

SQiPシンポジウム2025

ソフトウェア開発の全工程で適用可能な
トークン数を用いた規模計測方法の提案

アズビル株式会社 佐々木瑛太

2025/9/1



今日お話しするテーマ

ソフトウェア開発の全工程で適用可能な トークン数を用いた 規模計測方法の提案

「規模」って何？
⇒ソフトウェアの
「大きさ」「複雑さ」

「トークン」って何？
⇒「単語」みたいなもの

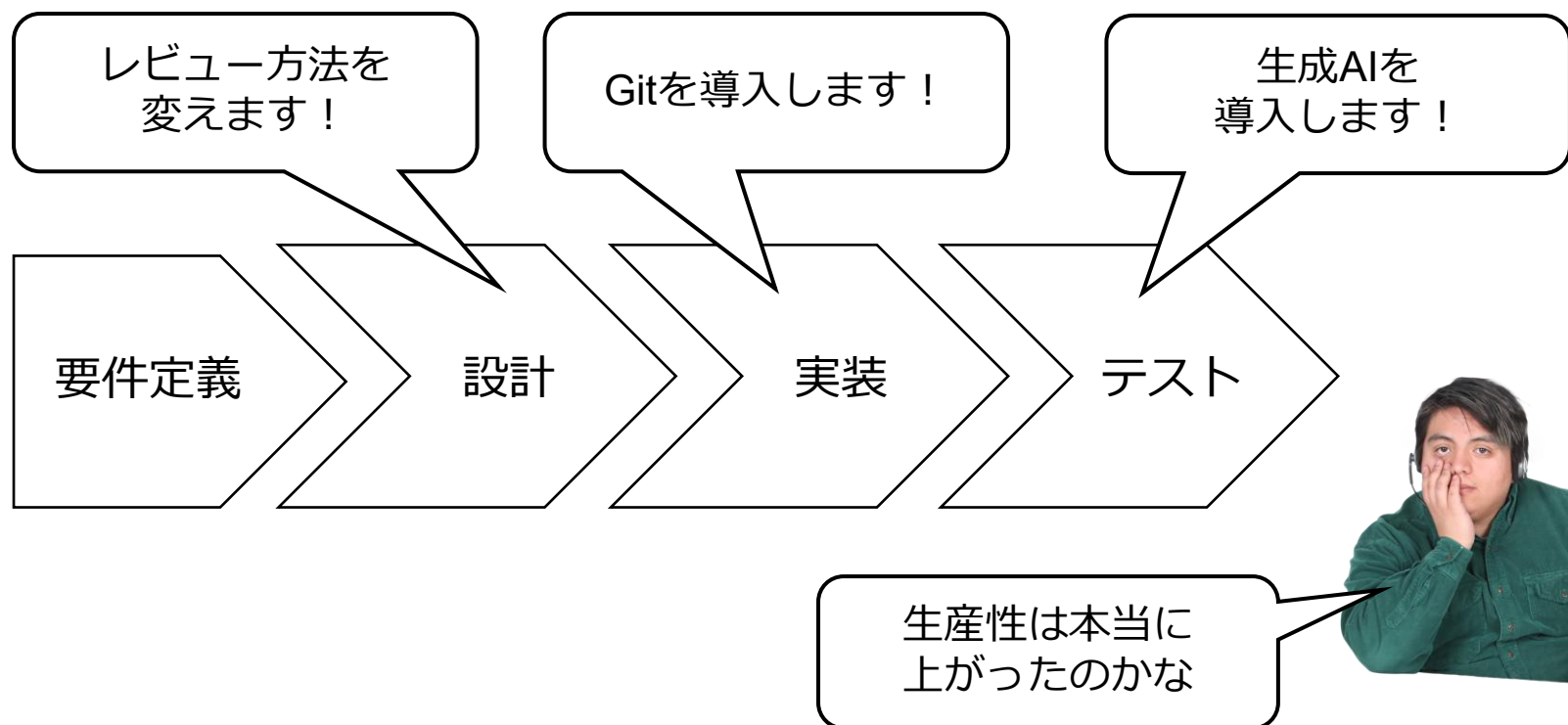
実装だけでなく
設計・テストでも使える

目次

- 背景と課題提起
- 先行研究
- 提案手法
- 実験方法と結果・考察
- おわりに

私が抱えている課題：「生産性チャレンジ」の評価

「生産性チャレンジ」の例



**生産性チャレンジ：有志が自分で設定したテーマで業務効率化を行う取り組み
全工程にわたる様々な取り組みの有効性を定量的に評価したい
⇒成果物の規模に着目する**

既存の規模測定手法の限界：コードの計測

このプログラムは何ステップ？

```
for (i = 0; i < 100; i++) printf("hello"); /* 何ステップ？ */
```

ではこれでは？

```
/* 何ステップ？ */  
for (i = 0; i < 100; i++)  
{  
    printf("hello");  
}
```

