ソフトウェア工学の基礎

この勉強会について

- ソフトウェア工学に関係する教科書・資料を各自で読み発表する輪講方式の勉強会
- 過去のログ
  - https://github.com/himrock922/software_research
- 前回のログ
  - himrock922→オートマトン、形式的な証明方法
  - kawai → 命題論理、述語論理、様相論理
- 輪講する内容の変更
  - himrock922→ ソフトウェア工学の基礎知識を総合的に学べるような輪講に変更

目次

- ソフトウェアについて
- ソフトウェア工学とは
- ソフトウェアプロセス
- ソフトウェアプロセスの評価
- プロセスプログラミング

ソフトウェアについて

- プログラミング言語という人工的に創作された言語で書かれる
  - そのような抽象的な記述でありながら、コンピュータを通して実世界に直接働きかけることが可能
- ソフトウェアの2つの性質
  1. 小説や論文などの作品と類似する性質(言語表現による創作結果という意味で)
  2. 実利的な工業製品として生産される性質(実世界への直接的な作用という機能から)
- そして、ソフトウェアを制作するプロセス
  - 言語による表現行為と人工物の設計開発という2つの側面を持つ
- 今回の発表からは、ソフトウェアをどう作ればよいか、何をアプローチとすれば良いのか考えられる発表にする

ソフトウェア工学とは

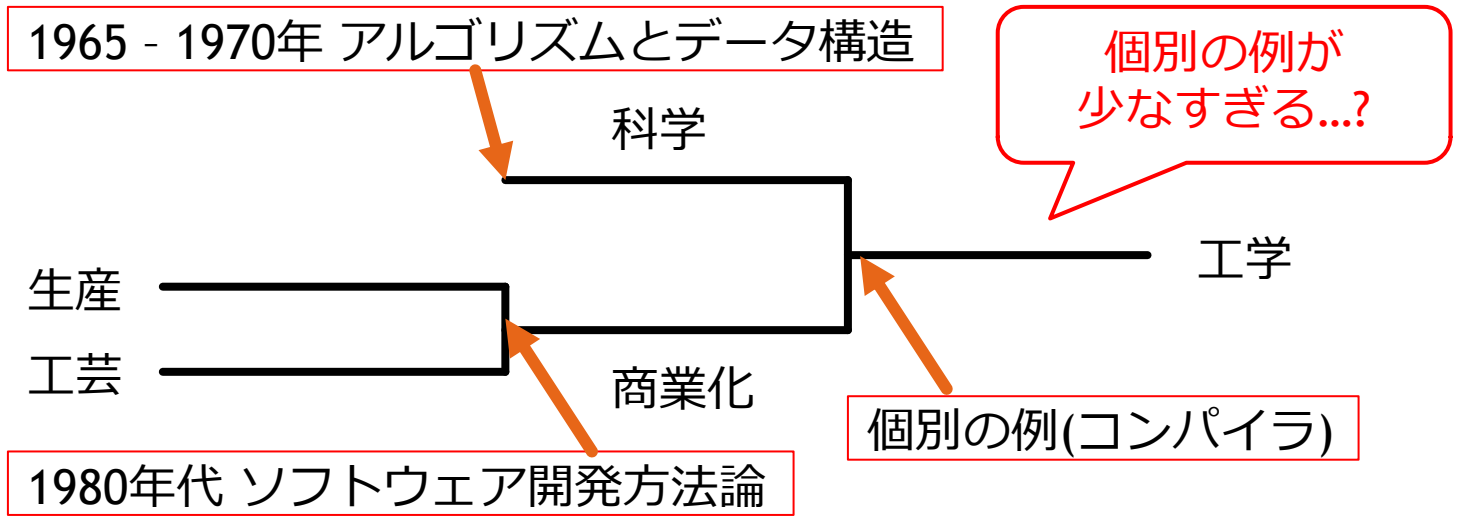
- ソフトウェアのもつ独特な性質に注目しながら、それを作るプロセスや手法を探求すること、またその分野
- ソフトウェア工学の定義(IEEE Std 610-1990)
  1. ソフトウェアの開発、運用、保守に対する、系統的で統制され定量化可能な方法
  2. (1)のような方法の研究
- ソフトウェアという抽象的なもの、更にはプログラミング言語で表現するという行為自体、人類が経験しなかったものといってもよい
- 3つの側面を考慮してソフトウェアを上手く開発してく
  1. 抽象度の高いソフトウェアを開発し保守するための純粋に技術的な側面
  2. 組織的に開発し管理するための管理的側面
  3. 利用者の満足度を上げ、また開発者のチーム内の協力体制を築き士気を上げるためのコミュニケーションや認知といった人間的側面

