

ソフトウェア 構造解析を用いた 結合テスト技法 の紹介

JaSST Hokkaido 2019

RICOH
imagine. change.

 **BACKSLASH**
DESIGN

リコーITソリューションズ株式会社

狩野薫・森孝司・増子聡

ビースラッシュ株式会社

山田大介

2019年8月30日

自己紹介

- 名前：狩野 薫
- 所属：リコーITソリューションズ株式会社
エンベデッドソリューション事業部
事業推進センター IT検証部
- テスト業務歴7年目
テストエンジニア 兼 テストマネージャ
 - クラウドシステム (5年)
 - ソリューションシステム (2年)
- 日頃思うこと：
「楽に楽しくテストしたい！」



- 本日本日お伝えしたいこと
 - テスト現場の困りごとを解決し、テストを楽にするために私たちが取り組んでいる結合テスト技法の紹介
- お伝えしたい方々
 - テストに悩んでいるテストエンジニア
 - 例えば・・・
 - 総合テストで障害が多発し、テストが計画通りに進まない（でも、やることや納期はかわらない）
 - 結合テストで期待した通りに欠陥が検出できない
 - 結合テストでテスト範囲の絞り込みに苦労している

1. テスト現場の困りごとから考えるテストアプローチ
2. 欠陥が発生するリスクをソースコードから評価する
3. リスクの評価結果をもとにテスト範囲を決定する



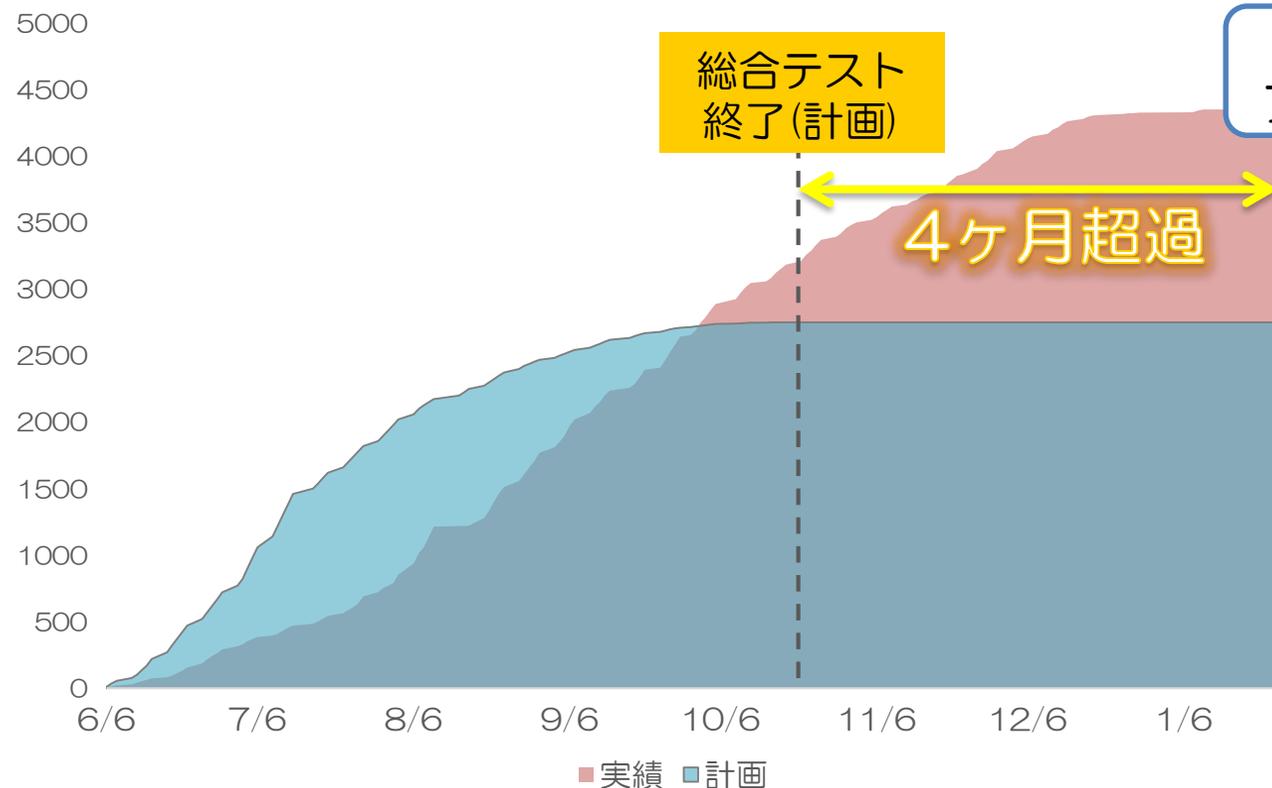
1.テスト現場の困りごとから考える テストアプローチ



テスト現場の困りごと【その①】

- 結合テストまでに検出すべき欠陥が総合テストで多発！

某プロジェクトの累積障害発生件数の推移



開始直後に
ブロッキング！

テストできない
機能が多数！



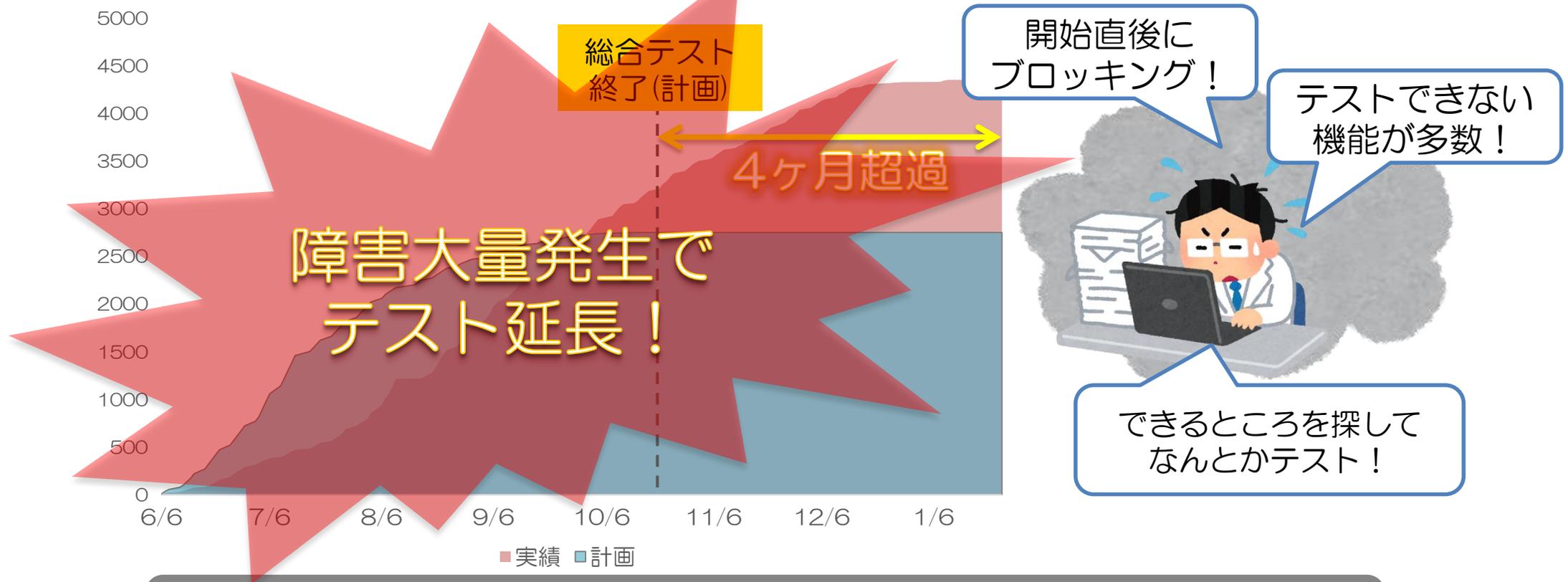
できるところを探して
なんとかテスト！



テスト現場の困りごと【その①】

- 結合テストまでに検出すべき欠陥が総合テストで多発！

某プロジェクトの累積障害発生件数の推移



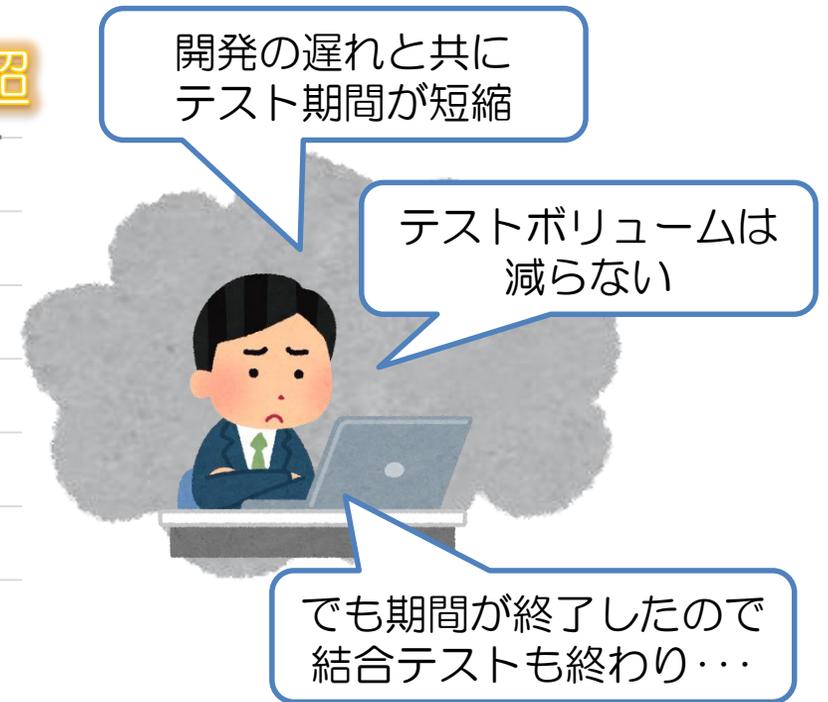
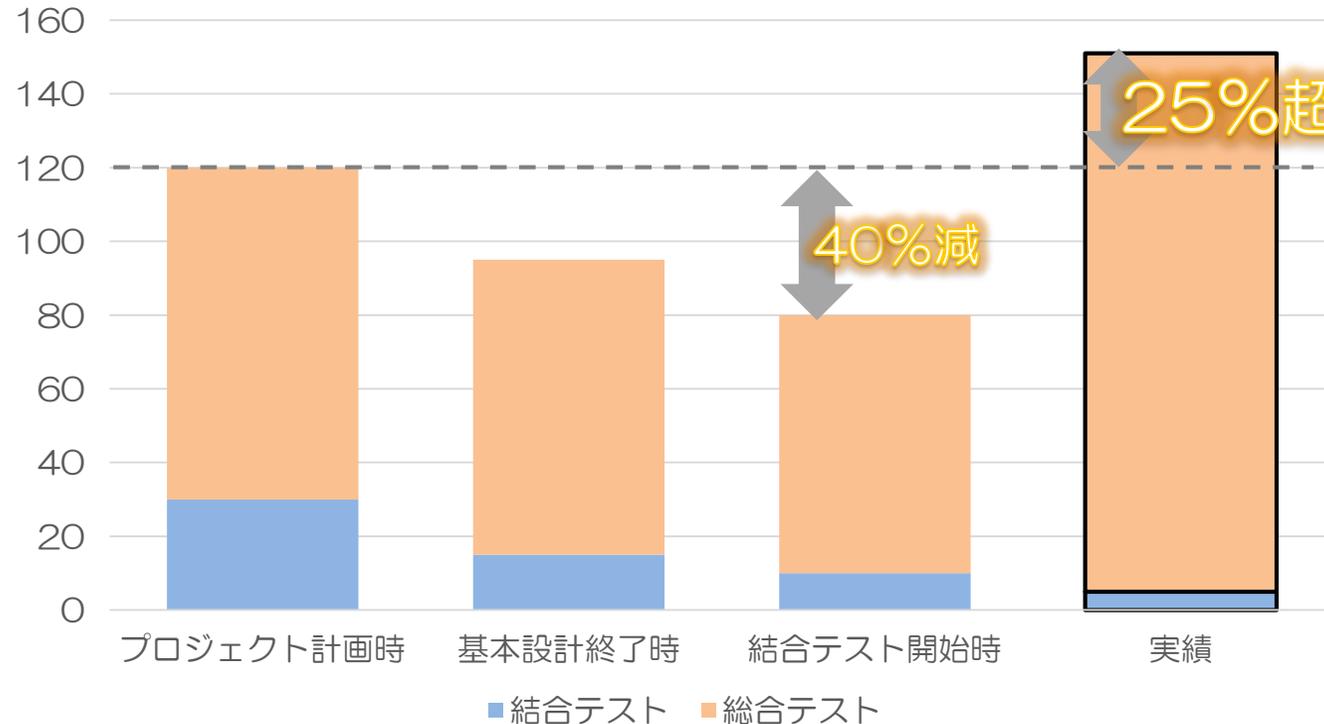
もっと前の工程で欠陥を出せていれば...



