



Software Testing Sa.Ga

～ ソフトウェアテスターの苦闘・工夫
の歴史を振り返る ～

辰巳 敬三
(ASTER)

松木 晋祐
(ACCESS)

2011年9月7日



目次

- ◎ コンピュータの歴史
- ◎ コンピュータゲームの歴史
- ◎ テストの考え方の歴史の変遷
- ◎ テスト技法の歴史
- ◎ テスト技術の研究動向
- ◎ まとめ



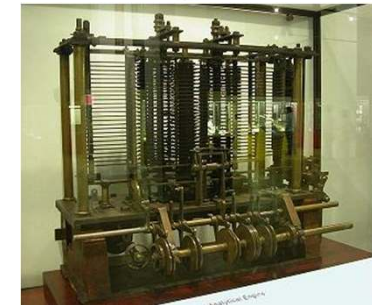
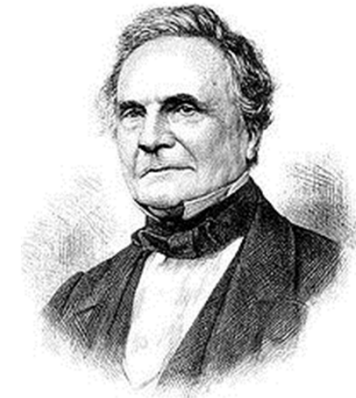
コンピュータの歴史

(1/3)

◎ コンピュータの原型

❖ Charles Babbage(英国)の「解析機関」

- 1837年に設計を開始 (完成には至らず)
- 入力(プログラムとデータ)はパンチカード
- 出力はプリンター、曲線プロッター、ベル



◎ 最初のプログラマー

❖ Ada Byron, Lady Lovelace

- 解析機関のプログラムを作成(1843年)
Babbageの講演記録の注釈の中に記述



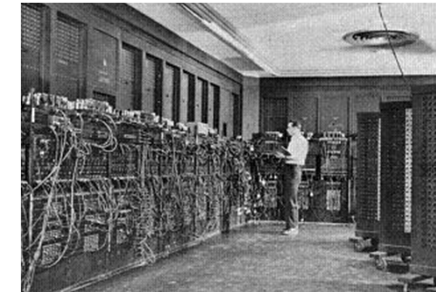
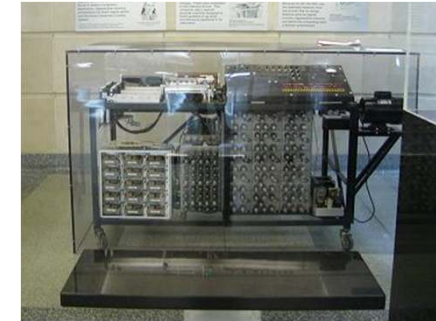


コンピュータの歴史

(2/3)

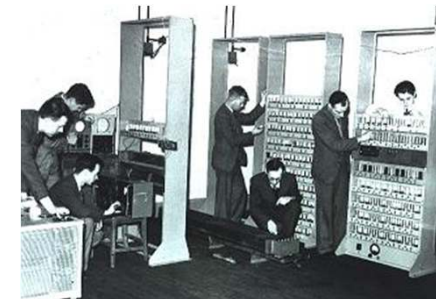
◎ 最初の電子計算機

- ❖ ABC (Atanasoff-Berry Computer), 1939年
 - 米国アイオワ州立大学のAtanasoffとBerryが開発
- ❖ ENIAC (Electronic Numerical Integrator and Computer), 1946年
 - 米国ペンシルバニア大学のEckertとMauchlyが開発



