

オープンソース開発プロセスの要素を導入したサーバ構築演習の授業設計と試験的評価

Instructional Design and Evaluation of Server Exercise Class with Elements of Open Source Development Process

大森 裕介
Yusuke Ohmori

深町 賢一
Ken' ichi Fukamachi

公立千歳科学技術大学
Chitose Institute of Science and Technology

1. まえがき

オープンソースソフトウェア(OSS)開発プロセスの要素を実践的なサーバ構築演習授業に導入し、社会で求められるIT 技術者の育成を目的とする学習モデルの提案を行う。本研究で提案する学習モデルを演習授業に実践することで、本学習モデルの利点や欠点、また運用上の困難さを調査し、本学習モデルの試験的評価を示す。また、実践的な演習により学生の主体的な学習に与えた効果を評価する。

2. 背景

日本で行なわれている主体的な学びや教育機関を離れた後に求められる学び直し、日本で行われている高等学校の情報教育、大学での実践的な情報教育について述べる。

2.1. 主体的な学習

近年、アクティブラーニングなどを用いて、批判的な思考や学びかたの学習、異質な集団での問題解決能力の習得に取り組んでいる[1]。一方で、図1に示したように、アクティブラーニングでは学生が安易な回答を求めることや派生知識に無関心、表面的な議論を行うこと、認知プロセスの外化(問題解決のために知識を使う、人に話す、書く、発表すること)を行わないことが課題である。

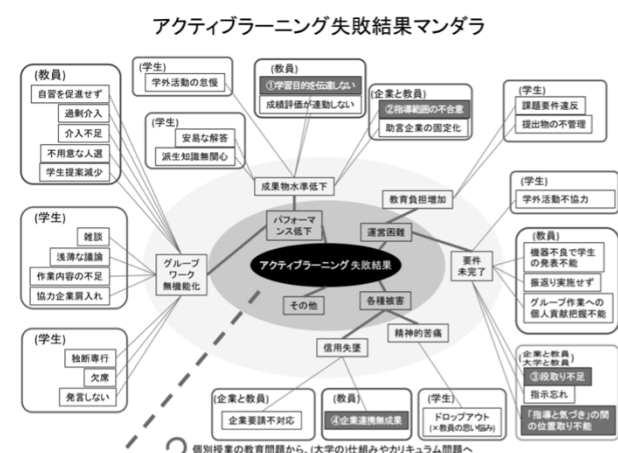


図1: アクティブラーニング失敗結果マンダラ(「アクティブラーニング失敗事例ハンドブック～産業界ニーズ事業・成果報告～」(中部地域大学グループ・東海 A チーム、2014 年(P3)) [2]より引用)

2.2. 生涯学習

学校教育を離れた人でも学び直しを行うことが求められている。経済産業省によると、リカレント教育において、学問追求と実践的教育のバランスに留意しつつ、実践的な職業教育の拡充を図る必要があると述べられている[3]。

主体的かつ意欲的に知識を身につけ社会に貢献することが社会人に求められている。

2.3. オープンソースソフトウェア開発

世界では、多くの技術者が意欲的にソフトウェア開発を行い社会に貢献する活動を行っている。その中でも、オープンソースソフトウェア(OSS)はソースコードが公開されており、自由な利用、再配布を認めているソフトウェアがオープンソースソフトウェアと定義されている。オープンソースソフトウェアの開発の多くは、無報酬で自発的に参加する開発者を中心として行われている。

OSS 開発では、場所や時間にとらわれず、多様な考えや思考を行う人が協働しソフトウェアを開発している。そのため、テキストを主に用いてコミュニケーションを行っている。OSS 開発に貢献する開発者は、知識を OSS 開発から習得することや、OSS 活動に貢献するため、自身の知識を OSS 開発に提供する。しかし、提供される情報はさまざまで、情報の妥当性を多様な意見を交わしながら議論し、評価しながら開発を行う。

情報サービス産業協会によると、多くの有能な技術者で形成されたコミュニティにおいて開発に参加する技術者には世界レベルで自身の能力を高められるばかりでなく、多くの人的ネットワークを得ることに繋がり、多くの先進的な情報を得る機会が増えることになると述べられている[4]。

