

テスト種別&観点カタログガイド

第1.2版

2018年8月16日



この作品は [クリエイティブ・コモンズ 表示 - 継承 4.0 国際 ライセンス](https://creativecommons.org/licenses/by-sa/4.0/) の下に提供されています。

テスト種別&観点カタログ©2018 TIS INC. クリエイティブ・コモンズ・ライセンス（表示-継承 4.0 国際）

変更履歴

No.	版数	変更日	区分	変更項目の番号・名称	変更内容
1	1.2版	2018/8/16	変更	表紙	ライセンス情報を付与

1. 本書について

本書（テスト種別&観点カタログ）は、TIS社内から収集した事例をもとに、アプリケーション開発で使用可能なテストの種別と観点を整理したカタログです。

テストの種別（以下、「テスト種別」）とは、アプリケーションの検証の目的によりテストを分類したものです。

例えば、「機能テスト」や「性能テスト」、「セキュリティテスト」などが含まれます。

本書では、「テスト種別カタログ」として、テスト種別をまとめた資料を提供します。

テストの観点（以下、「テスト観点」）とは、検出したい不具合または検証したい対象を踏まえて整理した、テストケースを考える上でのベースになるものです。

例えば、「単項目バリデーション - 必須バリデーション」、「DB制御 - 排他制御」などが含まれます。

本書では、「テスト観点カタログ」として、テスト観点をまとめた資料を提供します。

なお、本ファイル（テスト種別&観点カタログガイド）では、テスト種別カタログ、テスト観点カタログの概要、活用方法を説明しています。

また、別紙として、テスト種別を整理する際に想定した成果物の一覧と、テスト種別を工程に割り当てる例を添付しています。

1.1. 目的

本書の目的は以下の通りです。

- ・テストの工程(※)で実施する内容を整理する際に、テスト種別の単位で検討することにより、テスト工程の中身が具体的に想像し易くなり、ステークホルダー間での認識齟齬が抑制されること。
- ・テスト観点を使用することで、テストケースの作成を容易にすること。
- ・要件定義～プログラミングの工程で、考慮漏れや最低限決めておくべき点を、テスト種別やテスト観点をもとに確認することで、効率的に品質を作り込めるようにすること。

※：反復開発の場合は「工程」を、対応するアクティビティやイテレーション、スプリントなどに読み替えてください。

1.2. コンセプト

本書のコンセプトは以下の通りです。

- ・プロジェクト固有の事情に依存せず、横断的に利用できる。
- ・プログラム単体に対するテストから、システム全体に対するテストまで幅広くカバーする。
- ・手段に依存せず、プロダクトを検証するために必要な情報を提供する。

1.2.1. 本書のスコープ

1.2.1.1. 取り扱う内容

本書で扱うテスト種別、テスト観点は、以下を基準に整理しています。

- ・プロダクトのうち、アプリケーションに対する検証を対象とする。
 - ・アプリケーションの外観や動作結果など、アプリケーションを外部から見たり、動作させたりしたときの品質
 - ・ソースコードなどのアプリケーションの内部構造の品質

1.2.1.2. 取り扱わない内容

本書では、以下についてスコープ外として取り扱わないものとします。

- ・リリース後に実施されるテスト。（本番検証など）
- ・インフラに関するテスト。
 - ただし、アプリケーションの開発者がテストを実施すると想定したものについては記載する。
- ・検証を行う手段。
 - プロジェクトで採用した技術や求められる品質などをもとに、検証方法（静的解析・レビュー・プログラムを動かすテストなど）や使用する技法を検討すること。
- ・品質指標（品質の状態の確認・改善に使用するメトリクス）の基準値。
 - プロジェクトの特性により大きく左右されるため、各プロジェクトで類似プロジェクトや過去の実績の値を確認して検討すること。
- ・テストの実施順序。
 - 各工程の中でどのテスト種別から実施するか、テスト観点を元に作成したテストケースをどの順番で実施するかは、プロジェクトで求められる品質の優先度やテスト環境の要件などに応じて検討すること。

1.3. 想定している利用対象

1.3.1. 読者

本書の主要なターゲットは、以下を想定しています。

- ・アプリケーションの新規開発や保守開発のプロジェクトで、テスト計画やテストケースの作成またはレビューを担当する予定がある人。
ただし、本書ではテストに必要な基礎知識の解説は行わないため、テストの実施経験があることを前提としています。