テスト現場の困りごと【その②】

- 後工程のテストほど超短期間の欠陥出しが要求される！

某プロジェクトのテスト日数の推移





テスト現場の困りごと【その②】

- 後工程のテストほど超短期間の欠陥出しが要求される！

某プロジェクトのテスト日数の推移



開発の遅れと共に
テスト期間が短縮

テストボリュームは
減らない



でも期間が終了したので
結合テストも終わり…

時間がなくても危険なポイントはテストしたい



テスト現場の困りごと【その③】

■ テスト範囲の絞り込みが難しい



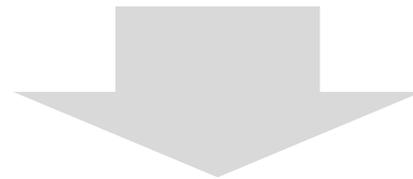
設計者や仕様書が不足してても
テスト範囲を絞りこめないものか...



結合テストで**期待通りに欠陥を出したい**

欠陥が出やすい**ポイントに絞ったテスト**がしたい

設計者や仕様書に**頼らずに入手できる情報**から
テスト範囲を絞りこみたい



解決案

- ソースコードから**欠陥が出やすい箇所を予測**する
- 予測した欠陥発生箇所をもとに、テスト範囲を決定する



2.欠陥が発生するリスクを ソースコードから評価する



欠陥の発生は予測できるの？



ソースコードから得られる情報で
欠陥の発生予測に使えるものはないかしら??

