一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SS2022-72(2023-03)

ソフトウェアアップサイクルのための事例共有システムの開発と評価

中田　匠哉† 　陳　　思楠† 　佐伯　幸郎†† 　中村　匡秀†,†††

† 神戸大学 〒 657–8501 神戸市灘区六甲台町 1–1
††† 理化学研究所・革新知能統合研究センター 〒 103–0027 東京都中央区日本橋 1–4–1
†† 高知工科大学 〒 782–8502 高知県香美市土佐山田町宮ノ口 185
E-mail: †tnakata@ws.cs.kobe-u.ac.jp, ††chensinan@gold.kobe-u.ac.jp, †††saiki.sachio@kochi-tech.ac.jp,
††††masa-n@cmds.kobe-u.ac.jp

あらまし　ソフトウェア開発においては，プログラミングの知識だけでなく，ドメイン知識や課題解決のアイデアが必要であることが知られている．しかし，技術を特定のドメインに適応するための経験的知識は，言語仕様ドキュメントや書籍などで得られる技術的知識に比べて獲得が困難である．そこで，過去の開発プロジェクトに蓄積された知識をアップサイクルすることで，開発効率の向上と既存知識の活用を目指すソフトウェアアップサイクルという考え方がある．本研究では，先行研究で提案されているアップサイクルの実績を共有するソフトウェアアップサイクル事例共有システムを実装し，評価実験を行う．評価実験の目的は，集合知によってアップサイクルが効率化するかどうかを評価し，事例共有システムの検索・登録機能の品質を評価することである．具体的には，被験者が 2 種類の開発タスクに取り組み，開発結果を事例共有システムに登録する実験を行い，結果を通じて集合知の効果やシステムの有効性・効率性・信頼性を調べる．

キーワード　ソフトウェアアップサイクル，ソフトウェア再利用，知識ベース，集合知，開発効率

Development and Evaluation of Case Knowledge Base for Software Upcycling

Takuya NAKATA†, Sinan CHEN†, Sachio SAIKI††, and Masahide NAKAMURA†,†††

† Kobe University, Rokkodai-cho 1–1, Nada-ku, Kobe, Hyogo, 657–8501 Japan
††† Riken AIP, 1–4–1 Nihon-bashi, Chuo-ku, Tokyo, 103–0027 Japan
†† Kochi University of Technology, 185 Tosayamadacho Miyanokuchi, Kami, Kochi, 782–8502 Japan
E-mail: †tnakata@ws.cs.kobe-u.ac.jp, ††chensinan@gold.kobe-u.ac.jp, †††saiki.sachio@kochi-tech.ac.jp,
††††masa-n@cmds.kobe-u.ac.jp

Abstract　In this study, a software upcycling case sharing system is implemented to share the achievements of previous upcycling studies, and evaluation experiments are conducted. The purpose of the evaluation experiment is to evaluate whether upcycling is made more efficient through collective intelligence and to evaluate the quality of the search and registration functions of the case sharing system. Specifically, the subjects undertake two types of development tasks and conduct experiments to register the development results in the case sharing system, examining the effect of collective intelligence and the effectiveness, efficiency, and reliability of the system through the results.

Key words　Software upcycling, Software reuse, Knowledge base, Collective intelligence, Development efficiency

1. はじめに

ソフトウェア開発では，プログラミングの知識だけでなくドメインの知識や課題解決のアイデアが必要とされる [1], [2]．情報技術に関する知識は，プログラミング言語の仕様ドキュメントを参照したり，個人や企業が執筆した書籍・資料・Web サイトを閲覧することで得られることが多い [3]．しかしながら，技術を特定のドメインに適応した知見や顧客の要求を達成するための工夫といった経験的知識を得ることは困難である [4]．リポジトリに蓄積された過去の開発プロジェクトにはソースコードだけでなく設計文書やアーキテクチャなどが含まれており，抽出すべき先人の知識が多く存在すると考えられる [5]．