1.3.2. プロジェクト

- ・本書は、以下の処理方式(※)をベースにテスト観点を整理しています。
該当する処理方式がないプロジェクトの場合でも、類似の処理に関するテスト観点は利用可能です。
 - ・ウェブアプリケーション
 - ・バッチアプリケーション
 - ・メッセージング
 - ・ウェブサービス
- ・本書は、アプリケーションの新規開発、保守開発のいずれのプロジェクトでも利用可能です。

※：本書ではNablarch5u13 アプリケーションフレームワークで提供している実行制御基盤に基づき、処理方式を定義しています。
詳細は、Nablarchアプリケーションフレームワークの解説書をご参照ください。
https://nablarch.github.io/docs/5u13/doc/application_framework/application_framework/nablarch/big_picture.html

1.3.3. タイミング

テスト計画やテストケース作成など、テストのアクティビティを行うタイミング（工程）で利用されることを想定しています。
他、要件定義～プログラミングのアクティビティでも、求められている品質とよくある不具合の考慮状況のチェックを行うために利用可能です。

1.4. 本書をまとめる上でもとにしたもの

1.4.1. テスト種別をまとめる上でもとにしたもの

要件定義～設計で作成される成果物(※)をもとにしてテスト種別をまとめています。
成果物とテスト種別が紐づいており、要件定義や設計における主要な検討事項についてどのテスト種別で検証するか明らかになっています。
そのほかに、TIS社内から収集した事例、「1.4.3. 参考文献」に挙げた文献を参考にしています。

※：想定した成果物は、「別紙1. 想定成果物一覧」を参照してください。

1.4.2. テスト観点をまとめる上でもとにしたもの

TIS社内から収集した事例をベースに、「1.4.3. 参考文献」に挙げた文献を参考に、まとめています。
テスト観点は、テスト種別ごとに整理しています。

1.4.3. 参考文献

独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター、つながる世界のソフトウェア品質ガイド ～あたらしい価値提供のための品質モデル活用のすすめ～、
<https://www.ipa.go.jp/files/000055008.pdf> (参照 2017-06-29)

独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター、非機能要求グレード、
<https://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html> (参照 2017-06-29)

独立行政法人情報処理推進機構 セキュリティセンター、安全なウェブサイトの作り方、
<https://www.ipa.go.jp/files/000017316.pdf> (参照 2018-02-16)

独立行政法人情報処理推進機構 セキュリティセンター、ウェブ健康診断仕様、
<https://www.ipa.go.jp/files/000017319.pdf> (参照 2018-07-31)

International Software Testing Qualifications Board作成、Japan Software Testing Qualifications Board翻訳、ISTQBテスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2011.J02、
http://jstqb.jp/dl/JSTQB-SyllabusFoundation_Version2011.J02.pdf (参照 2017-06-29)

World Wide Web Consortium (W3C)作成、ウェブアクセシビリティ基盤委員会 (WAIC)翻訳、Web Content Accessibility Guidelines (WCAG) 2.0、
<https://waic.jp/docs/WCAG20/Overview.html> (参照 2018-03-09)

The Open Web Application Security Project, OWASP Top 10-2013 : The Ten Most Critical Web Application Security Risks 日本語版、
https://www.owasp.org/images/7/79/OWASP_Top_10_2013_JPN.pdf (参照 2017-09-20)

