Engineering and Software Engineering
Michael Jackson

南山大学大学院　数理情報研究科
森下　月菜

シナリオ

- ソフトウェア工学
- 確立された工学分野での失敗
- ソフトウェア製品の失敗
- コンピュータシステムの周りの「環境」
- ソフトウェアエンジニアの対象
- ソフトウェア開発者に必要な技術と知識
- ソフトウェアとその問題領域
- 自然科学と工学の違い

- まとめ
- 参考文献

ソフトウェア工学

「モデル」「設計」「保守」
などの用語，概念は
既存の工学からきている

- 新しい工学分野である
  - 確立した工学分野の理論的基礎と実用的な形式に基づくべき
- ソフトウェア開発の失敗
  - 膨大な経済的損失と多くの被害に関わる
- ソフトウェア開発の成功
  - 過去の失敗の理解が必須

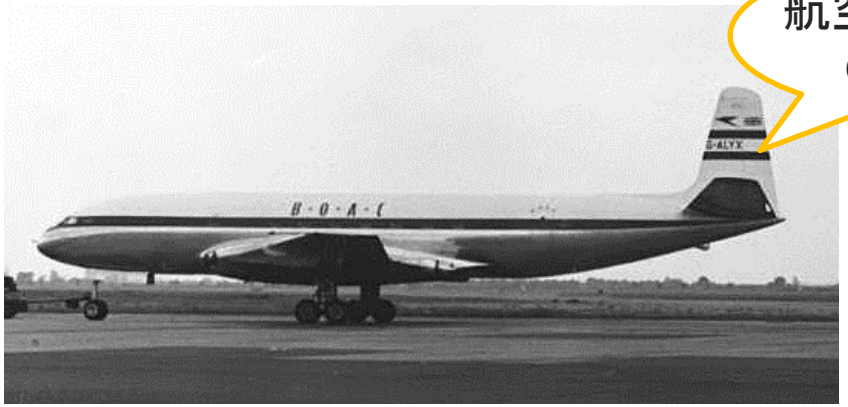
今回はソフトウェアと
その周辺の話題を紹介

確立された工学分野での失敗

- 長期実績上の局所的欠陥　＝　発展途上でよくある話
  - 例1　：　1986年　　スペースシャトル「チャレンジャー」の破損
  - 例2　：　1950年代　コメット1墜落事故
  - 例3　：　1940年　　タコマナローズ橋の落橋

