

オープンソース ERP パッケージに対する派生開発手法の提案

—開発プロジェクトの事例をもとに—

岸麻美 生井雅志 土山真由美

住所 〒221-0031 神奈川県横浜市神奈川区新浦島町1丁目1番25号テクノウェイブ100ビル7F

E-mail: asamim@aicom.co.jp

概要 当社は、システム企画から保守までのサービスを提供している。近年の顧客の要求を実現するための開発では、要求やスコープが曖昧で、短納期の開発も多い。そこで、品質管理と納期、コストを満たすために、オープンソースの ERP パッケージを用いたシステム導入を実施している。本報告は、オープンソースの ERP パッケージを利用した派生開発において、開発や管理手法を工夫することで、技術的なリスクを回避し、短納期で一定の品質を担保した開発を実現した事例を紹介するものである。

Proposal of derived development method for open source ERP

—Case of development project—

Asami Kishi, Masashi ikui, Mayumi Tsuchiyama

† HAMAGOMU AICOM INC.

Yokohama City 1 - 25 Shin-urashima - cho Kanagawa - ku Techno Wave 100 Building 7 F

E-mail: asamim@aicom.co.jp

Abstract We provide services from system planning to maintenance. Derived development to realize customer's additional request often has vague requirements and scope. In addition, there are many developments with short delivery times, it is also necessary to comply with delivery deadline compliance and quality collateral. In this research, in the derivative development using open source ERP, we introduce the case where the development was realized by maintaining the quality by effectively using the requirement definition method and the open source project management tool and development agreement is there.

Keyword open source、ERP、iDempiere

1. はじめに

1.1. 本事例の概要

本事例は、オープンソースソフトウェア(以下、OSSと記す)の ERP パッケージである iDempiere の派生開発を XDDP と対比することで、派生開発のポイントを提案するものである。変化の大きな市場環境の中で、各企業がシステム構築に OSS の ERP パッケージを採用する一助となることを期待している。

OSS は、OSS のコミュニティによって機能開発されている。本事例では、これを標準機能と称する。OSS の派生開発は、標準機能を最大限利用することが品質や納期を担保することにつながる。しかしながら、ソースが公開されているため

に、標準機能にも様々な変更を加えることも可能である。また、変更を加えると OSS のコミュニティが担保している品質を阻害する可能性もある。この問題に対し、iDempiere の教育、開発規約の整備、OSS を研究する組織との連携によって、課題解決した当社のプロジェクト事例を紹介する。

1.2. 当社の事業と取り組み

当社は、1970年に設立された横浜ゴム株式会社(以下、横浜ゴムと記す)の100%子会社である。システム企画から保守運用までの役務を提供している。顧客は横浜ゴム、横浜ゴム関連会社、一般企業と多岐にわたる。

横浜ゴム向けのシステム開発では、スクラッチ

開発をメインに取り組んできたが、2000年度よりERPパッケージであるSAP、近年はMicrosoft Dynamicsなど商用パッケージも利用している。また、2012年度より、OSSのERPパッケージであるADempiereを利用したシステム導入を開始した。さらに、2016年度より、様々なOSSを利用できるようにOSSの研究チームを設立した。

1.3. OSSを取り巻く状況

近年のビジネス環境の変化に対し、企業のニーズを満たす上でもOSSが注目されている^[1]。JISAも業務アプリ領域、共通アプリ領域、ミドルウェア領域の様々な分野にOSSの利用が進んでいると報告している^[2]。

利用が進んだ背景として、スクラッチ開発では実現できないシステム構築のスピードと、商用パッケージでは実現できない派生開発の自由度が考えられる。なお、商用パッケージでも派生開発に自由度が許諾されている場合もあるが、製品に準じた言語を習得する必要があり、システム導入外の費用と時間が発生するなどの課題がある。

なお、本事例における派生開発は表1のように定義する。

表1 本事例の派生開発の定義^[3]

種類	定義	本事例での定義
新規開発	要求仕様に書かれていることを実現すること	OSSコミュニティ側から提供される新機能 本事例では、標準機能として記述、パラメータ変更によるコーディングなしの設定変更が可能
派生開発	機能追加	OSSコミュニティが開発した機能とは独立した追加開発機能とする
	変更	OSSコミュニティが開発した機能に変更を伴う追加開発とする