1.5. 注意事項

- ・テスト種別カタログ、テスト観点カタログの内容について
本書は、幅広いプロジェクトで利用可能な内容を取りまとめていますが、記載内容の網羅により、対象システムの品質が担保されることを保証するものではありません。
プロジェクト固有の事情を反映したテストを行うためには、本書のテスト種別、テスト観点到追加・変更・削除を行う必要があります。
本書をプロジェクトに適用する方法は、「2. 活用方法」をご覧ください。
- ・テスト種別カタログ、テスト観点カタログに記載している要件定義～設計までの成果物について
テスト種別カタログ、テスト観点カタログには、検証内容の検討で必要になる要件定義～設計の成果物について、主要な例を挙げています。
該当の成果物がないと具体的なテストの実施内容を検討できないため、成果物の入手可否を確認することで、計画したスケジュールのフィジビリティを確認できます。
記載している成果物は、後述の例外を除き、以下のFintanコンテンツで定義されているドキュメントを指します。
 - ・要件定義フレームワーク (2.20)
 - ・Nablarch開発標準また、例外として、上記には定義されていないがプロジェクトで作成されることが多いと想定した以下のドキュメントも記載しています。
 - ・方式設計書
- ・成果物の読み替えについて
以下のプロジェクトの場合は、本書に記載されている要件定義～設計までの成果物について、読み替えが必要です。
プロジェクトで要件や設計を記載しているドキュメントやその他の記録先などに読み替えてください。要件や設計の検討結果であれば、形式は問いません。
 - ・上記のフレームワークや標準を利用していないプロジェクト
要件定義書や設計書を独自の体系で管理しているプロジェクト、要件や設計をWikiやユースストーリーなどファイル以外の形式で管理しているプロジェクトなど。
 - ・上記のフレームワークや標準をもとに独自のドキュメントを整備しているプロジェクト
上記のフレームワークや標準を元に、成果物の分割や統合を行っているプロジェクトなど。
 - ・テストケースを作成する時点で本書に記載されている成果物にあたる文書がまとまっていないプロジェクト
敢えて要件や設計を文書化するタイミングを遅らせているプロジェクトなど。

2. 活用方法

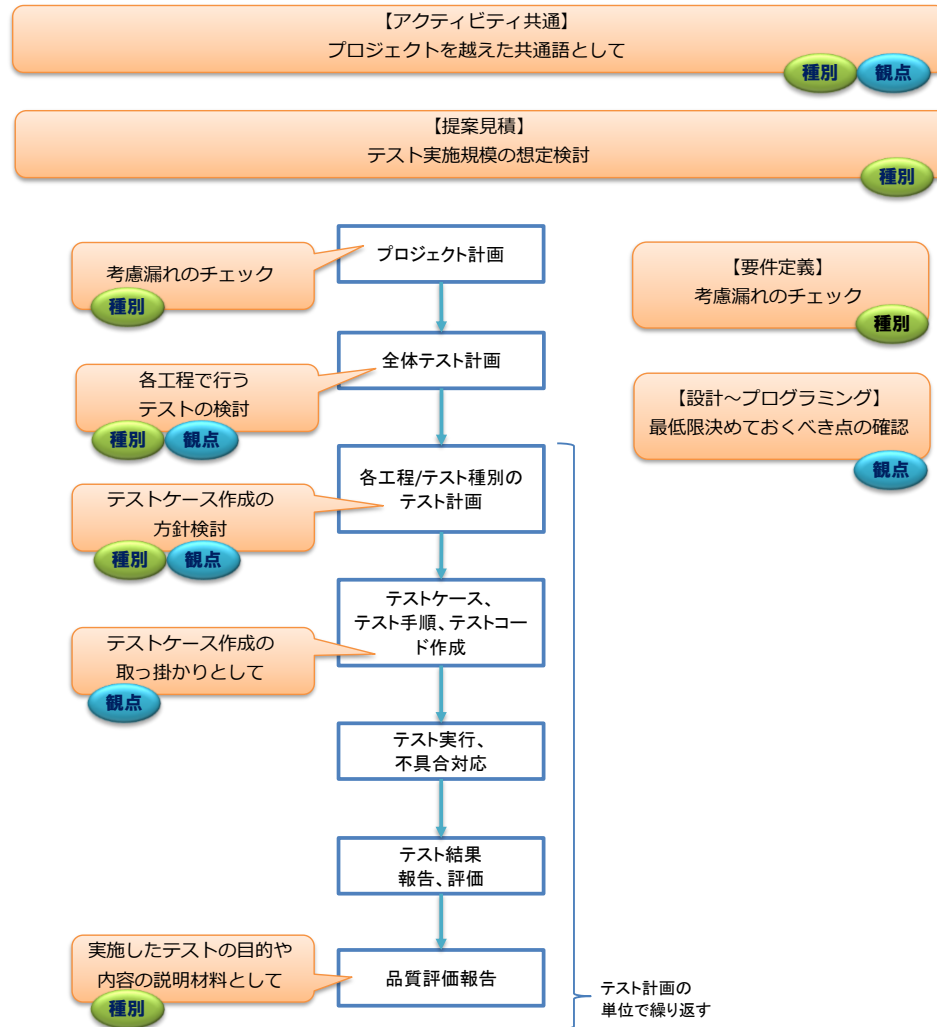
テスト種別、テスト観点のカatalogが、それぞれどのような場面で活用できるか、説明します。

