SS2010-WG7
「製品ファミリのアーキテクチャ品質」

参加者

- 吉村健太郎
  - (株)日立製作所
- 島敏博
  - セイコーエプソン(株)
- 坂井良治
  - 東京エレクトロン　ソフトウェア・テクノロジーズ(株)
- 林好一
  - (株)SRA
- 山内和幸※
  - (株)エクスモーション

※ WGコーディネータ

WG実施スケジュール

日	時	内容
06/09	終日	活動なし ※ 各自、ご興味のある公開プログラム等に参加して下さい
	18:00～	情報交換会
06/10	10:00～11:00	WG招待講演 「製品ファミリのアーキテクチャ設計」 岸知二（早稲田大学教授）
	11:00～12:00	ポジション・ステートメントの発表
	13:00～19:00	議論 1. 認識合わせ 2. アーキテクチャ品質について 3. アーキテクチャ評価について
	19:00～21:00	技術交流会
06/11	10:00～11:00	シンポジウム招待講演 「新たなソフトウェアが生み出されるきっかけとは？」 金子勇
	11:00～12:00	前日の議論の続き
	13:00～14:30	議論のまとめ

1. 認識合わせ

1. 認識合わせ

- WG活動のテーマに限定せず、まずは参加者が抱える疑問や悩みについて、率直に意見交換を行う
  - ただし、テーマを完全に逸脱したものについては扱わない

- これにより、テーマの議論を開始するための、参加者間での共通の知識基盤を作る

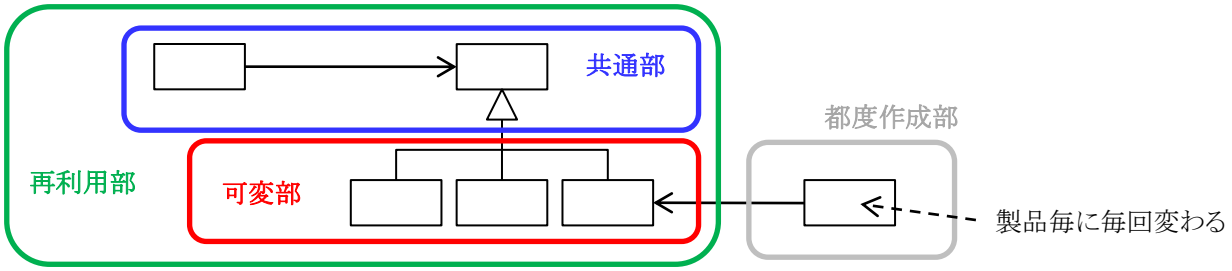
製品群開発の見える化ってうまくいってる？

- 例えば、共通部はココ、可変部はココとか…
  - ステレオタイプではプアすぎる
  - ツールで色付けしてやっていたりする場合もある

- 可変点（variation point）と変異体（variant）は分けて表現すればよいのでは？
  - 可変点にノート等で説明を付けておく手もある
  - ステレオタイプから、特定のビューに必要な情報だけをフィルタリングすることもできる
- 見る人に関心のある情報だけを見せるのがよい
  - ビューを切り替えることで、必要な情報だけが見えるように

用語の使い方は一緒？

- 共通部、可変部、再利用部、都度作成部は何を指す？



- 上図（島さんご提供）で十分
  - 再利用部は、どこまでを資産とするかで変わる
  - 都度作成部は、将来の製品で利用される可能性もあるので、再利用資産へのフィードバックが発生する場合もある
    - 都度作成部を再利用資産化するかどうかは、組織の状況による
  - 後で資産化した場合、製品開発側が資産になったものを使ってくれないことがある
    - 一番有能な人材に資産化させると、そうならないことが多い
    - 資産を使った場合と、そうでない場合のデータを取って見せると、資産を使う効果が説明しやすい
    - メンバー全員が「製品群を作る」というマインドにならないといけない

可変性を見つけるアプローチは？

- 以下のもの以外の方法があるのか？
  - 仕様書からフィーチャ図を作成する
  - コンパイルスイッチ、#ifdefマクロからフィーチャ図を作成する
    - 見えていない可変性は、フィーチャ図では拾えないのでは？

- 見えていない可変性とは？
  - 複数のものが「違う」ことは、容易に発見できる
- 見えていないのは共通性
  - 同じ機能を個別に作っている場合などに発生する
- そのような共通性はフィーチャ図には現われないのでは？
  - それはフィーチャ図の使い方次第
    - 可変性だけを表現する方法
    - 主要フィーチャ（機能、制約等）の構成／関係を表現する方法
  - 後者であれば、共通性も表現は可能
    - 実際、後者の使い方をしている実践者も多い

可変性を管理する方法は？

- 結局、Excelで#ifdefマクロを管理する方法しかないのか？

- 2つの視点が混在している
  - バインディング・メカニズム → 岸先生のご講演で紹介あり
  - 実際のバインディング
- 実際のバインディングは、いろいろな方法がある
  - if, ifdefで行う
  - Makefileで行う（Makefileを製品毎に分ける）
  - コードジェネレータを使う
  - ツール（GEARS, pure::variants, etc.）を利用する
  - etc.
- 設計／実装環境によっても、使える技術が異なる
- 組込みでは、Ifdefもよく使われる
  - Ifdefは振る舞いを切り替えるというより、構成を切り替えるために使っている
    - ファイルの選択等

