

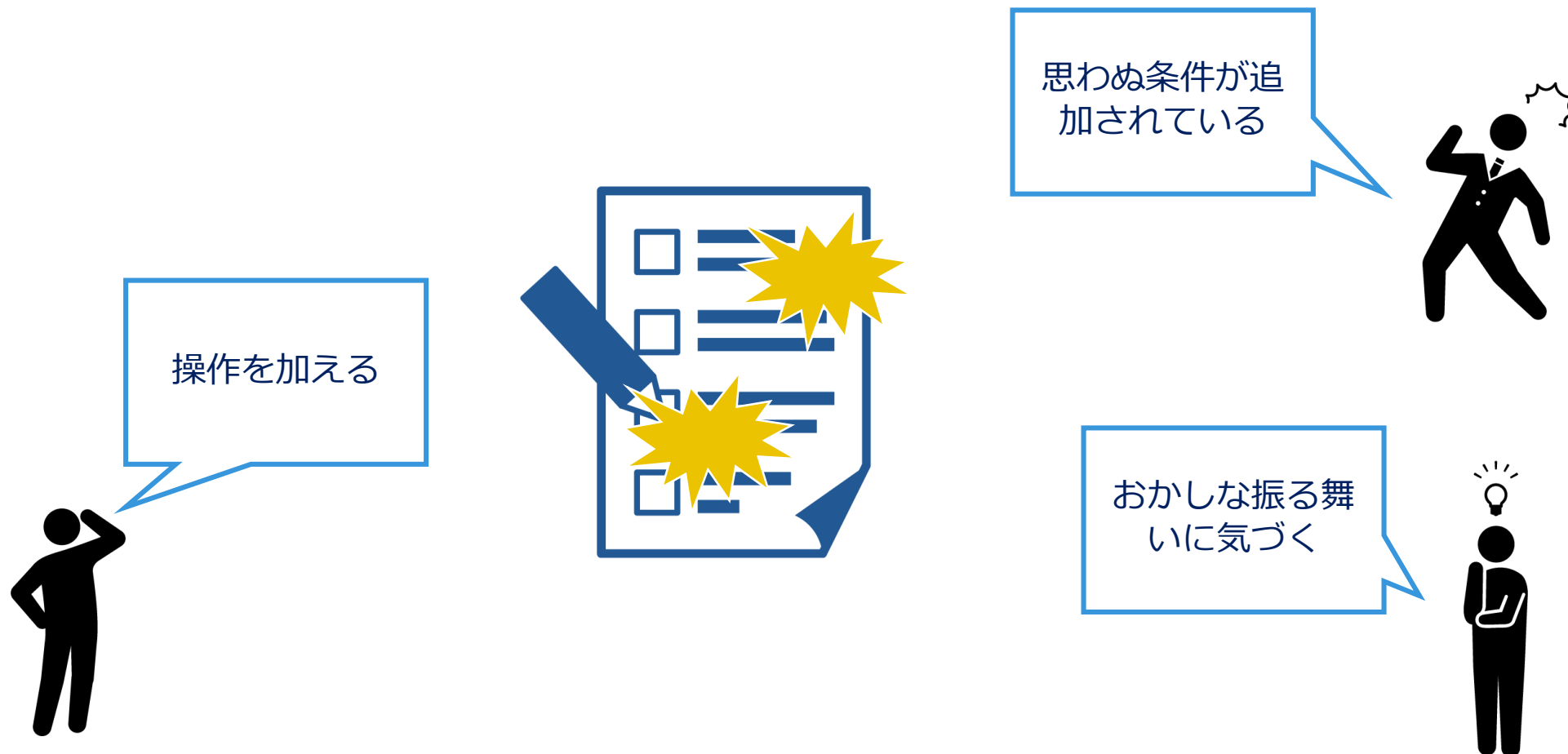
NGTの記法を応用した不具合分析からのテストの補強

2018年3月7日

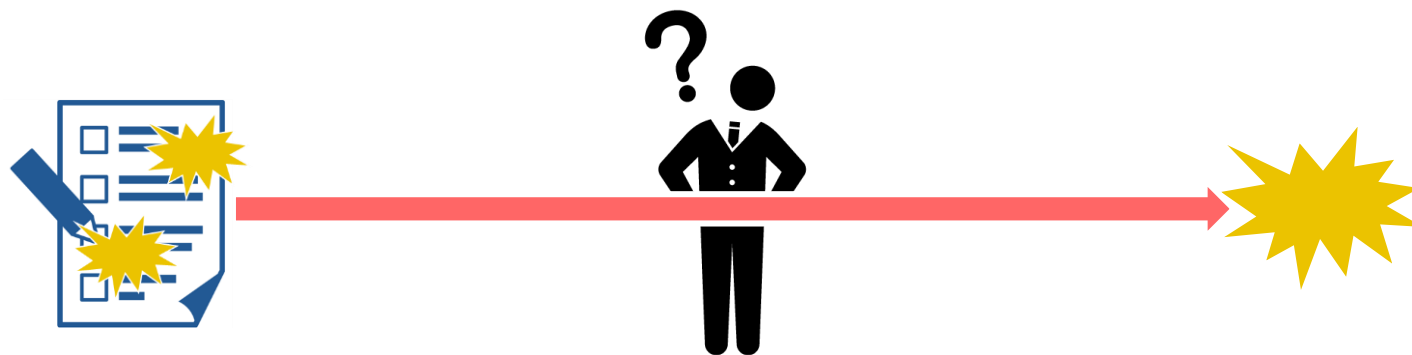
株式会社ベリサーブ
吉川 努

- 背景
- 問題と課題
- 取り組み内容
- NGTとは
- なぜNGTを活用しようと思ったのか
- NGTの応用方法
- 適用事例
- 結果
- 発見した不具合
- まとめ

