ソフトウェア開発現代史: 55%が変化に備えていない現実 — AI支援型開発時代のReboot Japan

Agile Japan Day2 [2025/11/14 金曜日]

ファインディ株式会社 CTO室 高橋 裕之

1989年より組込みエンジニアとして、電話網の交換機開発携わり国産OS CTRONをスクラッチで開発。また、BSD UNIXベースのTCP/IPプロトコル開発に従事。その後、メーカーでRTOSや組込みLinuxを基盤としたテレビ、デジタルカメラ、ビデオカメラなどの開発に16年携わる。
2005年からソフトウェア開発で起きるさまざまな問題に向き合うことを決意しSPI（ソフトウェアプロセス改善）の専門家へ転身。
問題を抱える開発チームに向き合いながら、計測エビデンスをもとにしたケイパビリティモデルに基づくプロセス改善活動を得意とする。
Findy Tech Blog 編集長。

ファインディ株式会社 CTO室
Staff Engineer (Engineering Excellence),
SPI Coach, Agile Coach

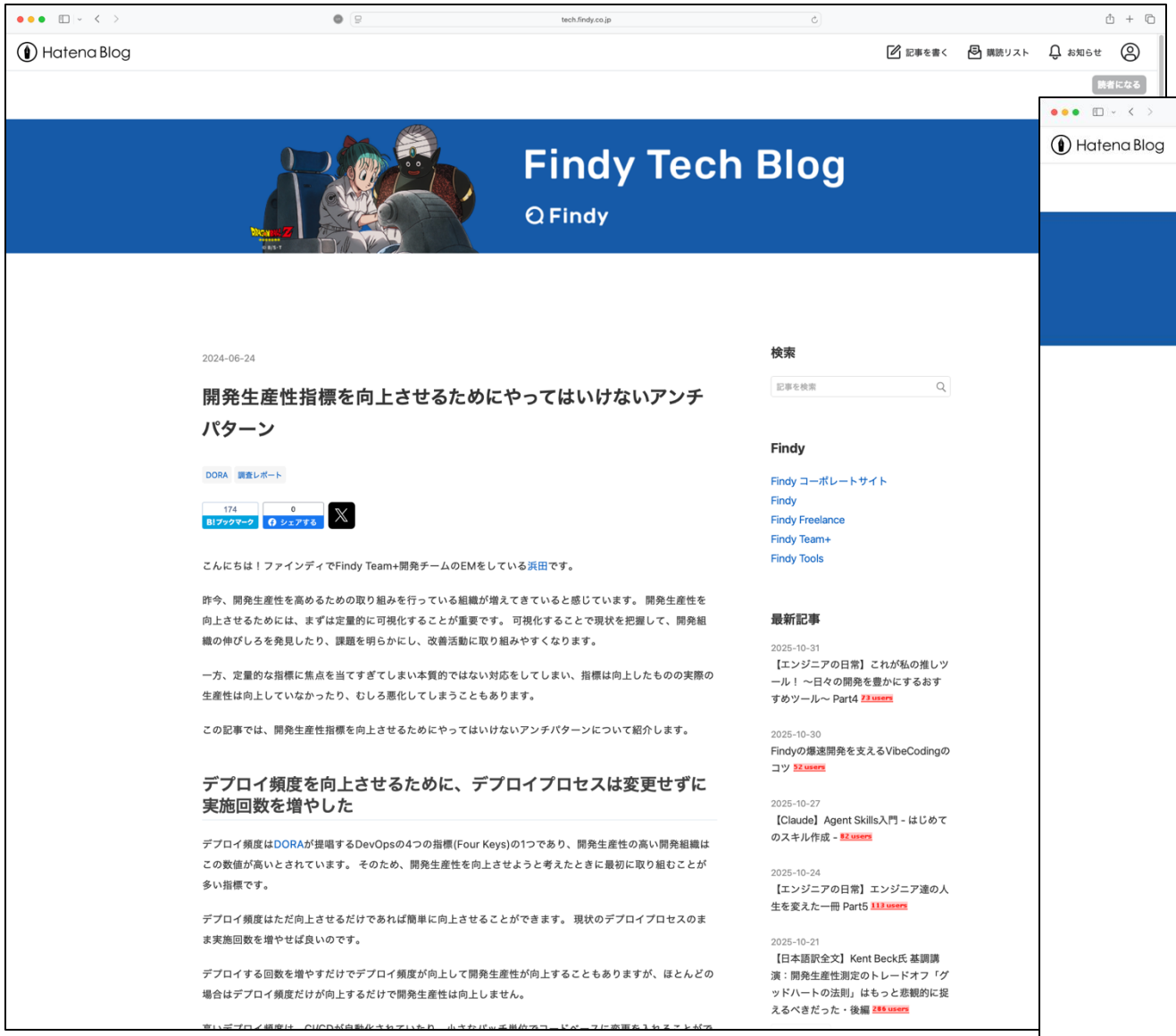
高橋 裕之 / Hiroyuki Takahashi

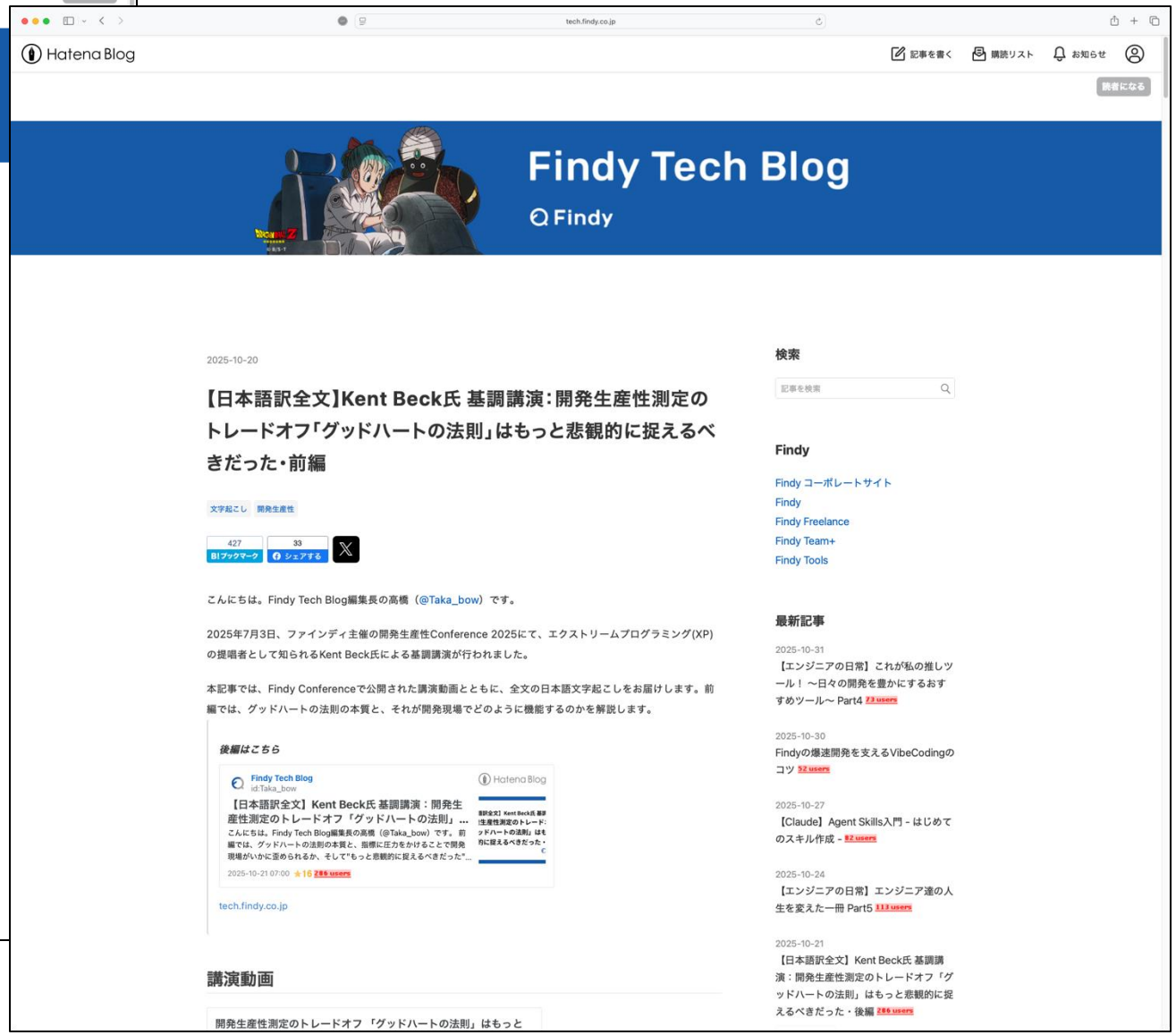
@Taka_bow
takabow
hiroyukitakah

- 1989年 株式会社ジェーシーイ
- 1993年 フリーランス
- 1995年 株式会社メイテック
- 1996年 日立通信システム株式会社（現・株式会社日立情報通信エンジニアリング）
- 2002年 ソニーデジタルネットワークアプリケーションズ株式会社
  コンシューマ向けカメラ開発、組織横断型プロセス改善（SEPG）に従事
- 2013年 ウイングアーク１ｓｔ株式会社