ソフトウェアアップサイクルは，既存プロジェクトに隠された知的資産を新規のソフトウェア資産として転換することで，

開発効率の向上と既存知識の活用を目指す考え方である [6]．アップサイクルにおける課題は，既存プロジェクトからのアップサイクル素材の抽出と抽出した素材を組み合わせるアップサイクル手法の構築の 2 つ存在する．先行研究の**ソフトウェアアップサイクル事例共有システム**は，アップサイクルの実績をアップサイクル事例として記録・共有することで集合知を形成し，アップサイクル手法の構築を効率化することを目的として提案されたシステムである [7]．利用者としてシステムに蓄えられた事例を参考にし，素材を活かした開発のアイデアを閃き，アップサイクルを実践し，さらに開発者として事例をシステムに登録することで，アップサイクル開発を効率的に循環させる．しかし，集合知によってアップサイクルを効率化できるか具体的な評価が存在していない．さらに，システム自体の具体的な実装と品質評価が存在しない．したがって，提案システムを用いたアップサイクルの評価実験が必要である．

本研究では，事例共有システムを具体的に実装し，評価実験を行う．バックエンドシステムと Web UI を実装し，実際に利用可能なシステムを実装する．評価実験の目的は，集合知によってアップサイクルが効率化するかどうかを確認するとともに，実装したシステムの検索・登録機能の品質を評価することである．具体的には，以下の 3 つのリサーチクエスチョンを設定する．

RQ1　集合知によってアップサイクルは効率化するか？

RQ2　事例共有システムの検索機能の有効性・効率性・信用性はどうか？

RQ3　事例共有システムの登録機能を用いた事例の登録はどれだけ時間がかかるか？

評価実験は 6 人の学生を対象に行う．被験者は，2 種類の開発タスクに取り組み，開発結果を事例共有システムに登録する．開発の様子の録画と実験後アンケートを通じて結果を得る．実験結果から 3 つの RQ と事例共有システムについて考察を行う．

2. 準　　備

2.1 ソフトウェアアップサイクル

ソフトウェアアップサイクルとは，環境問題においてリユースやリサイクルに近い概念として扱われる**アップサイクル**の考え方をソフトウェア工学に適用した概念である [6]．アップサイクルとは，製品の再利用においてもとの素材が持つ特徴を活かしながら全く新しいプロダクトを生み出す考え方である [8]．

例として，廃棄されるメガソーラーのソーラーパネルの再利用を考える．リユースでは，劣化した部品を交換するなどしてソーラーパネルとして再利用される．リサイクルでは，ガラスや金属といった資源に一度加工したうえで，様々な製品の素材として再利用される．アップサイクルではリユースやリサイクルと異なり，ソーラーパネルの性質を生かしたまま本来とは異なる文脈で用いる．具体的には，パネルの平面の頑丈な板という特徴に着目して脚を取り付けておしゃれなテーブルを作る，といった革新的な発想で新たな価値を生み出す考え方がアップサイクルである．