テスト実行者によって不具合が発見される場合がある



もしテスト実行者が発見してくれないと不具合は見逃されてしまう



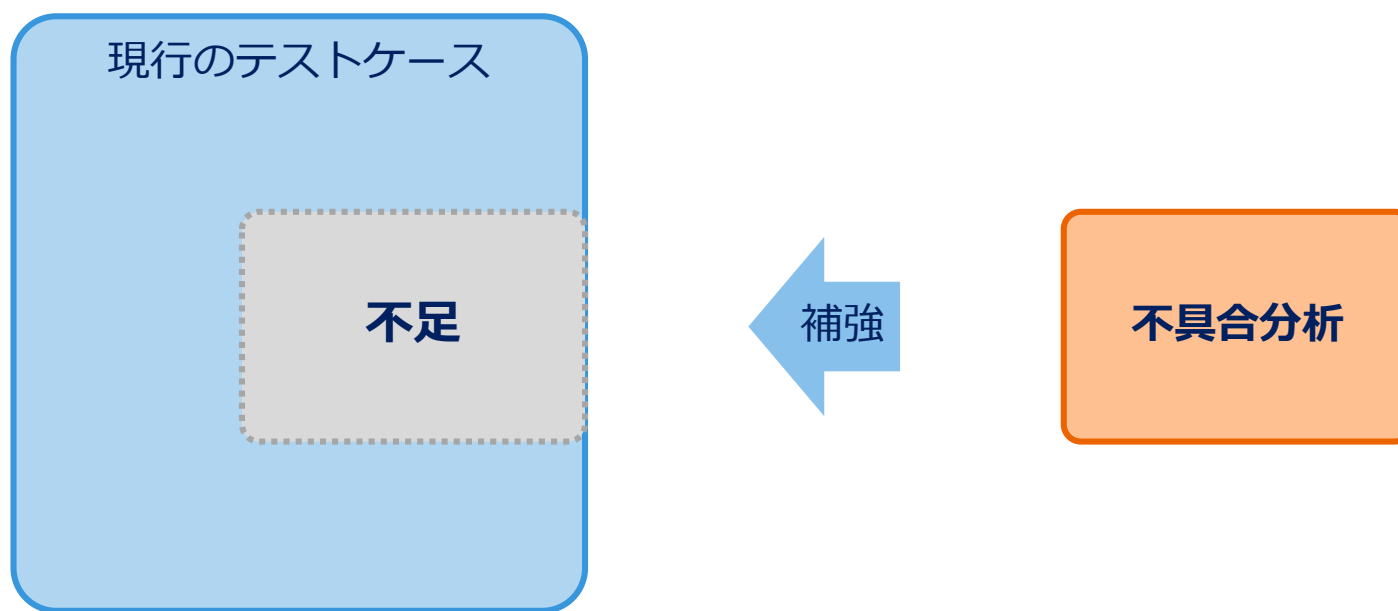
問題

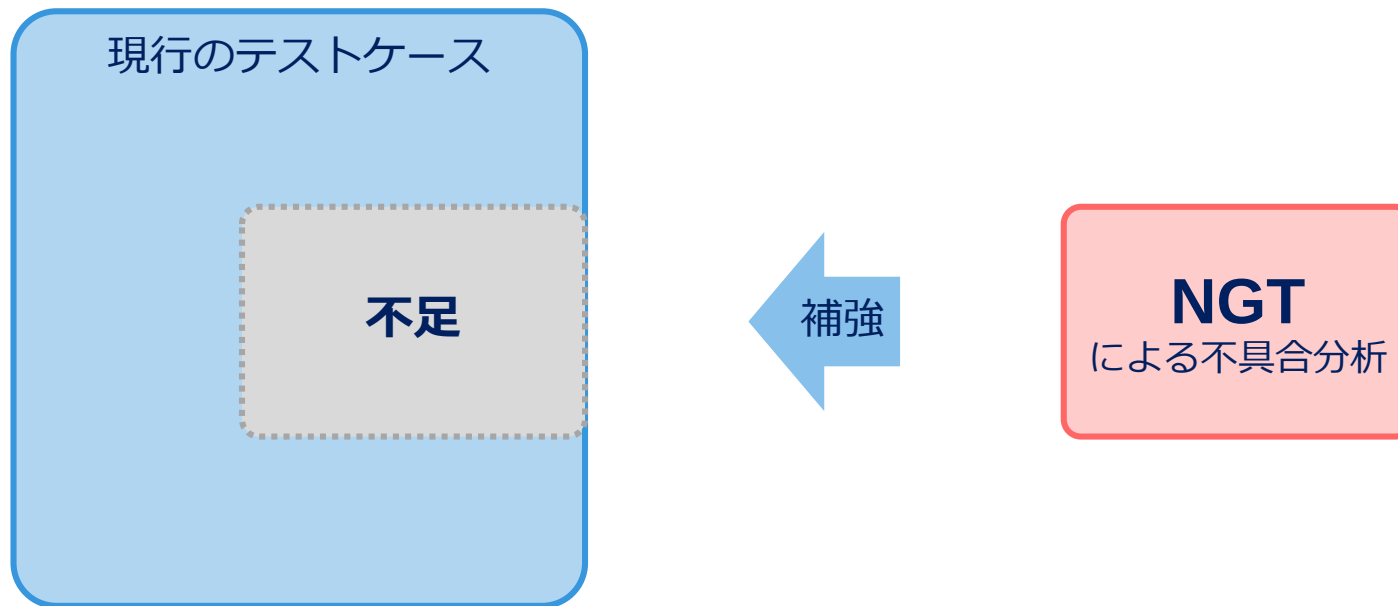
不具合の発見がテスト実行者に依存している

課題

プロジェクト期間内にテストを補強して
潜在している不具合を発見しなければならない







➤ NGTとは…

- Notation for Generic Testingの略
- 電気通信大学の西康晴氏が提唱する**テスト開発のための記法**

テストすべき事柄を「テスト観点」としてツリー上に展開していく

➤ NGTによるテスト観点図

- テストケースとテスト観点とを必ず**結びつける**ようにテスト設計する
- **親子関係をきちんと明示**し網羅性を保証する
- テスト観点を列挙・整理し全体像を把握する

