SKYCOM Guaranteed

ソースコードに存在する不適切なコメントの検出手法適用事例

2017年6月8日

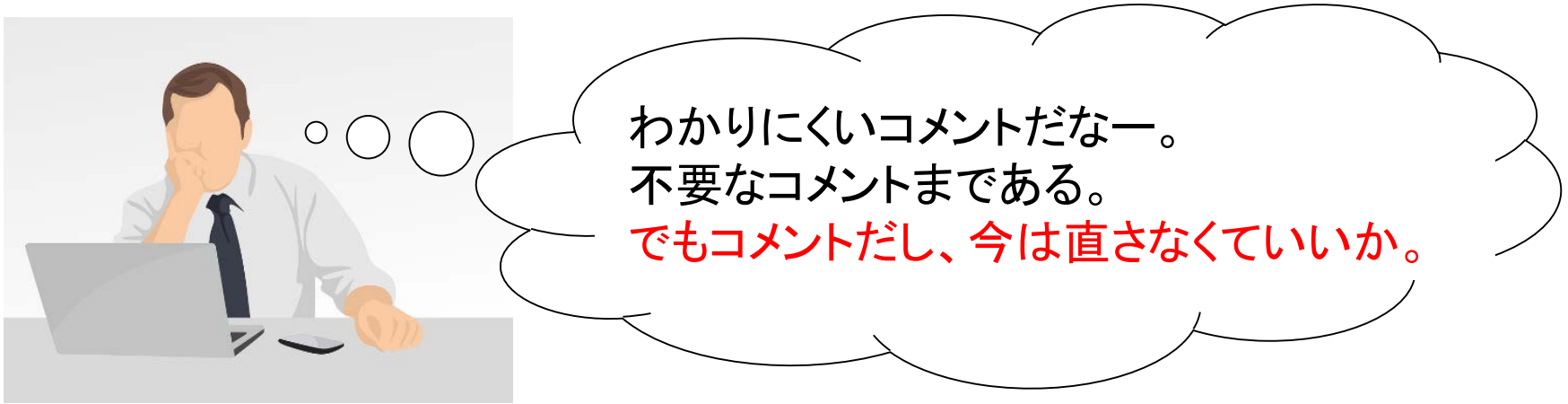
株式会社スカイコム
開発1部
甲斐 秀一

宮崎大学
工学研究科
田上 諭，片山 徹郎

背景と課題

■ 背景

ソースコードに用いられるコメントは、プログラムの処理には直接影響を与えないため、内容について重視されにくい傾向にある

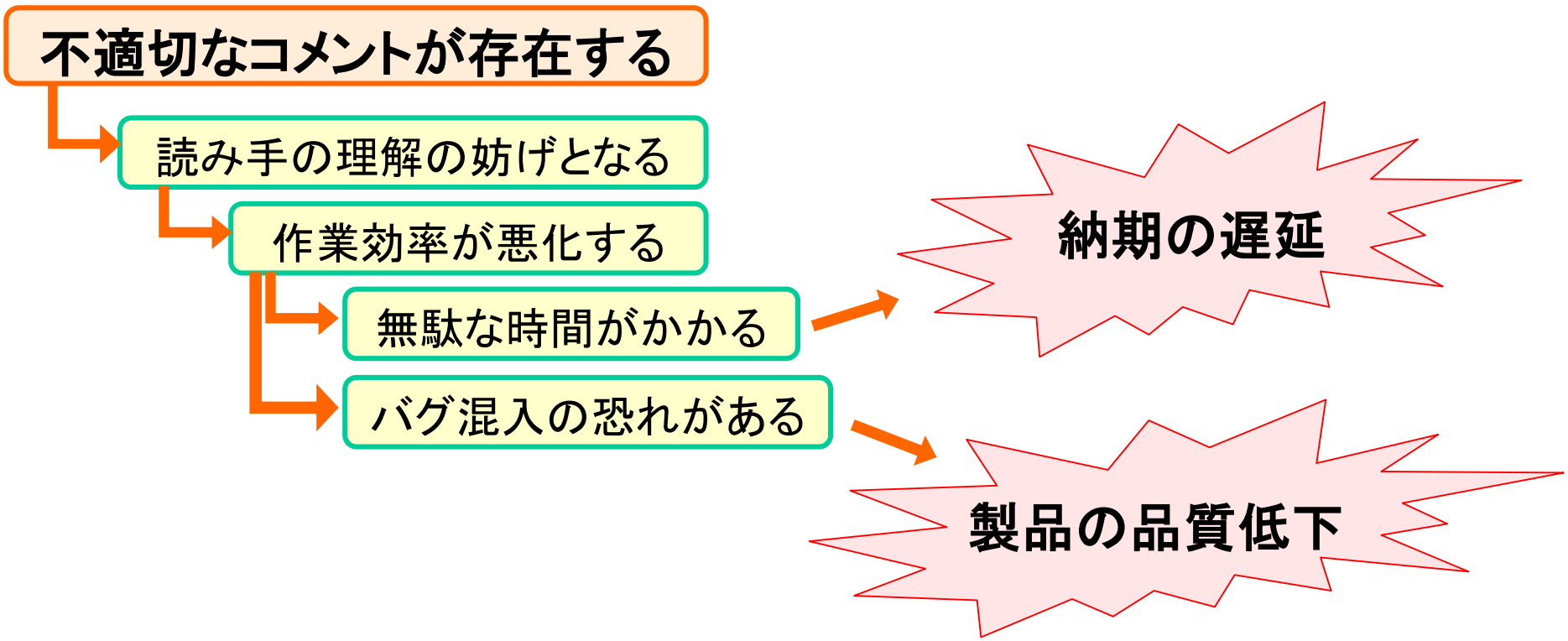


当社でもこのような状況が見受けられ、不適切なコメントを含んだソースコードが徐々に蓄積されていた

背景と課題

■ 課題

不適切なコメントは、以下のような問題を起こす可能性がある



取組みの目的

SKYCOM Guaranteed

■ 目的

現状、不適切なコメントを検出する手法は一般的に浸透していない

宮崎大学と協力して、不適切なコメントを効率的に検出する仕組みを構築したい

[期待する効果]

- 不適切なコメントを目視で確認する手間と不正確さの解消
- 検出した不適切なコメントの周辺処理を見直すことによる品質向上
- コメントの重要性を理解するための教育ツールの役割

