

特別研究報告

題目

プログラミング教育におけるテストケース通過状況比較システム

指導教員

楠本 真二 教授

報告者

長谷川 和輝

令和4年2月8日

大阪大学 基礎工学部 情報科学科

内容梗概

昨今のプログラミング教育では、教員に提示されたテストを達成することを目標に受講生が開発を行うというテストファーストな手法が採用されつつある。これにより、受講生にとって仕様に対する曖昧さが低減されるという利点がある。しかし、課題達成への作業の見積もりは容易ではなく、学生自身による進捗管理は困難である。この問題の解決策の一つとして他者との比較が挙げられるが、COVID-19の流行により机を並べての作業や会話といった対面での交流が減少し、他者の進捗状況の認知は困難になっている。そこで本研究では、Git上に記録された開発履歴を用いて、現在と過去の受講生の課題達成状況を可視化するシステムを提案する。各受講生はこのシステムを利用することで自分の達成状況を相対的に把握できる。適用実験では学部3年生を対象とした講義にこのシステムを導入し、自身の相対的な進捗状況の把握はできたか、また取り組むべき課題の優先順序付けはできたかについて確認した。

主な用語

プログラミング教育, CI ツール, 進捗, 可視化, 社会的比較

目次

1	はじめに	1
2	準備	3
2.1	テストファーストなプログラミング演習形態	3
2.2	COVID-19 の影響	3
2.3	TF 形態の問題	3
3	提案手法	5
3.1	seap の概要	5
3.2	seap の外観	5
3.3	進捗の回収	8
4	適用実験	10
4.1	実験の目的	10
4.2	題材講義	10
4.3	アンケートの設計	10
5	実験結果と考察	12
5.1	seap の使用状況	13
5.2	obj1 の達成可否と考察	13
5.3	obj2 の達成可否と考察	15
5.4	seap の良かった点, 改善点	15
6	おわりに	17
	謝辞	18
	参考文献	19

目次

1	現在の学生との比較画面	6
2	過去の学生との比較画面	7
3	進捗回収の流れ	9
4	seap の利用頻度	12
5	seap をいつ利用したか?	12
6	あなたの進捗は早かったか? 遅かったか?	13
7	他人との進捗差はどの程度だと思うか?	13
8	達成が容易なテストを把握できたか?	14
9	達成が困難なテストを把握できたか?	14

表目次

1 はじめに

効率的かつ効果的なプログラミング演習の実現を目的として、テストケースを活用した演習形態が広く採用されている [1] [2]. このテストファーストなプログラミング演習形態（以降、TF 形態）では、教員があらかじめ課題の仕様をテストとして規定し、受講生はそのテストの通過を目標としてプログラミング作業を進める. TF 形態は一般的な演習形態と比較して、仕様に対する曖昧さの低減、自動的な課題達成の可否判定、自動判定による自主的な開発の促進といった利点が存在する. さらに TF 演習の実現には、版管理システムや継続的インテグレーションツールなどのソフトウェア開発技術が活用される [3]. そのため実践的な開発技術を体験できる、という副次的な効果も期待できる [4].

TF 形態を含む一般的なプログラミング演習が抱える典型的な課題の一つとして、受講生自身による進捗管理の難しさが挙げられる [5]. 教育現場における開発作業は、基本的に受講生にとっては未経験の内容である. よって、与えられた課題の達成にの日数を要するのか、現在の作業速度で締め切り間に合うのか、優先的に実装すべき仕様やテストはどれか、といった作業の見積もりは容易ではない. また、言語処理や画像処理、アプリ開発等をテーマとする実践的なプログラミング演習では、まとまった規模の課題が一括して提示されることがある. この方法は、従来の受動的な教育形態と比べて、能動的かつ自主的な開発作業の促進が可能であり、事前にテストを配布する TF 形態との親和性も高い. 一方で、受講者による自主的な進捗管理の支援が重要な課題となる.

作業見積もりの難しさを緩和する一の方法は、他者との比較である. 自身の作業が他の大勢と比べて進んでいるのか・遅れているのかという相対的な情報は、進捗の是非を考える際の一つの基準となる. このような他者と自己との比較（社会的比較 [6] [7]）は人の取る自然な行動であり、進捗状況を含めた様々な領域の不確実さを低減させる効果がある [8]. しかしながら、COVID-19 の流行を契機とした社会のオンライン化により、社会的比較は容易ではなくなった. 対面でのコミュニケーションの機会が激減したため、その中で自然に得られていた他者の進捗状況という情報へのアクセスは困難になった. TF 形態での演習はオンライン化が容易である一方で、自主的な開発作業を促進しやすく、結果的に社会的比較の機会をより減少させる傾向にある. 効果的な TF 形態の実現のためには、社会的比較の仕組みが必要であると考えられる.