  プロセス改善コーチ兼アジャイルコーチとして活動。2018年よりソフトウェアプロセス＆品質改善部部長、製品品質管理責任者、オープンソース管理責任者を務める。
- 2021年 株式会社ビズリーチ
  プロセス改善コーチ兼アジャイルコーチとして活動。QAとプロセス改善部門のマネジャーとして、DevOpsアプローチによる開発透明性向上とDORA(Four Keys)メトリクスを用いた開発生産性による改善活動を実施。SODA（Software Outcome Delivery Architecture）構想を立案し、プロダクトの事業影響を定量化・可視化する組織変革をリード。
- 2024年 ファインディ株式会社
  ソフトウェアプロセス改善コーチ兼アジャイルコーチとして活動。
  エンジニア組織への新技術・方法論導入支援のイネーブルメントを担当。

Findy

Findy Tech Blog 編集長 - https://tech.findy.co.jp



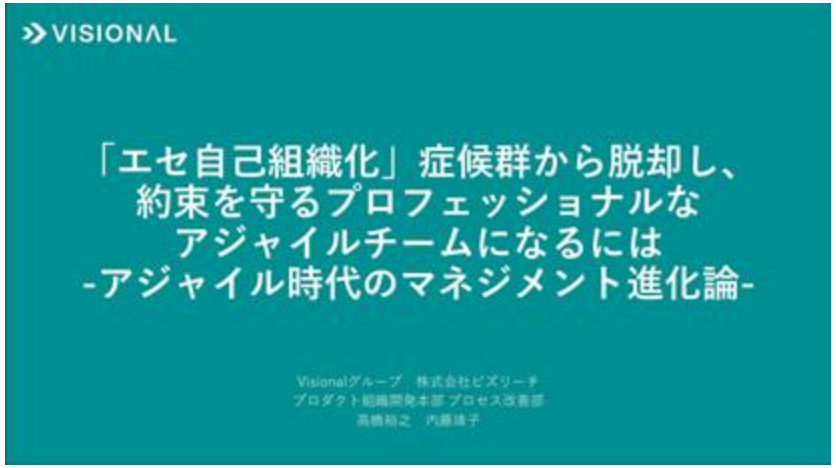


※画像は『ドラゴンボールZ』とのコラボが行われている2025年10月時点のものである © B/S・T

Portfolio

https://speakerdeck.com/takabow/ji-hua-hashu-kufalsedehanakuli-terumofalse

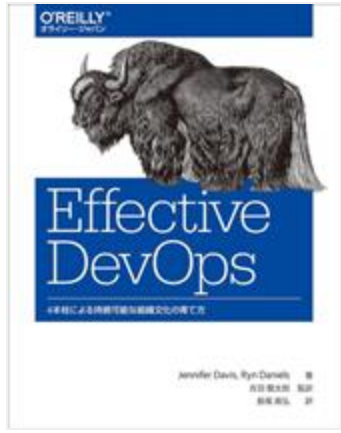
https://speakerdeck.com/takabow/wen-hua-de-fu-zhai-tofalsezhan-i-lao-pu-sohutoueakai-fa-hui-she-deaziyairubian-ge-woshi-gua-keta8nian-jian