ソフトウェアアップサイクルは，既存プロジェクトの一部分

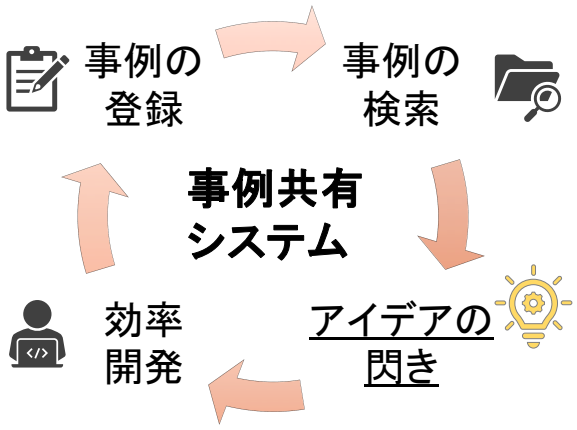


図 1: 事例共有システムの開発サイクル

であるコード・設計・文書等の**プロジェクト素材**を，アップサイクルによって新規の価値あるソフトウェア資産に転換する考え方である．加工元プロジェクトの実装詳細を捨て去り，価値ある動作や設計のみを新たなプロジェクトで利用する．ソフトウェアアップサイクルのアプローチは，**ニーズ指向**と**シーズ指向**の 2 つである．ニーズ指向アプローチは，再開発コストの低減による開発効率の向上を目的として，既存プロジェクトに作りこまれた英知を利用してアップサイクルするものである．シーズ指向アプローチは，未活用の様々な既存プロジェクト素材を活用することを目的として，革新的なプロダクトを創出するものである．ソフトウェアアップサイクルは，既存プロジェクトからアップサイクル素材を抽出する工程と，素材を組み合わせて開発する工程に分かれている．特に，抽出した素材をどのように組み合わせてアップサイクルを行うかは自明でなく，素材を用いた開発手法の構築が課題である．

2.2 先行研究：ソフトウェアアップサイクル事例共有システム

ソフトウェアアップサイクル事例共有システムは，アップサイクルの実績をアップサイクル事例として記録・共有することで集合知を形成し，未来のアップサイクルを効率化することを目的として提案されたシステムである [7]．アップサイクル事例とは，アップサイクルの実績やアイデアをまとめたデータ構造である．4 つの主要素によって構成され，`context` は目的・経緯・課題を，`materials` はアップサイクル素材の一覧を，`recipe` はアップサイクルの組み合わせ方・レシピを，`result` は結果を表している．`materials` が保持する各素材は，コード片・クラス図・ドメイン等任意のプロジェクト素材を文章と参考資料で説明したものである．事例モデルに関する予備的評価を行った先行研究から，アップサイクル素材の用途はソースコード再利用に限らずアーキテクチャやアルゴリズムなど幅広い価値が存在することが分かっている [9]．提案システムのユーザは，システムに事例を登録する登録者と，システムから事例を検索して開発の参考にする利用者の 2 種類である．利用者としてシステムに蓄えられた事例を参考にし，素材を活かした開発のアイデアを閃き，アップサイクルを実践し，さらに開発者として事例をシステムに登録することで，図 1 に示すアップサ

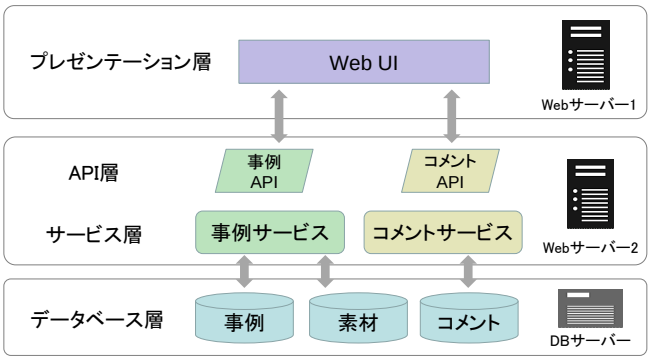


図 2: 事例共有システムの実装アーキテクチャ

図 3: PC 版の Web UI 画面

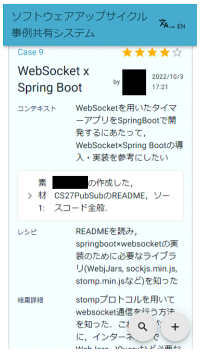


図 4: スマートフォン版の Web UI 画面

イクル開発のサイクルが構築できる．

提案システムの課題は 2 つある．まず，集合知によってアップサイクルを効率化できるか不明であり評価が求められる．さらに，システム自体の具体的な実装とシステムの品質に関する評価が存在しない．したがって，提案システムを用いたアップサイクルの評価実験が必要である．

