



ソフトウェアアップサイクル事例共有手法の評価と 今後の展望

神戸大学大学院 工学研究科

○中田匠哉 陳思楠 佐伯幸郎 中村匡秀

2024/02 第5回ソフトウェアエコシステムワークショップ

Session 2-2



発表概要

■ 着目する課題

- ◆ 発見した既存プロジェクト素材をどのようにして開発に活用するか？

■ 提案手法

- ◆ ソフトウェアアップサイクル事例共有システム(SUCCEEDシステム)
 - (A1) アップサイクル事例の定義
 - (A2) 事例の蓄積と活用

■ 評価実験

- ◆ SUCCEEDシステムを使った研究室メンバーによる開発実験

■ 今後の展望①

- ◆ ユーザの要望を叶えるサービス開発

■ 今後の展望②

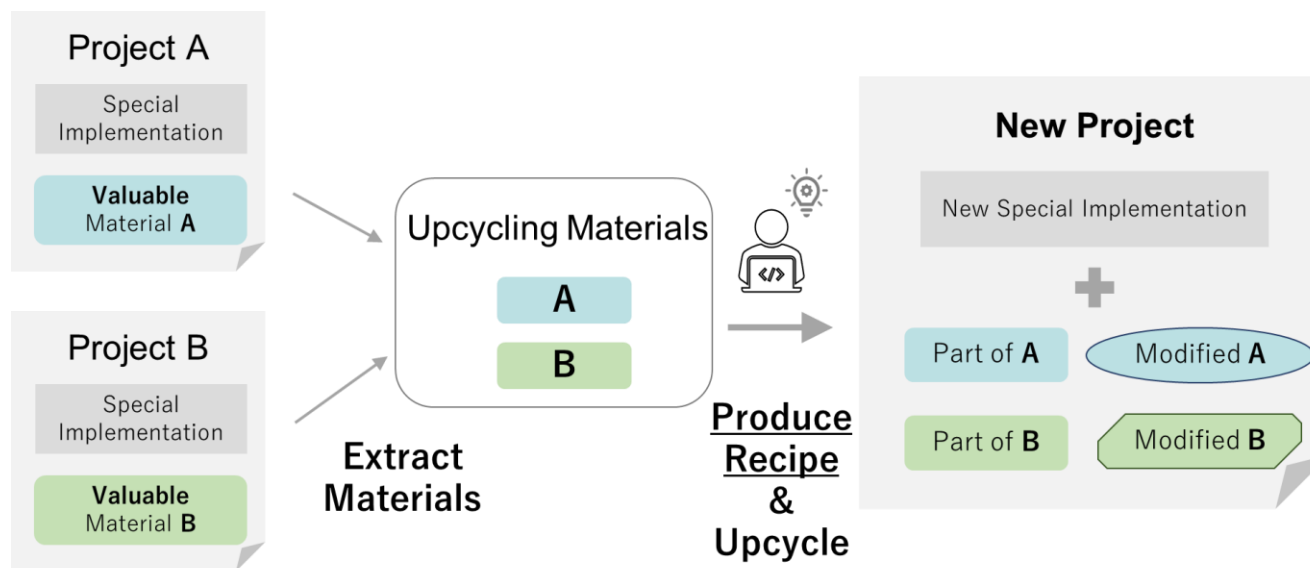
- ◆ 事例の蓄積・活用の自動化



ソフトウェアアップサイクル

- コード・設計・文書等の既存プロジェクトの一部(=素材)を、**新規の価値あるソフトウェア資産**に転換

- ◆ 実装詳細を捨て去り、価値ある動作・設計を保持・利用



- **課題**: 抽出した素材をどう組み合わせるアップサイクルするか
→どのようにアップサイクルのアイデアを生み出すか？



ソフトウェアアップサイクル事例共有システム

■ 目的：アップサイクルを**持続的かつ効率的**に行う

■ キーアイデア

◆ 他のアップサイクル事例を見て、新たなアップサイクルのアイデアを閃き、閃いたアイデアを**実践・共有**し、さらに新たなアイデアが生まれる

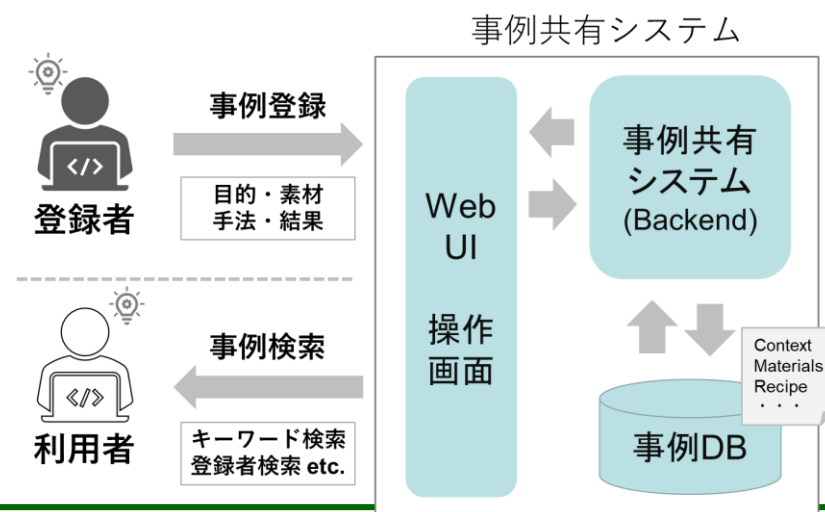
● **アップサイクル事例**：どんな目的のためにどんな素材を使ってどのようにアップサイクルし、どんな結果だったかを文章でまとめたデータ

