

藤原研研究室紹介

- ソフトウェア工学研究室 -

藤原 賢二

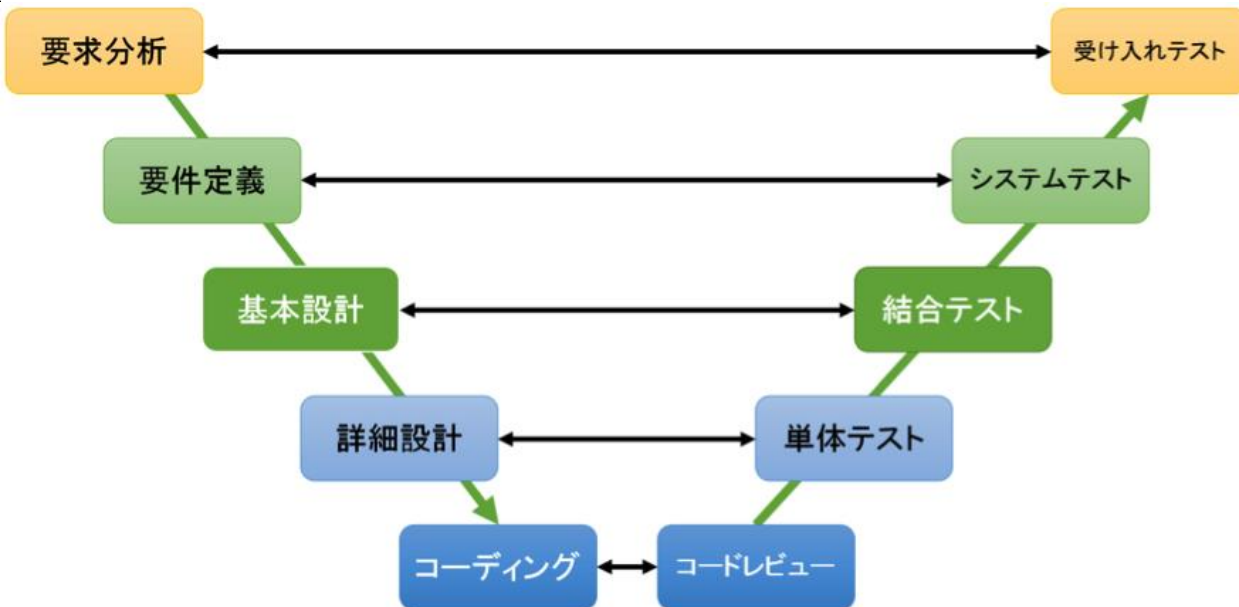
■ソフトウェア工学

- 要求分析から受け入れテストまでソフトウェア開発のあらゆる工程を工学的に支援する分野

■ソフトウェア工学（本流）の研究と並行してプログラミング教育の研究もしています