観点同士のつながりは線や矢印などで関連づける

出典：「テスト観点に基づくテスト開発方法論 VSTePの概要」VSTeP.130510.color.pdf

➤ 詳細化

■ ステレオタイプ

◆ 継承・部分・属性化・原因結果

■ ビューとビューポイント

➤ 関連

■ 組み合わせ関連

■ 順序依存関連

■ 関連ビューポイント

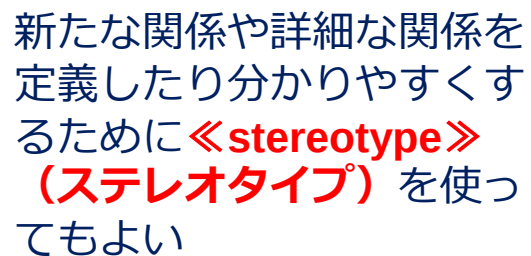
➤ テストフレーム

➤ テストコンテナ

ステレオタイプを使うことで
新たな関係や詳細な関係を定義したり、
わかりやすくすることができる

出典：「テスト観点に基づくテスト開発方法論 VSTePの概要」VSTeP.130510.color.pdf

■ テスト観点を階層的に記述していく

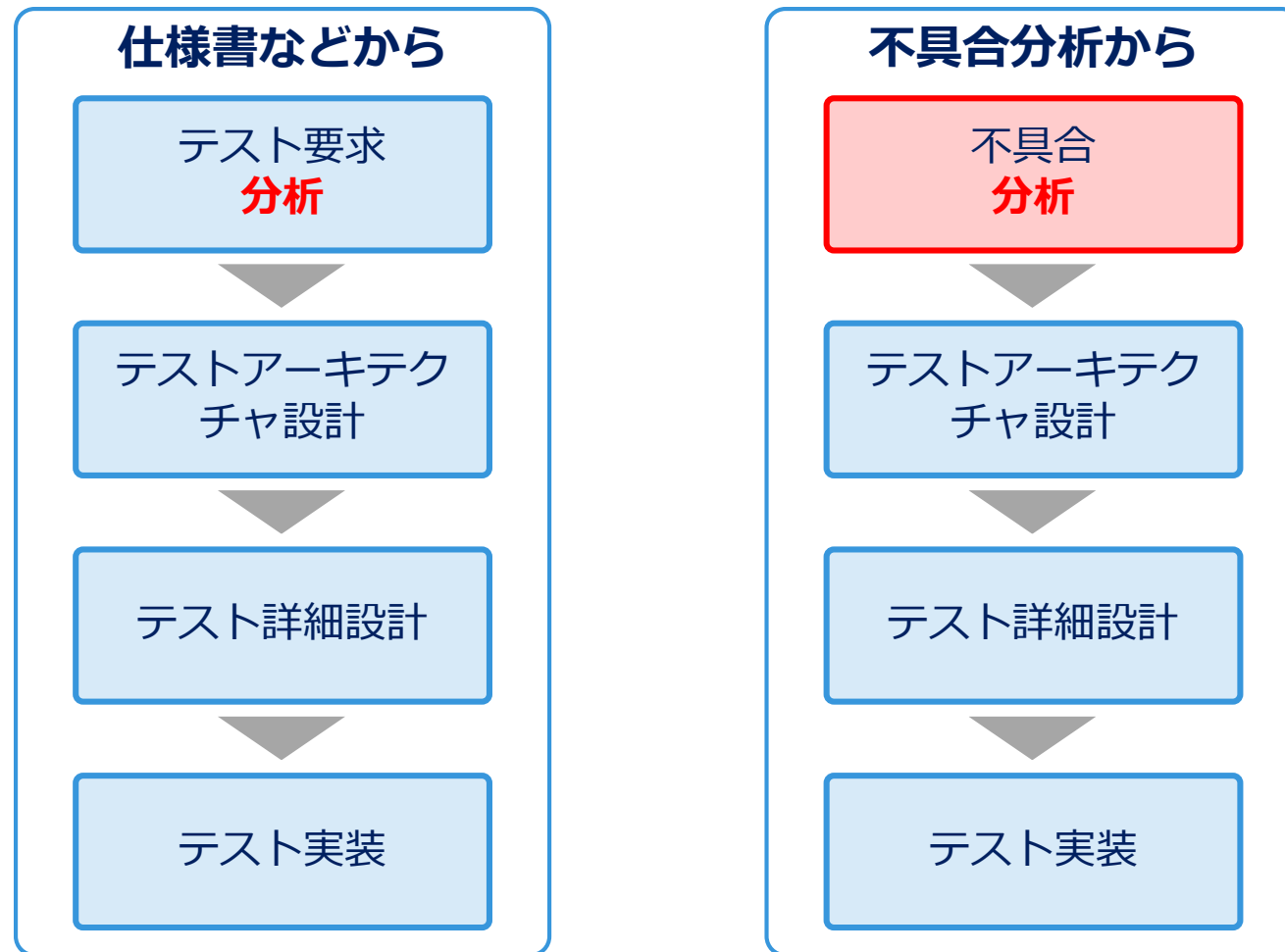


出典：「テスト観点に基づくテスト開発方法論 VSTePの概要」VSTeP.130510.color.pdf

なぜNGTを活用しようと思ったのか

VERISERVE

テスト開発プロセスの比較



なぜNGTを活用しようと思ったのか

VERISERVE

他の不具合分析のやり方ではしっかりこなかった

機能に対する傾向分析

どんなテストをすれば
いいかを導出しにくい

どんなときに不具合が起こりやすいかという点に着目して分析したかった

不具合の要素で分析

キーワードが並べられるだけで、それらをうまく結び付けられない

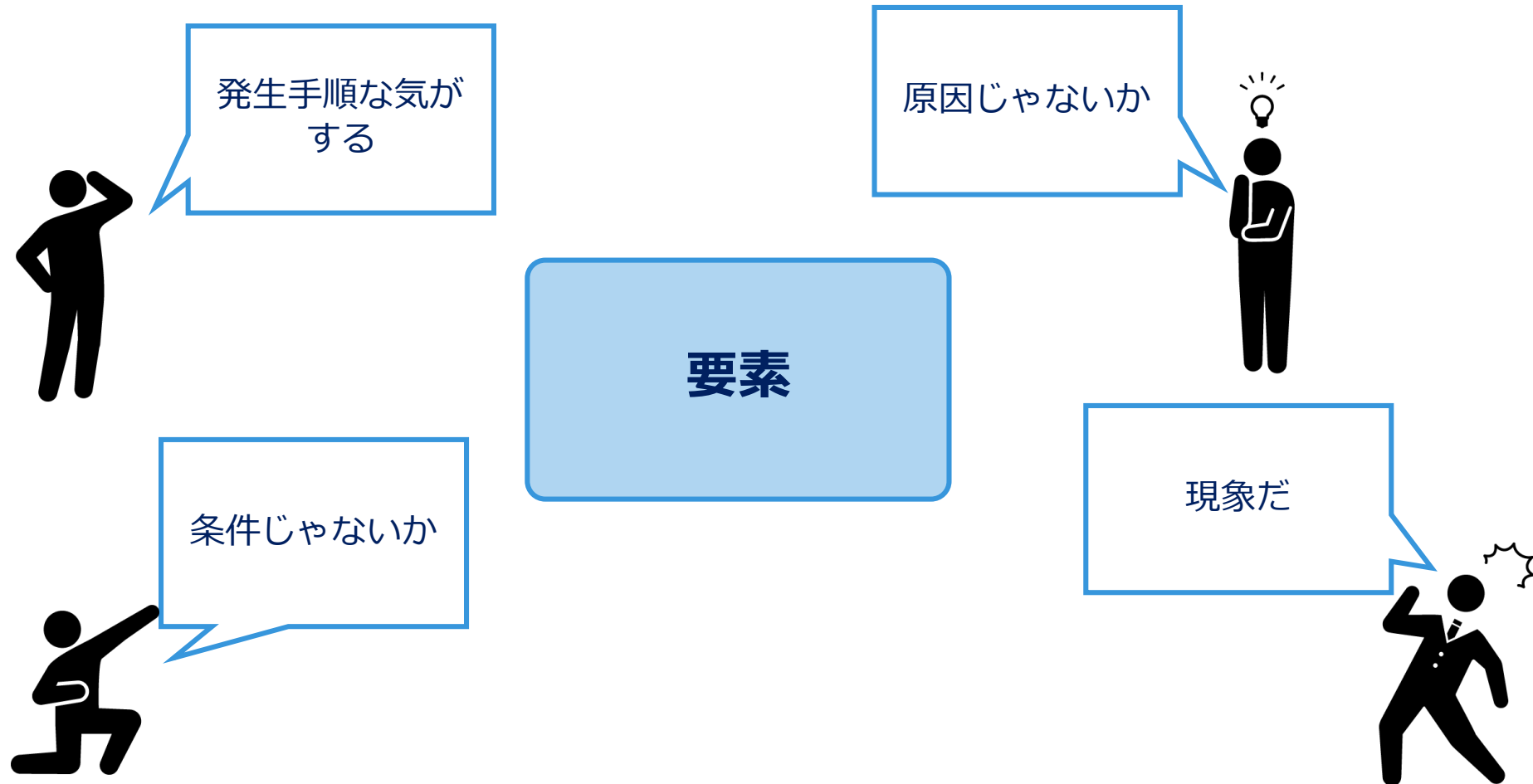
潜在的な結びつきを導出したかった

結びつきがない箇所こそ潜在的な関係性が存在している

なぜNGTを活用しようと思ったのか

VERISERVE

おもむろに不具合の要素を抽出したかった



1) NGTの記法を応用し不具合分析する

不具合分析にNGTの記法を応用して不具合の要素を抽出し、その要素間の関係性を結びつける

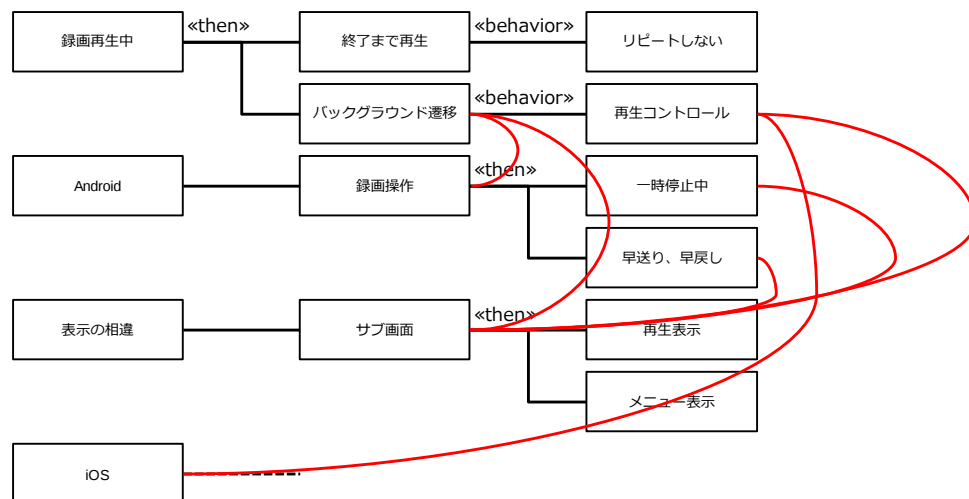


図 NGTを応用した例

2) フレームワークに当てはめる

結びつけた要素をフレームワークに当てはめてテストケースの形式に整理する

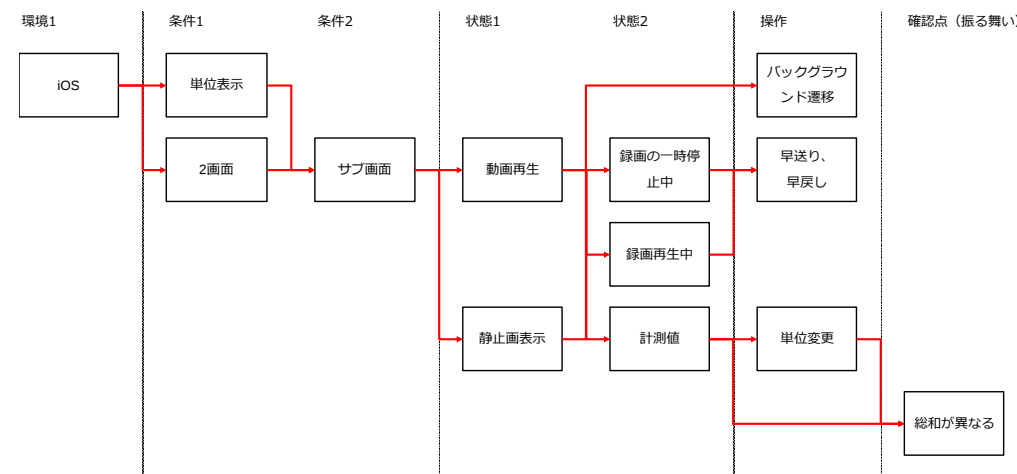


図 フレームワークの例

流れ

1) NGTによる不具合分析

①要素を
抜き出す



②要素を
つなげる



③つながりのない
要素をつなげる



2) フレームワークに当 てはめる

④フレームワークに
当てはめる

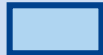
ステレオタイプに着目

要素を抜き出し、ツリー上に表現していくのはそのままに、
新たなステレオタイプを使って不具合分析しやすいようにした

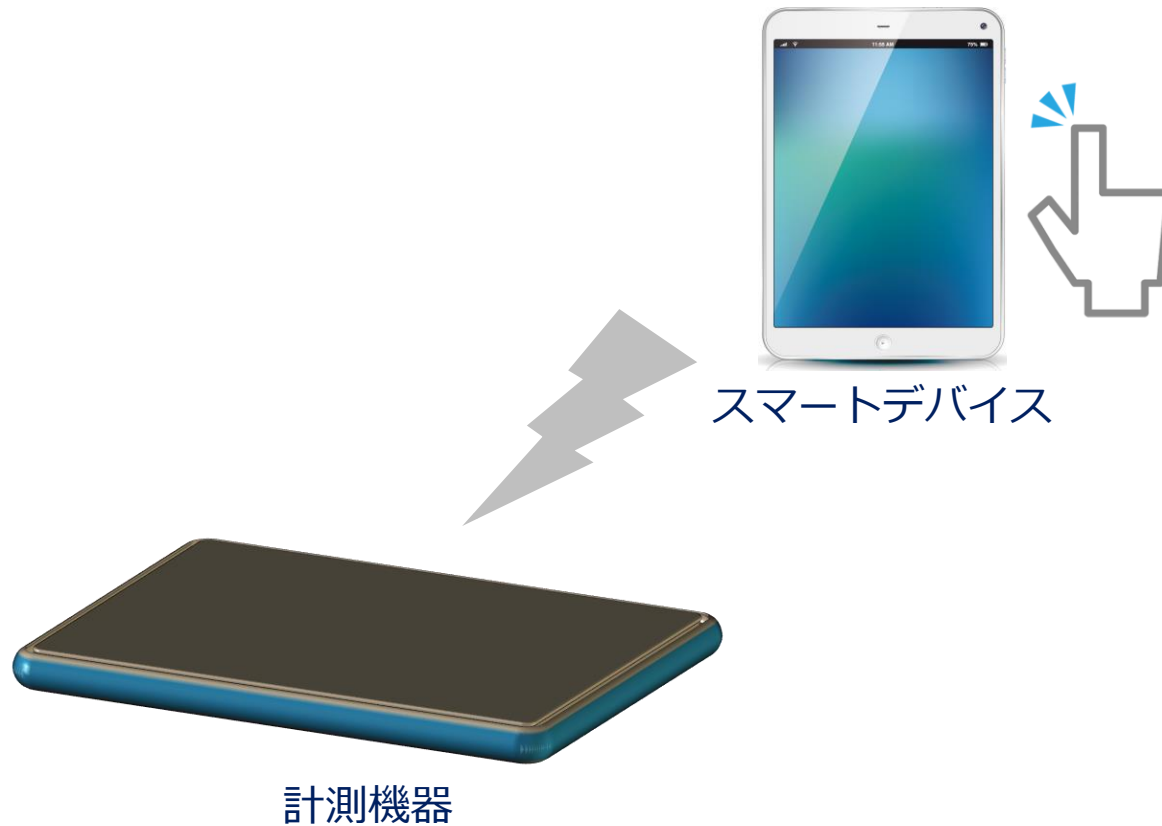
- «and» : 左側の要素と右側の要素を合わせて条件となる関係性を示す。並列を表す。
- «then» : 左側の要素の条件の後、右側の要素の状態になる関係性を示す。順序を表す。
- «after» : 左側の要素の条件の前提条件が右側の要素となる関係性を示す。逆順を表す。
- «behavior» : 右側の要素が不具合の現象を表す。

