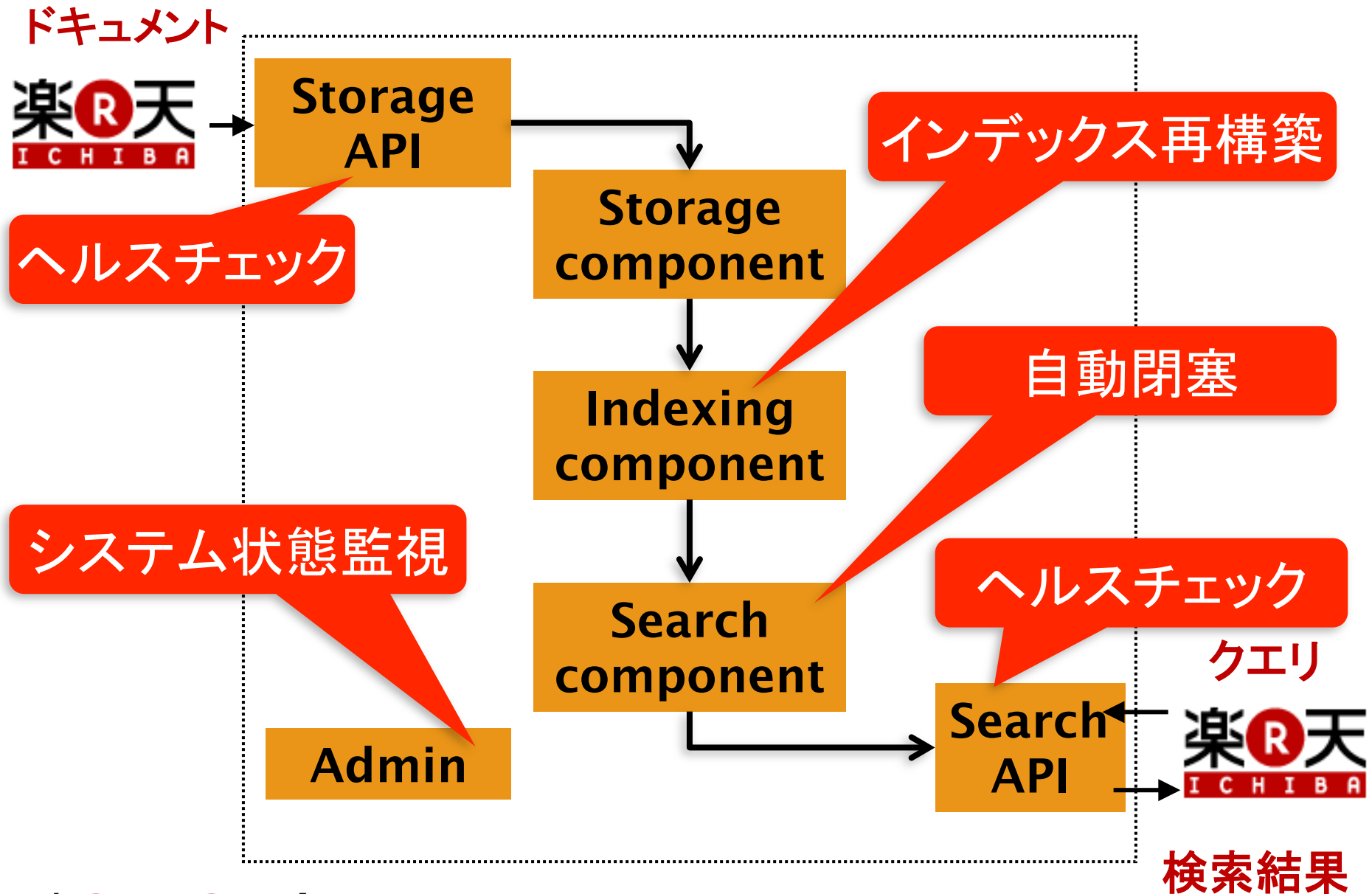
システムレベルの検証が必要な項目②: 可用性・運用性



システムレベルの検証が必要な項目②: 可用性・運用性



運用性・可用性に関連するコンポーネント



“継続的システムテスト”

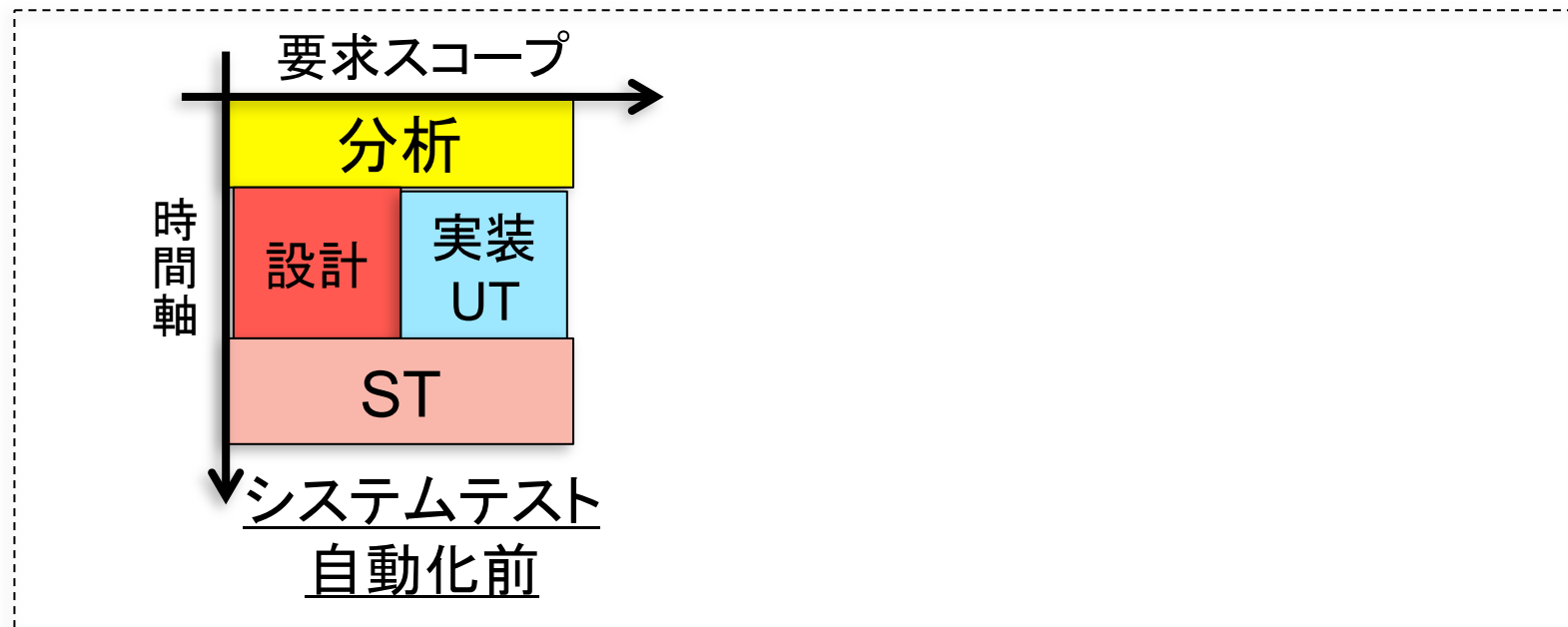
- コンセプト

検索ランキングや、運用性、可用性などの
システムレベルの検証も継続的に実行

“継続的システムテスト”

- コンセプト

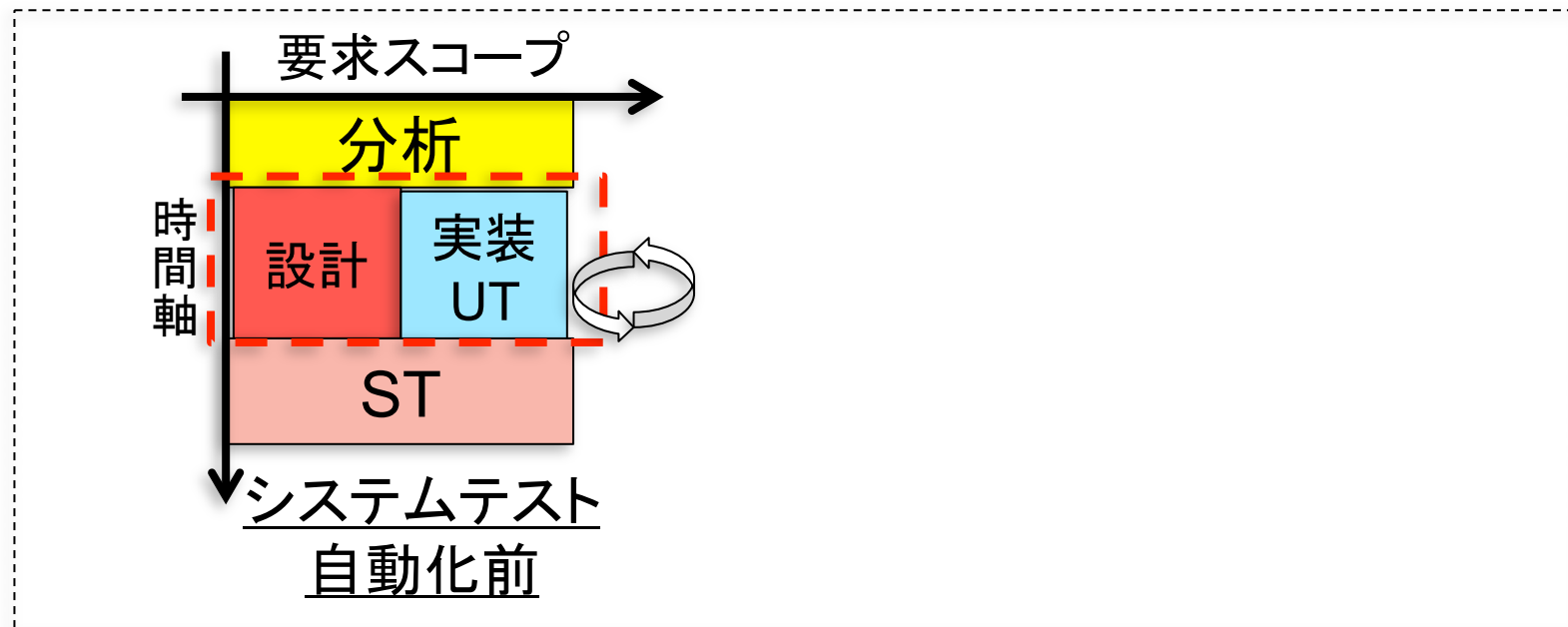
検索ランキングや、運用性、可用性などの
システムレベルの検証も継続的に実行



“継続的システムテスト”

- コンセプト

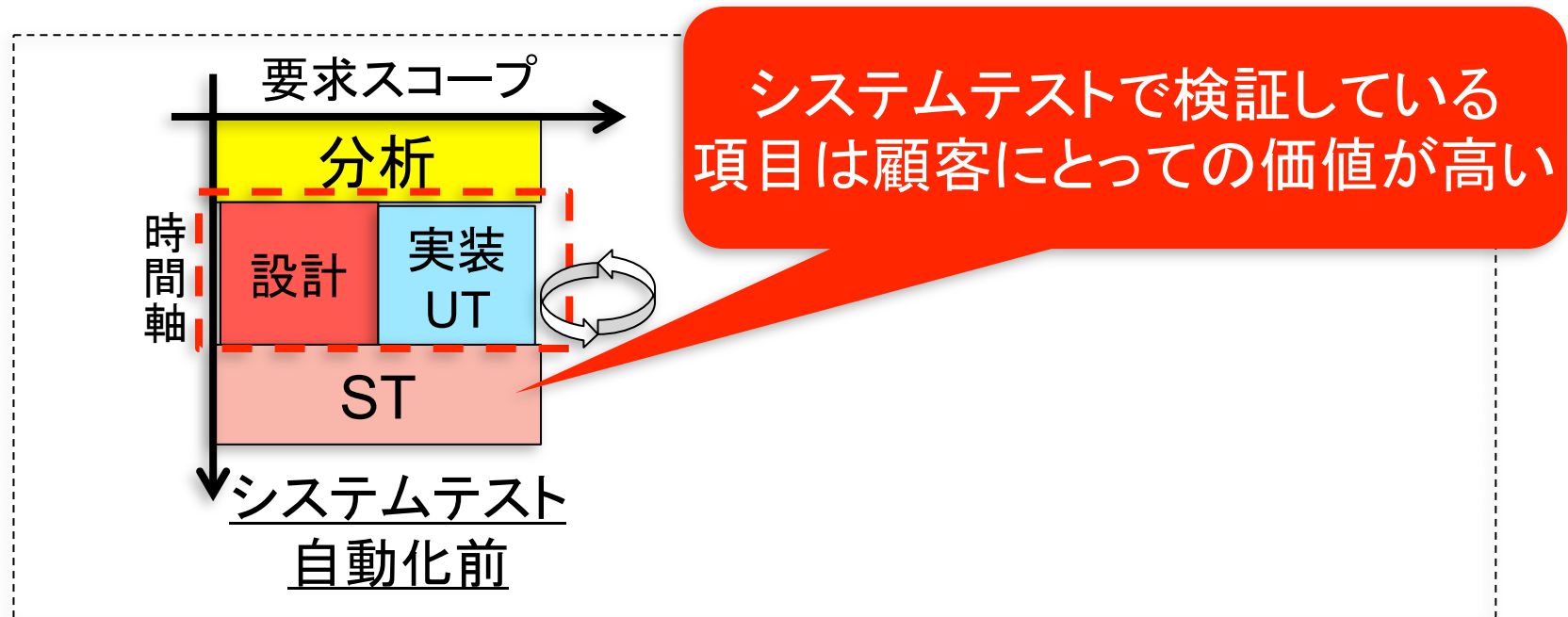
検索ランキングや、運用性、可用性などの
システムレベルの検証も継続的に実行



“継続的システムテスト”

• コンセプト

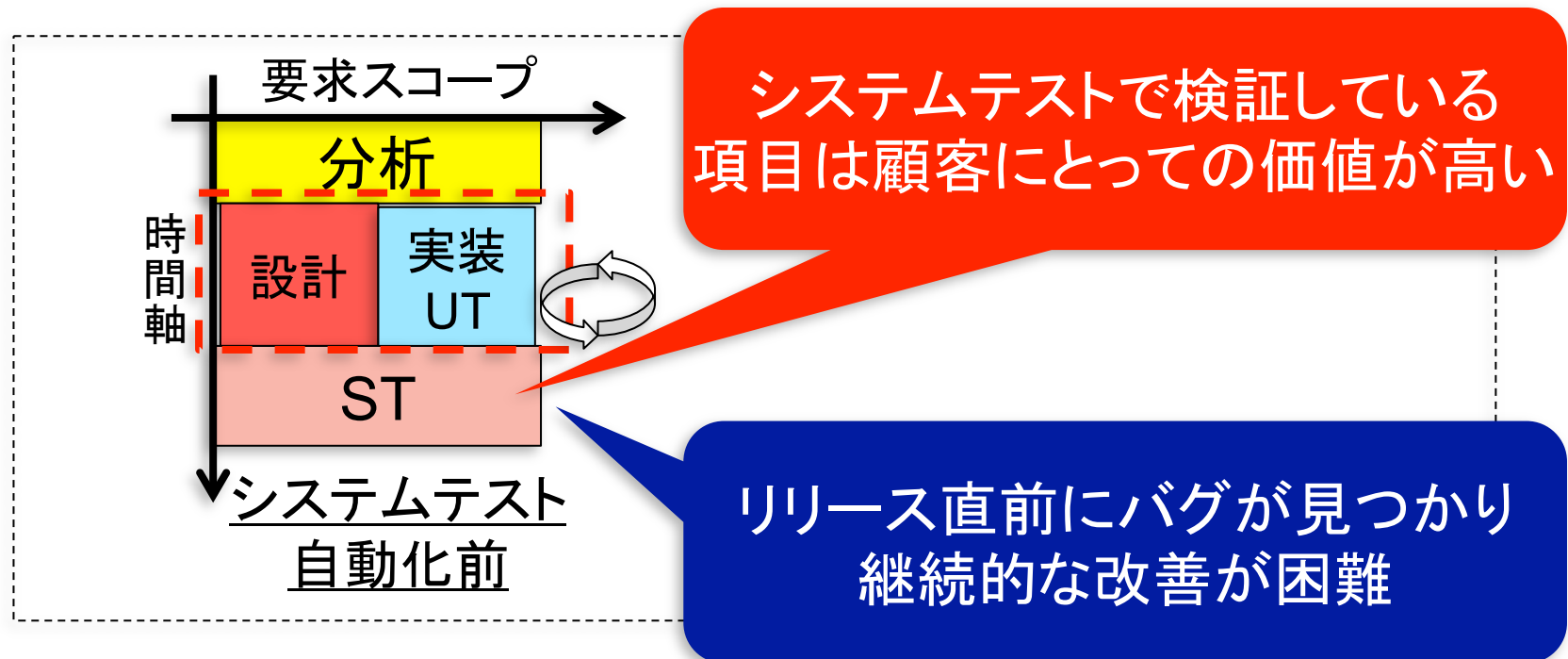
検索ランキングや、運用性、可用性などの
システムレベルの検証も継続的に実行



“継続的システムテスト”

• コンセプト

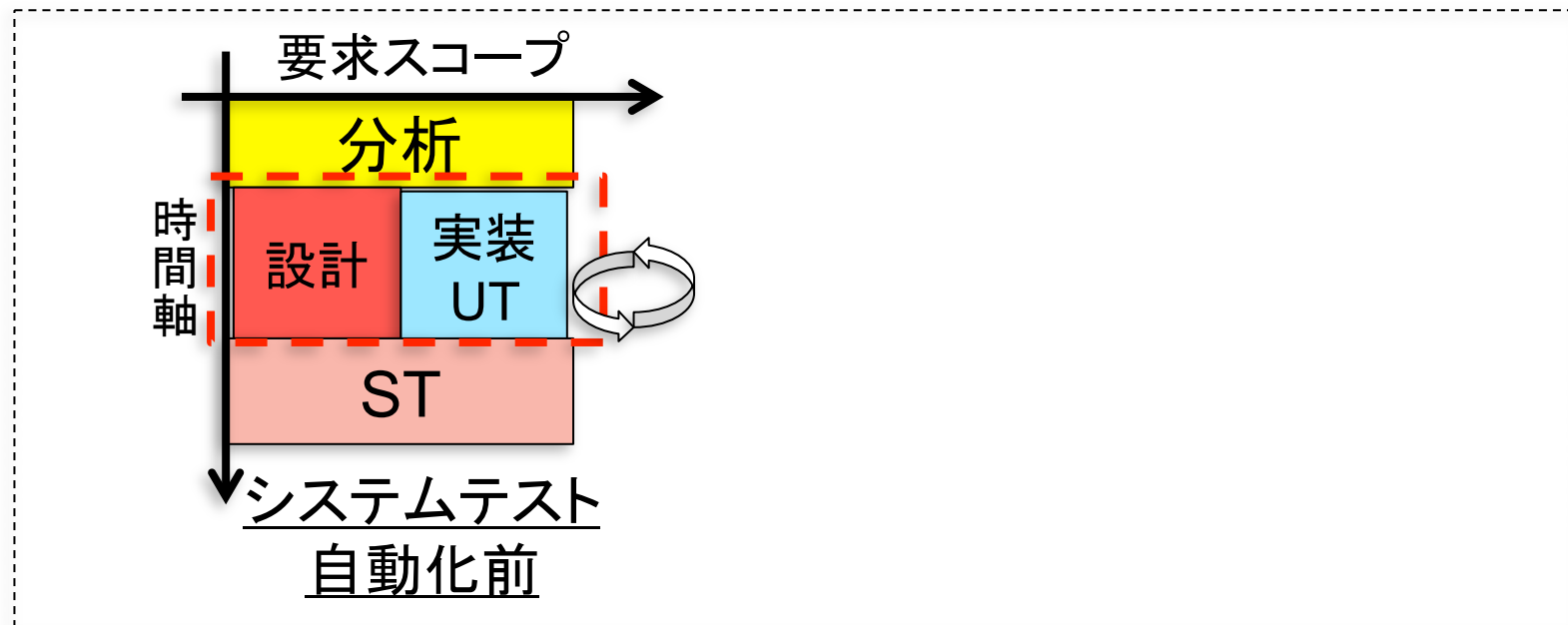
検索ランキングや、運用性、可用性などの
システムレベルの検証も継続的に実行



“継続的システムテスト”

