

The Impact of Code Review Coverage and Code Review Participation on Software Quality

コードレビューの範囲と参加数によるソフトウェア品質への影響

Shane McIntosh¹

Bram Adams³

Yasutaka Kamei²

Ahmed E. Hassan¹

¹Queen's University, Canada

²Kyushu University, Japan

³Polytechnique Montréal, Canada

¹{mcintosh, ahmed}@cs.queensu.ca

²kamei@ait.kyushu-u.ac.jp

³bram.adams@polymtl.ca

コードレビューについて

◎ (厳格な) コードレビュー

- ▶ レビューの品質を保証するために、厳格なレビュー基準を要求
 - ▶ 例
 - 対面での会議がある
 - レビュー担当者のチェックリストがある

◎ モダンコードレビュー

- ▶ 最近のレビュー環境で採用されているコードレビュープロセス
 - ▶ 例
 - メーリングリスト, GerritなどのWebアプリケーションを用いている
 - 非同期型でレビューを行うための対面で会議を行わない[1]
 - 普通のコードレビューと対比して簡易的なもの[1]

背景

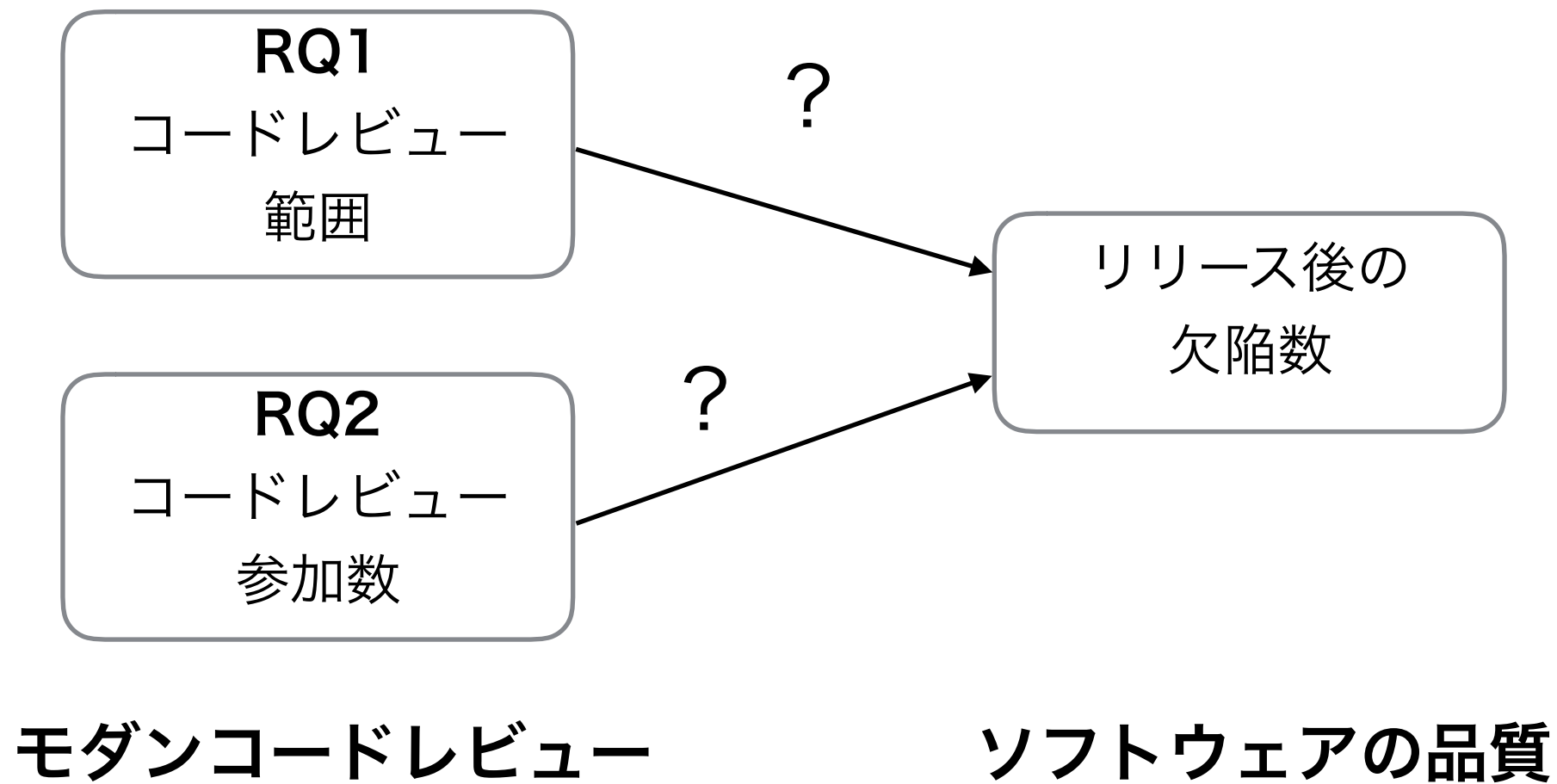
- ◎ コードレビューでソフトウェアの品質が向上することが示されている
 - ▶ しかし、これは厳格なコードレビューを行なった場合
- ◎ 最近は厳格でない簡易的なモダンコードレビューが主流
- ◎ 従って、モダンコードレビューのプロセスについて盛んに研究されている
- ◎ しかし、モダンコードレビューとソフトウェアの品質との関係についてはあまり研究されていない