SQIP研究会メトリクス演習コース 第2回資料 「規模の測定」 p11より引用

改行を多めに入れて
水増しできちゃうね



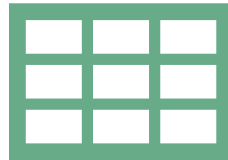
コードの規模のメトリクスとして、行数がよく使われる。が...
プログラムの表層的特徴（コーディングスタイル、改行の頻度...etc.）に
よって、少なからず影響を受けてしまう

既存の規模測定手法の限界：ドキュメントの計測

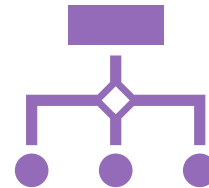
ドキュメント成果物の例



機能仕様書
Word文書



入力項目一覧
Excelシート



アクティビティ図
EAファイル

形式がバラバラ
お手上げた



設計やテストで作られるドキュメントは、成果物の形式がバラバラ（ワード文書, Excelシート, EAファイル...etc.）
⇒統一的な規模のメトリクスがない

課題提起

● 現在抱えている問題

- コードの規模は改行などによってばらつきやすい
- ドキュメントは形式がバラバラで規模の測定方法がない

● 提案手法が満たすべき要求

- 表層的特徴や形式の影響を受けにくい
- コードでもドキュメントでも規模を測定できる
- 自動的に測定できる
- 差分に対して規模を測定できる

課題を解決するために新しい規模のメトリクスを提案する

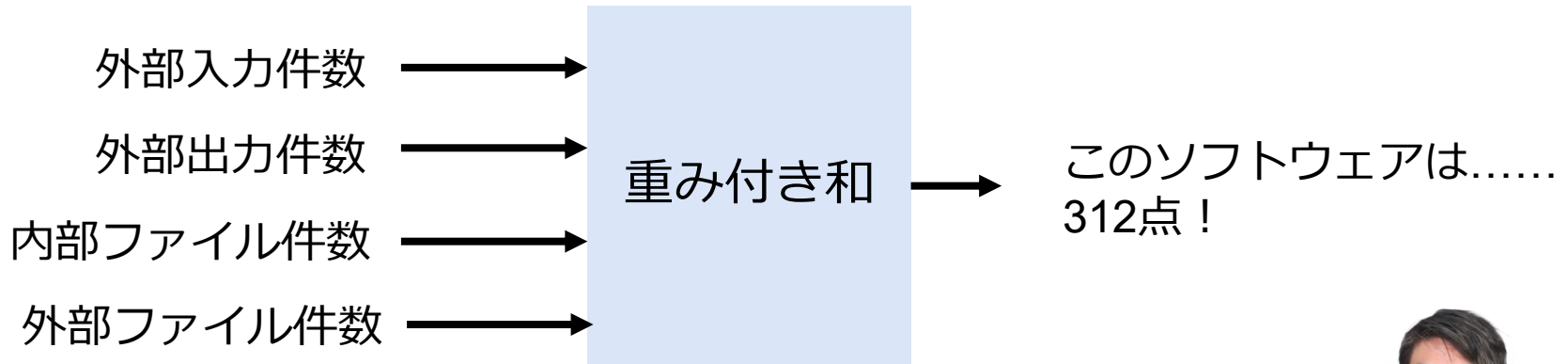
目次

- 背景と課題提起
- **先行研究**
- 提案手法
- 実験方法と結果・考察
- おわりに

機能ベースの規模：ファンクションポイント法

ソフトウェアの機能

ソフトウェアの規模



ちょっと
大変すぎる



ファンクションポイント法（FP法）は機能ベースで規模見積もりができるが...