2.4. 情報教育

日本の教育では積極的に IT 技術者の育成を行っている。大学教育では、実践的な技術者育成も重要視されており、その技術者像や技術者教育の到達目標や評価方法の提言もされている。

近年、基本情報技術者試験の資格取得を単位認定や入試優遇に用いている大学や授業カリキュラムを情報処理技術者試験の試験要綱を参考に策定している大学がある。大学教育のカリキュラムで情報処理技術者試験の試験要綱の知識を習得することが学生に求められている。

情報処理技術者試験とは、情報処理の促進に関する法律に基づき経済産業省が、情報処理技術者としての知識・技能が一定以上の水準であることを認定している国家試験である。情報処理技術者試験の試験要綱にはプログラミング

やネットワーク, OSS などについて出題されると示されている [5].

3. 目的

近年, 日本 の教育では, 積極的に IT 技術者の育成を行っている. 大学教育では, 実践的な技術者育成も重要視されており, その技術者像や技術者教育の到達目標や評価方法の提言もされている. 我々は, 発展的な知識や実践的な学習だけでなく, 社会で求められる IT 人材の開発姿勢や知識習得, 実践的な能力を学ばせることも重要と考えている. 姿勢や文化を学ぶ要素を授業設計に取り入れた場合の利点や欠点, また運用上の困難さについては明らかにされていない.

本研究では, オペレーティングシステムの仕組みについて学ぶ授業に実践的な演習とアクティブラーニングに加え, OSS 開発プロセスの要素を導入し, その有用性を評価する. OSS 開発プロセスとは, OSS 開発を行う過程である.

学生同士が認知しにくい状態にし, 異質な集団での交流する能力, 批判的思考の向上を目指す. 演習授業を行い, 実践的な学習を行う. 実践的な演習の中で OSS 開発プロセスを実践し, 教育機関を離れた後も主体的に学習できる IT 技術者を目指す学生に対する教育モデルの提案を行う.

4. 提案する学習モデル

OSS 開発プロセスの要素を導入した学習モデルを提案する. 本研究では, 実践的な学習を行いながら, 社会で求められる IT 人材を意識させるため IT 業界やオープンソースソフトウェア開発活動で用いられている GitHub を利用し, 次の学習姿勢の習得を促す.

- (a) 時間や場所にとらわれない情報共有
- (b) 問題の発見から解決までの履歴を残す
- (c) 問題解決までの学習経路の把握
- (d) 異質な集団に近い状態でのコミュニケーション
- (e) 認知プロセスの外化(外部に発表)

図 2 に学生間で情報共有し, 整理された情報から学習経路を把握する様子を示す. GitHub を利用したコミュニケーションは必然的にテキストベースになる. GitHub を用いているため, 時間や場所に関わらずコミュニケーションを行うことができる. テキストコミュニケーションでは情報や知識をまとめることが求められる. これにより言語リテラシー能力の必要性を理解させる. オープンソースソフトウェア開発活動のように匿名の状態です学生間のコミュニケーションを行い, 異質な集団に近い状態でのコミュニケーションを行うようにする. 他者からの情報は妥当性を評価し, 表面的な議論を行いにくい状態を作る. 学生全員に, GitHub で他者に貢献するように促した. また, 貢献に対して評価し有益だと思う情報にはコメントを行うことや, GitHub のスタンプ機能を用いて評価するように促した.

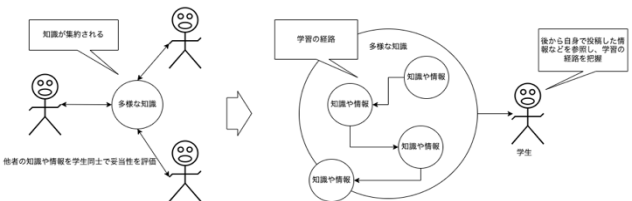


図 2 : 学生間で情報共有し, 整理された情報から学習経路を把握

知識の外化を促すため, 習得した知識や情報の言語化が得意な学生には, その知識を課外活動で技術ブログを執筆することにより外部に公開することも促した. 図 3 に技術ブログを執筆するために行う活動を示す.

習得した技術や知識を整理し, 再度妥当性を評価する. 技術ブログにするため, 整理された情報を言語化する. その後, 執筆した技術ブログを外部に公開し, 外部の集団から閲覧, 評価される.