NGT記法との比較

項目	NGT
テスト対象を表す	
テスト観点を表す	
テスト観点間の関連を表す	
順序依存関係を表す	
詳細化関連を表す	
新たな関係	《stereotype》

項目	今回の取り組み (囲み部分)
(要素として統合)	
要素を表す	
つながりのない要素間の関連を表す	
順序依存関係を表す	《after》 《then》
詳細化関連を表す	
新たな関係	《behavior》 《and》

適用したシステム



テスト設計 のやり方	仕様ベース
	機能分解したモノに対してテストタイプを当てはめていく

①不具合の情報から要素を抜き出す

N O	タイトル	手順	内容
0 1	バックグラウンドへ遷移したアプリに30秒以内に復帰した際、再生コントロールが正しく動作しない	1. 1画面で動画再生し、「一時停止」ボタンを押す。 2. メールアプリに遷移する 3. バックグラウンドに遷移したアプリを30秒以内にフォアグラウンドに戻す。	- アプリに戻った後、再生コントロールが正しく動作しない。 30秒以内にアプリ画面に復帰すると、動画の再生が早送り状態のままとなっている。

バックグラウンド遷移

再生コントロールが正しく動作しない

動画再生中

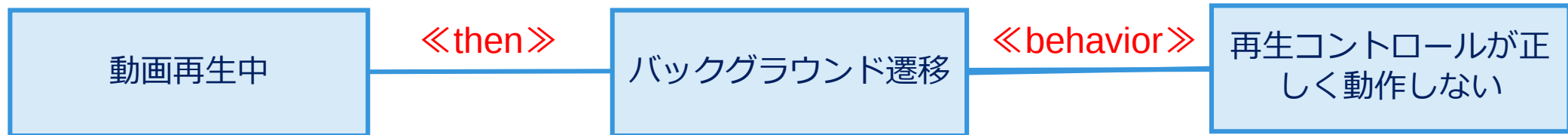
①不具合の情報から要素を抜き出す ～その方法～

- 発生しない条件との違いを抽出する
- 「環境」－「前提」－「手順（トリガー）」－「振る舞い」を意識する
- その中でポイントとなった箇所を特定する
 - 発生しない条件との違いが出ないような箇所は抜き出す対象にはならない。
 - 例えば異なる環境でも発生するなら環境は関係ないのでその要素は抽出しない。

これらの要素を抜き出すのでこれらを不具合報告に情報として記録しておく必要がある

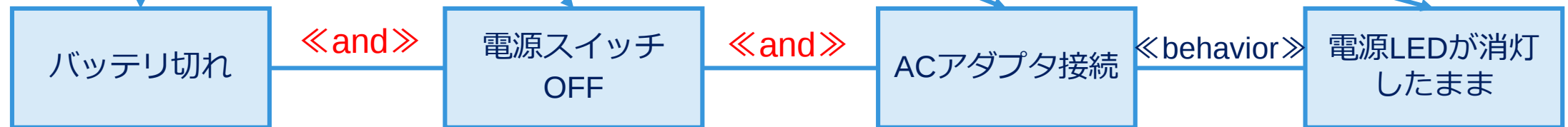
②応用した記法で要素をつなげる 《then》《behavior》の場合

N O	タイトル	手順	内容
0 1	バックグラウンドへ遷移したアプリに30秒以内に復帰した際、再生コントロールが正しく動作しない	1画面で動画再生し、「一時停止」ボタンを押す。 - メールアプリに遷移する - バックグラウンドに遷移したアプリを30秒以内にフォアグラウンドに戻す。	- アプリに戻った後、再生コントロールが正しく動作しない。 30秒以内にアプリ画面に復帰すると、動画の再生が早送り状態のままとなっている。



②応用した記法で要素をつなげる 《and》の場合

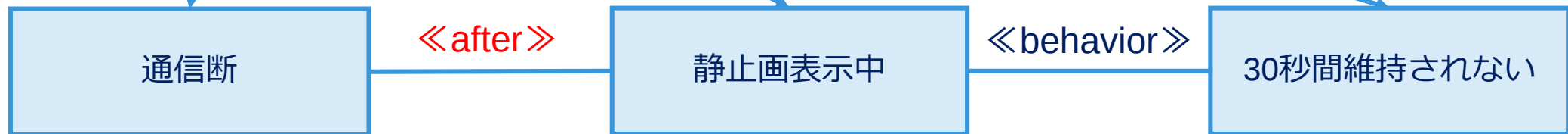
N O	タイトル	手順	内容
02	バッテリー切れ で 電源スイッチOFF の本体に ACアダプタを接続 して数分の後、点滅していた電源LEDが消灯したままとなる	【前提条件】 - 電源SW：OFF - ACアダプタ：接続 - バッテリー充電：無 【手順】 1.前提条件セット後、数分間経過させる	ACアダプタ接続している状態で、電源LEDが消灯したままとなり充電状態判別が不可となる



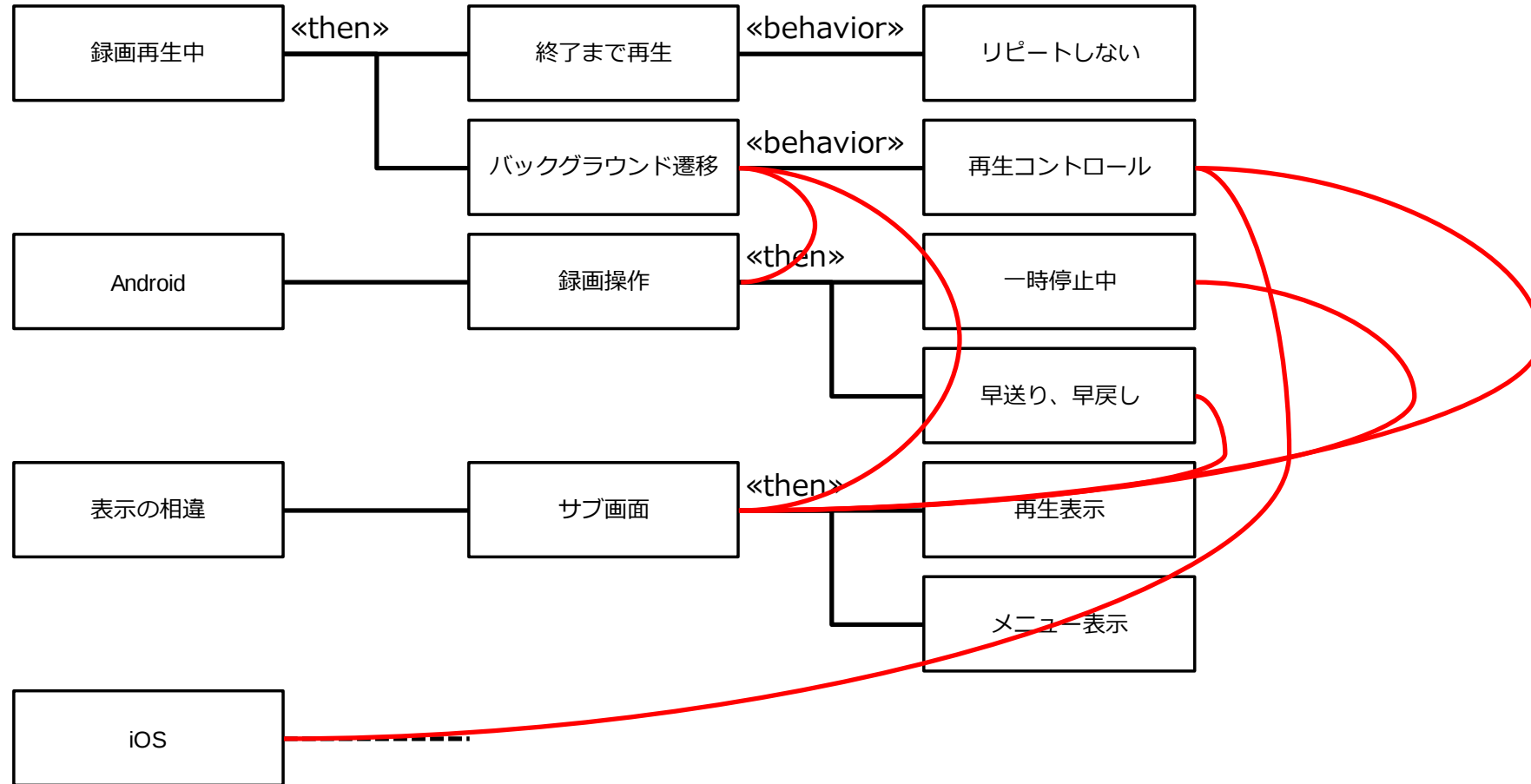
②応用した記法で要素をつなげる 《after》の場合

N O	タイトル	手順	内容
03	静止画表示中に、通信を切断すると、30秒経過しないうちに別の画面に遷移する（静止画表示が保持されない）	1. 静止画を表示する 2. 本体の電源スイッチをOFFにして無線LAN通信を切断する 3. アプリの画面状態を確認する	本体電源OFFの17,18秒経過後にエラー表示が出て、静止画表示が中断される

