第6回ソフトウェアパターン勉強会
解析結果例の発表

永和システムマネジメント
家永英治

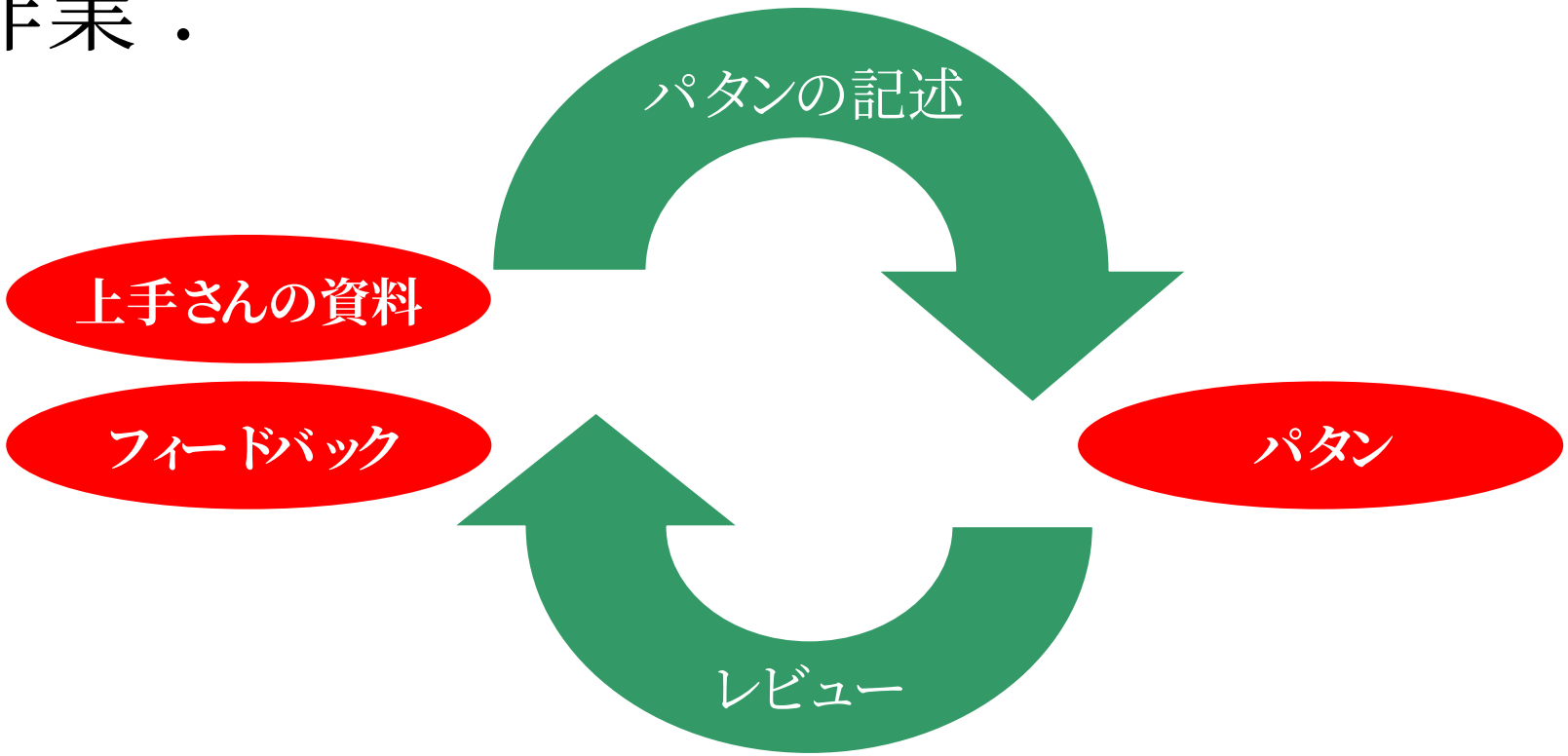
内容

ホスト系技術者のオブジェクト指向（OO）学習パタンランゲージ

- 作成過程の紹介
  - 苦労したこと、うまくなかったこと
  - うまくいった事
- パタン化の成果（一部）
- おわりに
  - ホスト系技術者のOO学習パタンランゲージはできたか否か
  - パタン化のポイント

