年末報告会2019

M2 倉地 亮介

単位取得状況

- 一般科目群
  - 一般科目：3/4 (残りプロフェッショナルコミュニケーションⅡ)
- 先端科学技術科目群
  - 序論科目：3/3
  - 基盤科目：12/12
  - PBL科目：2/2
- 研究活動科目群
  - ゼミナールⅠ・Ⅱ：2/2
  - コロキアムA・B：2/2

研究紹介

研究タイトルについて

- タイトル候補
  - ソースコードの類似性に基づいたテストコード自動推薦ツール ★★★
  - 類似コード検出ツールを用いたテストコード自動推薦 ★★★
  - テストコード再利用支援のための自動推薦ツール ★★
  - 再利用を目的とした，テストコード自動推薦ツールの提案 ★
  - テストコードの再利用支援を目的としたテストコード自動推薦ツール ★

研究内容

1. テストスイート自動推薦手法

- 類似コード検出技術を用いてOSSプロジェクト内のテストコードを特定するアルゴリズム

2. SuiteRec

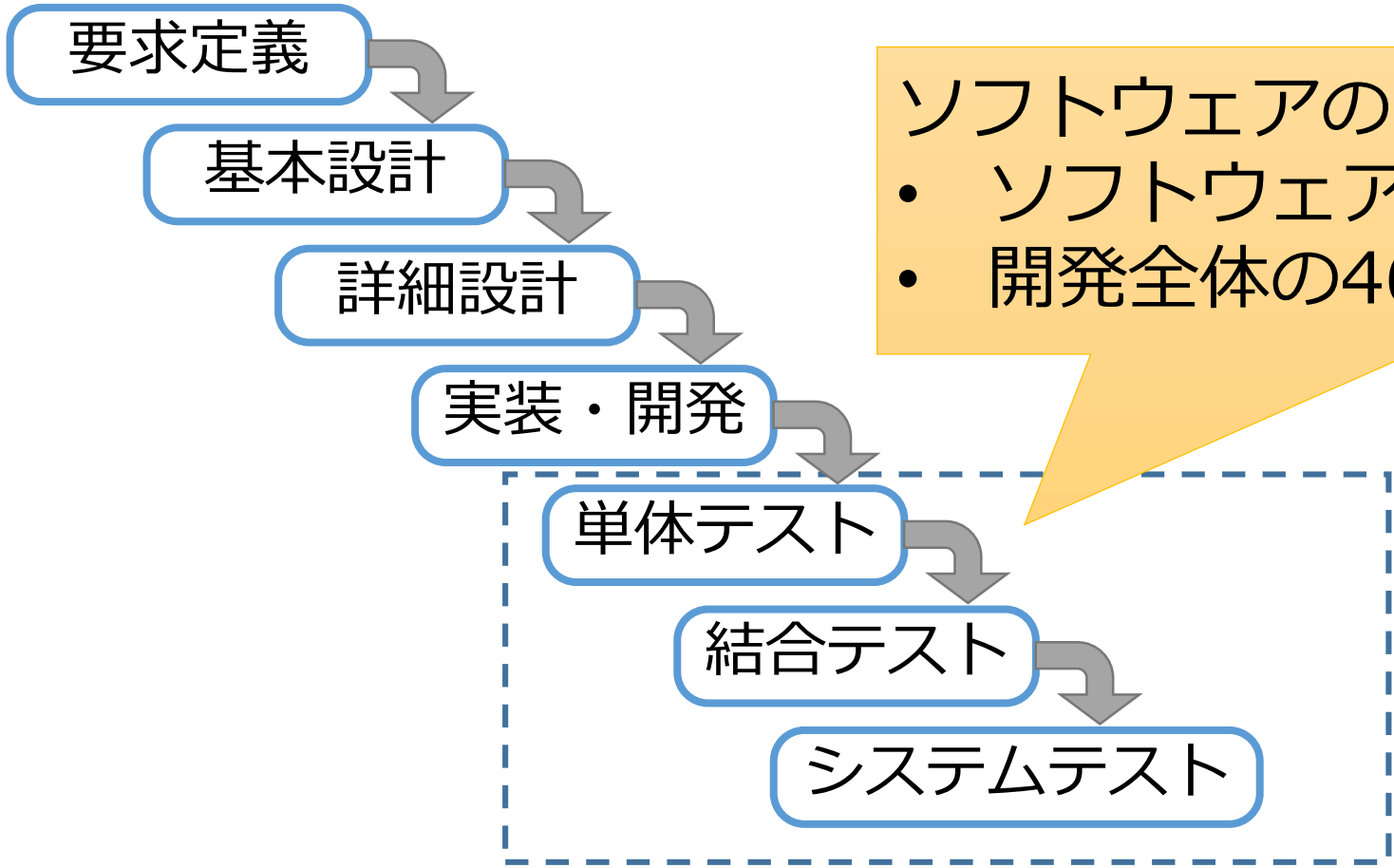
- 開発者が入力したコードに関連するテストスイートを表示する
- webアプリケーションとして実装された Interface

