

JaSST'18 Kansai

S3 B-2) テスト設計セッション

VERISERVE

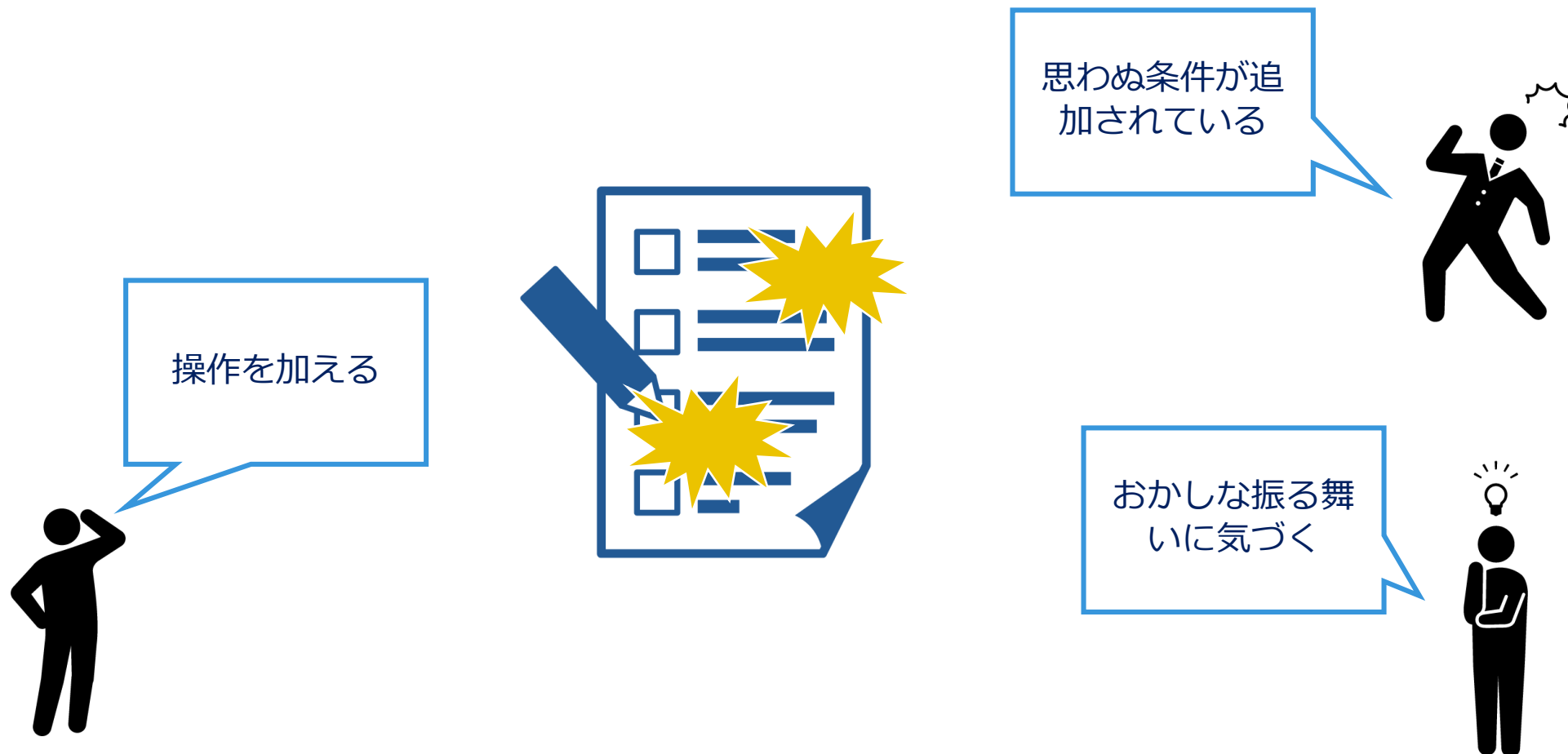
ロジックツリーによる不具合分析からのテストの補強

2018年6月15日

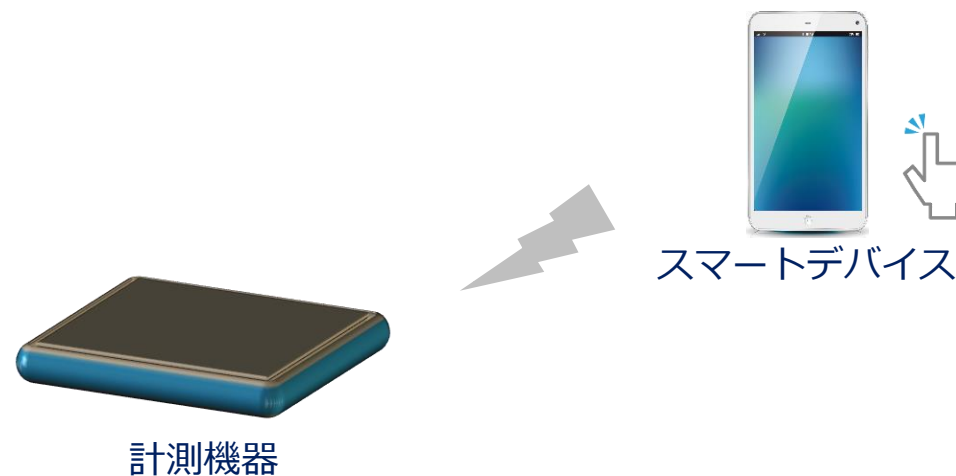
株式会社ベリサーブ
吉川 努

- 背景
- 問題と課題
- 取り組み内容
- **なぜロジックツリーで不具合分析をしようと思ったのか**
- やり方
- ロジックツリーの記法
- 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



N O	項目1	項目2	項目3	テストの意図	事前条件	手順	期待値
1	計測画面	計測画面の遷移	表示継続	30秒以内の復帰で静止画表示状態が保持されていること	1.静止画ファイルを表示する	1.無線LAN通信を切断し、30秒 以内 に再度無線LAN通信を接続する	静止画表示状態であること