対策

■ 本研究で取組む対策

ソースコード内の不適切なコメントを検出するツールの作成・利用

■ 本研究での宮崎大学と当社の役割

【宮崎大学】　【当社】

ツール開発　　社内アンケート実施、ツール利用・評価

不適切なコメントの定義

SKYCOM Guaranteed

■ 不適切なコメントの定義決め

1.不適切な可能性のあるコメントの候補を、宮崎大学共同研究チームで整理
2. 整理した候補について、当社開発部門内で賛否のアンケート調査を実施

■ アンケート調査結果

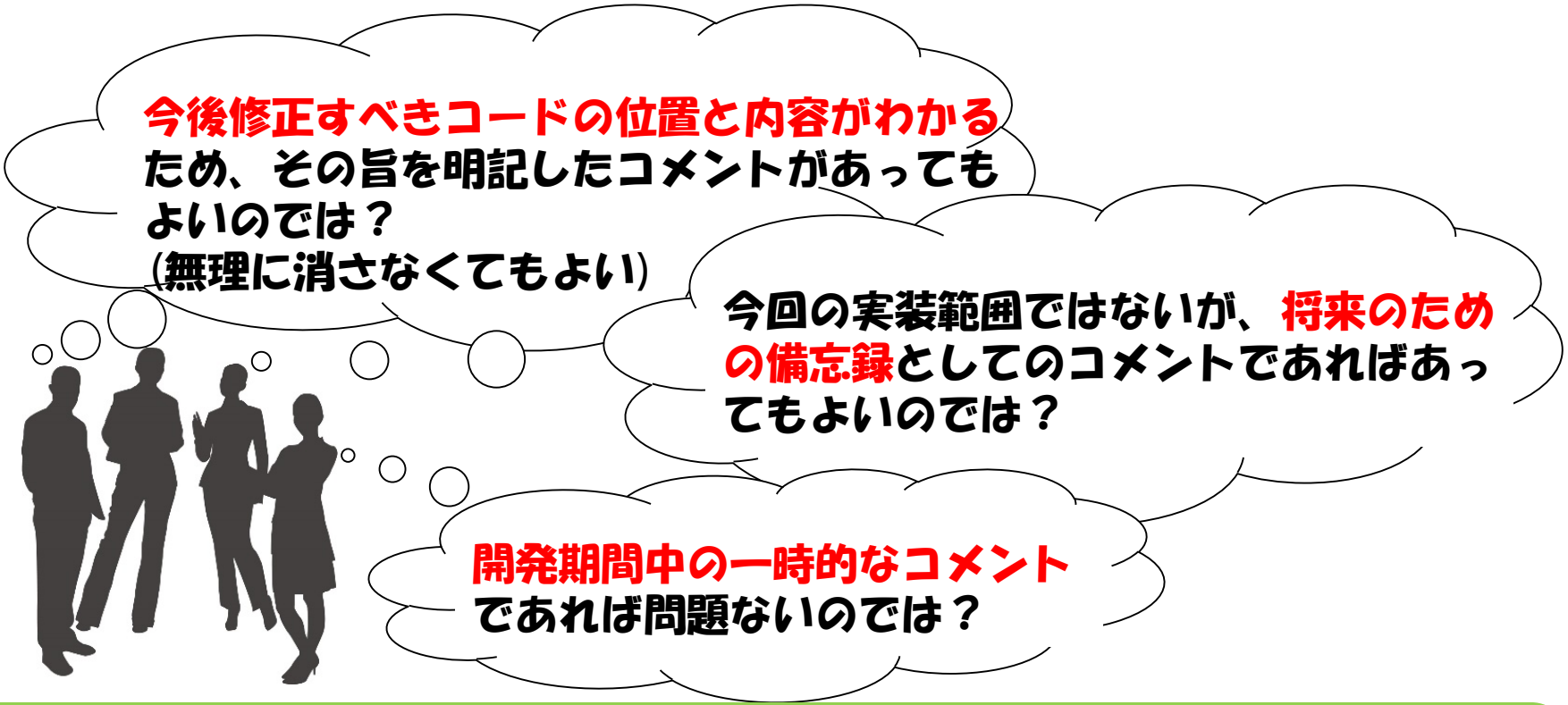
不適切な可能性のあるコメントの分類	例	不適切と考える理由	開発部門賛同率
改修履歴コメント	// 2017/03/03 yamada 修正	改修が重なるごとに見づらくなる	93.8%
TODOコメント	// その他は今後修正予定	いずれ忘れ去られたり、いつまでも後回しにされる	62.5%
意図が曖昧なコメント	// 暫定的な処理	読み手によって解釈が異なる可能性のあるコメントは、混乱を招く	87.5%

処理そのもののコメント	// ここで10回ループする
	while (i < 10) {
	  i++; // iをインクリメントする
	}
		コードを見たままのコメントでは意味がなく、その処理の理由や目的を記載すべき	93.8%

※TODOコメントについては、賛同率が他と比較して低い結果となった！
理由は次のスライドで説明する

不適切なコメントの定義

「TODOコメント」について、**不適切ではない**と判断した社員からは、
以下のような意見が挙がった。



「備忘録」としてTODOコメントを必要と考える社員もいた。
TODOコメントは不適切かどうかより、「**備忘録として検出する必要のあるコメント**」の意味合いが強いものとして、検出対象に含めた。

不適切なコメントの検出方法

SKYCOM Guaranteed

■ 検出手順

以下の2ステップで検出を行う。

STEP1：不適切なコメントとして検出する単語を定義

STEP2：不適切なコメント検出（検出ツールの実行）

■ STEP1：不適切なコメントとして検出する単語を定義

ソースコード分析結果、アンケート結果より、
検出対象の単語をデフォルトで以下のように定義した

不適切な可能性のあるコメントの分類	定義する単語・日付フォーマット
改修履歴コメント	yyyy/mm/dd、yyyy.mm.dd、yyyy-mm-dd、yyyy年mm月dd日
TODOコメント	TODO、将来、未対応、今後、予定
意図が曖昧なコメント	暫定、曖昧、未定、仮
処理そのもののコメント	ループ、繰り返し、ブレーク、インクリメント
理解不足コメント	なぜ、わからない、？