◎ 最初のプログラム内蔵型(ノイマン型)コンピュータ

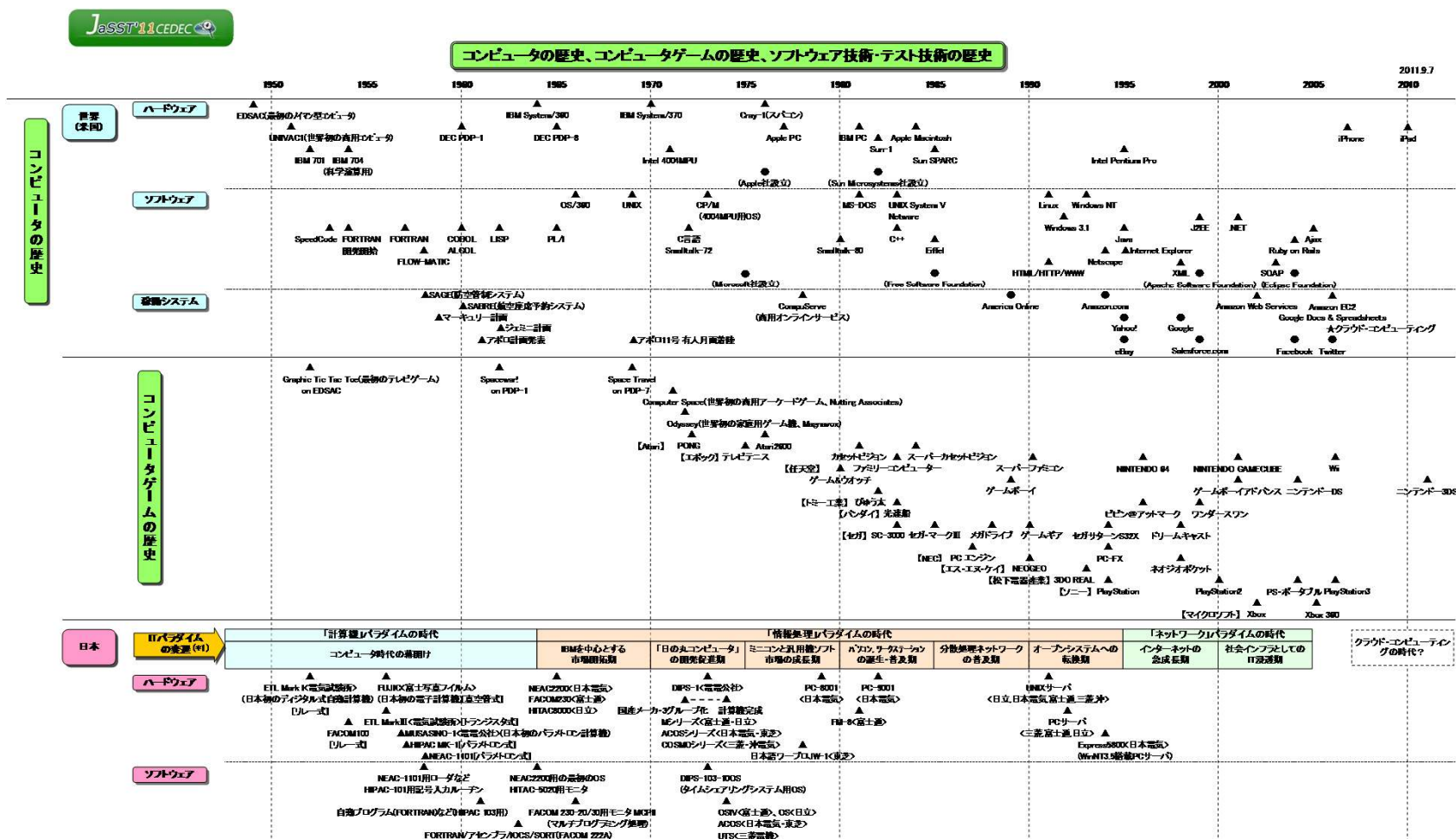
- ❖ EDSAC (Electronic Delay Storage Automatic Calculator), 1949年
 - 英国ケンブリッジ大学のWilkesらが開発





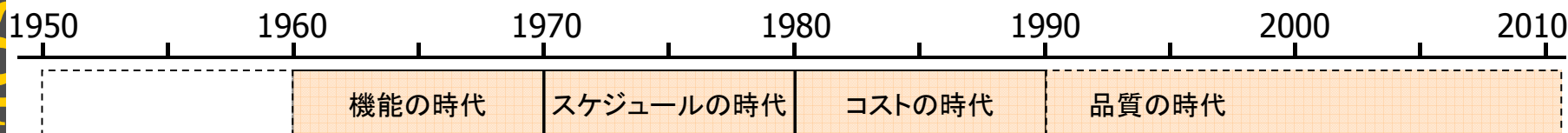
(3/3)

◎ 別紙の歴史年表参照





歴史的変遷：マネジメントの観点



◎ 1960年代: 機能の時代

- ❖ ITが社会に入り始めた
- ❖ 社会的ニーズに応える機能の開発が主要課題

◎ 1970年代: スケジュールの時代

- ❖ ソフトウェア危機(NATO会議, 1968年)
- ❖ 計画期間内にソフトウェア開発を完了させることが主要課題
- ❖ 工程を明確化したライフサイクルモデルが紹介され始めた

◎ 1980年代: コストの時代

- ❖ ハードウェアコストの継続的低減、パソコンの出現
- ❖ ソフトウェアの価格の高さに目が向き、コスト/生産性が主要課題
- ❖ 各種のコストモデル(FP, COCOMO, ...)が提案され、適用され始めた