本研究の目的は、TF 形態でのプログラミング演習における進捗把握の支援である. そのために、他者との進捗状況の比較を実現するシステム seap を提案する. seap では、TF 形態で与えられる各テストケースの通過状況を、進捗状況を表す指標と見なす. 全受講生に対する各テストの達成割合をリアルタイムで可視化することで、相対的な進捗状況の把握を実現する. さらに seap では現在の受講生だけでなく、過去の受講生との比較も可能である. これにより、各テストの達成にの期間を要するのか、過去の受講生がどのような順序でテストを達成させたか最も達成が困難なテストがどれなのか、といった

類推が可能となる。適用実験として、学部3回生を対象としたプログラミング演習に seap を導入し、その効果を確認した。実験の結果、自身の相対的な進捗状況の把握には寄与できたが、取り組むべき課題の優先順序付けにはあまり寄与できなかった。

2 準備

2.1 テストファーストなプログラミング演習形態

昨今、ソフトウェア開発の実践的な技術を取り入れた、演習形式のプログラミング教育形態が広まっている [1] [2]. 本論文では、この教育形態の一つであるテストファーストなプログラミング演習形態に着目する. テストファーストなプログラミング演習形態（以降、TF 形態）では教員が事前に提示したテストに通過したか否かを課題達成の可否と見なす. 従来のプログラミング演習では課題達成の可否の判定に一定の時間と労力が必要であり、受講生は自身の提出物が達成要件を満たしているか否かを即座に把握することができなかった. それに対し、TF 形態では、プログラムを提出した時点で即座に課題達成の可否がわかる [9] [10]. そのため、受講生は自発的に次の課題へ取り組むことができる. また、課題達成状況が自動的に把握できるので、教員は受講生全体の進捗状況を俯瞰して把握しやすく、進捗が遅れている受講生に対するサポートが可能になる. TF 形態の実現には、Git や CI ツールなど様々な開発支援技術が使われる.

TF 形態の具体的な実施の流れを説明する. まず教員がその演習における課題、すなわち作成プログラムの要件を規定する. 教員はその要件の達成可否を判定するテストスイートを作成し受講生に配布する. 以降、テストスイートを単にテストと呼ぶ. 受講生はプログラムを作成し、配布されたテストを用いて自身が作成したプログラムが要件を満たすか確認する. プログラムが配布された全テストに通過すれば課題達成となる.

2.2 COVID-19 の影響

2019 年から流行した COVID-19 の対策として、教育現場では非対面形式での講義が実施されてきた [11]. テストファーストな教育手法はオンライン形態のツールを使用するため、対面作業への依存を軽減できる [12]. そのため、非対面形式での講義との親和性が高く、プログラミング演習において広く採用されつつある.

しかし、非対面形式での講義が増加するにつれ、机を並べての作業や受講生間での会話の機会は減少した. そのため、受講生は自身の進捗が他の受講生と比較して進んでいるのか、遅れているのかを把握することが難しくなると考えられる. また、提示された締め切りに対してどのようなペースで課題に取り組むべきかについて友人と相談する機会も減少した [13] [14].

2.3 TF 形態の問題

本研究では TF 形態が抱える問題のうち、受講生間の交流が減少したことに起因する以下 3 つの問題を考える. これらはコロナ禍における受講生間の交流の減少で顕在化した問題であるが、TF 形態自体

が抱える問題であるといえる。

P1： 他者と比較したときの自身の進捗状況の把握

人は正確な自己評価を得ようとする際、他者との相対的な比較を基準とするという社会的比較論が知られている [6] [15] [16]。そのため、受講生が自身の進捗が進んでいるのか、遅れているのかを把握する際に、他の受講生との進捗度合いの比較は役立つ。しかし、コロナ禍において机を並べての作業や会話といった対面での交流機会の減少により他の受講生の進捗を把握しづらくなった。その結果、自分の進捗が他の受講生と比較して進んでいるのか、遅れているのかがわからず、不安になる可能性がある。また、他の受講生の進捗が把握しづらいことにより、自分の進捗が遅れているとは思わない受講生が増え、最終的な課題達成者が減少する可能性があるとも指摘されている [17]。コロナ禍が収束した後に広まることが見込まれるハイフレックス型講義*¹においても、従来の対面形式の講義に比べて他の受講生の進捗を把握しづらくなり、コロナ禍における講義と同様の状況になると想定される。

P2： 優先的に取り組むべきテストの把握

多くの場合、教員は難易度に応じた順序でテストを受講生に提示するが、演習の形態によっては難易度とは無関係の順序になっていることもある。特に言語初学者ではなく中級者以上を対象とした、発展的な内容（データ構造とアルゴリズムや、言語処理など）では大きなテスト群が一度に与えられることもある。また、テストの難易度の捉え方は教員によって様々であるため、実際に受講生が取り組むと後半のテストの方が簡単な場合もある。

*¹ ハイフレックス (HyFlex) 型講義とは、Hybrid-Flexible の略で、対面・同期オンライン・非同期オンラインが提供され、学生が自在に選択することができる講義形態を指す。

3 提案手法

3.1 seap の概要

TF 形態における問題の解決を目的として、他者との進捗比較を可能とするシステム seap を提案する。seap では、自身と現在の受講生との比較だけではなく、過去の受講生との比較も行えるようにする。また、seap における進捗はテストの通過状況と見なす。ソースコードの行数を進捗とする方法も可能だが、ソースコードの長さは設計方法や採用するアルゴリズムなどに依存するため、曖昧である。その点、テストは教員から提示されるため、受講生間で差異はなく、明確である。

まず、P1 の解決を目的として、自身と現在の受講生との比較を実現する。具体的には自身のテスト通過状況と全体に対するテスト通過者の割合を比較する。以降では、全体に対するテスト通過者の割合を達成率と呼ぶ。自身のテスト通過状況と達成率の比較は社会比較論 [6] [15] [16] における正確な自己評価を得ようとする際の自身と他者の相対的比較に相当する。そのため、自身の進捗が進んでいるのか、遅れているのか把握する上でこの比較は役に立つと考えた。