よく使われる結合テスト技法の問題

- 循環的複雑度が大きいコンポーネント中心のテスト

派生開発では、
循環的複雑度が大きい
コンポーネントが
多数存在している

複雑度:高
規模:大



循環複雑度が小さいが、
欠陥が発生している

※色の濃淡は複雑度の度合、
枠の大きさはコンポーネントの規模、を示す

循環複雑度だけでは欠陥の予測には不十分！

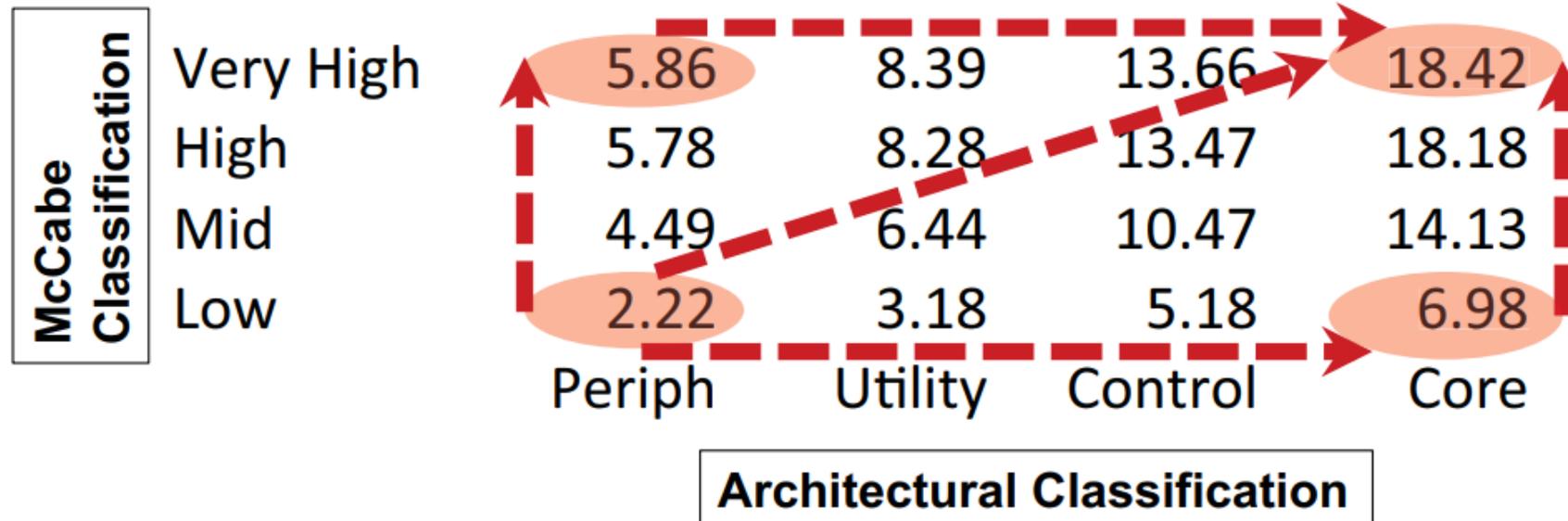
■ 欠陥発生予測の研究

- 複雑度が増すと欠陥密度が増加する
- 複雑度が増すと潜在的コードエラーが増加する

引用：ソフトウェア構造を見れば品質がわかる！構造メトリクスとバグの関係（スライド22）

■ Sturtevantの欠陥予測シミュレーション

- 循環複雑度と構造複雑度より欠陥を予測する



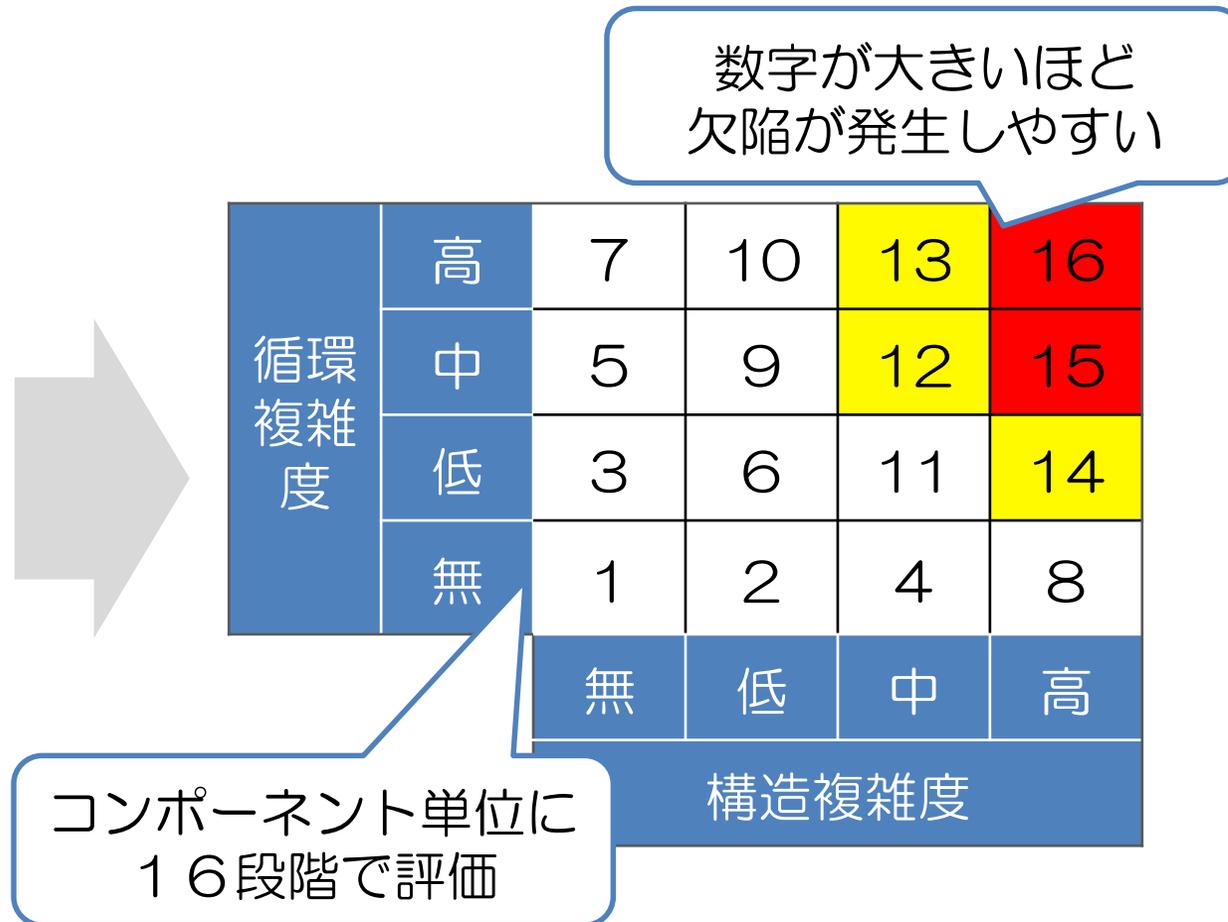
引用：Technical Debt in Large Systems: Understanding the cost of software complexity（スライド39）