◎ 1990年代: 品質の時代

- ❖ ソフトウェアに対する社会の依存度の増大
- ❖ ソフトウェアの故障や使いにくさが深刻な事態を招くことが広く認識

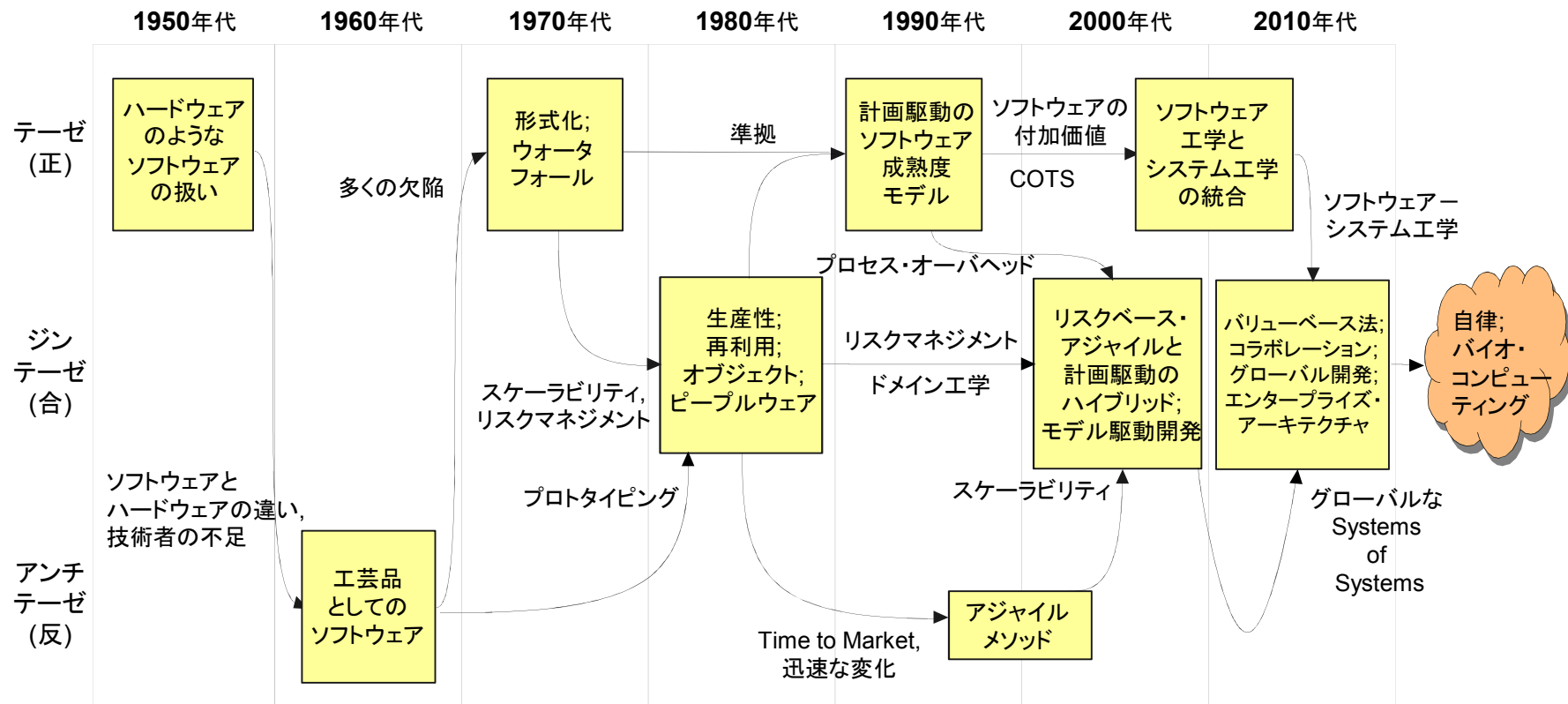
[出典] V. Basili and J. Musa, "The Future Engineering of Software: A Management Perspective," 1991

歴史的変遷：ソフトウェア工学の観点

1950	1960	1970	1980	1990	2000	2010
ハードウェア工学的	ソフトウェア工芸 (Crafting)	形式化とウォーターフォールプロセス	生産性と拡張性	並行プロセス vs. 順次プロセス	俊敏性と価値	

◎ ヘーゲルの進化ビュー(B. Boehm)

❖ ソフトウェア工学の歴史的変遷を弁証法で捉える



[出典] B. Boehm, A View of 20th and 21st Century Software Engineering, 2006のプレゼンテーションスライドから引用



目次

- ◎ コンピュータの歴史
- ◎ コンピュータゲームの歴史
- ◎ テストの考え方の歴史の変遷
- ◎ テスト技法の歴史
- ◎ テスト技術の研究動向
- ◎ まとめ



コンピュータゲームの歴史 (1/4)

◎ 最初のコンピュータ(?)ゲーム

- ❖ チェス機械: エル・アヘドレシスタ, 1912年
 - レオナルド・トーレス・ケベード(スペインの数学者)が開発
 - 内部の電気機械的な装置により盤面の状況を判断し駒の動きを決めることができた



◎ 最初のテレビゲーム

- ❖ Graphic Tic Tac Toe (EDSAC上で動作する三目並べ), 1952年
 - アレキサンダー・サンディ・ダグラス(ケンブリッジ大の学生)が開発
 - 人間とコンピュータの相互疎通に関する研究の一例として作製。画面の記録が残っている初のテレビゲーム





コンピュータゲームの歴史 (2/4)

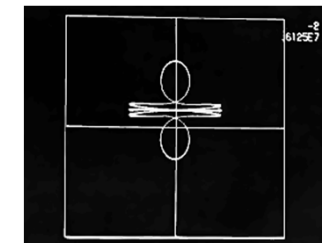
◎ コンピュータ/ソフトウェア開発とゲーム ～研究者の娯楽とUNIX誕生秘話～

❖ Spacewar!, 1962年

- 世界初のシューティングゲーム
- スティーブ・ラッセル(MITの学生)らが開発
- PDP-1(DECのミニコン、1960年リリース)上のデモンストレーションプログラム
- PDP-1はハッカー文化の母

❖ SpaceTravel, 1969年

- ケン・トンプソン(Bell Labs)がGECOS上で開発
- その後、PDP-7(1965年リリース)に移植、これがUNICS(UNIX)開発の契機となった





コンピュータゲームの歴史 (3/4)

◎ 商業としての成立

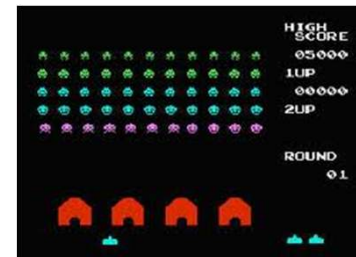
❖ 海外ゲーム産業

- 1971年：「コンピュータスペース」
世界初のアーケードゲーム
- 1972年：「オデッセイ」
世界初の家庭用ゲーム機
- 同年：「PONG」(Atari社)
世界で最初に商業的に成功