■ アプローチ：アップサイクル事例を**集合知**として手軽に記録・共有する
SUCCEEDシステムの開発

(Sharing Upcycling Cases with Context and Evaluation for Efficient Software Development)

◆ (A1)事例データモデルの定義

◆ (A2)事例の蓄積・活用の実現





(A1)アップサイクル事例データモデルの設計

■ 事例データモデル

- ◆ **context** : アップサイクルの目的・経緯・課題
- ◆ **materials** : 利用するアップサイクル素材の一覧
 - **detail** : **各アップサイクル素材の詳細を説明した文章**
 - **access** : アップサイクル素材へのアクセス方法(URL等)
 - **version** : 素材のバージョン
 - **discovery** : 素材を発見した手法・経緯
- ◆ **recipe** : アップサイクルのやり方・レシピ
- ◆ **result** : アップサイクルの結果
 - **detail** : 結果の詳細
 - **score** : 結果の点数評価 (1～5点、未実施のアイデアは0点)
- ◆ **creator** : 登録者
- ◆ **createdDate** : 登録日時



アップサイクル事例の例

context	WebSocketを用いたタイマーアプリをSpringBootで開発するにあたって、WebSocket × Spring Bootの導入・実装を参考にしたい	
material 1	detail	過去プロジェクトAのREADMEとソースコード全般
	access	リポジトリのURL
	version	コミットID
	discovery	WebSocketの使い方を先生に相談したら教えてくれた
recipe	READMEを読み、springboot × websocketの実装のために必要なライブラリ(Webjars, sockjs.min.js, stomp.min.jsなど)を知った	
result	detail	stompプロトコルを用いてwebsocket通信を行う方法を知った。これを足がかりに、インターネットでWebJars, JQueryなど必要な技術を調べていった。
	score	4
creator	学生〇〇	
createdDate	2022/10/01 12:00:00	



(A1)事例データモデルの予備的評価

■ システムを用いずにアンケート形式で事例を収集

- ◆ 対象者: 研究室の学生 7 名
- ◆ 手法: 対象者がアップサイクルに準じると考える開発事例を、事例モデルに則った形式で記入してもらう
- ◆ 結果: 事例 8 件、うち素材 10 件が集まった

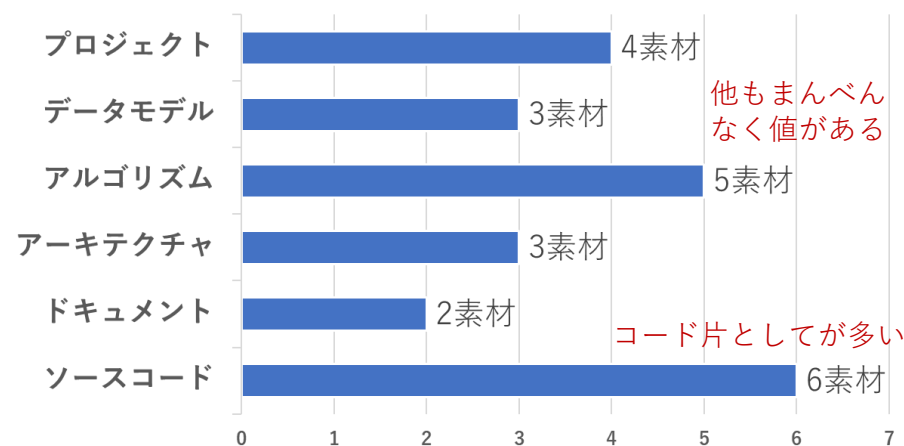
■ 事例のサンプルと分類

WebSocketを用いたタイマーの開発に、Pub/Subサービスのコード・ライブラリ依存を活用(結果:4点)

GASによる予定サービスの開発に、ほぼ同じ機能のJavaサービスからアルゴリズムを再利用(結果:5点)

Pythonでクラスタリング分析を行うために、他の分析のコードを再利用し、パラメータを微修正(結果:5点)

各素材が開発で担った役割を、重複を許してカウントしたグラフ





(A2)事例の蓄積・活用の仕組み

■ SUCCEEDシステムを用いた事例の蓄積と活用の流れ

◆ 蓄積 → 検索 → 閃き → 実践 → 蓄積

■ 蓄積

◆ 開発事例の蓄積:どんな開発を対象とするか?