1.4. iDempiereとは

iDempiereは、購買管理、在庫管理、販売管理、

生産管理、会計管理の機能を有し、多言語、多通貨に対応しているOSSのERPパッケージである。アメリカのComPiere社が提供しているOSSのERPであるCompiereをベースとし、ADempiere、iDempiereとして発展してきた。OSはLinuxで、データベースはOracleとPostgreSQLに対応している。当社は費用面を考慮し、すべての構成をOSSで構築する提案をしている。開発言語はJavaで、Web系とクライアント系の2種類のユーザーインターフェース（以下、UIと記す）が構築されている。メニュー体系や画面項目の変更、権限設定等はパラメータ設定の変更で対応することができる。

2. 開発事例概要

2.1. プロジェクトの目的

本事例では、横浜ゴムの働き方改革の一環である個人業務の可視化プロジェクトを事例として取り上げる。横浜ゴムでは2030年でも通用する働き方の検討を行っている。1つの施策として、売上に直結する成果だけでなく、他者や他部門への業務支援に対して個人の成果や貢献を可視化する仕組みを構築することとした。成果を可視化することで、無駄な作業の排除や社員のモチベーション向上につながる施策をとることを目的としていた。

2.2. プロジェクトの課題

個人業務の可視化プロジェクトは、既存システムの置き換えではない。さらに一部の部門で試行し、利用に即した形でシステムを発展させていくという試験的な要素を持っていた。したがって、要件は整理されていないが、動く仕組みを素早く提供する必要があった。また、個人業務を可視化することに付加価値があるわけではなく、得られる情報の活用が付加価値を持つことから、安価にシステム構築することも前提としていた。

さらに、システム利用にあたって、個人及び組織の負荷が上がることは望ましくない。したがって、簡単な操作性も求められていた。

2.3. 開発方法の選定

プロジェクトを実現するにあたり、以下の開発

方法を選択肢とした。

- ① 新規スクラッチ開発
- ② 商用の ERP パッケージの利用
- ③ OSS の ERP パッケージを利用した派生開発
- ④ 既存システムを流用した派生開発

4つの方法を、費用、品質、要求の実現性の観点で比較検討した。個人業務の可視化のため、業務のやり取りを受発注として、販売管理や購買管理の機能をもつシステムを利用する方針とした。さらに操作性の改善のために、派生開発ができることを評価し、③OSSのERPパッケージを利用した派生開発、すなわち iDempiere を利用した派生開発を採用した。

2.4. 開発のアプローチ

iDempiere を利用したシステム導入方法は、以下の手法をとる。

- ① 標準機能の利用判断
- ② 業務の実現方法の検討
- ③ 派生開発の設計
- ④ 派生開発の開発
- ⑤ 派生開発のテスト

はじめに、業務フローを用いて、標準機能を利用できる業務と、標準機能を利用できない業務に仕分ける。次に、標準機能を利用できない業務の実現方法について、業務を ERP パッケージに合わせるのか、業務に合う機能を派生開発するのかを検討する。そして、派生開発の対象となった業務の要件を元に、派生開発の設計、開発、テストを行う。なお、標準機能はパラメータ設定によるコーディングなしの機能利用変更ができる。また、派生開発の設計では、機能追加を行うのか、機能に変更を加えるのかの判断も発生する。

3. 課題

3.1. 標準機能の利用判断

2.4 の開発アプローチで示した①標準機能の利用判断、③派生開発の設計を行うには、ERP パッケージのもつ標準機能を詳細に理解する必要がある。商用の ERP パッケージの場合、研修、開発者向けの技術情報の提供、セミナー等の教育環境が整っている。また、PostgreSQL など普及が進んでいる OSS の場合、研修やセミナー等、教育機関が整っている。さらに、知識量を確認する指標となる資格制度もある。しかしながら、普及が充分ではない iDempiere は、開発コミュニティ

や自主的な勉強方法はあるが、教育機関は充分ではなく、資格制度はない。したがって、iDempiere の機能詳細、つまりは標準機能の知識を習得することが難しい。

3.2. 派生開発の設計

派生開発の設計において、機能追加を行うのか、機能に変更を加えるのかは、OSS のコミュニティが担保している品質を享受する上で重要な判断となる。設計方法によっては、OSS のコミュニティが提供する追加機能や不具合対応を享受できない。

商用の ERP パッケージには、標準機能と派生開発の区別、ひいては派生開発の機能追加と変更の区別が明確になるように、開発者の利用者 ID による権限制御がされている製品もある。開発者が変更できる範囲が明確であれば、派生開発の自由度は制限されるが、一定の品質を担保できる。しかし、iDempiere では、製品の内部的な仕様や構造を正しく理解しなければ、適切な派生開発の設計が行えない。