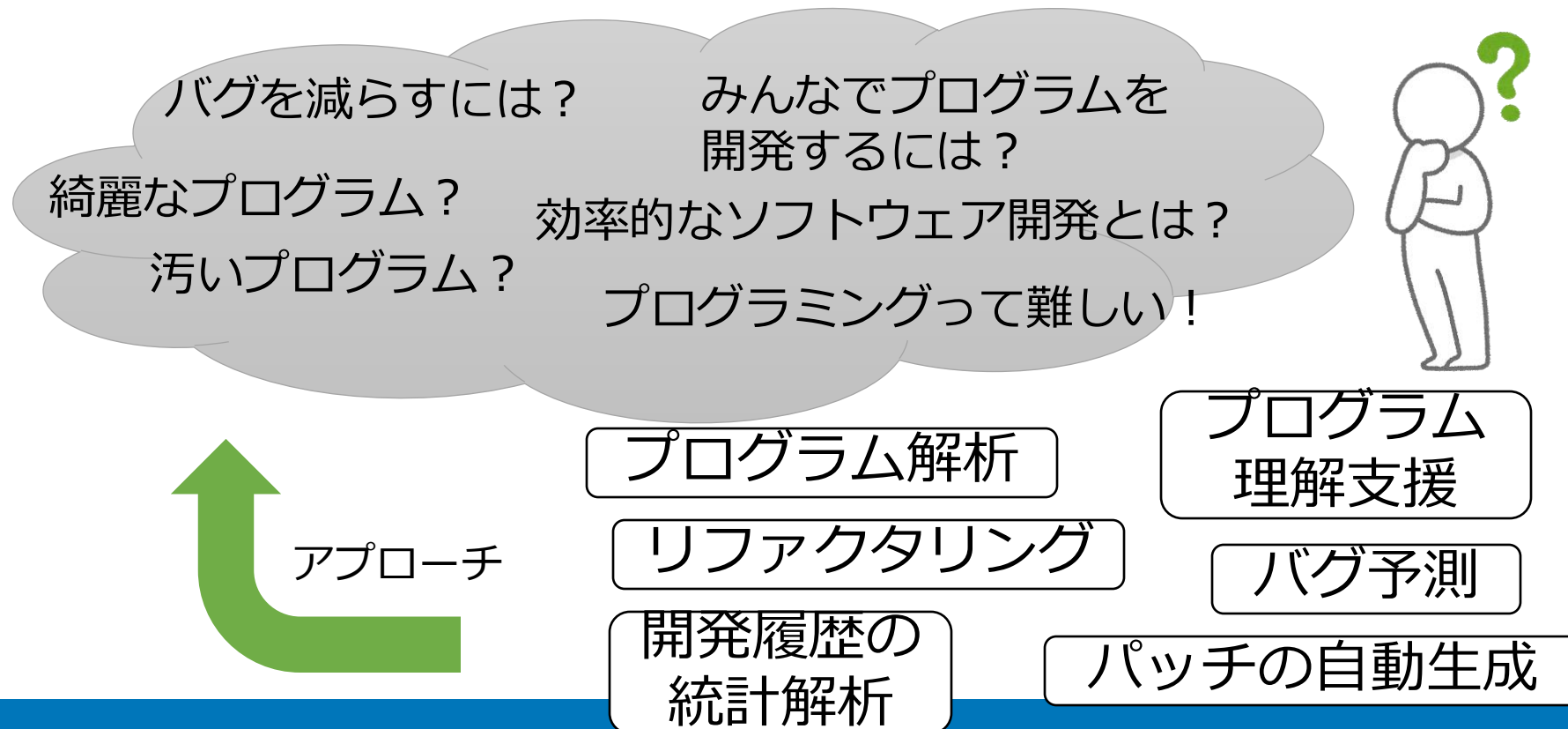
- 開発者（学生）の開発履歴分析
- （学生の）プログラム解析

ソフトウェア工学の範疇



ソフトウェア工学に 関する研究

- ソフトウェア工学の中でもプログラム解析・リポジトリマイニングを中心に研究
 - ソフトウェアの開発履歴を分析してより良いソフトウェア開発をするにはどのようにすればよいかをデータマイニングに基づいて検討