- コンセプト

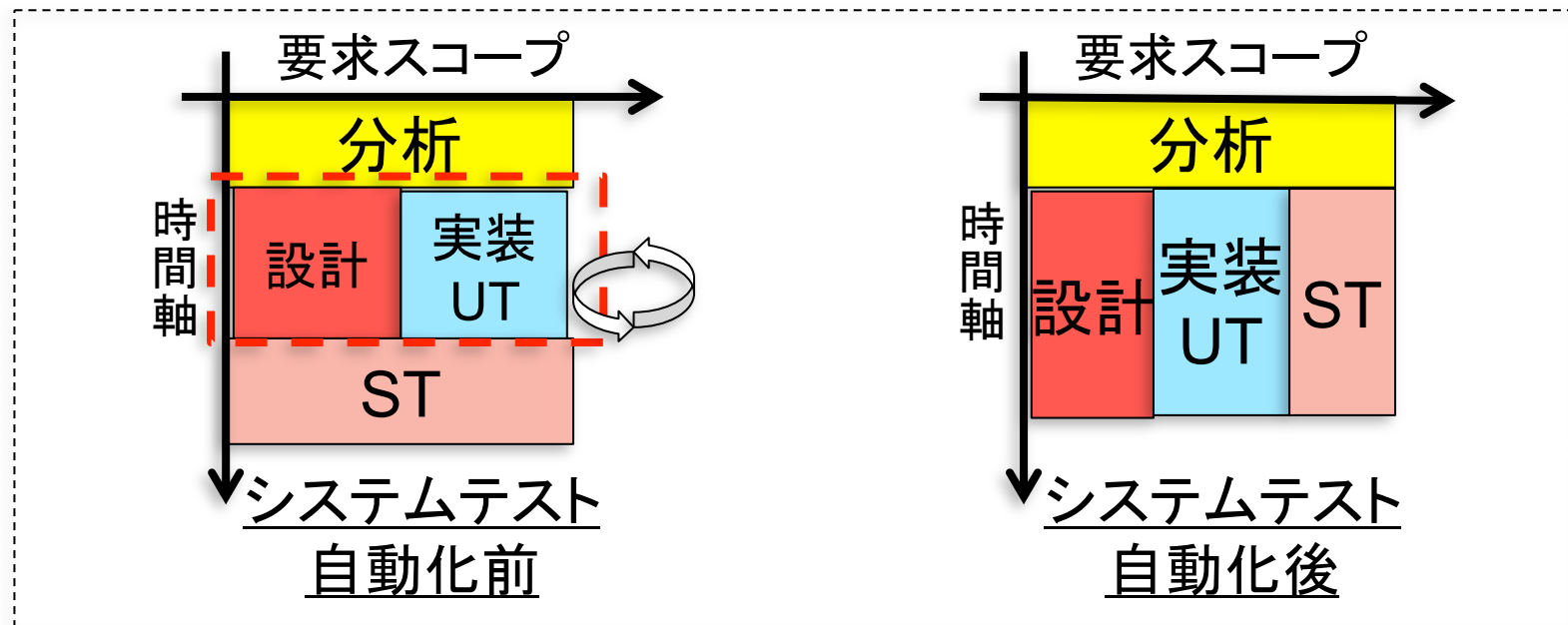
検索ランキングや、運用性、可用性などの
システムレベルの検証も継続的に実行



“継続的システムテスト”

• コンセプト

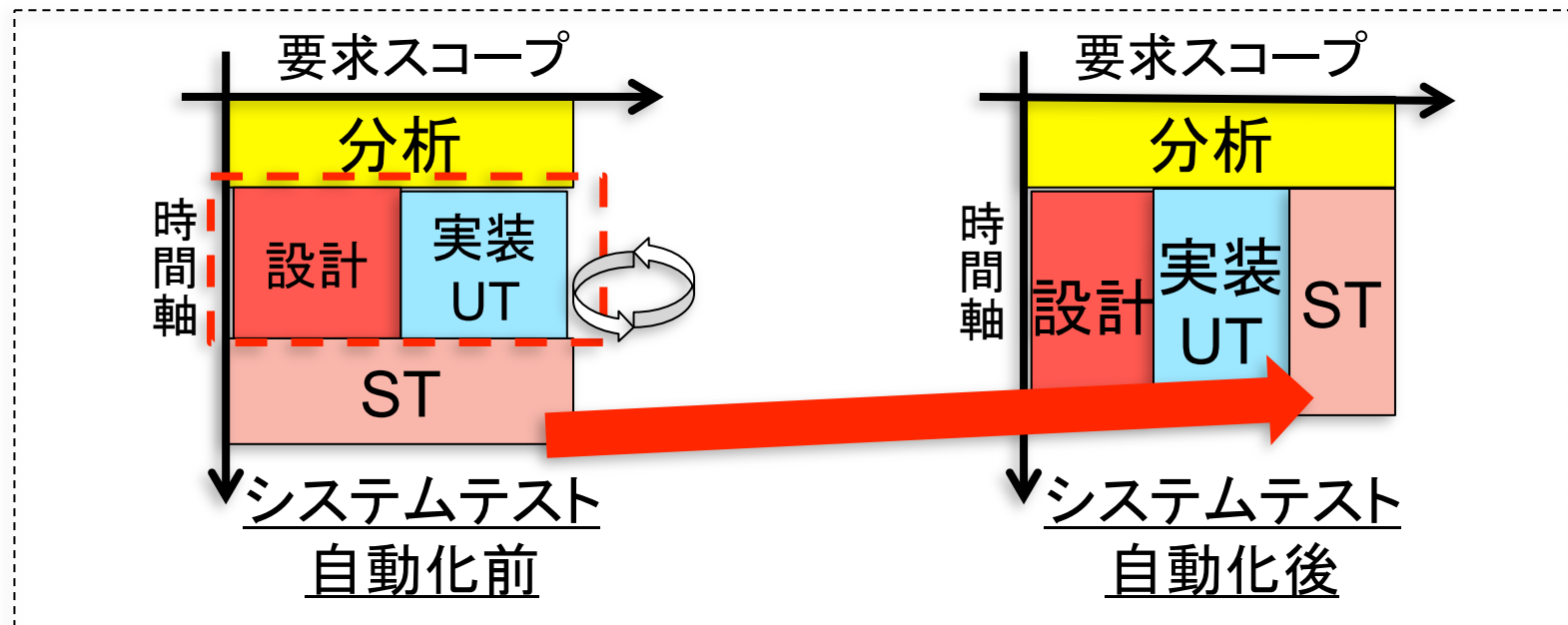
検索ランキングや、運用性、可用性などの
システムレベルの検証も継続的に実行



“継続的システムテスト”

• コンセプト

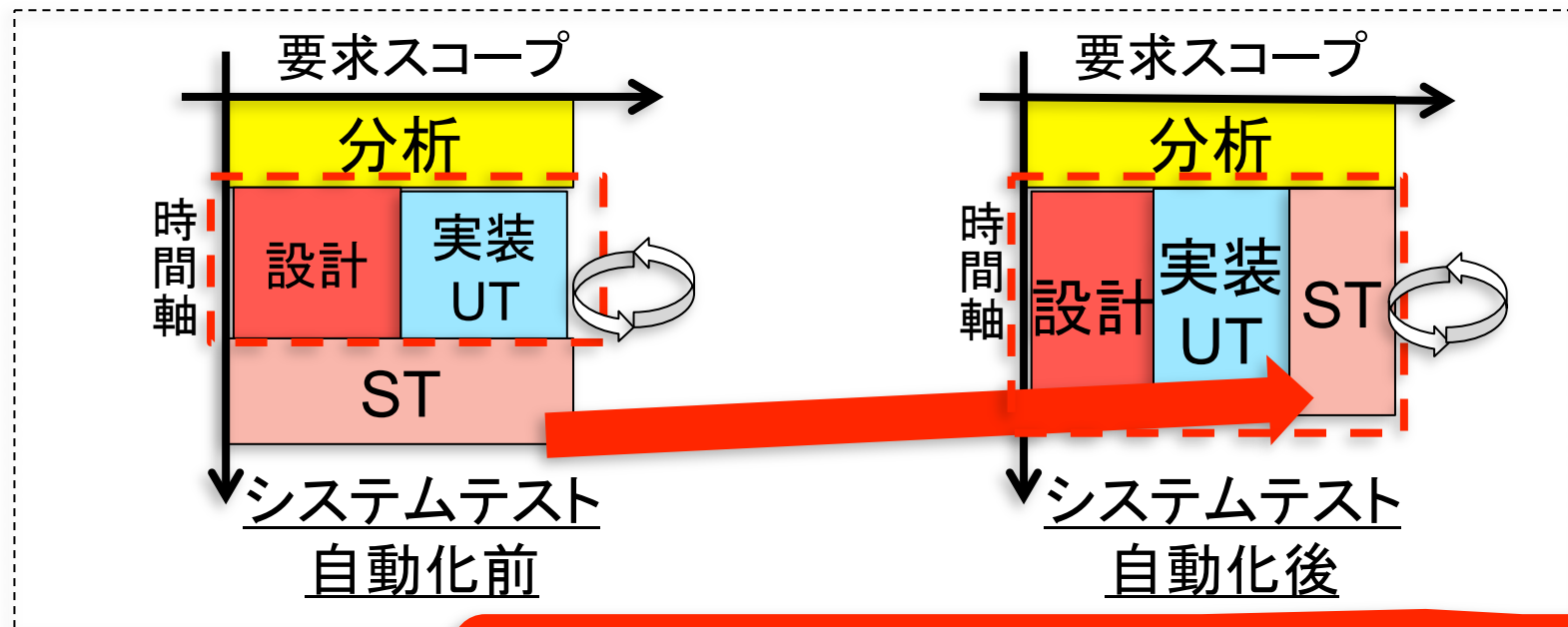
検索ランキングや、運用性、可用性などの
システムレベルの検証も継続的に実行



“継続的システムテスト”