さらに自身と過去の受講生との比較により、自身の進捗と大半の受講生の進捗の差を類推できる。例えば、自身の通過したテストについて、過去の受講生の達成率が自分の達成した 3 日後に 50% になったとする。このことから、自身の通過したテストについては 50% の受講生に比べて 3 日分進捗が進んでいると類推できる。また、自身が未通過のテストについて、過去の受講生の達成率が 7 日前に 50% になっていたとすれば、50% の受講生に比べて 7 日分進捗が遅れていると類推できる。

次に、P2 の解決を目的として、達成率を各テストごとに表示する。各テストについて現在の受講生の達成率や過去の受講生の達成率遷移を見ることで、通過が困難なテストや容易なテストがわかる。例えば、過去において講義開始からすぐに達成率が高くなるテストは通過が容易であり、締め切り直前まで達成率が低いテストは通過が困難だと考えられる。また、自分が未通過のテストの達成率を見て、多くの受講生が通過しているテストがあれば、教員だけではなく他の受講生にもそのテストを通過させるためのアドバイスを求めることができる。このようにして参考となる情報を提示することで、取り組むテストの優先度を定める補助ができると考えた。

なお、seap を web 上に公開することで、コロナ禍におけるの非対面の講義形式であっても、seap を使用できる。

3.2 seap の外観

seap の外観を図 1, 2 に示す。seap は以下 4 つの要素で構成されている。

- 比較対象年度, 課題選択一覧

比較対象年度： 2021 2020 2019 2018

課題： Lexer Parser Checker Compiler

学籍番号：

比較対象の日付：

s4.compilerの締め切りまであと”20日と22時間”

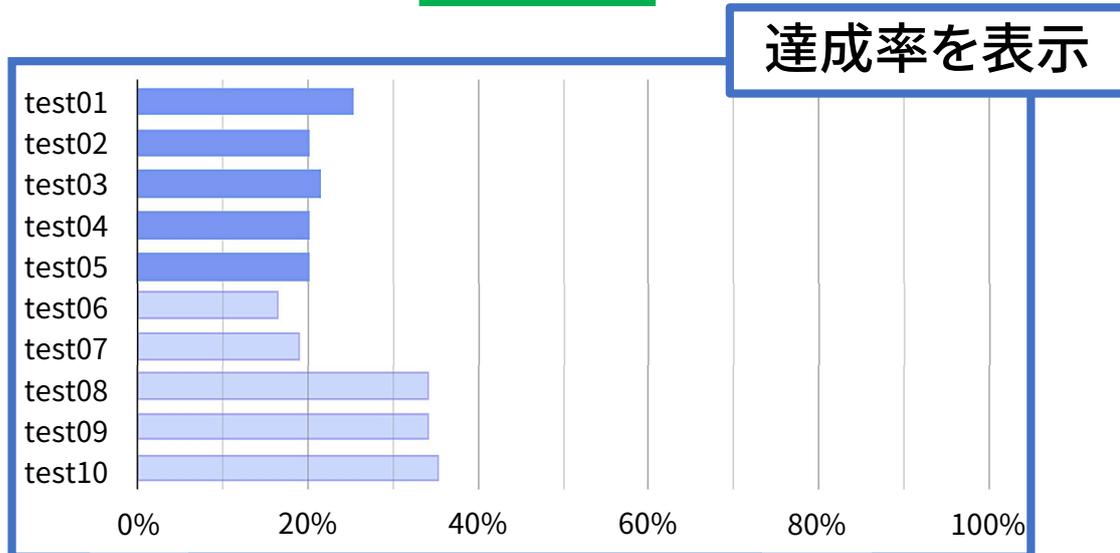


図1 現在の学生との比較画面

- 自身の ID 入力フォーム
- 比較対象の日付選択スライダー
- 各テストの達成率を示す棒グラフ

まず、比較対象年度を選択することで、現在の受講生か過去の受講生との比較のどちらを行うか選択できる。その下にある課題選択一覧で、確認したい課題を選択できる。なお、図1、2では、節にある適用実験の題材となった講義の課題が表示されている。下部にある棒グラフには各テストの達成率が表示される。中央部にあるタイムスライダーで締め切りまでの残り時間を設定することで、設定された時点での各テストの達成率が画面に表示される。また、学籍番号と記載されている欄に自身の ID を入力すると、自身の達成したテストが濃い青で色付けされる。これにより、自身のテスト通過状況と達成率の比較を可能にする。

比較対象年度： 2021 2020 2019 2018

課題： Lexer Parser Checker Compiler

学籍番号：

比較対象の日付：

s4.compilerの締め切りまであと”11日と20時間”