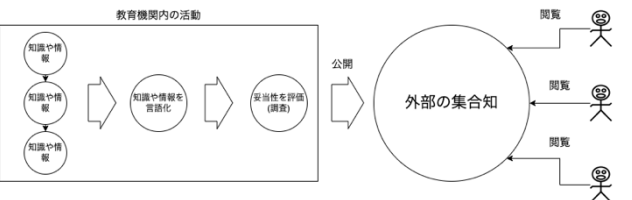


図 3 : 学習経路を把握し, 纏めた知識を外部に公開

本検証では, 実践的な学習も行う. 実践的な学習の中で派生知識に関心を持つように促す. また, 学生が学んだことがない知識や技術に触れさせることで, 学生の興味や関心を促す. 実践的な体験により学習意欲の向上を図る. 授業時間外でも演習に取り組める環境を用意する.

5. 授業設計

本検証は全 15 回の演習授業で行った. 授業構成は前半と後半で大きく分かれる. 多くの学生はサーバ構築未経験者のため, 前半では, テキストとワークシートを用いて, 基本的なアプリケーション構築を行い基礎から学習する. 後半は「なんらかの連携をするシステムを構築する」という自由な課題とした. 前述した (a)~(d) を促すため GitHub は演習全体を通して利用する.

図 4 に授業設計を示す. 前半の演習は第 1 回から第 7 回であり, 後半の演習は第 9 回から第 15 回である.

授業回数	第 1 回	第 2 回~第 6 回	第 7 回	第 8 回
授業形態	オンライン	オンライン	オンライン	オンライン
内容	調べ学習	LAMP 環境構築演習	LAMP 環境構築演習の口頭試問	E-learning のテスト
授業外学習	E-learning	E-learning	E-learning	指示なし
授業回数	第 9 回~第 14 回	第 15 回		
授業形態	オンライン	対面		
内容	自由課題	自由課題の成果発表		
授業外学習	指示なし	指示なし		

図 4 : 授業計画

5.1. 学習者

本学の情報システム工学科の学部 3 年の選択科目であるオペレーティングシステムを受講した学生である. 学生は, 本講義の E-learning で基本情報技術者試験に出題されるオペレーティングシステムの分野と同等の内容を学習している. 他の授業でデータベースやネットワークについても学習している.

5.2. 演習内容

前述した全 15 回の演習の前半の演習と後半の演習について述べる.

5.2.1. 前半の演習

前半の授業では、演習内容を示したサーバ構築演習を行った。Amazon Webservices(AWS)が提供する EC2 と呼ばれる仮想サーバ上に一般に LAMP と呼ばれる構成の WEB アプリケーションを構築し、実践的な演習を行った。アプリケーションの構成を図 5 に示し、アプリケーションの画面を図 6 に示す。実践的な演習を行うことで、発展的な知識習得や内発的モチベーションの向上を目指した。

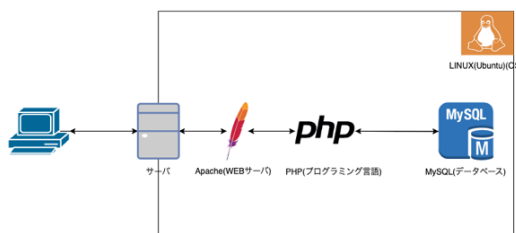


図 5 : LAMP 構成のアプリケーション構成図(OS の Linux, Web サーバーの Apache, データベースの MySQL, プログラミング言語の PHP を組み合わせた構成)

商品名 価格(円)

商品名 価格(円)

ノート 100円

図 6：アプリケーションのブラウザ画面

演習全体で時間や場所に関わらず AWS のサービスにアクセス可能な状態にし、いつでも演習に取り組めるようにした。[GitHub](#) も時間に関わらずコメントできるようにし、授業時間外もコミュニケーションが行える状態にした。

学生には、演習の内容を示すため、LAMP 構成のアプリケーション構築についてのワークシートを作成した。学生はワークシートに取り組みながら演習を行う。ワークシートでは演習に関することを幅広く設問にしている。サーバ構築を行うだけではなく、必要な動作の意味や仕組みを問う設問も用意した。

