

# ソフトウェアエンジニアの昨今

ソフトウェア研究会in秋葉原

池袋分室

講師 池田公平

# 石器時代(真空管)

- ソフトウェアは、電氣的な配線によって組み込まれた。(ハードの一部だった)
- やがて、外部記憶の命令を実行する方式が発明された。(ノイマン型)
- 現在のソフトウェアもノイマン型であり、基本構造は変わっていない
  - ノイマン型に代わる構造はいくつか発明されては消えていった。(もしくは研究途中)
  - 当時のソフトウェアエンジニアはハードウェアエンジニアと密接な関係があり、ソフトウェアのみのエンジニアリングが確立していたか不明。(調査不足)

# ソフトウェア工学(1960年代～1975年)

- NASAによる一大プロジェクトにより、ソフトウェア工学が生まれた
- ソフトウェアの開発を、基本設計・詳細設計・実装・テストなどに細分化し、大人数によって効率的なソフトウェア開発が可能になった
- このころ、ソフトウェア専門職としてのエンジニアが誕生したと思われる。
- マイクロコンピュータはまだ発明されていない
- ソフトウェア工学による開発の問題点として、「人月の神話」という本が著作され現在でもベストセラーとなっている。(1975年)

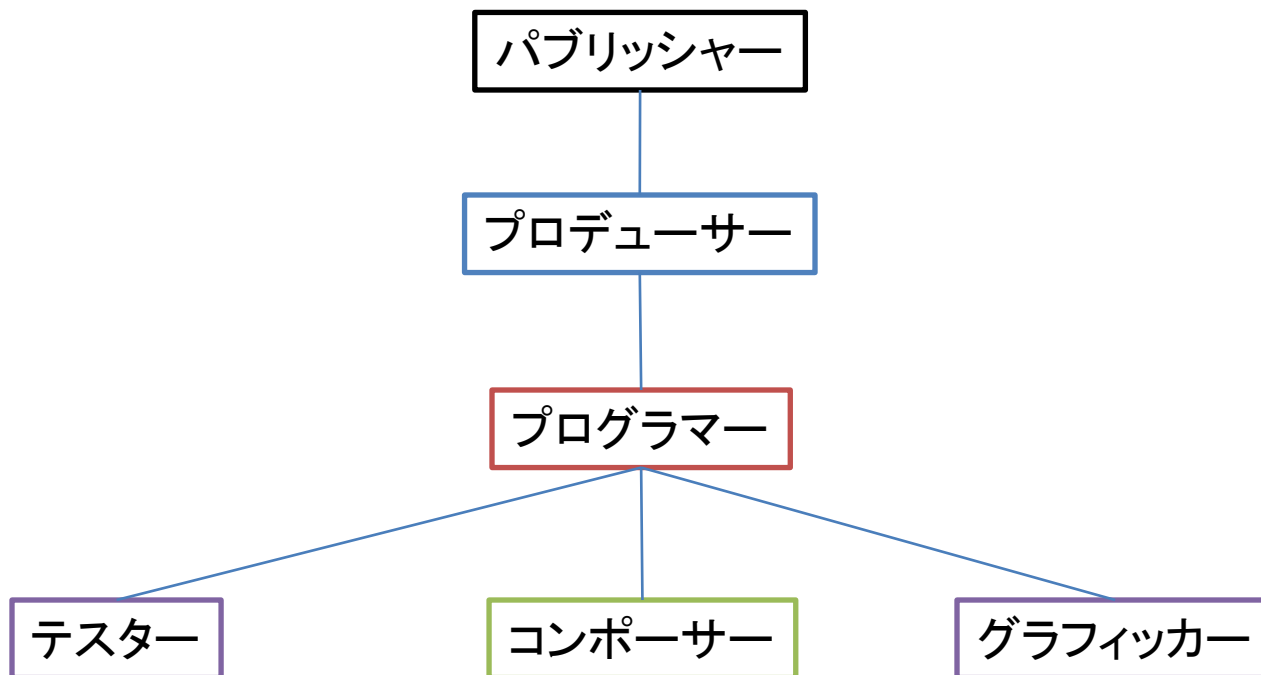
# マイコン時代(1976年～1982年)