目的

- ◎ モダンコードレビューとソフトウェアの品質の関係を定量的に調査する
 - ▶ モダンコードレビューでは、レビューされる範囲やレビューへの参加数とそのソフトウェアの品質に関係するのではないかという仮説を検証する

リサーチクエストション

- RQ1 : コードレビューの範囲とリリース後の欠陥数との関係
- RQ2 : コードレビューへの参加数とリリース後の欠陥数との関係



ケーススタディ

- 対象プロジェクト

- ▶ Ot
- ▶ VTK
- ▶ ITK

- 基準

- ▶ ①レビューの方針がある
- ▶ ②追跡できる

- 基準を満たさなかったプロジェクト（十分にコミットとレビューが関連づけられない）

- ▶ Android : 2%
- ▶ LibreOffice : 14%

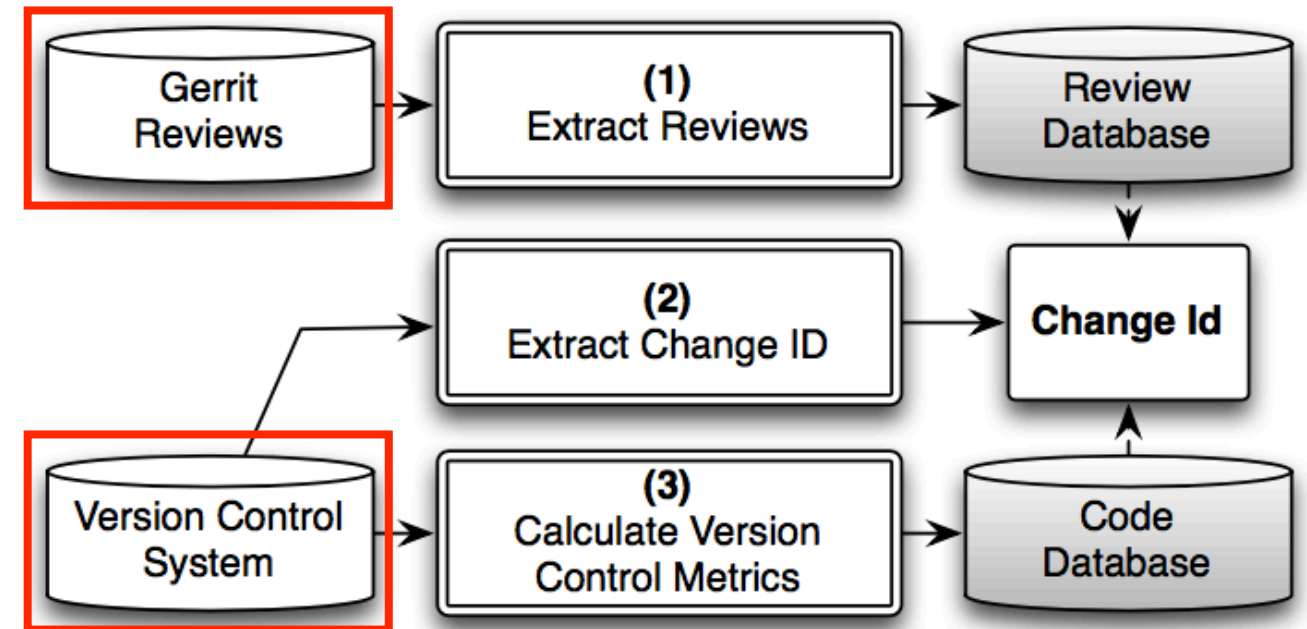
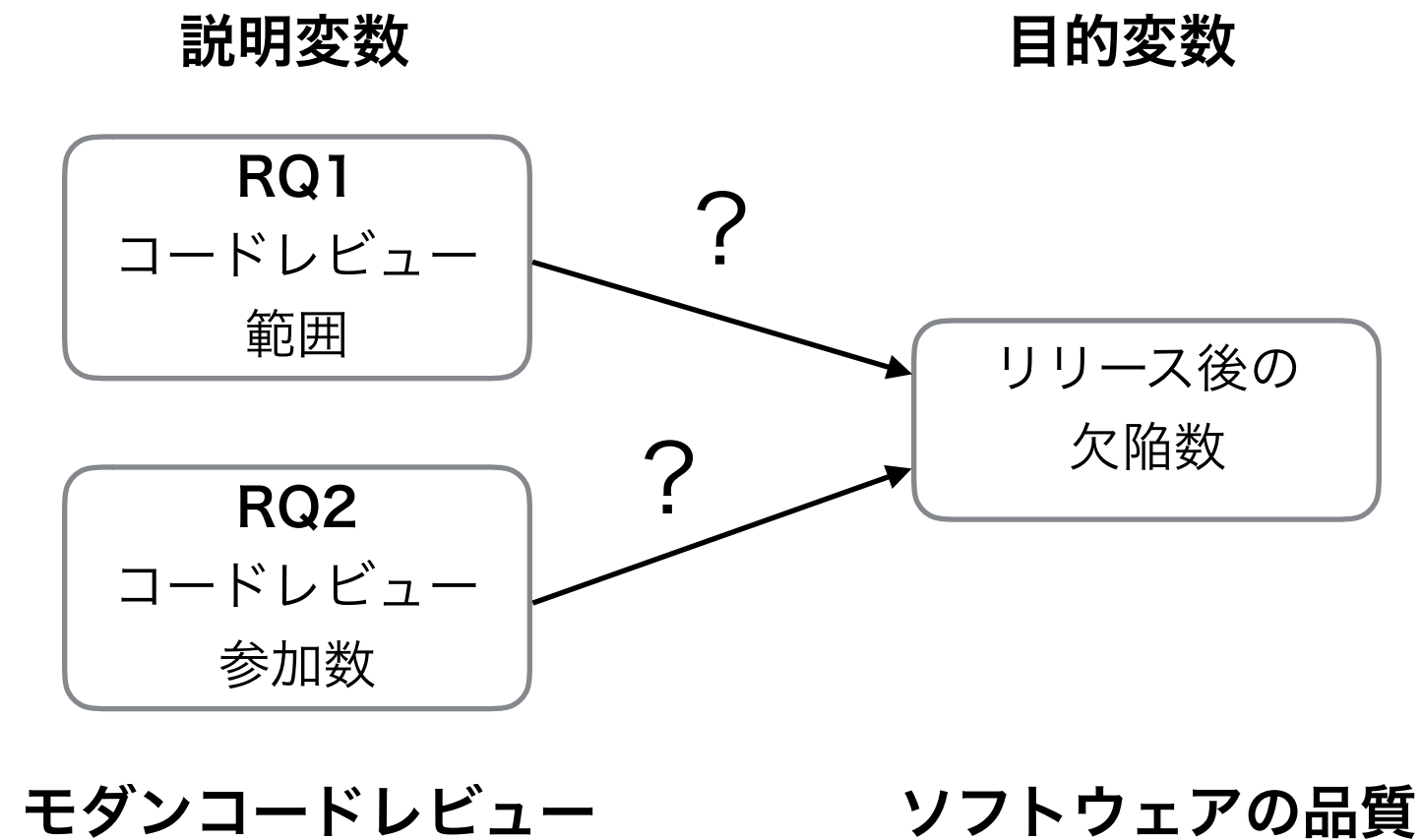


Figure 2: Overview of our data extraction approach.