3. 評価実験

- SuiteRecが開発者のテストコード作成をどれだけ支援できるか定量的・定性的に評価

背景

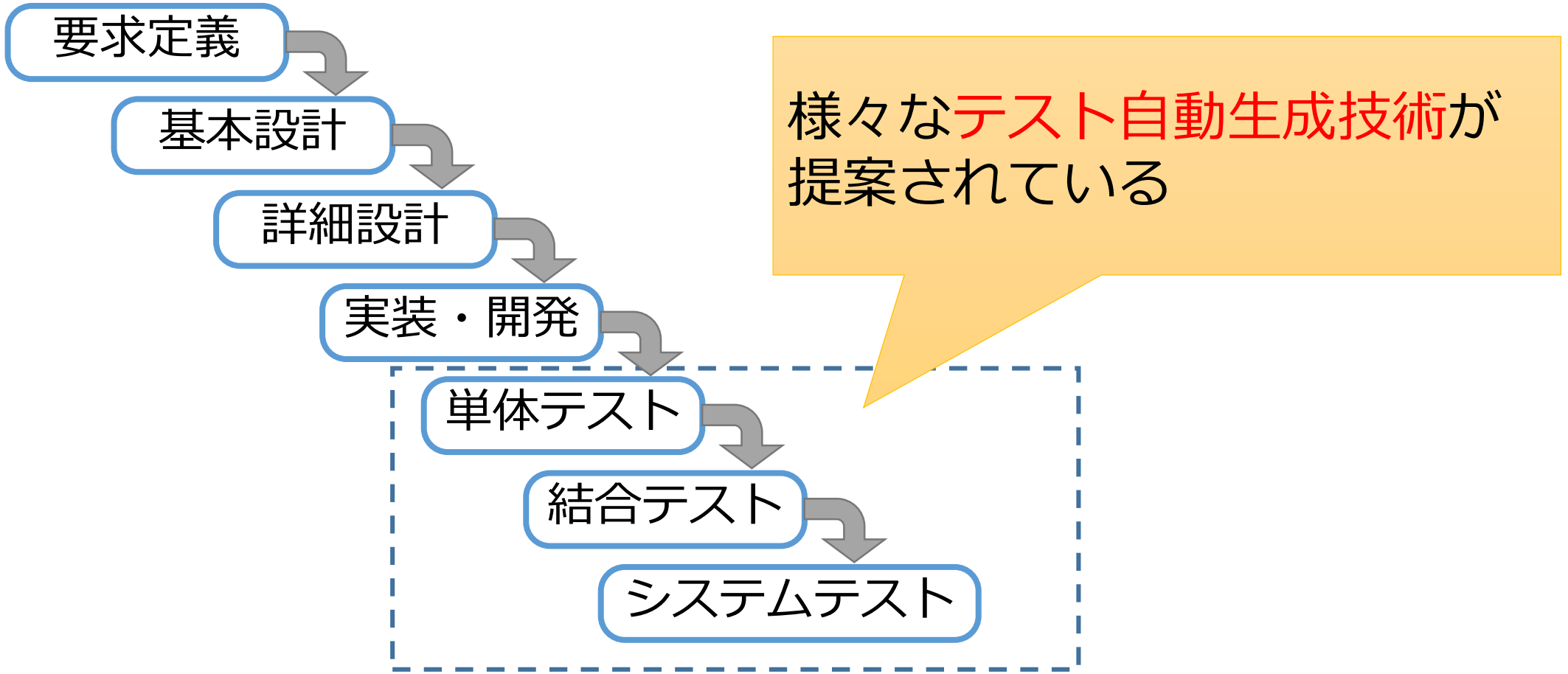
- ソフトウェアに求められる要件が高度化・多様化する一方で，
  ユーザからは品質確保やコスト削減に対する要求も増している



ソフトウェアのテスト工程
- ソフトウェアの品質を確かめる工程
- 開発全体の46%の費用を占める

背景

- ソフトウェアに求められる要件が高度化・多様化する一方で，
  ユーザからは品質確保やコスト削減に対する要求も増している



自動生成ツールにおける課題

- 自動生成されたテストコードは，保守作業を困難にする[1]
  - 対象コードの作成経緯や意図に基づいて生成されていないので開発者は理解しにくい
  - 開発者は自動生成されたコードを信用していない

テスト失敗の原因がテストコードの問題なのか，テスト対象のコードによるものなのか判断が難しい

[1] S.Shamshiri,J.Rojas,J.Pablo Galeotti,N.Walkinshaw,G.Fraser . How Do Automatically Generated Unit Tests Influence Software Maintenance?. Proc. of ICST, pp.239–249, 2018.