• コンセプト

検索ランキングや、運用性、可用性などの
システムレベルの検証も継続的に実行

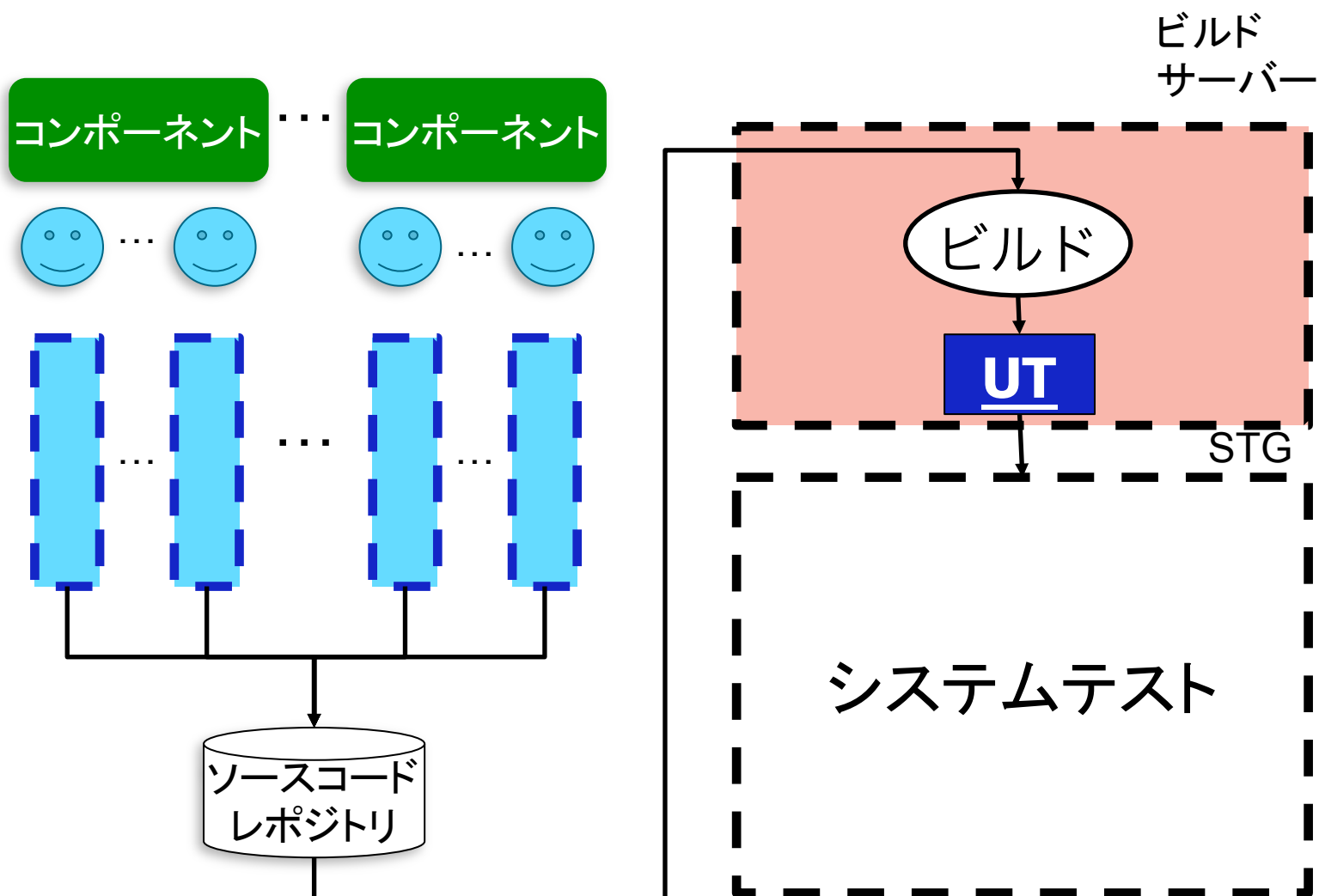


顧客にとって価値の高い領域での
開発者や分析者の継続的な改善をサポート

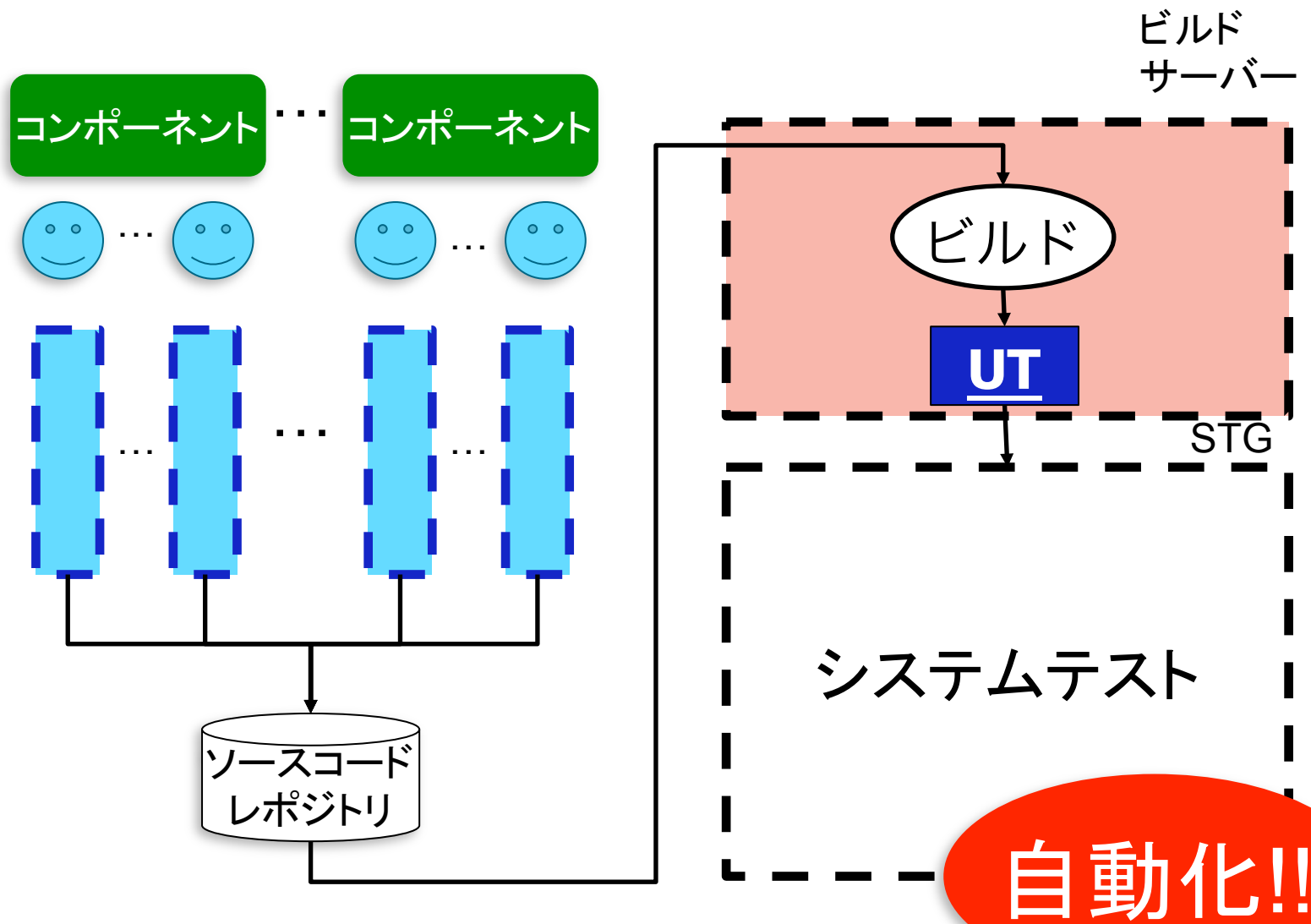
継続的 システムテスト

- コンセプト
- 課題と解決方法

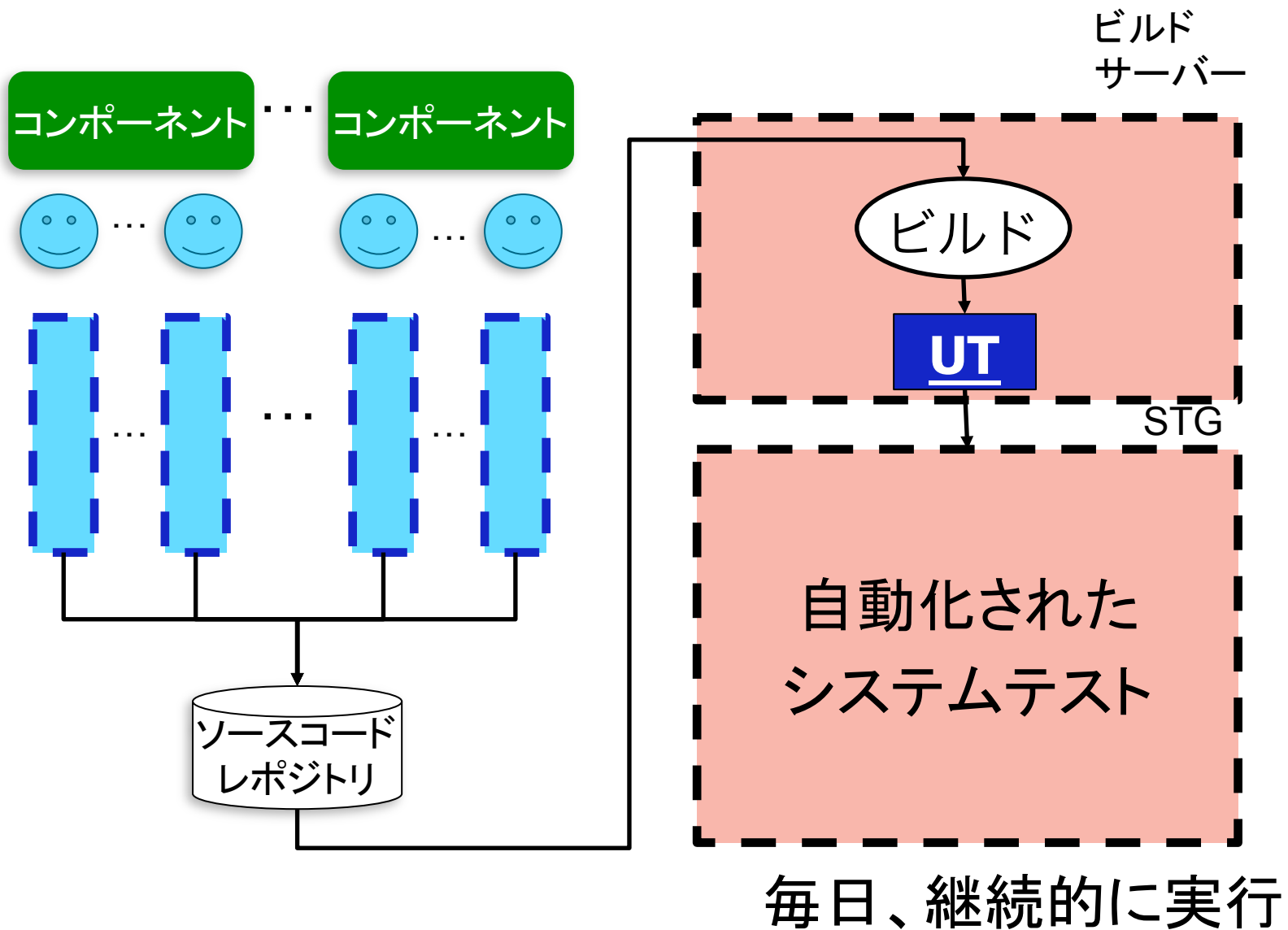
継続的インテグレーション



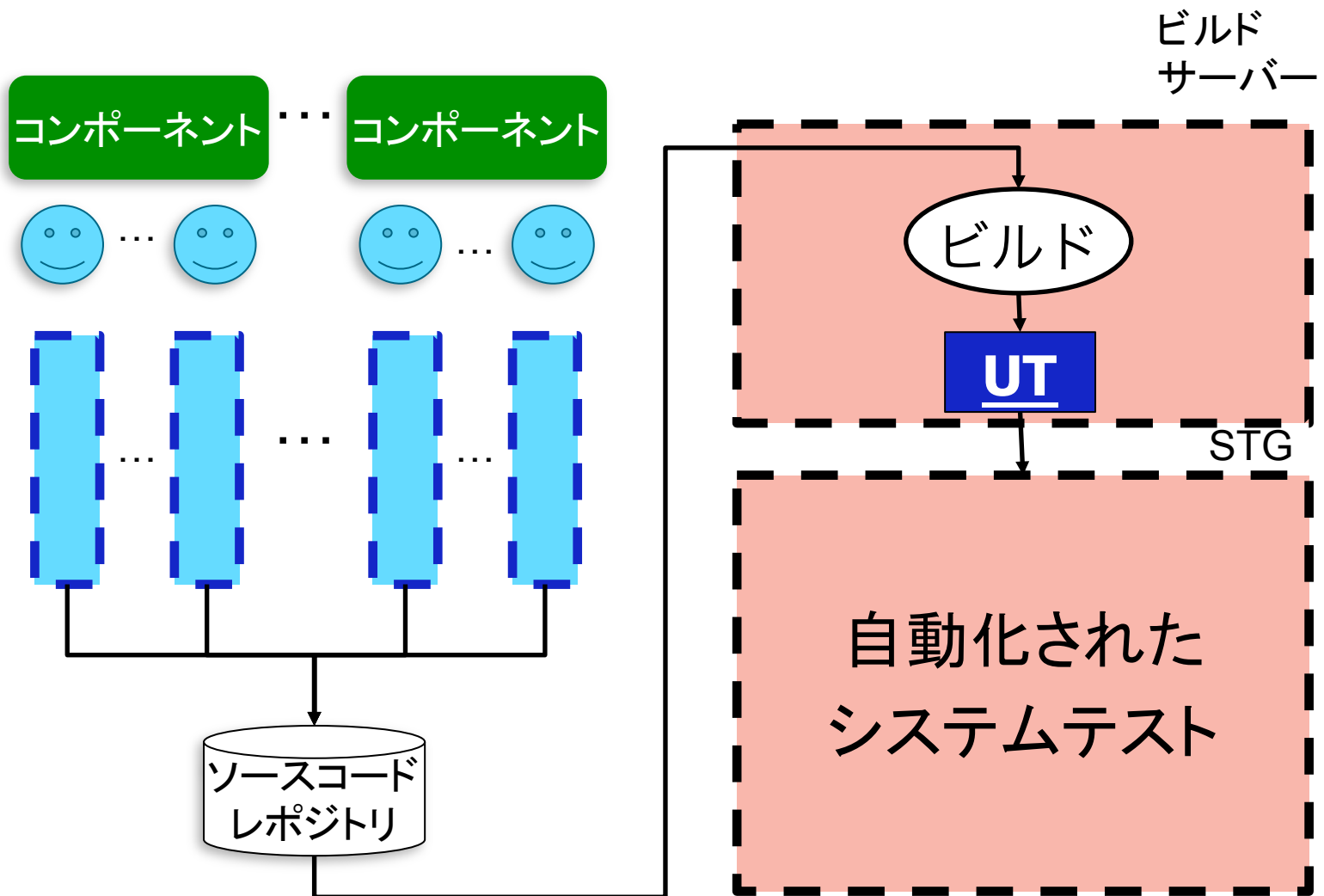
継続的インテグレーションから継続的システムテストへ



継続的システムテスト概要図



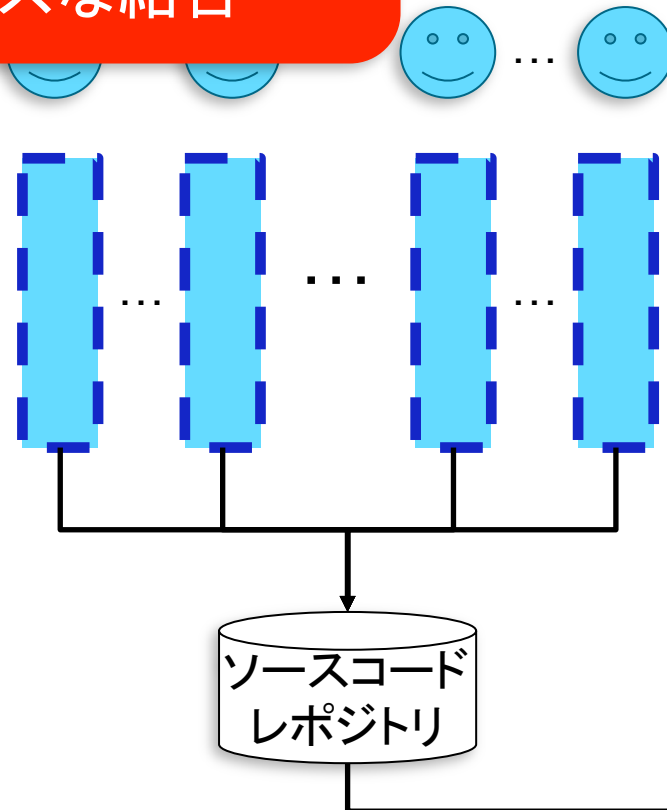
継続的システムテスト: 課題



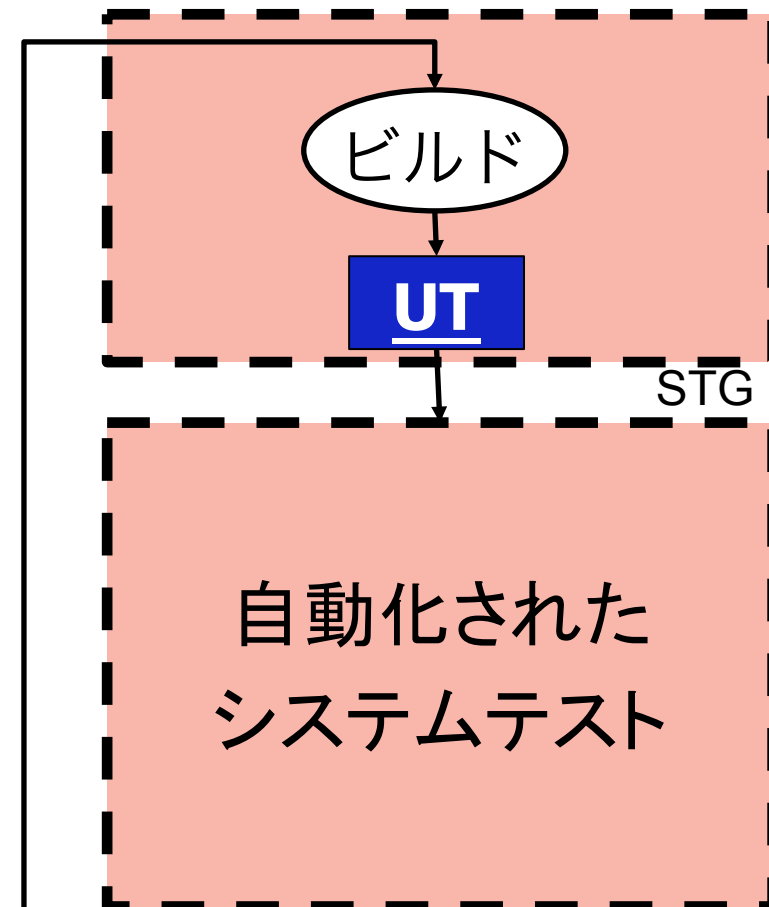
継続的システムテスト: 課題

① 複数のチームや
コンポーネントの
スムーズな結合

コンポーネント



ビルド
サーバー



継続的システムテスト: 課題

① 複数のチームや
コンポーネントの
スムーズな結合



ソースコード
レポジトリ

② システムテストの
自動化は高コスト?

ビルド
サーバー

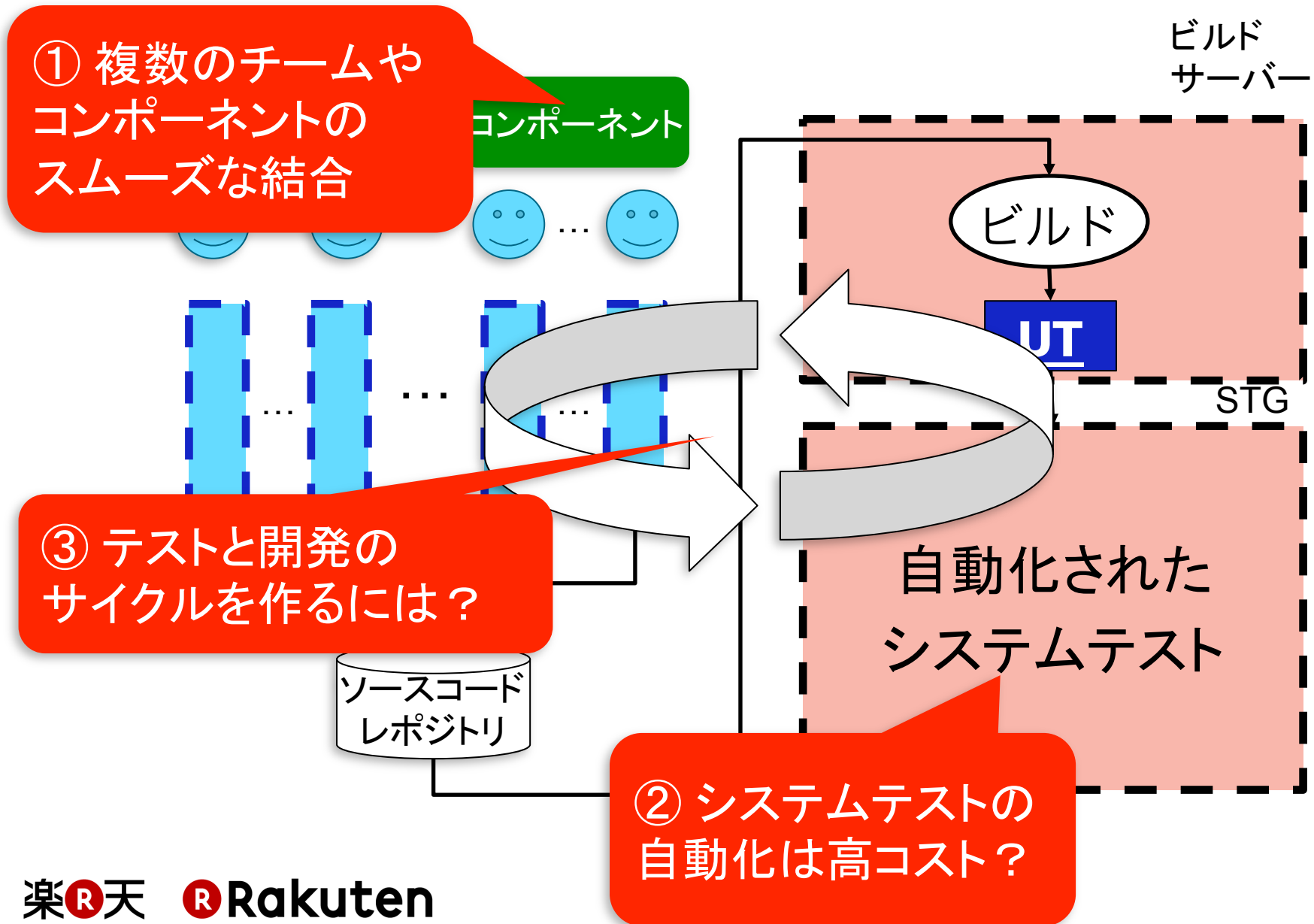
ビルド

UT

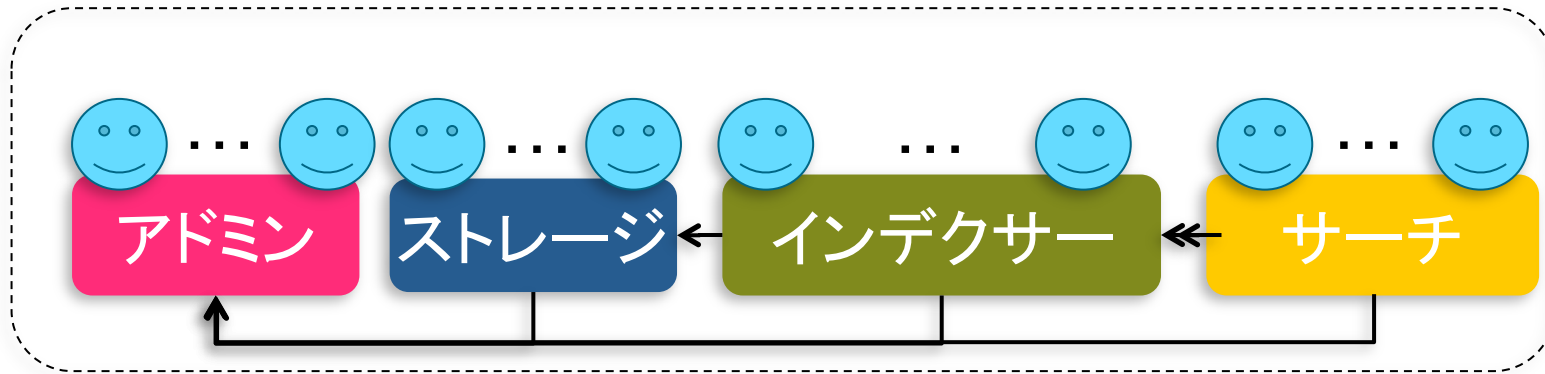
STG

自動化された
システムテスト

継続的システムテスト: 課題



自動化された継続的システムテストの課題①: スムーズな結合



問題点:

- チームの属人的な設計
→ テスト自動化可能性に問題
- 低レイヤのコンポーネントのバグ
→ 他のコンポーネントの結合やテストに支障

課題①への取り組み: Software Engineer in Test (SET)

<http://googletesting.blogspot.jp/2010/03/google-is-hiring-sets.html>

役割: 自動化の専門部隊
テスト自動化方針に則った自動化

課題①への取り組み: Software Engineer in Test (SET)

<http://googletesting.blogspot.jp/2010/03/google-is-hiring-sets.html>

役割: 自動化の専門部隊

テスト自動化方針に則った自動化

開発チーム

コンポーネント	開発者			Software Engineer in Test
Search		...		
Indexer		...		
Storage		...		
Admin		...		

機能の開発工程

テスト自動化方針



課題①への取り組み: Software Engineer in Test (SET)

<http://googletesting.blogspot.jp/2010/03/google-is-hiring-sets.html>

役割: 自動化の専門部隊

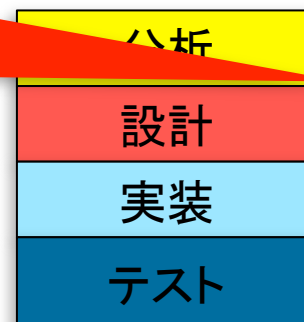
テスト自動化方針に則った自動化

開発チーム	コンポーネント	開発者		Software Engineer in Test
	Search		...	
	Indexer		...	
	Storage		...	
	Admin		...	