ソフトウェア工学の範囲

- 米国で読まれるソフトウェア工学の教科書の著者
  - Roger S. Pressman(https://en.wikipedia.org/wiki/Roger_S._Pressman)
  - Ian Sommerville(https://en.wikipedia.org/wiki/Ian_Sommerville_(academic))
  - Shari Lawrence Pfleeger
- ソフトウェア工学の知識体系を整理して記述することを目的とした文書
  - SWEBOK(IEEEとACMの共同作業)
- 計算機科学の標準カリキュラムを作成する活動の最中で、作成されたソフトウェア工学の分野を取り上げた文書
  - J97(計算機科学のカリキュラム: 情報処理学会)
  - CC2001(計算機科学のカリキュラム: IEEEとACMが共同作業)

ソフトウェア工学の工学としての成熟度

- カーネギーメロン大学 Mary Shawの主張



- **Systems Software Research is Irrelevant(Rob Pike)**
- ソフトウェアの開発において銀の弾丸はないが、それらを地道に発展・普及・利用する努力を続けることが大事

ソフトウェアプロセス

- ソフトウェア工学が扱う対象の基本的な構成要素は二つ
  1. プロダクト

     エンジニアリングの過程で生成される中間製品や文書を含めたすべての生産物を指す
  2. プロセス

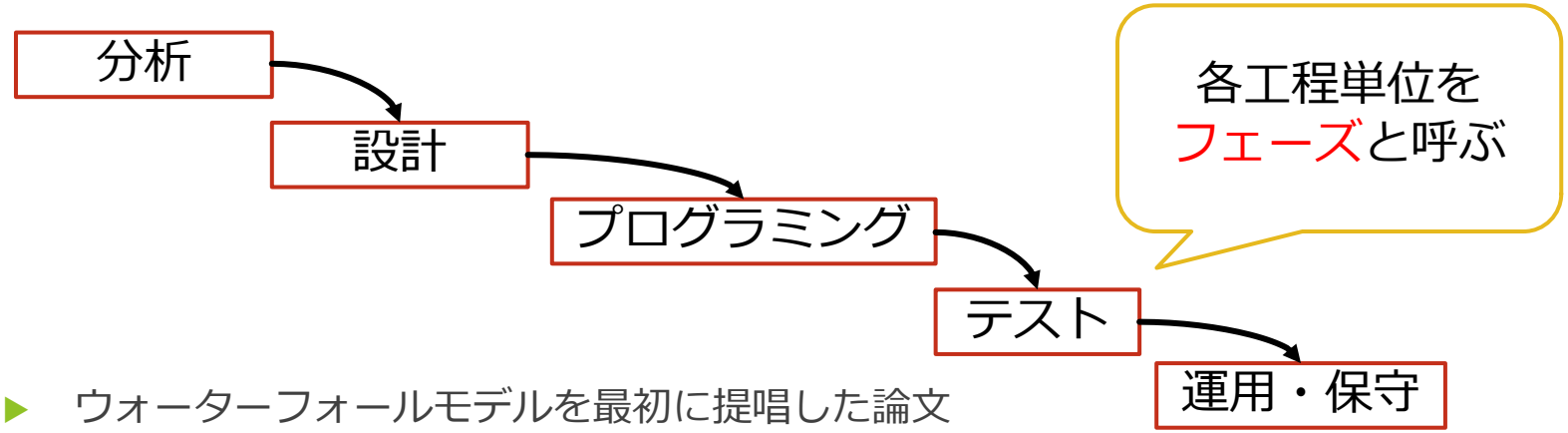
     プロダクトを生み出す工程
- ソフトウェア工学にとって、ソフトウェア開発のプロセス自身を方法論的な考察の対象にすることは重要