- 何かしら既存の知見を活用した開発
- 明確にアップサイクルを意識して開発したかどうかは問わない

◆ 開発者が手動で事例を記述する

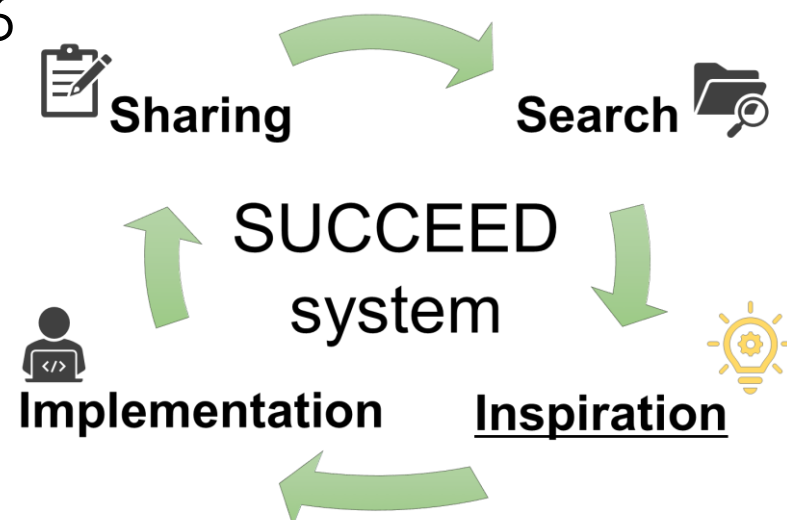
- なぜ、何を使って、どうして、どうだったか

◆ SUCCEEDシステムに事例を登録する

■ 検索

◆ 目的の事例を検索する

- キーワード検索
- 最近の事例、昔の事例





(A2)事例の蓄積・活用の仕組み

■ 閃き

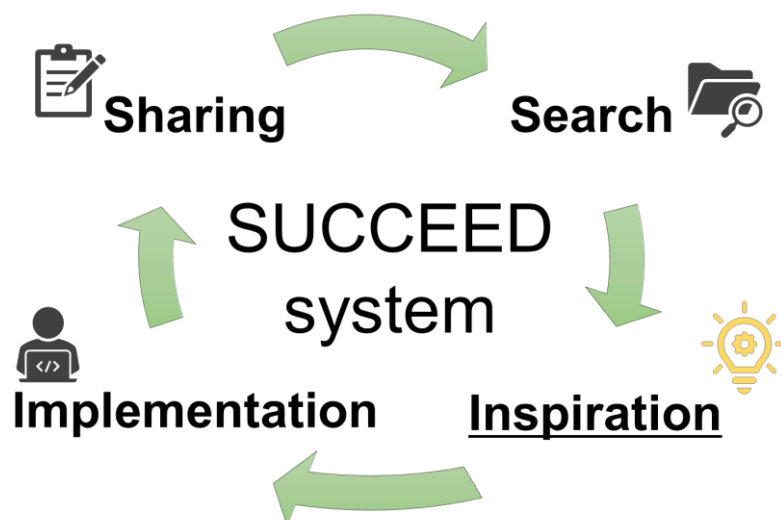
- ◆ 事例の各要素を閲覧し、アップサイクルの手法を閃く
 - context要素から、開発背景が近い事例を見つけ出す
 - materials要素から、アップサイクルに使う素材を得る(URL、リポジトリ)
 - recipe要素から、どのようにアップサイクルするかを知る
 - result要素から、過去の成功・失敗体験を知る

■ 実践

- ◆ 素材を用いたアップサイクルを実践する
- ◆ どのようにアップサイクルを行ったか、結果はどうだったか記録しておく

■ さらに蓄積

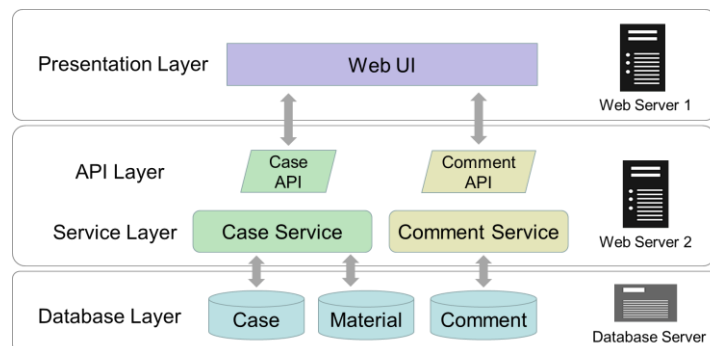
- ◆ SUCCEEDシステムに実践の記録を事例として登録する





(A2)SUCCEEDシステムの実装

- キーワード検索・ハイライト機能
- いいね・コメントによるフィードバック





評価実験の準備

■ RQ.集合知を用いたアップサイクルは開発をどう改善するか？

■ 概要

- ◆ 被験者に異なる2回のアップサイクル開発タスクに取り組んでもらう
- ◆ 開発にSUCCEEDシステムを用いた群と用いない群において、開発の経過や成果物などに差があるかどうか確かめる