3. 実　　装

3.1 アーキテクチャ

システムの実装アーキテクチャを図 2 に示す．システムの構築には 3 つのサーバーを用いた．1 台目の Web サーバー（Web サーバー 1）は，ユーザが操作する Web User Interface（Web UI）を実現するサーバーである．TypeScript，Next.js，Material UI といった JavaScript 系の環境で実装した．2 台目の Web サーバー（Web サーバー 2）は，事例登録等のシステムの内部処理を実現するサーバーである．Kotlin，Spring Boot，Tomcat といった Java 系の環境で実装した．データベースサーバー（DB サーバー）は，事例・素材・コメントのデータベースを Relational Database で実現するサーバーである．本アーキテクチャの特徴は，フロントエンドとバックエンドの分離である．Web サーバー 2 が事例登録等の処理を行う Application Programming Interface（API）を提供し，Web サーバー 1 が Web UI を介して API を呼び出すことで動作する．

3.2 UI デザイン

PC 版とスマートフォン版の Web UI 画面をそれぞれ図 3，4 に示す．UI デザインの特徴は，検索・事例表示・登録の 3 カラムレイアウトである．また，各事例への Good ボタン機能とコメント機能を追加し，利用者から登録者へのフィードバックを可能にした．検索・登録の入力補助として，アップサイクル事例の概念を説明するヘルプテキストを充実させた．

4. 実　　験

4.1 目　　的

本実験の目的は，集合知によってアップサイクルが効率化するかどうかを確認するとともに，実装したシステムの検索・登録機能の品質を評価することである．特に，SQuaRE [10] の利用時品質指標における，有効性・効率性・信用性を評価する．リサーチクエスチョンとして，以下の 3 点を設定した．

RQ1　集合知によってアップサイクルは効率化するか？

RQ2　事例共有システムの検索機能の有効性・効率性・信用性はどうか？

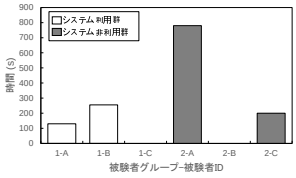
RQ3　事例共有システムの登録機能を用いた事例の登録はどれだけ時間がかかるか？

4.2 実 験 設 定

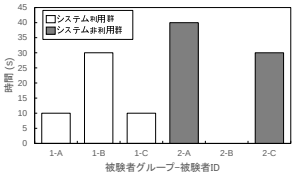
実験の被験者は，神戸大学中村研究室に所属する修士課程の学生 6 名である．被験者はランダムにグループ 1，グループ 2 に 3 人ずつ分ける．実験の流れは，まず，事例共有システムへの知識を均一化するためにシステムの事前学習および検索・登録操作の練習を行う．その後，2 種類の開発タスク（タスク 1，タスク 2）に 30 分ずつ取り組み，開発の様子を録画する．各タスク終了後，開発事例を事例共有システムに登録し，登録の様子を録画する．全てのタスク終了後，アンケートに回答する．

グループ 1 とグループ 2 の違いは，タスクに取り組む際にシステムを利用するかどうかである．グループ 1 は，タスク 1 を事例共有システムを最低一度は使いつつ，他の検索システムを自由に併用して開発に取り組む**システム利用群**である．一方グループ 2 は，タスク 1 を事例共有システムを使わず，他の検索システムのみを利用して開発に取り組む**システム非利用群**である．タスク 2 では，システム利用の有無が逆になり，グループ 1 がシステム非利用群，グループ 2 がシステム利用群となる．

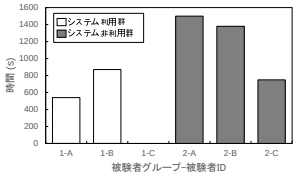
実験に用いる事例共有システムは 3. で実装したシステムである．実験開始前にシステムに 24 件のアップサイクル事例が登録された状態で実験を行った．