チャレンジャーの打ち上げ

航空宇宙工学
の失敗例



英国海外航空のコメットMk.I

航空・金属工学
の失敗例

土木工学
の失敗例

タコマナローズの海峡とタコマナローズ橋

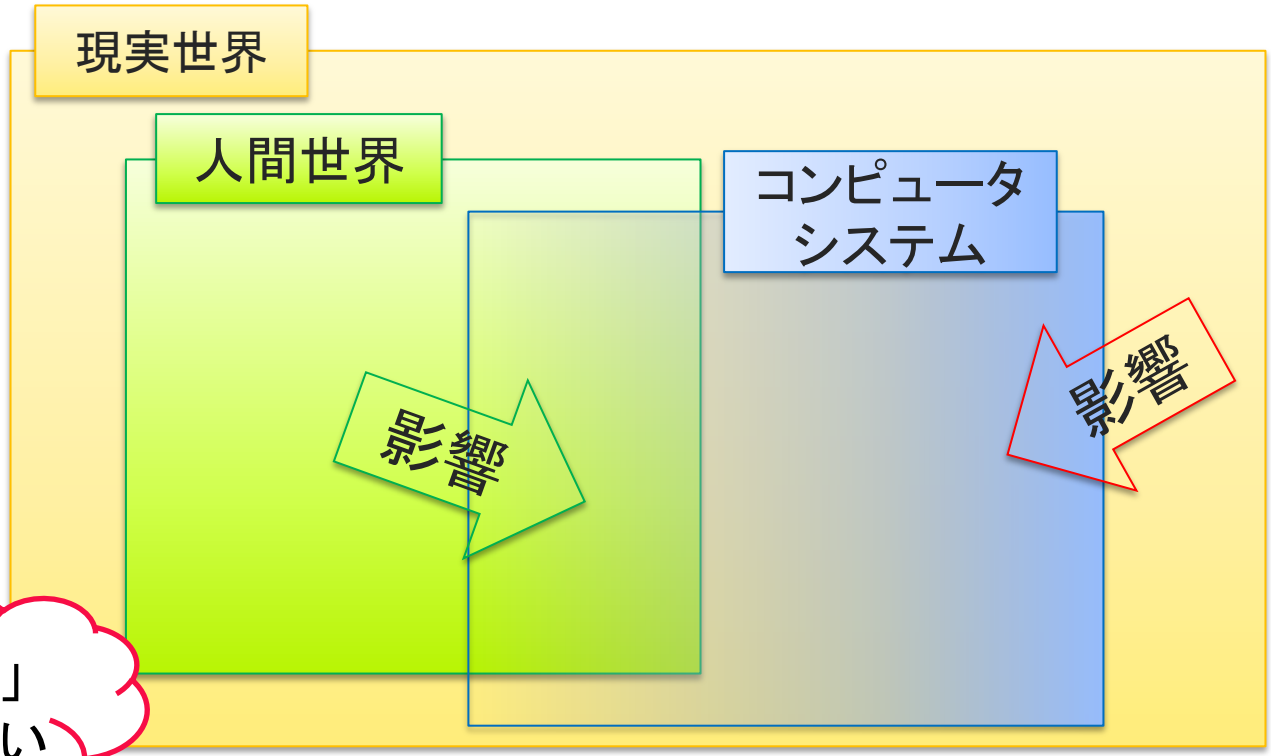
出典：Wikipedia

ソフトウェア製品の失敗

- クリティカルシステムの失敗に大きく影響
  - 例　：　放射線治療機器　セラック25
    - カナダ原子力公社とフランスCGR-MeV社によって開発・製造
    - 制御に用いられたソフトウェアのバグにより，過度の被曝事故を起こす
      - 照射位置が正しい位置でないにもかかわらず電子線ビームが患者に直接当たり事故が発生
      - 従来機セラック20のコードに潜んでいたバグによるもの