テスト資産はコア資産化されているのか？

- 製品コード／モデル自体よりも、テスト資産の方が管理が厄介なのでは？

- （技術）管理の問題と、可変性の扱いの問題がある
  - 管理： PL固有の話ではない
  - 可変性の扱い： 製品コード／モデルの場合と同様にできる
- ある資産が進化した場合、そのテスト資産も同様に進化しないといけないが、これが難しい？
  - 難しさよりも、量の影響が大きいのでは？
    - 一般的には、”製品コード ＜ テストコード” であるため
- フィーチャの選択によって異なる組み合わせのテストは、通常、製品開発側で検査している
  - 変更に対して頻繁に大きく変わるなら、コア資産化するべきではない
- テストの視点から見た「再利用性」というのも、認知されてきた感がある

2. アーキテクチャ品質について

アーキテクチャの何が問題なのか？(1)

- 現在のアーキテクチャが、適切なものなのかどうかがわからない
  - 製品ファミリの開発において、製品がQCDの点で効率的に(狙ったとおりに)開発できているのかどうか？
- 現在のアーキテクチャで、この先も大丈夫なのかどうかがわからない
  - 機能や製品バリエーションが増加していく中で、果たして耐えうるのか？
  - それとも、もはや抜本的な手を打つ時期に来ているのか？
- これらを知る手段はあるのか？
  - これが、本WGのモチベーション
  - 上記問題に対して、GQM法を利用して深掘りしていく

アーキテクチャの何が問題なのか？(2)

- 前述の問題の鍵はどこにあるのか？
  - アーキテクチャの品質がどういう状態にあるかを知ることができれば、これらの問題に対する解が得られるのでは？

- 製品(群)開発の総工数に対して:
  - 新規機能数が支配的な悪影響を与えている場合
    - 開発するための技術を向上させないといけない
      - コードベース → モデルベース開発
      - ビッグバンテスト → 早期検証
      - 技術者のスキル獲得、人材の投入等
  - 開発製品数が支配的な悪影響を与えている場合
    - 開発しないで製品を出すための技術を向上させないといけない
      - 再利用のしやすさ／違いの作りやすさ
      - 解析や変更のしやすさ
      - 製品の試験のしやすさ
    - すなわち、保守性が重要と言える

製品ファミリ・アーキテクチャの保守性（1）

- 製品ファミリ開発における保守性とは？
  - ISO-9126の定義

    解析性、変更性、安定性、試験性（標準適応性は除く）
  - これらで十分に問題に答えられるか？ → 何か足りない

- 製品ファミリ固有の視点は？
  - **可変性**（variability ＝ 機種展開容易性）
    - 既に予期されている変更への対応
    - 空間的な変化
  - **進化可能性**（scalability）
    - 予期されていない、将来的な変更への対応
    - 時間的な変化

製品ファミリ・アーキテクチャの保守性（1）

- 前述の6つの特性と、問題とを結びつけると…

G1：現在のアーキテクチャが、適切なものなのかどうかがわからない

Q：解析性、変更性、安定性、試験性、可変性

G2：現在のアーキテクチャで、この先も大丈夫なのかどうかがわからない

Q：進化可能性

- これらを「製品ファミリのアーキテクチャ品質」の主要素として捉えて、議論を進めていくことにする

保守性の計測

- 前述の品質をどのように計測するか？
  - 品質特性をQとすると、Mには行きつかない
    - 抽象的すぎる
  - 6つのサブ特性を更に細分化し、計測可能なQに落とし込まなければならない

- ベーシックなメトリクスで、ある程度のものは測れる
  - コード行数、(サイクロマティック)複雑度、凝集度／結合度、etc.
  - ただし、可変性と進化可能性については、ベーシックなものでは定量化できない

- OOの設計原則／パターン等を適切に適用することでも、保守性の大きな向上は達成できる
  - しかし、これを定量的に証明するのは困難

可変性／進化可能性の細分化

- 可変性
  - 予測した変更に対して、どの程度の工数で対応できているか？
    - 計測方法#1： コア資産からの製品導出にかかる時間
    - 計測方法#2： 実対応時間÷予定対応時間
    - 計測方法#3： 1変更当たりの箇所（行数／パッケージ数／etc.）
- 進化可能性
  - 新規要件に対して、何カ所の変更が発生したか？
    - 計測方法： 変更したパッケージ／クラス／メソッド数
- 時間をメトリクスにするのは困難
  - 正確性が低い、計測しづらい、事後でないとわからない…

→ 可能であれば避けたい

産からの要望

- 産の立場としては、可変性／進化可能性に対する計測アイデアが欲しい
  - 幾つかは先に挙げたとおり
- 是非、学の立場の方々に、多くのQMを定義／発見して頂きたい
  - その有用性、信頼性については、産の側で評価可能
  - あるいは、産学連携して進めていくこともできるだろう

3. アーキテクチャ評価について

アーキテクチャ評価について

- 参照アーキテクチャ自体は実行不可能
  - 可変点が決定(固定／束縛)されていないため
- (製品ファミリの)アーキテクチャの専門家に伺ってみたが、現時点では特別な方法はないとのこと
  - アーキテクチャ評価手法は幾つも提案されている
  - これが製品ファミリというコンテクストでは、可変性(可変点)に対するケアが加わる
- 動的側面の評価については、まだ有効な手法がない
  - 是非、学の新規アイデアに期待したい