❖ 国内ゲーム産業

- 1976年：「ブロック崩し」
全国の喫茶店にテーブル筐体が設置
- 1978年：「スペースインベーダー」
喫茶店がゲームセンターへ業態変更
- 1980年：「ゲーム&ウォッチ」
- 1982年：「ファミリーコンピュータ」





コンピュータゲームの歴史 (4/4)

◎ そしてオンラインへ

❖ ウルティマオンライン/ディアブロ

- 1997年：MMO、MOという規模の分類を代表するソフトが奇しくも同年に発売

❖ ファンタシースターオンライン

- 2000年：国内家庭用ゲーム機でのMORPGの中興の祖

❖ ファイナルファンタジーXI/ラグナロクオンライン

- 2002年：前者は家庭用ゲーム機初のMMORPG、
後者は最多プレイヤー数を誇る韓国製MMORPGの覇者

❖ ソーシャルゲーム

- 2007年：Facebookアプリ「FarmVille」が8000万ユーザを獲得
- 2009年：「怪盗ロワイヤル」のヒットにより日本でも認知される



コンピュータゲームと“バグ” (1/2)

◎ 時代ごとに移り変わる“バグ”の意味

❖ 黎明期(1970年代)

- スケールがとても小さくオープンソースソフトウェアの感覚でユーザ自身が開発者でもあった

❖ 産業化時代(1980年代)

- 海外では品質という概念が成立する前に、粗製濫造のツケを業界全体で負わされた「アタリ・ショック」が発生
- 一方国内では「裏技」「高等テクニック」として発生させること自体が賞賛された牧歌的な時代

❖ 家庭用全盛期(1990年代)

- ゲームソフトの回収や、ブランドの失墜などバグが深刻味を帯びてくる。ゲームデバッグ専門の会社も登場

❖ オンライン時代(2000年代)

- 課金モデルとソーシャルグラフの発達により、ひとつのバグがビジネス自体の存続を危うくする可能性を帯びる一方、その修正も即座に行えるという側面も併せ持つ



コンピュータゲームと“バグ” (2/2)

◎ ゲームにおけるバグの二面性

『正のバグ』

- ❖ 名古屋撃ち(スペースインベーダー)
- ❖ キャンセル(ストリートファイターII)
- ❖ 無限増殖(スーパーマリオブラザーズ)

『負のバグ』

- ❖ 要件を把握できていない
 - 少牌がシステムに組み込まれていない
 - ガンシューティングで照準が最初からズレている
 - キャッチャーが後ろを向いている
- ❖ ゲーム性を著しく損なう
 - ボス攻略後特定のボタンを押下すると100%フリーズ
 - 対戦スゴロク型ゲームで偶数と奇数が必ず交互に出る
 - 強力な装備が入手できるカジノで**838861枚のコインが4G(ゴールド)**で買ってしまう



目次

- ◎ コンピュータの歴史
- ◎ コンピュータゲームの歴史
- ◎ テストの考え方の歴史の変遷
- ◎ テスト技法の歴史
- ◎ テスト技術の研究動向
- ◎ まとめ



テストの考え方の歴史的変遷 (1/3)

◎ ソフトウェアテスト前史

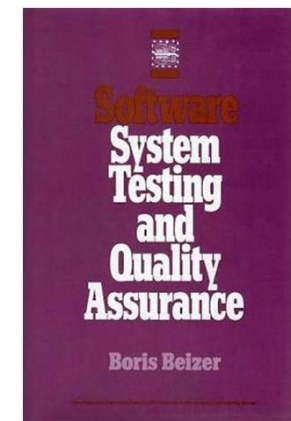
❖ 最初のテスターは？

Boris Beizerによると・・・

- 『テストやデバッグの最初の議論はAdaのメモに遡る』
- Software System Testing and Quality Assurance [Beizer,1984]

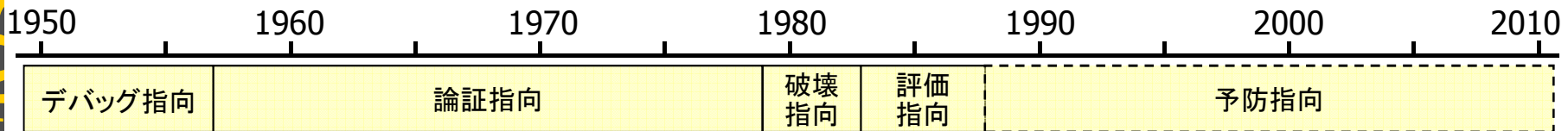
Ada Lovelace女史はプログラマの第1号として、19世紀、Charles Babbageの壮大にして未完成に終わったコンピュータのプログラム作成にたずさわったが、「あと1週間あれば大丈夫です、バベッジ博士。それですべて終わります」と何度も言っていたにちがいない。幸いにもハードウェアが完成しなかったため、Lovelace女史は机上デバッグ以後の段階に進まずにすんだ。

(ボーリス・バイザー,「ソフトウェアテスト技法」,1994,日経BP)





テストの考え方の歴史的変遷 (2/3)



1. **デバッグ指向**の時代(~1956年)
 - デバッグとテストが区別されていなかった時代
2. **論証指向**の時代 (1957年~1978年)
 - デバッグとテストという二つのフェーズを区別
 - テストはプログラムが仕様を満足していることを提示するためのもの
3. **破壊指向**の時代 (1979年~1982年)
 - テストはエラーをみつけるつもりでプログラムを実行する過程<Myers>
4. **評価指向**の時代 (1983年~1987年)
 - ライフサイクルを通じた評価活動の中にテストが位置付けられた
 - ソフトウェアの品質は単一の技法で保証できるものではない
5. **予防指向**の時代 (1988年~)
 - ライフサイクルと並行して進められる予防指向のテストプロセス
 - 開発の上流工程からテスト設計を開始