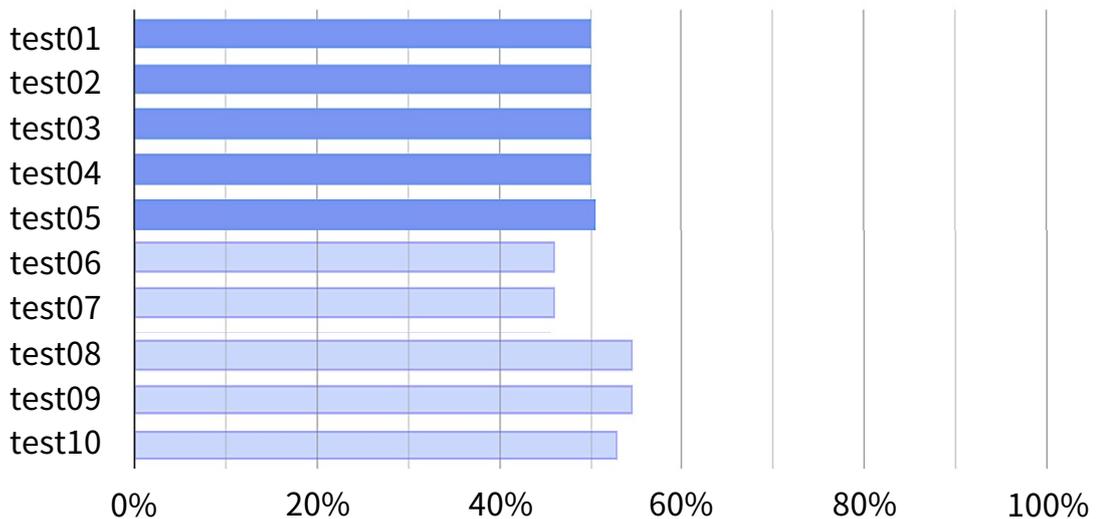


図2 過去の学生との比較画面

図1は2021年を選択しているため、現在の受講生と比較している画面である。下部の棒グラフには、締め切り20日と22時間前の各テストの達成率が表示されている。この受講生のIDは09B99008であり（以降、この受講生を09B99008と呼ぶ）、締め切り20日と22時間前の時点でtest01～test05までのテストは通過しているが、test06～test10は通過していない。未通過のテストの達成率を見るとtest06、test07に比べ、test08～test10の達成率が高いため、次に取り組むべきはtest08～test10であると考えられる。また、既に通過しているtest01～test05に比べ、test08～test10は達成率が高いため、テスト通過までにtest01～test05の時ほど時間と労力がかからないと目算できる。

また、図2は2020年を選択しているため、過去の受講生と比較している画面である。下部の棒グラフには、2020年における締め切り11日と20時間前の各テストの達成率が表示されている。09B99008

が既に通過している test01～test05 のテストは、2020 年において締め切り 11 日と 20 時間前になると約 50% の受講生が通過しているとわかる。このことから、図 1, 2 で指定している時刻の差分である 9 日と 2 時間分は約半数の受講生よりも進捗が進んでいると考えられる。また、09B99008 が通過していない test06～test10 のテストは、2020 年において締め切り 11 日と 20 時間前になると約 50% の受講生が通過している。このため、図 1, 2 で指定している時刻の差分である 9 日と 2 時間分は約半数の受講生よりも進捗が遅れていると考えられる。

3.3 進捗の回収

Git 上に記録された開発履歴を利用して進捗（テストの通過状況）を回収する。進捗回収の流れを図 3 に示す。受講生の各コミットごとにおいて、git-checkout し、テストを実行する。その後、テストの実行結果を図 3 の下にあるように、JSON 形式で集計する。この処理を過去の受講生全員分に対して行う。

現在の受講生についてはリアルタイムに進捗を取得するため、webhook を活用する。受講生が自身のリモートリポジトリへ git-push した時点で seap へ通知するように webhook を設定する。通知を受けると seap は対象の受講生のリモートリポジトリをローカルへ git-clone する。その後、最新コミットで git-checkout しテストを実行する。実行結果は過去の受講生の時と同様の JSON 形式で、集計済みの実行結果に追記していく。