欠陥発生リスクの評価方法

- 2つの複雑度を組み合わせてリスクを評価する

循環複雑度	リスク
1~10	無
11~20	低
21~50	中
51~	高

構造複雑度		リスク
FanIn	FanOut	
Low	Low	無
High	Low	低
Low	High	中
High	High	高





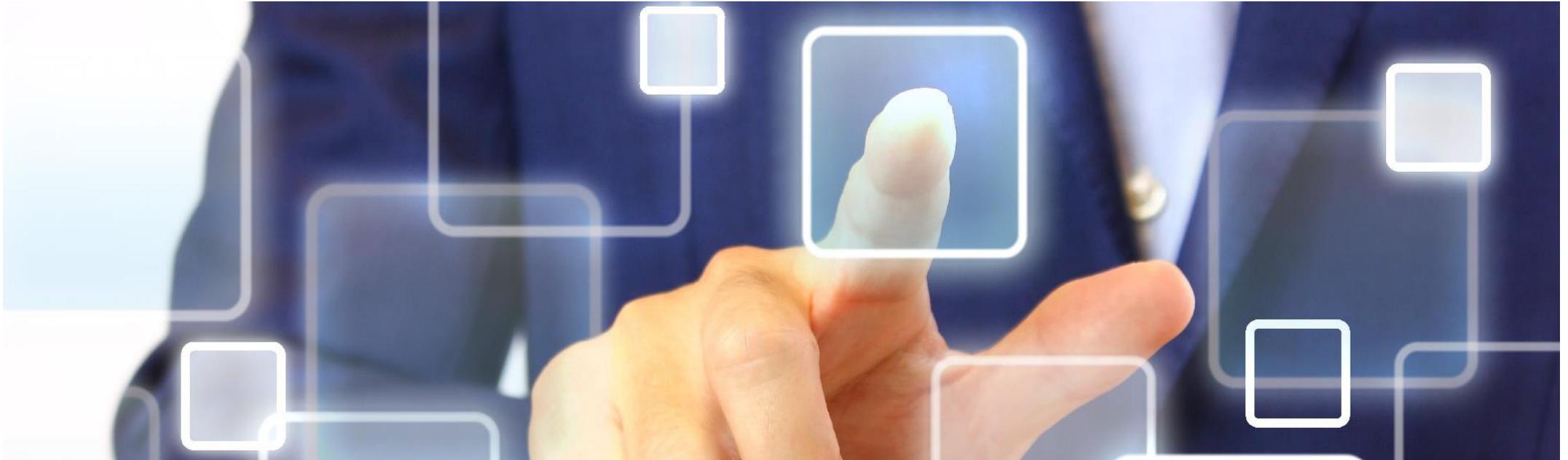
3. リスクの評価結果をもとに テスト範囲を決定する

実例を使って
紹介します



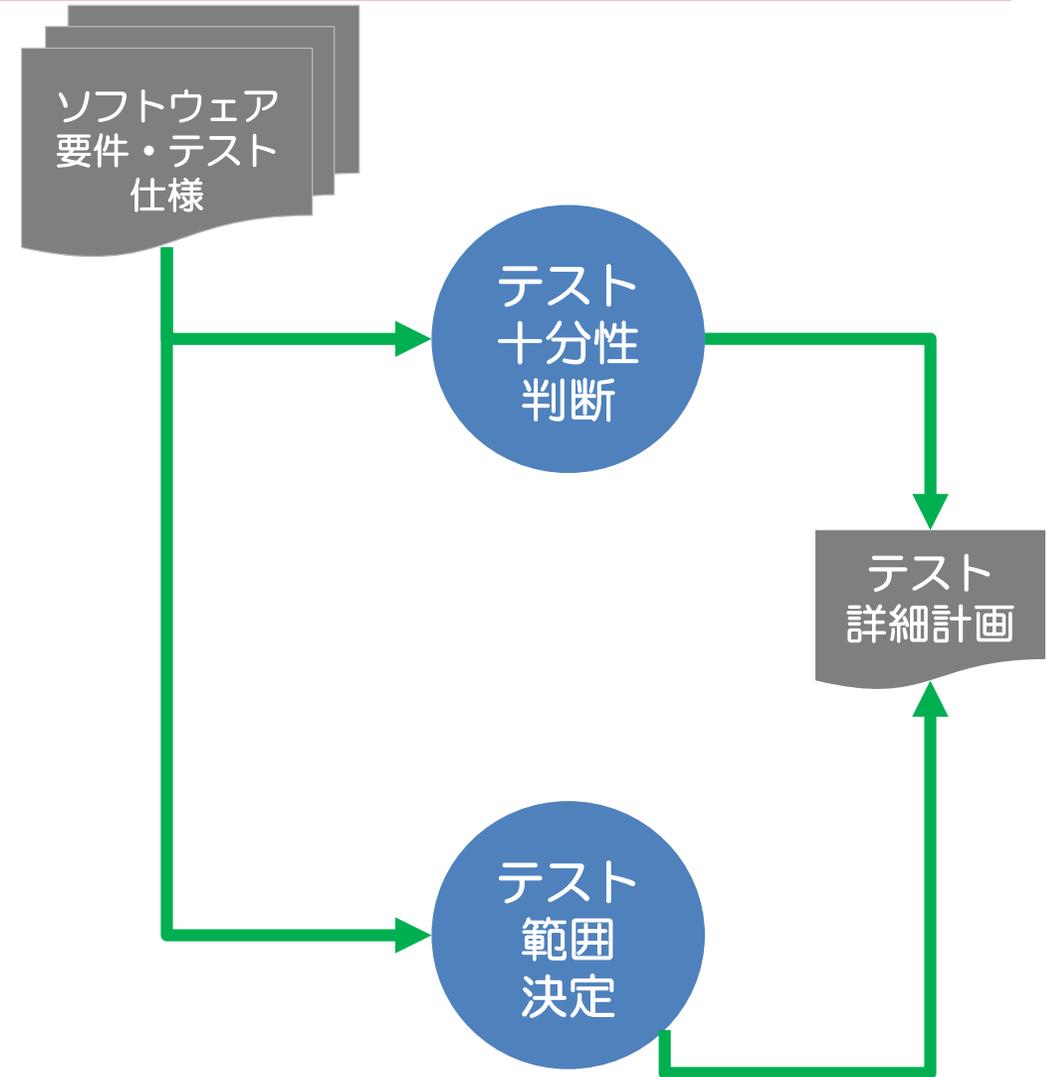
ただ今実践中！

- テスト対象の概要：
 - 製品名：ビースラッシュ(株) AtScope V5.0 (2020年リリース)
 - 特徴：ソースコードから様々な構造図の出力を行うことで、ソフトウェアの設計意図の「発掘」を支援する静的解析ツール。
 - 開発言語：C#.NET