■ 被験者

- ◆ 弊研究室の学生・教員のうち、日常的に開発を行うほぼ全員の**11名**
- ◆ **実験群**:タスク中にSUCCEEDシステムで**必ず1回は検索する**
 - 検索結果を活用するかは被験者の自由とする
- ◆ **対象群**:タスク中にSUCCEEDシステムを**使わない** →閃き、実践が変化？
- ◆ どちらの群もタスク中にその他の好きなツールを使える
- ◆ 被験者は半数ずつグループに分け、2回のタスクで群を入れ替える

■ SUCCEEDシステム

- ◆ 事前に事例を24件投入

	タスク1	タスク2
グループ1	実験群	対象群
グループ2	対象群	実験群



評価実験の流れとタスク

■ 開発タスク1

- ◆ 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 (JavaScriptを想定)

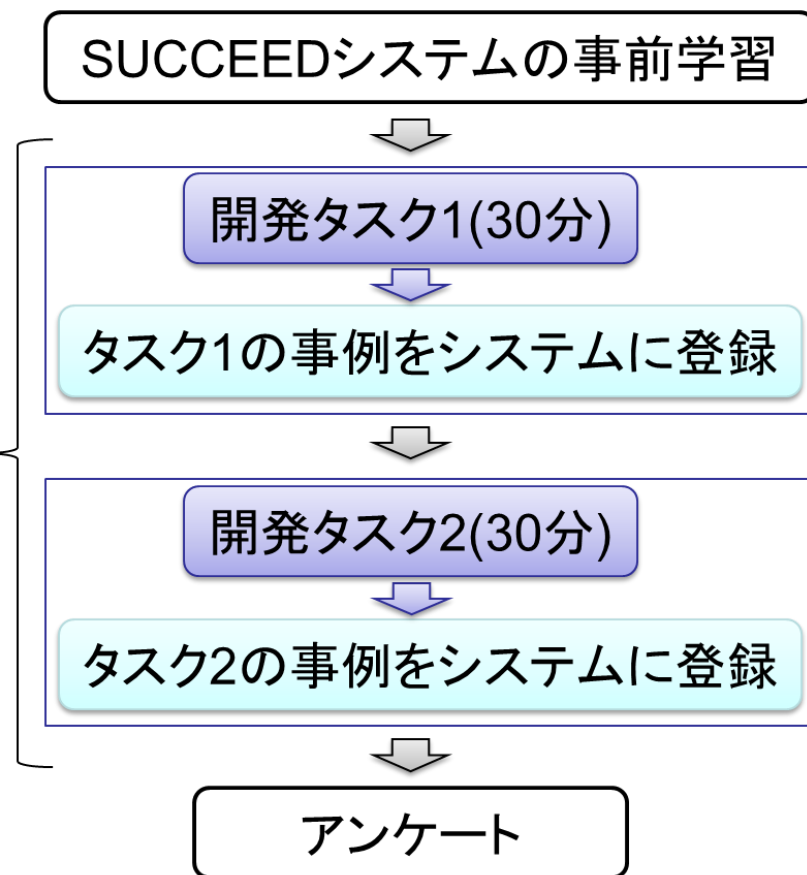
■ 開発タスク2

- ◆ シフト自動作成ツール向けの簡易アルゴリズムの実装 (Pythonを想定)