ソフトウェアリポジトリマイニング

5



データマイニング！



ソースコードの履歴

メール履歴



議論の履歴



コードレビュー
の履歴

ソフトウェアテスト
の履歴



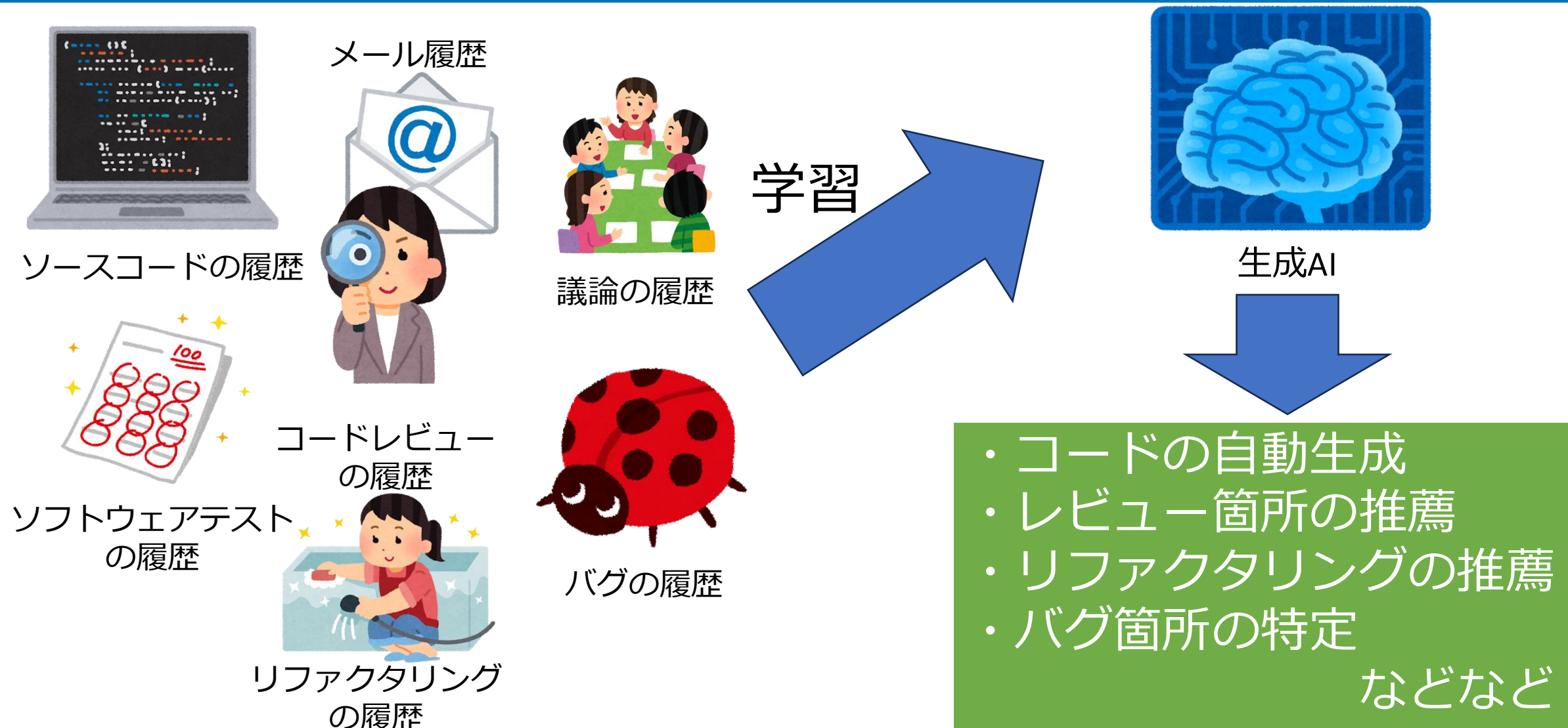
リファクタリング
の履歴



バグの履歴

生成AI技術のソフトウェア開発への活用

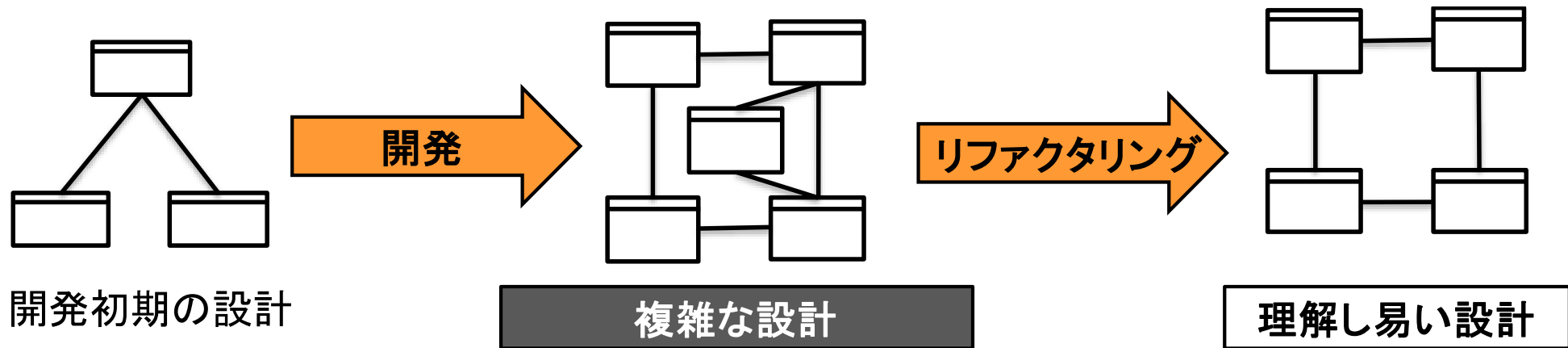
6



リファクタリング (1)

7

- ソースコードの品質を保って開発をするのは難しい
 - 品質：可読性，設計に対する変更の容易さ
 - 開発者はソースコードを深く理解した上で適切な変更を行う必要がある
 - 開発者は変更が容易な設計を保つ必要がある
- ソフトウェアの設計を改善するための技術として，**リファクタリング**が注目されている。