テストスメル

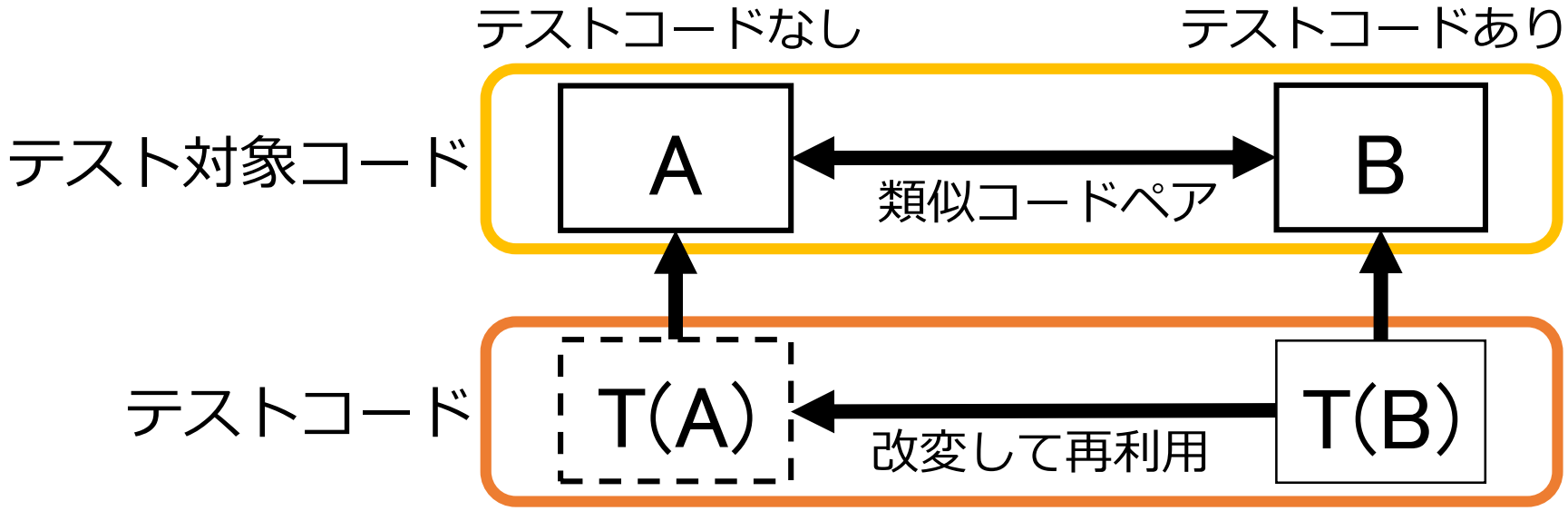
- テストコードのよくない実装を表す指標
  - Van Deursenら[4]は11種類のテストスメルのカタログを提案した (現在では21種類に拡張されている)
  - テストスメルの例: Assertion Roulette

```
@MediumTest
public void testCloneNonBareRepoFromLocalTestServer() throws Exception {
    Clone cloneOp = new Clone(false,
         integrationGitServerURIFor("small-repo.early.

    Repository repo = executeAndWaitFor(cloneOp);

    assertThat(repo, hasGitObject("ba1f63e4430bff267d                    8afc1d6294db0ccc"));

    File readmeFile = new File(repo.getWorkTree(), "README");
    assertThat(readmeFile, exists());
    assertThat(readmeFile, ofLength(12));
}
```