N O	項目1	項目2	項目3	テストの意図	事前条件	手順	期待値
1	計測画面	計測画面の 遷移	表示継続	30秒以内の復帰で静止画表示状態が保持されていること	1. 静止画ファイルを表示する	1. 無線LAN通信を切断し、30秒以内に再度無線LAN通信を接続する	静止画表示状態であること

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

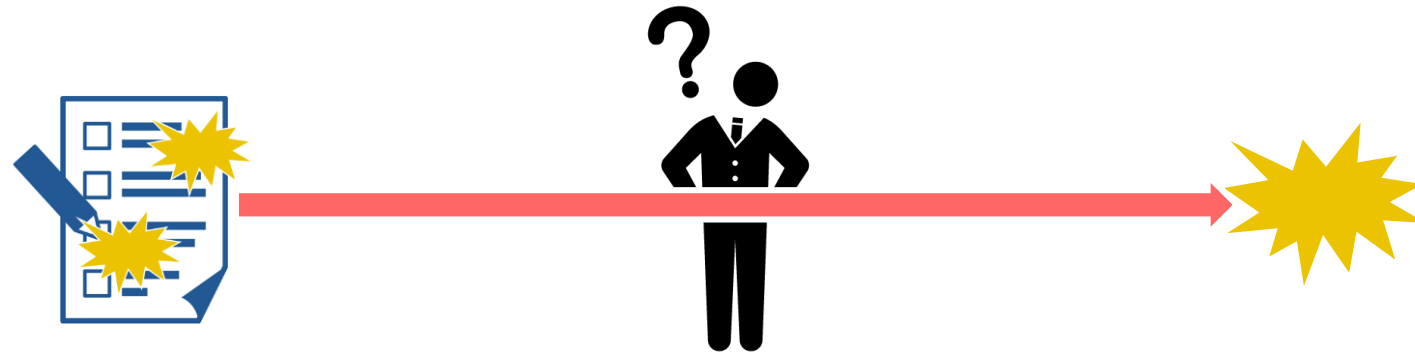
本体電源OFFの17,18秒経過後にアプリ側に無線LAN通信切断のダイアログが出て静止画表示が中断される

N O	項目1	項目2	項目3	テストの意図	事前条件	手順	期待値
1	計測画面	計測画面の 遷移	表示継続	30秒以内の復帰で静止画表示状態が保持されていること	1. 静止画ファイルを表示する	1. 無線LAN通信を切断し、 30秒以内 に再度無線LAN通信を接続する	静止画表示状態であること