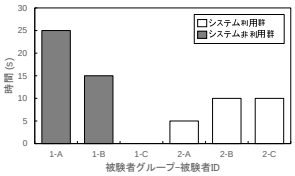
(a) タスク 1 の手指検出にかかった時間



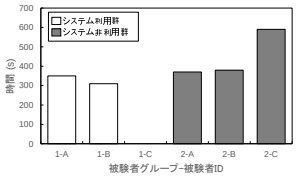
(b) タスク 1 のエージェントの知識到達にかかった時間



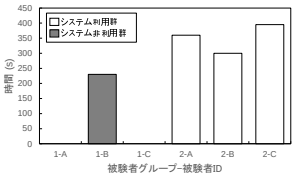
(c) タスク 1 の手指検出の知識到達にかかった時間



(d) タスク 2 のシフト生成アルゴリズムの知識到達にかかった時間



(e) タスク 1 の事例登録にかかった時間



(f) タスク 2 の事例登録にかかった時間

図 5: 時間計測の結果

4.2.1 実験で取り組んだタスク

タスク 1 は，バーチャルエージェントとの対話におけるハンドジェスチャーによる入力機能のモックの実装である [11]．タスクで指定されたバーチャルエージェントは JavaScript で実装されているものである．指定のバーチャルエージェントに関する知識は神戸大学中村研究室がクローズドに保有する知識であり，Google 検索などのオープンな検索ツールでは詳細な知識を得ることは不可能である．バーチャルエージェントに関する知識を得る方法としては，クローズドな研究室内 Wiki，ソフトウェアリポジトリ，事例共有システムによる検索がある．ハンドジェスチャー機能の実装には 2 つのステップがある．ステップ 1 は，Web カメラを用いて手指の様々な位置座標を検知することである．ステップ 2 は，手指座標からハンドジェスチャーの種類を識別することである．

タスク 2 は，シフト自動作成ツール向けの簡易アルゴリズムの実装である．本タスクでは，開発言語として Python を指定した．シフト自動作成問題は一般的に最適化問題として解かれることが多い．一方，中村研究室がクローズドに保有する知識として，貪欲法に近い簡易アルゴリズムが存在し，システムに登録された事例からのみアクセスできる．簡易アルゴリズムは，アルゴリズムに関する基礎知識があれば容易に思いつくことが可能な難易度である．

タスク設定の特徴は 2 点ある．1 点目として，タスク 1 とタスク 2 の開発言語を分けることで，短期的な言語仕様の慣れによる開発速度の変化の影響を小さくしている．2 点目として，タスク 1 とタスク 2 の開発にあたって要求する知識として，システムに登録されたクローズドな知識かつシステムを利用せずに獲得が可能な知識を設定している．これによって，取り組む開発タスクは単なる開発ではなくクローズドな既存知識を用いたアップサイクルタスクとなる．

表 1: Q2，Q3 による検索の成功・失敗に関するアンケート結果

タスク	検索ツール	成功/件	失敗/件
1	システム	3	0
1	その他	4	1
2	システム	2	1
2	その他	4	2

4.3 結　　果

4.3.1 時 間 計 測

タスクで開発に取り組む様子とシステムに事例を登録する様子の録画から様々な作業時間の計測を行った．結果を図 5 の棒グラフに示す．タスク 1 とタスク 2 でシステム利用群とシステム非利用群のグループが入れ替わっているため，分かりやすさのためにシステム利用群を白色，システム非利用群を黒色で表現している．ただし，図 5a の被験者 1-C のデータのように，録画の失敗によってデータのいくつかが欠損したため，得られなかったデータは表示していない．また，タスク 1 およびタスク 2 を制限時間の 30 分以内で完了した人はいなかったため，タスクの完了時間のデータを得ることはできなかった．ただし，タスク 1 においてタスク開始から手指検出成功までのステップ 1 の完了時間の取得に成功した．

図 5a は，タスク 1 の手指検出サブタスクの完了にかかった時間を計測した結果である．図 5b，図 5c，図 5d は，知識の検索にかかった時間を計測した結果である．図 5e，図 5f は，タスク 1，2 のそれぞれでシステムへの事例登録にかかった時間を計測した結果である．