ライフサイクルモデル

- ソフトウェアの開発計画から設計開発、運用、保守、廃棄にいたる過程を標準的なモデルとして表すもの
- 企業におけるソフトウェア開発の標準工程は何らかのライフサイクルモデルに基づき作られ、プロジェクト管理もモデルに基づいて実施されている
- ソフトウェアにおけるモデルの目的
  1. 標準的なソフトウェア開発手段を定め、開発担当者の作業をガイドする
  2. 開発プロジェクトを管理するのに準拠する管理モデルとして用いる
  3. 開発の方法論、使用するツールや開発環境、文書体系などを標準的に定めるための基盤作り
- このようなソフトウェアプロセスのモデルは、標準的なものをベースとして、個々の開発組織ごとにさらに個々のプロジェクトごとに設計される必要がある

ウォーターフォールモデル

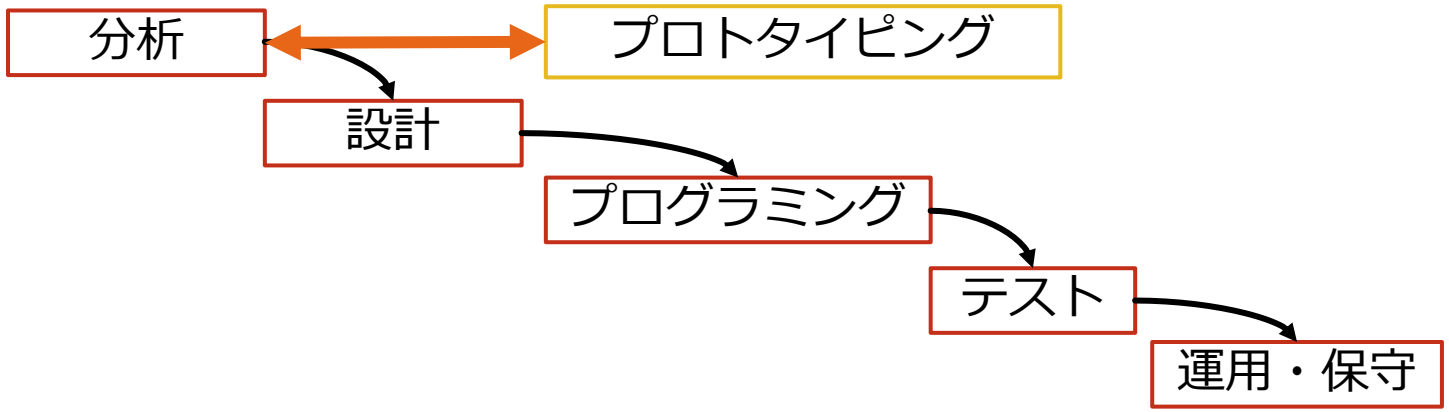
- ライフサイクルモデルのうち、もっとも古くからあるモデル



- ウォーターフォールモデルを最初に提唱した論文
  - W. W. Royceによるもの(と言われている)
- ウォーターフォールモデル自体に問題があると言われる2つの実現困難なこと
  1. フェーズ間に明確な区切りを置くとともに、その間はきちんと形式化された文書で受け渡す
  2. フェーズの手戻りを極力なくす