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

本体電源OFFの**17,18秒経過後**にアプリ側に無線LAN通信切断のダイアログが出て静止画表示が中断される

17秒経過させなければ不具合は見逃されていた



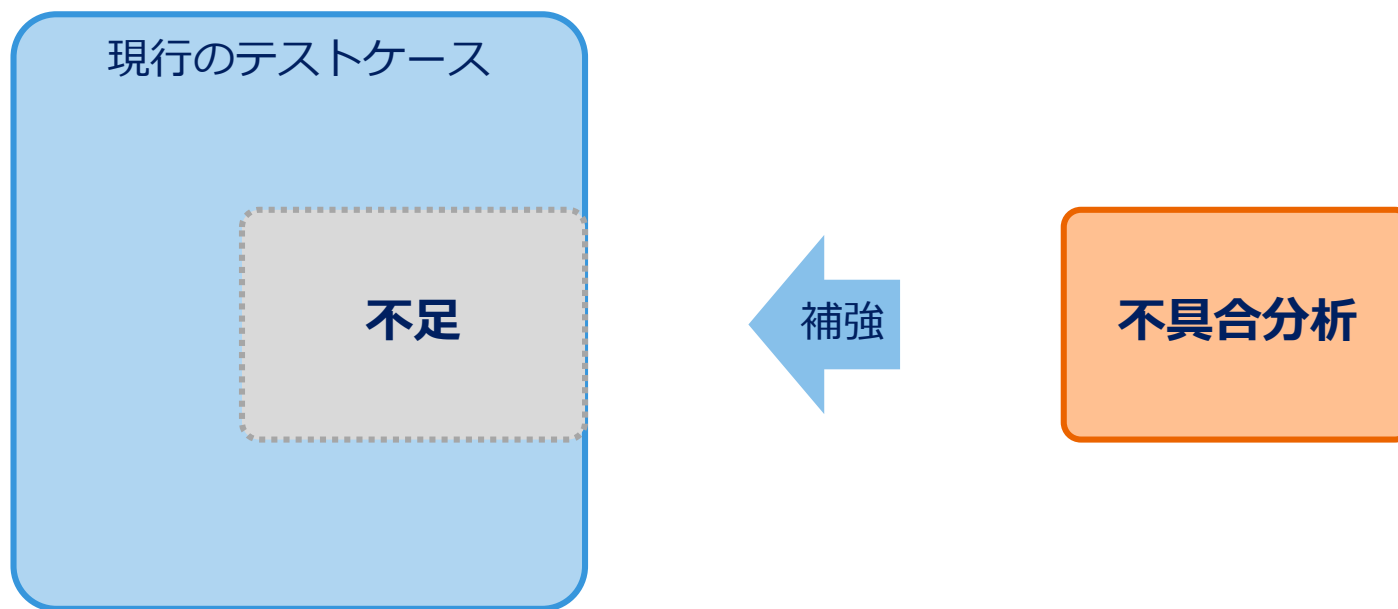
問題

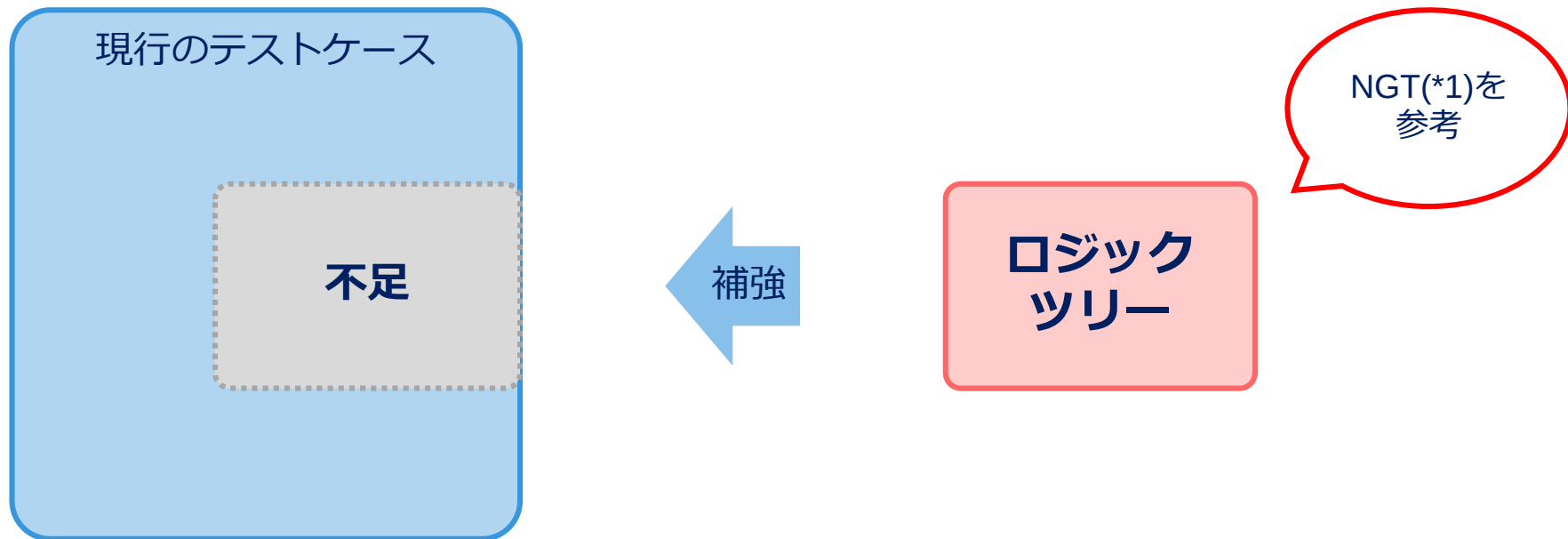
不具合の発見がテスト実行者に依存している

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







*1 : 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

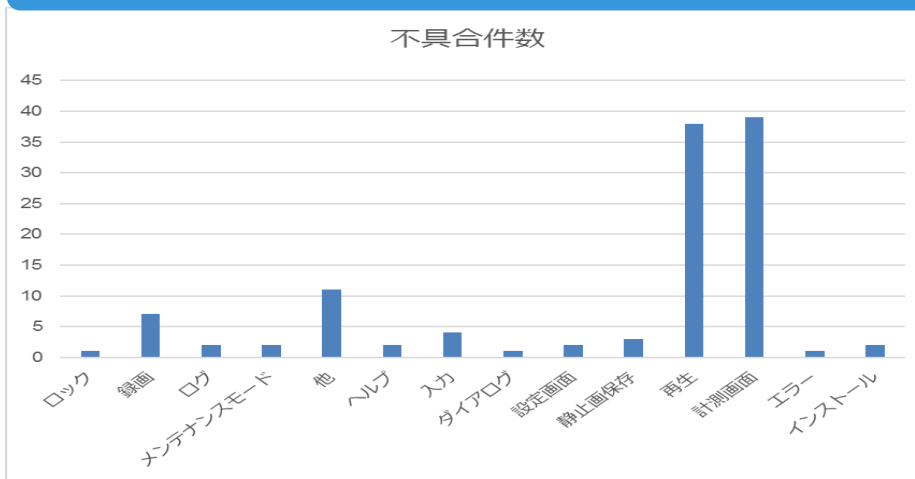
出典 : 「テスト観点に基づくテスト開発方法論 VSTePの概要」VSTeP.130510.color.pdf

なぜロジックツリーで不具合分析をしようと思ったのか

VERISERVE

他の不具合分析のやり方では納得できなかった（１）

機能に対する傾向を分析するやり方



傾向の出た機能に対して
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

なぜロジックツリーで不具合分析をしようと思ったのか

VERISERVE

他の不具合分析のやり方では納得できなかった（２）

不具合の要素を抽出して分析するやり方

NO	タイトル	手順	内容	キーワード
01	バックグラウンドへ遷移・・・	・・・	・・・	バックグラウンド
02	バッテリー切れで電源・・・	・・・	・・・	電池切れ
03	静止画表示中に・・・	・・・	・・・	通信断
...	・・・	・・・	・・・	再生