レビューデータと
VCSのデータを
変更IDで関連づける

アプローチ



- 対象プロジェクトから, 対象メトリクスを抽出
- 抽出したメトリクスから, モデルを構築
- モデルの分析, 説明変数の影響を分析

抽出するメトリクス

製品メトリクス

変更プロセスの
メトリクス

人的要因

RQ1

RQ2

	Metric	Description
Prod	Size	Number of lines of code.
	Complexity	The McCabe cyclomatic complexity.
Process	Prior defects	Number of defects fixed prior to release.
	Churn	Sum of added and removed lines of code.
	Change entropy	A measure of the volatility of the change process.
Human Factors	Total authors	Number of unique authors.
	Minor authors	Number of unique authors who have contributed less than 5% of the changes.
	Major authors	Number of unique authors who have contributed at least 5% of the changes.
	Author ownership	The proportion of changes contributed by the author who made the most changes.
Coverage (RQ1)	Proportion of reviewed changes	The proportion of changes that have been reviewed in the past.
	Proportion of reviewed churn	The proportion of churn that has been reviewed in the past.
Participation (RQ2)	Proportion of self-approved changes	The proportion of changes to a component that are only approved for integration by the original author.
	Proportion of hastily reviewed changes	The proportion of changes that are approved for integration at a rate that is faster than 200 lines per hour.
	Proportion of changes without discussion	The proportion of changes to a component that are not discussed.

抽出するメトリクス

- ◎ **製品メトリクス**（リリース時点のソースコードから計算）
 - ▶ コンポーネントの大きさ
 - ▶ コンポーネントの複雑度
- ◎ **変更プロセスのメトリクス**（リリースより6ヶ月前まで）
 - ▶ リリース前に混入された欠陥
 - ▶ Churn
 - リリース前にコンポーネントに追加 / 削除 されたソースコードの総数
 - ▶ 変更エントロピー
 - 変更プロセスの複雑さがどのようにファイルに分散されるかを測定（？）
 - (原文) Change entropy measures how the complexity of a change process is distributed across files.
- ◎ **人的要因**
 - ▶ コードの所有権など
 - どれくらいの割合でそのコンポーネントを変更しているか

各RQでそれぞれ追加するメトリクス

◎ RQ1

- ▶ レビューされた変更の割合
- ▶ レビューによって 削除 / 追加 された割合

◎ RQ2

- ▶ 自己承認された変更の割合
 - 変更元の作成者が統合のために行った承認
- ▶ 急速にレビューされた割合
 - レビューが早すぎたらいけない
- ▶ 議論がないレビュー

分析方法

◎ モデルを構築 (MLR: Multiple Linear Regression)

- ▶ MLR: リリース後の欠陥の発生率 (ソフトウェア製品の公式のリリースの不具合) を図るためのもの

▶ 式

- 複数線形回帰

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

目的変数

リリース後の欠陥数

説明変数

各メトリクス

- ◎ (1) 全てのメトリクスを含むモデルと, (2) それぞれ1つのメトリクスを除いたモデルのに対してカイ二乗検定を行う (有意水準5%)

◎ 説明変数の影響分析

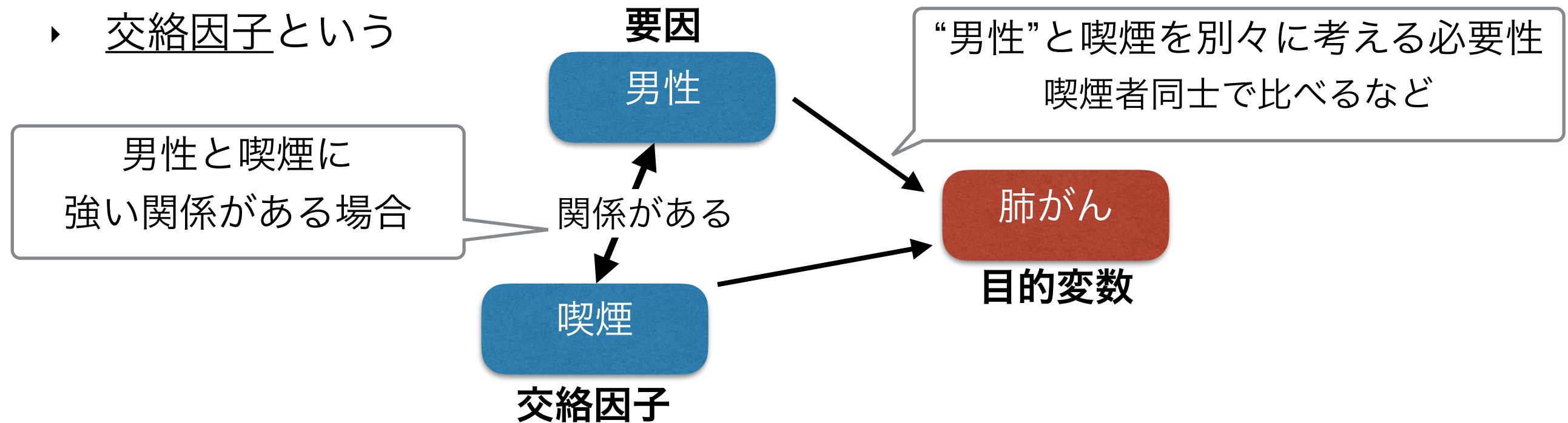
- ▶ 説明変数がリリース後の欠陥発生率に及ぼす影響を計算 (CPDC)