テストの考え方の歴史的変遷 (3/3)

◎ Beizerの『テスト道』 (電通大・にし先生の命名)

❖ テスターの精神生活 “tester’s mental life”

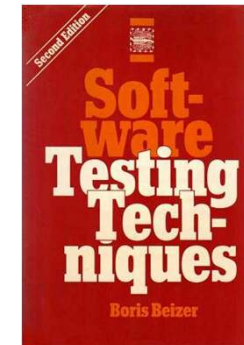
フェーズ0：テストとデバグには何の差もない。デバグ以外にはテストには特別な目的はない。

フェーズ1：テストの目的は、ソフトウェアが動くことを示すことである。

フェーズ2：テストの目的は、ソフトウェアが動かないということを示すことにある。

フェーズ3：テストの目的は、何かを証明することではなく、プログラムが動かないことによって発生する危険性のある許容範囲にまで減らすことである。

フェーズ4：テストは行動ではない。**テストをしないで品質の高いソフトウェアを作るための精神的な訓練**である。



[出典] B. Beizer, "Software Testing Techniques, 2nd Ed.," 1990

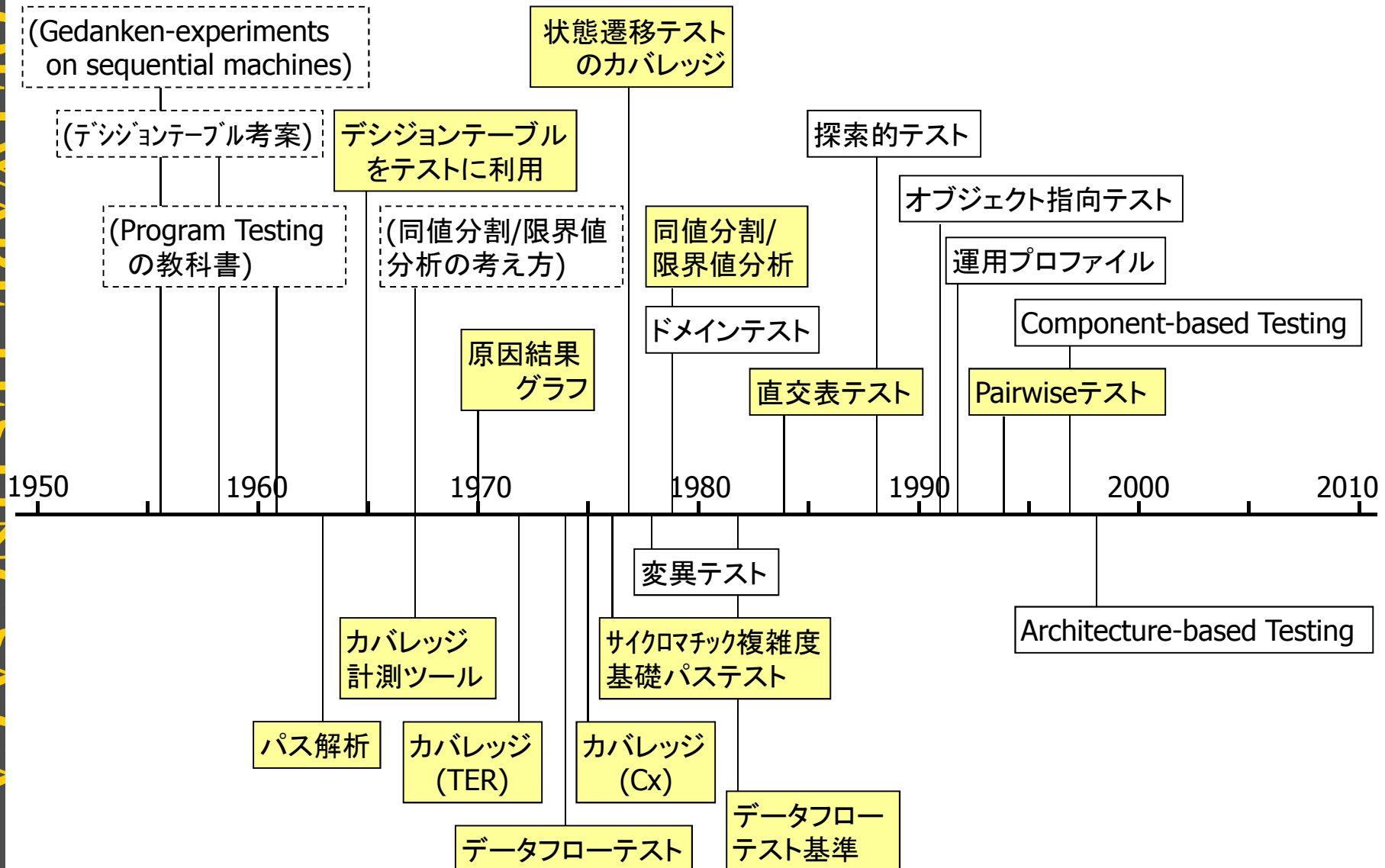


目次

- ◎ コンピュータの歴史
- ◎ コンピュータゲームの歴史
- ◎ テストの考え方の歴史の変遷
- ◎ **テスト技法の歴史**
- ◎ テスト技術の研究動向
- ◎ まとめ