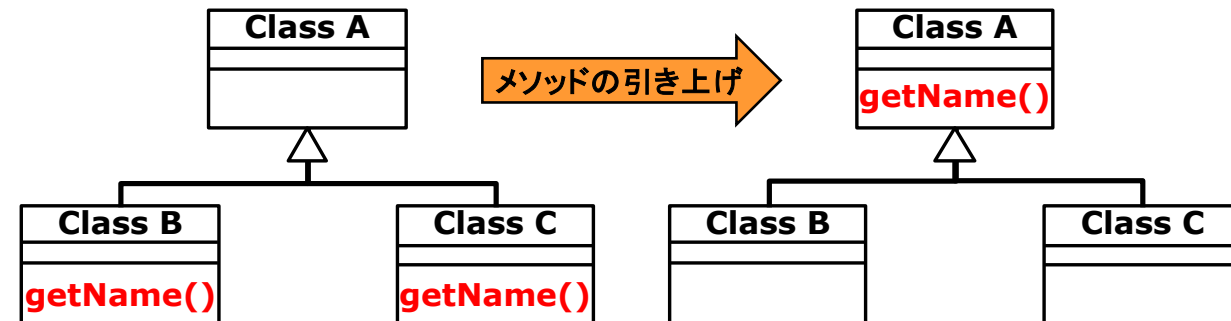
リファクタリング (2)

- ソフトウェアの外部的な振る舞いを変更することなく、内部的な構造を改善すること[1].



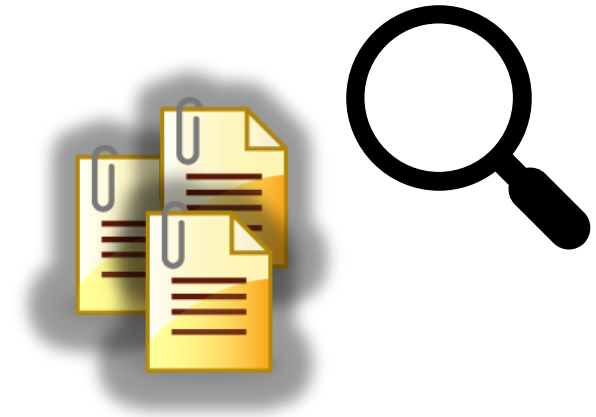
- リファクタリングの例:

- 変数名の変更, メソッド名の変更
- メソッドの引き上げ, メソッドの抽出・移動



リファクタリング推薦

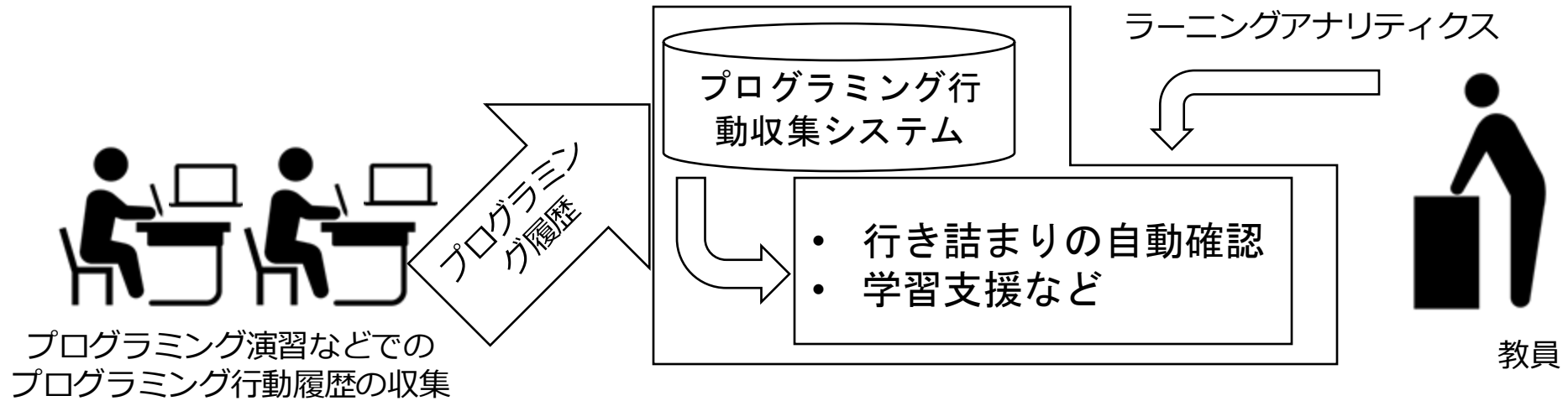
- 開発中のソースコードからリファクタリングすべき箇所を特定するのは開発者にとって負担となる
- 過去のリファクタリング履歴に基づき、リファクタリングすべき箇所を特定する
- (単純な) 機械学習を用いた手法については2014年時点で実施済み
 - 現在は深層学習を用いたリファクタリング推薦手法の開発を行っている