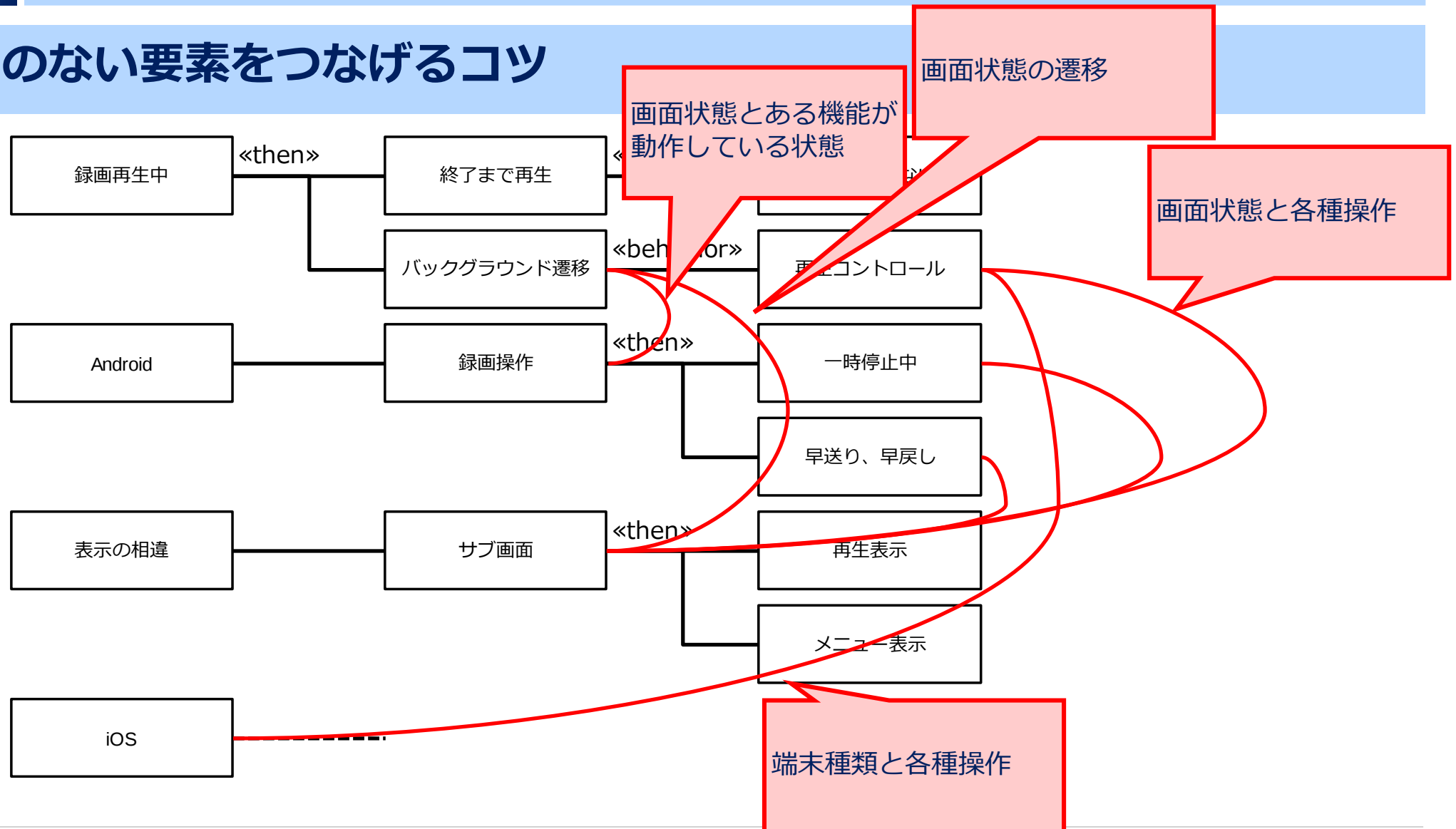
30秒以内とは状態を保持する時間に関する仕様



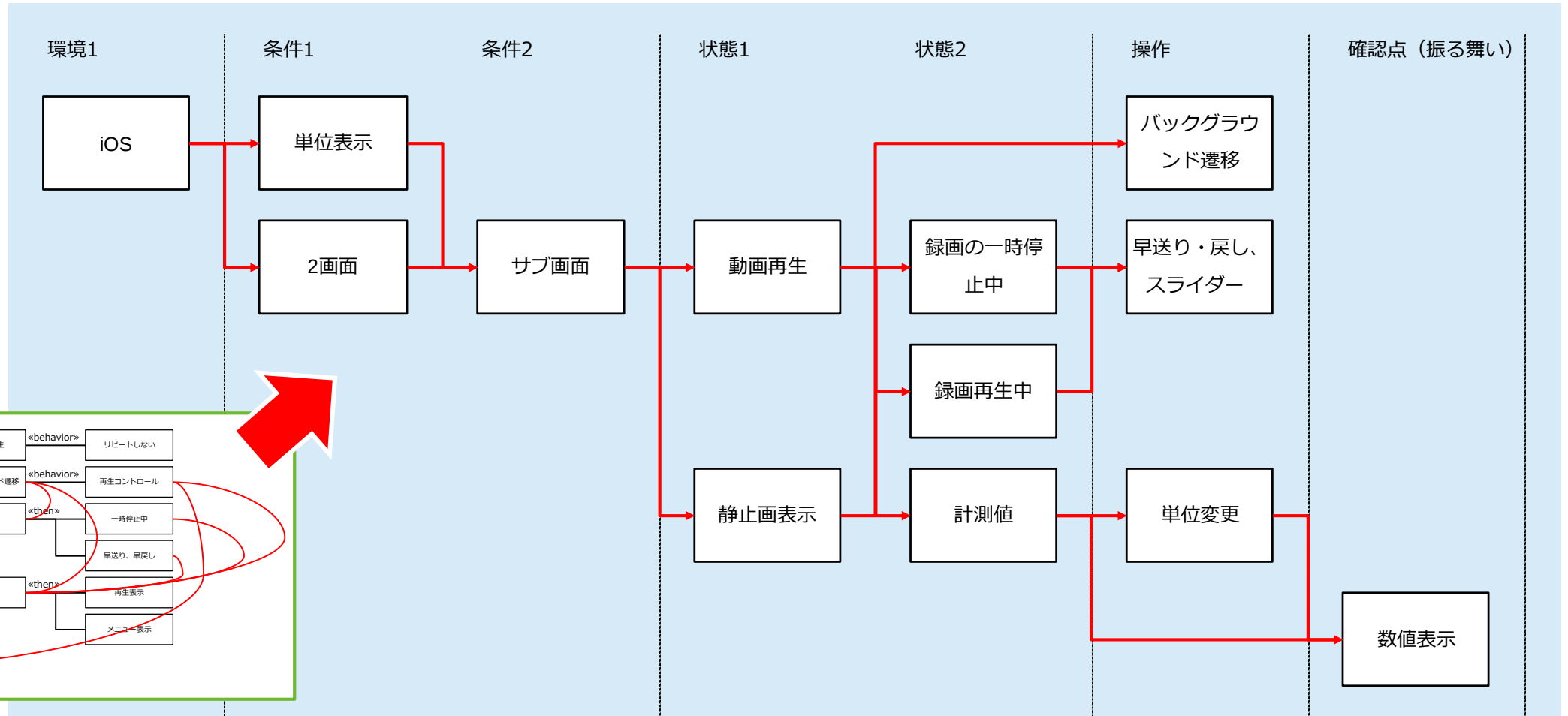
③つながりのない要素をつなげる

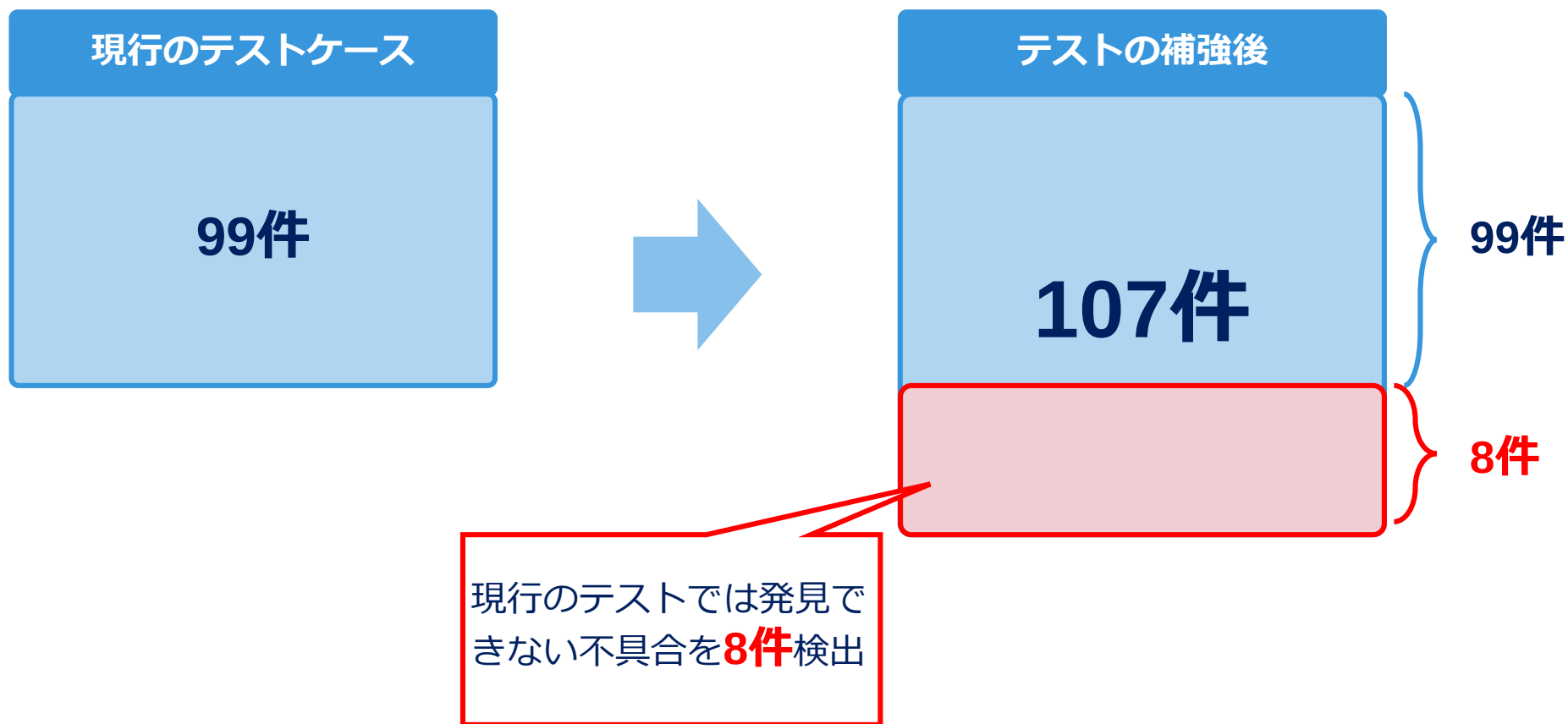


つながりのない要素をつなげるコツ