- ・ 成果物ベースではなく、ソフトウェア全体に対する測定になってしまう
- ・ **自動的な計測が困難**

⇒今回は機能量での測定は採用しない

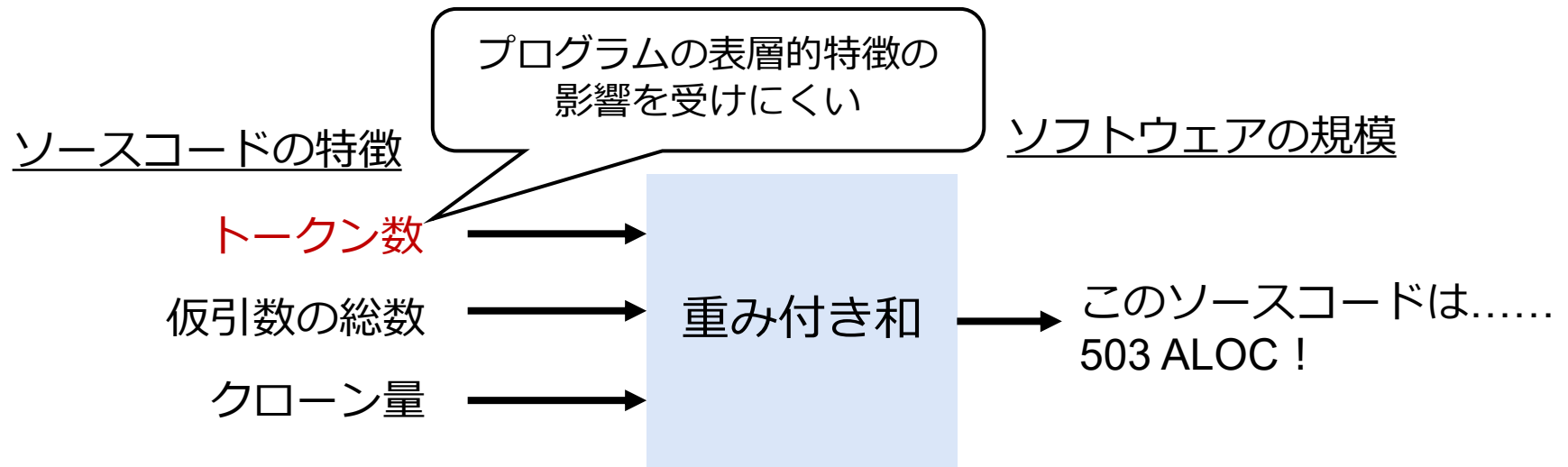
「トークン」でばらつきを抑える

● トークンって何？

- 広義の単語（先の例では変数、演算子...etc.）

● Adjusted Length of Code (ALOC) [1]

[1] 門田暁人, 内田眞司, 松本健一「実装者に依存しないプログラム規模尺度の構築の試み」 FOSE2010

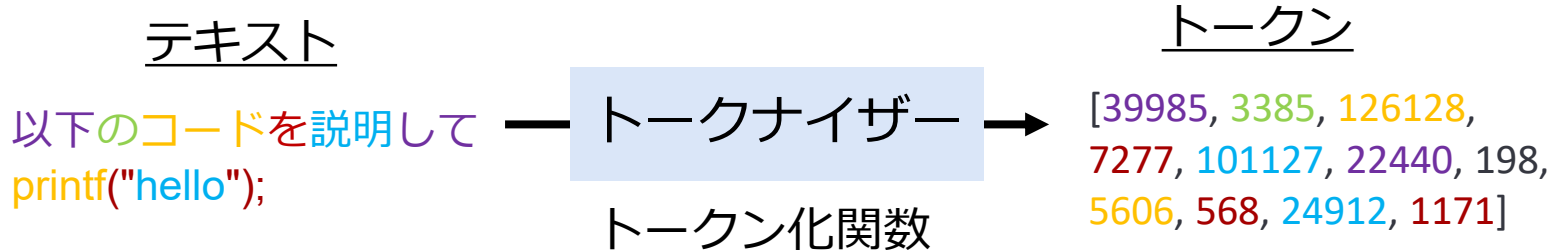


先行研究において、実装工程の規模のメトリクスに「トークン」を取り入れることで、規模の測定結果のばらつきを抑えられることが示されている

トークナイザーによるトークンへの変換

● トークナイザー：LLMにおけるトークン化関数

- 「バイト対符号化」というアルゴリズムによって、学習データから「単語」の集合を定めている
- コードも自然言語も（日本語も！）トークン化できる



トークナイザーを使えば、テキスト形式であればなんでも
まとめてトークンに変換できる

ドキュメントのテキスト形式への変換

● Markdown [2]

非テキスト形式の
ドキュメント



Word文書



Excelシート

Markdown
(テキスト形式)

仕様書
概要
この仕様書は...