自動化の観点から
分析や設計に
積極的に関わる

楽天 Rakuten

機能の開発工程



テスト自動化方針

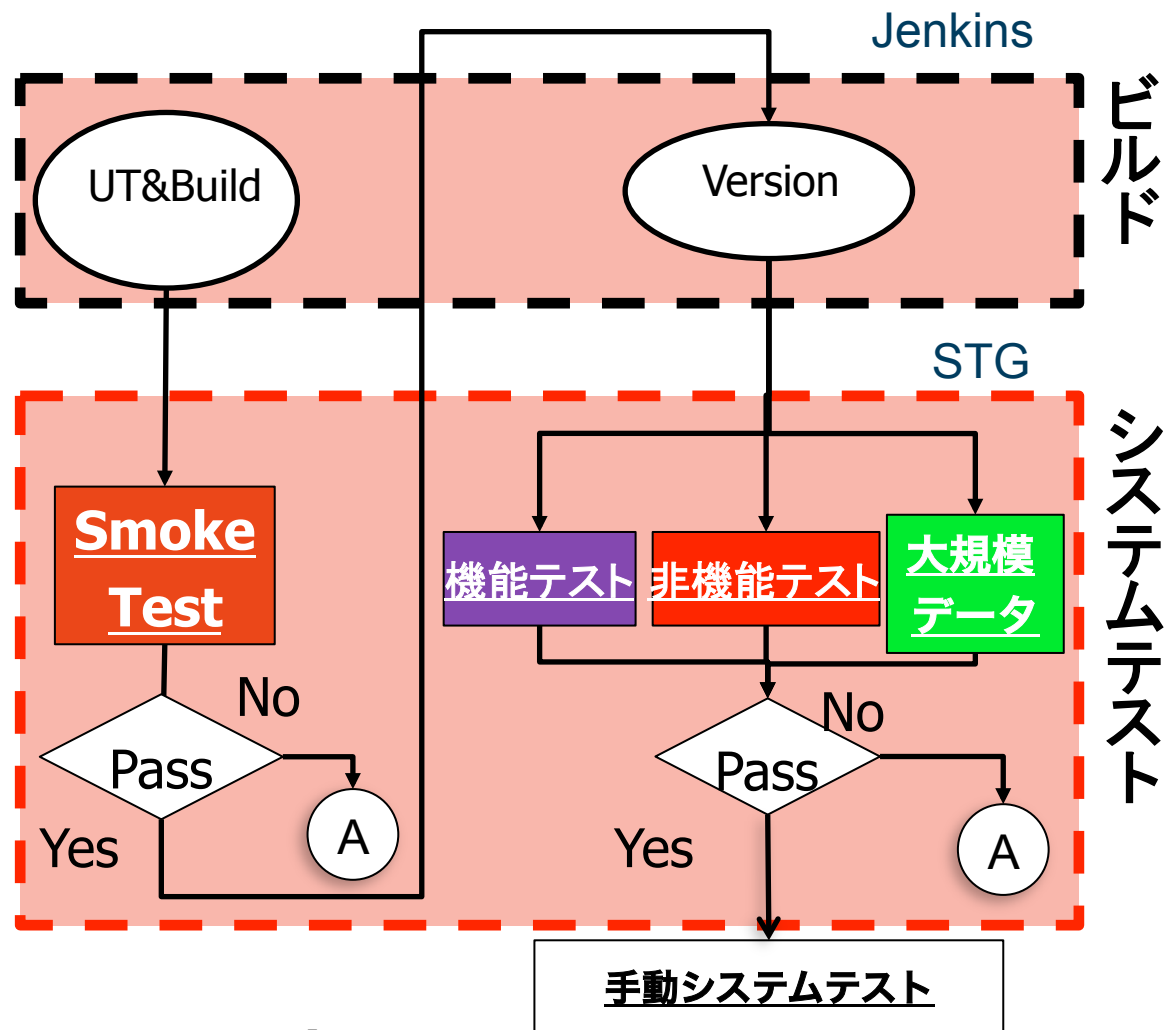


課題①への取り組み: :スモークテスト

- 最低限の結合を保証するスモークテストを自動化

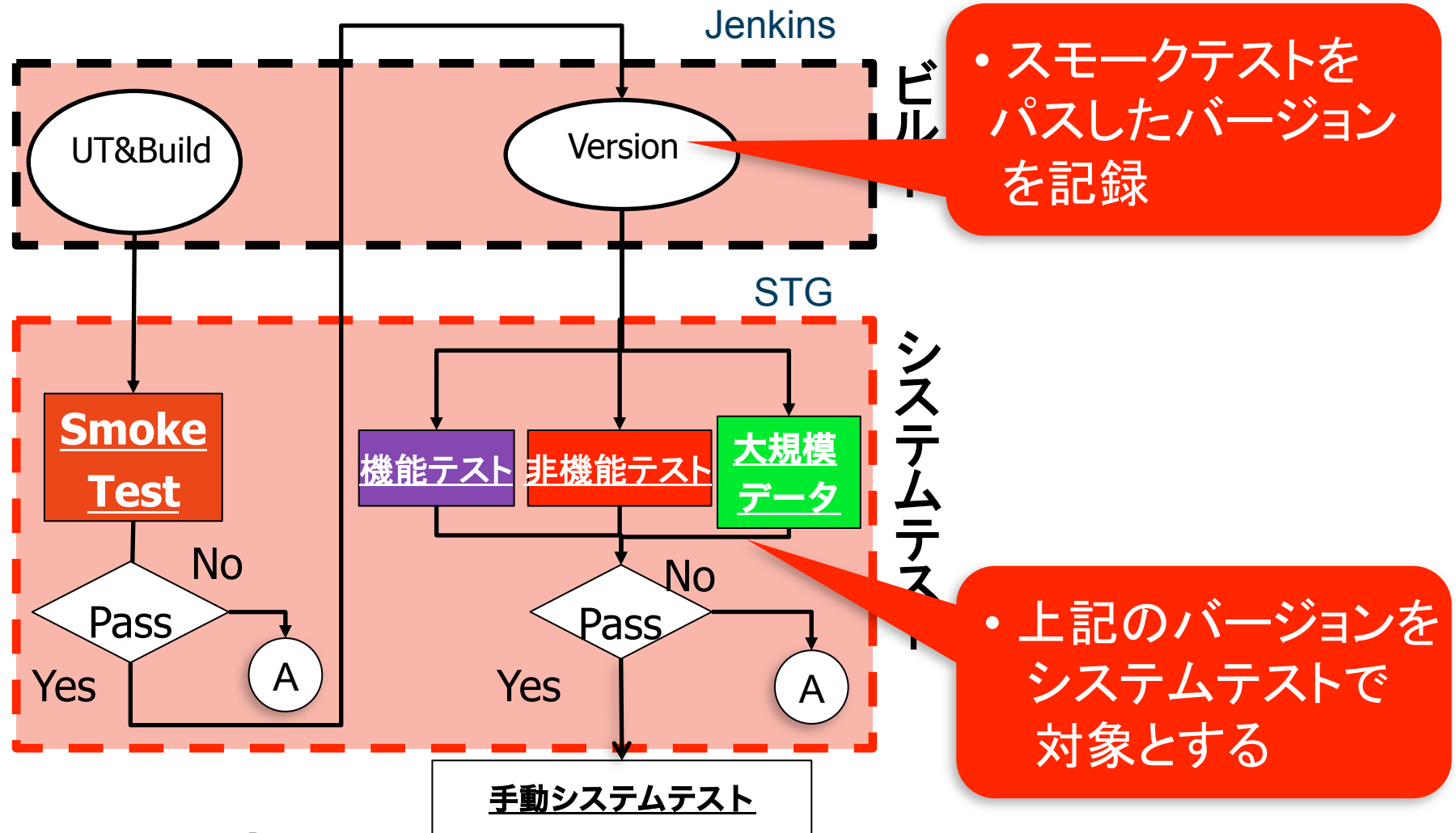
課題①への取り組み: :スモークテスト

- 最低限の結合を保証するスモークテストを自動化



課題①への取り組み: :スモークテスト

- 最低限の結合を保証するスモークテストを自動化



自動化された継続的システムテストの課題②: テストにまつわるコスト

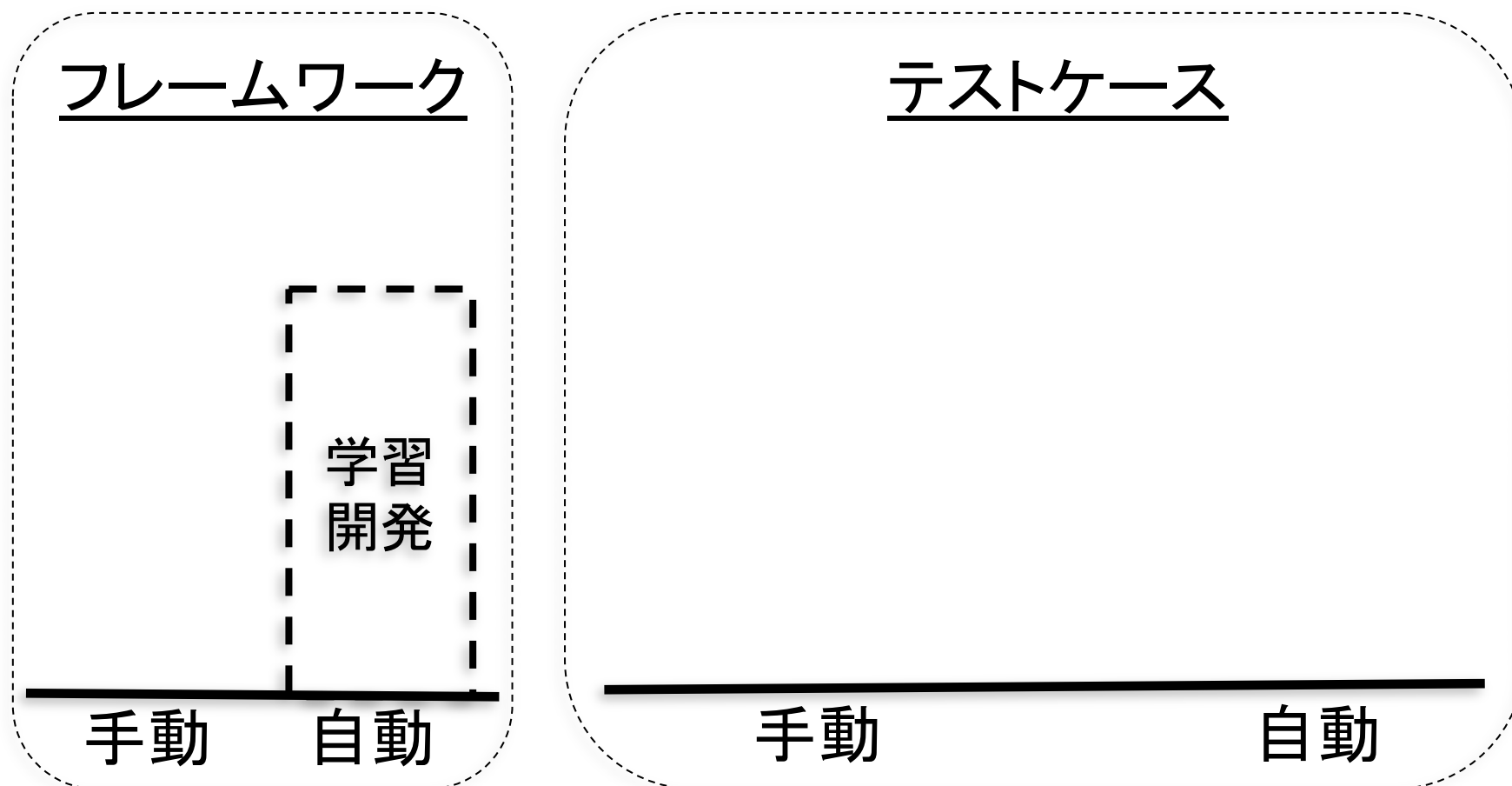
フレームワーク

手動 自動

テストケース

手動 自動

自動化された継続的システムテストの課題②: テストにまつわるコスト



自動化された継続的システムテストの課題②: テストにまつわるコスト

フレームワーク

フレームワークの
学習や開発は
必要な投資

学習
開発

手動

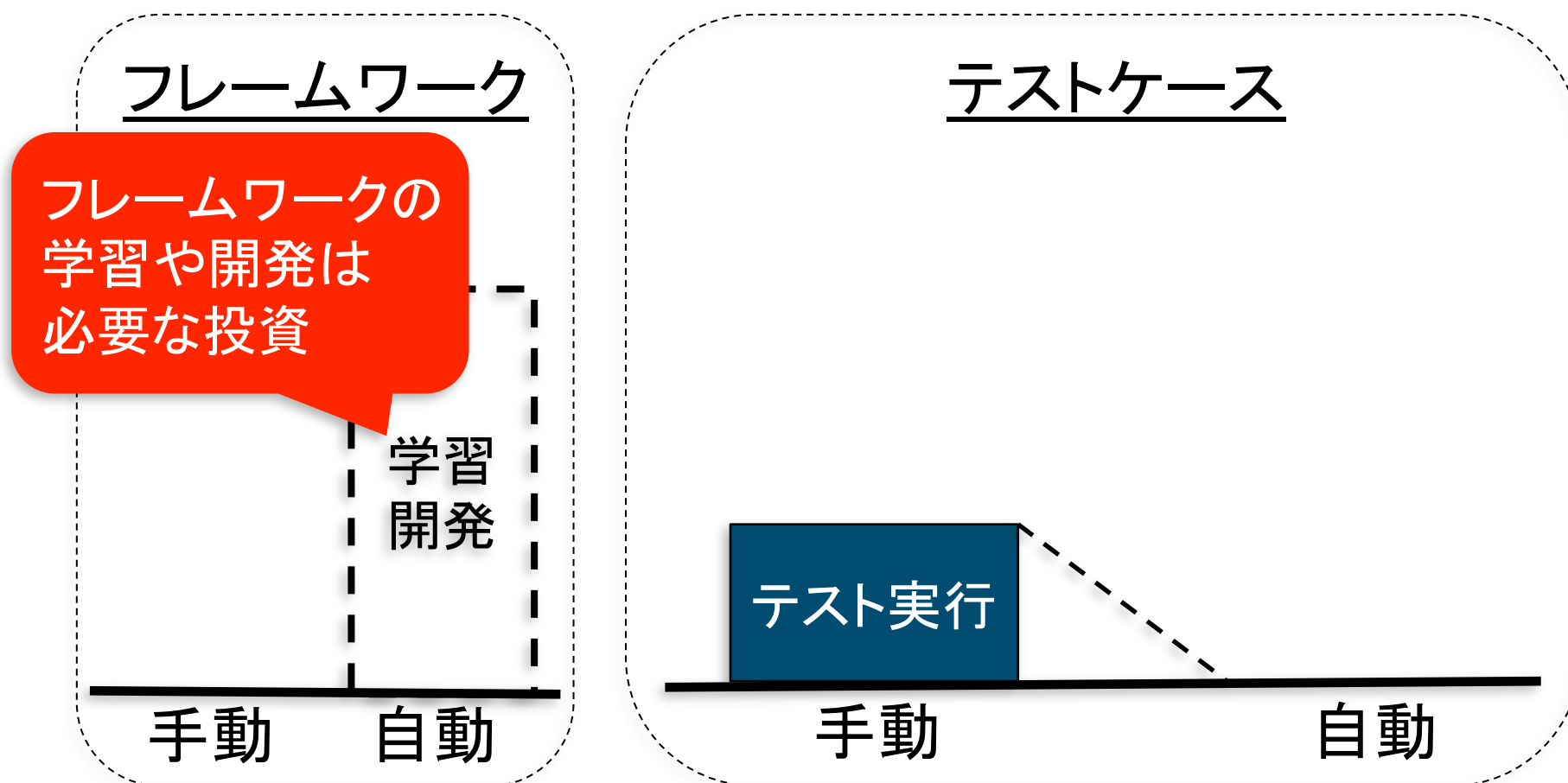
自動

テストケース

手動

自動

自動化された継続的システムテストの課題②: テストにまつわるコスト



自動化された継続的システムテストの課題②: テストにまつわるコスト

フレームワーク

フレームワークの
学習や開発は
必要な投資

学習
開発

手動

自動

テストケース

テスト実行

手動

自動

実行コストは自動化で0に

自動化された継続的システムテストの課題②: テストにまつわるコスト

フレームワーク

フレームワークの
学習や開発は
必要な投資

学習
開発

手動

自動

テストケース

テスト実行

テスト実装

手動

自動

実行コストは自動化で0に

自動化された継続的システムテストの課題②: テストにまつわるコスト

フレームワーク

フレームワークの
学習や開発は
必要な投資

学習
開発

手動

自動

テストケース

テスト保守
原因調査
テスト実行

テスト保守

原因調査

テスト実装

手動

自動

実行コストは自動化で0に

自動化された継続的システムテストの課題②: テストにまつわるコスト

テスト実装や、原因調査、保守の
コストを抑えるための工夫が重要

フレームワーク

フレームワークの
学習や開発は
必要な投資

学習
開発

手動

自動

テストケース

テスト保守
原因調査
テスト実行

テスト保守

原因調査

テスト実装

手動

自動

実行コストは自動化で0に

課題②への取り組み: ドメイン固有テスト言語

検索結果のテストケースを表現する
ドメイン固有テスト言語(DSTL)を定義
→ 実装・保守コストを低減

```
{
  "$name": "mv-000-int",
  "$request": {
    "$query": "key_field:*",
    "$filter": null,
    "$facet": "field:mv_int type:value.count limit",
    "$sortby": null,
    "$otherparams": "hits=20"
  },
  "$expected": {
    "$httpstatuscode": 200, "$apistatuscode": 0, "$nu
    "$docs": [ {"key_field": "mv-001"}, {"key_field"
    "$facetcounts": {
      "facet_queries": {},
      "facet_fields": {"mv_int": ["200", 2, "300", 4
      "facet_dates": {},
      "facet_ranges": {}
    }
  }
},
```

テストツール: Ngauto

ドメイン固有テスト
言語で記述された
シナリオ

- アクション1
デプロイ
- アクション2
データ投入
- アクション3
検索クエリ



Ngauto
(内製の
テストツール)

テストツールの中で
アクションに対する実際の
振る舞いを定義

“検索クエリ” アクションの例)

```
Report report  
    = issueSapiQuery(testId,query);
```

...

```
int expectedHttpStatus  
    = getExpectedHttpStatus(testId);  
int actualHttpStatus  
    = report.getActualHttpStatus()  
TestCase.assertEquals(  
    message,  
    expectedHttpStatus,  
    actualHttpStatus );
```

...

課題②への取り組み: トレーサビリティ

JUnitベースのテストツール

- JenkinsやIDEの画面から**バグの追跡が可能**
- 原因調査コストの低減

テストケースが失敗した場合の画面

Failed

com.rakuten.gsp.ngauto.tests.relevancy.abtest.updatemode.RelevancyEvaluateService.executeTestProcedure[0]

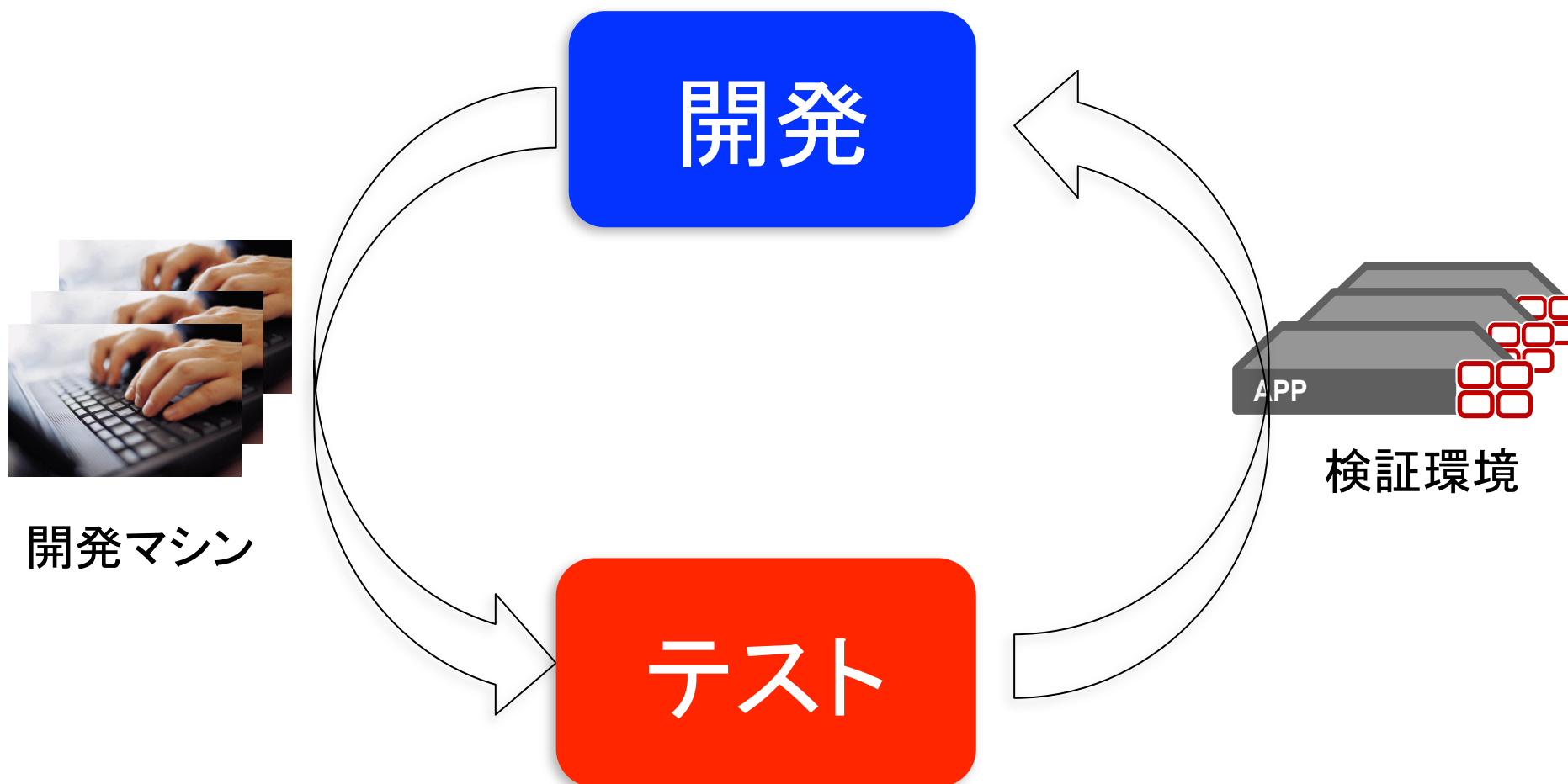
Failing for the past 9 builds (Since #41)

Took 7.3 sec.
[add description](#)

Error Message

```
The NDCG changed. Test ID: 0      Query = http://dev-agspapi004z:8082/search-api/mona_search/search?
sid=test&authkey=test&queryversion=2.0&query=cf_text:"ダウンロード"&filter=query_id:1&hits=30&sortby=relevancy      Expected numDocs = 20
Expected documents = [{"docid":"00000295653-00010010879-1-query-1","relevancy_score":"2"}, {"docid":"00000227676-00010016426-1-query-
1","relevancy_score":"3"}, {"docid":"00000227676-00010017239-1-query-1","relevancy_score":"3"}, {"docid":"00000295114-00010333398-1-query-
1","relevancy_score":"3"}, {"docid":"00000259391-00010002967-1-query-1","relevancy_score":"3"}, {"docid":"00000217759-00010012880-1-query-
1","relevancy_score":"3"}, {"docid":"00000206544-00010012478-1-query-1","relevancy_score":"3"}, {"docid":"00000205968-00010005929-1-query-
1","relevancy_score":"3"}, {"docid":"00000214151-00010007451-1-query-1","relevancy_score":"3"}, {"docid":"00000221501-00010000056-1-query-
1","relevancy_score":"3"}, {"docid":"00000258910-00010015762-1-query-1","relevancy_score":"3"}, {"docid":"00000221501-00010000400-1-query-
1","relevancy_score":"3"}, {"docid":"00000234015-00010001265-1-query-1","relevancy_score":"3"}, {"docid":"00000282518-00010001929-1-query-
1","relevancy_score":"3"}, {"docid":"00000202702-00010045978-1-query-1","relevancy_score":"3"}, {"docid":"00000221501-00010299711-1-query-
1","relevancy_score":"3"}, {"docid":"00000221501-00010712561-1-query-1","relevancy_score":"3"}, {"docid":"00000225360-00010005561-1-query-
1","relevancy_score":"3"}, {"docid":"00000205968-00010005630-1-query-1","relevancy_score":"3"}, {"docid":"00000230900-00010002959-1-query-
1","relevancy_score":"3"}]      Actual numDocs = 20      Actual documents = [{"docid":"00000259391-00010002967-1-query-
1","relevancy_score":"3"}, {"docid":"00000221501-00010000056-1-query-1","relevancy_score":"3"}, {"docid":"00000225360-00010005561-1-query-
1","relevancy_score":"3"}, {"docid":"00000230900-00010002959-1-query-1","relevancy_score":"3"}, {"docid":"00000205968-00010005630-1-query-
1","relevancy_score":"3"}, {"docid":"00000206544-00010012478-1-query-1","relevancy_score":"3"}, {"docid":"00000221501-00010000400-1-query-
1","relevancy_score":"3"}, {"docid":"00000221501-00010712561-1-query-1","relevancy_score":"3"}, {"docid":"00000205968-00010005929-1-query-
1","relevancy_score":"3"}, {"docid":"00000214151-00010007451-1-query-1","relevancy_score":"3"}, {"docid":"00000258910-00010015762-1-query-
1","relevancy_score":"3"}, {"docid":"00000202702-00010045978-1-query-1","relevancy_score":"3"}, {"docid":"00000217759-00010012880-1-query-
1","relevancy_score":"3"}, {"docid":"00000221501-00010299711-1-query-1","relevancy_score":"3"}, {"docid":"00000234015-00010001265-1-query-
1","relevancy_score":"3"}, {"docid":"00000227676-00010016426-1-query-
1","relevancy_score":"3"}, {"docid":"00000227676-00010017239-1-query-1","relevancy_score":"3"}, {"docid":"00000295114-00010333398-1-query-
1","relevancy_score":"3"}, {"docid":"00000295653-00010010879-1-query-1","relevancy_score":"2"}]      Previous NDCG = 0.963034      Current
NDCG = 1.000000
```