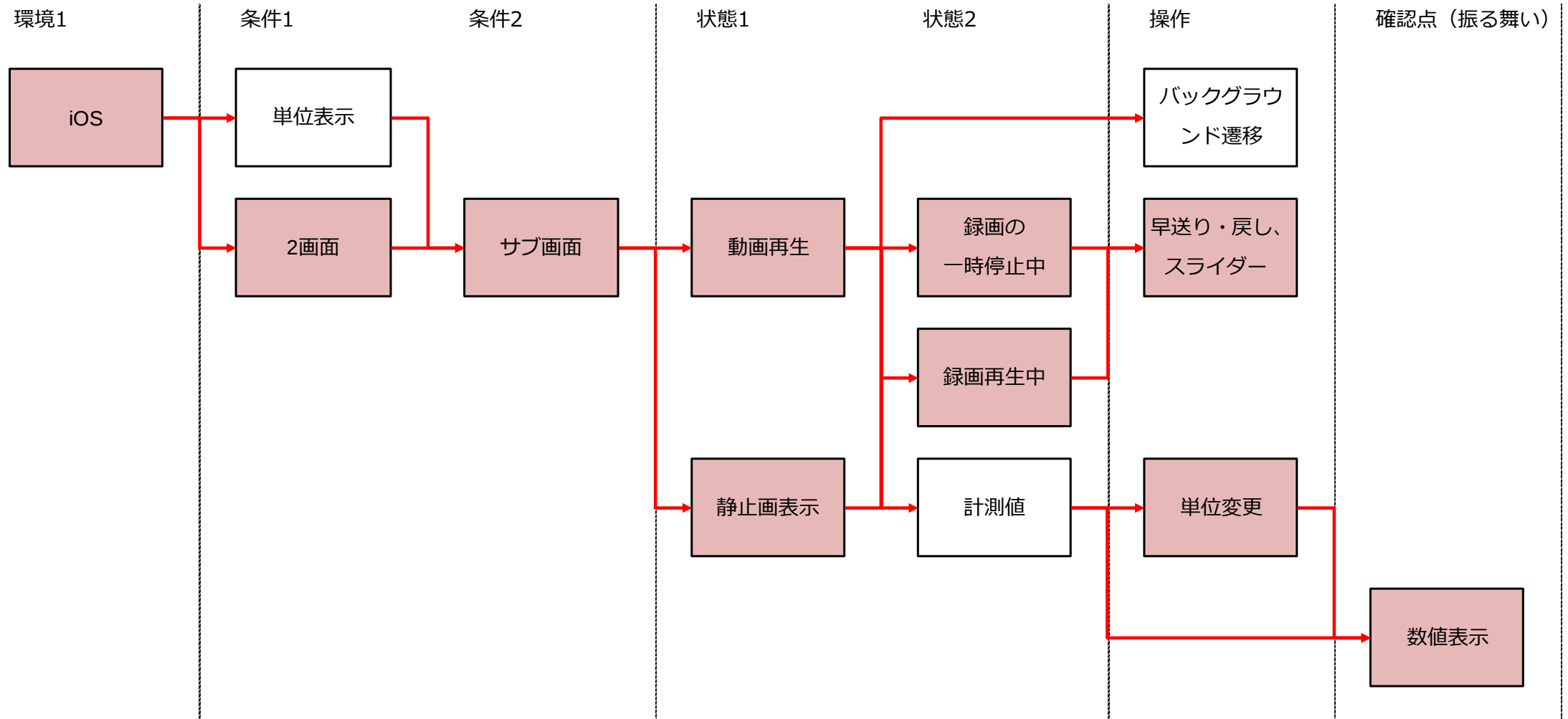
④ フレームワークに当てはめる





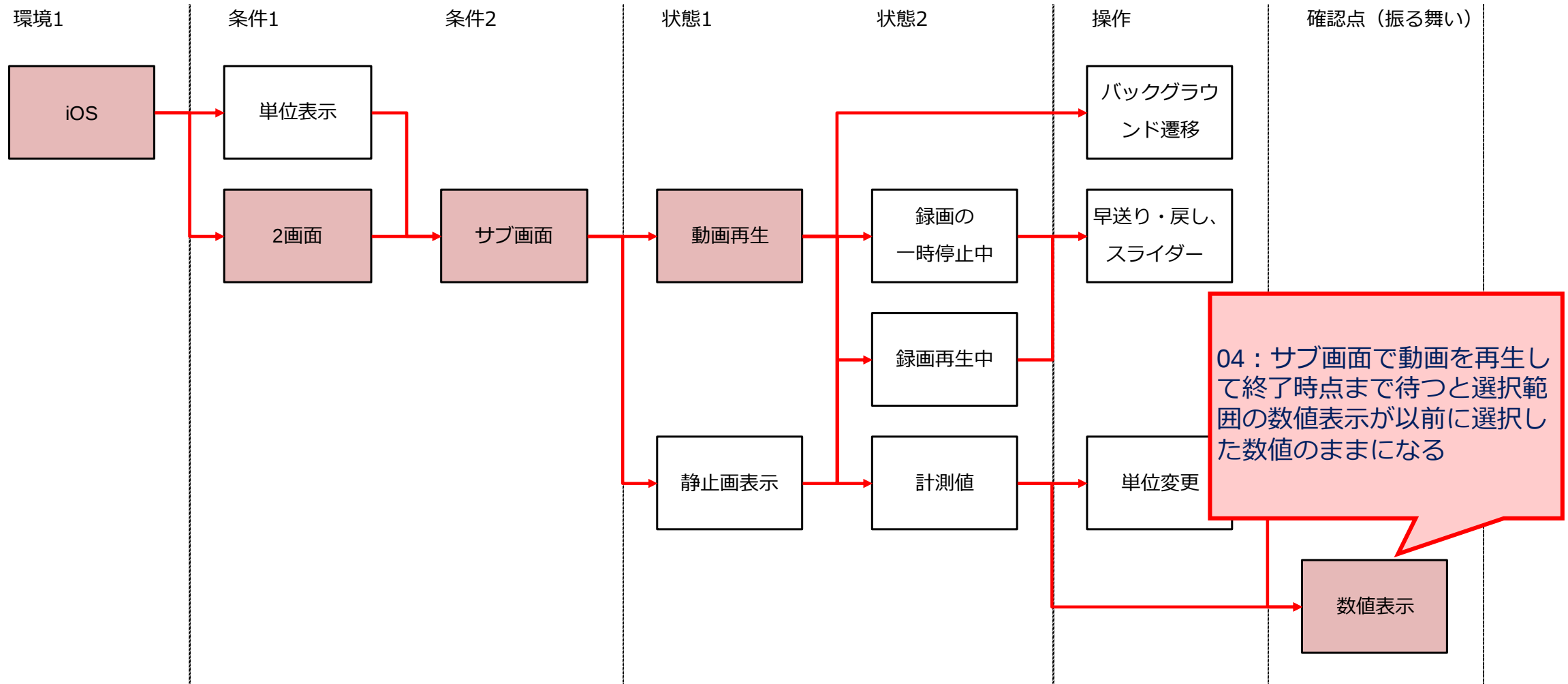
発見した不具合

VERISERVE



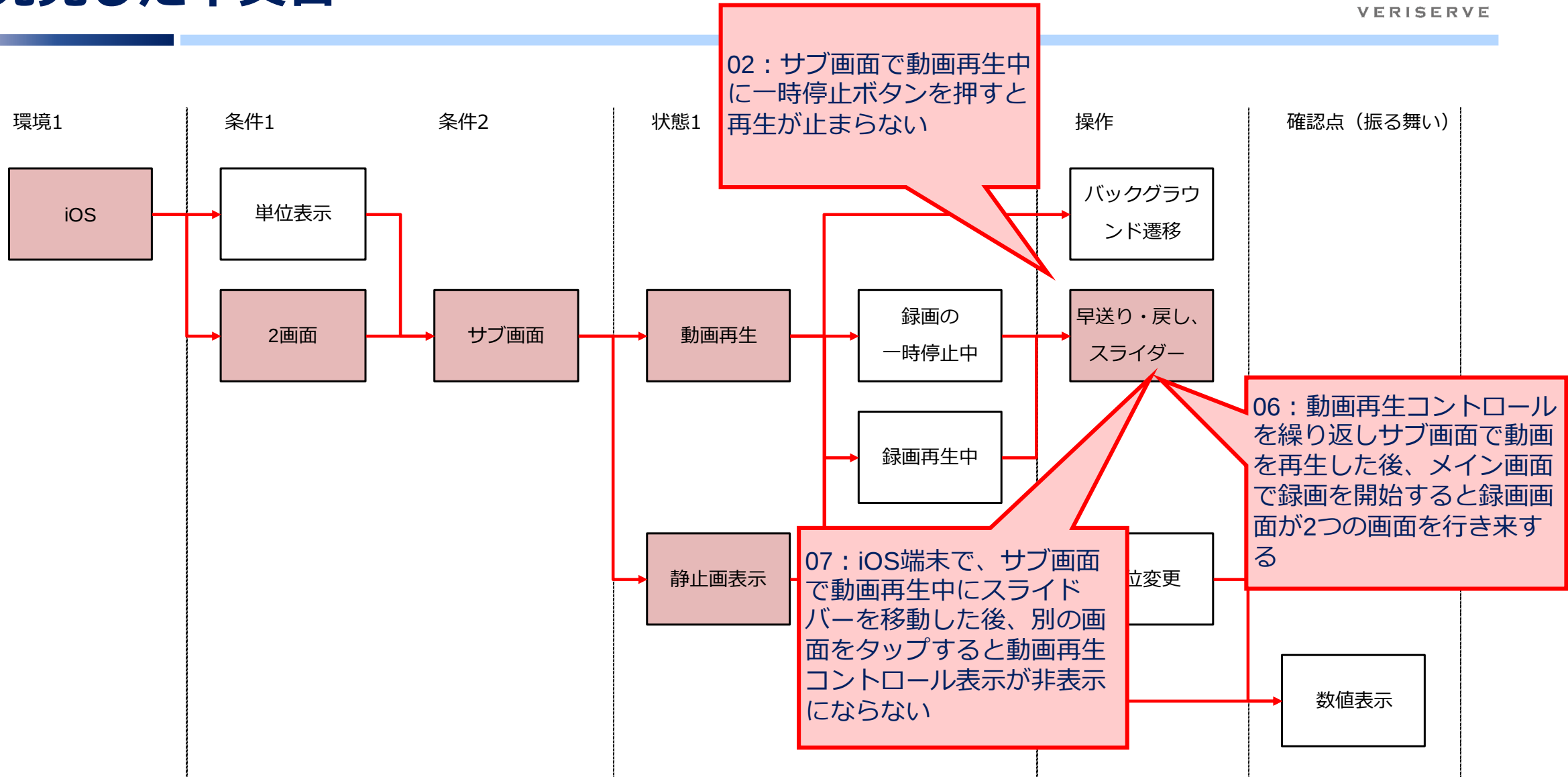
発見した不具合

VERISERVE



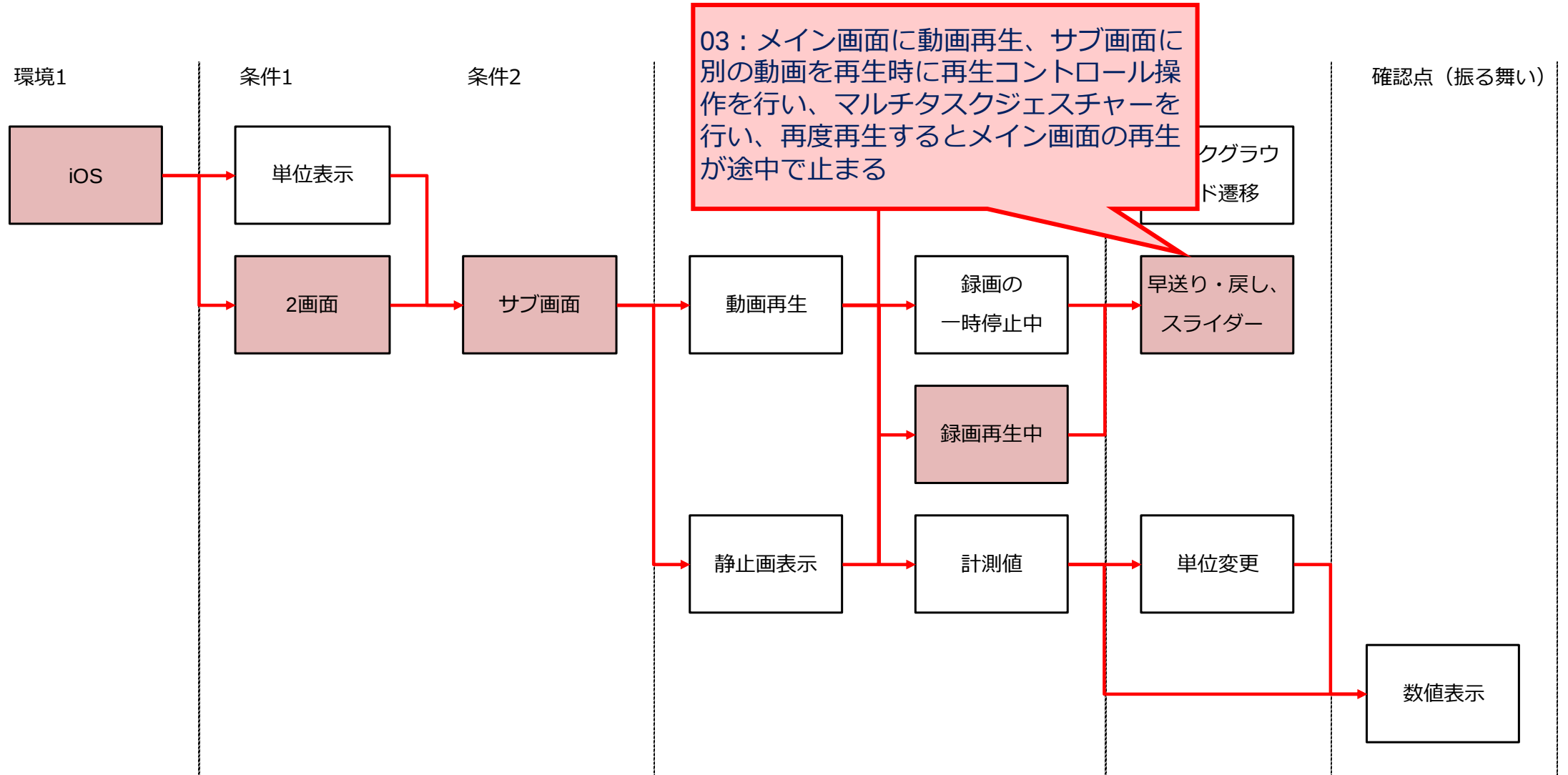