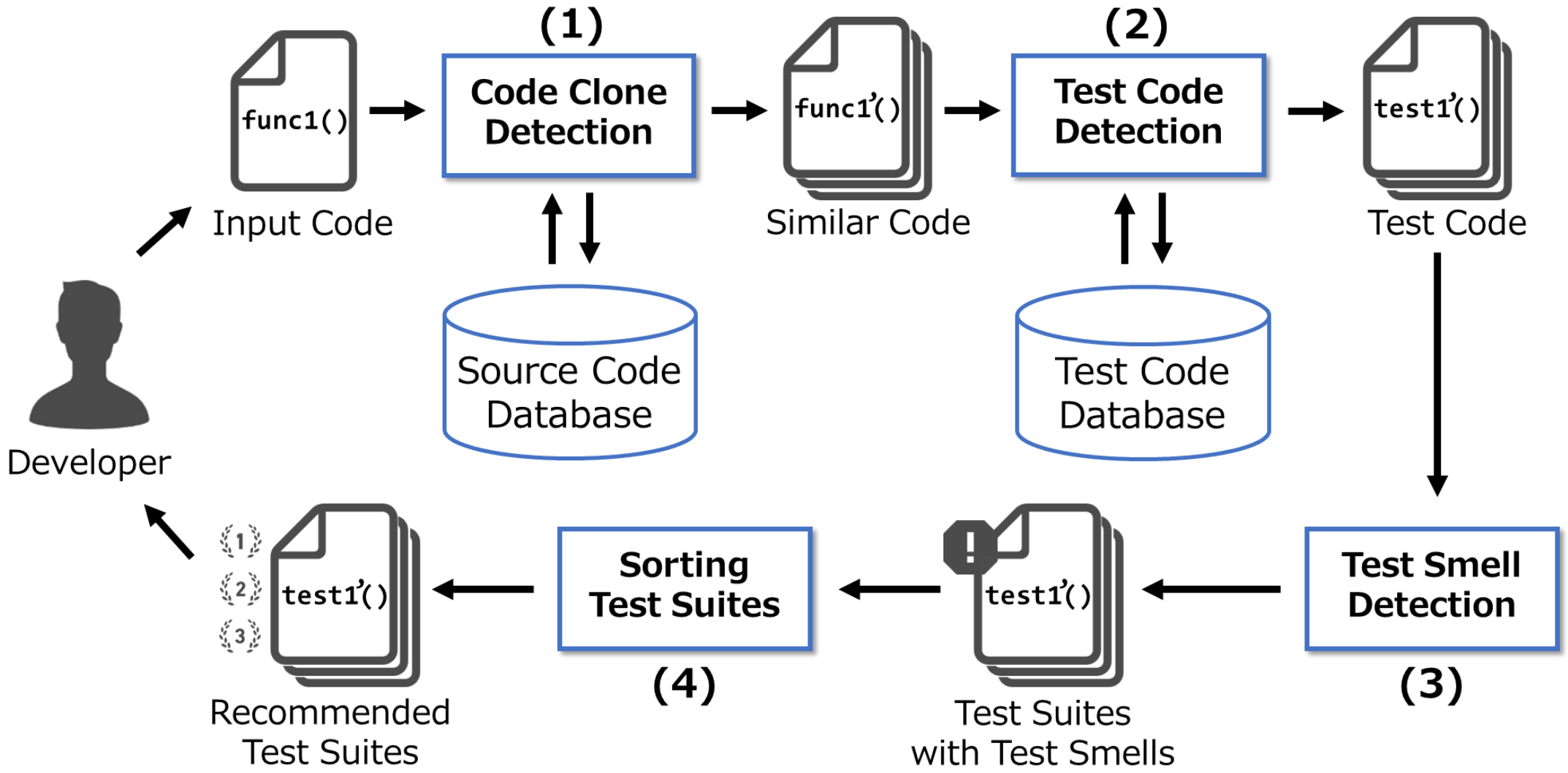
複数のassert文が存在する

研究目的とアイディア

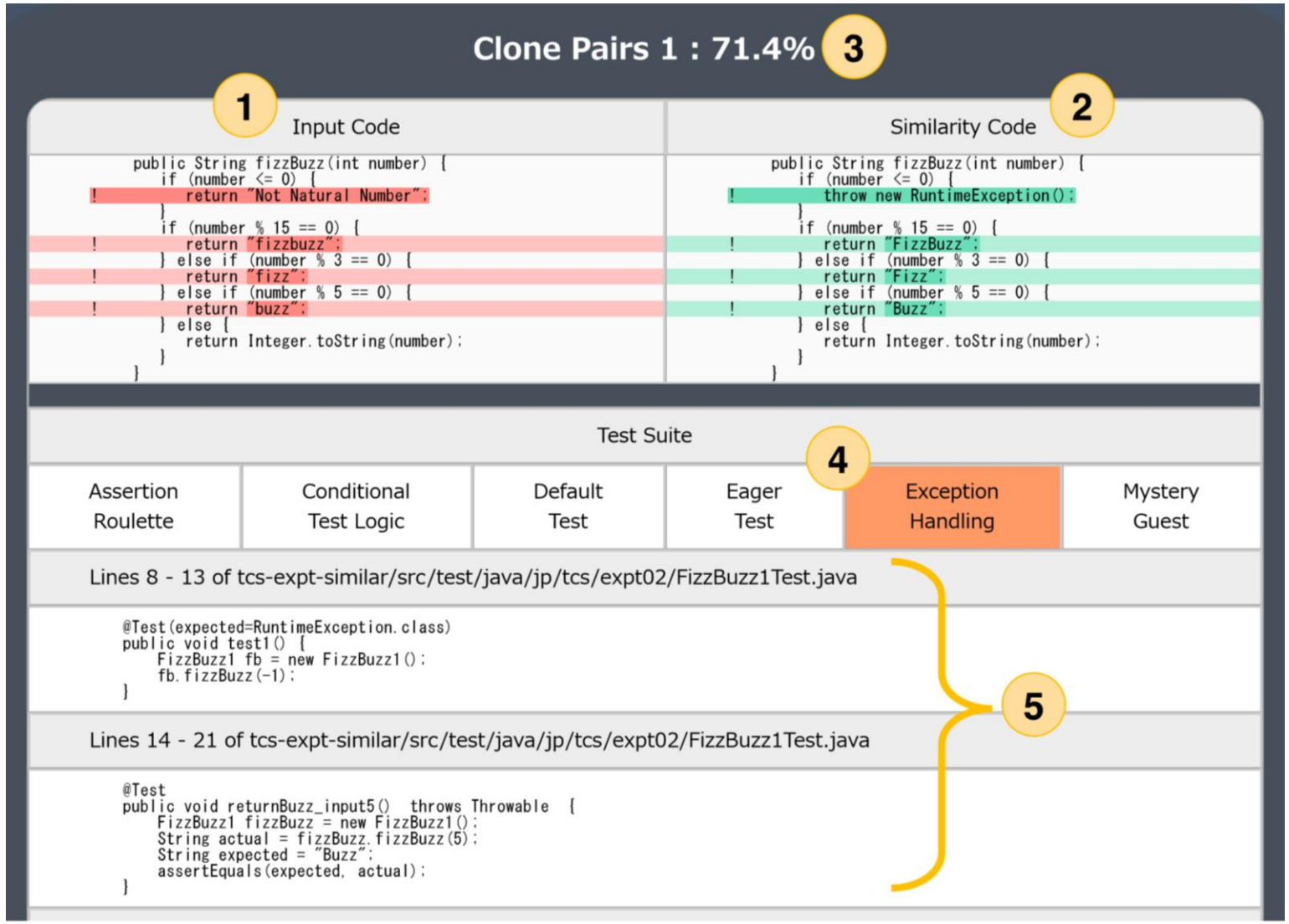
- 目的: 既存の高品質のテストを推薦することで開発者を支援
  - 命名規則に従った可読性の高いテストコードを利用できる
  - 人によって作成された信頼性の高いテストコードを利用できる
- アイディア: 類似コード間でテストコードを再利用



提案ツールの概要


(1)
func1()
Code Clone Detection
func1'()
(2)
Test Code Detection
test1'()
Input Code
Similar Code
Test Code
Source Code Database
Test Code Database
Developer
test1'()
Sorting Test Suites
test1'()
Test Smell Detection
(4)
(3)
Recommended Test Suites
Test Suites with Test Smells

推薦テストスイート



① 入力コード

② 類似コード

③ 類似度(UPI)

④ テストスメル

⑤ テストスイート

Research Question

- RQ1: 提案ツールは，高いカバレッジを持つのテストコードの作成を支援できるか？
- RQ2: 提案ツールは，テストコード作成時間を削減できるか？
- RQ3: 提案ツールは，品質の高いテストコードの作成を支援できるか？