テスト技法の歴史





デシジョンテーブルテスト

- ◎ プログラムの仕様を条件部とアクション部からなるデシジョンテーブルに整理してテストケースを作成

<歴史>

- ❖ 1958年：システム設計の道具としてGE社やSutherland社でデシジョンテーブルを考案
- ❖ 1960年：GE社はデシジョンテーブルから直接ソースコードを生成するプログラム言語TABSOLを開発
- ❖ 1965年：RCA社のScheffがデシジョンテーブルで表現したテストケースを入力とするテストツールを発表
- ❖ 1975年：Goodenough & Gerhartはプログラム構造に基づくテストをデシジョンテーブルで設計する方法 (Condition table method)を提案

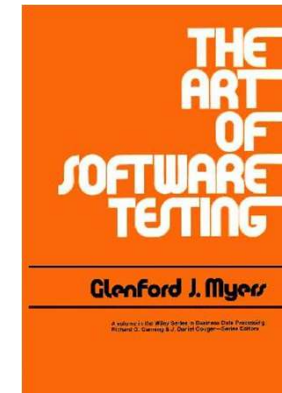


同値分割、限界値分析

- ◎ 入力データを「同じ出力結果が得られるグループ」に分割して各グループ(同値クラス)の代表値、および同値クラスの端に着目してテストケースを作成

<歴史>

- ❖ 同値分割、限界値分析という名称でテスト技法として確立したのは、1979年のIBMのMyersの"ソフトウェア・テストの技法"
 - "同値"という概念や用語は数学からと思われる
- ❖ Myers以前から考え方はあった
 - 1967年のElmendorfの論文には、ソフトウェアの機能のバリエーションを変数とその状態で定義し、状態として正常値、最大値、最小値、それらを越える値を含める考え方が書かれている





原因結果グラフ

- ◎ 外部仕様を、原因(入力条件)と結果(出力や状態)の論理関係、および原因間の関係(制約条件)を示すグラフで表現し、それにもとづいてテストケースを作成

<歴史>

- ❖ Myersの"ソフトウェア・テストの技法"(1979年)で紹介されたことにより広く知られるようになった
- ❖ 技法を考案したのはElmendorf(1970年頃)
 - スイッチング回路の故障点の発見のための手法を元に、制約条件の考え方を追加してソフトウェアの機能を記述できるようにした
 - 原因結果グラフからテストケースを作成するツール
TELDAP(TEst Library Design Automation Program)を開発。ハードウェアの論理回路のテストパターン作成ツールを元に作成



状態遷移テスト

- ◎ プログラムの仕様を状態遷移図や状態遷移表を使って整理し、それにもとづいてテストケースを作成

<歴史>

- ❖ 状態遷移図や状態遷移表は、元来は有限オートマトン(有限状態機械:FSM)の仕様を記述するためのもの
 - デジタル回路の設計ソフトウェア、コンパイラのための字句解析(言語理論)、通信プロトコルのような有限状態をとるシステムのチェックをするソフトウェアなどの分野に用いられている計算モデル
- ❖ 1956年のMooreの論文"Gedanken-experiments on sequential machines"(順序機械の思考実験)がFSMベースのテスト生成の研究の先駆け
- ❖ 1978年にChowが状態遷移テストにおけるカバレッジ基準"n-switch cover"を提案



直交表/Pairwiseテスト

- ◎ 任意の二つ(またはn個)の因子間で組み合わせが網羅されるようにテストケースを作成

<歴史>

- ❖ 実験計画法(直交表)は1920年代に英国のFisherが農業実験の合理化のために開発した手法。これをもとに田口玄一博士が品質工学(タグチメソッド)を開発
- ❖ テストへの直交表の適用は、1983年に富士通で開始、同時期に米国でもMandlがAdaコンパイラのテストに適用
- ❖ 米国で活用されるようになったのは、AT&T社(Bell Labs, Bellcore)の成果が発表された1990年代半ばから。
 - 1992年に直交表によるテスト設計ツールOATSを開発
 - 1994年に制約条件を考慮した組み合わせ表ツールCATSを開発
 - 1994年に逐次的な手法によるPairwise生成ツールAETG を商用化



制御フローテスト (1/2)

- ◎ プログラムの制御の流れをフローグラフで表現し、基準に従ってグラフを網羅するテストケースを作成

<歴史>

❖ グラフ理論の適用

- 1960年にソフトウェア開発(コンパイラ)へグラフ理論を適用(Karp)
- 1963年にテスト設計へ適用。フローチャートをグラフ化し、ブール行列に変換後、分岐点に着目してテストを設計(MillerとMaloney)
- 1976年にMcCabeがグラフ理論を元にプログラムの複雑度を表すサイクロマチック複雑度、基礎パステスト法を提案

❖ カバレッジの計測

- 1960年代前半には意図したルートが通過したかどうかの計測開始
- IBM社のPoughkeepsie地区でコードカバレッジ・ツールを開発
 - COBOLとFORTRANソースプログラムを対象にしたハードウェアの命令語モニタ[Warner,1964]
 - ステートメントカバレッジとブランチカバレッジアナライザ[Hirsh,1967]



制御フローテスト (2/2)

<歴史> (つづき)

❖ カバレッジ基準

- TER (Test Effectiveness Ratio:テスト有効度) [Brown,1972]
 - TERとして命令網羅、分岐網羅を定義
- C0, C1, C2, ... [Miller,1975]
 - 1975年時点では、
C0 : プログラムの直感、C1 : 命令網羅、C2 : 判定条件網羅、...
 - 1977年頃に、
C0 : 命令網羅、C1 : 判定条件網羅、C1p : 判定条件/条件網羅、
C2 : C1基準+ループ網羅、...