見出し付き文書を
テキストで表現

表を
テキストで表現

名前	メールアドレス
佐々木	a@test.com

Markdownとは、文章を簡単な記号で構造化できる表記方法
⇒見出し付き文書（Word）や表形式の文書（Excel）など、
様々な形式のドキュメントをテキスト形式で表せる！

目次

- 背景と課題提起
- 先行研究
- **提案手法**
- 実験方法と結果・考察
- おわりに

提案手法

● Step1

- ・ 成果物の収集

● Step2

- ・ （必要に応じて）非テキスト形式の成果物をMarkdown化

● Step3

- ・ トークン化

● Step4

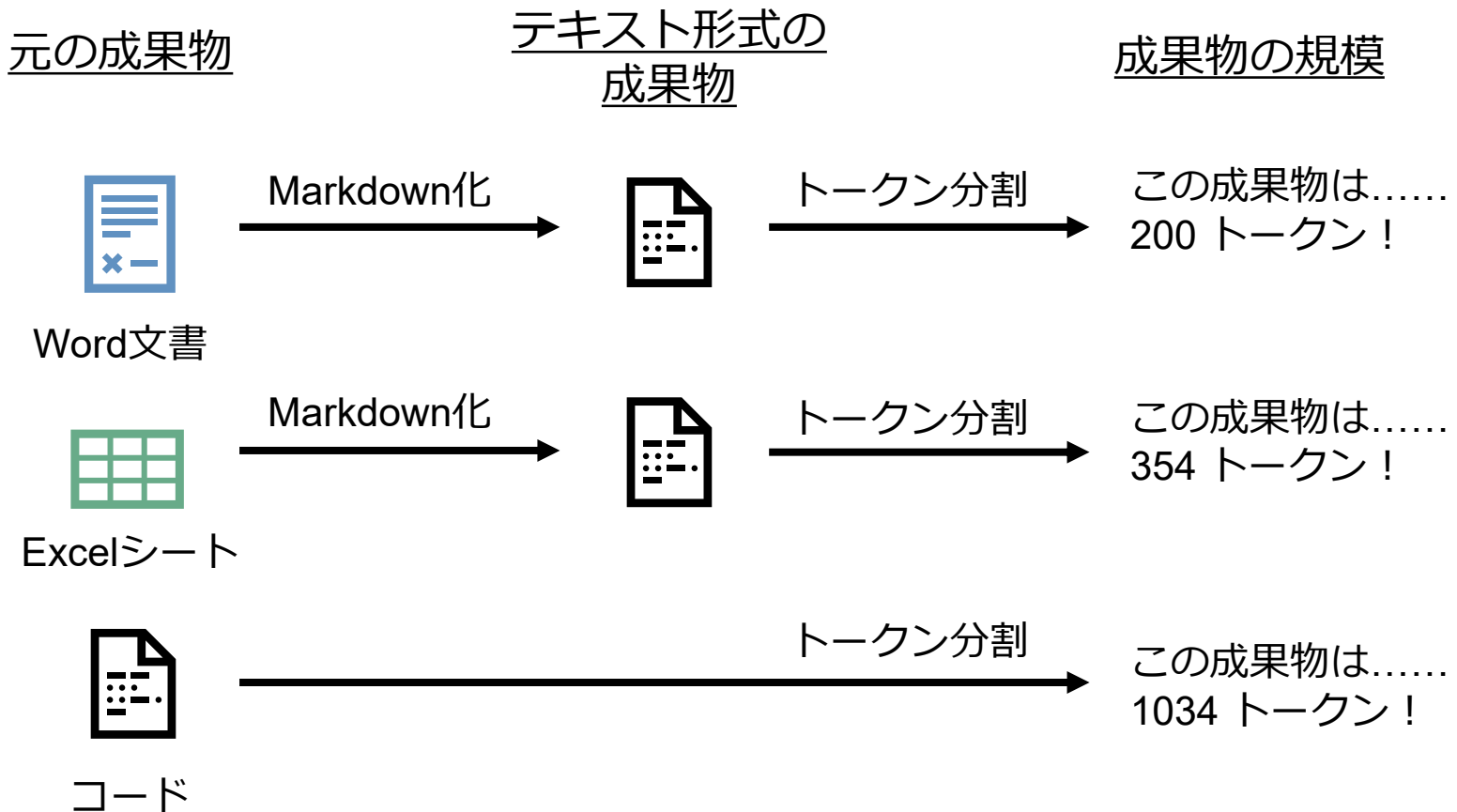
- ・ 増分トークンの検出

● Step5

- ・ （増分）トークン数の計測

- ・ コードでもドキュメントでも統一的に規模を測定できる！
- ・ トークンベースにすることで、よりばらつきの少ない測定が可能になる！

提案手法の具体例



差分の規模の測定方法

変更前のテキスト

以下のコードを説明して
`printf("hello");`



変更後のテキスト

このコードを説明してください
`printf("hello world");`



差分の規模

増分トークン数は.....
「この」「してください」「world」で
3 トークン！