- RQ4: 提案ツールの利用は，開発者のテストコード作成タスクの認識にどう影響するか？

実験結果(RQ1,RQ2)

- **RQ1:** 提案ツールは，高いカバレッジを持つのテストコードの作成を支援できるか？

A. 条件分岐が多いプログラムのテストコードを作成する際，提案ツールの利用はカバレッジ(C1)を向上するのに役立つ可能性がある

- **RQ2:** 提案ツールは，テストコード作成時間を削減できるか？

A. 提案ツールの利用は，推薦されたコードを理解し編集が必要なので開発者は，テストコード作成に多くの時間を費やす可能性がある

実験結果(RQ3,RQ4)

- **RQ3:** 提案ツールは，品質の高いテストコードの作成を支援できるか？

A. 開発者は，提案ツールによって推薦される高品質のテストスイートを参考にすることで品質の高いテストコードを作成できる

- **RQ4:** 提案ツールの利用は，開発者のテストコード作成タスクの認識にどう影響するか？

A. 提案ツールの利用した場合，開発者はテスト作成タスクを容易だと認識し，作成したテストコードに自信が持てる

まとめ

- 類似コード検出技術を用いて，既存の高品質のテストコードを推薦するツールを提案
- 提案ツールを定量的・定性的に評価
  1. 条件分岐が多いコードのテストを作成する際にコードカバレッジの向上に効果的であること
  2. 提案ツールを使用して作成したテストコードは検出されたテストスメルの数が少なく品質が高いこと
  3. 提案ツールの利用した場合，開発者はテスト作成タスクを容易だと認識し，作成したテストコードに自信が持てる