不適切なコメントの検出方法

SKYCOM Guaranteed

■ STEP1:不適切なコメントとして検出する単語を定義 (定義ファイルの設定)

[定義ファイルの特徴]

- XML形式
- 検出する単語を追記して使用
- 正規表現が使用可能

拡張が容易

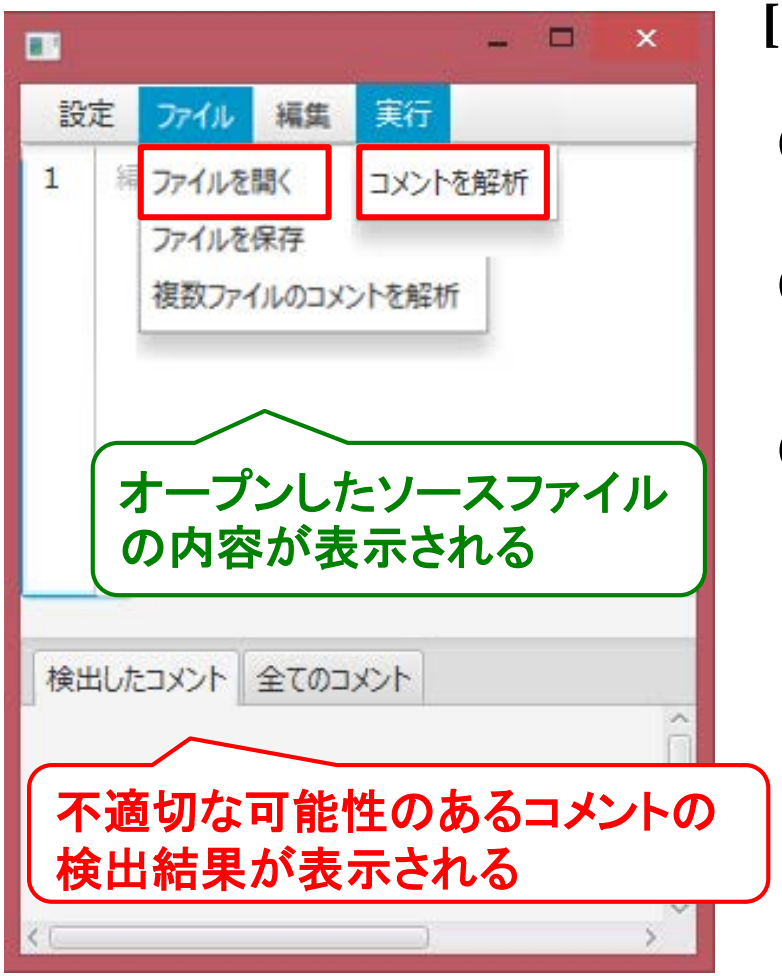
```
<?xml version="1.0" encoding="UTF-8" ?>
<comments>
   <comment type="TODOコメント">
      <regularExpression>TODO</regularExpression>
      <regularExpression>将来</regularExpression>
   </comment>

   <comment type="改修履歴コメント">
      <!-- 検出する日付フォーマットは以下
         yyyy/mm/dd、yyyy.mm.dd、yyyy-mm-dd、yyyy年mm月dd日 (yyyyは2桁以上4桁以下)-->
      <regularExpression>(?=.*¥d{2,4}[.¥-/年]¥d{1,2}[.¥-/月]¥d{1,2}日{0,1})</regularExpression>
   </comment>
</comments>
```

不適切なコメントの検出方法

SKYCOM Guaranteed

■ STEP2：不適切なコメント検出

STEP1の定義ファイルを読み込んでコメント解析を行うツール(宮崎大学で開発)を用いて検出を実行



[実行手順]