■ タスク設定の特徴

1. SUCCEEDシステム内の事例に、タスクに役立つが他の検索手法でそれほどの手間なく見つかるか、または必須ではない事例がある
2. アップサイクルタスクである
 - タスク1:レガシーコードの活用、 タスク2:様々なアルゴリズム

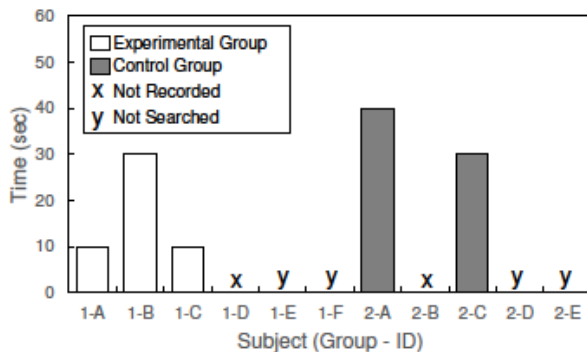
画面
録画



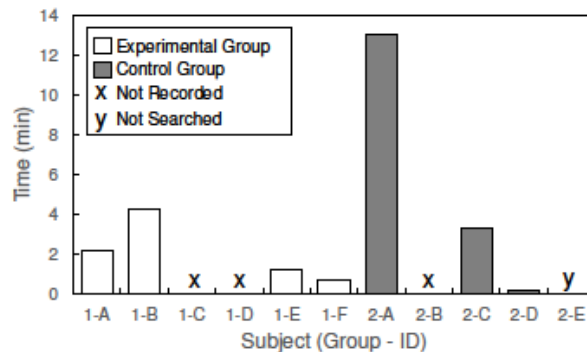


評価実験の時間測定結果

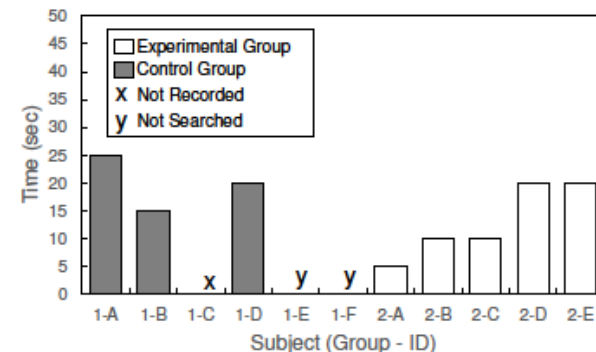
- 画面録画から検索・開発・登録時間を測定
- 知識の検索にかかる時間



エージェント



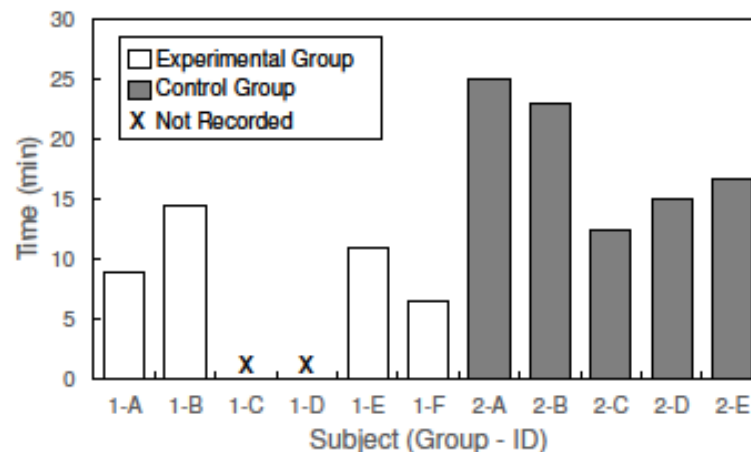
手指検出



シフト生成アルゴリズム

■ 開発にかかる時間

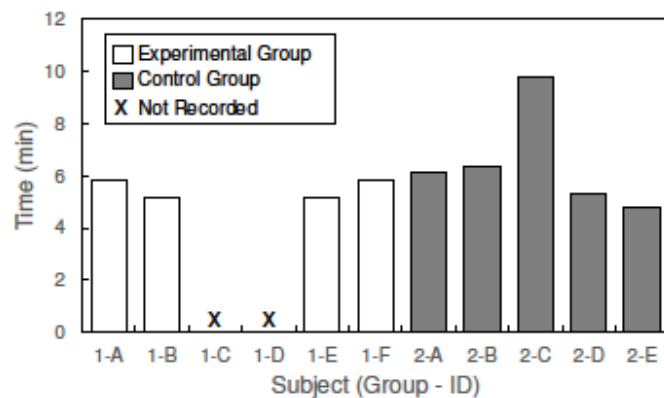
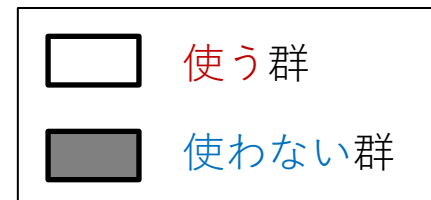
- ◆ タスク1のみ計測に成功
- ◆ タスク2は完答がほぼいない
 - 開発途中の比較は困難