キーワードが羅列されるだけで
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

1) ロジックツリーで不具合分析する

不具合分析にロジックツリーを使用して不具合の要素を抽出し、その要素を結びつける

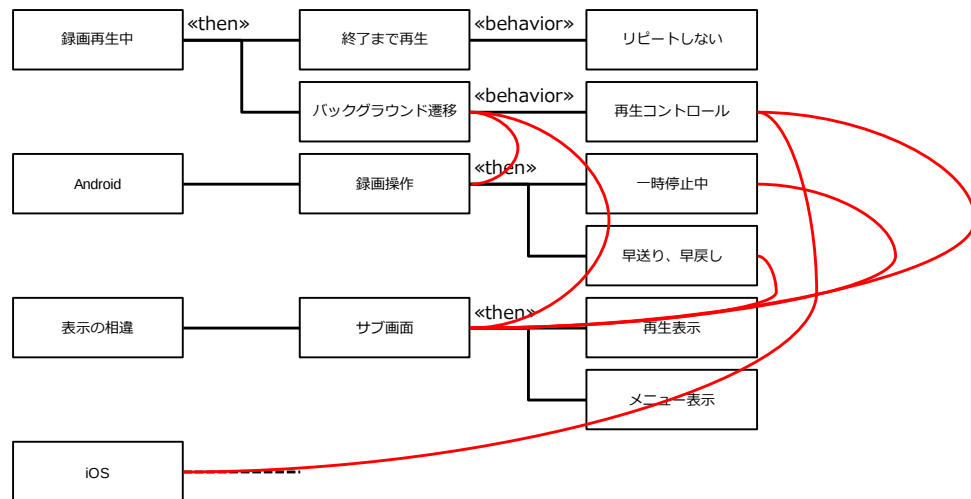


図 ロジックツリーで分析した例

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

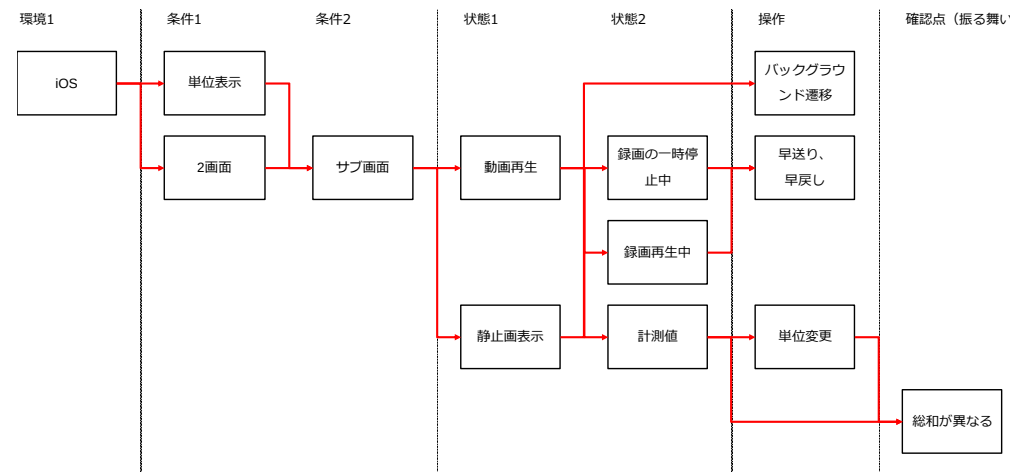


図 フレームワークの例

1) ロジックツリーによる不具合分析

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

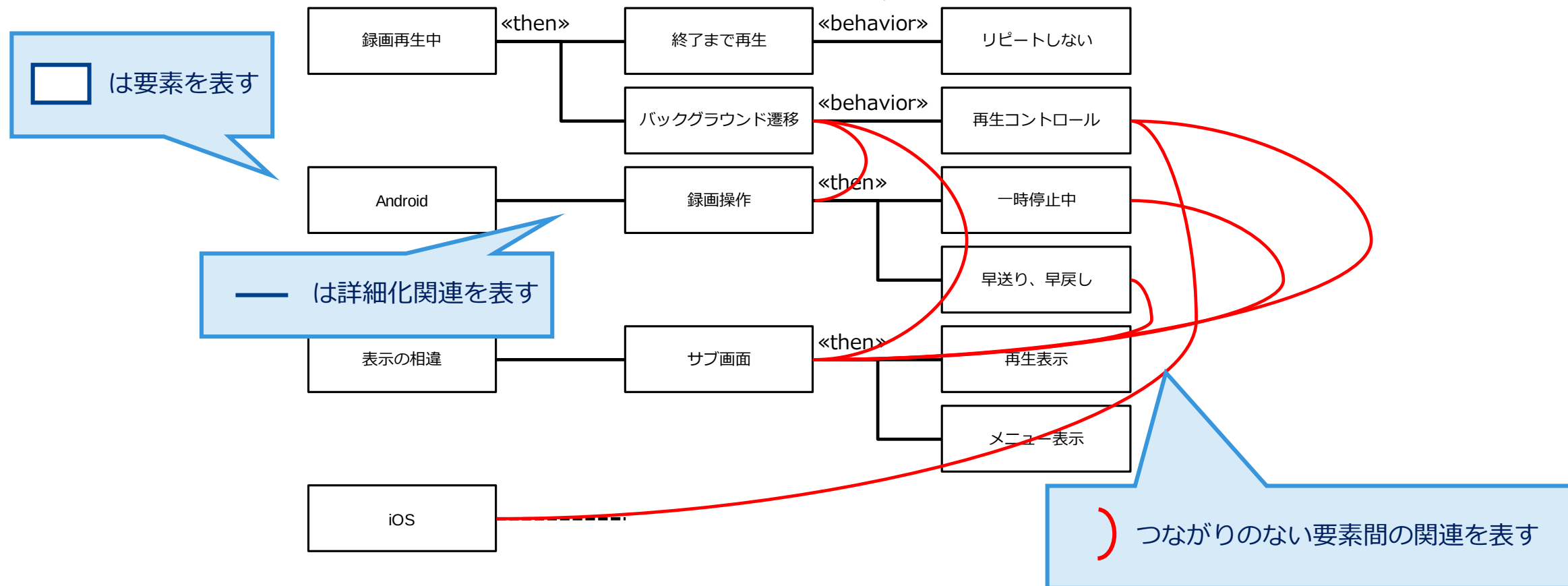
④フレームワークに
当てはめる

ロジックツリーの記法

VERISERVE

➤ 要素を中心に、**階層的**に記述していく

《 》は要素間の関連性を表す



参考：「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

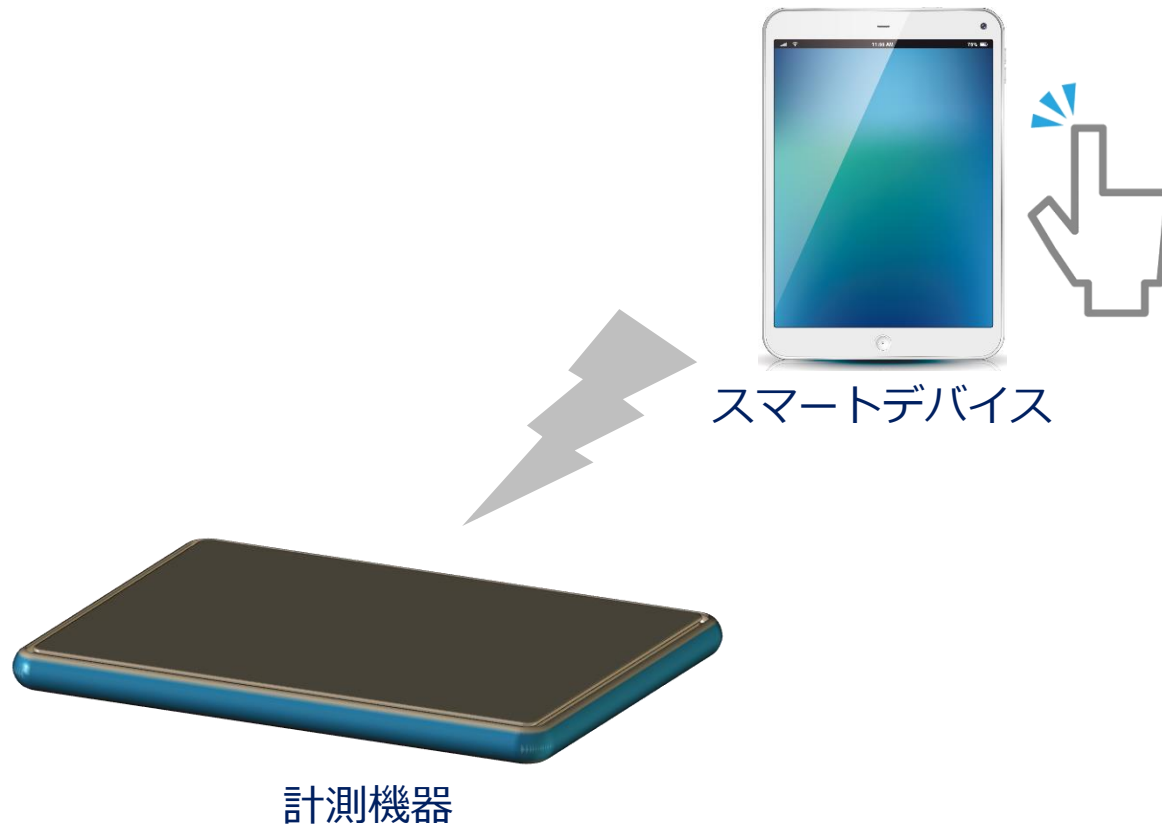
要素の関連性に着目

要素間の関連性を明確にすることで羅列だけでは見えないものを見えるようにした

- «and» : 左側の要素と右側の要素を合わせて条件となる関連性を示す。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連性を示す。順序を表す。