① STEP1の定義ファイルを既定の場所に配置する

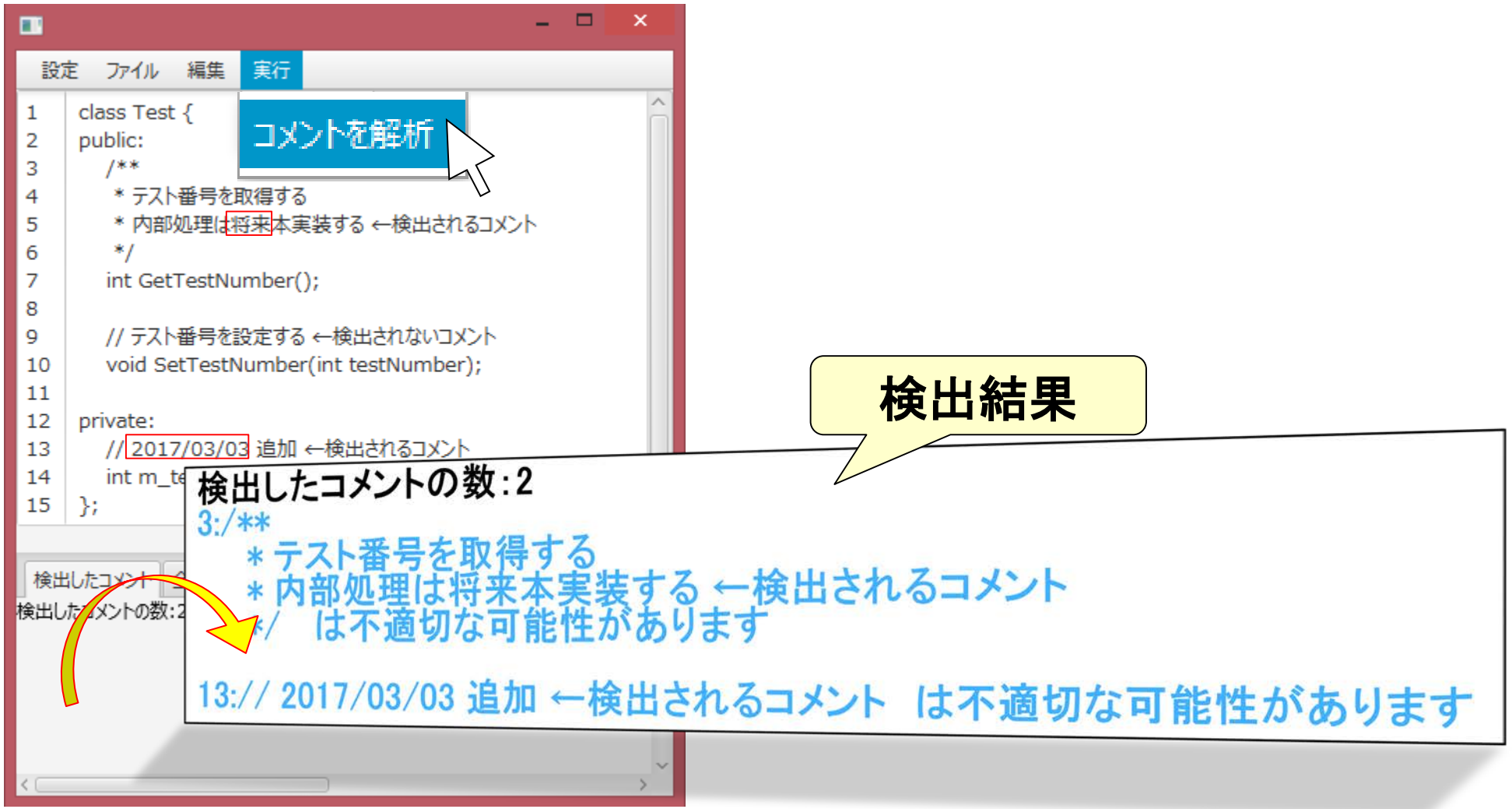
② 解析対象のソースファイルを指定してオープンする
※メニューバーより、「ファイル」→「ファイルを開く」