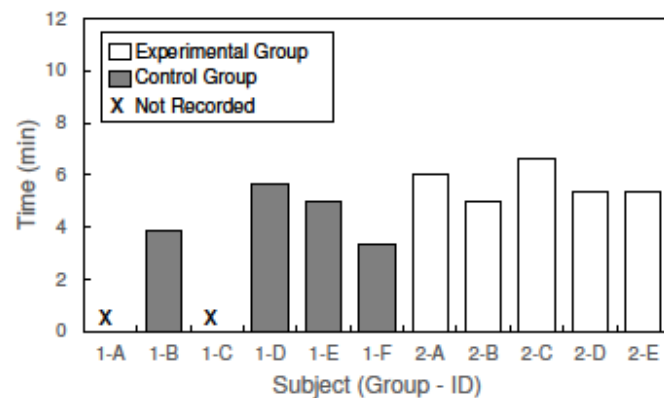


評価実験の時間測定結果

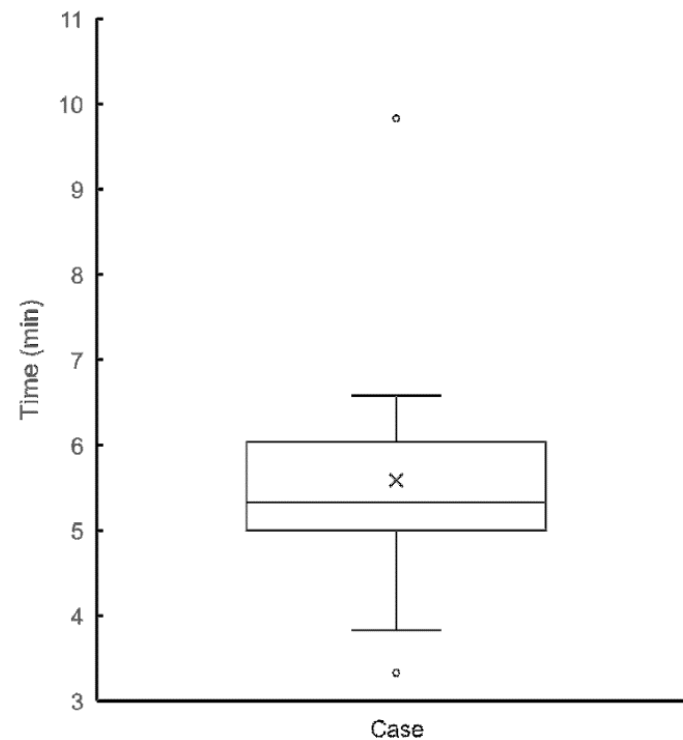
■ 登録にかかる時間



タスク1



タスク2



全タスク



評価実験のアンケート結果

■ タスクに関連する知識を検索できたか

◆ SUCCEEDシステムを使用した場合（対象群は全て未使用）

タスク	成功(人)	失敗(人)	未使用(人)	成功率
1	6	0	5	100%
2	4	1	6	80%
合計	10	1	11	91%

◆ その他の検索システムを使用した場合

タスク	成功(人)	失敗(人)	未使用(人)	成功率
1	7	1	3	88%
2	8	2	1	80%
合計	15	3	4	83%

■ その他の検索システムは何を使ったか

◆ Google検索、ChatGPT、研究室Wiki、研究室リポジトリ



評価実験の考察(検索と登録)

■ 検索機能の有効性

- ◆ SUCCEEDシステムを使ってうまく検索できた人が多かった

■ 検索機能の効率性

- ◆ 目的の知識に到達するまでの時間を比較
- ◆ SUCCEEDシステムの方が他の検索より平均時間は小さい(検定なし)

■ 検索機能の信頼性

- ◆ ほぼ全員が検索結果が信用できると答えた

■ 登録機能の有効性

- ◆ 自然言語による自由記述でも内容の大きな欠落は見られなかった

■ 登録機能の効率性

- ◆ 実験では約5～7分かかった
 - 登録する事例の品質を上げようとするほど登録時間が延びる



評価実験の考察(システム全体)

■ システム全体の有効性

- ◆ 前提1. そもそもタスク中でアップサイクルが行われていたのか?
 - 録画中で検索やChatGPTを通じて過去の知見を得て活用していた
- ◆ 前提2. 被験者が作成したソースコードは正しく動作していたのか?
 - 実装が完了した箇所までは正しく動作していた
- ◆ 実験群と対象群で成果物の品質に違いはあるか?
 - 対象群では、ドメイン知識の検索に失敗して要件を満たさない、Google検索で見つけた手法の活用方法が分からず実装できない、等の例がある
 - 実験群ではこういった失敗例をカバーできていた