データフローテスト

- ◎ プログラムの制御フローを元に、変数の定義、変数の使用の関係に着目してパスを選択することによりテストケースを作成

<歴史>

- ❖ 1960年後半、コンパイラ分野でプログラム最適化技術としてデータフロー解析の研究が進められた(IBM社のAllenら)
- ❖ 1974年の米国コロラド大学のOsterweilとFosdickのデータフローテストの論文がテスト技術としては最初と思われる
- ❖ その後、データフローテスト基準の提案や制御フローテストも含めた構造テストの基準間の包含関係の研究が進められている(RappsとWeyuker,1982 など)



テスト技法の発想のベース

◎ 様々な分野のアイデアの活用

テスト技法	ベース技術の分野
デシジョンテーブルテスト	システム設計技術
同値分割、限界値分析	経験則、数学
原因結果グラフ	ハードウェア論理回路の試験技術
状態遷移テスト	有限オートマトン
直交表/Pairwiseテスト	実験計画法(統計学)、数学
制御フローテスト	グラフ理論、ハードウェアモニター
データフローテスト	コンパイラのプログラム最適化技術



目次

- ◎ コンピュータの歴史
- ◎ コンピュータゲームの歴史
- ◎ テストの考え方の歴史の変遷
- ◎ テスト技法の歴史
- ◎ **テスト技術の研究動向**
- ◎ まとめ



テスト技術の研究動向

◎ A. Bertolino, "Software Testing Research: Achievements, Challenges, Dreams," 2007

- 2007年 ソフトウェア工学国際会議(29th ICSE) Future of Software Engineering trackの論文
- Bertolinoはソフトウェアエンジニアリング基礎知識体系(SWEBOK)の5章 Testingの執筆者
- ❖ **Achievements** : 達成できたテーマ
- ❖ **Challenges** : 夢の実現に向けた研究課題
- ❖ **Dreams** : 4つの夢



テスト技術の6つの側面

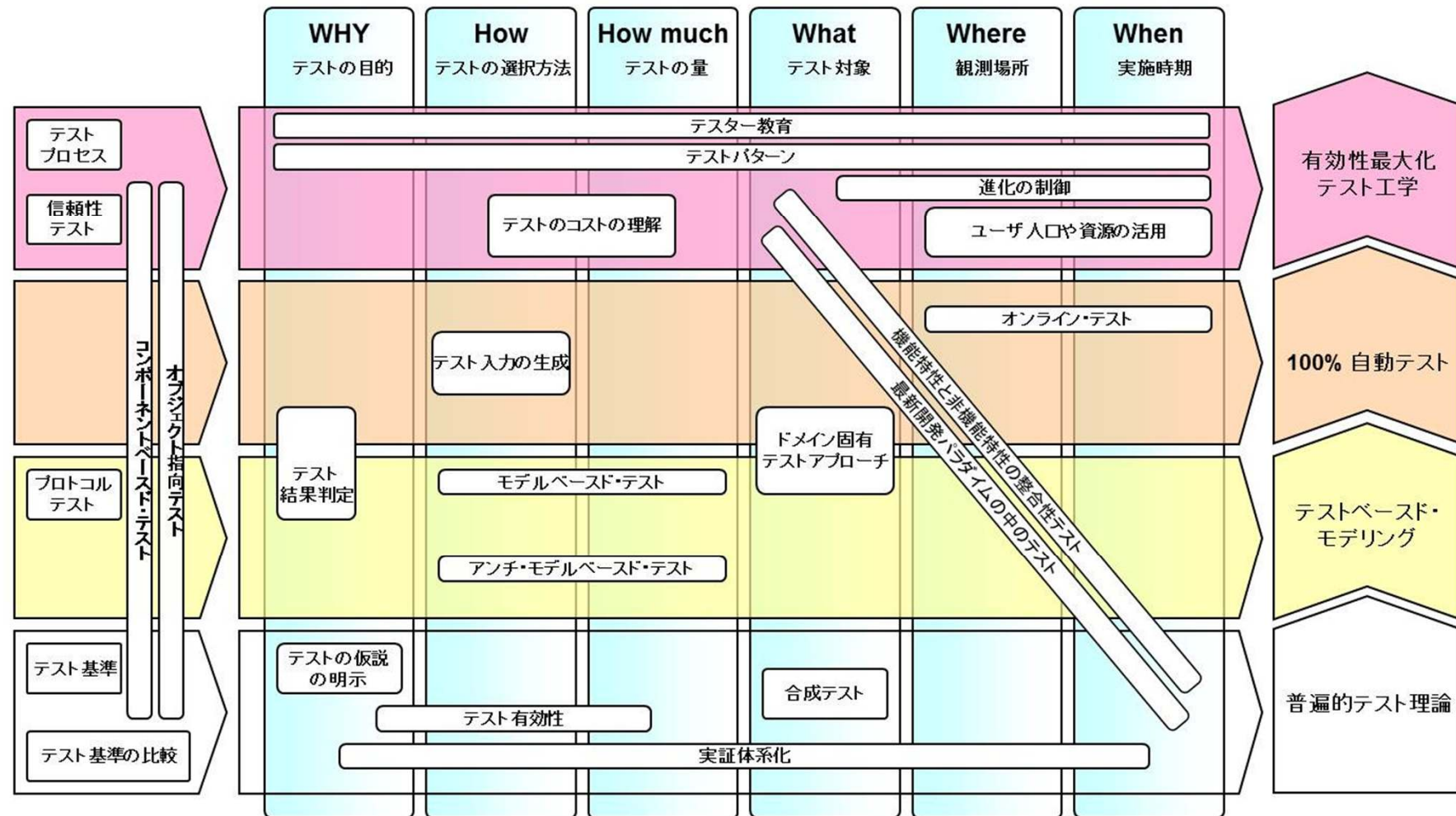
- ◎ テストは
 - ❖ 実行サンプルの観測と結果の判定
- ◎ テスト技術を特徴づける6つの側面
 - ❖ **WHY : テストの目的**
バグの検出、製品の出荷判断、UIのユーザビリティ評価、...
 - ❖ **WHAT : テスト対象**
ユニット、コンポーネント／サブシステム、システム(全体)
 - ❖ **HOW : テストの選択方法**
アドホック、ランダム、系統的な方法
 - ❖ **WHERE : 観測場所**
社内(in house)、疑似環境、実環境
 - ❖ **HOW MUCH : テストの量**
テスト充分性、網羅度合い、信頼性尺度
 - ❖ **WHEN : 実施時期**
ライフサイクルのどの時点で