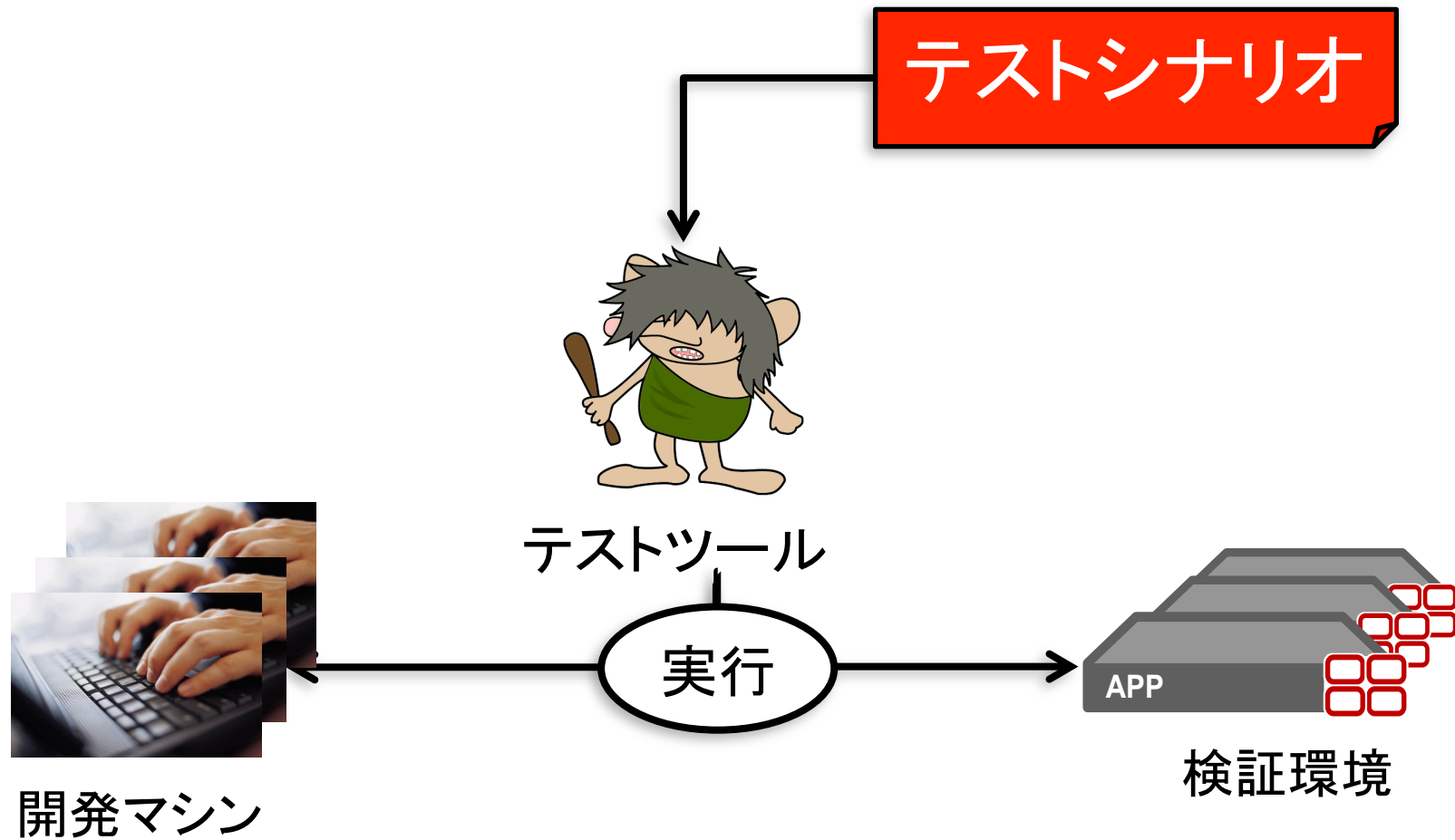
自動化された継続的システムテストの課題③: テストと開発のサイクル



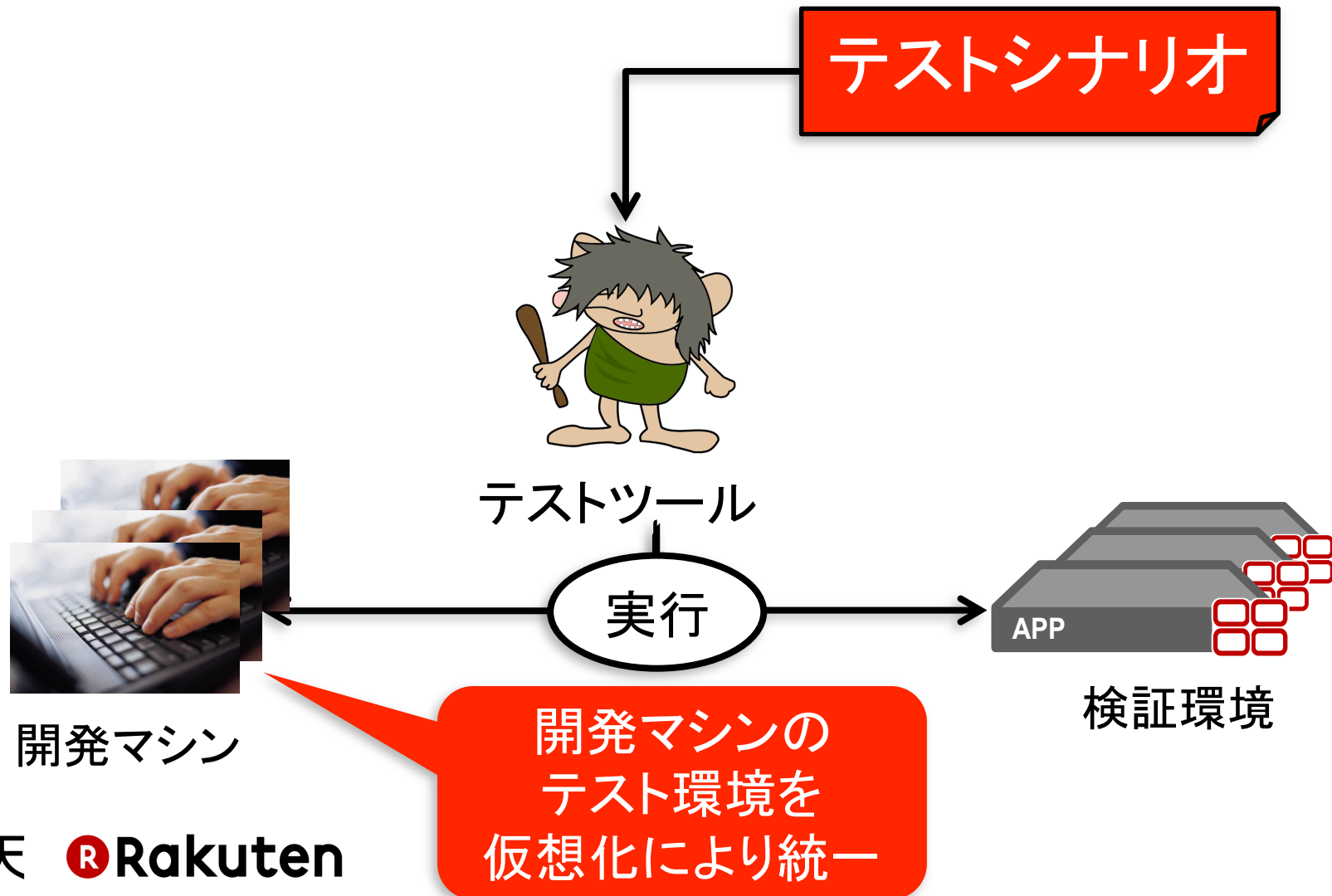
自動化された継続的システムテストの課題③: テストと開発のサイクル



課題③への取り組み:シナリオ実行の透過性

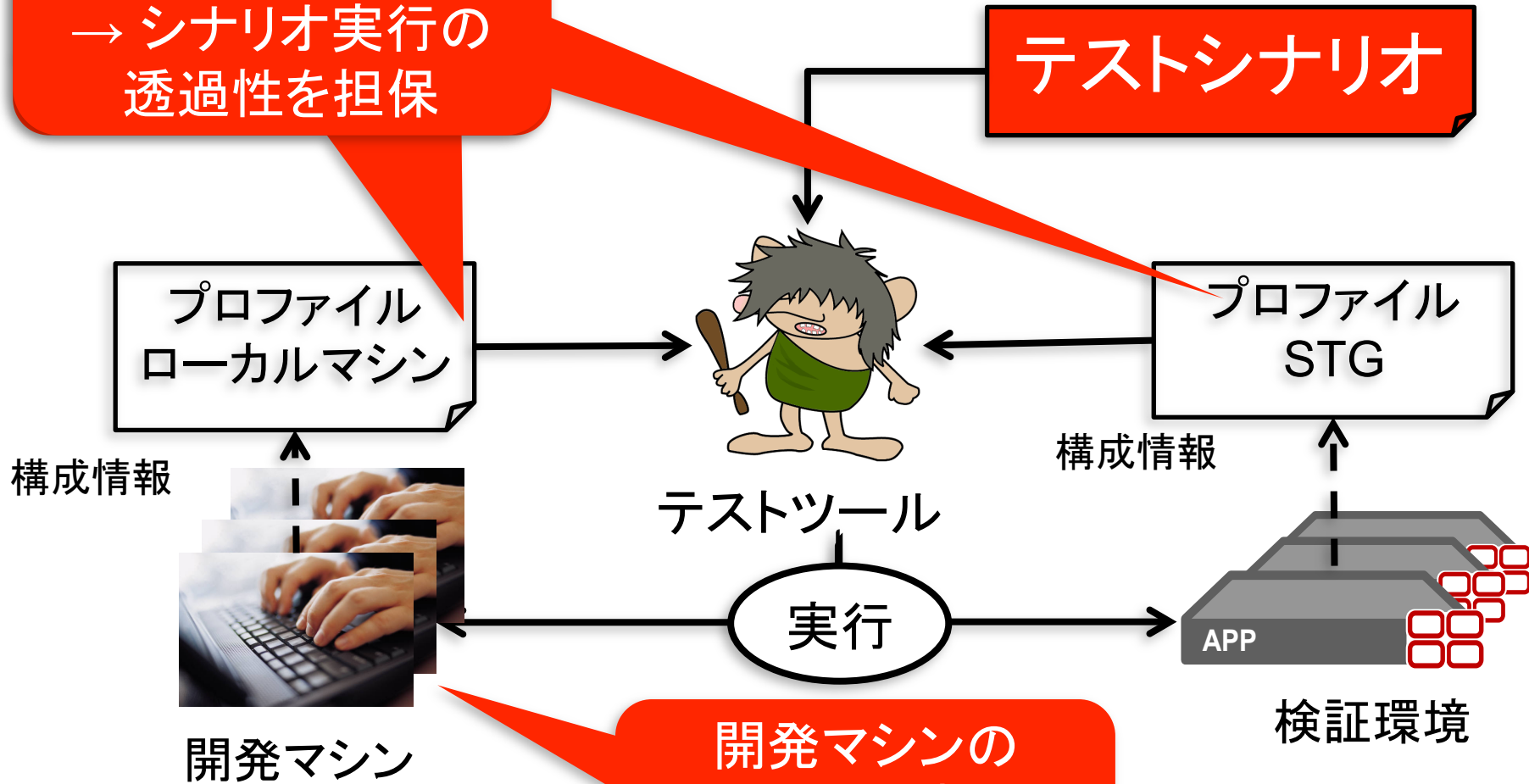


課題③への取り組み:シナリオ実行の透過性



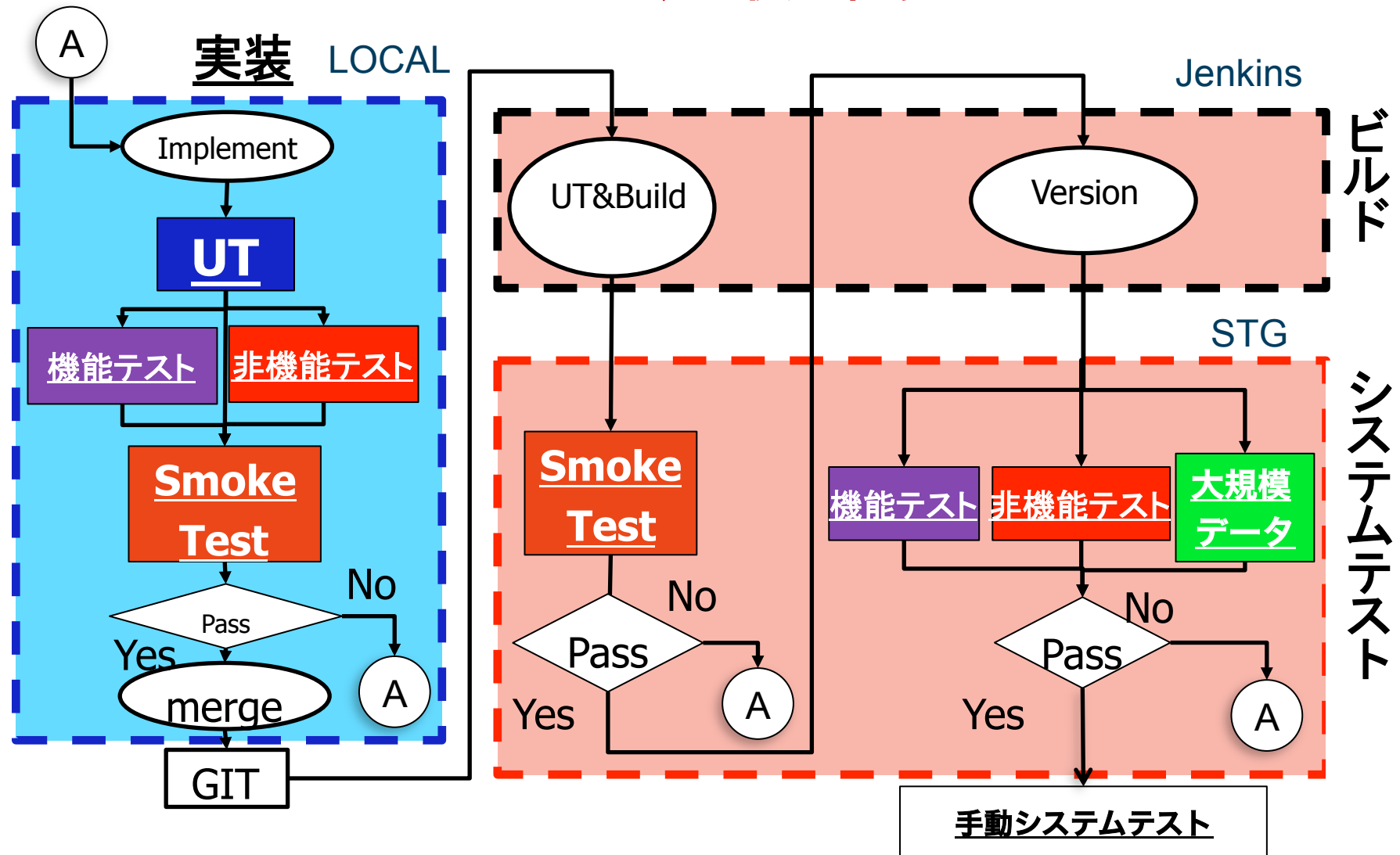
課題③への取り組み:シナリオ実行の透過性

環境の構成情報を
テストシナリオから分離
→ シナリオ実行の
透過性を担保

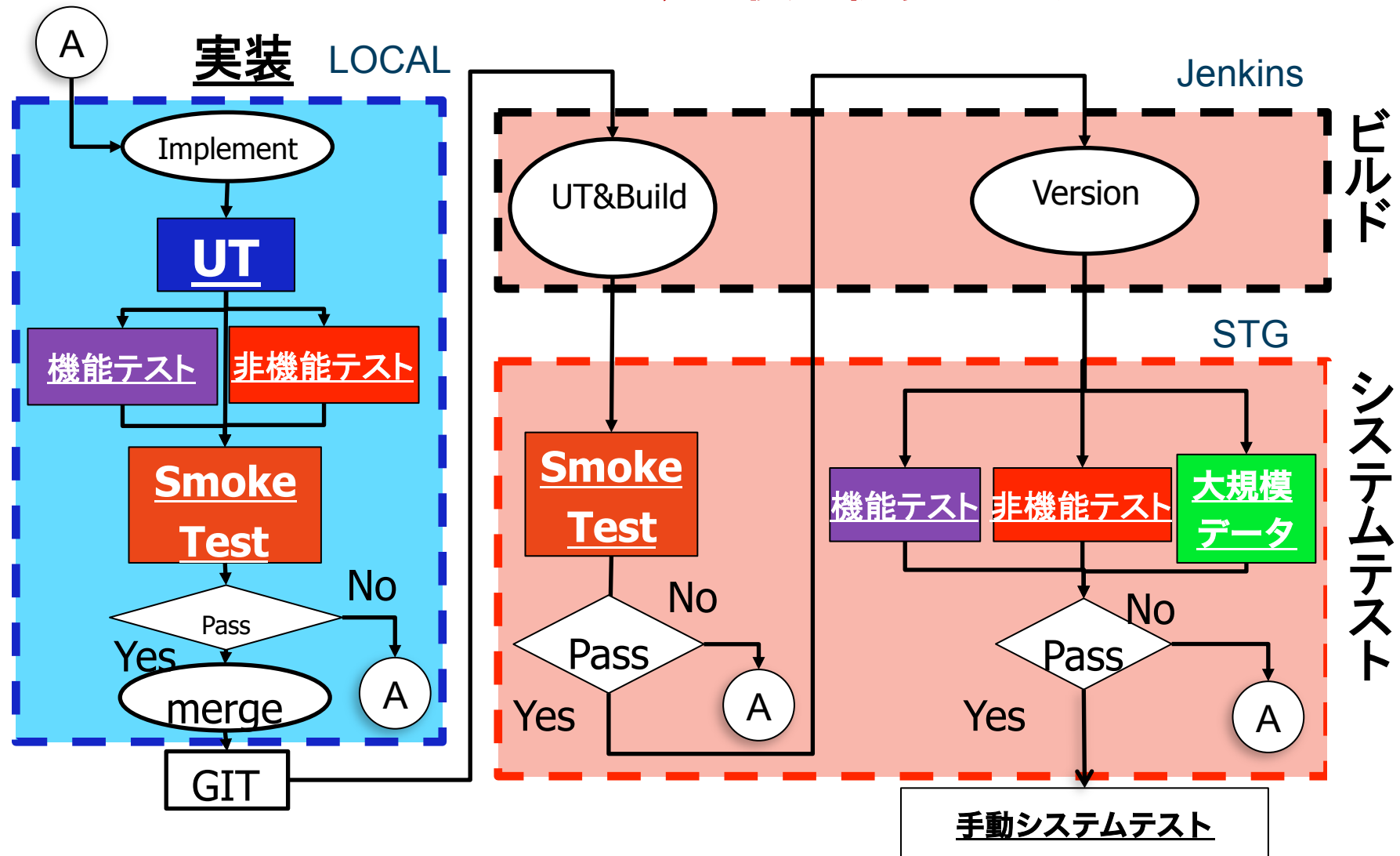


開発 サイクル

システムテスト自動化後の開発サイクル



システムテスト自動化後の開発サイクル



自動化されたすべてのテストを毎日、継続的に実行

楽天 Rakuten → リードタイム、バグ修正日数が最短1日に

評價

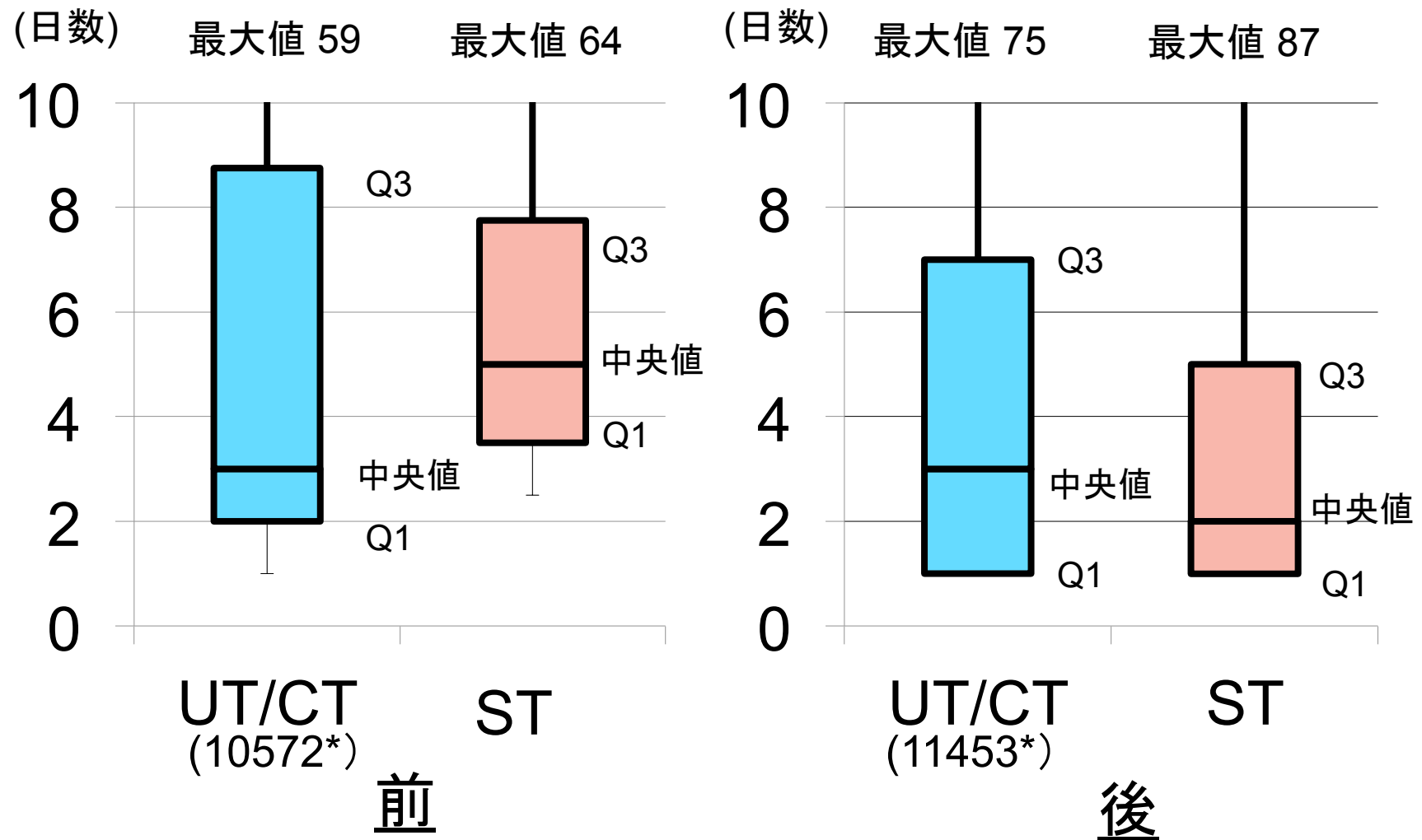
システムテスト自動化の効果の評価方法

- 自動化前後のバグ修正日数の比較
- 継続的システムテスト適用後の累積バグ数

参考情報: 自動化前後のテストの比較

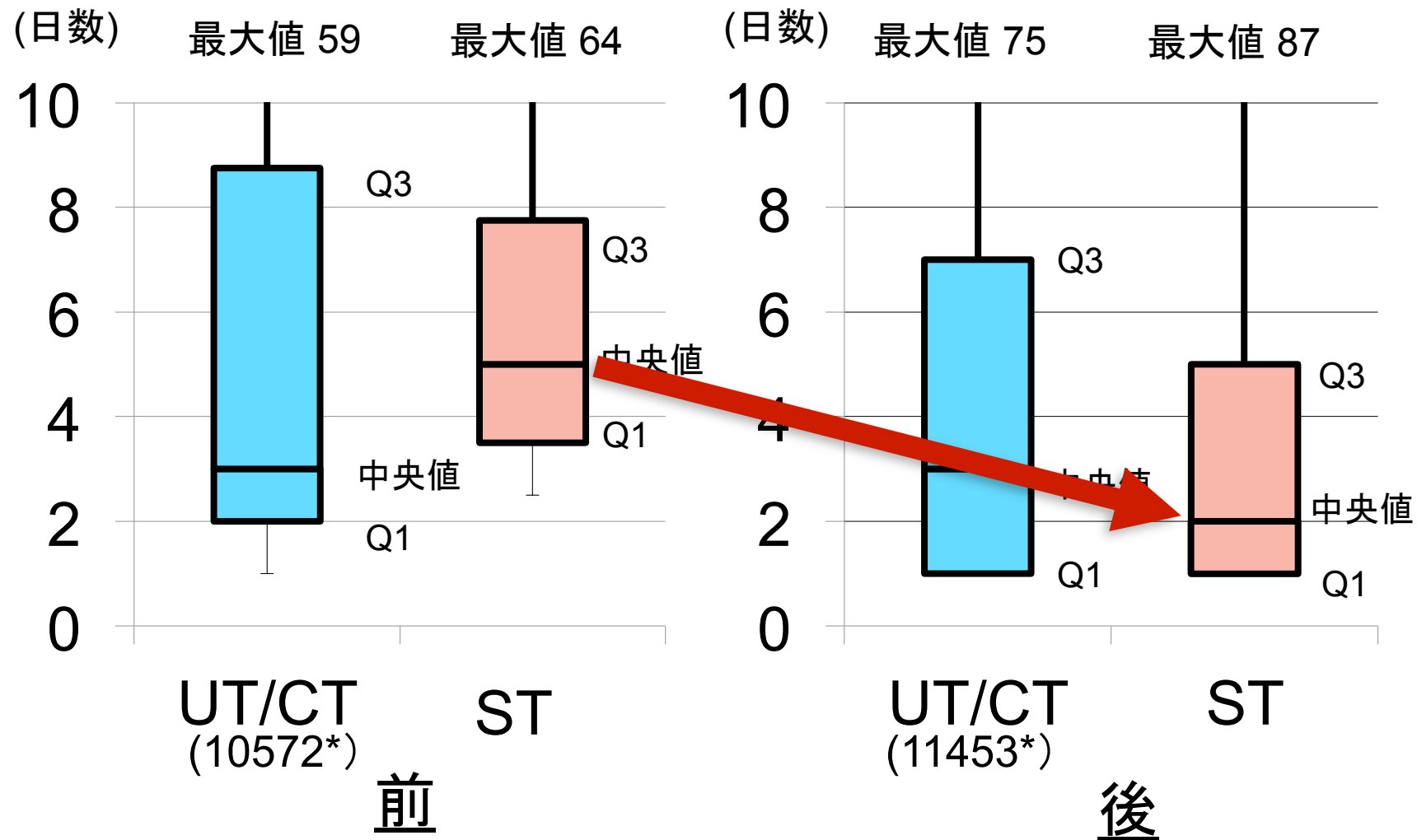
継続的システムテスト	前	後
自動化テストのカバレッジ	機能テストの一部 (604件)	機能テスト 非機能テスト 大規模テスト (2887件)
SET		○
スモークテスト	○	○
ドメイン固有テスト言語		○
バグのトレーサビリティ		△
テストシナリオの透過性		△