4. 施策

4.1. iDempiere 教育

iDempiere では、一般的な研修や教育機関が充実していない。したがって、標準機能の利用判断や派生開発の設計の知識の習得は利用者に委ねられる。

当社では、2012 年に横浜ゴムグループで iDempiere の前進である ADempiere のシステム導入を行った経験をもとに、独自の教育テキストを作成した。さらに、2016 年から、iDempiere のシステム導入を担うにあたり、教育テキストを改定した。また、iDempiere の開発コミュニティの情報で有用性が確認できた情報を社内で共有する仕組みを構築しており、プロジェクト参加者に対して、これらを活用した社内教育を行っている。

4.2. 開発規約

当社では、スクラッチ開発の開発規約と同様に、OSS の ERP パッケージについても、開発者のスキルによって品質や開発内容にばらつきが出ないように、開発規約を設けている。さらに、2016 年度に、研究チームにより OSS で提供されている標準機能に影響を与えない設計基準をまとめ

た開発規約が構築された。開発規約は、本来、命名規約やコーディング規約といった開発の可用性や保守性を用意にするものである。しかし、iDempiereの開発規約では、プログラム構造に関するガイドラインも整備している。

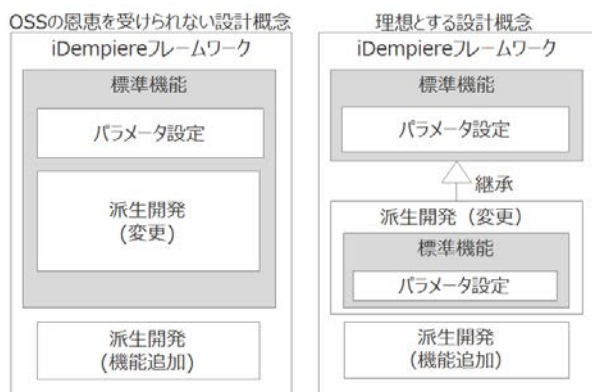


図1 設計開発の概念

2012年当時、iDempiereの前進であるADempiereのシステム導入を開始した直後は、図1に示すようなOSSの恩恵を享受できない設計が多数あった。結果として、OSSコミュニティが提示した不具合対応においても、個々にプログラムを変更し適応するなど追加コストが発生した。開発規約は、この経験を活かし、派生開発の設計を支援する施策として構築した。

4.3. 開発体制

標準機能の利用判断、派生開発の設計の多くは、4.1や4.2に示すような施策で補完できると考えていた。しかし、教育テキスト、情報共有の仕組み、開発規約は過去の経験をもとにした施策であり、すべての標準機能の利用判断、派生開発の設計の知識を補完できるわけではない。そこでプロジェクト体制に、図2に示すような研究チームのサポートを組み込んだ。研究チームはプロジェクト専任での配置ではなく、知識や判断の補完を目的とした。また、一部機能については、研究チームをレビューとして、標準機能の利用方法や派生開発設計の妥当性を客観的に評価することとした。

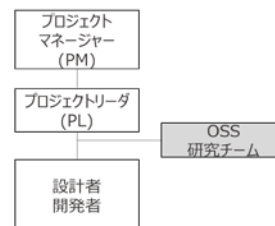


図2 体制イメージ

5. 試行

5.1. iDempiere 教育

プロジェクト参画者のうち、開発者以外は全員iDempiereのシステム導入は初めてであった。参画者の経験とiDempiereの教育に要した日数を、表2に示す。

表2 プロジェクト担当者の経験と教育

役割	経験			教育日数
	他 ERP	他 OSS	JAVA	
PM	有	有	有	1
PL	無	無	有	4
設計者 1	有	有	有	2
設計者 2	有	有	有	1

iDempiereの機能や概念は教育テキストで理解できた。しかし、実際の業務と機能の利用判断はテキスト教育だけでは不十分であった。

一例として、プロジェクトでは、個人間の業務支援を受発注と捉え、販売管理、購買管理機能を利用する方針とした。しかし、業務支援者が作業を受注、被支援者が作業を発注する場合、作業支援の業務可視化に、2回以上の操作が必要となる。実際には口頭で業務支援の契約は成立しており、受発注は1回の操作で完了したいという要望があった。教育テキストの知識だけでは、この問題を解決することができなかった。また、社内の情報共有の仕組みに必要な情報が不足しており、結果としてOSSのコミュニティを個別に調査することもあった。