$$CPDC(x_i) = \begin{cases} 0, & \text{if } PDC(x_i) \leq 0 \\ \lceil PDC(x_i) \rceil, & \text{otherwise} \end{cases}$$

予測モデル構築の前処理

- 変数が複数ある場合, 変数間の関係により分析結果の信頼性が低くなる可能性がある

- 交絡因子という



- 対処

- 多重共線性がないか確認
 - 共変量間の相関関係を確認 (今回は, $|\rho| > 0.7$ で除外)
 - 分散拡大係数 (VIF) を求める (今回は $VIF < 5$ になるまで)

RQ1, レビュー範囲と欠陥数：結果1

Table 3: Review coverage model statistics. ΔAIC indicates the change in AIC when the given metric is removed from the model (larger ΔAIC values indicate more explanatory power). *Coef.* provides the coefficient of the given metric in our models.

	Qt				VTK		ITK	
	5.0.0		5.1.0		5.10.0		4.3.0	
Adjusted R^2	0.40		0.19		0.38		0.24	
Total AIC	4,853		6,611		219		15	
	Coef.	ΔAIC	Coef.	ΔAIC	Coef.	ΔAIC	Coef.	ΔAIC
Size	◇		0.46	6**	0.19	223.4*	◇	
Complexity	◇		◇		◇		◇	
Prior defects	5.08	106***	◇		3.47	71***	0.08	13***
Churn	◇		◇		†		†	
Change entropy	◇		◇		◇		◇	
Total authors	‡		†		‡		‡	
Minor authors	2.57	49***	10.77	210***	2.79	50***	1.58	23***
Major authors	†		†		†		†	
Author ownership	◇		◇		◇		◇	
Reviewed changes	-0.25	9***	◇		-0.30	15***	◇	
Reviewed churn	†		†		†		†	

† Discarded during correlation analysis ($|\rho| > 0.7$)

‡ Discarded during VIF analysis (VIF coefficient > 5)

Statistical significance of explanatory power according to Drop One analysis:

◇ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

相関係数

AICの差？

RQ1
レビュー範囲
メトリクスを追加

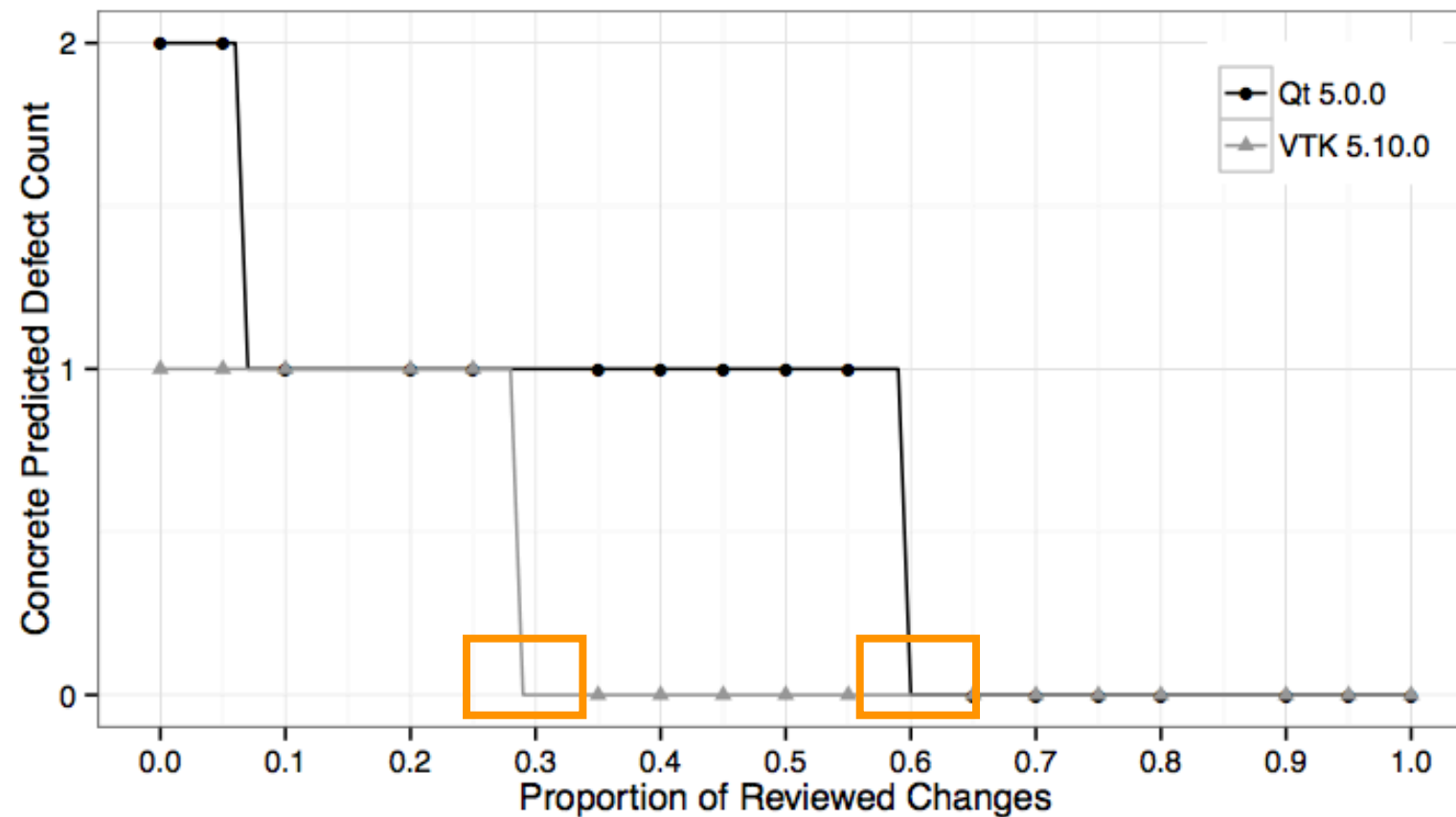
どの変数との関係を
指すのか？

- レビュー範囲が大きいコンポーネントは、リリース後の欠陥が少ない傾向がある

予測モデルの分析

- AIC（赤池の情報量基準）
 - ▶ ロジスティック回帰の適合度を表す指標
 - ▶ いくつかのモデルがどの程度実際の観察データにフィットするかを比較する尺度
 - ▶ 値が小さいほどいいモデル

RQ1, レビュー範囲と欠陥数：結果2



- レビュー範囲が0.29（VTK）または0.6（Qt）未満のコンポーネントには少なくとも一つはリリース後の欠陥が含まれている

RQ2, レビュー参加と欠陥数：結果1

RQ2
レビュー参加
メトリクスを追加

	Qt				ITK	
	5.0.0		5.1.0		4.3.0	
Adjusted R ²	0.44		0.26		0.25	
Total AIC	4,328		1,639		71	
	Coef.	ΔAIC	Coef.	ΔAIC	Coef.	ΔAIC
Size		◇	0.08	4*		◇
Complexity		◇		◇		◇
Prior defects	4.20	68***	0.95	28***	0.70	5*
Churn		†		◇		†
Change entropy		◇		◇		◇
Total authors		‡		†		‡
Minor authors	2.06	24***	3.22	85***	1.01	8**
Major authors		†		†		†
Author ownership		†		†		◇
Self-approval	1.34	11***		◇		◇
Hastily reviewed		◇	0.55	8**		◇
No discussion	0.83	4*	0.74	15***	0.70	4*

† Discarded during correlation analysis ($|\rho| > 0.7$)