ソフトウェアテストの研究ロードマップ

Software testing research roadmap

[出典] A. Bertolino, "Software Testing Research: Achievements, Challenges, Dreams," FOSE'07, p.85-103, 2007, Figure 1



Achievements

Challenges

Dreams



達成できたテーマ

- ◎ テストプロセス (Testing process)
- ◎ テスト基準 (Test criteria)
- ◎ テスト基準の比較 (Comparison among test criteria)
- ◎ オブジェクト指向テスト (Object-oriented testing)
- ◎ コンポーネントベースド・テスト (Component-based testing)
- ◎ プロトコル・テスト (Protocol testing)
- ◎ 信頼性テスト (Reliability testing)



4つの夢と研究課題

(1) 普遍的テスト理論 (Universal test theory)

- ❖ テスト技術の支援と育成に役立つ包括的な理論

[研究課題]

テストの仮説の明確化、テスト有効性、合成テスト、実証データ体系化

(2) テストベースド・モデリング (Test-based modeling)

- ❖ 効果的にテストできるモデルを開発者が考慮(cf. テスト容易化設計)

[研究課題]

モデルベースド・テスト、アンチ・モデルベースド・テスト、テスト結果判定

(3) 100%自動テスト (100% automatic testing)

- ❖ テスト入力の生成手法、テストプロセス自動化手順、統合テスト環境

[研究課題]

テスト入力の生成、ドメイン固有テストアプローチ、オンライン・テスト

(4) 有効性最大化テスト工学 (Efficacy-maximized test engineering)

- ❖ テスト手法、ツール、プロセスに関する費用対効果の高い工学技術

[研究課題]

ソフトウェアの進化の制御、利用人口・資源の活用、テストパターン、テストのコストの理解、テスター教育



まとめ

◎ テストの考え方の進化

テストの考え方の歴史的な変遷は単なる歴史ではなく、テストや品質保証の組織、品質マネジメントの成長過程として参考になるのではないのでしょうか。

◎ テスト技術の進化

テスト技術は開発技術のみならず、様々な分野の技術を学び、活用することで進化してきました。テストは言わば**”知恵の総合格闘技”**です。

先人がそうであったように、**遊び心(エンターテイメント)**を**忘れず**、新しいテスト技術の歴史をつくっていきましょう。

ご清聴ありがとうございました



画像の引用元について

- ◎ 本資料では以下のサイトの画像を利用させていただきました。

チャールズ・バベッジ

<http://ja.wikipedia.org/wiki/チャールズ・バベッジ>

解析機関

<http://ja.wikipedia.org/wiki/解析機関>

エイダ・ラブレス

<http://ja.wikipedia.org/wiki/エイダ・ラブレス>

Atanasoff-Berry Computer

http://en.wikipedia.org/wiki/Atanasoff-Berry_Computer

ENIAC

<http://en.wikipedia.org/wiki/ENIAC>

EDSAC

<http://www.cl.cam.ac.uk/conference/EDSAC99/reminiscences/>

レオナルド・トーレス・ケベード(チェス機械: エル・アヘドレシスタ)

<http://ja.wikipedia.org/wiki/レオナルド・トーレス・ケベード>

Graphic Tic Tac Toe

<http://soratobucan.blog60.fc2.com/blog-entry-168.html>

PDP-1

<http://ja.wikipedia.org/wiki/PDP-1>

The Spacewar! Java applet

<http://en.wikipedia.org/wiki/Spacewar!>

PDP-7

<http://research.microsoft.com/en-us/um/people/gbell/Digital/timeline/1964-3.htm>

Space Travel

<http://www.uvlist.net/game-164857-Space+Travel>

コンピュータースペース

<http://ja.wikipedia.org/wiki/コンピュータースペース>

ポン (ゲーム)

[http://ja.wikipedia.org/wiki/ポン \(ゲーム\)](http://ja.wikipedia.org/wiki/ポン_(ゲーム))

スペースインベーダー

<http://dankai-j.jugem.jp/?eid=113>