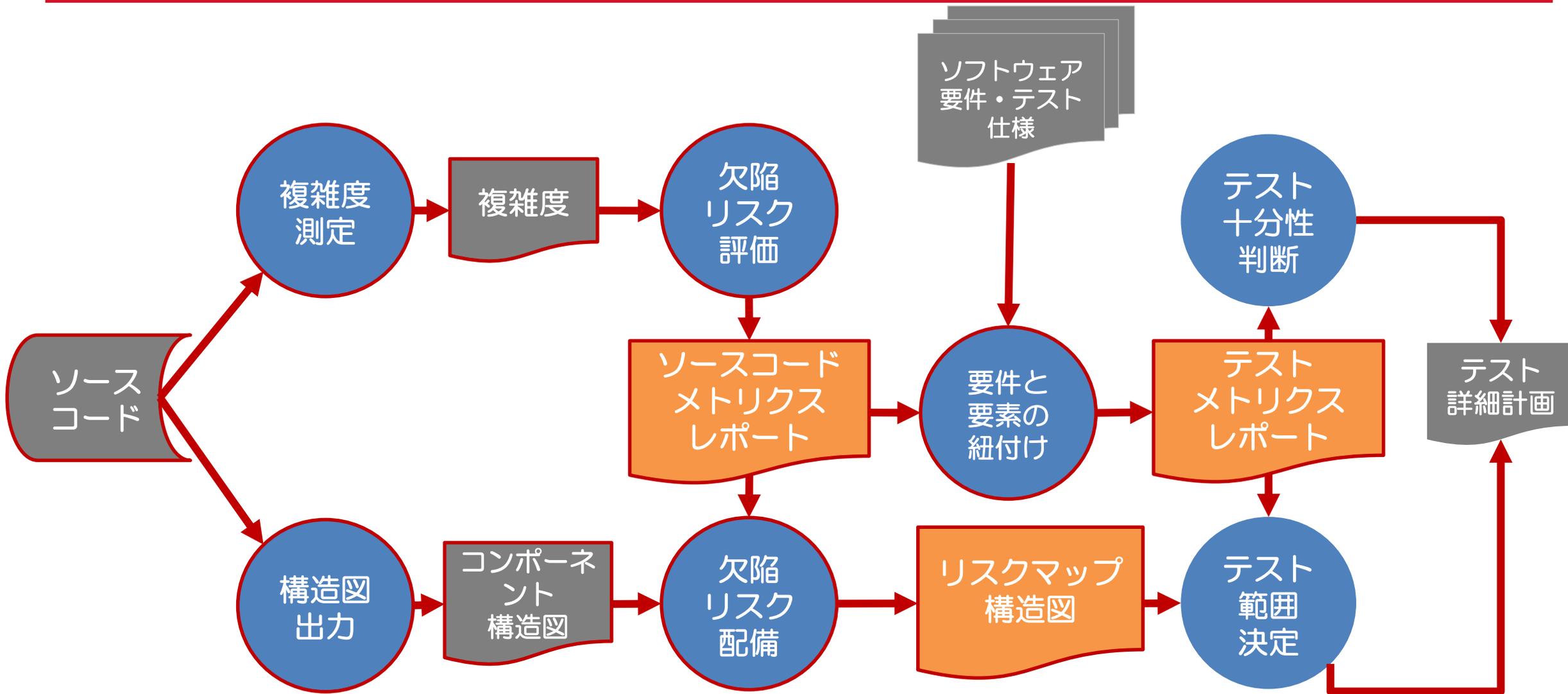




従来のプロセス

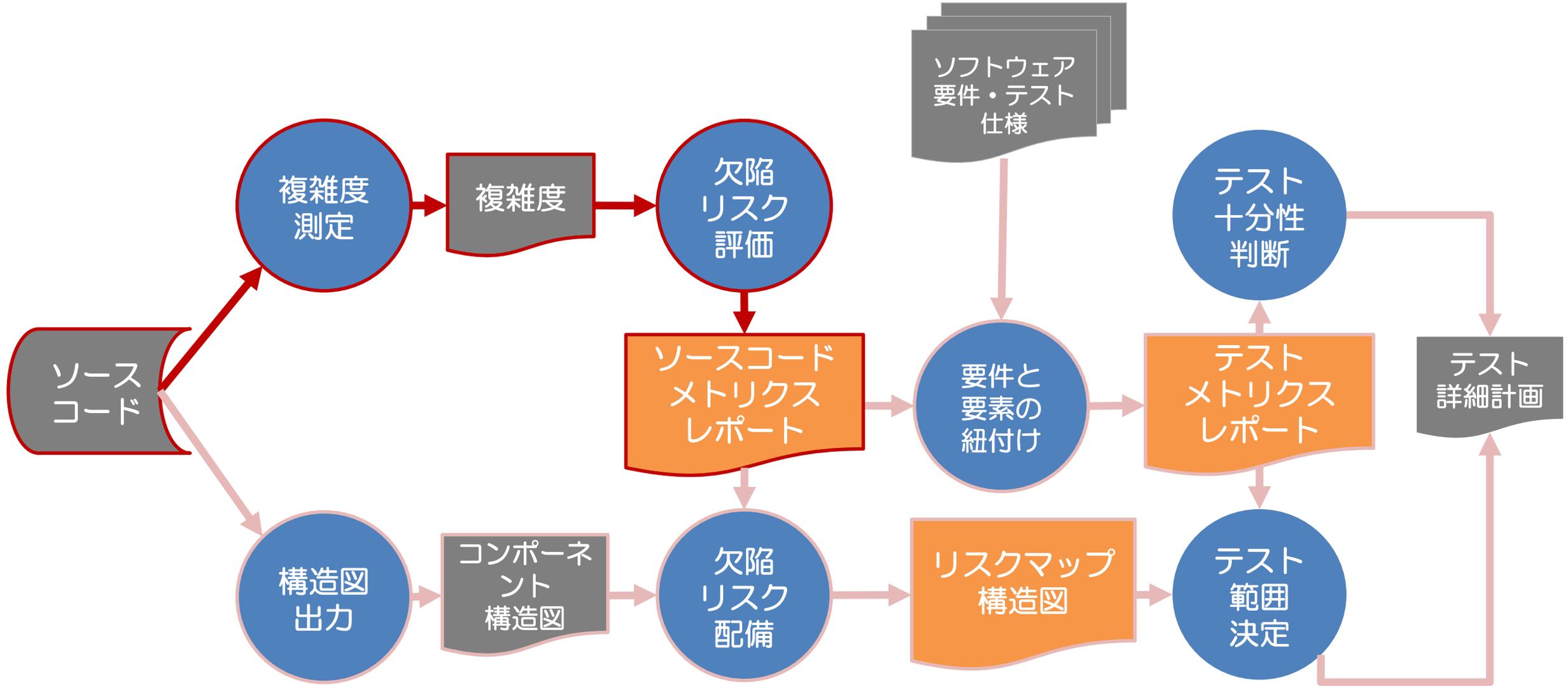


技法適用後のプロセス





技法適用後のプロセス



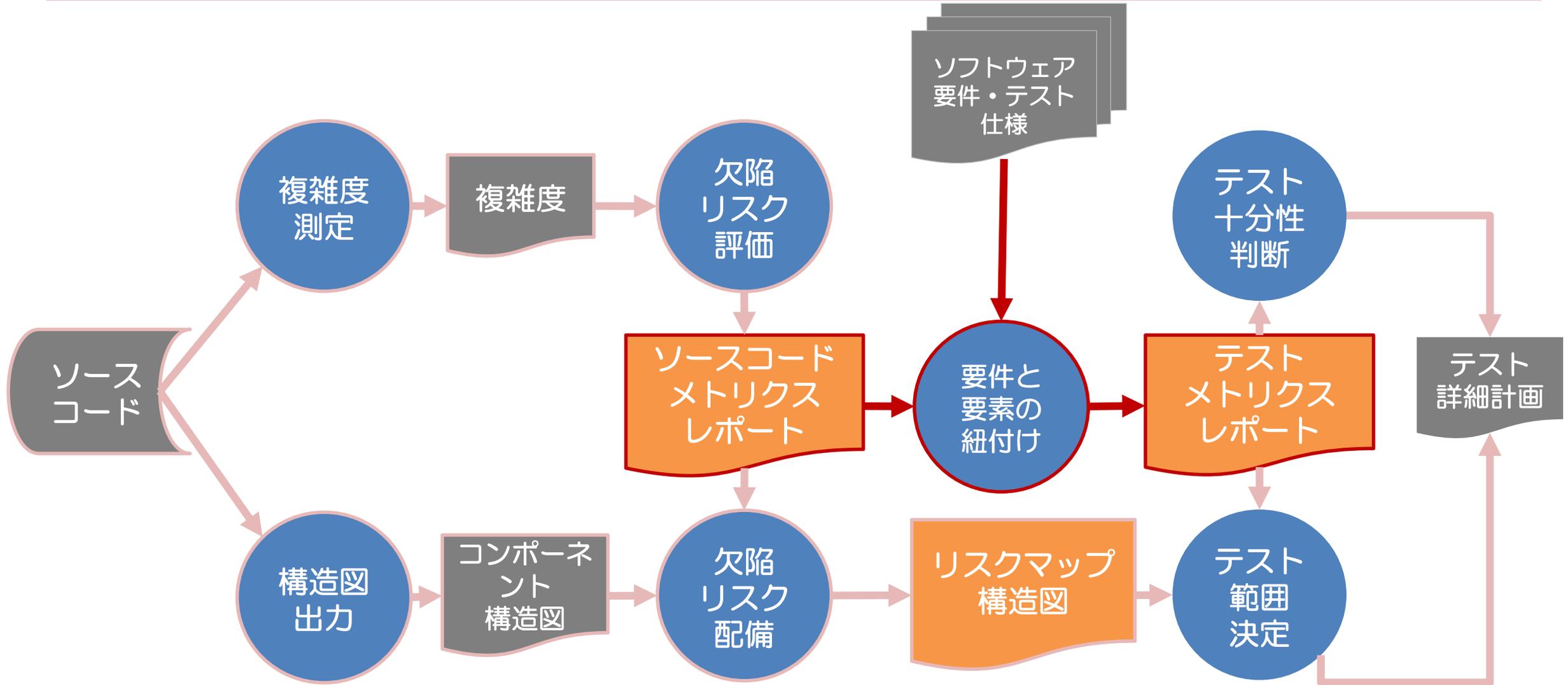
ソースコードメトリクスレポート

ソフトウェアリスク	システムレベル	欠陥リスク	12	11	11	7	5	5	4	4	4	4	4
		構造複雑度	中	中	中	無	無	無	中	中	中	中	中
		関数複雑度	中	低	低	高	中	中	無	無	無	無	無
ソフトウェアメトリクス	コード行数総和	CountLineCode	932	92	152	146	545	100	68	235	58	82	
	複雑度総和	SumCyclomatic	176	30	29	41	133	20	14	33	11	20	
	最大複雑度	MaxCyclomatic	25	13	13	316		6	5	5	2	10	
	最大ネスト深さ	MaxNesting	4	1	5	17			1	1	1	1	
	直接影響度閾値	DirectFanIn	2	2									
	直接依存度閾値	DirectFanOut	40	16	1								
	間接影響度閾値	IndirectFanIn	26	26									
	間接依存度閾値	IndirectFanOut	309	309	312	10	3	59	309	310	309	309	309
			_18_Drawer.EADrawer.EAGraphic	_03_ToolFactory.CreateFactory	_18_Drawer.GraphvizDrawer.Grap	_12_Parser.TokenAnalyzer	_12_Parser.VariableDefinitionDet	_12_Parser.VariableAccessDetect	AtScopeProject.AtScopeTool	AtScopeProject.StructureChart	_01_AtScope.AtScope	_01_AtScope.Boot	_01_AtScope.LicenseBoot