ウォーターフォールモデルに代わるソフトウェアモデル

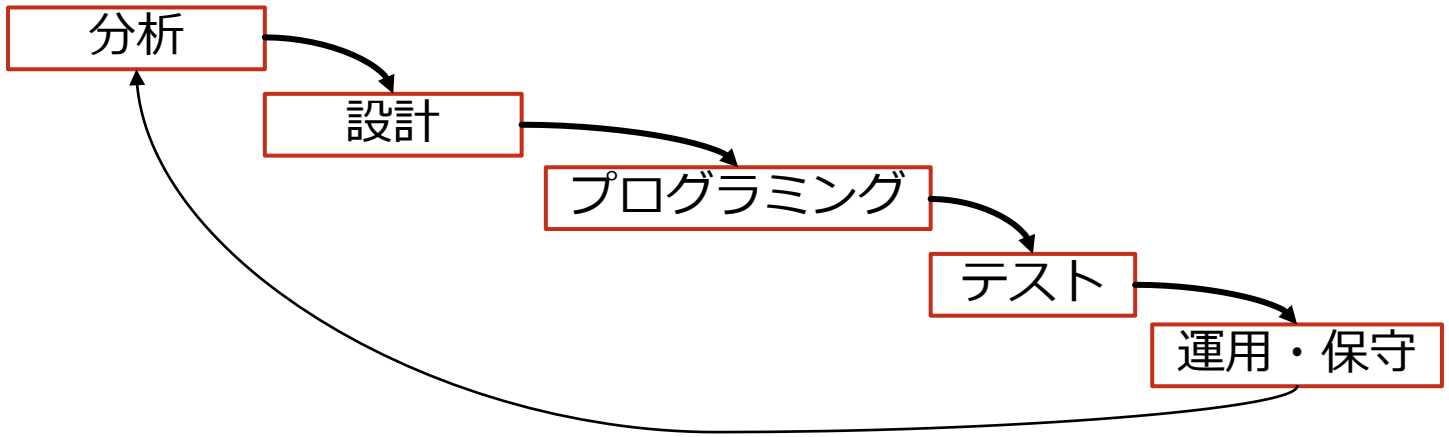
- プロトタイピング型モデル
  - 最終的に運用するシステムを作る前に、プロトタイプを作りそれを評価するプロセスを要求分析フェーズに組み入れたモデル



- プロトタイプは繰り返し手直しされるが、適切なタイミングで仕様を確定し、以降はウォーターフォールに従う

ウォーターフォールモデルに代わるソフトウェアモデル 2

- スパイラルモデル
  - 初めは小さな機能範囲のシステムを実現し、それを改良していくプロセスを繰り返すモデル



- スパイラルモデルにおける開発のプロセスがXtreme Programming(XP)という方法論がしばしば、採用されている

ソフトウェアプロセスの評価

- CMM(Capability Maturity Model)がプロセスの成熟度を評価するモデルとして著名
- 5段階の比較的単純な評価

レベル1: 初期

プロセスはほとんど定義されておらず、ソフトウェアの開発が偶発的もしくは個人の能力に依存している

レベル2: 反復可能

管理するための基本的な管理プロセスは確立しており、過去の経験を活かし、そのプロセスを繰り返すだけの仕組みはもっている状態

レベル3: 既定義

管理と開発の両面からソフトウェアプロセスが文書化され、標準化されて組織に組み込まれており、定義済みの標準プロセスに手を加えて利用している

レベル4: 管理下

ソフトウェアのプロセスとプロダクトについて、くわしい定量的データが収集され、それに基づいて分析、管理が行われている

レベル5: 最適化

定量的な分析のフィードバックによりプロセス改善が不断におこなわれている

プロセスの改善活動

- 前述の評価で重要なのはランク付けではなく、プロセスの改善につなげること
- 各組織が具体的な改善内容を策定
- SPIN
  - ソフトウェアプロセス改善活動。地域単位、組織単位で促進
- SE-CMM
  - CMMから発展したソフトウェアを特に対象にしたモデル
- CMMI
  - CMMから発展したシステムエンジニアリングモデル、製品とプロセスの統合モデルなどから策定
- ISO9000
- SPICE
- …etc

ソフトウェアプロセスの観察と改善

- 具体的にソフトウェアプロセスの改善を行っていくには、既存のプロセスを定量的に観察し、その結果に基づいて改善案を構築する必要がある
- 改善案作成のアプローチ
  - 事例研究の分析を積み重ねていくもの
  - 認知科学の手法を適用するもの
  - 品質管理、とくにTQM(Total Quality Management)
- 具体的なソフトウェアプロセスの観察例として、大規模プロジェクトにおけるプロセスの手戻りを分析した事例
  - 伊藤暁人: 大規模システム開発におけるプロジェクト管理問題. ソフトウェアシンポジウム'91論文集, D9-D16, 1991
  - T.Tamai and A.Itou: Requirements and design change in large-scale software development – Analysis from the viewpoint of process backtracking. In 15th International Conference on Software Engineering, pp. 167-176, Baltimore, Martland, U.S.A., May 1993.