本演習に取り組む学生で、サーバ構築を経験している学生は少ないため、基礎知識の習得のため、ワークシートに準拠したサーバ構築演習の資料[6]を作成した。資料では、派生知識に関する外部の資料の提示も行った。また、演習で用いたコマンドや知識を解説している。詳細については参考になるインターネット上の資料を明示しており、知識を学生自身で深められるようにしている。これにより、ワークシートを参考に演習に取り組み、不明点は資料を参照しながら理解を深められるようにした。

また、サーバ構築未経験の学生が問題を認知し、問題の調査を行うことは困難だと考えられた。そのため、前提知識や経験が浅い学生に対して、自身で問題を認知し、調査するための支援するための仕組みが必要だと考えた。サーバ構築演習支援ツール[7]を開発し、前半の演習で導入した.AWS の EC2 内でツールを実行することができる。ツールによりどこに問題が存在しているのかを調査し、演習者に解決策や問題点を提示することができる。

ワークシートに取り組むことは指示したが、資料を参照することやサーバ構築演習支援ツールを用いることを提案したが、用いることを指示していない。

[illegible]

図7：(左)ワークシート、(右)参考資料

PHP_get_db_data	PHPからDBのデータが取得できるか確認(コードの改訂が完了)	よくできました!
PHP_get_db_data	PHPからDBのデータが取得できているか確認	よくできました!
Appendix(Python(C01))	Apechの設置を確認	よくできました!
Appendix(Python(C01))	Pythonのバージョンを確認	よくできました!
Appendix(Pur(C01))	Apechの設置を確認	よくできました!
Appendix(Pur(C01))	Perlのバージョンを確認	よくできました!
Appendix(Python(Flash))	Flashがインストールされているか確認	よくできました!
Appendix(Python(Flash))	uGDSがインストールされているか確認	よくできました!
Appendix(mod-vngl-py)	libapache2-mod-vngl-pyがインストールされているか確認	よくできました!
Appendix(Python(Flash))	uGDSの設置を確認	よくできました!
Appendix(Python(Flash))	Apechの設置を確認	よくできました!
Appendix(Python(Flash))	Pythonのコードを確認	間違っています... Pythonにコードが誤りがあります。 /var/www/html/flash/p/main.pyを確認しましょう。
Appendix(Python(Flash))	PyMySQLがインストールされているか確認	よくできました!
Appendix(Python(Flash))	Pythonのコードを確認(データベースからappleが取得できているか)	間違っています... Pythonにコードが誤りがあります。 /var/www/html/flash/p/main.pyを確認しましょう。

図 8：サーバ構築演習支援ツールが動作している様子

5.2.2. 後半の演習

「なんらかの連携をするシステムを構築する」という自由な課題では、前半の LAMP 構成のアプリケーション構築演習で学んでいないことに積極的に学び、サーバ構築演習に取り組むように促した。後半の課題でも EC2 が利用できる状態にした。また、利用可能な AWS の他のサービスも紹介した。最後の授業では、知識の外化を行うため他の学生の前で後半の授業の成果物の発表を行った。

5.3. オープンソースソフトウェア開発プロセスの要素

OSS 開発プロセスの要素を導入するため, GitHub を用いた学生同士での議論や外部に知識を公開するための技術ブログ執筆について述べる.

5.3.1. GitHub

GitHub を用いたコミュニケーションは学生全員で行い、学生同士が認知しにくい状態で行った。これにより、情報の妥当性の評価、異質な集団での活動を目指した。**GitHub** のコメントの書き方にルールを設け、学生に書くべき内容を示した。これにより、学びかた(学習プロセス)の把握、基礎的リテラシーの向上を図った。

図 9 に本検証で用意した **GitHub** の様子を示す[8]。自身の課題をタイトルにし、スレッドとして投稿する。学生らは、同様な課題のコメントを参照し、自身の演習環境に適用する。学生同士がお互いに認知しにくい状態でディスカッション

することで、情報の妥当性などを学生自身で考慮しながらコメントを読むことができる。

スレッドの投稿に記載する内容は、以下のことについて書くように提示した。「何をしようとしているのか」「何故それがしたいのか」「エラーログ」「エラーの解決のために行ったこと」の4つの項目である。1つの問題についての会話の流れがスレッドに集約され、学習の流れを把握することができる。