■ システム全体の効率性

- ◆ タスク1の実験群の平均開発時間: **10分15秒**
- ◆ タスク1の対象群の平均開発時間: **18分26秒**

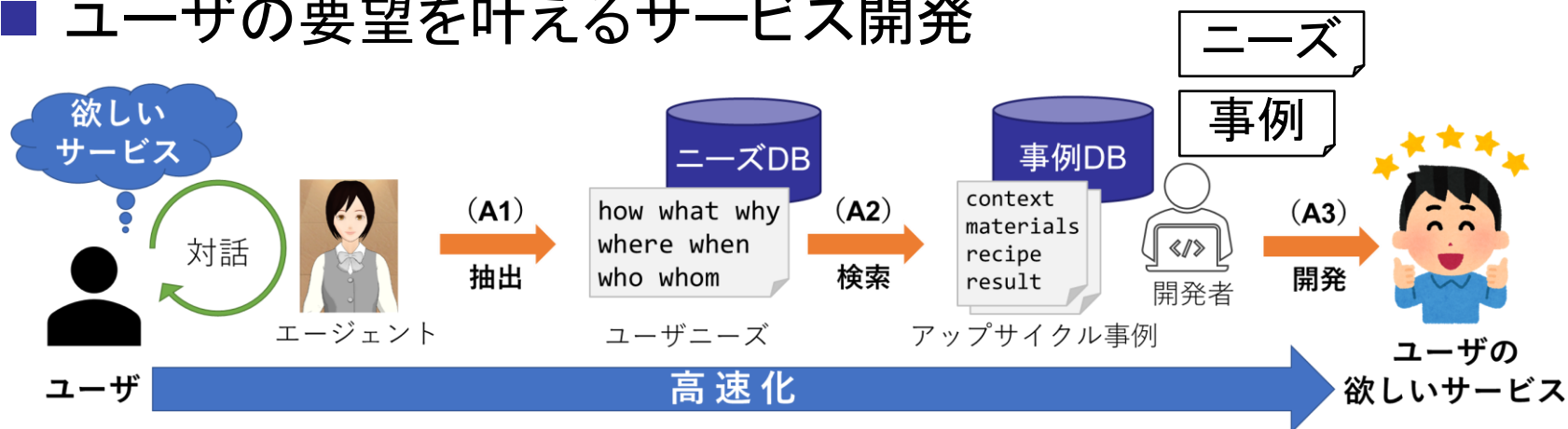
■ 弊研究室ではシステムにより開発の改善がある程度できた

- ◆ 結果を他組織に汎化させることは、環境・被験者の条件が極めて複雑であることから今後の課題となる



今後の展望①

■ ユーザの要望を叶えるサービス開発



■ キーアイデア

- ◆ ユーザが欲しいサービスの実現に役立つアップサイクル事例を発見

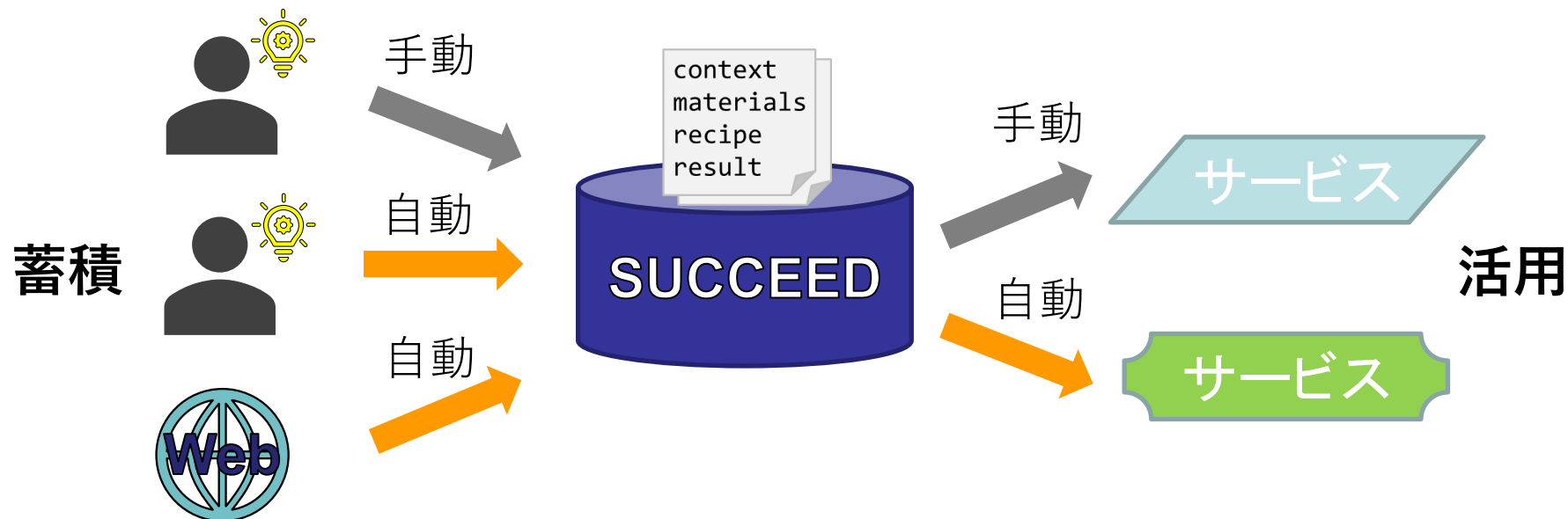
■ アプローチ

- ◆ (A1) ユーザニーズをヴァーチャルエージェントで自動抽出
- ◆ (A2) ユーザニーズをもとに事例を自動取得
- ◆ (A3) ユーザニーズと事例を参考にしてサービス開発