③ 「②」でオープンしたファイルを解析する
※メニューバーより、「実行」→「コメントを解析」
※定義ファイルに従って解析される

不適切なコメントの検出方法

SKYCOM Guaranteed

■ STEP2：不適切なコメント検出（検出ツールの実行）



ツールの評価・効果

SKYCOM Guaranteed

■ ツールの評価方法

1.当社開発部門の複数ソースコードに対してツールを実行

2.実行結果を元に、以下の式で算出する「適合率」「再現率」が高いほど高評価

適合率 ＝ 検出した不適切なコメント数 ／ 検出した全コメント数

⇒検出結果の中に、どれだけ検出すべき不適切なコメントが含まれたか

再現率 ＝ 検出した不適切なコメント数 ／ 検出すべき不適切なコメント数

⇒検出すべき不適切なコメントのうち、どれだけ検出できたか

再現率は、100%が必須

検出すべきコメントのすべてを検出できなければ漏れが発生し、ユーザーの判断以前の問題となるため、100%である必要がある

実例

全コメント (5行)	
// コメント1	検出すべき不適切なコメント
// コメント2	
// コメント3	
// コメント4	適切なコメント
// コメント5	

ツール実行

検出した全コメント(4行)
// コメント2
// コメント3
// コメント4
// コメント5

適合率 ＝ 2／4 ＝ 50.0%

再現率 ＝ 2／3 ＝ 66.6%

ツールの評価・効果

SKYCOM Guaranteed

■ 評価手順

① 総ステップ数が5000行以上のプロジェクトを評価対象として選択する

↓

② ①のプロジェクト内の全ソースから、検出すべきコメントとその総数を記録する

↓

③ ①のプロジェクト内の全ソースに対し、ツールでコメント解析を実行して、検出したコメントとその数を記録する

↓

④ ②と③で記録したコメントが一致するものの総数を記録する

↓

⑤ ②③④を元に、「**適合率**」「**再現率**」を算出する

ツールの評価・効果

SKYCOM Guaranteed

■ 評価結果

評価対象：2つのプロジェクト

プロジェクト①

全ソース行数：6854 step
内)コメント行数：1990 step

実行回数	検出する単語の定義	検出すべきコメント数	検出した全コメント数	一致したコメント数	適合率	再現率
1回目	ステップ1で定義した単語	23	31	20	64.5%	87.0%
2回目	「おそらく」「かもしれない」等を追加	23	29	23	79.3%	100.0%
3回目	「ループ」「繰り返し」等を除外 (すべて適切なパターンだったため)	23	27	23	85.2%	100.0%

内訳）改修履歴コメント：6件
TODOコメント：7件
曖昧なコメント：2件
理解不足コメント：8件

検出精度UP!!

プロジェクト②

全ソース行数：9394 step
内)コメント行数：1585 step

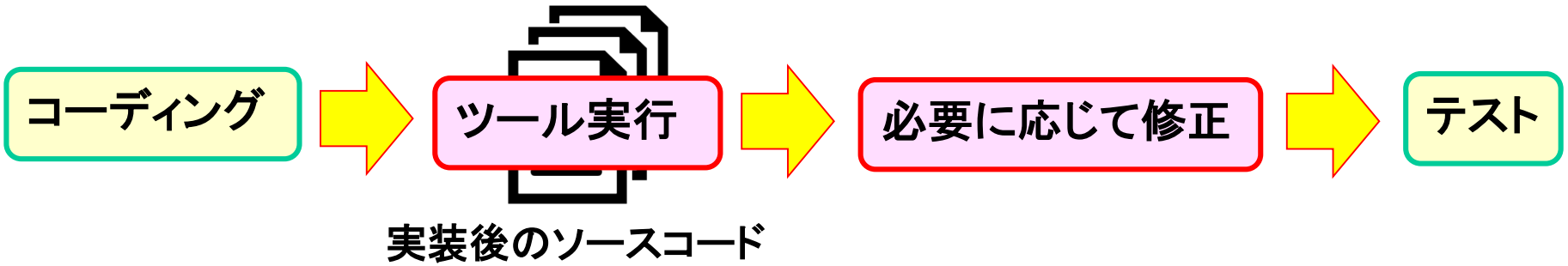
実行回数	検出する単語の定義	検出すべきコメント数	検出した全コメント数	一致したコメント数	適合率	再現率
1回目	プロジェクト①で3回実行後の定義を使用	24	26	24	92.3%	100.0%

検出精度維持!!