図9：本検証で作成されたスレッド

5.3.2. 技術ブログ

4で述べた(d)を促すため、学生に技術ブログの執筆を勧めた。執筆の方法(技術ブログのプラットフォームについて紹介、ブログのテーマ例を提示)を行った。執筆のテーマ例は、「後半の演習で取り組みたい内容」や「今までで学んできた知識や技術」などを学生に提示した。執筆した学生には、執筆したブログを Qiita Advent Calendar[9]に登録するように促し、Qiita Advent Calendar から他の学生もブログを閲覧できるように管理した。

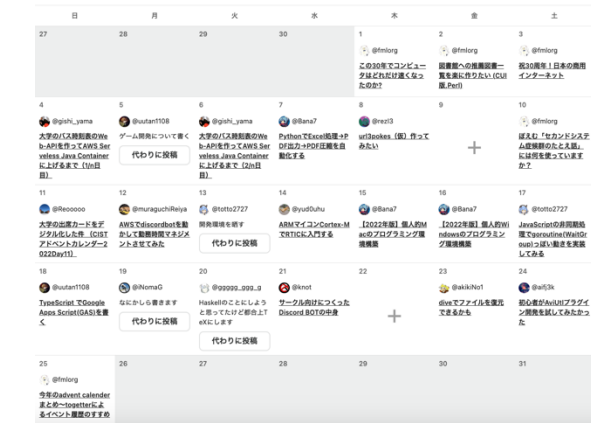


図10：本検証で用いた Qiita Advent Calendar の様子(本検証に参加している学生以外にも技術ブログを登録している)

6. 評価

前半の演習の事前事後アンケートの差分、授業全体を通しての GitHub のコメント履歴およびインタビューをもとに評価を行う。

6.1. 前半の演習の主体的な学び

前半の演習前と演習後にアンケートを行った。同じような質問を2回行い、学生の主体的な学びの変化を以下に示す。このアンケートでは ARCS モデルを参考に作成した。ARCS モデルとは、注意 (Attention)、関連性 (Relevance)、自信 (Confidence)、満足感 (Satisfaction) の4要素の頭文字を

とった学習モデルである[10]。この4要素が学習意欲を向上させる要因になると言われている。本研究で提案する学習モデルにおいても、学習者の ARCS モデルの要素を向上させることは、主体的に学習を促すために不可欠であると考えたためである。

検証前では、満足感を調査することはできないため、「注意」「関連性」「自信」の3項目をアンケートにより調査した。

表1：ARCS モデルのアンケート結果(n=25)

項目	向上	変化なし	低下
注意	9人(36%)	15人(60%)	1人(4%)
関連性	8人(32%)	10人(40%)	7人(28%)
自信	13人(52%)	10人(40%)	2人(8%)

表1より、3割以上の学生が主体的な学びに良い効果があった。一方で、「関連性」が低下した理由で、サーバ構築演習と学生自身が興味ある機械学習などの知識との関連性が明らかにならなかったことが低下の原因であるとアンケートの自由記述で書かれていた。

6.2. 同期、非同期なコミュニケーション

事後アンケートにのみ設けた設問「(本検証以外で行っている)グループワークで積極的に発言できたか」「GitHub で積極的にコメントできたか」をまとめると、21名(84%)の学生は GitHub のコメントよりグループワークの方が積極的に発言できたと回答した。

アンケートやインタビューで分かった非同期なコミュニケーションと同期的なコミュニケーションの特性を表2に示した。それぞれのコミュニケーションの「学生の他者貢献に対する積極性」、「学生が感じるコミュニケーションの難易度」、「コミュニケーションを行い問題解決するまでの時間」、「学生が発言することに対して抵抗があるか」のについて示した。

表2：非同期なコミュニケーションと同期的なコミュニケーションの特性

	積極性	難易度	時間	抵抗の有無
同期的なコミュニケーション	高い	低い	短い	ない
非同期なコミュニケーション	低い	高い	長い	ない

テキストコミュニケーションの情報共有はグループワークと比較し、積極的に情報共有が行われにくいことが分かった。また、GitHub のコメントには、自身の考察や知識の情報がなく、他者に情報が十分に伝達できないことがあった。一方で、テキストコミュニケーションの時間や場所に縛られないことや、情報を整理する時間が十分にあることなどをメリットに感じつつ、テキストコミュニケーションは同期的なコミュニケーションと比較し難しいと感じている学生が多い。