5.2. 開発規約

開発規約はプロジェクト参加メンバーに周知した。しかし、過去にシステム導入経験の実績がない機能の開発では、派生開発の機能追加と変更の判断ができなかった。

例えば、今回の開発では、月間集計した個人の売買情報をファイルで添付して関係者にメール

で通知したいという要求があった。iDempiere の標準機能では、パラメータ設定にすることで、メールを通知する機能が存在していたが、ファイルを添付する機能が存在しなかった。そこで、添付ファイル付きのメール通知ができる機能を派生開発することとした。本来、設計開発には3パターンがあった。

- ① 派生開発(変更)として、標準機能をすべて直接改修する。
- ② 派生開発(変更)として、標準機能を生かし、メール通知処理を変更開発する。
- ③ 派生開発(機能追加)として、メール通知処理を個別に追加開発する。

①は開発規約違反となり、OSS コミュニティの対応を享受できない可能性がある。②、③の場合、派生開発による標準機能への影響は限定的になる。プロジェクトでは、③を採用した。プロジェクトの振り返りにおいて、機能の汎用性から OSS コミュニティに知見として提示すべきであった。

5.3. 開発体制

研究チームの支援により、情報がうまく共有できた例として、マスタの利用方法の判断があげられる。

プロジェクトでは、個人間の様々な業務を取引として実現するため、取引先マスタや品目マスタの利用を検討していた。プロジェクトの設計者は、標準の品目マスタで業務を管理する方針とし、不要な項目は削除し、不足項目を派生開発の変更で対応する案を考えた。しかし、マスタの中でも、品目マスタは必ず利用するマスタであり、派生開発による変更は様々なプログラムへの影響が想定された。そこで、研究チームと利用方法について協議を重ねた。結果、当初想定していた設計では、予定しない組織設定などが必要になることから、設計方法を変更することとした。設計者が気になる事項について、チャットやメールなども含め、プロジェクト期間中、研究チームと確認を行っていた。

6. 結果

6.1. iDempiere 教育

iDempiere 教育、情報共有の仕組みにより、品質管理への一定の貢献はできた。ただし、定量的に予測できるコスト削減効果としては教育コストの方が高いという結果であった。

単体テスト工程の障害発生率について、過去の OSS 開発から算出した平均値は 10.5%であった。一方、本事例では 9.0%でありバグ発生数が抑えられた。障害発生率の低下により抑えられた障害件数は 3 件と想定される。障害 1 件あたりの対応工数はプロジェクト平均で 6 時間であったことから、単体テストの工数を 2.25 人日削減することができた。結合テストやシステムテストでの障害発生による手戻り工数の削減を考えればこの倍の効果は想定される。定量的に予測される削減工数以外にも副次的効果として、設計や開発の生産性が高くなったことは考えられる。

しかし、以降のプロジェクトではメンバ全員が iDempiere の未経験者で、教育工数がプロジェクトメンバの人数に比例して増加することは極めて少ないと考えられる。したがって、今後のプロジェクトでも数値を取得し、継続的な効果を測定していく。

また、技術情報を共有する仕組みについても同様であり、経験プロジェクトが増えることで、知識や知見を増やすことができる。同様の効果測定を今後も続けていく。

6.2. 開発規約

開発規約については、派生開発の機能追加と変更の選定のすべてを記録できないことから、定量的な結果判定が難しい。品質面での効果があったという設計者、開発者の意見はあった。しかしながら、本事例で着目する派生開発の機能追加と変更の判断においては、開発規約の情報不足が確認できた。

仮に、5.3 で示した品目マスタの変更判断などが開発規約として記載されていたとする。プロジェクトの全 17 機能の開発に品目マスタの影響があったと考えた場合、各機能で品目マスタに関するテスト項目が 1 件ずつ削減されると仮定する。1 機能あたりの平均テスト時間が 8.2 時間、平均テスト件数 14 件より、1 件あたりのテスト時間が 0.6 時間となり、17 機能で約 10 時間、つまり 1.25 人日のコスト削減ができる。したがって、総合的にはコスト、納期へ貢献できる。

ただし、開発規約と照らし合わせながら設計することは設計者の負担となる。今後、商用パッケージのように、派生開発の機能追加と変更の判断が明確にできるような開発フレームワークの検討が必要である。