評価結果: バグ修正日数の比較



*テストメソッド数

評価結果: バグ修正日数の比較



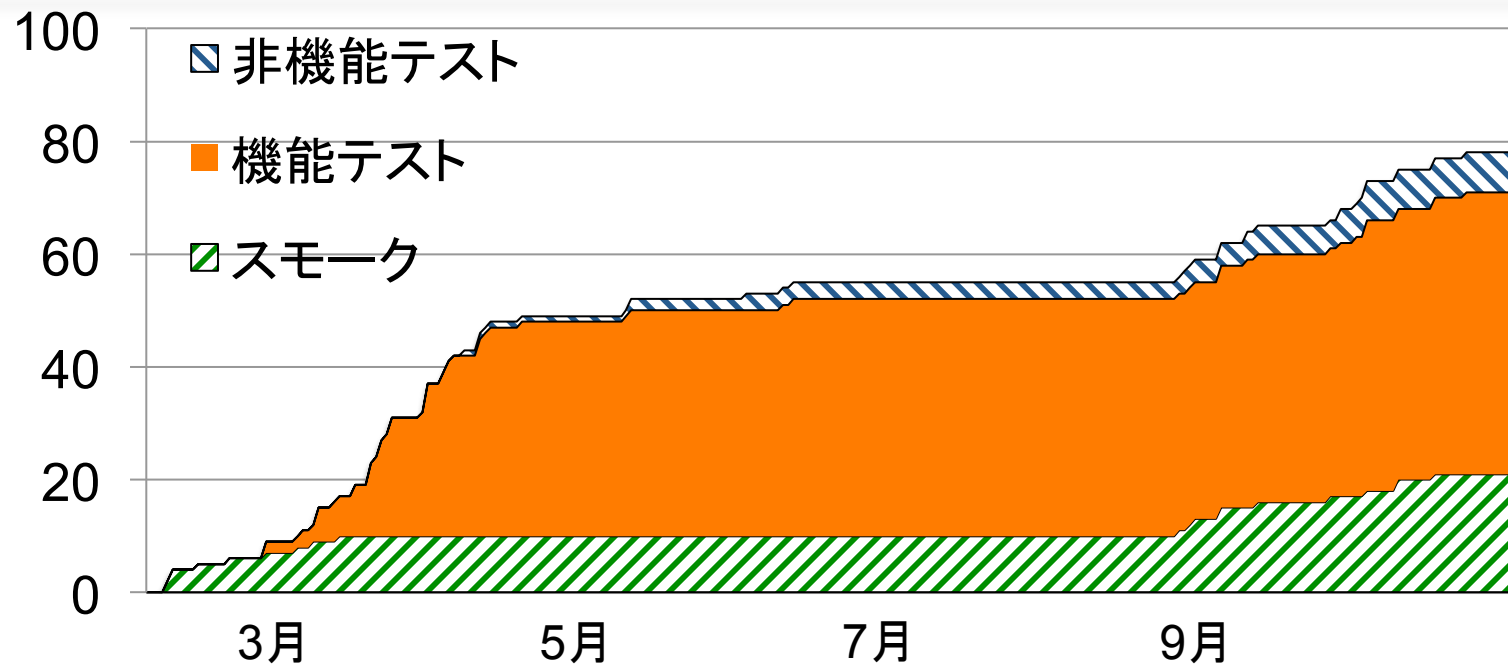
*テストメソッド数

楽[®]天 [®]Rakuten

システムテストで発見された
バグの修正日数が大幅に改善

自動化された継続的システムテスト適用後の累積バグ数

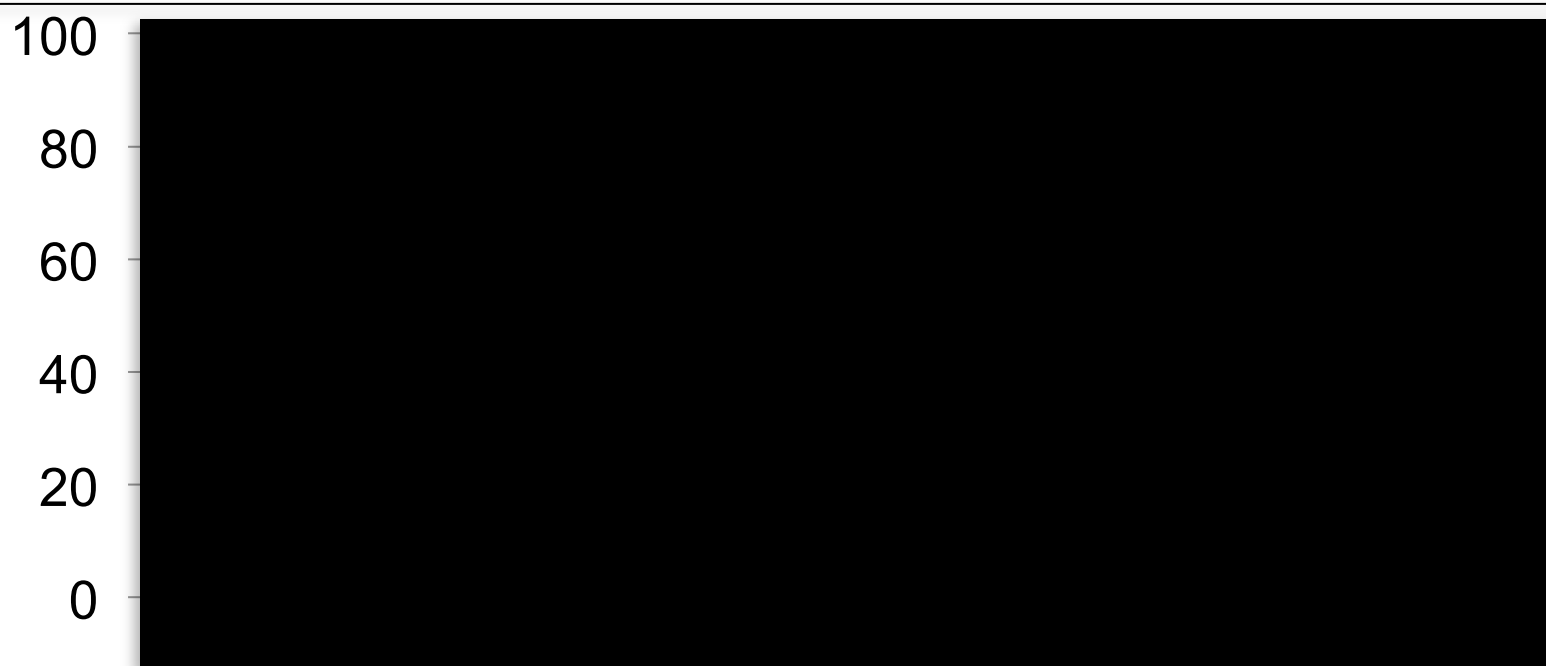
開発



テスト
自動化

自動化された継続的システムテスト適用後の累積バグ数

開発



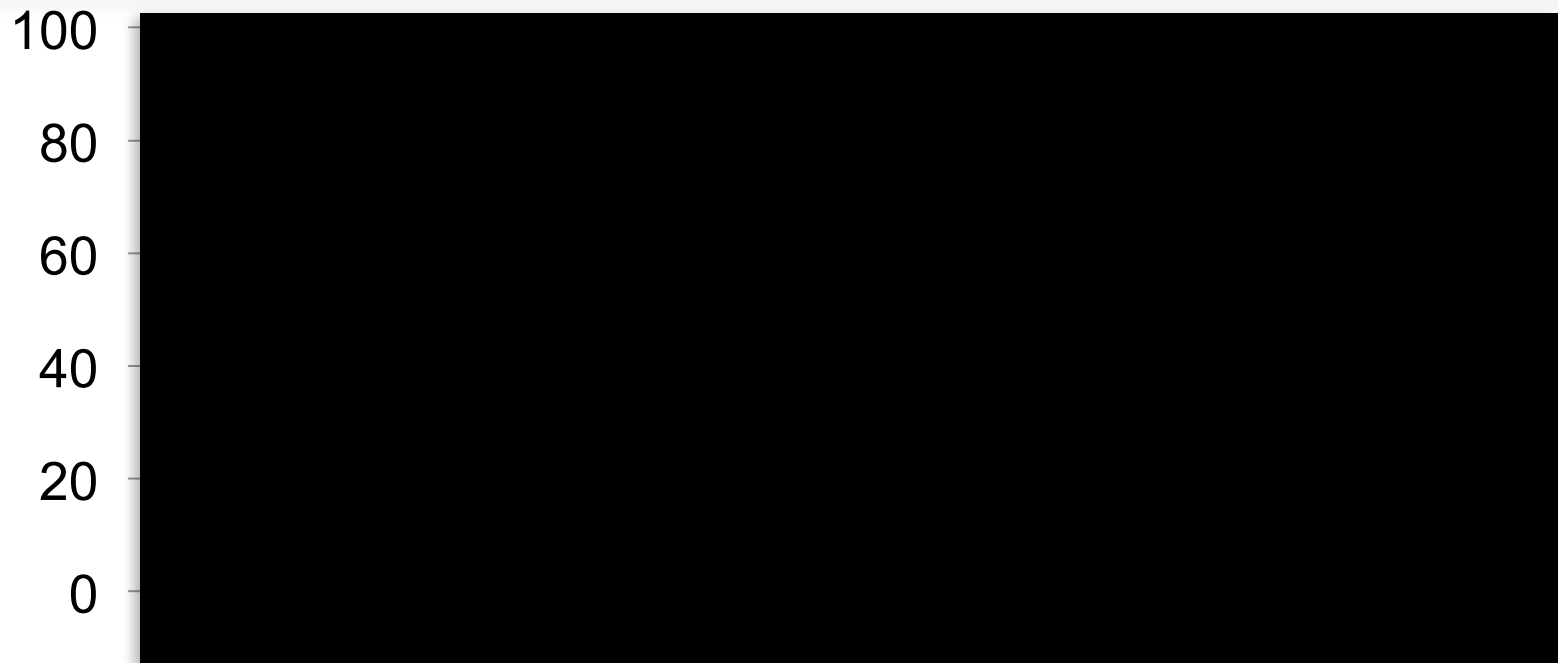
テスト
自動化

自動化された継続的システムテスト適用後の累積バグ数

開発

大きな
機能要件

受け入れ
テスト



テスト
自動化

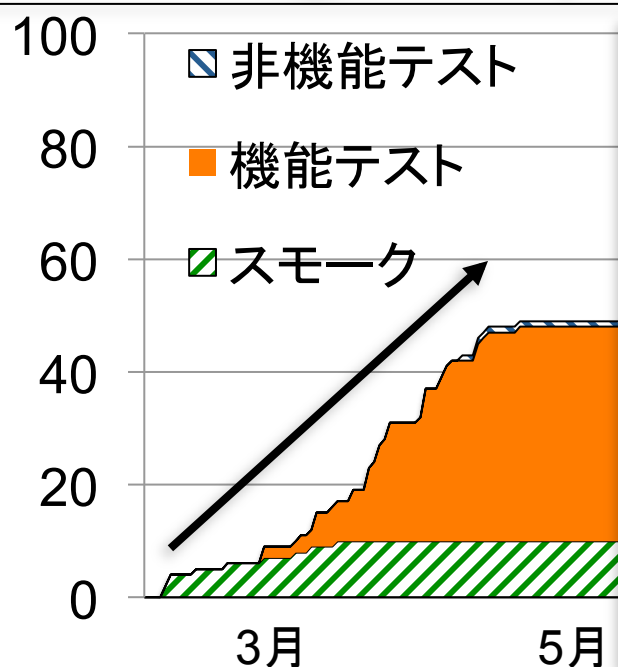
機能テスト
自動化完了

自動化された継続的システムテスト適用後の累積バグ数

開発

大きな
機能要件

受け入れ
テスト



テスト
自動化

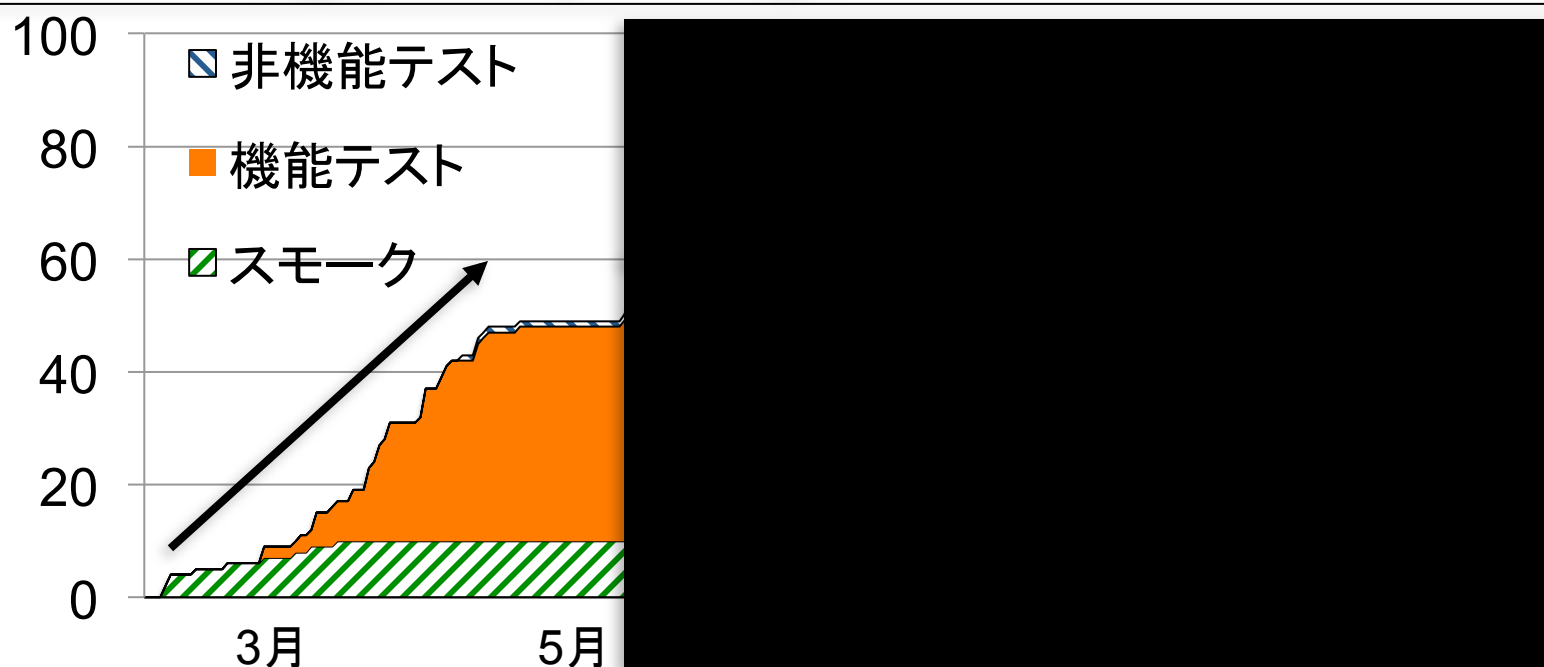
機能テスト
自動化完了

自動化された継続的システムテスト適用後の累積バグ数

開発

大きな
機能要件

受け入れ 断続的な
テスト 小さな要件



テスト
自動化

機能テスト
自動化完了

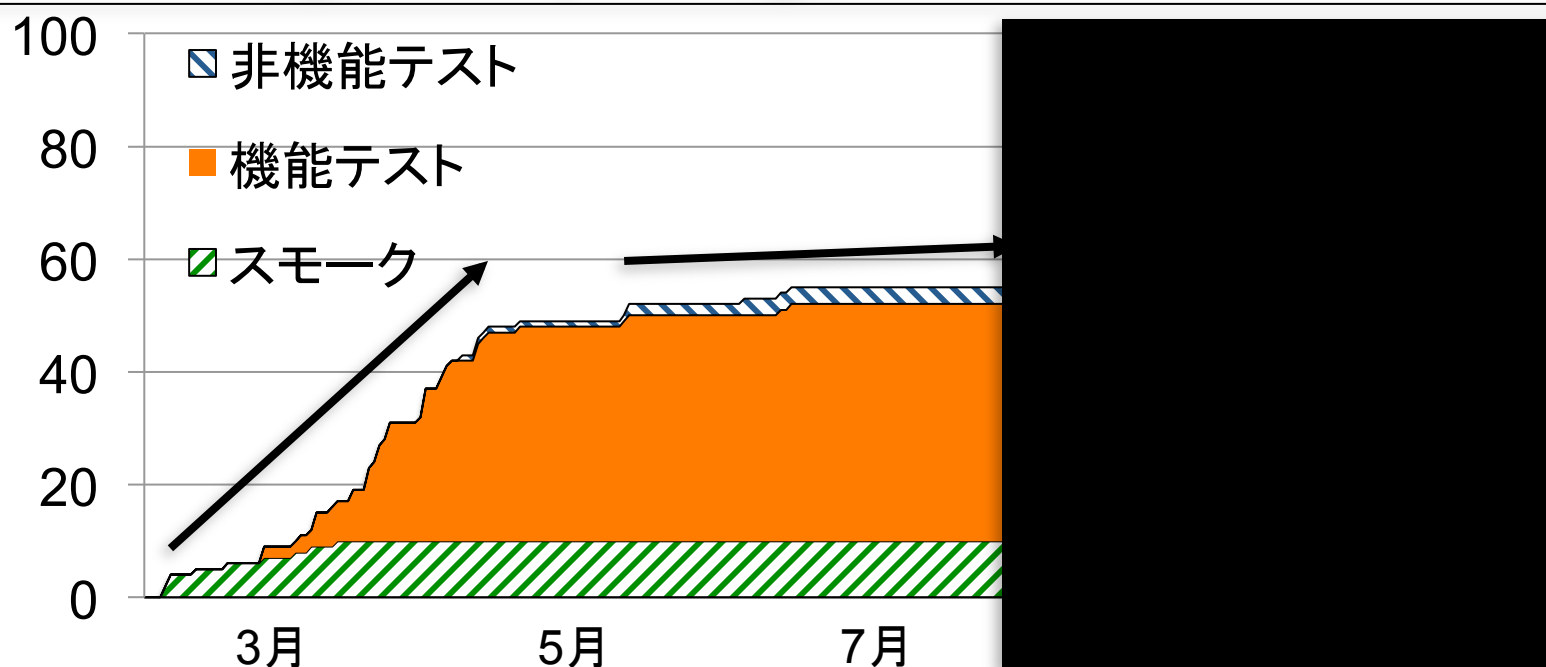