- «after» : 左側の要素の前提条件が右側の要素となる関連性を示す。逆順を表す。
- «behavior» : 右側の要素が不具合の現象を表す。

参考：「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

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

適用事例

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

④フレームワークに
当てはめる

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。

動画再生中

《then》

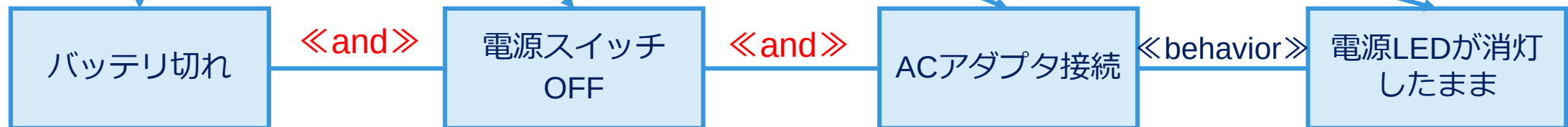
バックグラウンド遷移

《behavior》

再生コントロールが正しく動作しない

②要素をつなげる 《and》の場合

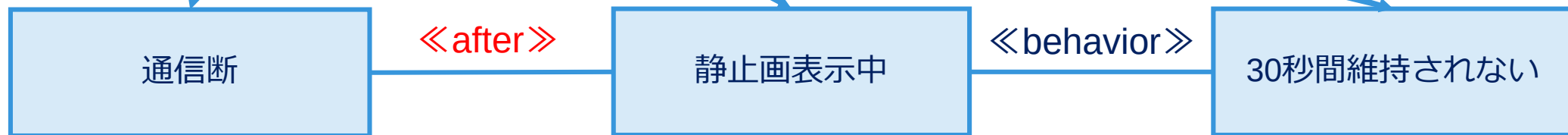
N O	タイトル	手順	内容
0 2	バッテリー切れで電源スイッチOFFの本体にACアダプタを接続 して数分の後、点滅していた電源LEDが消灯したままとなる	【前提条件】 ・ 電源SW：OFF ・ ACアダプタ：接続 ・ バッテリー充電：無 【手順】 1.前提条件セット後、数分間経過させる	・ ACアダプタ接続している状態で、電源LEDが消灯したままとなり充電状態判別が不可となる