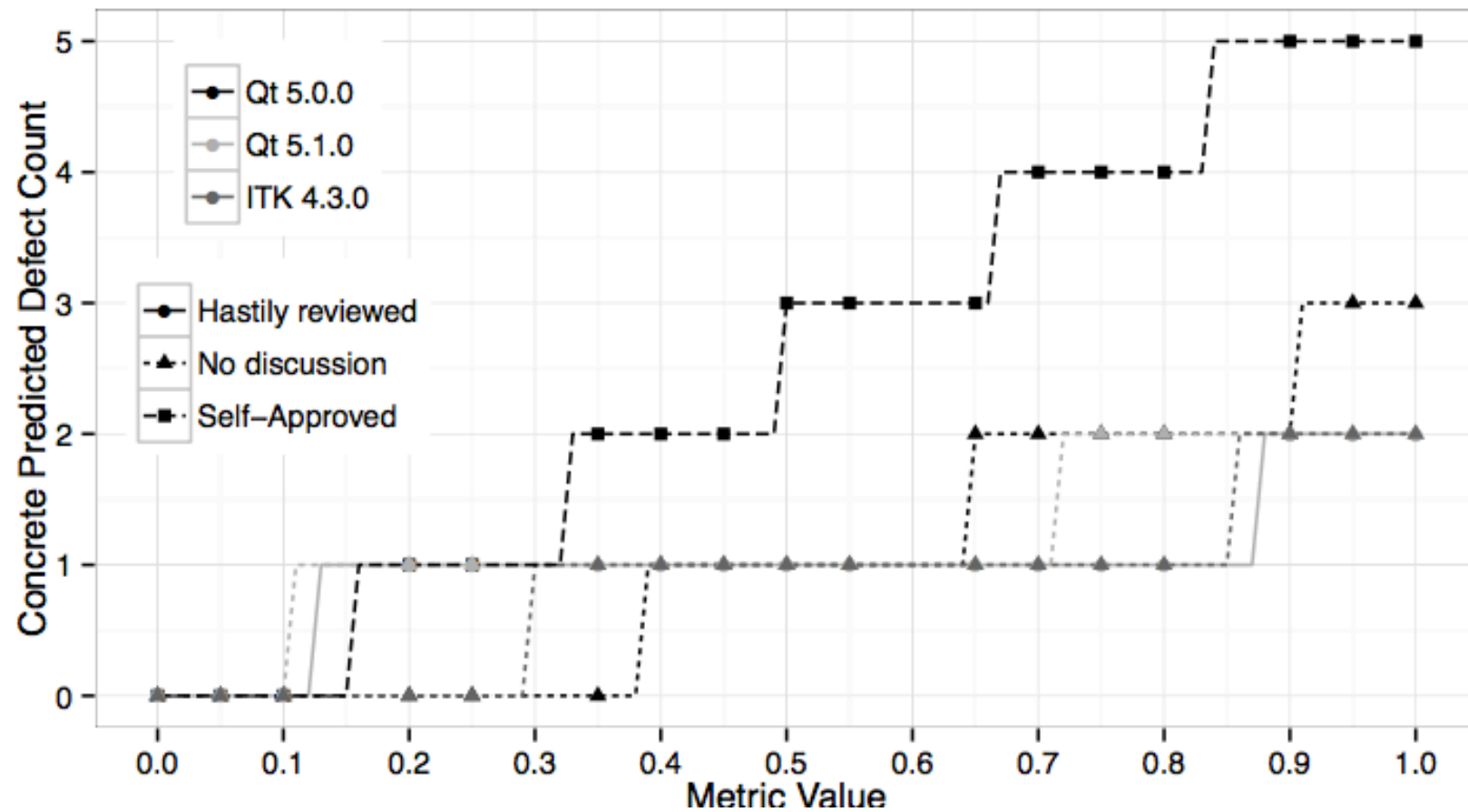
‡ Discarded during VIF analysis (VIF coefficient > 5)

Statistical significance of explanatory power according to Drop One analysis:

◇ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

- ◎ コードレビューへの参加率が高いコンポーネントは、リリース後の欠陥が少なくなる傾向がある

RQ2, レビュー参加と欠陥数：結果2



- コードレビューの参加率が低いコンポーネントは, リリース後の欠陥数が多い傾向がある
- 議論のないレビューはリリース後の欠陥数の増加と関係している

まとめ

- ◎ コードレビューレビュー 範囲 / 参加数 が少ないコードは, リリースされたソフトウェアの品質に係する
 - ▶ コードレビュー範囲のメトリクスから, ソフトウェアの欠陥数を説明できる場合がある
 - 今回は4プロジェクト中2つで説明できた
 - ▶ コードレビュー参加メトリクスのうち2つは, Qtでリリースされた製品の欠陥モデルに大きな係がある
 - ▶ レビューの参加数が小さいコンポーネントは, 大きいコンポーネントに比べて欠陥数が5つ大きくなる

