

効率的なソフトウェアアップサイクルのための 事例知識ベースの構築と予備的評価

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

2022/12 第4回ソフトウェアエコシステムワークショップ Session 4-1

アップサイクルとは

■ (広義の)アップサイクル

- ◆ もとの素材の特徴を活かしつつ、別のモノに生まれ変わらせる
- ◆ 例：廃棄されたメガソーラーの再利用

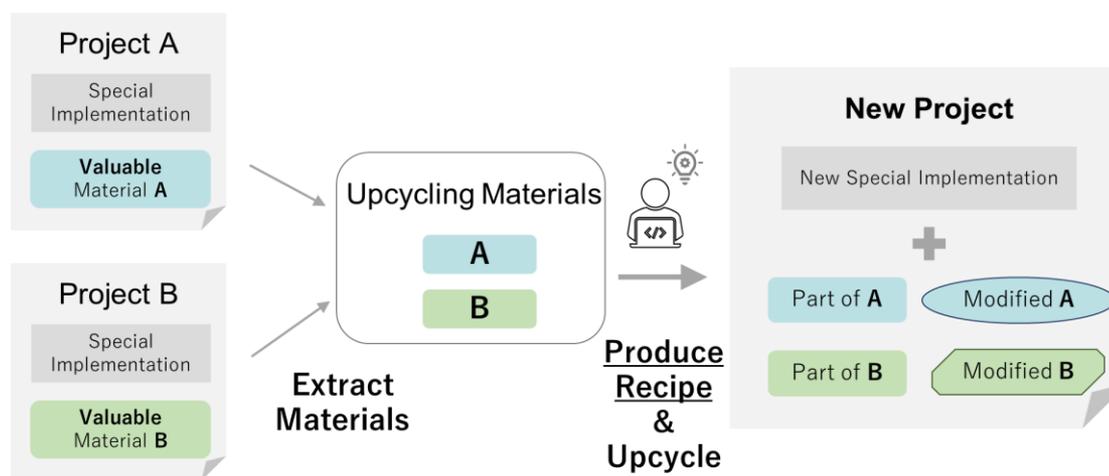
| | リユース | リサイクル | アップサイクル |
|---|--|---|--|
| 方法 | そのまま使う | 資源に分解して使う | 特徴を活かしつつ 手を加えて使う |
| 成果物 | 元のプロダクト | 素材、資源 | 新しいプロダクト |
| 【例】メガソーラー の再利用 | 解体して、家庭用 途にリユース | ガラス、金属に分解し、 リサイクル | おしゃれなテーブルに アップサイクル |
|  |  |  |  |

ソフトウェア開発の文脈で活かせる考え方

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



- アップサイクルにおける2種類のアプローチと効果

- ◆ 【ニーズ指向アプローチ】 再開発コストの低減による開発効率向上
- ◆ 【シーズ指向アプローチ】 未活用の既存技術を活用した価値創造

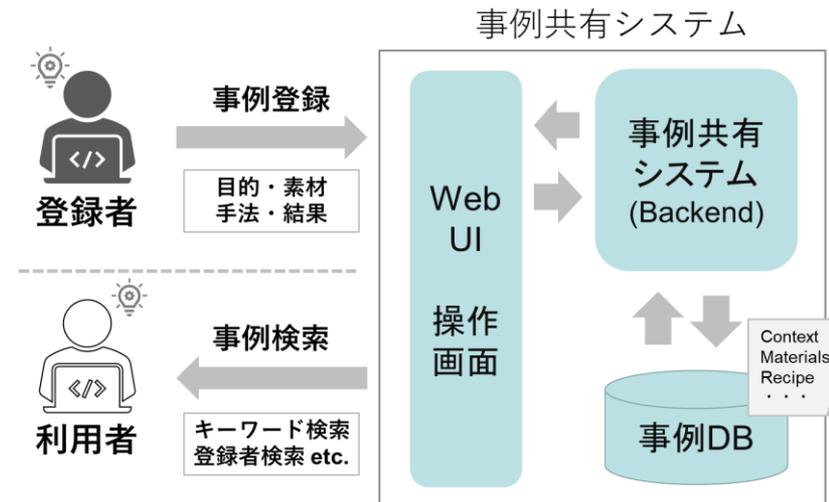
- **課題**: 抽出した素材をどう組み合わせせてアップサイクルするか

目的とアプローチ

- 目的：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、**未来のアップサイクルを効率化する**
 - ◆ **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
- 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
知識ベースシステムを検討する

- ◆ (A1)事例データモデルの設計
- ◆ (A2)事例データモデルの予備的評価
- ◆ (A3)事例共有システムの設計・構築



(A1)アップサイクル事例データモデルの設計

■ 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
 - **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