シナリオの
透過性

自動化された継続的システムテスト適用後の累積バグ数

開発

大きな
機能要件

受け入れ
テスト 断続的な
小さな要件

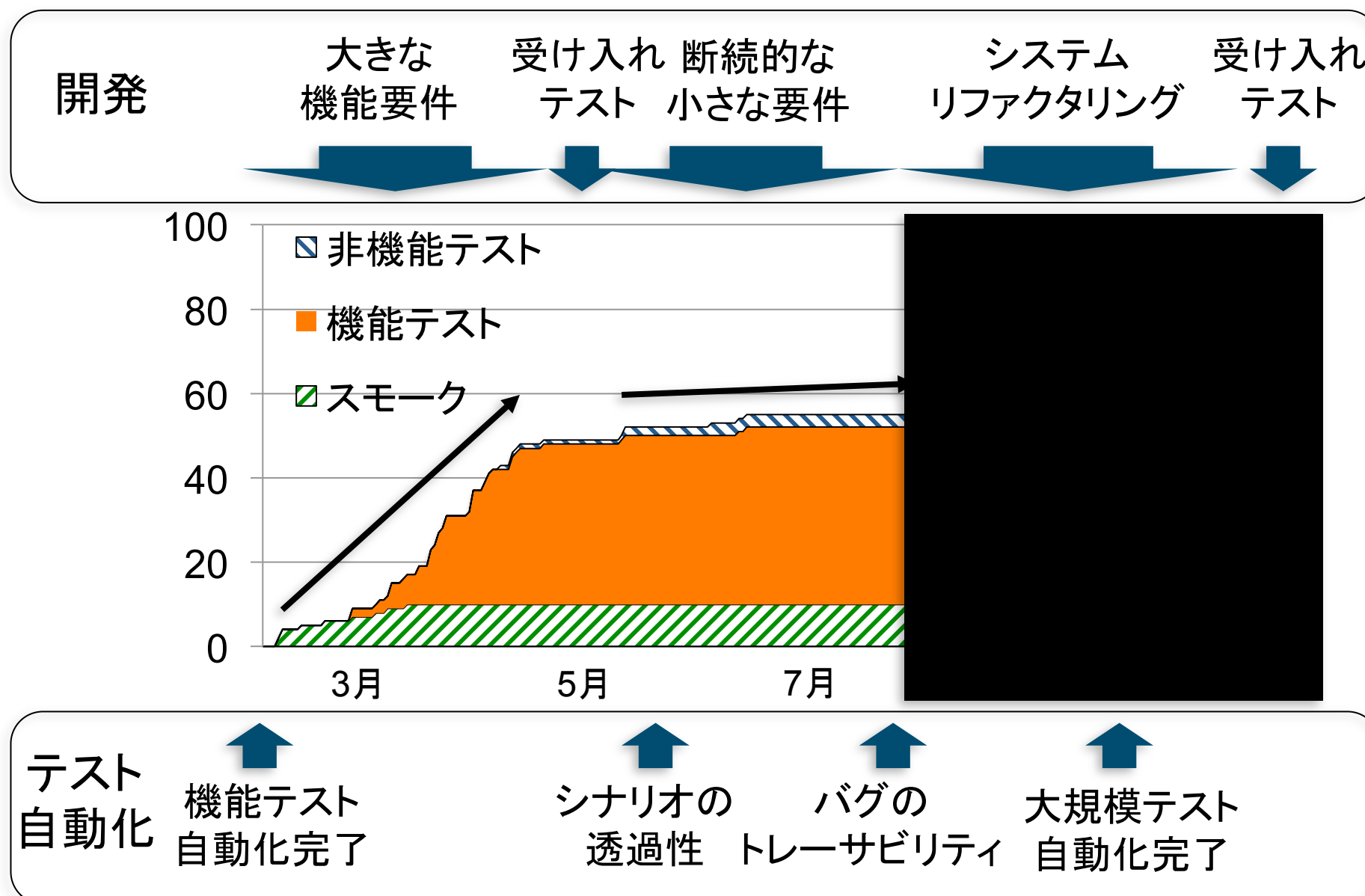


テスト
自動化

機能テスト
自動化完了

シナリオの
透過性

自動化された継続的システムテスト適用後の累積バグ数



自動化された継続的システムテスト適用後の累積バグ数

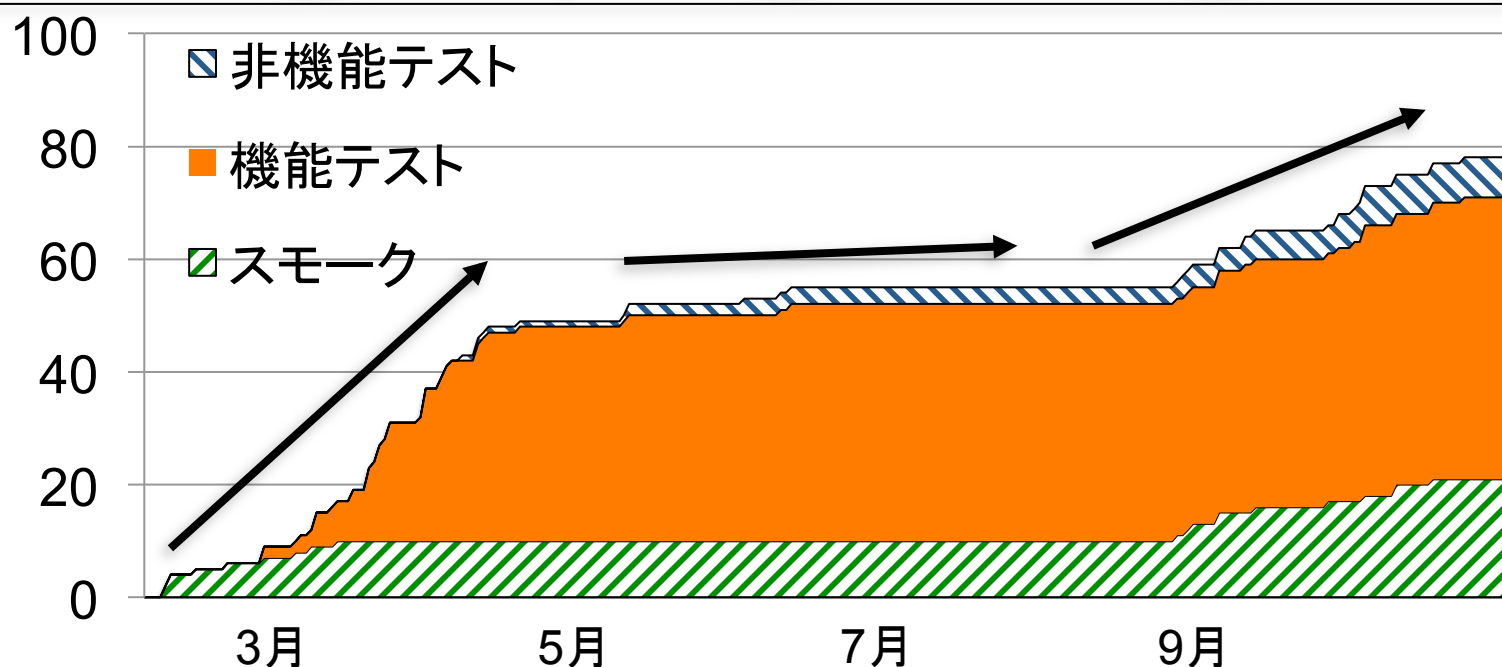
開発

大きな
機能要件

受け入れ
テスト 断続的な
小さな要件

システム
リファクタリング

受け入れ
テスト



テスト
自動化

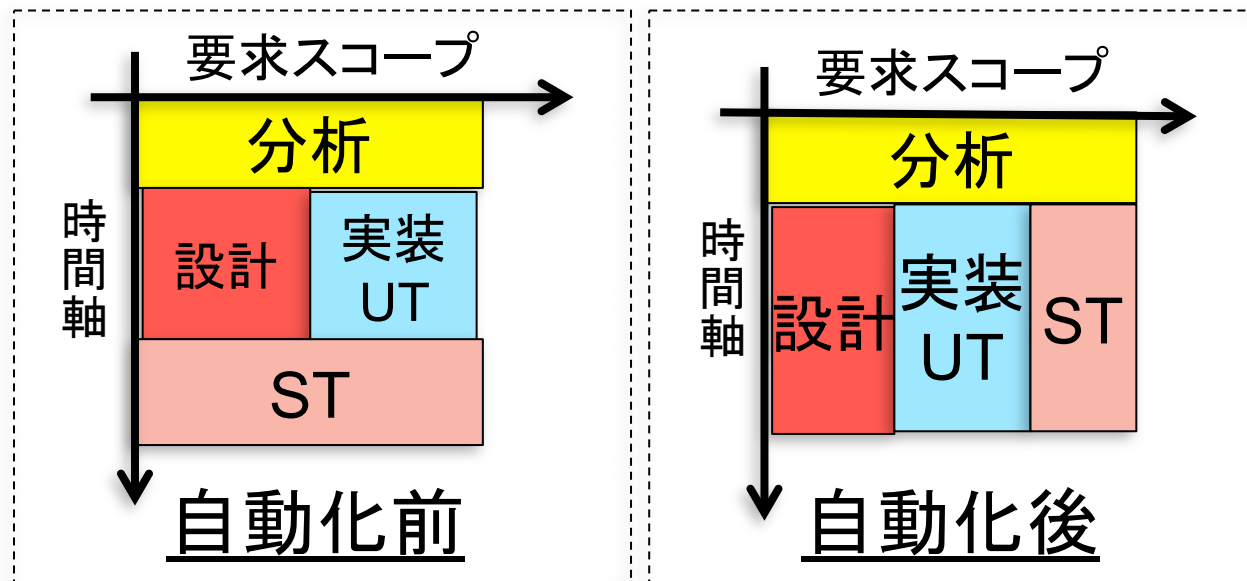
機能テスト
自動化完了

シナリオの
透過性

バグの
トレーサビリティ

大規模テスト
自動化完了

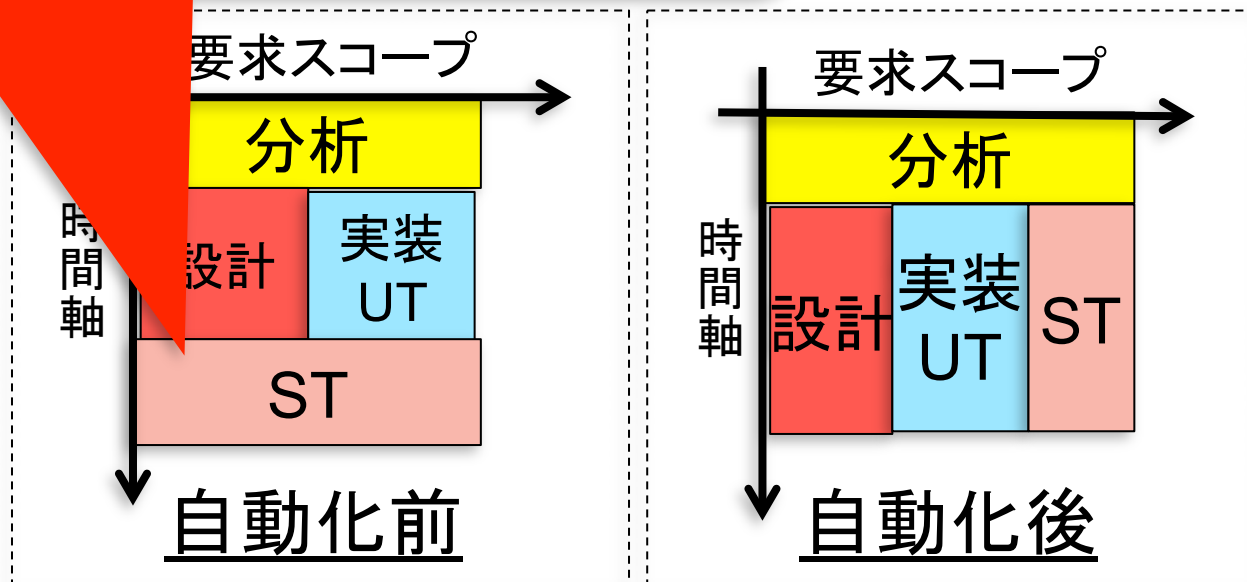
継続的システムテストについてまとめ



継続的システムテストについてまとめ

自動化前:

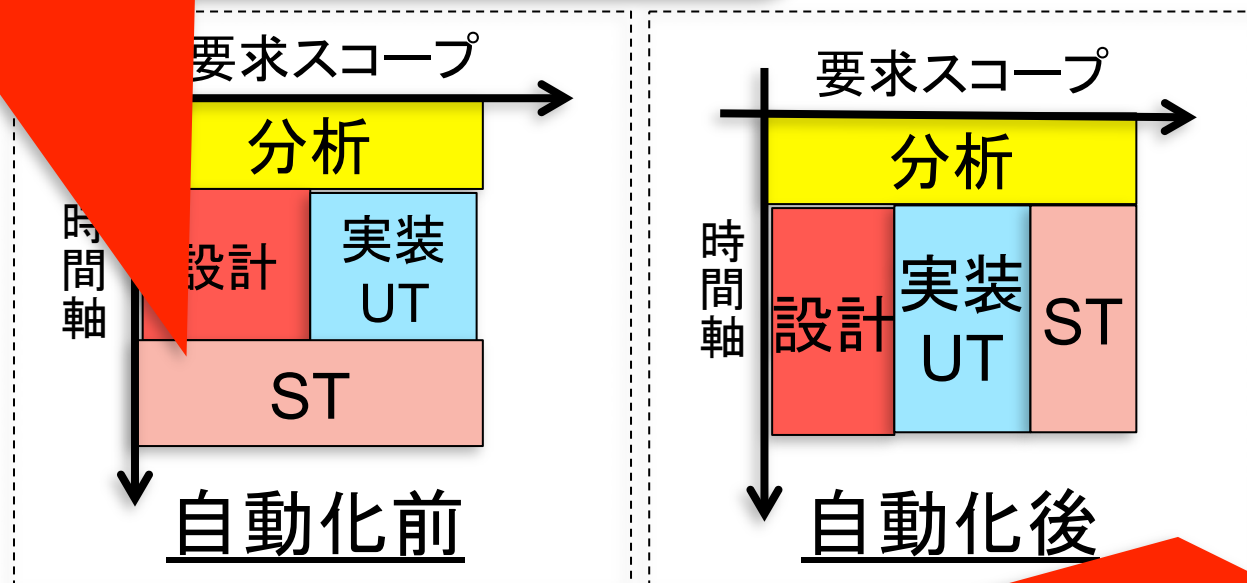
リリース直前にバグが見つかり
てんやわんやしていた



継続的システムテストについてまとめ

自動化前:

リリース直前にバグが見つかり
てんやわんやしていた

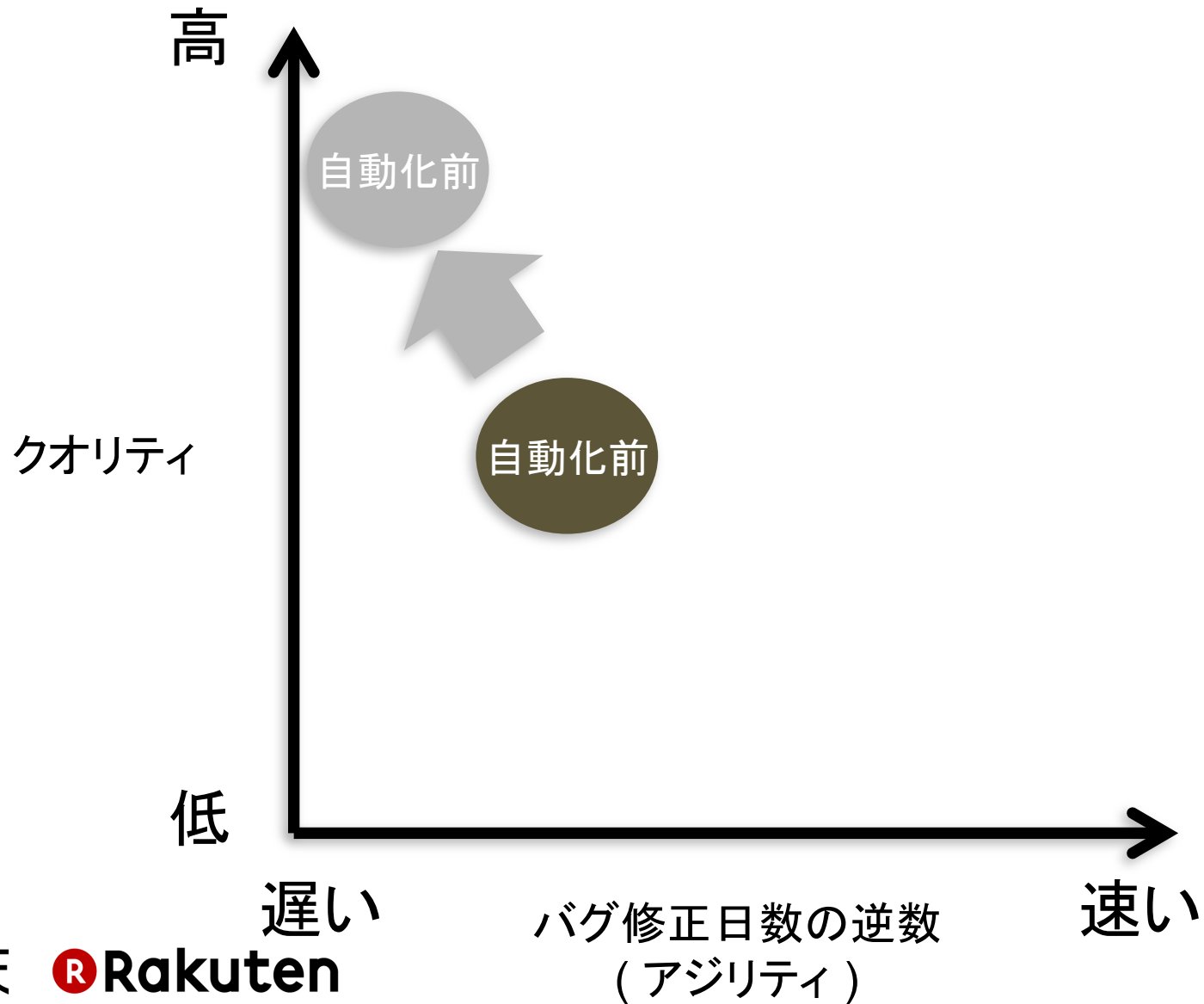


自動化後:

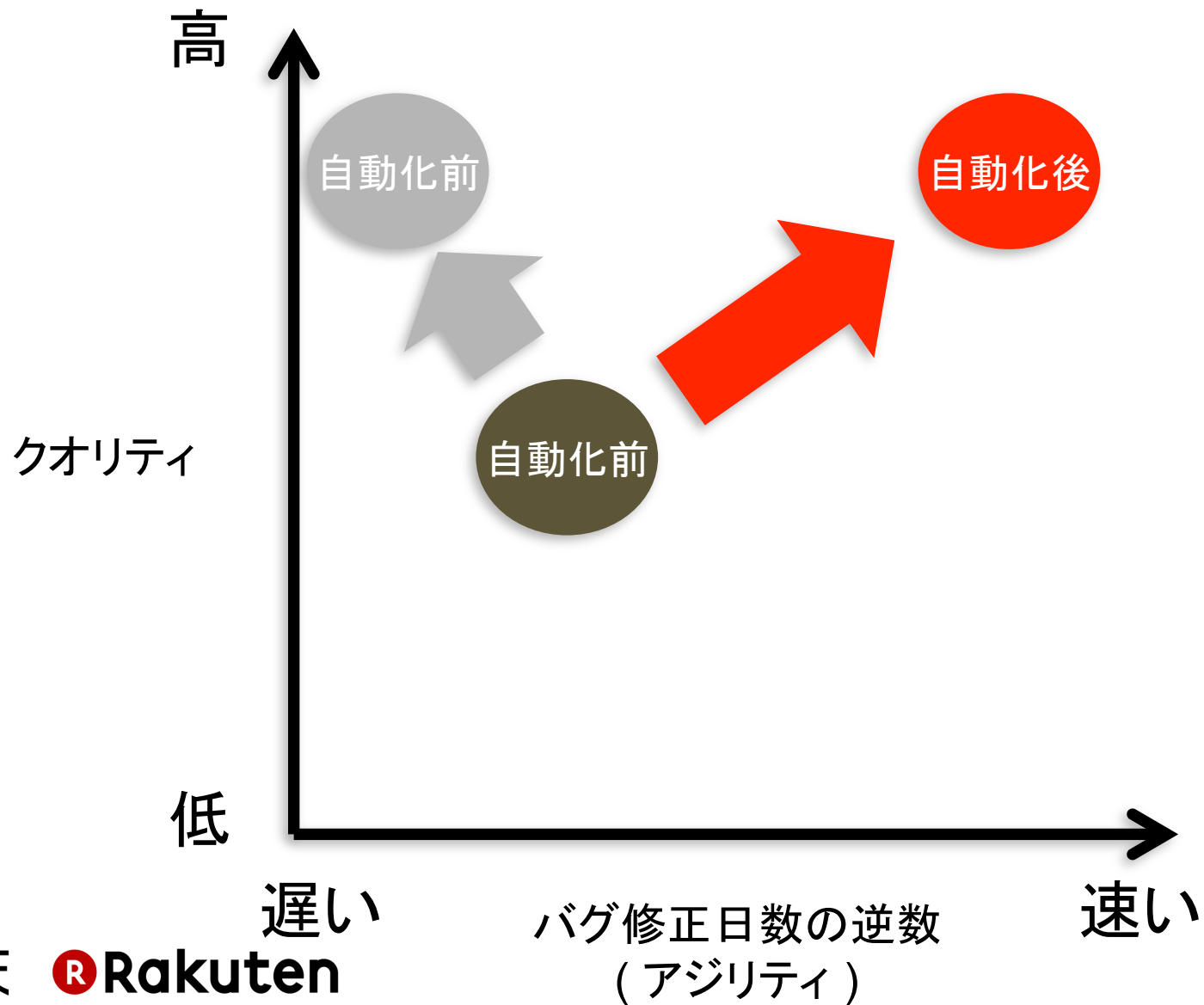
開発者や分析者が積極的かつ安全な改善を
継続的に行う事が可能に！

まとめと 今後の課題

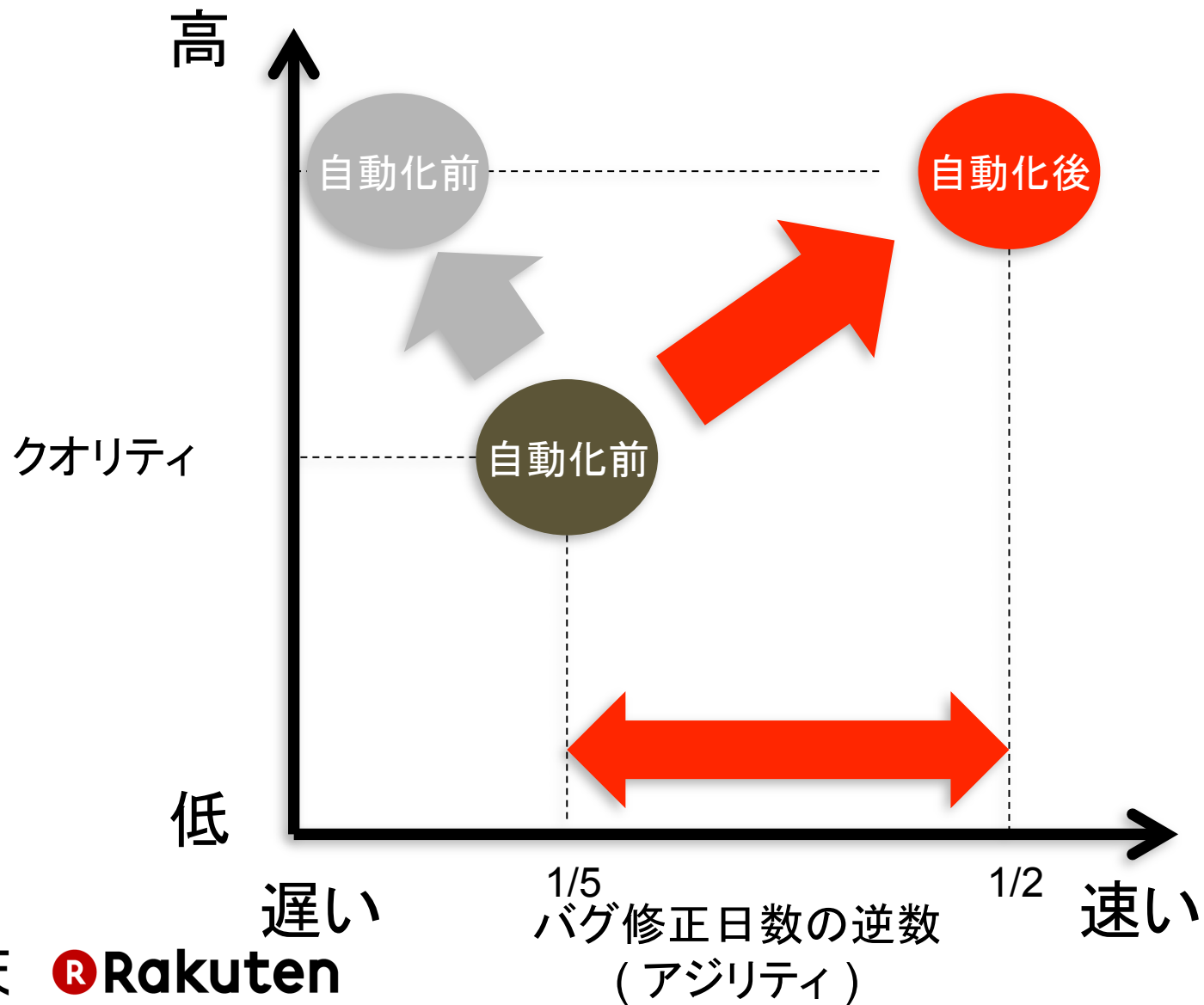
まとめ: 継続的システムテストの意味



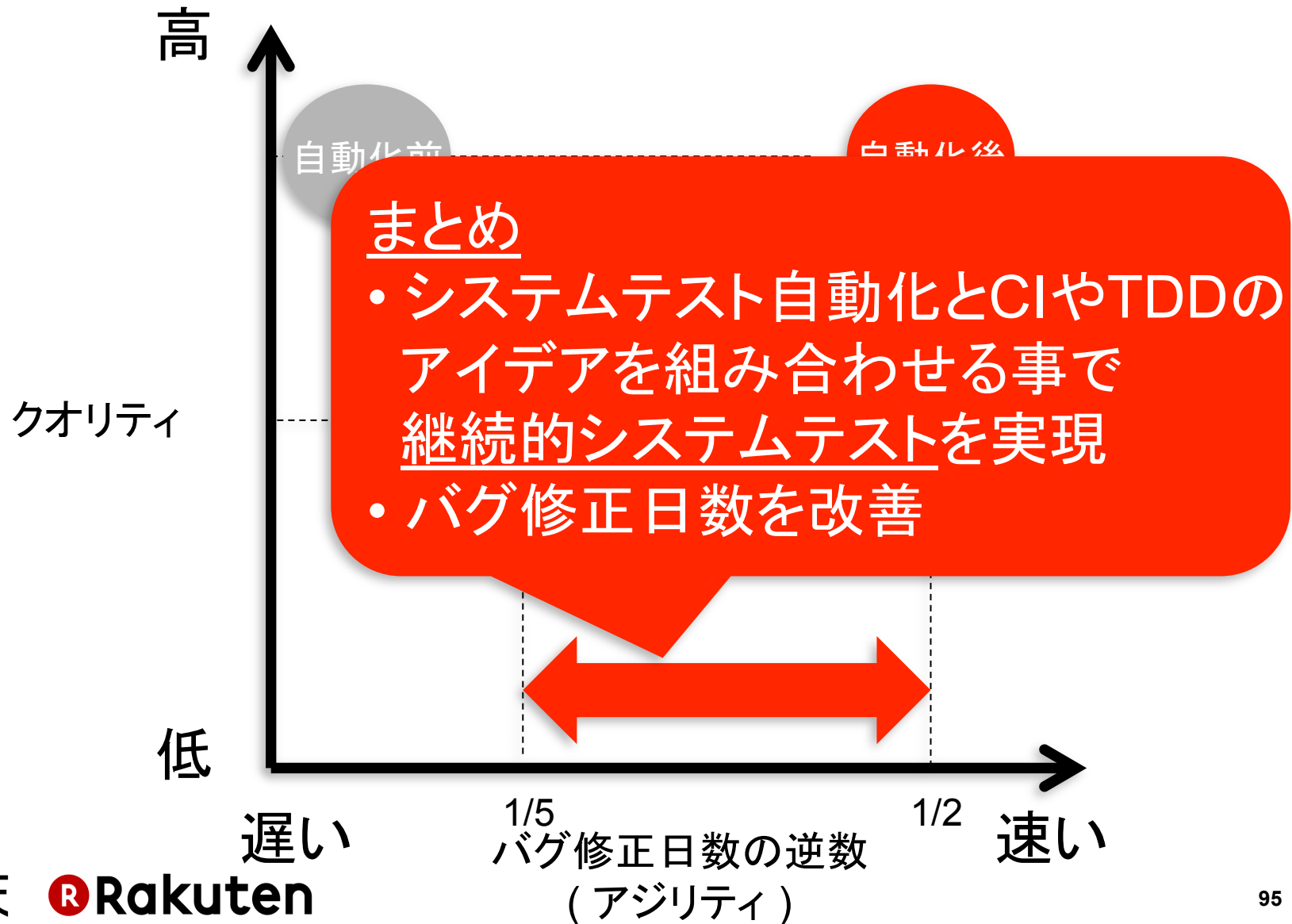
まとめ: 継続的システムテストの意味



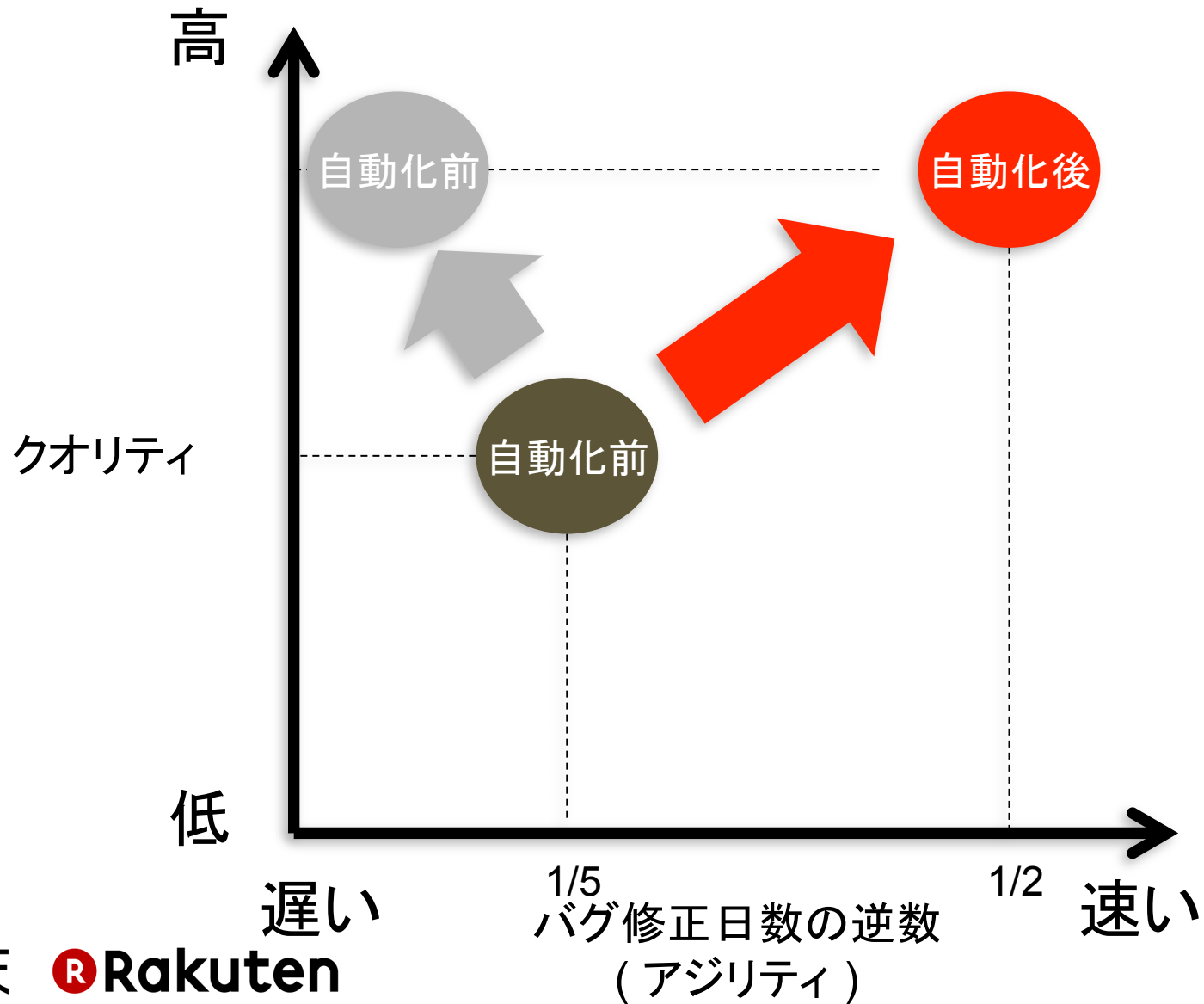
まとめ: 継続的システムテストの意味



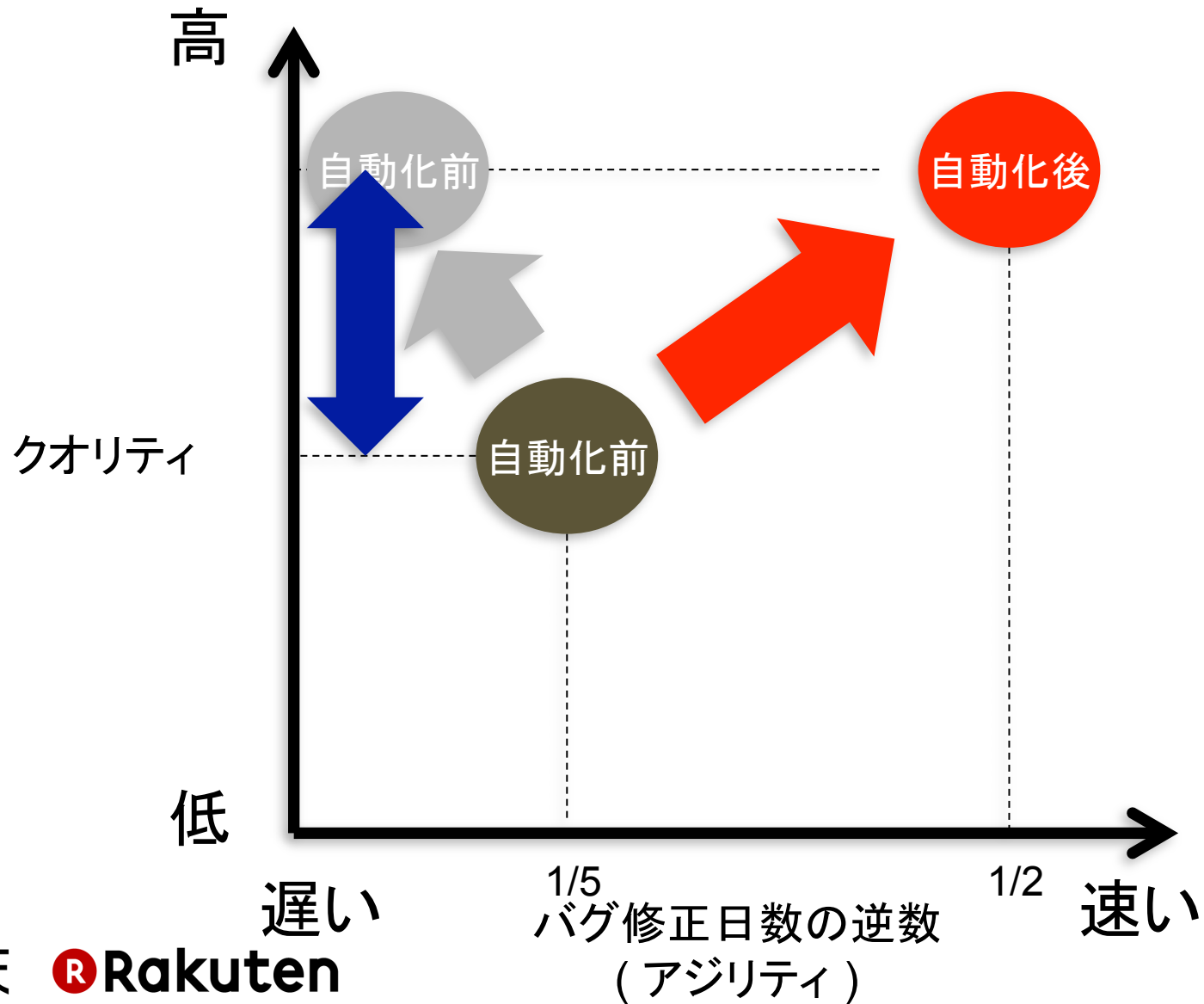
まとめ: 継続的システムテストの意味



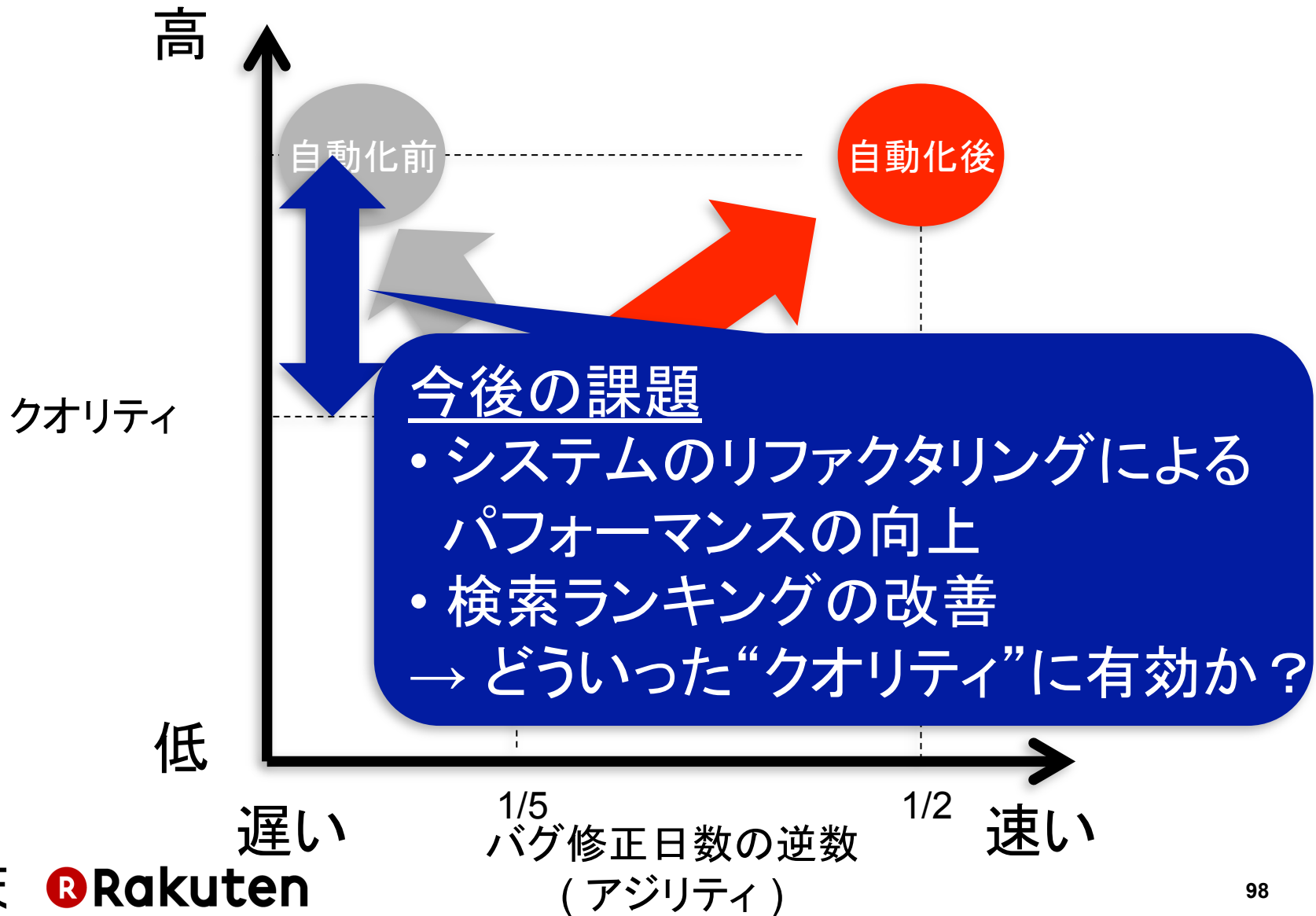
まとめ: 継続的システムテストの意味



まとめ: 継続的システムテストの意味



まとめ: 継続的システムテストの意味



謝辞



**Good to see
system test
getting some**