6.3. 開発体制

本事例でもっとも効果があった施策は研究チームのサポートであった。数値に現れないが、細かな確認も含め、知識が集約されていることで、設計者、開発者のプロジェクト推進の安心感を構築できた。

結果として、障害発生率の抑止や顧客ニーズへの応答、さらに納期への貢献ができた。

しかし、開発体制に研究チームを補完することは常時対応できる施策ではない。研究チームの目的は、iDempiere も含む OSS 全般の調査研究である。本事例では、研究チームのサポート実績は会議で 10 時間であった。チャットやメールによるサポートもあったが、これについては埋没している。

本事例では成功要因ではあるが、6.1、6.2 で示した予防効果と比較すると、開発規模によって費用対効果は充分ではない可能性が示唆される。

7. 考察

7.1. 結果のまとめ

本事例では、OSS の ERP パッケージの派生開発において、3 つの施策を試行した。

- ① iDempiere 教育
- ② 開発規約
- ③ 開発体制

結果として、顧客の望む品質・納期・コストを満たす開発ができた。

iDempiere 教育では、ERP パッケージ全体の知識向上により、派生開発を減らす提案を行った。また、プロジェクトメンバ全員の知識が向上することで派生開発の品質レベルを向上させた。開発規約では、派生開発の機能追加と変更の判断を行い、OSS コミュニティの恩恵を受ける開発を一部実現できた。開発体制では、開発規約で補助できない派生開発の機能追加と変更の判断を補助し、プロジェクト全体の品質・納期・コストに貢献した。いずれの工夫も品質、納期、費用に貢献したといえる。

しかし、iDempiere の教育、情報共有の仕組み、開発規約において、経験のない OSS の機能をどのように取り込んでいくか、開発体制については、支援できる規模をどのように判断するかが課題である。

7.2. XDDP との対比

本事例では、プロジェクト当初から XDDP と照らし合わせた推進をしていない。プロジェクト完了後、7.1 の課題を解決する手段としてプロジェクトの工夫と XDDP との対比を行った^[4]。本事例の開発プロセスを XDDP における派生開発の変更パターンとの PFD と比較した結果を図 3 に示す。

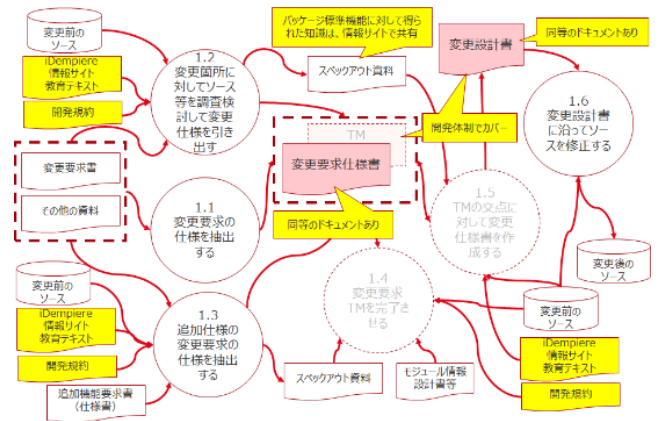


図 3 PFD の比較^[5]

本事例の開発プロセスは概ね XDDP の PFD と一致していた。

施策①の iDempiere の教育によって教育テキストや情報共有サイトを作成したことにより、変更前のソースから得られる情報の精度を向上させることに効果があった。また、施策②の開発規約をプログラムの構造レベルで整備したことで、変更仕様の検討や、変更設計書を作成するときに ERP パッケージの持つ品質を担保する変更方法の検討がしやすくなった。

しかしながら、XDDP の主要成果物に着目すると、一部本事例で補完できない成果物が存在する。

変更設計書は本事例で作成した設計書も差分のみの記載しており、記載内容も XDDP で提唱されている項目を網羅している。また、変更要求仕様書に当たるドキュメントは、2.4 で示した機能利用判断において、標準機能をそのまま活用する部分、派生開発で対応する部分を明確にする過程でアウトプットする成果物に類似していた。USD M の表記方法には則っていないものの、変更要求とその理由、変更仕様が機能ごとに整理している。

補完できない成果物はトレーサビリティ・マトリックス(以下、TM と記す)である。iDempiere は ERP パッケージであると同時に、開発のフレームワークの側面も持っており、表 3 に示したように、開発で実装種類は定義されている。したが