今後の計画

- 提案ツールの推薦ランキングに関する評価を行う(年内に)
- 修論書き始める(年明けから)
- プレゼン資料作成(1月中旬)
- 被験者の数を増やすかも(阪大・KITとか)
- IWSC通れば2月に発表
- SuiteRecの改善・拡張
  - クリップボード(コピー＆ペーストがしやすいように)
  - 自動編集機能
  - 複数の類似コード検出ツールに対応して，検出できる類似コードの幅を広げる

評価実験に関する補足資料

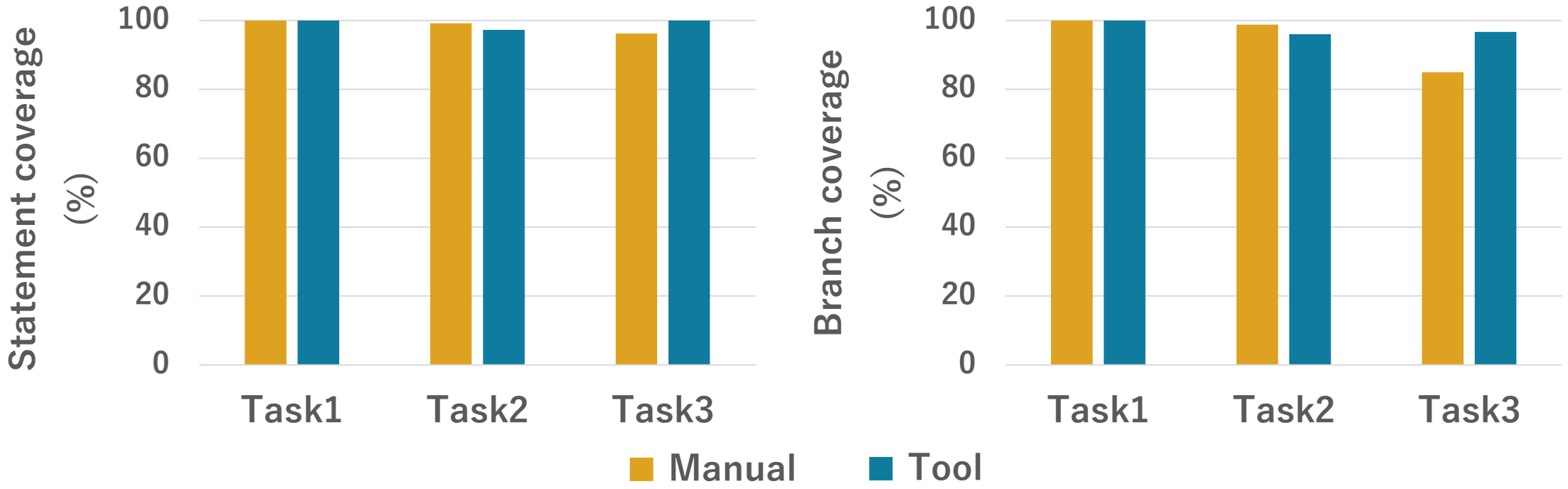
評価実験

- 提案手法を被験者のテスト作成をどの程度支援できるかを定量的・定性的に評価
- 実験概要
  - 情報科学を専攻する修士の学生10人に3つの問題のテストコードを作成してもらう

	Task1	Task2	Task3
概要	典型的なFizzBuzzの関数	第1引数に応じて計算方法を変更し，計算結果を返す	2つの入力値に基づいて試験の合否を判定する
分岐数	8	16	24