パタン化作業の概要

- 資料 :上手さんの資料
- 人 :私、天野、懸田
- 作業：



作成過程

ボトムアップだけでは無理がある

まず、上手さんの資料と自分の経験を頼りに、
【背景】【課題】【解法】の構造でパタンを1つ1つ書くことを試みる。

■うまく書けなくて途中で挫折
- そもそも、パタン全体で何を伝えたいか不明
- 全体の目的や構成が曖昧のため、曖昧な【背景】【課題】【解法】しか書けない

そこで・・・

トップダウンと併用すると書きやすい

- パタン全体の目的や構成を決めた
  - 全体で誰に何を伝えたいか（目的）を設定
  - パタン全体のカテゴリーを設定

■書きやすかった
- だれに何を伝えたいかを決めると、文の言い回しなどパタン全体で統一感がでて書きやすかった
- カテゴリーを穴埋めするような感覚でパタンを挙げることができた
- カテゴリーにより【背景】【課題】がまとめやすくなった

作成過程

ホスト系技術者のオブジェクト指向学習のためのパタン（ランゲージ）

- 目的
  - パタンランゲージで学習者に効率的、効果的なOO学習を支援する
    - 読者は、ホスト系技術者でOOを学習する人
    - OO学習のつぼをパタンで理解できる

成果物

OOの学習パタンのカテゴリ

＃OOのテスティングの学習パタン
？？

OOのモデリングの学習パタン
- OOPを事前に学習しよう
- エンティティモデル作成に注力しよう
- オブジェクト指向＝クラス思考
- モデルを動かそう

OOのプログラミングの学習パタン
- デバッガを使ってOOを把握しよう
- ポリモフィズム
- 深入り禁止

総合的OOの学習パタン
- OOマップをGET

学習のための学習パタン
- 過去の開発経験を忘れよう

#小さな目標
#Aha! へぇー!
#師匠 •弟子 •同志を持とう

成果物

レビューでパタン品質が上がる

- 天野、懸田からのフィードバック
  - パタン名は直感的に内容がわかる方が良い
  - 言葉が難しい、言葉が適していない
  - 他に、●●パターンがあるよ！
  - パタンランゲージにはなってない
  - 読者をOO学習者に絞ったのはなぜ？
    - 教育者の方が良いのでは？
    - 分けない方が良いのでは？

難しかったこと

■難しかったこと
- 背景、課題の設定が妥当かの判断
- レビュー
  - どのようにレビューを進めたらいい？
  - どのように成果物の評価をしたらいい？
- 名前付け
  - ネーミング方法 思いつかない・・・)
- ランゲージとしてのパタン間のつながりをどのように発見し、まとめるか

作成過程

成果（一部）

過去の開発経験を忘れよう

【背景】

一般に人は、「過去の経験を活用して問題を解決しようとする」「新しいことを学ぶことに不安や苦痛を感じる」など行動特性を多かれ少なかれ持っている。そのため、過去の経験がOO習得の阻害要因になったり、OOを嫌ったりして、OO学習が進まなくなってしまう場合がある。

【課題】

- OOの新しい経験を習得するには？
- OO学習に対して防衛的にならないためには？

【解法】

過去の開発経験を忘れよう

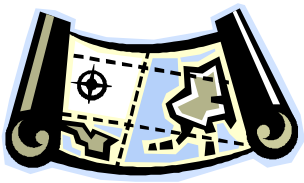
ステップ１:自分が今までの開発経験を持っていることを理解しよう。
ステップ２:一時的に従来の開発経験を忘れよう。
ステップ３:新しくOOを1から学習しよう。
ステップ４:今までの開発経験とOOと融合して、優れた開発者を目指そう。

【関係】

- とくに、「エンティティのモデリングに注力しよう」でこのパタンを適用しよう。
- 「Aha」と併用して、学習を楽しもう。

成果物

OOマップをGET



【背景】
・OOの学習要素は多い。闇雲に学習を進めると、OOの理解がプログラミングのみに偏ったり、学習が途中で止まってしまったりと、OO習得が思うようにはかどらない。

【課題】
・OO学習の最初の基本方針は？
・どのようにして、効果的、効率的にOOを習得する？

【解法】
・まずは全体を勉強しOOの全体観をつかもう。
ステップ１:OOのプログラミングの要点を学ぼう。
ステップ２:OOのモデリングの要点を学ぼう。
ステップ３:OOとプロセスの関係の要点を学ぼう。