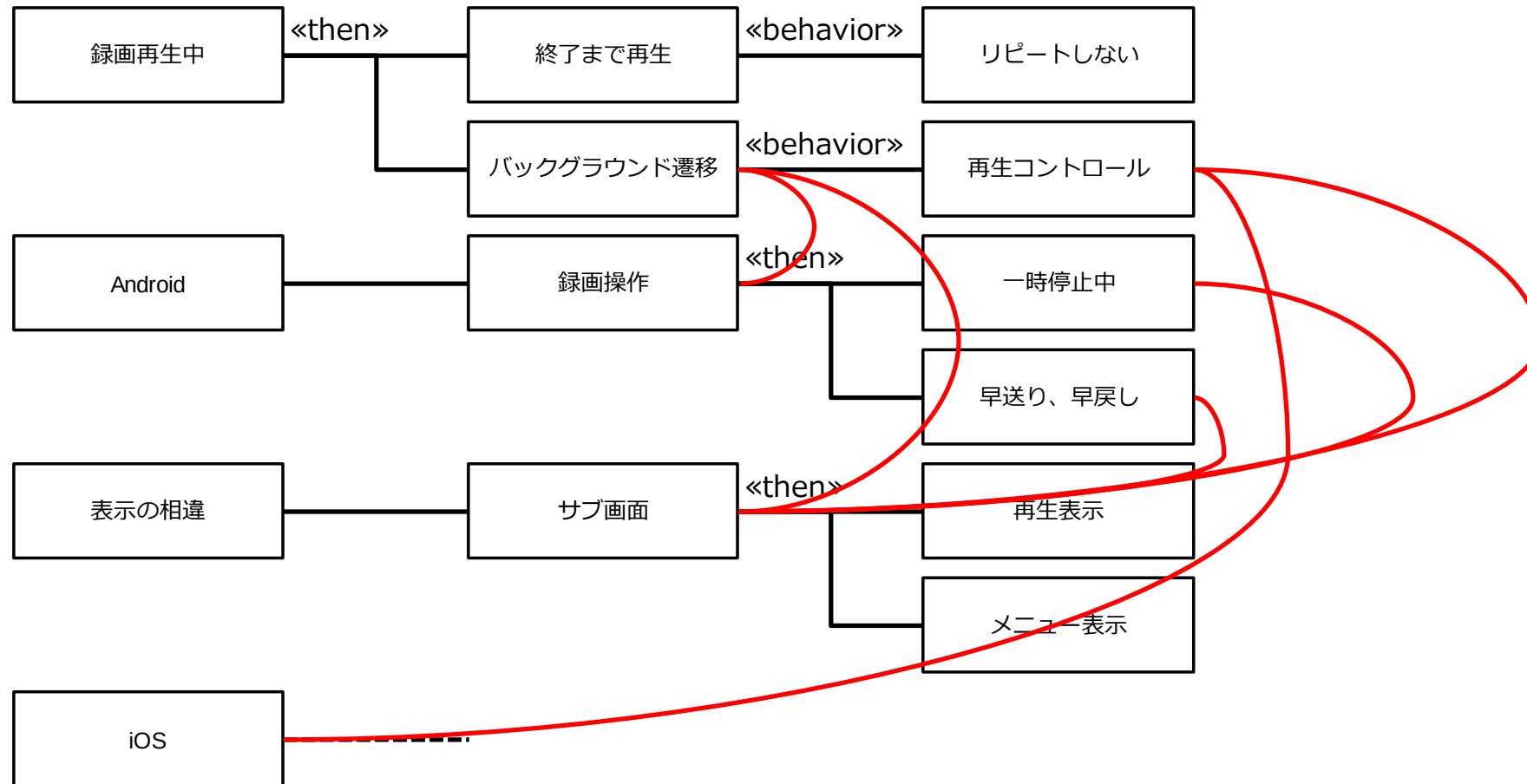
②要素をつなげる 《after》の場合

N O	タイトル	手順	内容
0 3	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移移する（静止画表示が維持されない）	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