減った分のトークンは
数えない

変更前後の成果物をトークン分割し、
変更後に追加された分のトークン数を数える

Excel成果物の扱い

- 空のセルをMarkdown化すると、文字列に変換されてしまう
⇒ 規模を正しく見積もれない！
⇒ 空のシートからの増分を計測する

空のExcelシート

項目1	項目2	項目3

Markdown変換前は
空きセル

Markdown化

項目1	項目2	項目3
Nan	Nan	Nan
Nan	Nan	Nan
Nan	Nan	Nan

比較

Markdown変換後は
「Nan」という文字列

成果物Excelシート

項目1	項目2	項目3
値1	値2	
値3	値4	

Markdown化

項目1	項目2	項目3
値1	値2	Nan
値3	値4	Nan
Nan	Nan	Nan

空シートから
増えた分

目次

- 背景と課題提起
- 先行研究
- 提案手法
- **実験方法と結果・考察**
- おわりに

実験方法

● 評価の観点

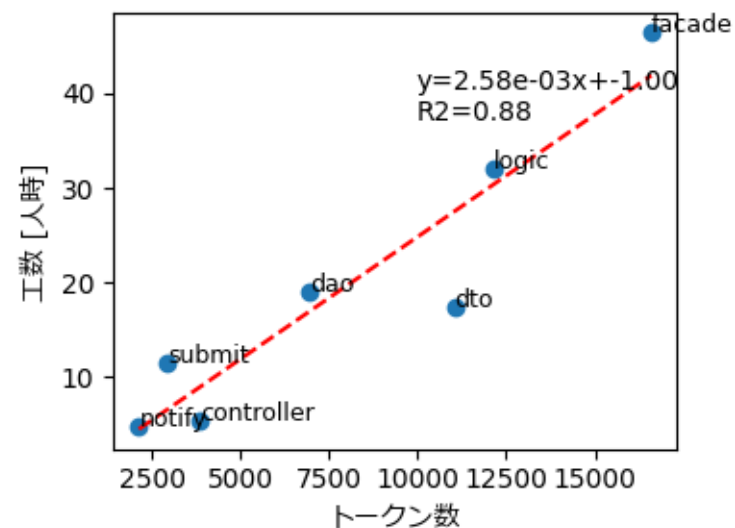
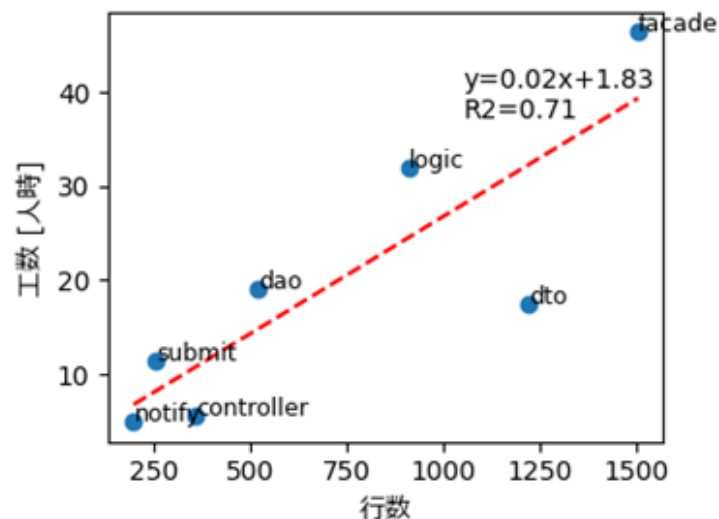
- 自動で、簡単に測れるか
- 様々な形式のドキュメントやコードに適用可能か
- 作業前と後での計測が可能か
- 同一条件（同一PJでの同一人物の作業）で生産性が同一となるか