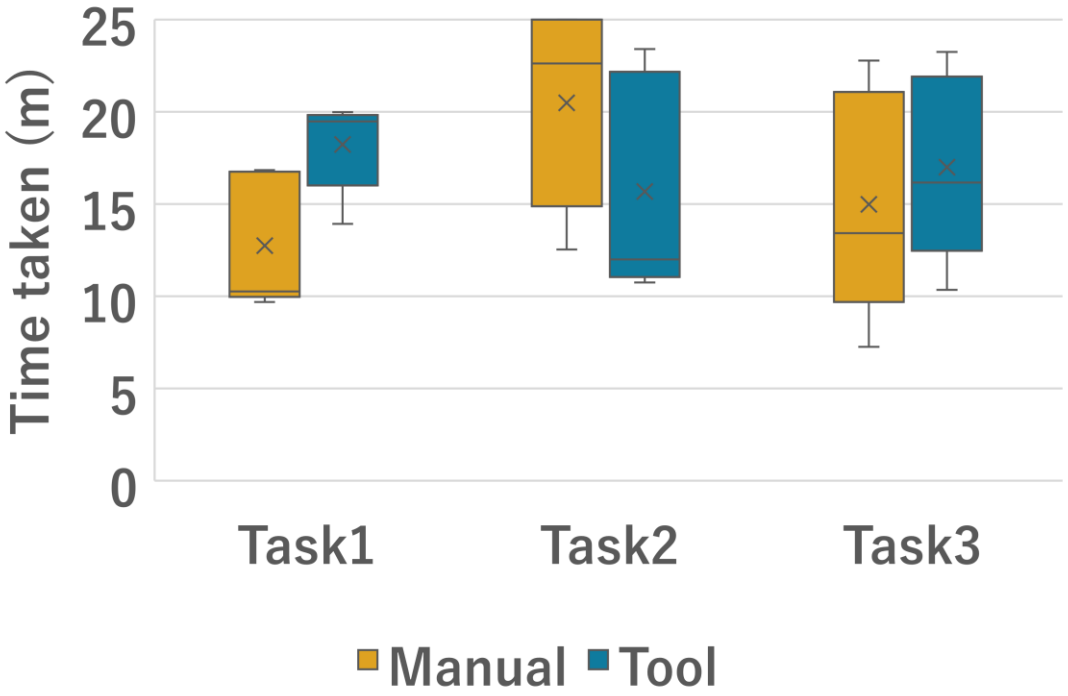
  - ツールを使用した場合とそうでない場合で被験者が作成したテストコード比較する
  - 実験後にテスト作成タスクに関するアンケートに回答してもらった

RQ1:提案ツールは，高いカバレッジを持つのテストコードの作成を支援できるか？



条件分岐が多いプログラムのテストコードを作成する際，提案ツールの利用はカバレッジ(C1)を向上するのに役立つ可能性がある

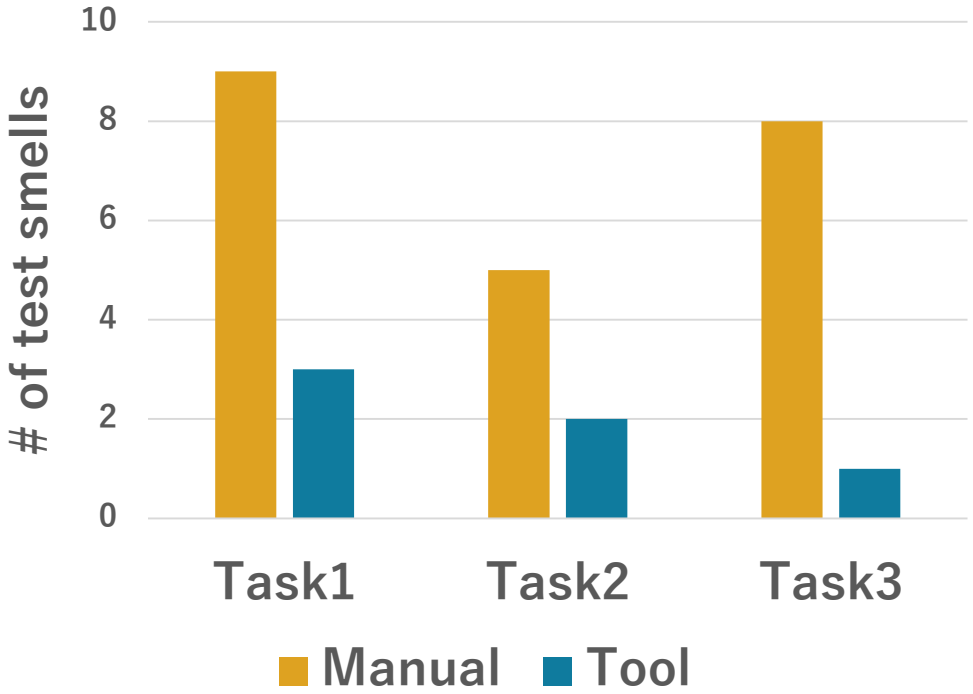
RQ2: 提案ツールは，テストコード作成を削減できるか？



- Task1とTask3は，ツールを使用した場合，タスク終了までの時間が長い
- 一方で，Task2はツールを使用しない場合の方がタスク終了までの時間が長い
- 多くの開発者は無駄なテストを作成するのに多くの時間を費やした可能性がある

提案ツールの利用は，推薦されたコードを理解し編集が必要なので
開発者は，テストコード作成に多くの時間を費やす可能性がある

RQ3: 提案ツールは，品質の高いテストコードの作成を支援できるか？



- すべてのTaskにおいて，ツールを使用し場合検出されたテストスメルの数が少ない
- 多く検出されたテストスメル
  - Assertion Roulette
  - Default Test
  - Eager Test