ソフトウェアエンジニアは
ディペンダビリティを第一に考えるべき

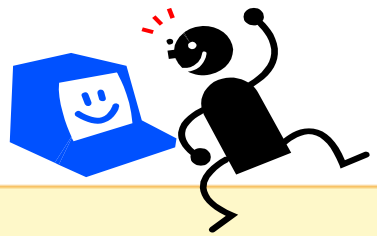
コンピュータの周りの「環境」



「外にある」
だけではない

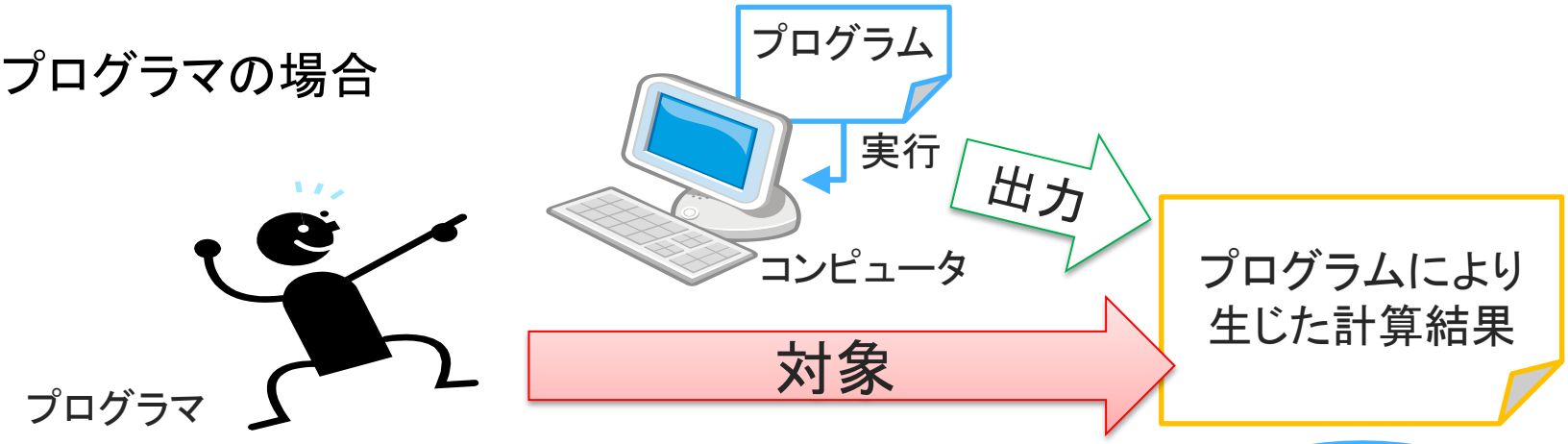
- 現実世界と人間世界がコンピュータシステムに与える影響

ソフトウェアが全く機能しない　or　ソフトウェアが適切に機能する



ソフトウェアエンジニアの対象

- プログラマの場合



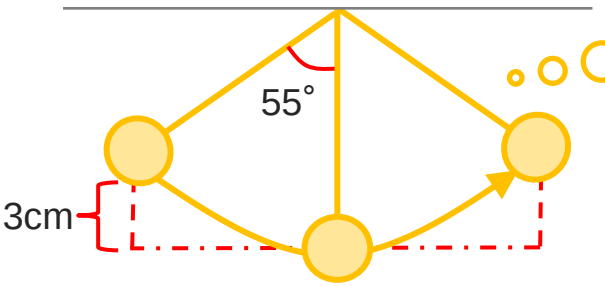
- ソフトウェアエンジニアの場合

コンピュータが物理世界と人間世界に
接触するためエンジニアの対象も広くなる

自然科学と工学の違い

- 自然科学
  - マシンの操作原理の知識
  - 例　：　振り子がどう動くか

興味があるのは原理



軽くて伸び縮みしない糸を使用すると仮定

空気抵抗がないと仮定

- 工学
  - マシンの操作原理の知識　＋　操作原理に耐える物理特性の知識
  - 例　：　自動車がどう動くか　＋　空気抵抗

興味があるのは現実

摩擦はどれだけ生じるか

空気抵抗はどれだけ生じるか

工学では現実世界に則した見解が必要

ソフトウェア開発者に必要な技術と知識

- プログラミングに関わる形式的な記号処理
- 関係のある問題世界とそれに関わる知識や技術

- 例１　：　数列の計算プログラムを作る
  - ✓ 必要な知識
    - ✓ 記号処理
    - ✓ 数列の知識
    - ✓ 定理

    } 問題世界の知識・技術
- 例２　：　飛行船を飛ばすプログラムを作る
  - ✓ 必要な知識
    - ✓ 記号処理
    - ✓ 物理学
    - ✓ 物理法則
    - ✓ 制御工学

    } 問題世界の知識・技術

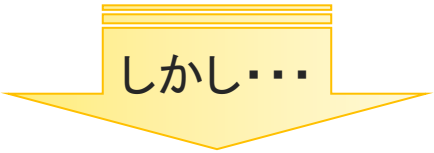
必須

プログラムの知識と
技術だけでは不十分

プログラムに関する知識・技術以外の
知識・技術も重要

ソフトウェアとその問題領域

- 「ソフトウェアとその問題世界の間の親密な関係は，形式的な
  プログラム仕様を間に入れることで分離されるべき」(ダイクストラの主張)



- 切り離して考えることは不可能
  - ソフトウェアコンポーネントは問題世界と密接に関わる
    - ABSの例 ： 路面が濡れている → スリップしないようにABSが働く

問題世界のコンテキスト

問題世界の
コンテキストに依存

問題領域とソフトウェアの振舞いを
区別した形式仕様は書けない

確立された分野の教訓

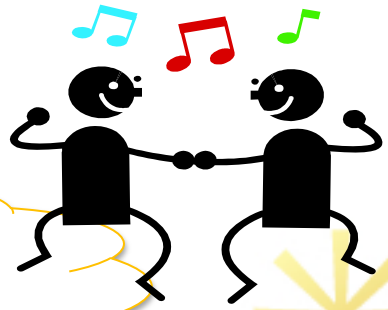
- ソフトウェア工学
  - 確立された分野で培われたものを取り入れることで成立

- ソフトウェア工学が現れた頃のエンジニアの見解
  - 確立された分野はその分野の対象物だけが対象
  - e.g)　機械工学は機械を作るための工学＝　機械のことだけを考えれば良い