プロセスプログラミング

- ソフトウェアを開発するプロセスを手続的なプログラムとして記述し、「実行」しようというアイディア
  - 以降、プロセスを記述する形式言語やモデル、それを動かすプロセス支援環境の提案が数多く発表された
- プロセスプログラミングの目標
  1. 実際のプロセス、とくに人間の行動が主要な要素となる開発過程を観察、分析し、改善方法を探る
  2. プロセスの形式的記述に適した形式システム、言語の研究および実際の記述実験をおこなう
  3. プロセスという概念を核とするソフトウェア開発環境を開発する

プロセスプログラミングの代表的な研究例

- プロセスの形式記述/モデル化
  - T. Katayama: A hierarchical and functional software process description and its enaction. In Proceedings 11th International Conference on Software Engineering, pp. 343-352. IEEEE, May 1989.
  - K. Inoue, T. Ogihara, T.Kikuno, and K. Torii: A formal method for process descriptions. IN Proceedings 11th International Conference on Software Engineering, pp. 145-153, May 1989
- プロセス中心ソフトウェア開発環境
  - Arcadiaプロジェクト
    - プロセス記述用の言語とその記述および実行支援システムの開発・検証
  - Marvelプロジェクト
    - ルールベースによるプロセス記述手段とオブジェクトソースによるソフトウェアの生成物管理とを組み合わせた開発環境の提供

プロセスプログラミングの代表的な研究例(2)

- プロセスの設計と実行
  - プロセス設計の考え方
    - Prescriptive(規定的)
      - プロセスを手順として示す。具体的で実行と結びつけやすいが、柔軟性に欠ける。
    - Proscriptive(制約的)
      - プロセスの満たすべき条件を記述する。プロセスの動的な変化に対応しやすいが、実行との間にギャップする。
  - ただし、プロセスをソフトウェア開発プロジェクトの1件ごとに、全く新たに設計しなおすことは考えにくいので、基本となるプロセスを前提にカスタマイズするのが現実
- プロセスの実行とは?
  - プロセスの様々な特性を自動的に計測し、数値管理を行ったり等の開発支援やプロセスを自動実行するようなこと

プロセスプログラミングの代表的な研究例(3)と研究の今後

- スケジューリングの記述方法
  - ほとんどのモデルが取り扱っているのがプロセスのスケジューリング
  - 作業単位ごとに起こる事象を記述したり、作業から発生する事象から間接的にスケジュールを求めたり
- オブジェクト管理
  - プロセスで生成され、また利用されるオブジェクトの管理
  - プロセスとの関連から、リポジトリシステムのような一貫性を保持する技術など
- プロセスプログラミングの研究の今後
  - ソフトウェアプロセスの研究例は1990年代に集中している
  - その後、ソフトウェアアーキテクチャの方に関心が向く
  - プロセスとプロダクトのサイクルで研究トレンドが交互に変わっていくようになる?

まとめ

- ソフトウェアについて
- ソフトウェア工学とは
- ソフトウェアプロセス
- ソフトウェアプロセスの評価
- プロセスプログラミング

- 既存のプロジェクトであれば、既存のプロセスをカスタマイズすることが一般的だが、研究として何が課題だったか考えられるようになることは大事

参考文献

- ソフトウェア工学の基礎 (岩波オンデマンドブックス)
  - 著者: 玉井 哲雄
  - https://www.amazon.co.jp/%E3%82%BD%E3%83%95%E3%83%88%E3%82%A6%E3%82%A7%E3%82%A2%E5%B7%A5%E5%AD%A6%E3%81%AE%E5%9F%BA%E7%A4%8E-%E5%B2%A9%E6%B3%A2%E3%82%AA%E3%83%B3%E3%83%87%E3%83%9E%E3%83%B3%E3%83%89%E3%83%96%E3%83%83%E3%82%AF%E3%82%B9-%E7%8E%89%E4%BA%95-%E5%93%B2%E9%9B%84/dp/4007304572/ref=dp_ob_title_bk