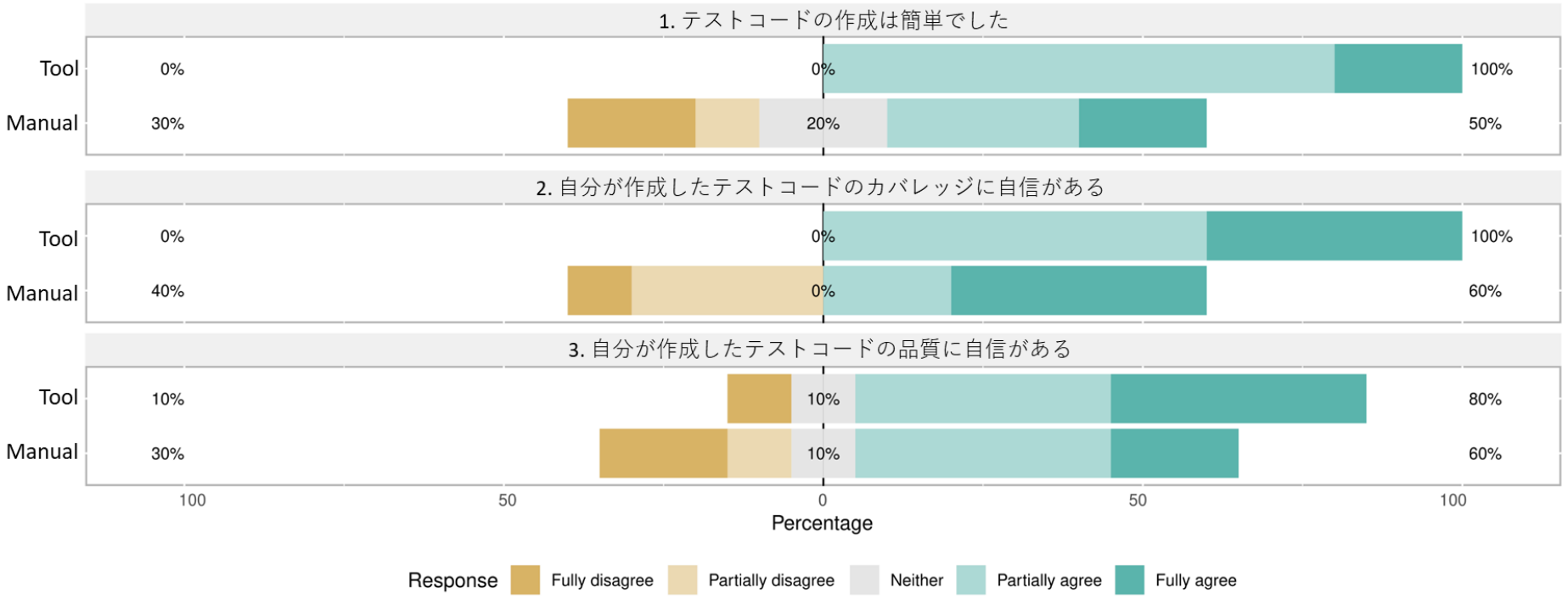
開発者は，提案ツールによって推薦される高品質のテストスイートを参考にすることで品質の高いテストコードを作成できる

RQ4: 提案ツールの利用は，開発者のテストコード作成タスクの認識にどう影響するか？

- 被験者に実験タスク終了後にアンケートを実施した

	項目	5段階評価 強く反対・反対・どちらでもない・賛成・強く賛成				
1	テストコードの記述は簡単でした(ツール不使用)	1	2	3	4	5
2	テストコードの記述は簡単でした(ツール使用)	1	2	3	4	5
3	作成したコードの品質に自信がある(ツール不使用)	1	2	3	4	5
4	作成したコードの品質に自信がある(ツール使用)	1	2	3	4	5
5	作成したコードの品質に自信がある(ツール不使用)	1	2	3	4	5
6	作成したコードの品質に自信がある(ツール使用)	1	2	3	4	5

RQ4: 提案ツールの利用は，開発者のテストコード作成タスクの認識にどう影響するか？



提案ツールの利用した場合，開発者はテスト作成タスクを容易だと認識し，作成したテストコードに自信が持てる

関連研究

- 類似コード間のテスト再利用
  - Zhang[1]らは，クローンペア間でコードを移植を行い，移植前と移植後のテスト結果を比較しその情報を基にテストを再利用するツールGrafterを提案した.
  - Sohaら[2]は，開発者が詳細な再利用計画を決めることで，コード片を再利用する際に，テストスイートの関連部分を半自動で再利用および変換を行うツールSkipperを提案した.

SuiteRecは，既存ツールと2つの視点で異なる
- OSS上からテストコードを検出することができる
- クローンペア間のテスト再利用計画は開発者に委ねていること