同様に…

ソフトウェアのことだけを考えれば良いという誤解が生じた

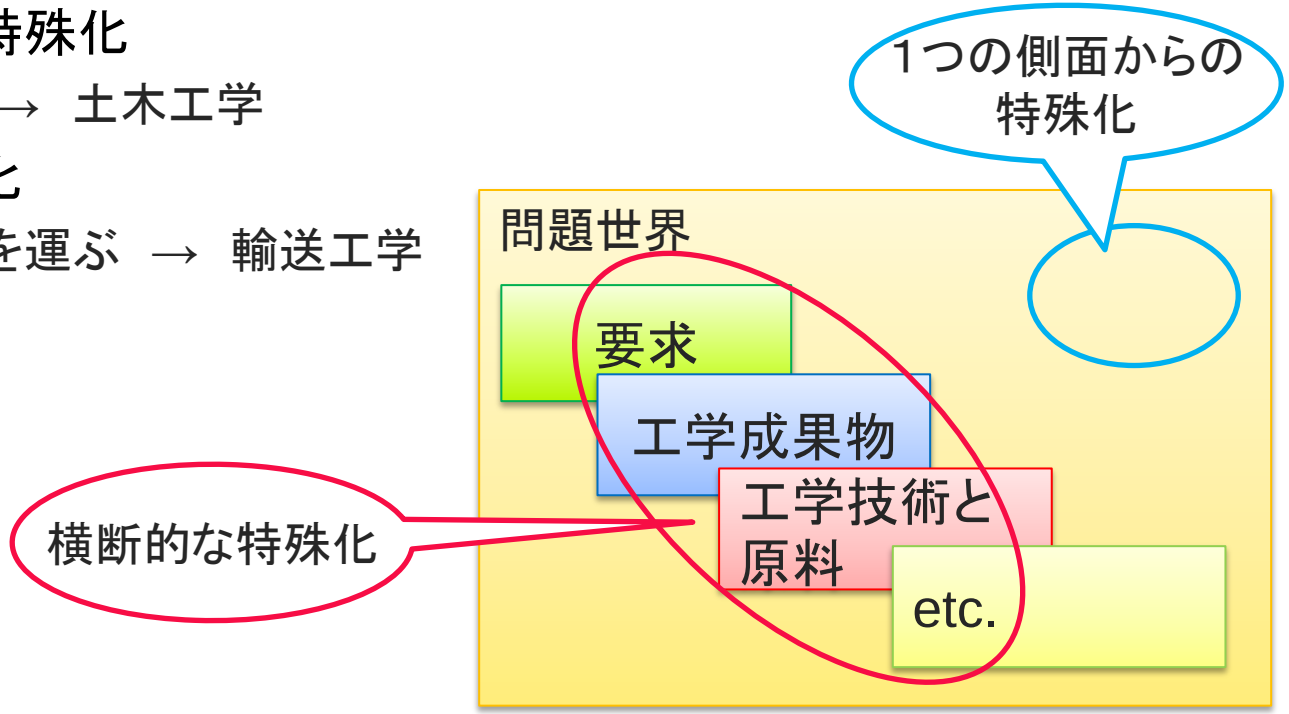
実際そんなことはない！



ソフトウェア工学は他の分野のものを
うまく組み合わせて成り立っている

特殊化の持つ側面

- 工学成果物に関する特殊化
  - e.g.) 車を作る → 自動車工学
- 問題世界に関する特殊化
  - e.g.) 川の氾濫 → 土木工学
- 要求に関する特殊化
  - e.g.) 効率よく物を運ぶ → 輸送工学
- その他

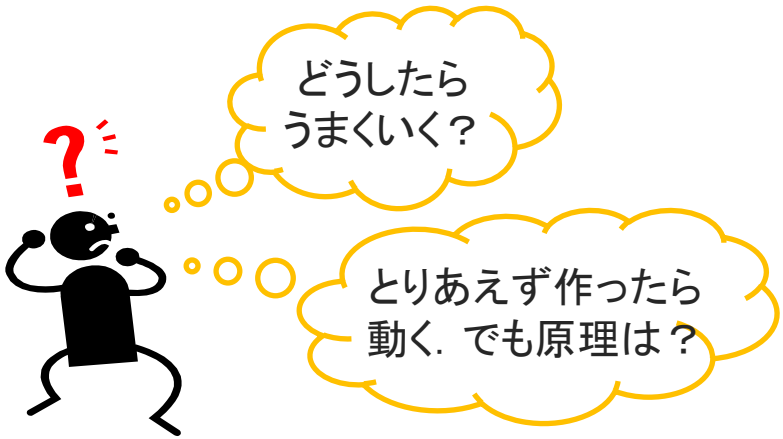


様々な側面から特殊化は可能

一般的な設計と斬新な設計

- 一般的な設計
  - 経験則が存在
    - 古くからの知識が蓄積されている
    - ほとんどの人が理解可能

- 斬新な設計
  - 経験則が存在しない
    - 新しいものを作る
    - 成功の推測が出来ない
    - ほとんどの人が理解不能
  - 技術として十分に確立されていない



新しいものを作る時には技術として
確立できるよう原理や構造を示す必要がある

まとめ

- ソフトウェアエンジニアはディペンダビリティを第一に考えるべき
- コンピュータの周りにある世界はコンピュータへ影響を与える
- エンジニアの興味の対象は物理世界と人間世界にも広げる必要がある
- プログラムに関する知識・技術以外の知識・技術も重要
- 工学では現実世界に則した見解が必要

参考文献

- Michael Jackson, Engineering and Software Engineering, 2011, The Future of Software Engineering, Pages 100-114.
- Wikipedia, http://ja.wikipedia.org/wiki/.
- 組み込みソフトウェア工房, http://d.hatena.ne.jp/sessamian/20100227.
- 中島　震, 形式手法入門　ロジックによるソフトウェア設計, オーム社, 2012.
- 「正当性検証と妥当性確認」講義資料

Engineering and Software Engineering
Michael Jackson
END

南山大学大学院　数理情報研究科
森下　月菜