4.3.2 アンケート

タスク後のアンケートでは以下の項目についてタスク 1 とタスク 2 のそれぞれで回答を得た．

Q1　タスクの解決に必要な知識のうち以前から持っていたものをすべて選択してください（タスク 1 の選択肢：手指認識・指定バーチャルエージェントの概要・JavaScript・HTML・その他自由記述，タスク 2 の選択肢：シフト作成アルゴリズム・アルゴリズムに関する一般的な理解・Python・その他自由記述）

Q2　システムを用いて目的の事例を検索することができましたか（選択肢：できた・できなかった・システム無しに割り当てられた）

Q3　システム以外のツールを用いて目的の情報を検索することができましたか（選択肢：できた・できなかった・使っていない）

Q4　システムの検索結果として得られた事例の内容は信用できましたか（選択肢：できた・できなかった・システム無しに割り当てられた）

Q5　システム以外の検索結果として得られた内容は信用できましたか（選択肢：できた・できなかった・使っていない）

表 2: 検索による知識到達にかかるグループ別平均時間

検索した知識	システム非利用群の用いた検索ツール	システム非利用群の平均時間/s	システム利用群の平均時間/s	平均時間の差/s
エージェント	リポジトリ	35	17	18
手指検出	Google 検索	490	193	297
シフト自動生成	Google 検索	10	8	12

Q6　システム以外の検索方法として何を用いましたか（自由記述）

Q1 の結果，タスクの解決に用いるプログラミング知識に関する知識はほぼ全員があらかじめ持っていた．タスク 1 で用いる JavaScript に慣れていない被験者が 2 人で，タスク 2 で用いる Python は全員が問題なく利用できた．Q2，Q3 の結果，検索の成功・失敗に関するアンケート結果は表 1 の通りとなった．Q4，Q5 の結果，タスク 1 でシステムの検索手段が信用できないと答えた被験者が 1 人いた．また，全てのタスクでシステム以外のその他の検索手段の検索結果が信用できると答えた．Q6 の結果，事例共有システム以外で使った検索手段として，全員が Google 検索を挙げた．研究室内のクローズドな Wiki やソフトウェアリポジトリを検索に利用した人もいた．

4.3.3 録　　画

タスクと事例登録の様子の録画から明らかになったことを述べる．まず，全員の被験者がパソコンでシステムを利用し，スマートフォンでは利用しなかった．また，タスク 2 では，システム利用群はシステムの検索によって発見した簡単なアルゴリズムを用いた実装を目指したが，システム非利用群は Google 検索で「シフト自動生成」と入力した際に多数ヒットするある程度数学的な知識が必要な数理最適化を用いた解放で解こうとしており，タスクに取り組む手法が全く異なった．数理最適化で解こうとした群は，事前知識が足りなかったため実験時間内のソースコードの改良に失敗していた．検索ワードのファジーさ（大文字小文字の違いの許容，類似結果の表示）という点では，システムより Google 検索等の他検索サービスがより優れているため，システムを用いた検索に数回失敗している様子が見受けられた．一方で，Google 検索での検索に失敗している例としては，発見した Web ページがタスクに役立つかを確認するための閲覧時間が長く，結果として不要なページの閲覧にタスクの時間を多く消費した人が多かった．また，Google 検索の利用目的として，開発指針を決める大枠の知識以外にもプログラミング言語に関する簡単な仕様を得るために利用している様子が多く見受けられた．

4.3.4 登録された事例

Google 検索で得た Web サイトの知識を素材とした事例では，素材の要素として Web サイトの URL と Google 検索で得たことしか書かれていないものが多かった．一方で，どのような検索ワードによって得られたかまで記載している事例も存在し，該当事例の登録時間は被験者の平均よりも長かった．Web サイトの素材に関しては，バージョンに関して記載している素材は存在しなかった．タスクが完了できなかった事例においても，開発の途中まで役に立った事例に関しては高い点数が付けられていた．特に，コードをコピーしてそのまま利用できたと記載されている事例の点数が高い傾向にあった．