LAMP 構成のアプリケーション構築のワークシートが順調に進んだと感じた学生と、順調に進まなかったと感じた学生に分けて集計した結果を以下に示す。この2つの区分は学生の自己申告である。第7回の授業に行ったアンケートの「ワークシートの演習は順調に進みましたか」の設問に対して、「順調に進んだ」と回答した学生は「ワークシートが順調に進んだ学生」とした。反対に、「Issue を使いながら進めた」「順調に進まなかった」と回答した学生は「ワークシートが順調に進まなかった学生」として集計した。アンケ

ート回答者が 25 名に対して、「ワークシートが順調に進んだ学生」は 9 名、「ワークシートが順調に進まなかった学生」は 16 名だった。学生を前述した 2 つのグループに分けて集計した結果、「ワークシートが順調に進んだ学生」の方が「ワークシートが順調に進まなかった学生」よりも本検証で用いた GitHub で他者に積極的に貢献しない傾向があった。一方で、本検証以外の授業で行なわれているグループワークでは、グループ間で大きな差は見られなかった。

6.3. サーバ構築演習支援ツール

サーバ構築演習支援ツールに関する設問では、12 名 (48%(n=25))の学生が「役に立った」と回答した。

6.4. GitHub に投稿されたコメント

GitHub に投稿された内容には情報が整理された投稿もあった。一方で、一部の投稿で質問すべき問題やエラー情報が整理されておらず解決までに時間がかかっていた。

6.5. 技術ブログ

教員からの働きかけにより、3 名 (12%(n=25))の学生が技術ブログを執筆し、認知プロセスの外化に取り組んだ。

また、学生によって執筆された技術ブログを参考に後半の自由な演習課題に取り組んだ学生がいた。

6.6. インタビュー

4 名の学生にインタビューも行い、学生の特性に関して調査した。インタビューの対象者は、GitHub のコメント履歴や技術ブログの執筆の有無、アンケートの結果から、傾向が異なる学生を選択した。(I)学内外かかわらず主体的に課外活動に取り組む学生は、新しいことへの挑戦を苦痛に感じていない。この演習でも自身で課題を設定し、能動的に学んでいた。また、技術ブログの執筆も行っていた。執筆は教員からの助言があったためと回答している。興味を持っていることを実践できない原因として、教員や友人の後押しが必要だと感じている。(II)課外活動に消極的な学生は、主体的に学ぶことに苦痛を感じている。この演習では、自身で課題を設定できず、能動的に課題に取り組んでいない。

6.7. 前年度のアンケートと比較

前年度の今年度の本授業に対して学生が期待したことについて学ぶことができたかについての匿名アンケート結果を示す。このアンケートは、第 15 回目の授業終了後に行った。「授業をつうじて伸ばしたいと思ったスキルが伸びましたか」の設問に対しての学生の回答を以下の表に示す。前年度のアンケート回答者数は 22 人、今年度は 19 人が回答した。結果を以下の表 5.7.1 に示す。

表 3 : (左)前年度,(右)今年度

項目	人数	割合	項目	人数	割合
とても伸ばせた	3 人	13.6%	とても伸ばせた	9 人	47.4%
伸ばせた	16 人	72.7%	伸ばせた	9 人	47.4%
伸ばせなかった	3 人	13.6%	伸ばせなかった	1 人	5.3%
全く伸ばせなかった	0 人	0%	全く伸ばせなかった	0 人	0%

「伸ばしたいと思った(思っていた)スキルについて教えてください」の設問では、前年度と今年度の学生のどちらも「システムがネットワークに繋がる仕組み」「クラウド構築のスキル」「複数のシステム・サービスを連携して一つ

のシステムを作るスキルや、サーバの構築スキル」などの回答していた。本検証を行った講義に学生が求める授業内容は前年度と今年度で大きな差がないと考えられる。前年度と今年度では前半の演習の有無が大きく異なる。表 3 より、前年度と比較し、今年度の学生は授業冒頭で思っていた学びたいことを前半の演習授業を通じて伸ばせたと感じている学生が増加した。