【関係】

成果物

エンティティのモデリングに注力しよう

【背景】

OOに慣れないと経験にとらわれてしまい、機能をそのまま分割したクラス、状態を持たないstaticメソッドだけのクラス、巨大コントローラクラス、メニュー＝クラスといったOOのメリット恩恵を得られないモデルを作成する傾向がある。

【課題】

・どのようにして有用なクラス分割を習得する？

・機能の段階的詳細クラス、巨大コントローラクラス、メニュー＝クラスに陥らないためには？

【解法】

・エンティティのクラス図に注力しよう

　・まずエンティティ（E）のモデリングに注力しよう。

　・次に、バウンダリ（B）コントロール（C）をEに追加して、機能の実現を確認しよう。

　・BCを取り除いても、Eが残ることを確認しよう。

　・Eが複数の機能（ユースケース）にまたがって共有されることを確認しよう。

【関係】

・「過去の経験を忘れて」、エンティティのモデリングを学ぼう。

成果物

おわりに

結論：パタンランゲージ化はできなかった

- パタンはいくつか挙がったが・・・
- パタン同士がつながったパタンランゲージまで発展できたとは言えない。
  - 【関係】に他のパタンの名前を列記しただけでは、パタンランゲージとは呼べない（By天野）
  - 活き活きとする状況を表現できなければ、パタンランゲージじゃない（By天野）

パタンを書く際のアドバイス

- 【背景】【課題】【解法】を穴埋めの感覚で書いてみる
  ⇒フォーマットがあると書きやすい
- パタン全体の目的を書いてみる
- パタン全体のカテゴリーを書いてみる
  ⇒方向性を決めると書きやすい
- レビューでフィードバックを得る

⇒一人では分からなかったことを発見できる

パタン化の活動を発展させるためには

- パタンの書き方の明確化
  - 書くためのこつ、ポイント
- パタンを書く動機付け
  - 最優秀パタン賞
- レビューの方法の明確化
  - レビューのチェックポイント
- フィードバックを得やすい環境作り
  - 読んでフィードバック
  - 使ってフィードバック

ご清聴ありがとうございました。

ホスト系技術者のオブジェクト指向学習のためのパタン（ランゲージ）

ホスト系技術者のオブジェクト指向学習のためのパタン（ランゲージ）

- 目的
  - パタンランゲージで学習者に効率的、効果的なOO学習を支援する
    - 読者は、ホスト系技術者でOOを学習する人
    - OO学習のつぼをパタンで理解できる

成果物

OOの学習パタンのカテゴリ

＃OOのテスティングの学習パタン
？？

OOのモデリングの学習パタン
- OOPを事前に学習しよう
- エンティティモデル作成に注力しよう
- オブジェクト指向＝クラス思考
- モデルを動かそう

OOのプログラミングの学習パタン
- デバッガを使ってOOを把握しよう
- ポリモフィズム
- 深入り禁止

総合的OOの学習パタン
- OOマップをGET

学習のための学習パタン
- 過去の開発経験を忘れよう

#小さな目標
#Aha! へぇー!
#師匠 •弟子 •同志を持とう

学習のための学習パタン

#Aha,へぇー

【背景】
・OOを学ぶには、覚えることが多く、時間がかかることが知られている。学習者は継続的な学習が必要となる。

【課題】
・なにを動機にして、OOを学び続けたらいい？

【解法】
・Ahaを実践しよう。Ahaとは、学習者の理解度でわからない状態から、わかった状態に遷移する際に学習者が感じる「喜び」を指す。
・「Aha！ポリモフィズム、すげっ！」「Aha！テスト簡単！BCEのEに注力しよう」とOO学習で新しい発見を楽しもう。発見が続けば、継続的なOO学習の可能性が高くなる。

【関係】
・「同志」で「Aha」を共有すれば、「Aha」の喜びが倍になり、継続的に学習が可能になる。
・「師匠」は「弟子」の「Aha」を引き出そう。

過去の開発経験を忘れよう

【背景】

一般に人は、「過去の経験を活用して問題を解決しようとする」「新しいことを学ぶことに不安や苦痛を感じる」など行動特性を多かれ少なかれ持っている。そのため、過去の経験がOO習得の阻害要因になったり、OOを嫌ったりして、OO学習が進まなくなってしまう場合がある。