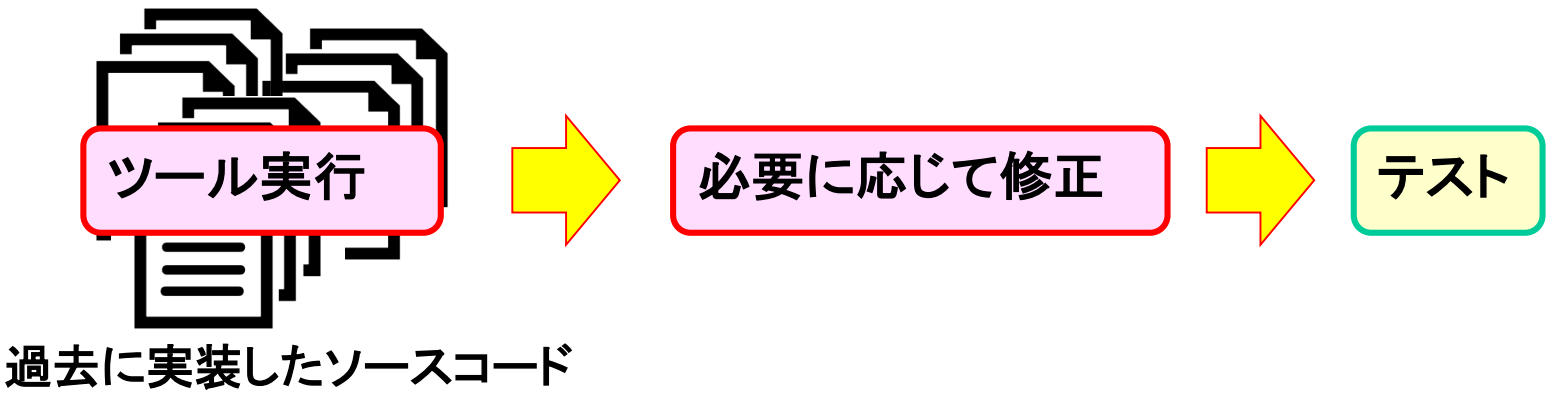
ツールの評価・効果

■ ツールの利用例

開発終了前に実施 ・・・不適切なコメントの混入を防止



古いソースで実施 ・・・既存の不適切なコメントを検知して改善



ツールの評価・効果

SKYCOM Guaranteed

■ ツールの評価・効果まとめ

- **拡張性の確保**
  ⇒ツールを実行して検出結果の傾向を分析し、定義ファイルを拡張していくことで検出精度が上がった

- **コメントのチェック時間短縮、正確性アップ**
  ⇒不適切なコメントを目視で確認する必要がなくなり、過去の実装も含めて容易に見直しが可能となった

- **潜在する課題の検出**
  ⇒検出した不適切なコメントの位置から、以下の検出に成功した
  - 周辺処理に潜在する課題（TODOコメント、曖昧なコメント）
  - 見直すべき疑わしい処理（理解不足コメント）

理解不足コメントとして分類される「？」「おそらく」「かもしれない」などで検出されたコメント内容は、ほとんどの場合(15件中13件が)周辺処理を疑わざるを得なかった

残課題

SKYCOM Guaranteed

■ 残課題

本研究で、コメントに関連して以下の問題も挙がった

コメントに関する問題	問題と考える理由
書くべきコメントの記載がない	必要なコメントが少ないほど、コードの可読性も低下する。
処理内容とは異なるコメントがある	誤った理解をする可能性が非常に高く、メンテナンス時のバグ混入に繋がる。
不要コードがコメントアウト状態になっている	コードの可読性が低下する。 また、コードのグレップ時にコメントアウトされた無効なコードが含まれる場合があるため、作業の妨げにもなる。

品質向上のためには、これらの検出も必要！！

まとめ

■ 本研究を通して実感したこと

・社員が不適切と考えるコメントは、ある程度共通していることが確認できた

・不適切なコメントを容易に検出することは、品質・作業効率の向上に繋がることが確認できた

・コメントに関して、暗黙的に様々な課題が存在することが確認できた

これらの確認ができたことで、改めてコメントの重要性を認識した。

今後、製品のさらなる品質向上を実証できるよう、将来的にはAI技術の利用も視野に入れながら、継続して本研究を進めていきたい。