● 実験方法

- あるプロジェクトのコードについて、**作成に要した工数と提案する規模メトリクス（=トークン数）をプロット**し、その関係性を確認する。また、従来の規模メトリクスである**行数との比較**も行い、**生産性のばらつきがトークン数によってどの程度抑えられるか**を検証する。
- 同様に、あるプロジェクトの**Word文書やExcel表などのドキュメント**についても、**作成工数とトークン数をプロット**する。ここでは、差分データも含めて分析を行い、**ドキュメント形式や「差分/新規作成」といった違いによらず、同様の傾向が見られるか**を検証する。

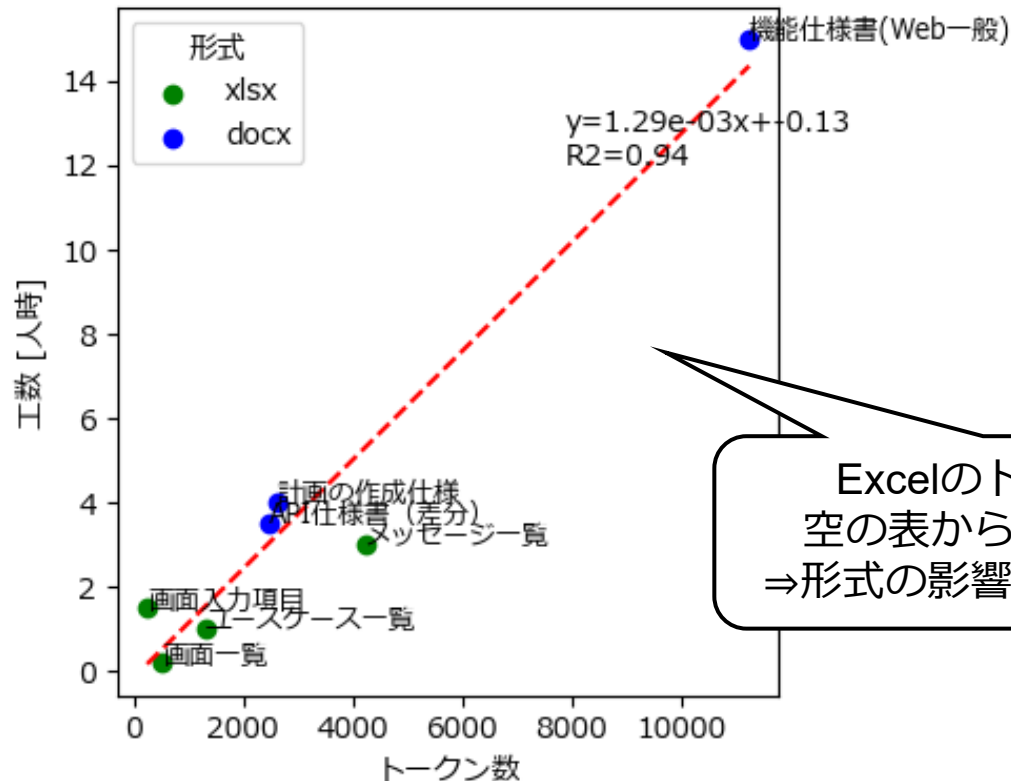
実験結果

● 実装工程への適用



R2値を比較した結果、行数を使用した生産性よりもトークン数を使用した生産性のほうが、ばらつき（表層的特徴の影響）が小さいことを確認した

実験結果の考察：Excel成果物の傾向の違い(2/2)



- ・ 新規作成/差分測定の違いがあっても、傾向が同じ
- ・ Word文書/Excelシートの違いがあっても、傾向が同じ

目次

- 背景と課題提起
- 先行研究
- 提案手法
- 実験方法と結果
- おわりに

まとめ

- **トークン数を使用した成果物の規模計測方法を考案した**
 - コードとドキュメント（Word文書、Excelシート）の両方に適用可能
 - 自動で簡単に計測できる
 - ※ ほぼOSSのみで実現できる
 - 同一条件下（同一人物による同一PJでの作業）において、おおむね工数に比例する
- **⇒様々な形式の成果物について、作成時の生産性を統一的に評価できるようになった**

残課題

- **テキスト化が難しいドキュメントの規模測定**
 - 例1：画像
 - 例2：EAファイル（UML設計ソフトの専用ファイル）
- **応用範囲の拡大**
 - 例：品質に与える影響

azbil

あ す み る、ア ズ ビ ル。