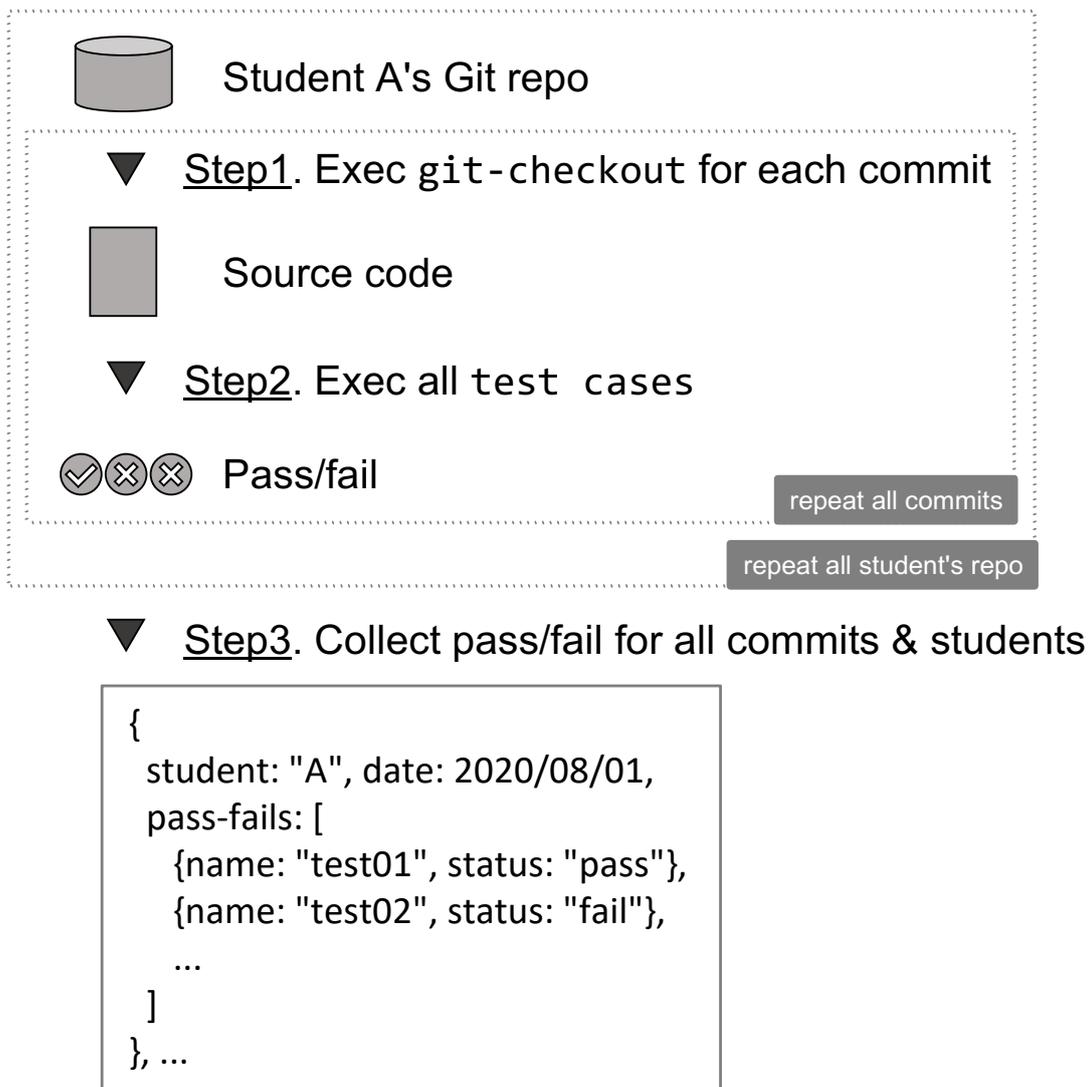


図3 進捗回収の流れ

4 適用実験

4.1 実験の目的

本研究では, seap を TF 形態に導入することで 2.3 節にて列挙した課題を解決できたかを確認するために適用実験を行った. 適用実験を行うにあたり, 以下の項目を目的として設定した.

obj1: 自身の相対的な進捗状況の把握

obj2: 取り組む課題の優先順序付け

なお, obj1 については自身と他の受講生との相対的な進捗差を具体的な日数や時間で把握できたかについても調査した.

4.2 題材講義

seap の適用実験として, 大阪大学基礎工学部情報科学科の学部 3 年生を対象として開講されている情報科学演習 D*² (以降, 演習 D) に seap を導入した. 受講生には字句解析器の作成, 構文解析器の作成, 意味解析器の作成, コンパイラの作成の 4 つの課題が順に課される. 本論文では 3 つ目と 4 つ目の課題への導入の結果を述べる. 1 つ目の課題から導入する方法も考えられるが, seap 導入前後での比較を可能とするために, 3 つ目の課題より導入した. 各課題におけるテストの数は 1 つ目の課題のみ 38 ケース, それ以外の課題については 37 ケースである. なお, 今年度の演習 D はハイフレックス型講義であった.

受講生が seap を使用するか任意とし, 希望者は 3.3 節にあるデータ収集対象から外すようにした. そのため, 演習 D の受講生のうち seap の使用を希望する人に対して研究目的によるデータ収集の同意を得た. 結果として, 全受講生 78 名中 77 名から同意を得られた.

なお, 3 つ目の課題の締切後から 4 つ目の課題の締切後にかけて受講生に対してアンケートを実施する. アンケートでは 4.1 節で列挙した目的に加えて, seap の利用実態と要望についても調査する.

4.3 アンケートの設計

obj1 については, 自身の進捗について他の受講生に比べて進んでいると思ったか, 遅れていると思ったか, わからなかったかという選択肢を設けて質問することで確認する. その後, 自身の進捗が進んでいると回答した人, 遅れていると回答した人のそれぞれに具体的にどの程度進んでいると思ったか, 遅れていると思ったかについて質問する.

*² <https://loki.ics.es.osaka-u.ac.jp/>

また、取り組むテストの優先順序付けには、過去の受講生の達成率の推移が参考になると考えられる。そのため、obj2については過去の受講生や現在の受講生の達成率を見ることによって、達成が容易なテストや達成が困難なテストが把握できたかを質問することで確認する。