7. 考察

本研究で提案する学習モデルをサーバ構築演習に適用することで、学生の ARCS モデルの関連性、注意、自信の項目を向上させることができた。これは、インタビューやアンケートの結果から、他のプログラミングの授業との関連性が明らかになったことや、周辺知識が身についたことが要因だと考えられる。また、前半の演習課題が完了できたことが自信につながっていることも考えられる。

テキストコミュニケーションでは、同期的なコミュニケーションと比較し、積極的な情報共有が行われにくいことが分かった。一方で、テキストコミュニケーションの時間や場所に縛られないことや、情報を整理する時間が十分にあることなどをメリットに感じつつも、テキストコミュニケーションは同期的なコミュニケーションと比較して難しいと感じている学生が多い。アンケートの結果から、対話する相手の知識レベルや抱えている課題を認知するまでの時間が影響していると考えられる。

評価で述べた(I)のような学生は、少し後押しすることで、OSS 文化へ一歩を踏み出せる学生であり、本検証で、技術ブログで認知プロセスの外化を行った。一方、(III)のような学生は、派生知識に無関心であるように見受けられ、課外活動に対し苦痛を感じる学生もいる。GitHub の履歴から、技術文書化する能力が低いため、テキストコミュニケーションが不得手であり、うまく質問もできず、課題に能動的に取り組めていないと推測される。

8. 議論

学生に課題を設定させて取り組ませる課題は、「主体的に課外活動に取り組まない傾向がある学生」や「自身で調査し理解を深めながら学習することに苦痛を感じる学生」にとっては難易度が高いことが分かった。また、教員が示す課題や知識習得に焦点を当てた課題に対して不満を感じる学生がいることが分かった。このような学生は、短時間で完成させることができる課題や自身が興味ある題材の課題に関心を示すと考えられる。

このような学生にとって、同じような感性を持つ学生が体系的に情報をまとめた資料に取り組むことが有効だと考える。提案として、図 11 のような学びの様子を示す。意欲的な学生らが好きな題材について体系的に情報をまとめ公開する。公開された情報を他の学生が参照し、短時間で再現し達成感を感じる。再現後に、理解を深め、自身で変更を加える。これにより、意欲的な学生から多様な情報やアイデアが提供され、意欲的でなかった学生も意欲的な学生に影響され課題に取り組むことができると考えられる。

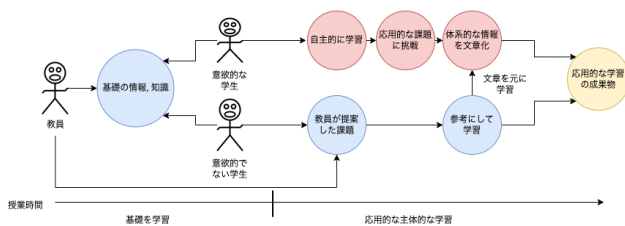


図 11：意欲的な学生と意欲的でない学生の学び

本検証では、全 15 回の授業で技術ブログを用いて認知プロセスの外化を生間以外の集団に対して行うように促した。しかし、ブログを執筆した学生は 3 名であり検証に参加した学生の 1 割程度だった。また、この執筆された技術ブログを参考に自由課題に取り組んだ学生も数名いた。本検証では、少数の学生だったが、意欲的な学生が主体的に活動し、執筆した体系的な情報である技術ブログが、意欲的でなかった学生に影響を与え、技術ブログを参考に自由課題に取り組ませたと推測される。

このように、意欲的な学生が、大学生活を通して楽しみながら情報を纏め社会に公開する活動などを通して社会に貢献することで、教育機関を離れた後も主体的に学び、社会の集団に意欲的に価値を提供していくと考えられる。また、意欲的な学生が纏めた体系的な情報を他の学生も閲覧可能な状態にすることで、意欲的でない学生に良い変化を与えられると期待される。

本検証では期間が短く調査することはできなかったが、OSS 開発プロセスを取り入れることで期待される長期的な学生の学びを図 12 に示す。特定の授業だけではなく、様々な授業を通じて、長期的に前述した活動を行うことで、より多くの学生が社会に貢献し、多くの学生に良い影響をもたらすと考えられる。