って、変更仕様と機能種別を整理すれば、フレームワークの概念に合わせてプログラムの構成や変更方法の大枠は固まる。

表 3 iDempiere の主な機能種別

機能種別	概要
標準画面	データベースの 1 テーブルと対になる編集画面を生成する機能 主にパッケージの設定で生成が可能。項目の配置、レイアウトは制限される
パネル・フォーム	データベースの複数のテーブルを集約して 1 画面に表示する、独自のレイアウトで一覧を表示するなど、テーブルとレイアウトを自由に設計してプログラミングできる機能
プロセス	オンラインでバッチ処理を実装する機能
インポート	データベースの 1 つのテーブルと対になって、CSV ファイルなどの外部ファイルをインポートして一括でデータを登録する機能

機能種別ごとにクラス構成を図式化することで、変更対象のクラスを類推できる。さらに変更仕様と照らし合わせることで、具体的にどの機能種別に変更を加えるべきなのかを判断できる。しかし、変更による影響度を判断することはできない。本事例では、TM に変わるものとして、②開発規約③開発体制を施策した。結果として効果は獲られたものの、6.3 で述べたように、開発体制については機能強化チームの負荷や、組織として対応できないケースなど課題も残る。

この課題に対応する 1 つの案として、XDDP の TM の概念を取り入れるのは効果的と推察される。TM の横軸に、iDempiere の派生開発で検討しなければいけない要素を配置することで、変更仕様の内容によって、ある程度交点になる組み合わせがパターン化されるものと推測される。TM の有用性を検討するために、本事例で紹介したプロジェクトの一部の機能について TM を整理し、その有用性について机上検証した。

結果、TM の横軸にパラメータ設定の変更、ロジックの変更の切り口で変更要素を配置することで、変更仕様の種類によって、パラメータ設定だけで対応できるものと、ロジックの変更が必要なものが整理された。TM を作成することで、今まで暗黙値として認識されていた iDempiere の開発ノウハウが視覚的にわかる状態で整理できた。TM を作成することは、変更仕様に対する変

更箇所の特定だけではなく、iDempiere のノウハウの整理にも有効であった。現時点ではデータが少ないため、変更内容と TM のパターンの傾向や、見積りや変更仕様書の記載内容との連動についてはさらなるデータの蓄積と分析が必要である。

以上のことから、今後、OSS の ERP パッケージである iDempiere でも XDDP を利用することでプロジェクトの品質納期コストに貢献できると考えられる。

8. まとめ

本事例は、当社の OSS の ERP パッケージである iDempiere に対する派生開発の取り組みとして、教育、情報共有の仕組み、開発規約の整備、体制強化による短納期で品質を確保した開発の事例を紹介した。また、XDDP とこれらの施策の関係を整理することで、その効果ポイントと更なる改善の可能性について考察し、TM の概念を取り入れることで更なる改善が見込まれると結論付けた。

近年、OSS を利用した開発は増えており、今後もその潮流は続くものと思われる。OSS を利用するデメリットを抑え、どのようにメリットを享受するか、この鍵は派生開発のプロセスを整理することにあると考えている。

本事例は 1 事例であり、現段階では普遍的な利用に貢献できる情報ではない。しかし、XDDP を用いて、OSS の ERP パッケージ導入の実績を積み上げ、派生開発のポイントを明確にしていく。さらに、変化の大きな市場環境の中で、各企業がシステム構築に OSS の ERP パッケージを採用する一助となるよう今後も情報発信していく。

文 献

- [1] 論説委員 奥平和行, ものづくりを変えた「オープン」, 日経新聞 電子版, 2018/02/01
<https://www.nikkei.com/article/DGXMZ026372310R30C18A1TCR000/>, 2018/03/31
- [2] 一般社団法人情報サービス産業協会, デジタルビジネスへの挑戦 情報サービス産業白書 2017, 株式会社インプレス, 2017
- [3] 清水吉男, 派生開発 XDDP-なぜ特化したプロセスが必要なのか?,
<https://www.ipa.go.jp/files/000049830.pdf>, 派生開発推進協議会(AFFORDD), 2015
- [4] 清水吉男, 「派生開発」を成功させるプロセス改善の技術と極意, 株式会社技術評論社, 2007
- [5] オープンソース ERP システム iDempiere(アイデンピエレ) Lab
<https://www.compiere-distribution-lab.net/idempiere-lab/>, 2018/03/31