2.1. 活用場面

以下の図は、テストに関するアクティビティ例と、各アクティビティでのCatalogの使いどころ（オレンジの角丸四角形オブジェクト）を示しています。

「種別」（緑の楕円オブジェクト）、「観点」（青の楕円オブジェクト）は、それぞれ「テスト種別Catalog」「テスト観点Catalog」を表し、各資料をご使用いただける場面を示しています。

例）プロジェクトを越えた共通語として、「テスト種別Catalog」に定義されたテスト種別を使用する。



2.2. 活用の流れ

本書を活用する流れについて、新規開発プロジェクトの後、継続して保守開発に取り組む場合の例を示します。
なお、プロジェクトで行われるアクティビティのうち、テストに関係する部分のみを抜粋して記載しています。

新規開発プロジェクト	<p>事前準備</p> <p>…テスト種別カタログとテスト観点カタログを、プロジェクト用に編集する。（「2.3. 事前準備のポイント」参照）</p> <p>なお、以降は編集後のテスト種別カタログ、テスト観点カタログをプロジェクト用として使用する。</p> <p>↓</p> <p>全体テスト計画作成</p> <p>…プロジェクト計画などで決められた工程の定義と、プロジェクト用のテスト種別カタログをもとに、どの工程でどのテスト種別を実施するか決定する。</p> <p>優先度の高いテスト種別がある場合は、プロジェクト用のテスト観点カタログをもとに、検出したい不具合を具体的に検討し、ステークホルダーと早期に合意しておく。</p> <p>テスト種別とテスト工程のマッピングのイメージは、「別紙2.工程へのテスト種別の割り当て例」を参照。</p> <p>↓</p> <p>各工程テスト計画作成</p> <p>…プロジェクト用のテスト種別カタログとテスト観点カタログをもとに、テストの目的や検証内容を再確認した上で各工程のテストケース作成方針を決定する。</p> <p>↓</p> <p>テストケース作成</p> <p>…テストケース作成方針を踏まえて、プロジェクト用のテスト観点カタログを参照しながらテストケースを作成する。</p> <p>↓</p> <p>テスト実行～品質評価</p>
保守開発プロジェクト	<p>全体テスト計画作成</p> <p>…新規開発プロジェクトで作成した全体テスト計画とプロジェクト用のテスト種別カタログをもとに、追加・削除が必要なテスト種別の有無を確認し、保守開発プロジェクトの全体テスト計画検討のインプットとする。</p> <p>↓</p> <p>各工程テスト計画作成</p> <p>…新規開発で作成したテストケース作成方針とプロジェクト用のテスト観点カタログをもとに、追加・削除が必要な方針の有無を確認する。</p> <p>該当する方針があれば、テストケース作成方針を修正する。</p> <p>↓</p> <p>以下、新規開発プロジェクトのテストケース作成以降と同様のため割愛。</p>

2.3. 事前準備のポイント

2.3.1. 事前準備

プロジェクトの特性にあわせ、本書のテスト種別、テスト観点を追加・変更・削除する必要があります。（以降、この作業を「テラリング」と表現します。）

本書では、プロジェクト固有の事情に依存しにくい、横断的に利用できる内容を扱っているため、プロジェクトの品質目標やテストの対象とする範囲を踏まえたテラリングが必須となります。

テラリング後のテスト種別、テスト観点をプロジェクト用の資料としてご使用ください。

- ・プロジェクトではどのような品質がどの程度求められているか、どの外部システムまでをテストの対象とするか、テラリング前に確認しておく必要があります。品質目標やテストの対象とする範囲を踏まえて、本書を参考に必要なテストを決定する想定です。