発見した不具合

VERISERVE



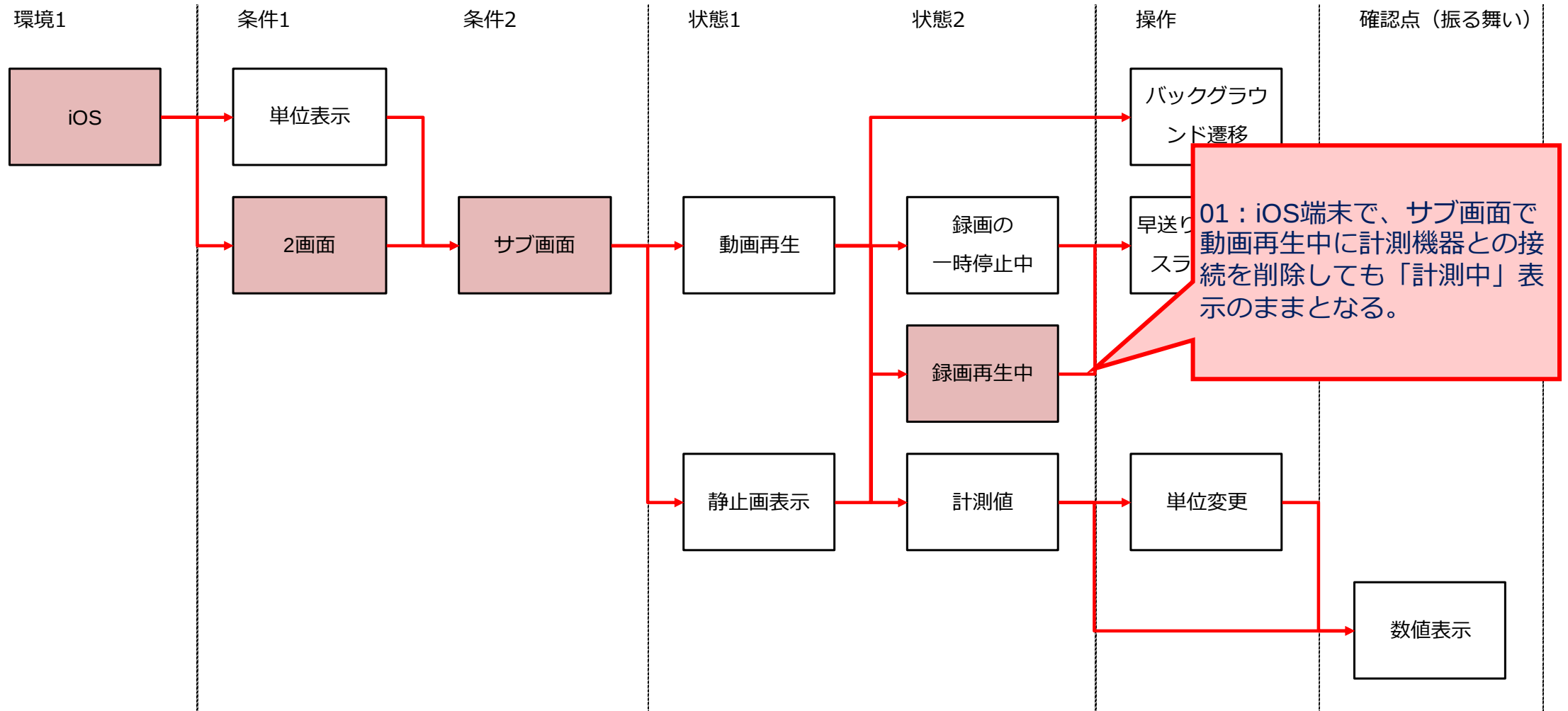
発見した不具合

VERISERVE



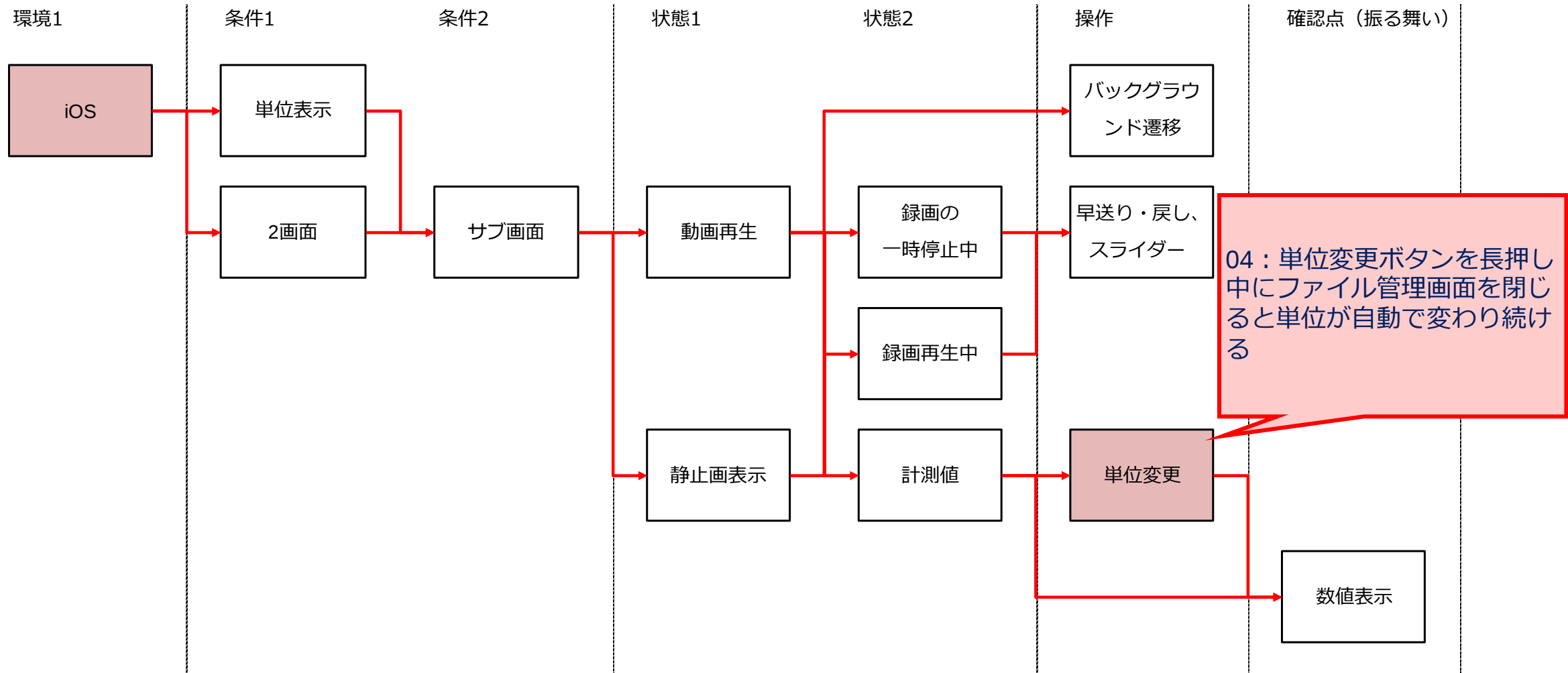
発見した不具合

VERISERVE



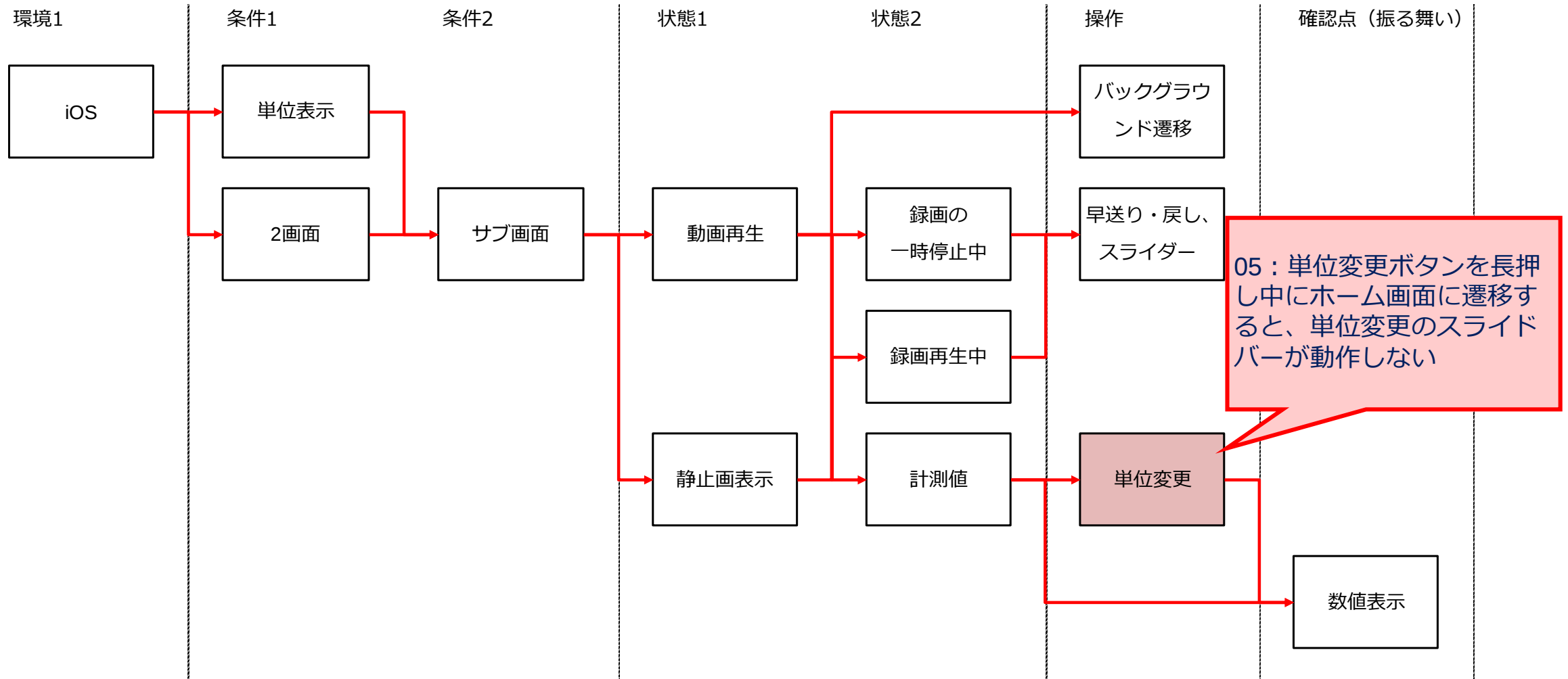
発見した不具合

VERISERVE



発見した不具合

VERISERVE



- 次のテストのための分析としても利用できる。
- テスト実行者によって発見された不具合が多いな、と感じたときに有効
 - 不具合の発見がテスト実行者にほとんど依存していない場合は有効ではない
- 異なる手法で設計されたテストケースに対して有効
 - NGTで設計されたテストケースに対しては有効ではない
- NGTを実践する取っ掛かりとして有効