なお、これらに加えて seap の利用実態と要望の調査のため、利用頻度とそのタイミング、使用した感想や改善点についても質問する。

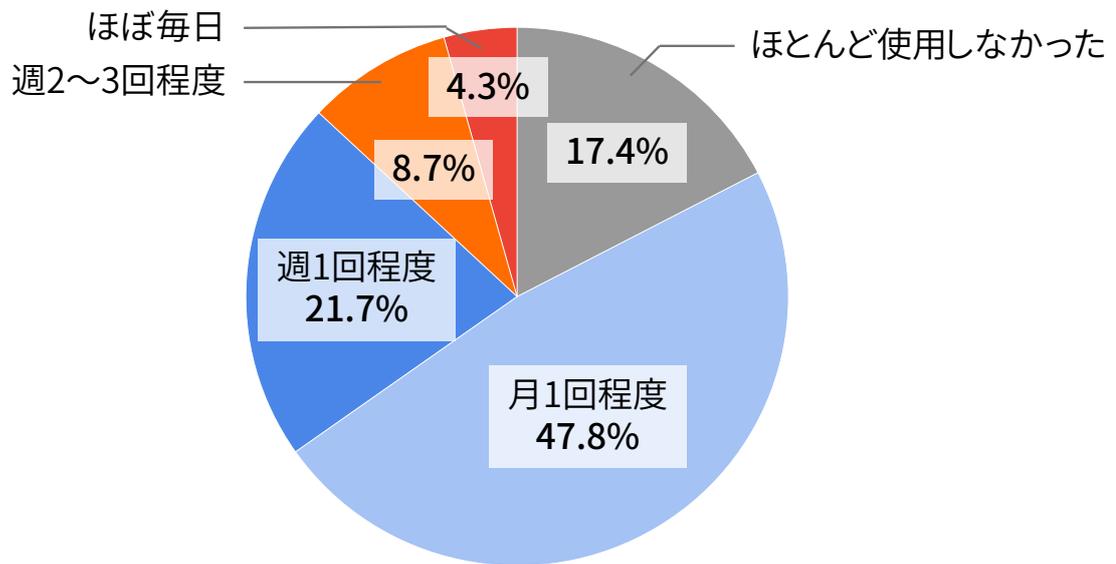


図4 seap の利用頻度

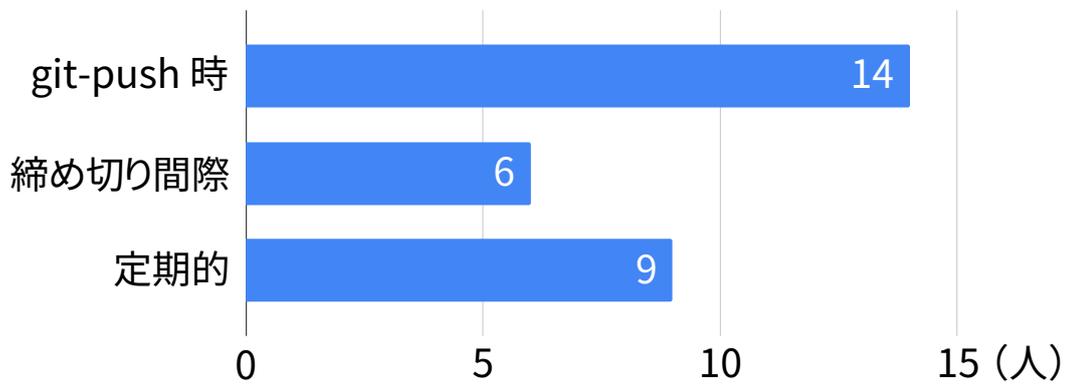


図5 seap をいつ利用したか？

5 実験結果と考察

演習 D の受講生に対してアンケートを行った結果、77 名中 23 名から回答が得られた。

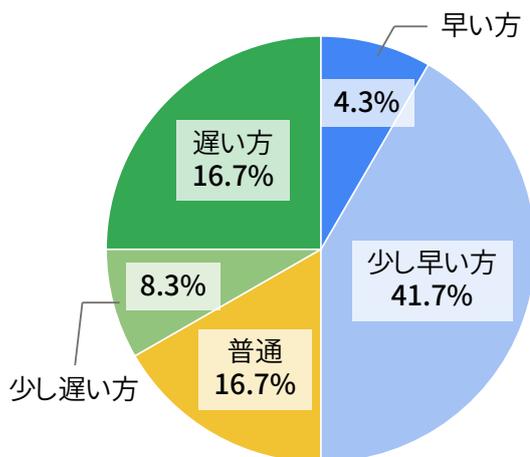


図6 あなたの進捗は早かったか？遅かったか？

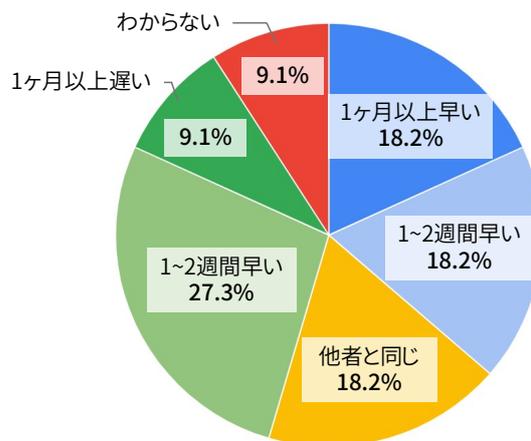


図7 他人との進捗差はどの程度だと思うか？

5.1 seap の使用状況

seap の利用頻度についてのアンケート結果を図4に示す。図4より、回答者の約34%が週1回以上利用し、回答者の約47%が月1回程度の利用をしていたことがわかった。また、seap をいつ利用したかのアンケート結果を図5に示す。図5より、回答者23名中14名がgit-push 実行時にseap を利用していることがわかる。

seap にはgit-push された時点でのプログラムのテスト通過状況が反映されるため、図5のような結果になったと考えられる。また、今年度の受講生のコミット数は平均14.5回であり、演習Dの開講期間は約6ヶ月であることや、git-push の実行回数はコミット数よりも少なくなることを考えると、図5のような結果になったのは、git-push の実行回数が約月1回であったためと考えられる。このことからseap の利用頻度を上げることができれば、git-push の回数が増え、コミット粒度が細かくなることが期待される。

5.2 obj1 の達成可否と考察

アンケートの結果を図6に示す。自身の相対的な進捗が把握できず、わからないと回答した受講生はいなかった。以降、5.2節では相対的な進捗を単に進捗と呼ぶ。この結果からseap は受講生の進捗把握に寄与できたと考えられる。