- 8ビットマイクロプロセッサが発明され、ホビーとして誰でもコンピューターに触れるようになる。
- プログラミングはマシン語で、ハードウェアの知識は不可欠
- 大型コンピューターのソフトウェアとマイコンのソフトウェアとは隔たりがあった
- マイコンのソフトは、設計からテストまで一人の人間が行うのが普通だった

# 8ビットゲーム機時代(1982年～1985年)

- PC6001,ファミコン、MSXなどの8ビットゲーム機がヒットする
- プログラミングができなくても、アプリケーションユーザーがPC(マイコン)を買うようになる
- ゲームプログラミングは、デザインから実装までほとんど一人で行うのが普通だった。グラフィックスと音楽を別の人間が担当することはあった

# 8ビットゲーム機時代の開発体制



プログラマーが実質的にディレクターとなりゲーム制作のほとんどすべてのことを仕切っていた。1982年ごろから、グラフィックや音楽は分業化されたが、プログラマーがメインであることはしばらく続いた

# 8ビットゲーム機時代の開発プロセス

企画

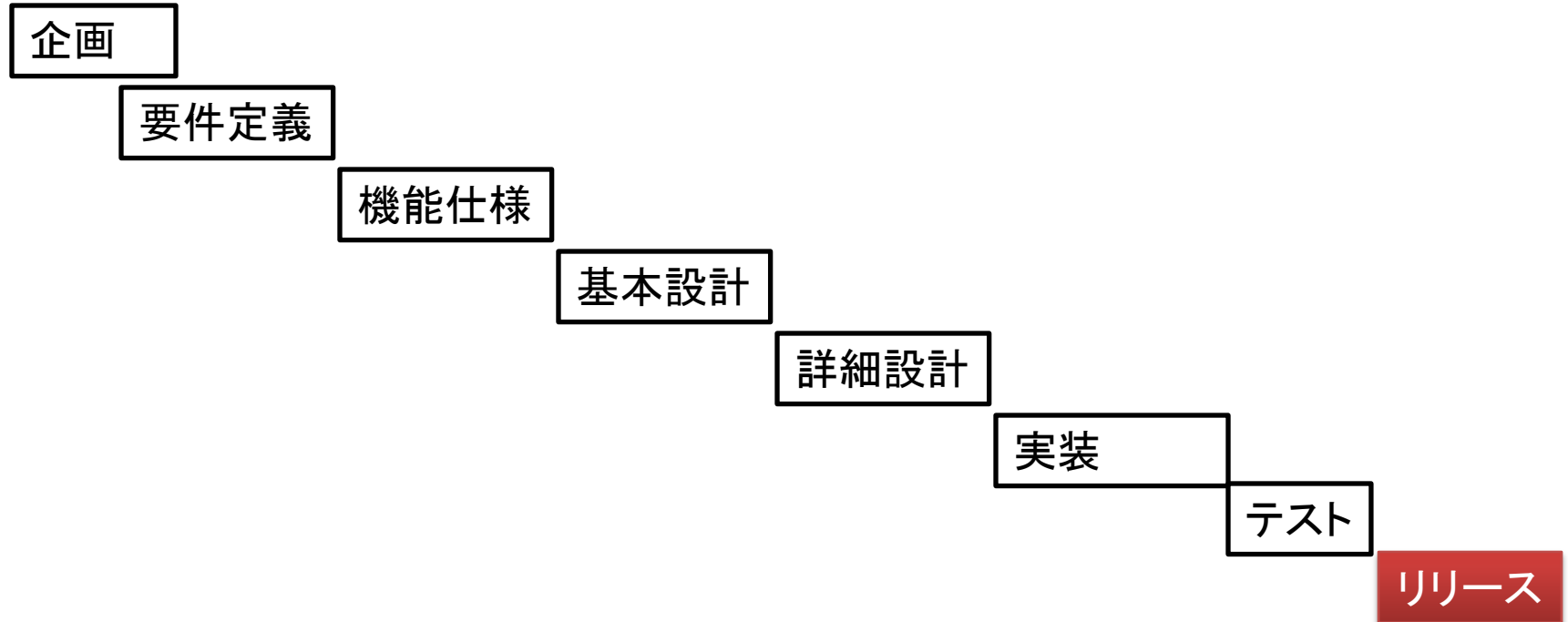
開発

デバッグ & テスト

リリース

開発のイニシアティブはプログラマにあり、ほとんどすべての決定事項はプログラマが管理していた。  
その結果、開発が長引きリリースが遅れることは日常茶飯事だった。

# ソフトウェア工学によるソフトウェア開発プロセス



開発のプロセスは細分化され、それぞれの担当者が自分の責任の範囲内で作業を行い、次のステップに託すスタイル。  
進捗度合いが分かりやすく、遅れの出にくい方式と考えられていた。



# ソフトウェア工学による開発の利点

- 作業が細分化されるため、多くの人員を効率的に配置できる
- それぞれのステップで検証を行うため、責任の所在がわかりやすい
- 設計と実装を分離できるため、業務委託や外注作業がやりやすい
- 進行がステップ毎なので、進捗度合いが把握しやすい

# 企業にとってのソフトウェア工学の利点

- ソフトウェア工学は、大企業による大きなプロジェクトに向いている