4.4 考　　察

4.4.1 RQ1：集合知によってアップサイクルは効率化するか？

図 5a より，タスク 1 の手指検出サブタスクの平均達成時間は，システム利用群は 11 分 45 秒，システム非利用群は 20 分 10 秒であった．システムを用いた群の平均達成時間が 8 分 25 秒短くなり，42%減少した．しかしながら，平均達成時間に有意な差であるか検定を行うには標本数が少ないため，実験の結果から集合知によってアップサイクルが効率化するかは不明である．今回の実験の成果として，得られたデータにおける平均達成時間を比較するとシステム利用群が数値上で短くなったという，アップサイクルの効果に関する今後の研究への一つの知見が得られた．

4.4.2 RQ2：事例共有システムの検索機能の有効性・効率性・信用性はどうか？

検索機能の有効性は，表 1 の検索の成功・失敗に関するアンケート結果から評価する．ここで，有効性の評価における検索機能の目的は，利用者が検索したい知識を得ることとする．システムの検索機能を用いた検索では 6 件中 1 件が検索に失敗した．その他の検索ツールを用いた検索では 11 件中 3 件が検索に失敗した．このことから，Google 検索や Wiki といった従来ツールと比較して，システムによる検索機能の有効性は大きく劣っていないと考えられる．

検索機能の効率性は，表 2 に示す検索によって知識に到達したグループごとの平均時間で評価する．結果から，全ての検索においてシステムを利用することで平均検索時間が減少していることから，システムの検索機能の効率性がある程度示された．これは，多くの被験者が利用した Google 検索で得られた知識は，知識をどのように開発に役立てるかが簡潔にまとめられていないため，必要な知識かどうか確認するために時間がかかったと考えられる．

検索機能の信頼性は，検索結果に信頼できるかどうかのアンケートで評価する．システム以外の検索結果は全ての人が信用できると答えたのに対し，システムの検索結果は信用できないと答えた人が 1 人いた．回答者にインタビューしたところ，検索で見つかった事例が参照する素材プロジェクトにドキュメントが書かれていなかったため頼りない知識だと感じたと回答した．このことから，素材のドキュメントや事例の各項目の記述文の詳細さといった事例データの品質によって信頼性が変化すると考えられる．

4.4.3 RQ3：事例共有システムの登録機能を用いた事例の登録はどれだけ時間がかかるか？

図 5e，図 5f の結果から，事例登録の平均時間は 6 分 5 秒で，標準偏差は 1 分 44 秒であった．被験者によって登録時間のばらつきがあることがわかった．特に登録時間が長かった事例には各項目が詳細に記述されていたことから，事例登録に手間取る以外にも登録者が事例の質を上げようとすることも登録時間が長くなる原因となる可能性がある．

4.4.4 実験を通して得られた知見

両方のタスクにおいて 6 人全員がタスクを完了することができなかったが，サブタスクの時間計測によって知見が得られた．被験者のプログラミング技術・知識に応じてタスク完了度は変化する．複数のサブタスクを設定し，タスク完了度と各サブタスクの到達時間を適切に計測できる実験手法の構築が必要である．また，アップサイクル実験においては，システムに事例として登録された未知の既存知識と開発に必要な既知の既存知識を組み合わせて開発を行うため，適切なタスクの設定が必要である．

タスク 2 において，システム利用群とシステム非利用群でタスクへのアプローチが全く異なった．システムの検索機能と Google 検索では得られる知識が異なることが原因であると考えられる．システムの検索機能の効果として，開発時間の短縮だけでなく，開発者が必要な知識を得る過程でアップサイクルの知見を渡すことによって，開発アプローチを変化させる効果があると考えられる．

表 2 から，手指検出とシフト自動生成のように，システムの利用・非利用に関わらず求める知識の種類によって平均検索時間が大きく異なるものがあることがわかった．検索対象の知識の種類によってタスクの達成度や達成時間にどのような差があるか検討が必要である．

システム利用群におけるシステムと Google 検索の併用の様子から，システムは既存の検索手法の機能のすべてを代替するものではなく，既存手法と併用する開発補助ツールの一つという役割で用いられると考えられる．細かいプログラミング言語の仕様は事例共有システムでは網羅することが困難であるため，従来の検索システムの併用が必要となる．