今後の展望②

■ 事例の蓄積・活用の自動化



■ キーアイデア

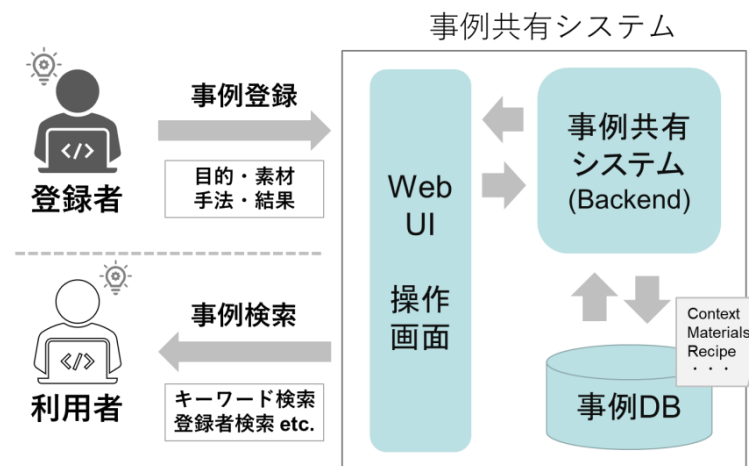
- ◆ LLMを活用し、知見→事例、事例→ソースコード、の変換を行う

■ 展望①と組み合わせ、ユーザニーズの抽出からサービス開発まで自動化できないか？



まとめ

- アップサイクルは、コード・設計・文書等の既存プロジェクトの一部(=素材)を、新規の価値あるソフトウェア資産に転換
- 課題
 - ◆ 抽出した素材をどう組み合わせるアップサイクルするか
- キーアイデア
 - ◆ アップサイクル事例を容易に記録・共有するSUCCEEDシステムを構築することで、アップサイクルの閃きの好循環を狙う
- アプローチ
 - ◆ (A1)事例データモデルの定義
 - ◆ (A2)事例の蓄積・活用の実現
- 弊研究室で実験を行った
- 今後の研究
 - ◆ ユーザニーズと連携
 - ◆ 事例蓄積・活用の自動化





APPENDIX



アップサイクル効率化のための課題

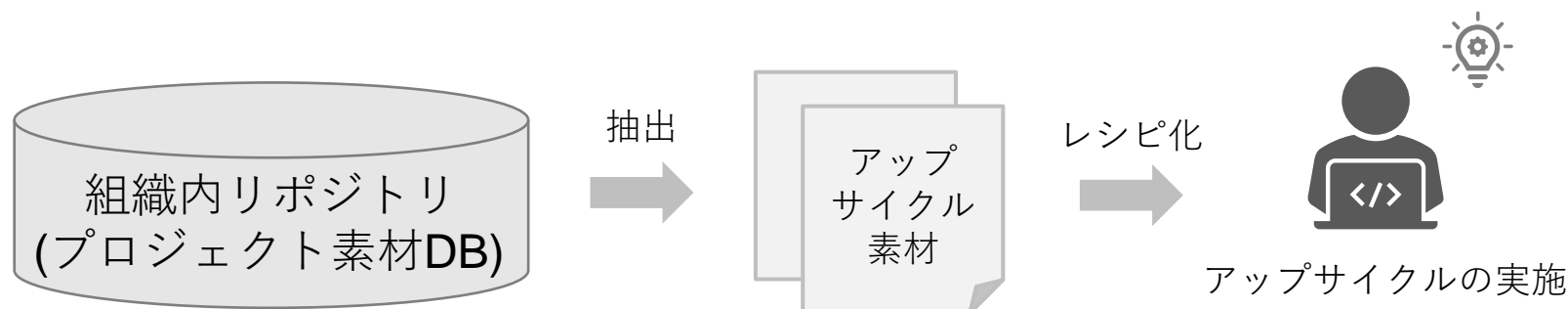
■ アップサイクル素材の資産化・発見支援

- ◆ 組織内リポジトリに蓄積された膨大なプロジェクト素材から、アップサイクルに活用できる素材を抽出
 - この素材を**アップサイクル素材**と呼ぶ
- ◆ 発見支援手法の関連研究
 - プロジェクトコーパス



■ アップサイクル素材の使い方・レシピ

- ◆ プロジェクト素材の使い方を発見し、アップサイクルのレシピを作成
- ◆ 先行研究は存在しない



蓄積するアップサイクル素材の定義

■ アップサイクル素材の定義

◆ どんなプロジェクト素材か

- 素材の説明に加えて、メタ情報として素材へのアクセス方法も必要

◆ 素材はいつのバージョンのものか

- プロジェクト素材は更新されるため、適切なバージョンの記載が必須

◆ 素材をどのように発見したか

- 発見経緯は価値ある知的資産

■ プロジェクト素材の種類例

◆ プロジェクト全体、ソースコードファイル、関数

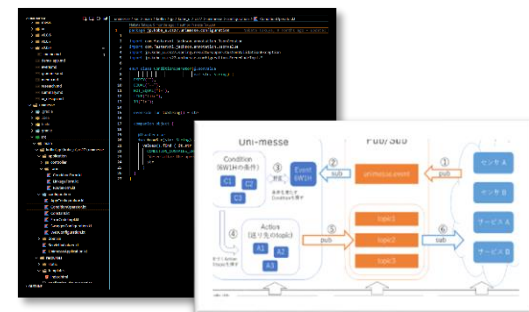
◆ 公開されたAPI、ドキュメント

◆ テストケース、プロジェクト設計ドキュメント (UML・データモデルなど)

■ 素材の発見経緯の例

◆ 過去の開発経験・アップサイクル経験から

◆ 発見手法を用いて (プロジェクトコーパスなど)





アンケート項目

- 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)
- 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)
- 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)
- 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)
- 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)
- 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)