特別講義録フォーマット

第 7 回特別講義 レポート

日時	2021 年 11 月 12 日（金）10:00～12:00
実施形態	オンライン（Zoom）開催
テーマ	プロダクトライン開発の考え方 – 共通性を確立して可変性を確保する
講師名・所属	林 好 一 氏 Y's Workshop 代表 兼 ソフトウェアプロセスエキスパート
司会	岩井 慎一 氏（株式会社デンソー／本研究会 基礎コース 主査）
アジェンダ	● PLE を志向する動機 ● 共通性と可変性 – 複数システムを意識した仕様 ● フィーチャモデル – 共通性と可変性の表現、利用 ● PLE のプロセスと体制 ● PLE にまつわる神話
アブストラクト	プロダクトライン開発 （PLE） とは、広義の派生開発であり、共通性を持つ複数のシステムを視野に入れる開発方法である。単品開発とは異なり、再利用を強く意識する。そのため、複数システム間の共通性を識別する。他方、システムごとに異なる点は可変性として識別し、各システムで追加変更削除が可能になる仕組みを設ける。 本講義では、PLE の基本的な考え方を解説する。
講義の要約	

◆講師紹介
林 好 一 氏
Y's Workshop 代表 兼 ソフトウェアプロセスエキスパート

<略歴>
ソフトウェアプロセスの分野での支援が主な業務であり、2003 年からソフトウェアプロダクトライン開発を対象に加え、関連する調査研究、定義・モデリング、教育、メンタリングや課題識別・解消等の支援活動を行なう。またコミュニティ活動にも積極的に参加し、ワークショップ、セミナー、カンファレンス、勉強会等の企画、運営、そして成果の広報に携わる。2014 年からは Automotive SPICE に沿ったプロセス改善にも従事している。
1983 年から 2019 年まで株式会社ＳＲＡ勤務、以後個人事業主と DNV ビジネス・アシュアランス・ジャパン株式会社契約社員の二足のわらじを履く。

研究論文や著書
共訳：Klaus Pohl ら著、『ソフトウェアプロダクトラインエンジニアリング―ソフトウェア製品系列開発の基礎と概念から技法まで』、エスアイビーアクセス、2009 年

その他（学位、表彰、学会活動）
日本 SPI コンソーシアム（JASPIC）プロダクトライン分科会リーダ
Software Product Line Conference 2011 Tutorial Co-Chair

1. プロダクトライン開発とは？

- プロダクトライン開発とは、Product Line Engineering (PLE)のこと。
- なぜプロダクトラインで開発する必要があるのか。
- 低コスト、高品質、短納期が実現できる。
- 低コストに注目されがちだが、高品質というのがメリットとなる。
  →再利用部分は品質の保証が可能なため。
- 根幹は再利用（ソフトウェアの再利用、プロセスやモジュールの再利用など）
  →バージョン管理される製品が中心の考え方だが、そういった製品でなくても適用が可能。

2. 共通性と可変性

- プロダクトラインはコア資産を作成するという考え方。
- コア資産から発生させて新しいシステムを作っていくようなイメージ。
- コア資産そのものもアップデートしていく必要がある。
- 共通性を基に資産を形成。可変性に着目して製品を導出していく。
- 具体的な導入例としてプリンタードライバーの開発があり、欠陥を 96%減らすことに成功した。

3. どうやってプロダクトライン開発をすすめるのか

- 製品で共通な機能や特徴（フィーチャ）を見つける。ここでいう共通な部分というのは概念的に共通というという意味である。
- プログラムモジュール単位で考えるのではない。概念的にとらえなるべく多くの共通点を見つける。その中で可変の部分はどの箇所になるのかをしっかりと見極める。
- 車で例えるならばエンジンがあるというのは共通性となる。しかしどのタイプのエンジンを使うかは可変性ととらえる。

4. コア資産をフィーチャモデルで表す

- フィーチャモデルを使ってコア資産を表現する。
- フィーチャモデルは FORM 方式で記載する。
- 簡単には作成できない。製品計画やマーケッティング顧客分析などあらゆる情報を集めて作ることが大切。
- フィーチャモデルを作ってから、アーキテクチャを作る。アーキテクチャが決まればコンポーネントが決まる。
- フィーチャモデルは表形式になったりツリー状になったりする。