【課題】

- OOの新しい経験を習得するには？
- OO学習に対して防衛的にならないためには？

【解法】

過去の開発経験を忘れよう

ステップ１:自分が今までの開発経験を持っていることを理解しよう。

ステップ２:一時的に従来の開発経験を忘れよう。

ステップ３:新しくOOを1から学習しよう。

ステップ４:今までの開発経験とOOと融合して、優れた開発者を目指そう。

【関係】

- とくに、「エンティティのモデリングに注力しよう」でこのパタンを適用しよう。
- 「Aha」と併用して、学習を楽しもう。

小さな目標

【背景】
・OOの学習範囲はプログラミング、UML、モデリング、ツール、プロセス、パターン・・・と広い。そのため学習者は、その広さに、しり込みしてしまう場合がある。
【課題】
・どのようにして、学習要素の多さに圧倒されず、学習を続ける？
【解法】
・小さな目標を持とう。
・目標をクリアして達成感を味わおう。
・上の二つを繰り返し、少しづつ確実にOOを習得しよう。
・小さなことからこつこつと！
【関係】
・Aha、と関連する。

#師匠 •弟子 •同志を持とう

【背景】

•個人が標準的テキストを使ってOO習得ができるほど、OOの 技能 ノウハウは、体系化、言語化されていない。そのため、テキストのみでは、効率的 •効果的にOOのコツを習得するのは難しい。

【課題】

•OOの ノウハウ •かん •コツをどのように習得する？

•どのように効率的効果的にOOを学習する？

【解法】

•師匠を持とう。ペアプロ、コードインスペクション、モデルの レビュー等で師匠からフィードバックを得て、OOの理解を得よう。

•弟子を持とう。人に説明する機会を得て、OOの理解を深めよう。

•同志を持とう。 同志でOOの理解を共有しよう。

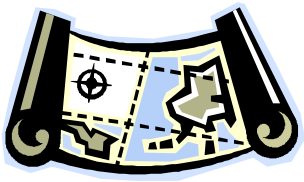
•近くに師匠 •弟子 •同志が見つからない場合は、インターネットで探そう。

【関係】

同志」と Aha」を共有しよう。

総合的OOの学習パタン

OOマップをGET



【背景】
OOの学習要素は多い。闇雲に学習を進めると、OOの理解が
プログラミングのみに偏ったり、学習が途中で止まってしまったりと、
OO習得が思うようにはかどらない。
【課題】
OO学習の最初の基本方針は？
どのようにして、効果的、効率的にOOを習得する？
【解法】
まずは全体を勉強しOOの全体観をつかもう。
ステップ１:OOのプログラミングの要点を学ぼう
ステップ２:OOのモデリングの要点を学ぼう
ステップ３:OOとプロセスの関係の要点を学ぼう
【関係】

OOのプログラミングの学習パタン

IDEを使おう

【背景】
・開発環境構築を素早く構築し、プログラミングのサイクル（プログラム作成・コンパイル・実行）の回転率を上げると、学習フィードバックの機会が多くなり学習が進む。
【課題】
・どうしたら素早く開発環境を整えられる？
・どうしたらプログラミングのサイクルを素早くできる？
【解法】
・IDEを使おう
・メリット
　・クラスパスの設定等の環境構築が簡単。
　・プログラミングしやすい。コンパイルしやすい。
【関係】

学習言語にJavaを選択しよう

【背景】

・OOの要所を学習したい。だが、言語によってはメモリ管理等の難しい要素も学習する必要がある。

【課題】

・プログラミングでOOの要所を学ぶのに適した言語は？

【解法】

・JavaでOOを学習しよう

・メリット

  ・メモリ管理不要

  ・豊富な情報

  ・わかり易い言語仕様

【関係】

ポリモフィズム

【背景】
・ポリモフィズムは、オブジェクト指向の重要な項目の 1つである。
【課題】
・どのようにしてポリモフィズムを学ぶ？
【解法】
・プログラミングでポリモフィズムを学ぼう。
  ・ポリモフィズムのしかけを知ろう
  ・条件分岐と対比して、ポリモフィズムの利点を知ろう
【関係】
・「IDE」を使って、ポリモフィズムの理解を深めよう。