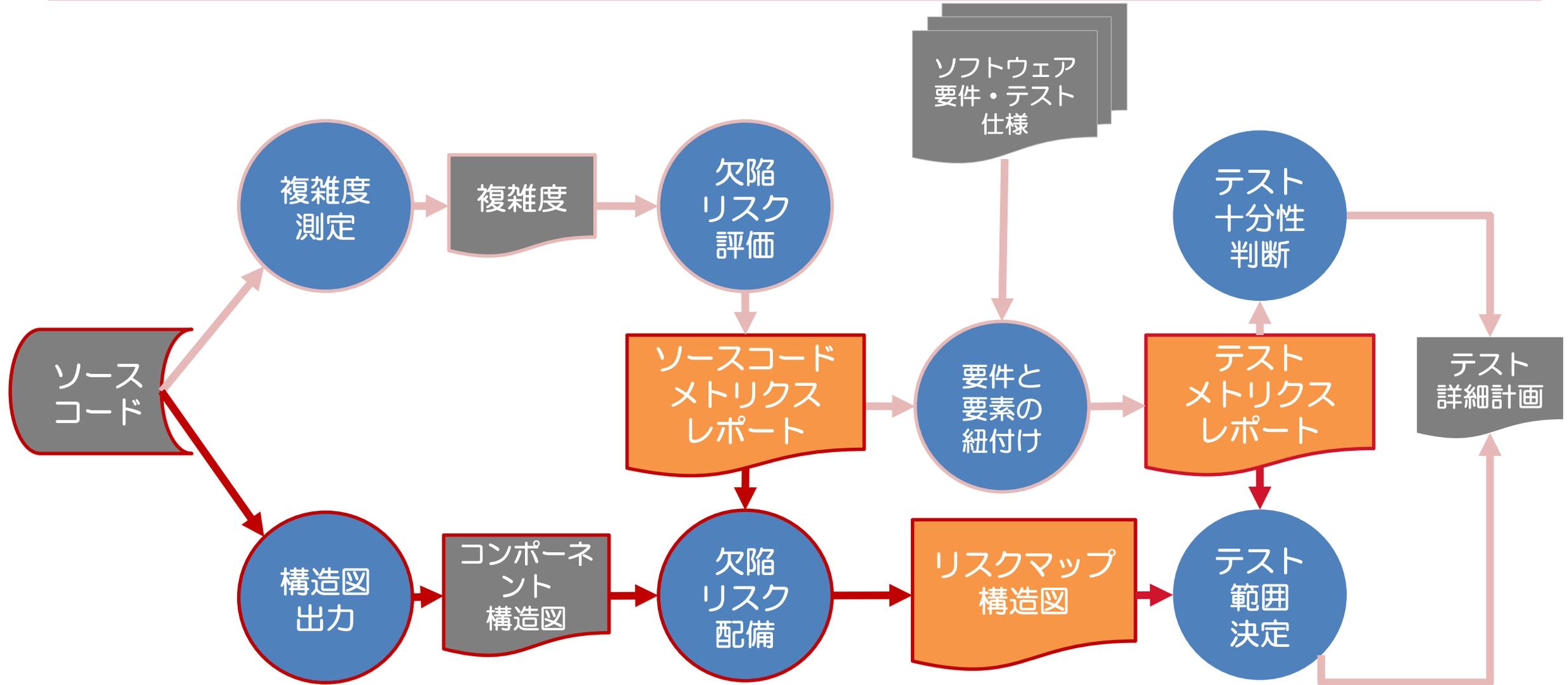
欠陥発生リスクが高い
コンポーネントを特定！



技法適用後のプロセス

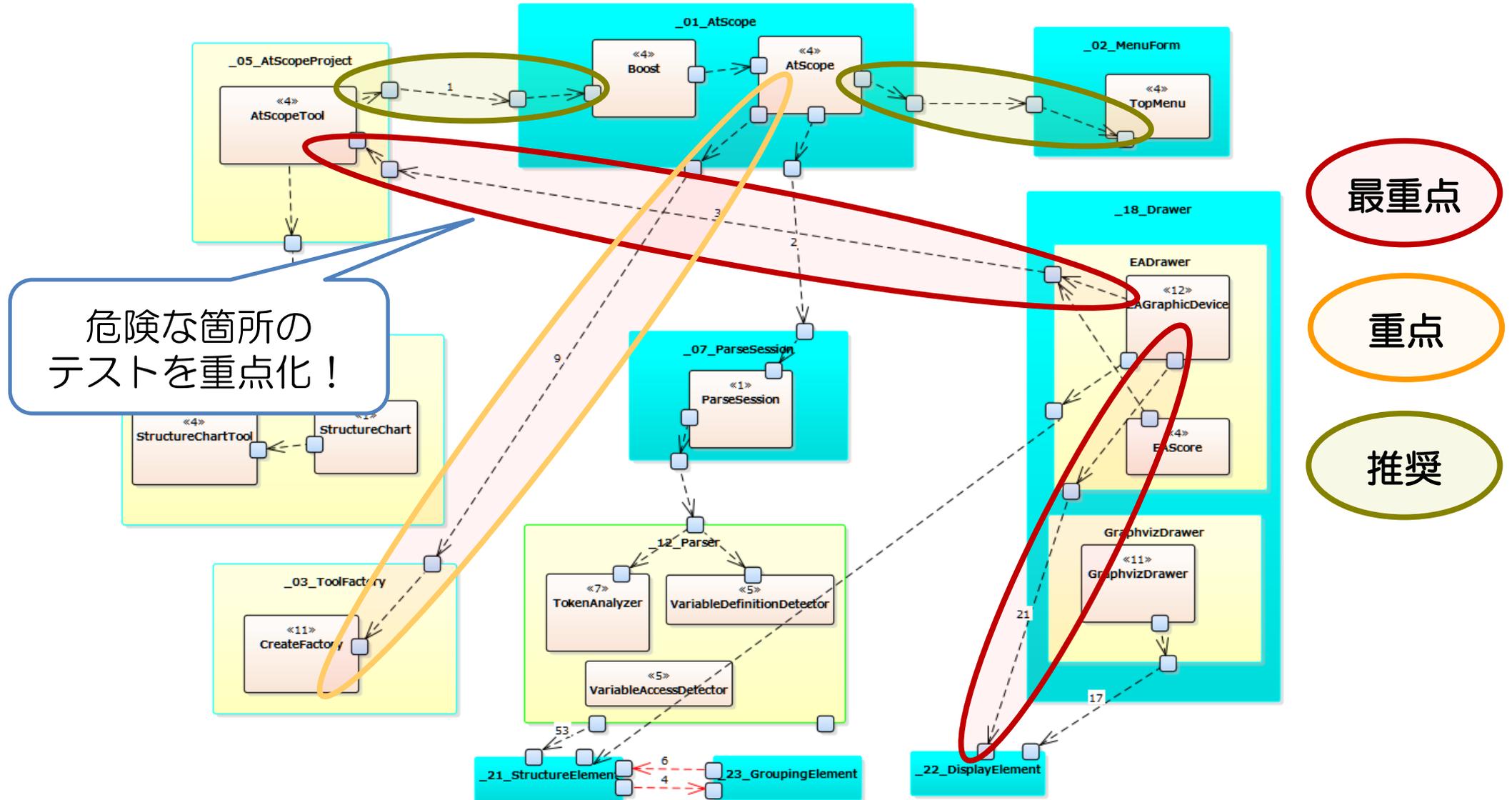


■ 技法適用後のプロセス





リスクマップ構造図



解決案

- ソースコードから**欠陥が出やすい箇所を予測**する
- 予測した欠陥発生箇所をもとに、テスト範囲を決定する



欠陥が出やすい**ポイントに絞った**テストがしたい

設計者や仕様書に**頼らずに入手できる情報**から
テスト範囲を絞りこみたい

結合テストで**期待通りに欠陥を出したい**

解決案

- ソースコードから**欠陥が出やすい箇所を予測**する
- 予測した欠陥発生箇所をもとに、テスト範囲を決定する



複雑度に着目することで、
欠陥発生箇所の絞込みが可能に！

ソースコードをテストベースにすることで、
確実に入手可能に！

現在プロジェクトで実践中



この活動の効果と課題【その①】

- 効果：無理なく納得しながらテスト活動ができる！

- Case 1 :

「全網羅テスト」「手探りのテスト」が不要に。

欠陥発生リスクを認識することで、戦略的なテストが行えるようになってきている。

- Case 2 :

テストチームの強みのひとつに。

理論的な武器を持つことで、上流からの品質向上に積極的に関与できるようになり、品質向上に貢献できているという実感が強く持てるようになってきている。



この活動の効果と課題【その②】

■ 副次的効果：開発チームの気づき&信頼関係の構築に貢献！

– Case 1 :

認識しづらい欠陥発生リスクの評価結果をマネジメント層へフィードバックすることで、改修によるリスクを考慮した開発方針が打ち出せるようになった。

– Case 2 :

担当モジュールの影響範囲を考慮せずに改修し続け、突如ソフトウェアが動かなくなることが過去にあった。開発者が理解していない情報を提供することで、開発者の意識改善&開発チームとの信頼関係を深め、対等な関係を築けるようになった。

■ チャレンジ中の課題：

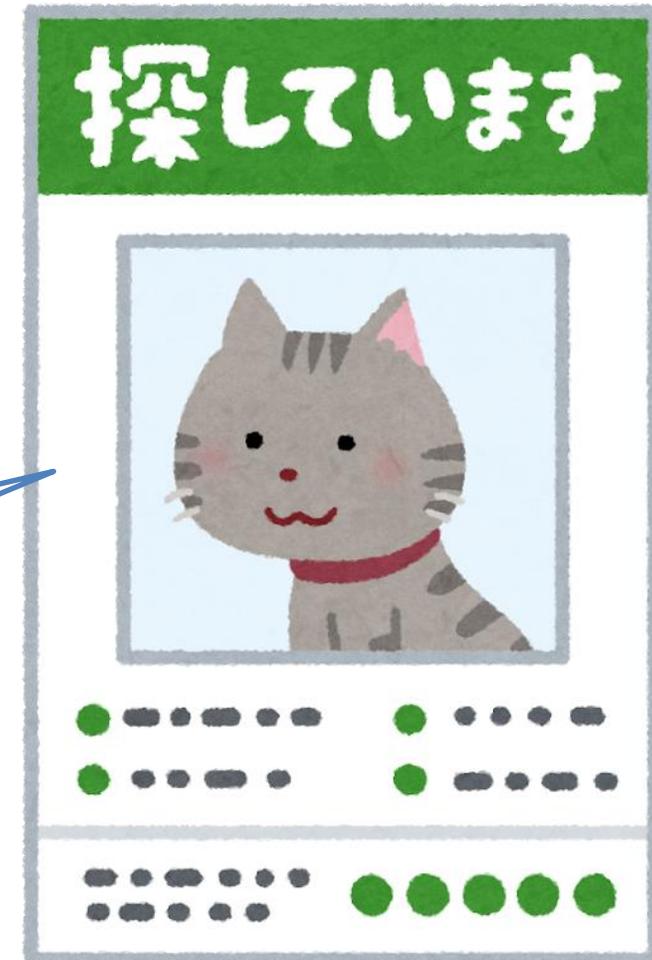
- テスト条件を決定し、テストケースを自動生成する



- 従来とは違う仕事のやり方を取り入れ、意識を変えてゆくことはとても難しいですが、この活動がテストチームの価値向上につながると考えています
- 「楽に楽しくテスト」できるよう、引き続きチャレンジしてゆきます！

募集中

- テスト条件の抽出等アイデアをお持ちの方
- この結合テスト技法をやってみたい方



ご清聴ありがとうございました

RICOH
imagine. change.

 **BACKSLASH**
DESIGN