また、自身の進捗を把握できた理由について質問したところ、以下のような回答があった。

seap 上の自分が達成していない課題の提出状況が高いことが多かったため。

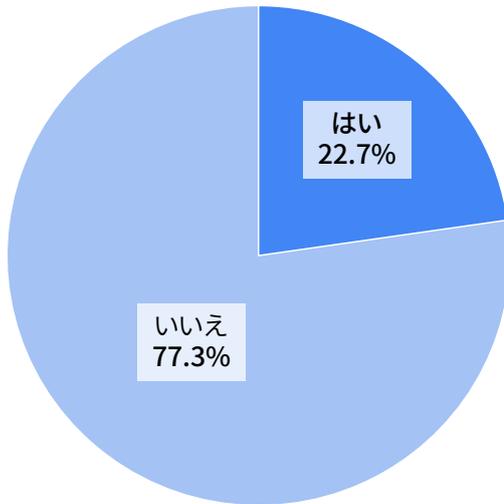


図8 達成が容易なテストを把握できたか？

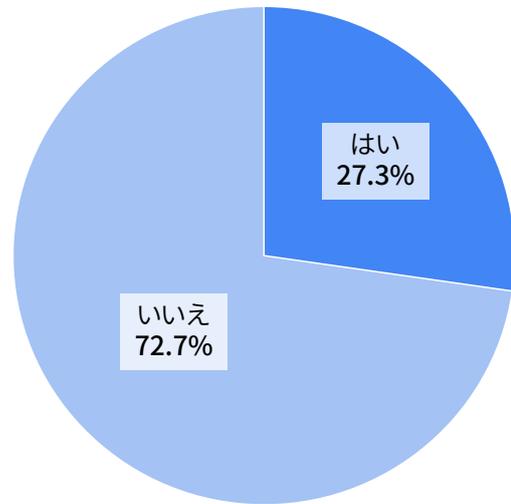


図9 達成が困難なテストを把握できたか？

4つ目の課題の全テスト通過段階で seap を見たとき、全体の達成率は2割ほどであったため。

全体進捗率が低い中で実装を完了できたため。

これらの回答から、進捗の把握をするタイミングについて、以下の2通りがあることがわかった。

1. 課題に取り組んでいる最中
2. 課題達成時

比較的進捗が遅れていると回答した受講生からは1の様な回答が多く、進捗が進んでいると回答した受講生からは2の様な回答が多かった。進捗が遅れていると感じた受講生は、遅れているという不安から、課題に取り組んでいる最中に進捗の把握を試みると考えられる。逆に、進捗が進んでいると感じた受講生は、課題に取り組んでいる最中は不安を感じないため、進捗の把握を試みず、達成時に自身の進捗に対して評価を下そうと進捗の把握を試みると考えられる。

また、自身の進捗を把握できた理由について以下のような回答もあった。

締め切り当日に課題を終えたため。

締め切りの1週間前くらいに課題を達成したため。

このように、比較的進捗が進んでいると回答した受講生の中には、自身が課題を達成した日と締め切り日との差を見て自身の進捗を把握した者もいた。

なお、他者と自身の進捗差はどの程度だと思いかという質問に対する回答を図7に示す。図7から90%の回答者が他者と自身の進捗差はどの程度か把握できていた。どのように把握したかについて質問したところ、以下のような回答が得られた。

過去の達成率を見て、自分が達成した1週間後くらいに約50%になったため。

締め切り3ヶ月前ほどに課題達成したため。

前者のように、seapを利用して具体的な差の把握をした回答者もいたが、後者のように、自身が課題を達成した日と締め切り日との差を見て具体的な差の把握をした回答者も多かった。

以上より、seapには自身が課題を達成した日と締め切り日との差を見るよりもわかりやすく自身の進捗を把握する機能が求められる。

5.3 obj2の達成可否と考察

アンケートの結果を図8, 9に示す。達成が容易なテストを把握できたと回答した受講生は22%程度に止まった。また、達成が困難なテストを把握できたと回答した受講生は27%程度に止まった。

達成が容易なテスト、達成が困難なテストを把握するには、タイムスライダーを動かし、達成率の変化を見る必要がある。しかし、seapでは変化の様子を1つの画面で俯瞰して見ることができない。そのため、受講生にはタイムスライダーを動かすという手間をかけてしまう。結果として、達成が容易なテスト、達成が困難なテストを把握する方法に気がつくにくくなったと考えられる。

また、演習Dは2017年よりTF形態を採用しており、毎年、受講生からの指摘を受け、テストの並び順や品質の改善をしてきている。そのため、既に演習Dのテストが受講生にとって適した難易度順に並んでいて、取り組むテストの優先順序を自ら付ける必要がなかった可能性もある。

以上より、seapの改善として、達成率の変化を1つの画面で俯瞰できるようにすることが考えられる。また、適用実験の改善として、敢えてテストを無作為に並べ、受講生に取り組むテストの優先順序を考えることの重要性を感じてもらえるようにすることが考えられる。

5.4 seapの良かった点、改善点

seapの良かった点や改善点について、記述式のアンケートを行った。良かった点については以下のような意見が得られた。

達成率が低いときに課題を終えることができ、優越感に浸れた。

競争心によって課題へのモチベーションが上がった。

自分に発破をかけることが出来て課題を早く終わらせることができた。コロナ禍には必要なツールだと思った。

このように、seapによって進捗の早い受講生にとっては自身の進捗状況について前向きな気持ちをもたらしたり、競争心や切迫感により受講生が課題に取り組むきっかけを生み出したりという副次的な効果が得られた。

また、改善点としては以下のような意見が得られた。

一度課題を達成すると、タイムスライダを戻しても“達成”扱いのままになり、“いつ達成したか”を振り返ることができない。

自分の進捗状況に対する定性的フィードバック機能があると嬉しい。現在の自分の進捗が過去や現在の平均値と比べてどうだ、というのを返してくれるとユーザビリティは高まる。

全テストを通過している人がどの程度いるのかも知りたかった。

このように、seapにはシステムとしての改善点が多く残されている。

6 おわりに

本研究では、自身の進捗を現在の受講生や過去の受講生と進捗を比較するシステム seap を提案した。適用実験として、学部 3 年生を対象とした講義にこのシステムを導入し、取り組む課題の優先順序付けにはあまり効果が見られなかったが、自身の相対的な進捗の把握において効果があることを確認した。