IDEの機能でOO基礎を理解しよう

【背景】
- クラスとオブジェクトの違いがわからない。
- ポリモフィズムがわからない。

【課題】
- どのようにして、オブジェクト指向の基礎を理解する？

【解法】
- IDE（Eclipse等）のデバッガのウォッチ機能を使って一つのクラスで複数のインスタンス（オブジェクト）が生成できることを確認しよう。クラス変数とインスタンス変数とを比較しよう。
- IDEのメソッド定義の移動とデバッガのステップ実行でプログラムの動きを確認してポリモフィズムを理解しよう。

【関係】
- 「オブジェクト図を描こう」と併用して、オブジェクトを把握しよう。

深入り禁止

【背景】

・プログラマの性として、プログラミングの面白さについついはまってしまう場合がある。しかし、現在、オブジェクト指向は実装の領域だけではなく分析・設計といった領域まで発展している。実践的なOO開発者には広い視野が求められる。

【課題】

・プログラマのみに特化した開発者ではなく幅広い視野を持ったオブジェクト指向開発者になるには？

【解法】

・OOPをほどほどに学習したら、プログラミングのモデリングの学習に移行しよう。

【関係】

・「OOマップをGET」を特殊化したもの。

OOのモデリングの学習パタン

モデルを動かそう

【背景】

・プログラムは動くため学習フィードバックが得やすい。だが、モデルは、直接動かないため、具体的な学習フィードバックが得られない。結果、理解が難しい場合がある。

【課題】

・どのようにして、慣れない抽象的なモデルを学習する？

・モデルとプログラムの関係性をどのように理解する？

【解法】

・描いた抽象 ・要約のモデルを具体 ・詳細のプログラムの対応関係を確認しよう。

・プログラムで具体的学習フィードバックを得よう。

・プログラムとモデルとの対応関係を確認しよう。

・プログラムでモデルが妥当だったかを検討しよう。

【関係】

・「師匠」にモデルを見てもらって、学習フィードバックを得よう。

オブジェクト指向≠クラス思考

(オブジェクト図を描こう)

【背景】

・モデリングで、クラス図の作成に注力しがち。結果、「オブジェクト」の視点が欠けた有用でないクラス図を作ってしまう。

【課題】

・どのようにして、オブジェクトを考慮した有用なクラス図を作成する？

【解法】

・オブジェクト図を描いて、オブジェクトを把握しよう

・オブジェクト図からクラス図を描こう

・クラス図からオブジェクト図（or相互作用図）の一例を作成し、クラス図が妥当か検討しよう

【関係】

・「IDEの機能でOO基礎を理解しよう」と併用でオブジェクトの理解を深めよう。

・巨大コントローラクラス、メニュー＝クラス、を作ってしまうようであれば、「エンティティのモデリングに注力しよう」を適用しよう

エンティティのモデリングに注力しよう

【背景】

OOに慣れないと経験にとらわれてしまい、機能をそのまま分割したクラス、状態を持たないstaticメソッドだけのクラス、巨大コントローラクラス、メニュー＝クラスといったOOのメリット恩恵を得られないモデルを作成する傾向がある。

【課題】

・どのようにして有用なクラス分割を習得する？

・機能の段階的詳細クラス、巨大コントローラクラス、メニュー＝クラスに陥らないためには？

【解法】

・エンティティのクラス図に注力しよう

　・まずエンティティ（E）のモデリングに注力しよう。

　・次に、バウンダリ（B）コントロール（C）をEに追加して、機能の実現を確認しよう。

　・BCを取り除いても、Eが残ることを確認しよう。

　・Eが複数の機能（ユースケース）にまたがって共有されることを確認しよう。

【関係】

・「初心」に戻ってエンティティのモデリングを学ぼう。

OOPを事前に学習しよう

【背景】

・OOプログラミングを学習せず、モデリングのみを学習すると、実装不可能なモデルを作成してしまう。

【課題】

・どのようにして実装可能なモデルを作成する能力を習得する？

【解法】

・まずOOPで実装を学習しよう。

・それからOOのモデリングを学習しよう。

・モデルとコードの関係を学習しよう。

【関係】