プログラミング教育に 関する研究

ソフトウェア工学×プログラミング教育

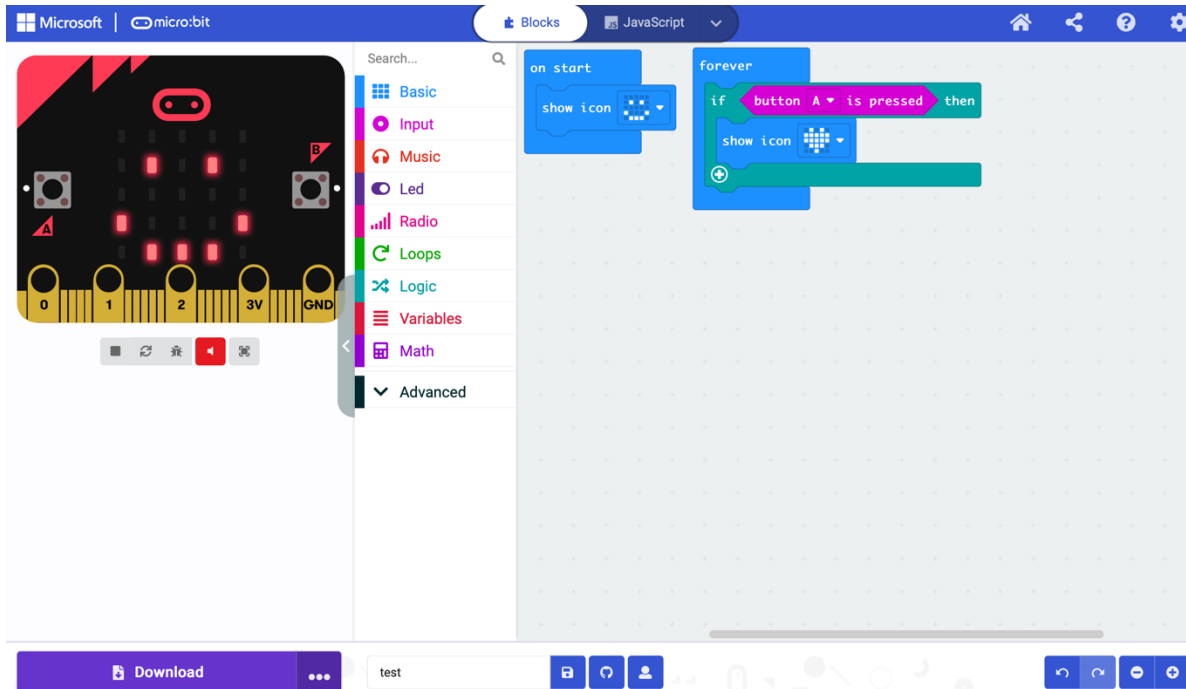
- ソフトウェア工学の知見を活かしたプログラミング教育に関する研究も行っています