本稿ではテストの通過状況を進捗と見做して適用実験を行った。しかし、テストの通過状況以外にも、コードメトリクスの計測や、レポートにおいて記述した文字数の計測なども、課題における進捗を考える上で役立つ。そのため、上記の進捗に対してシステムを拡張することや、受講生にアンケートを行って更なるデータを収集すること、seap 導入によるコミット粒度の変化の分析、演習 D 以外の講義への導入を見据えたシステムの拡張性を高めることなどが挙げられる。

謝辞

本論文執筆にあたり、添削指導をしていただいた柁本先生，研究内容に対する的確な指摘をしてくださった肥後先生，楠本先生，並びに執筆において様々なアドバイスをしてくださった同研究室の学生や先輩方，この場をお借りして感謝申し上げます。

私は本研究において，演習 D の受講生に社会的比較を行ってもらうように適用実験を設計しました。演習 D の受講生に社会的比較をさせるからには，自分もやらなければならないという使命感の元，私自身も同研究室の学生や先輩，並びに他研究室の学生と社会的比較を行いながら本論文を執筆しました。その結果，自分独りでは到底生み出せなかったであろう進捗を生むことができました。改めて，ありがとうございました。

また，実験に協力していただいた上に，卒業研究の応援までしてくださった演習 D 受講生の皆様。皆様の応援が大きな精神的支えとなり，本論文を完成させることができました。本当にありがとうございました。

1 年間という短い間でしたが，研究というものは独りでするのではなく，あらゆる人との関わりのもと成り立つものだと思えて実感できたかけがえのない時間でした。最後に，本論文をここまでご精読いただきました読者の皆様に多大なる感謝を申し上げます。

参考文献

- [1] Christopher Douce, David Livingstone, and James Orwell. Automatic test-based assessment of programming: A review. *Journal on Educational Resources in Computing*, Vol. 5, No. 3, pp. 4–es, 2005.
- [2] David Janzen and Hossein Saiedian. Test-driven learning in early programming courses. *The ACM Special Interest Group on Computer Science Education Bull.*, Vol. 40, No. 1, pp. 532–536, 2008.
- [3] Jon Bowyer and Janet Hughes. Assessing undergraduate experience of continuous integration and test-driven development. In *Proc. International Conference on Software Engineering*, pp. 691–694, 2006.
- [4] Florian Huber and Georg Hagel. The application of continuous practices in higher computer science education - a systematic literature review. In *Proc. International Convention on Information, Communication and Electronic Technology*, pp. 1618–1623, 2021.
- [5] Rodrigo Pessoa Medeiros, Geber Lisboa Ramalho, and Taciana Pontual Falcão. A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, Vol. 62, No. 2, pp. 77–90, 2019.
- [6] Leon Festinger. A theory of social comparison processes. *Human Relations*, Vol. 7, pp. 117–140, 1954.
- [7] J SULS. From the cradle to the grave : Comparison and self-evaluation across the life-span. *Psychological Perspectives on the Self*, Vol. 1, pp. 97–125, 1982.
- [8] Mark D. Alicke. *Evaluating Social Comparison Targets*, pp. 271–293. Springer US, 2000.
- [9] A. Kyrilov and D. C. Noelle. Do students need detailed feedback on programming exercises and can automated assessment systems provide it? *Journal of Computing Sciences in Colleges*, Vol. 31, No. 4, pp. 115–121, 2016.
- [10] J. Carter, J. English, K. Ala-Mutka, M. Dick, W. Fone, U. Fuller, and J. Sheard. How shall we assess this? In *Proc. Working group reports from ITiCSE on innovation and technology in computer science education*, Vol. 35, pp. 107–123, 2003.
- [11] Pinaki Chakraborty, Prabhat Mittal, Manu Sheel Gupta, Savita Yadav, and Anshika Arora. Opinion of students on online education during the COVID -19 pandemic. *Human Behavior and Emerging Technologies*, Vol. 3, No. 3, pp. 357–365, 2020.
- [12] Enrique Barra, Sonsoles López-Pernas, Álvaro Alonso, Juan Fernando Sánchez-Rada, Aldo

- Gordillo, and Juan Quemada. Automated assessment in programming courses: A case study during the COVID-19 era. *Sustainability*, Vol. 12, No. 18, p. 7451, 2020.
- [13] Muhammad Adnan and Kainat Anwar. Online learning amid the covid-19 pandemic: Students' perspectives online learning amid the covid-19 pandemic: Students' perspectives. *Journal of Pedagogical Sociology and Psychology*, Vol. 2, pp. 45–51, 2020.
- [14] Christopher Irwin and Zane Berge. Socialization in the online classroom. *E-Journal of Instructional Science and Technology*, Vol. 9, No. 1, pp. 1–7, 2006.
- [15] 大久保暢俊. 社会的比較による自己評価と対人関係. 東洋大学人間科学総合研究所紀要, 第 10 巻, pp. 111–121, 2009.
- [16] Paul Gilbert, John Price, and Steven Allan. Social comparison, social attractiveness and evolution: How might they be related? *New Ideas in Psychology*, Vol. 13, pp. 149–165, 1995.
- [17] Xu Yang, Deli Li, Xiaoqiang Liu, and Jianguo Tan. Learner behaviors in synchronous online prosthodontic education during the 2020 covid-19 pandemic. *The Journal of Prosthetic Dentistry*, Vol. 126, pp. 653–657, 2021.