5. ま と め

本研究では，ソフトウェアアップサイクルを効率化する手法であるソフトウェアアップサイクル事例共有システムに具体的な実装を与え，評価実験を行った．実装では，3 つのサーバーを用いてデータベース，バックエンドシステム，Web UI を構築した．評価実験では，6 人の学生を対象としてシステム利用群とシステム非利用群に分かれて 2 種類の開発タスクに取り組むアップサイクル実験を行った．結果，サブタスクに対してシステム利用群の平均達成時間が 42%短くなるという結果になった．また，システムを利用した検索機能は Google 検索などの他システムと比較して有効性・効率性・信頼性の観点で大きく劣っているとはいえないことが分かった．システムへの事例登録にかかる時間は約 6 分であった．

課題として，アップサイクル実験において被験者のプログラミング技術・知識に応じて変化するタスク完了度をはかるために，アップサイクルの評価に適切な実験手法の構築が必要であることが明らかになった．また，本研究を通じて得られた知見を活かして提案システムの改善に繋げたいと考えている．

謝辞 本研究の一部は JSPS 科研費 JP19H01138, JP20H05706, JP20H04014, JP20K11059, JP22H03699, JP19K02973, 特別研究員奨励費 22J13217，および，立石科学技術振興財団の研究助成を受けて行われている．

文 献

[1] S. Saeed, N. Jhanjhi, M. Naqvi, and M. Humayun, “Analysis of software development methodologies,” International Journal of Computing and Digital Systems, vol.8, no.5, pp.446–460, 2019.

[2] P. Oukes, M. vanAndel, E. Folmer, R. Bennett, and C. Lemmen, “Domain-driven design applied to land administration system development: Lessons from the netherlands,” Land Use Policy, vol.104, p.105379, 2021.

[3] Y. Wu, S. Wang, C.-P. Bezemer, and K. Inoue, “How do developers utilize source code from stack overflow?,” Empirical Software Engineering, vol.24, pp.637–673, 2019.

[4] H. Belani, M. Vukovic, and Ž. Car, “Requirements engineering challenges in building ai-based complex systems,” 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)IEEE, pp.252–255 2019.

[5] T. Siddiqui and A. Ahmad, “Data mining tools and techniques for mining software repositories: A systematic review,” Big Data Analytics, eds. by V.B. Aggarwal, V. Bhatnagar, and D.K. Mishra, pp.717–726, Springer Singapore, Singapore, 2018.

[6] K. Terakawa, S. Chen, and M. Nakamura, “Preliminary study of reasoning existing projects’ descriptions based on classname word elements, ” 23rd ACIS International Summer Virtual Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD2022-Summer), 2022.

[7] T. Nakata, S. Chen, S. Saiki, and M. Nakamura, “A study of case sharing system for efficient and innovative software upcycling,” Proceedings of 2022 International Conference on Data and Software Engineering (ICoDSE), pp.6–11, Nov. 2022.

[8] J.-S. Kwan, “Based on the perspective of sustainability, the characteristics of upcycle fashion design,” Fashion & Textile Research Journal, vol.14, no.1, pp.13–23, 2012.

[9] 中田匠哉，陳 思楠，佐伯幸郎，中村匡秀，“効率的なソフトウェアアップサイクルのための事 例知識ベースの予備的評価，” ソフトウェア工学の基礎ワークショップ FOSE2022，pp.227–228，Nov. 2022.

[10] 独立行政法人情報処理推進機構，“つながる世界のソフトウェア品質ガイド，” https://www.ipa.go.jp/files/000055008.pdf，May 2015. (Accessed on 2/20/2023).

[11] H. Ozono, S. Chen, and M. Nakamura, “Encouraging elderly self-care by integrating speech dialogue agent and wearable device,” 8th International Conference, ITAP 2022, Held as Part of the 24th HCI International Conference, HCII 2022, vol.LNCS 13331, pp.52–70, May 2022.