所感

- 難しい

- ▶ 統計的な知識, ソフトウェア工学的な知識が必要？
- ▶ 結果の表が直感的に理解できない
 - 共変量相関分析, VIFで除去された変数はわかるが, その代わりに何の変数を使っているのかが読み取れない

製品メトリクス（リリース時点のソースコードから計算）

- ◎ **コンポーネントのサイズ**

- ▶ コンポーネント（ディレクトリ）のコード行数

- ◎ **コンポーネントの複雑度**

- ▶ **Scitools Understand[†]**で測定（McCabeの循環的複雑さを使用）
- ▶ 複雑度はコンポーネントのサイズと高い相関があるため、コンポーネントのサイズで割る
 - ▶ コードの複雑度の測定結果の影響を軽減するため

[[†]]<http://www.scitools.com/documents/metricsList.php?#Cyclomatic>

データ抽出

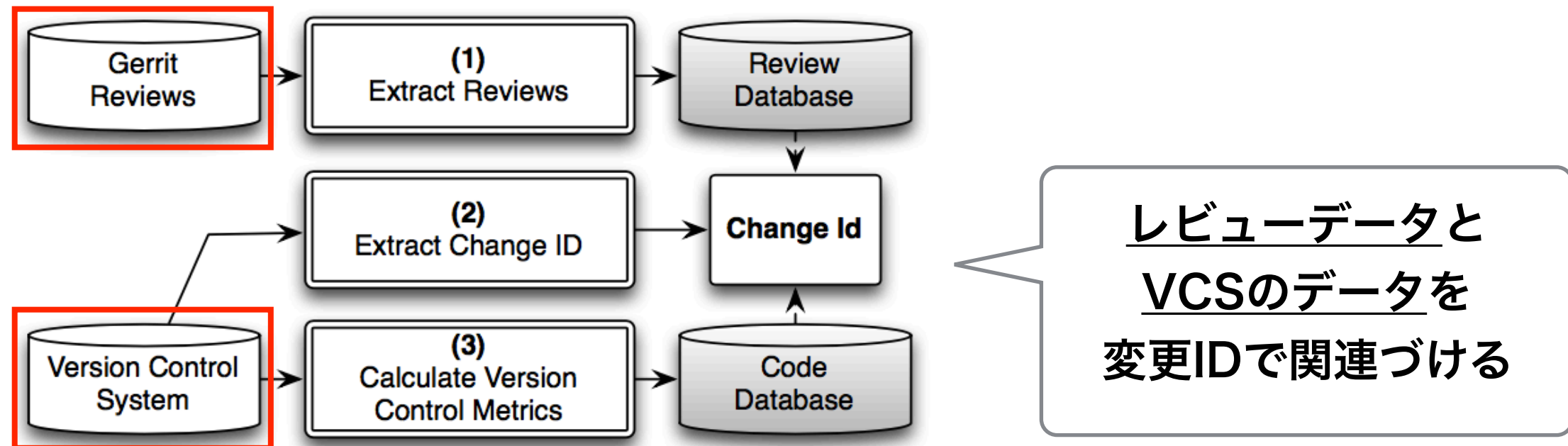


Figure 2: Overview of our data extraction approach.

(1) レビューを抽出

- 各レビュー, 関係する人, レビュー内容の詳細 (Qtのデータセット > 他のPJにも拡張)

(2) 変更ID (Gerrit) を抽出

- コミットメッセージから抽出

(3) バージョン管理メトリクスを計算

- コミットメッセージから, 「バグ, 修正, 欠陥, パッチ」などのキーワードが含まれる識別子の共起を検索しそのコミットを抽出

分散拡大係数（VIF）とは

[Minitab 17 についての 詳細](#)

分散拡大係数（VIF）では、予測変数の間に線形の関係がない場合と比較して、推定回帰係数の分散が拡大する程度を測定します。

回帰分析において多重共線性（予測変数間の相関）がどの程度存在するかを記述するために使用します。多重共線性は、回帰係数の分散性を増大させてその係数の安定性を損ねたり解釈を困難にしたりするため、問題の原因となります。

以下のガイドラインに従ってVIFを解釈します。

VIF	予測変数のステータス
VIF = 1	相関なし
$1 < \text{VIF} < 5$	穏やかに相関
$\text{VIF} > 5 \sim 10$	強く相関

VIF（分散拡大係数）の計算方法

- R_j^2 : 決定係数

$$\text{VIF}_j = \frac{1}{1 - R_j^2}$$