- ・仕様変更などがあった場合は、適宜テラリング前のテスト種別、テスト観点をもとに見直しを行ってください。

また、本書の新バージョンがリリースされた場合は、最新情報の反映のため、テラリング後のテスト種別、テスト観点の見直しを行うことを推奨します。

2.3.2. テラリングの手順

テラリングする際は、以下の順で考えることを推奨します。

- ・テスト種別カタログのテラリング

1. プロジェクトで決めたテストの目的に基づいて、必要なテスト種別を選択する。
2. プロジェクト固有の目的を持ったテストなど、カタログに相応しいテスト種別がない場合は独自に追加する。

- ・テスト観点カタログのテラリング

3. 1、2でテラリングしたテスト種別に紐づくテスト観点から、プロジェクトに必要な観点を選択する。
4. プロジェクト固有の条件を反映する必要があるテスト観点を編集する。
5. プロジェクト固有の観点など、カタログに該当するテスト観点がない場合は独自に追加する。

2.3.3. テラリングをする際に考慮が必要なポイント

テラリングで考慮が必要なポイントを挙げます。

以下に開発対象やチーム構成に依存しにくいポイントを挙げていますが、業務特性やテスト環境の制約などプロジェクトの特性も考慮してテラリングしてください。

また、テラリング後の資料は、テスト担当者、アーキテクト、業務有識者など複数名で検討やレビューを行うことを推奨します。

2.3.3.1. 内容をテラリングする場合

2.3.3.1.1. テスト種別のテラリングの考慮ポイント

- ・プロジェクトで決めたテストの目的に対して、選択したテスト種別に過不足がないか？
- ・プロジェクトが行う作業や開発の範囲、チーム分けの単位から見て、選択したテスト種別が妥当か？
例：移行作業はスコープ外のプロジェクトのため、移行テストは対象外とする。

2.3.3.1.2. テスト観点のテラリングの考慮ポイント

- ・プロジェクトで採用している処理方式（ウェブアプリケーション、バッチアプリケーションなど）に応じたテスト観点が選択できているか？
- ・テスト対象の入出力の手段及び種類、処理内容を踏まえてテスト観点に過不足ないか？
例：ファイルを扱う処理がないので、ファイルに関するテスト観点は削除する。
- ・類似プロジェクトの過去の不具合や障害と同様の事象を検出できるテスト観点が含まれているか？

2.3.3.2. カタログの項目列を追加、削除する場合

テスト種別カタログ、テスト観点カタログの利用場面やプロジェクトの特性により、項目列を追加、削除したほうが使いやすい場合があります。

例：テスト計画やテストケースに盛り込む必要があるテスト種別やテスト観点が漏れていないことを確認するために、チェック欄を追加する。

例：テスト種別ごとやテスト観点ごとに、検証を行う手段（利用するテスト実行ツール名など）を書く欄を設ける。

本書でテスト種別を整理する際に想定した、要件定義～設計で作成される成果物の一覧です。

一覧にない成果物をもとにしたテスト種別は、一部（テストされる可能性が高いと考えられるもの）を除き本書では取り扱いの対象外としています。

成果物は、本書の「1.5. 注意事項」配下、「テスト種別カタログ、テスト観点カタログに記載している要件定義～設計までの成果物について」のドキュメントを参考に、実装に繋がる成果物のみを抽出しています。

（例：「概算見積り」は実装に繋がる内容ではないので一覧に記載しない。）

凡例

「成果物分類」の記号は、以下のような基準で記載しています。

○	テスト対象を合格とする際に、その判断基準となる定義・設計を記載している主要な成果物。 該当成果物は、テスト種別カタログの「インプットとなる主要な成果物」に、テスト種別と紐付けて記載している。
△	テストの前提条件や合格条件となる詳細情報を記載している成果物。本書では「○」とした成果物を補うものとして扱う。 該当成果物を参照しているなど関連のある「○」の成果物名を併記している。
×	本書で取り扱わない内容を記載する成果物。
※	扱いが上記に当て嵌まらないもの。「備考」を参照。