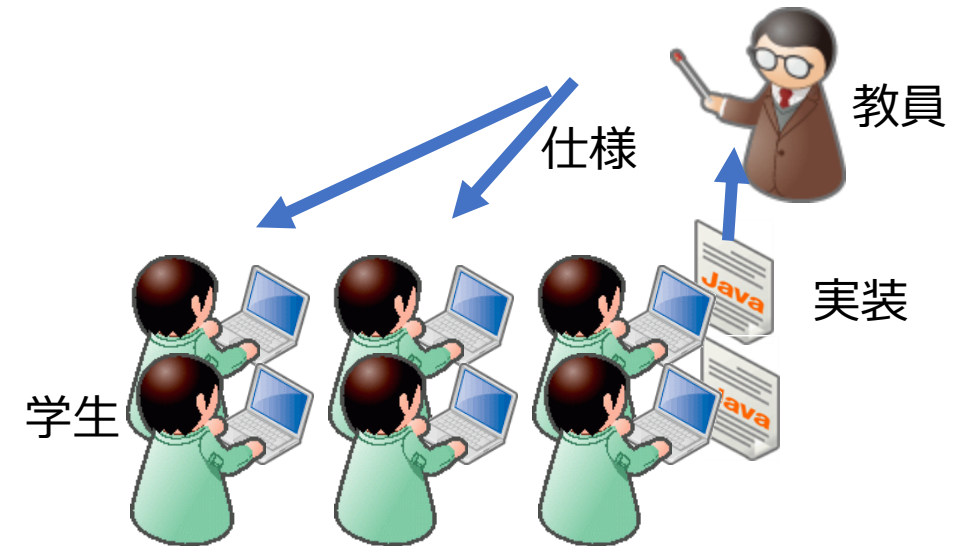
- どの辺が「ソフトウェア工学」を活かしているのか？
 - 学習者の行動履歴を分析
 - 具体的なソースコードなどを解析

プログラミング教育に関する研究

12



小・中学生を対象とした
ブロックプログラミングの教育支援
(画像はMicrosoft社の開発環境)



個別学習

教員・サポートスタッフ



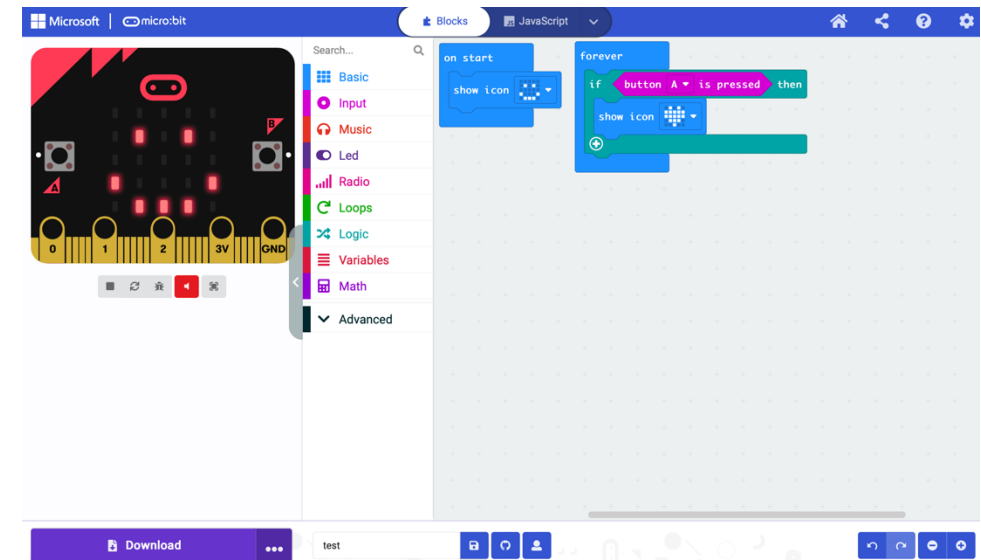
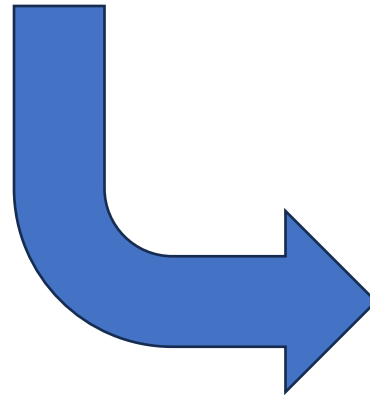
共同学習

大学生を対象とした
プログラミング教育支援

小・中学生を対象としたプログラミング教育支援

- 大学生のプログラミング演習におけるプログラミングの履歴を収集
- 履歴を分析することで学生の行き詰まりを特定する手法を提案
- 評価結果から行き詰まりを特定できることを確認

適用範囲を小・中学生に拡大



小・中学生を対象とした
ブロックプログラミングの教育支援
(画像はMicrosoft社の開発環境)

生成AIを活用したプログラミング教育支援

- 生成AIを活用した作問支援
- 生成AIを活用した仮想開発者を用いた共同開発の学習支援

大規模言語モデルを活用した学生の個別学習支援