5. プロダクトライン開発のプロセス

- プロセスは 2 段階になる。
- まずはコア資産を開発する。それからアプリケーションの開発を行う。アプリケーションの開発からコア資産に戻すパターンもある。
- いきなり全部をやろうとするのではなく今ある社内の資産から順次実施していく。
- エンジニアレベルで考えて作成するものではない、もう少し上のレイヤーのメンバーを参画させて議論を進める。
- 2 つのアプローチがある。
  <事前準備型>
  あらかじめにコア製品を作る。初期投資が高く期間も大きい。また、関連ドメインを熟知していて安定していて初期投資の余裕がある企業に向いている。

  <都度対応型>
  将来性や拡大が見込める場合に限りコア資産を作る。ピークを過ぎたら都度開発に戻す。

- 開発体制
  <中央統御型>
  一元的に管理・開発ができるが、コストの問題が発生する。彼らは何かを作るわけではないため。利益を上げるまで時間がかかる。

  <協調開発型>
  各部門で開発したものを全社で共有。費用は分散されるが、他部門のニーズにあうコア資産の開発や協調は難しい。スケジュールや責任問題。仕様変更などの課題がある。

- アジャイル開発では不要なものは作らないという考え方なので、製品全体の設計などを考えて開発を進める必要がある。
- 製品のライフサイクルが目まぐるしく変わるが、都度フィーチャモデルを見直して取捨選択をしていく必要がある。

6. オススメのプロダクトライン開発の進め方

- 既存の資産をコア資産化する。この時既存資産の改善・改修や整理は行わない。やってしまうと先に進めなくなる。
- 具体的には、①既存資産のコア資産化②フィーチャモデルの作成と資産登録③ポートフォリオや製品計画に従って開発するの 3 階層が望ましい。
- 狩野氏の品質モデルに従って資産化を進めるのもよい。
  当たり前品質→優先的にコア資産化。

一元的品質→次にコア資産化　　…など

7. 質疑応答

＜質問＞
PLE とマイクロサービスの考え方との違いについて教えてください。
＜回答＞
重なっている部分は多い。マイクロサービスもちゃんとやろうとしたら背後にあるサービス体系の理解が必要となる。どちらも目的は同じだが、実現方法や技術が異なる。

<質問>
バグ修正はコア資産に対して行い、個々の製品に対して個別に修正することはしない、という理解で正しいでしょうか？
<回答>
その通り。コア資産へ対応が基本となる。コア資産をベースとして各製品を対応していく。

<質問>
PLE の実践では、コア資産のファイル管理（SCM）が難しいと思います。Git や GitHub などでのリポジトリ管理どのようにすればよいでしょうか？ たとえば、どのような単位をリポジトリとするかなどです。
<回答>
統一して管理したいものをコア資産化して、なるべくまとめる（一つのファイルにしていく）。バリエーションがある場合はインターフェースを同じにする。フィーチャモデルを部門ごとに作成し、階層化している企業もある。また、コンフィグなどを用いてルール化し、管理しやすくする。

<質問>
よいフィーチャモデルを作るためにはマネジメントや顧客をうまく巻き込んで「どういう価値が提供できるか、大切か」という共通ビジョンを確立する必要があるかと思います。
この辺をどう実践していけばよいか、経験やご意見をお聞かせください。
<回答>
利害関係者を巻き込んでどれだけ意見をあつめられるかというのが最終的なポイントとなります。よいシステムアーキテクチャを作るのと同じようなスキルが必要と考えます。

<質問>
「フィーチャモデル」とは、特に IT ベンチャーのような企業では、「ビジネスモデル」とほぼ同義なのかと理解しましたが、如何でしょうか？
<回答>
フィーチャモデルとビジネスモデルは両輪であると考えている。フィーチャモデルを「能力」レイヤーに加えるとより良いものが作成できる。

<質問>
「自動化」について紹介がありましたが、例えば、要求、設計資料、ソース、テストケース等、どの部分をどのように自動化すればよいでしょうか。
<回答>

Gear というツールなどがあるが、Gear 以外にもツールは多く存在するので特性を踏まえて利用することが大切。システムやプログラムだけでなくドキュメントをプロダクトライン化しても効果はある。

以上