(A2)事例データモデルの予備的評価

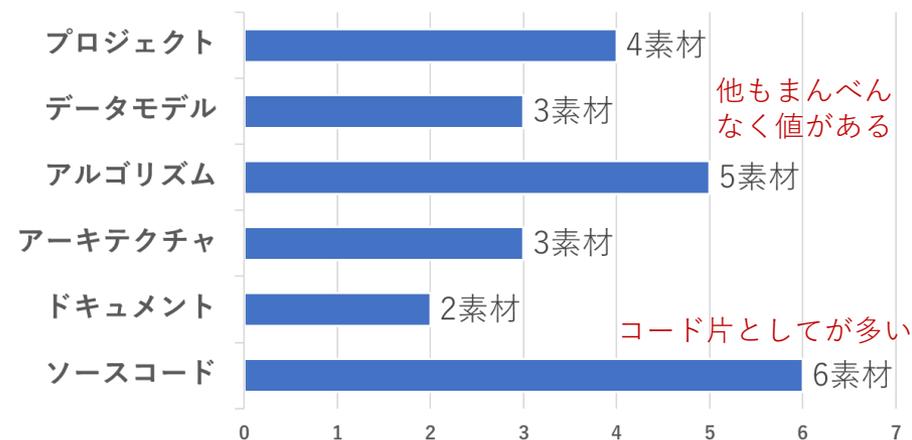
- システムを用いずにアンケート形式で事例を収集
 - ◆ 対象者: 研究室の学生 7 名
 - ◆ 手法: 対象者がアップサイクルに準じると考える開発事例を、事例モデルに則った形式で記入してもらう
- 結果: 事例 8 件、うち素材 10 件が集まった
- 事例のサンプルと分類

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)

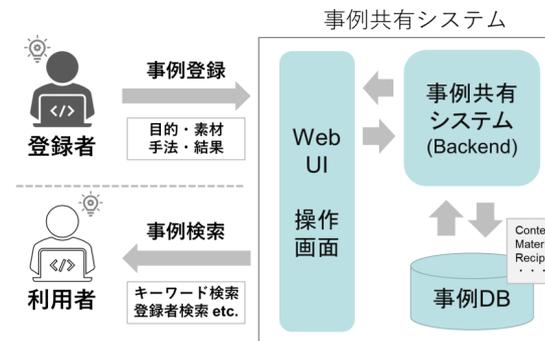
Pythonでクラスタリング分析を行うために、他の分析のコードを再利用し、パラメータを微修正(結果:5点)

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



(A3)事例共有システムの構築

- フロントとしてWeb UIを作成
- UIの特徴
 - ◆ 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



ソフトウェアアップサイクル事例共有システム

JA EN

ID

キーワード

Spring

検索

リセット

#17

Spring Boot × Swagger REST APIの仕様書自動生成 ★★★★☆

by 長谷碧 2022-10-12T16:19:40

コンテキスト KSPのバックエンド開発において、SwaggerというAPI仕様書を作成するツールを使いたい。**Spring** Bootで作られたREST APIでは、自動生成できるらしい。

素材1 中村先生が作成したcs27pubsubのSwaggerのconfigファイル部分

素材2 素材1でswaggerが動かない事例を解決策として挙げられていた記述

レシピ 素材1のSwaggerConfiguration.javaをそのまま移植して、パス等の設定を変更し、application.propertiesに素材2の記述を追加した。

結果詳細 素材1のSwaggerのconfigファイルの追加だけでは、バージョンが新しい**Spring** Bootが使えないため、少し追加調査が必要だった。

0 🗨️

#9

WebSocket x **Spring** Boot ★★★★☆

by 寺川航平 2022-10-31T17:21:48

コンテキスト WebSocketを用いたタイマーアプリを**Spring** Boot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

素材1 中村先生の作成した、CS27PubSubのREADME、ソースコード全般。

レシピ 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

結果詳細 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。

タイトル

コンテキスト

素材リスト

> 素材1 🗑️

+

レシピ

結果詳細

登録者

★★★★☆

登録

リセット

考察

- (A1)提案した事例データモデルの特徴
 - ◆ 素材を文章で説明することで、コード片のみならず図なども蓄積可能
 - ◆ まだ実行・試していないアイデアも投稿できる
- (A2)事例データモデルの予備的評価の結果
 - ◆ 素材のバージョン情報の欠落が多い
 - ◆ 事例の自己評価が「良い」が多い
 - ◆ コード再利用以外の素材活用も幅広く見られた
- (A3)事例共有システムの設計・構築
 - ◆ 開発スキルの習熟度に寄らず、あらゆる人がカジュアルに投稿できる
 - 集合知の形成
 - 精度の低い情報が混ざる危険
 - ◆ 利用頻度をいかに高めていくかの工夫が必要

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・共有する知識ベースを構築することで、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設計
 - ◆ (A2)事例データモデルの予備的評価
 - ◆ (A3)事例共有システムの設計・構築
- 今後の研究
 - ◆ システムを活用した本実験
 - ◆ 蓄積した事例を活用した分析