成果物 参考元	成果物名		成果物分類		備考
Fintan『要件定義フレームワーク (2.20)』	業務要件定義書	組織一覧	△	業務フロー	
		アクター一覧	△	業務フロー	
		業務階層定義	△	業務フロー	
		業務フロー	○	-	
		イベント一覧	○	-	
		状態遷移モデル定義	○	-	
		業務ルール定義	○	-	
		データフロー	○	-	
		概念データモデル定義	△	システム機能設計書	
		CRUD	△	システム機能設計書	
	システム要件定義書	システム機能一覧	△	システム機能設計書	
		システム機能一覧	△	システム機能設計書	業務要件定義成果物の同名成果物と同一。
		システム機能俯瞰図	△	システムフロー	
		システムフロー	○	-	
		画面機能要件定義	△	システム機能設計書	
		バッチ機能要件定義	△	システム機能設計書	
		帳票機能要件定義	△	システム機能設計書	
		外部IF機能要件定義	△	システム機能設計書	
		論理データモデル定義	△	システム機能設計書	
		CRUD図	△	システム機能設計書	
		非機能要件定義	※	-	各非機能要件を取りまとめた資料。必要に応じて、各成果物から参照する。
		可用性要件定義	○	-	
		性能・拡張性要件定義	○	-	
		運用・保守要件定義	○	-	
		監視要件定義	○	-	
		バックアップ要件定義	○	-	
		ネットワーク要件定義	×	-	インフラに関する記載のため対象外。
		移行性要件定義	○	-	
		セキュリティ要件定義	○	-	
		システム環境要件定義	×	-	インフラに関する記載のため対象外。
Fintan『Nablarch開発標準』	アプリケーション設計標準	UI標準	△	システム機能設計書	
		UI部品カタログ	△	システム機能設計書	
		カラーユニバーサルデザイン対応ガイド	△	システム機能設計書	
		DB設計標準	△	システム機能設計書	
		共通コンポーネント設計標準	△	共通コンポーネント設計書	
		コーディング規約	○	-	
		システム機能一覧	△	システム機能設計書	
		システム処理フロー	△	システム機能設計書	
		システム機能設計書	○	-	
		画面遷移図	○	-	
		画面一覧	△	システム機能設計書	
		リクエスト一覧	△	システム機能設計書	
		外部インターフェース一覧	△	システム機能設計書	
		外部インターフェース設計書	△	システム機能設計書	
		サブシステムインターフェース一覧	△	システム機能設計書	
		サブシステムインターフェース設計書	△	システム機能設計書	
		共通コンポーネント一覧	△	共通コンポーネント設計書	
		共通コンポーネント設計書	○	-	
		コード設計書	△	システム機能設計書	
		メッセージ設計書	△	システム機能設計書	
		帳票一覧	△	システム機能設計書	
		帳票設計書	△	システム機能設計書	
		メール設計書	△	システム機能設計書	
		ドメイン定義書	△	システム機能設計書	
		単語辞書	△	システム機能設計書	
		ER図	△	システム機能設計書	
		テーブル一覧	△	システム機能設計書	
		テーブル定義書	△	システム機能設計書	
		ビュー定義書	△	システム機能設計書	
		採番一覧	△	システム機能設計書	
		トリガー定義書	△	システム機能設計書	
		物理データモデル設計書	△	システム機能設計書	
		ネット・ジョブフロー	○	-	
		データ移行設計書	○	-	
		環境運用設計書	○	-	
その他	方式設計書	※	-		必要に応じて、各成果物から参照する。

別紙2. 工程へのテスト種別の割り当て例

工程ごとにテスト種別を割り当てた例を示します。
なお、工程とテスト種別の組み合わせはプロジェクトの特性に応じて検討することを前提としており、以下のように工程を定義すべきと定めるものではありません。

工程	テスト種別	テスト実施スケジュール																	
		4月		5月		6月		7月		8月		9月		10月		11月		12月	
プログラミング	構文チェック																		
	機能テスト（モジュール単位）																		
	機能テスト（画面・バッチ単位）																		
	セキュリティテスト																		
	性能テスト																		
結合テスト	機能テスト（ユースケース単位）																		
	業務シナリオテスト																		
	セキュリティテスト																		
システムテスト	業務シナリオテスト																		
	データ互換性テスト																		
	構成テスト																		
	性能テスト																		
	ストレステスト																		
	ボリュームテスト																		
	ロングランテスト																		
	障害テスト																		
	運用シナリオテスト																		
	移行テスト																		

移行テストで生成した
XXデータを各テスト環
境へ

1回目(手作成データ)

2回目(移行データ)

1回目(データ量が少)

2回目(データ量が本番同等)

3回目(データ量、スケジュールが本番同等)

データ量が少

データ量が本番同等