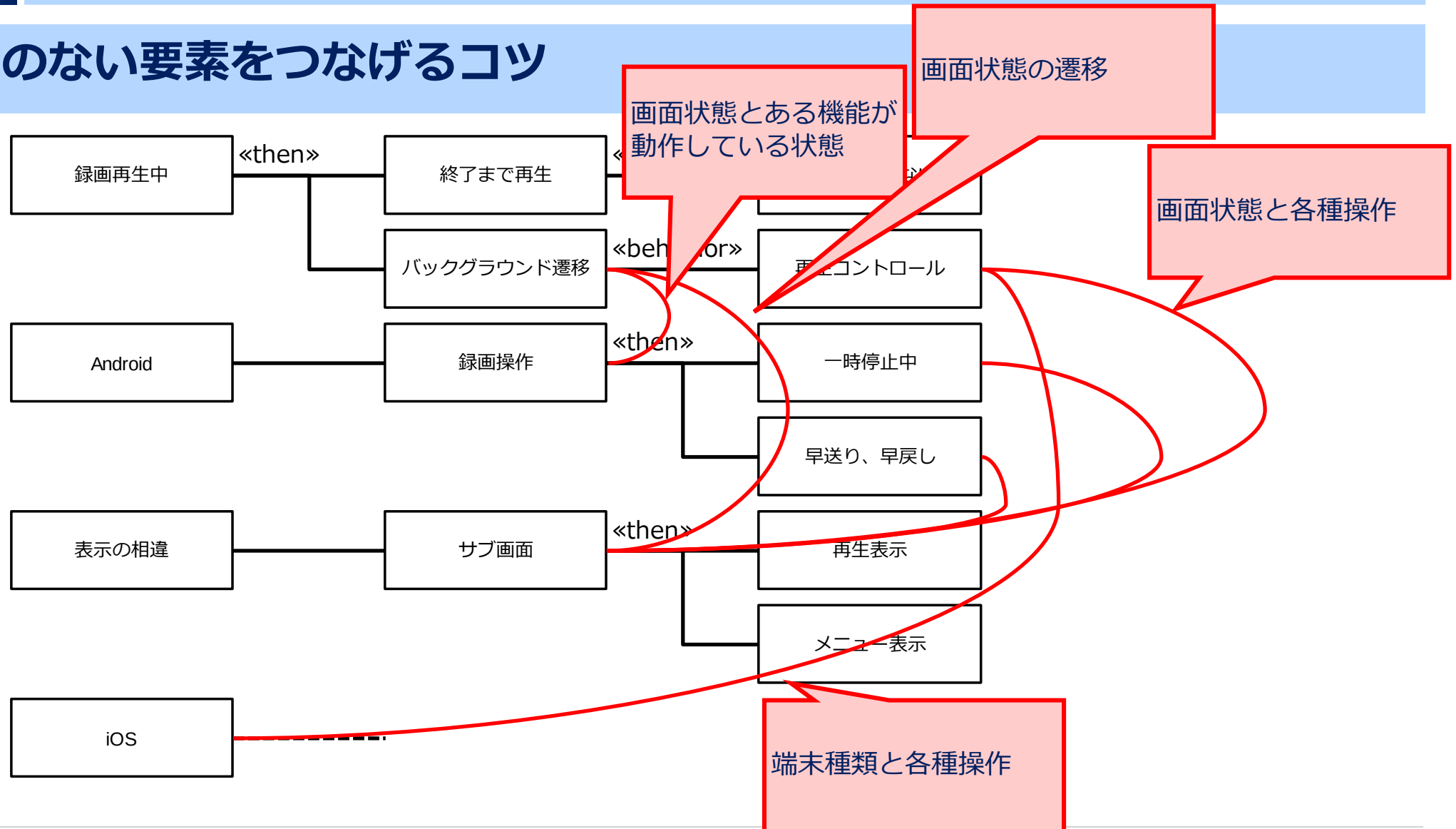
30秒以内とは状態を維持する時間に関する仕様



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



適用事例

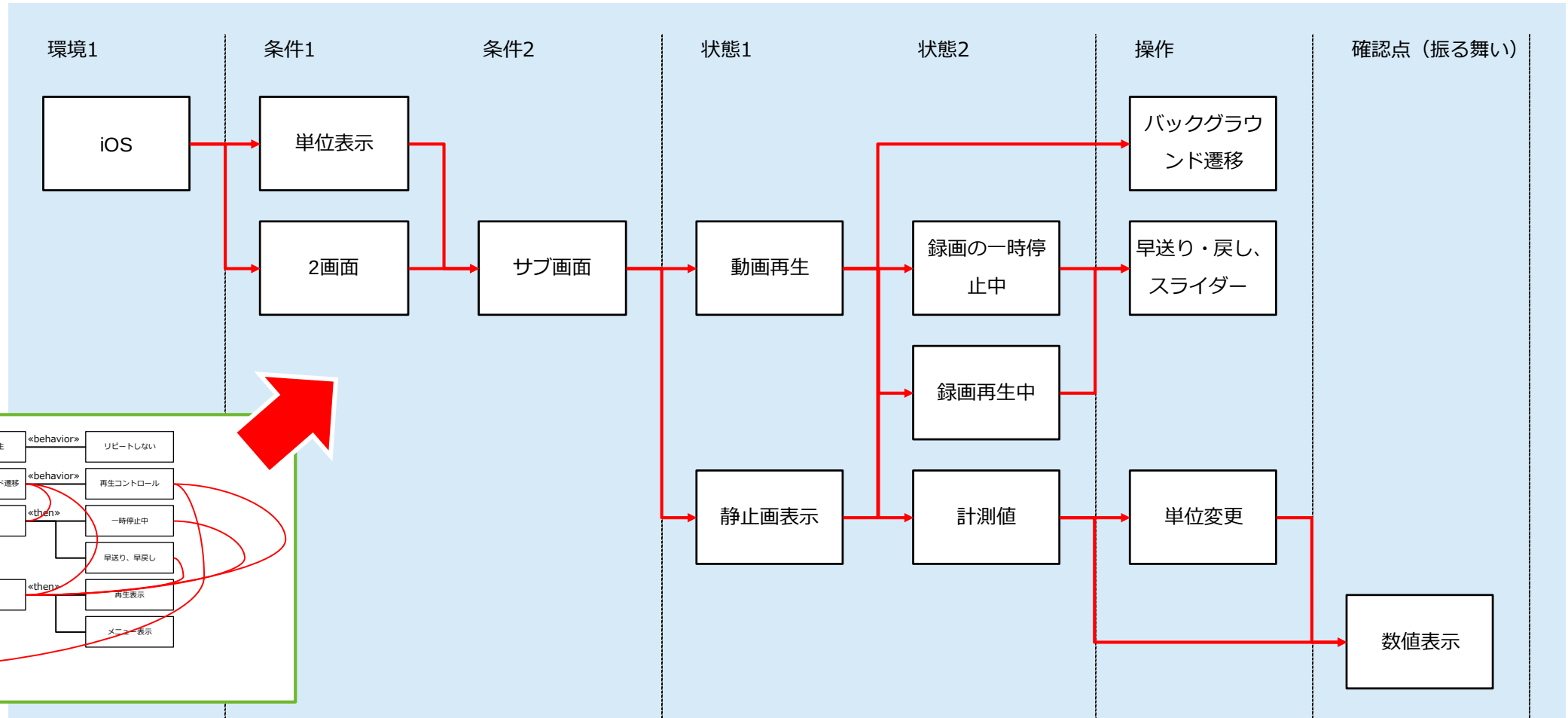
①要素を
抜き出す

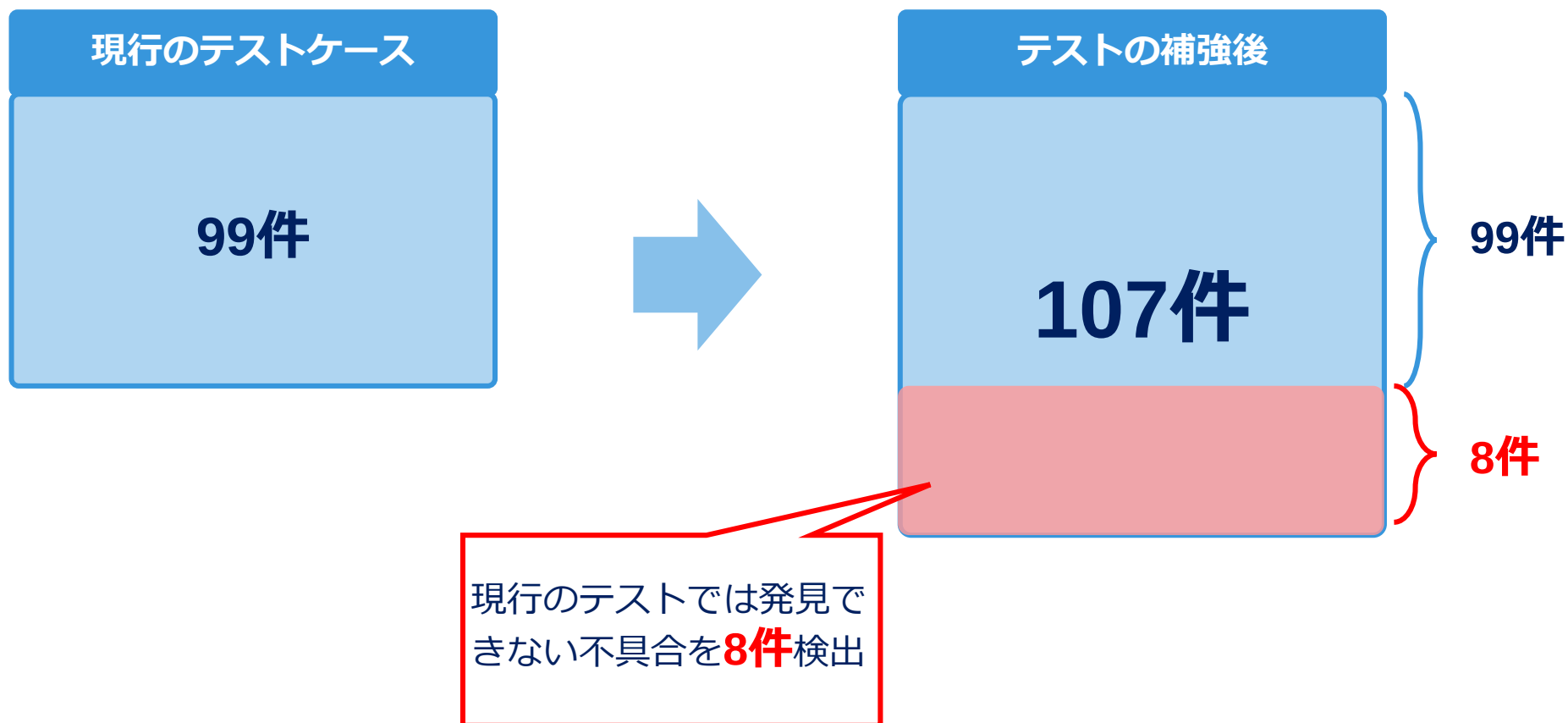
②要素を
つなげる

③つながりのない
要素をつなげる

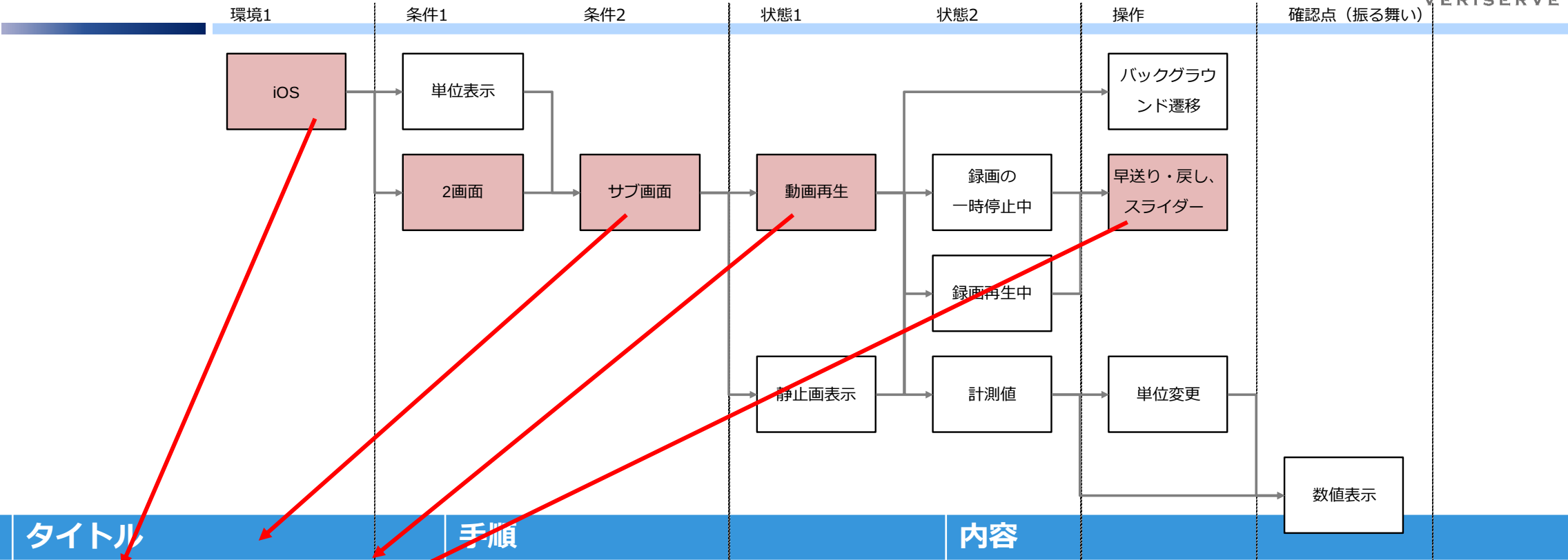
④フレームワークに
当てはめる

④フレームワークに当てはめる





発見した不具合例



NO	タイトル	手順	内容
100	iOS端末で、サブ画面で動画再生中にスライダーを移動すると、動画再生コントロール表示が非表示にならない	【前提条件】 サブ画面で動画を再生する 【手順】 - 再生コントロール上のスライダーを移動する - 計測画面部分をタップする - 動画再生コントロールの表示状態を確認する	- タップ時に、動画再生コントロールが一瞬消えるが、すぐに表示される - 次にタップすると動画再生コントロールは非表示となる - メイン画面では発生しない - Android端末では発生しない

➤ このやり方の有効な場面

- 次のテスト分析に活用できる
- テスト実行者に依存して発見される不具合が多いな、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
 - ◆ ロジックツリーで設計（分析）されたテストケースに対しては有効ではない

➤ 納得するために

- “手法ありき”ではなく実現したいことのために手法を適用する

参考資料 : 「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