- 作業分担のやりやすさ、業務委託のやりやすさも大企業向けである
- 設計を企業側でおこなうことで、製品のポリシーや企業イメージを守りやすい
- 実装とテストを外注することで、企業内のコストを削減できる

# 企業側の開発方針

- 製品のコンセプト・企業イメージ・ポリシーにかかわる部分は企業側が受け持つ
- 基本的な設計作業までは企業側で行う
- 実装とテストは、下請け企業に外注し、コストダウンを図る
- 企業側のリスクは減り、下請けのリスクは増大する

# その結果何が起きたか？

- 企業内での実装エンジニア(プログラマ)が不要になった
- プログラミング技術の蓄積、再利用、発展が企業内で困難になった
- 優秀なプログラマは大企業を去り、中小のソフトハウスで活躍するようになる
- 大企業は、コストダウンのためより安い下請け企業に業務を発注するようになる
- 価格競争により価格破壊が起こり、品質の低下、仕事や技術の海外流出がはじまる
- 昨今のソフトウェアの不具合による重大なトラブルには、こんな背景があるのではないか？

# 2010年におけるサラリーマンエンジニアの境遇

- 収入の低下
- 残業の増加
- 多忙によるモチベーション・勉強意欲の低下
- 自己啓発、技術継承など業務以外のことができなくなる
- リストラ、会社が無くなる不安
- 他の業種への転換・転職

# 2010年におけるフリーエンジニアの境遇

- 月単価の低下
- 業務の多忙化(単価低下を補うため)
- 仕事を探すことが困難(海外に仕事を奪われる)
- 世代の空洞化(儲からない仕事に就きたがる人はいない)

# エンジニアとして生き残るには

- 作業の効率化により、時間当たりの作業量を増やす ⇒ 低単価への対応
- 製品品質の向上 ⇒ 発展途上国のエンジニアと差別化
- マネのできない、オリジナリティのある製品を作る ⇒ 単価の安い海外エンジニアと同じ土俵で勝負しない

# 生き残るために

- 先端技術の習得
- 広範囲におよぶ知識の習得
- 方法論の習得
- 情報収集
- 企画から実装まで一貫したプロジェクトをめざす