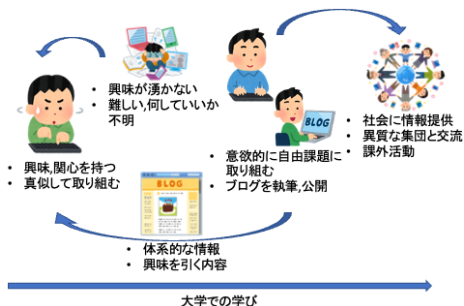


図 12：間接的に意欲的な学生が他の学生に与える影響

9. おわりに

我々は、OSS 開発プロセスをサーバ構築演習という実践的な演習授業に導入し、利点や欠点、また運用上の困難さについて調査した。多くの学生にとってサーバ構築演習は初めての体験になるため、授業の前半は基礎的な学習、演習を行った。これが自信につながり、全体をとおして、よく取り組めたように感じているようである。実際、後半の自由課題の取り組みは様々で学生自身で学んだ技術が取り入れられていた。一方で、OSS 開発プロセスの要素に積極的に取り組んだ学生は二割程度にとどまった。サーバ構築演習と同様に OSS も多くの学生にとって初めての概念であるため、半年の授業が短いだけなのか、概念になじめないのか、動機に共感できないのかなどの取り組みなかった原因を調査し、今後それらについて検討する必要がある。また、本授業で OSS を紹介したのは今後 IT 技術者になりたいと考える学生にと

って重要であると我々は考えているためである。OSS 開発プロセスを取り入れ、実践的な学びを行ったことが実を結んだかどうかは 5 年 10 年先にならないと分からない。この追跡調査も必要と考える。いずれにせよ我々は OSS を紹介し、OSS 運動に気づいた学生たちにとって良いキッカケとなったことを願っている。

参考文献

- [1] 国立教育政策研究所(2013).「国立教育政策研究所プロジェクト研究（平成 21～25 年度）「教育課程の編成に関する基礎的研究」」. 文部科学省. https://www.mext.go.jp/b_menu/shingi/chousa/shotou/095/shiryo/_icsFiles/afieldfile/2013/07/18/1336562_01_1.pdf, (参照 2023-02-20)
- [2] 東海 A(教育力)チーム(2014).「アクティブラーニング失敗事例ハンドブック」. 文部科学省. <https://www.hedc.mie-u.ac.jp/pdf/ALShippaiJireiHandbook.pdf>, (参照 2023-02-20)
- [3] 産業人材政策室(不明).「我が国産業における人材力強化～「人生 100 年時代」の「働く」と「学ぶ」の一体化～」. 経済産業省. https://www.meti.go.jp/committee/kenkyukai/sansei/jinzai-ryoku/pdf/001_02_00.pdf, (参照 2023-02-20)
- [4] 社団法人情報サービス産業協会. (2005).「オープンソースビジネスに取り組む SI 企業のための企業ポリシー策定ガイドライン」. <https://www.jisa.or.jp/Portals/0/report/16-J013.pdf>. (参照 2023-02-20)
- [5] 独立行政法人 情報処理推進機構(2022).「情報処理技術者試験・情報処理安全確保支援士試験 試験要綱」. https://www.jitec.ipa.go.jp/1_13download/youkou_ver5_1.pdf, (参照 2023-02-20)
- [6] LAMP 構成のアプリケーション構築資料. <https://github.com/lectures-fmlorg/os-2022/blob/main/pdf/docs.pdf>, (参照 2023-02-20)
- [7] サーバ構築支援ツール. <https://github.com/OHMORIYUSUKE/grech>, (参照 2023-02-20)
- [8] GitHub, Inc.(不明).「lectures-fmlorg/os-2022」. <https://github.com/lectures-fmlorg/os-2022/issues>, (参照 2023-02-20)
- [9] Qiita Inc.(不明).「CIST (公立千歳科学技術大学) Advent Calendar 2022」. <https://qiita.com/advent-calendar/2022/cist>, (参照 2023-02-20)
- [10] J.M. ケラー 著 鈴木 克明 監訳. 学習意欲をデザインする. 北大路書房. 2010