https://speakerdeck.com/visional_engineering_and_design/number-rsgt2023

https://speakerdeck.com/visional_engineering_and_design/devopsdays-tokyo-2024

https://speakerdeck.com/takabow/zhi-zao-ye-tosohutoueahaben-dang-nigong-cun-dekiteitanoka-pin-zhi-tosupidowowen-izhi-su-dc76f0ad-2d4c-4f2a-bf4d-cecbceb7eb5e



技術翻訳レビュー

Findy

会社紹介

会社概要

会社名	**ファインディ株式会社 / Findy Inc.**
代表取締役	**山田 裕一朗**
設立	2014 年 2 月　※ 本格的な事業開始は2016年7月
社員数	**376 名**
資本金	**27億5,386 万円**　※ 資本準備金含む
住所	**東京都品川区大崎1-2-2 アートヴィレッジ大崎セントラルタワー 5階**
事業許可番号	13-ユ-308478
サービス	・IT/Webエンジニアの転職サービス「Findy」 ・ハイスキルなフリーランスエンジニア紹介サービス「Findy Freelance」 ・経営と開発現場をつなぐAI戦略支援SaaS「Findy Team+」 ・開発ツールのレビューサイト「Findy Tools」 ・テックカンファレンスのプラットフォーム「Findy Conference」 ・顧客価値を追求する、AI時代の製品開発マネジメント「Findy Insights」等
投資家	**グローバル・ブレイン、ユナイテッド、SMBCベンチャーキャピタル、KDDI、 JA三井リース、みずほキャピタル、博報堂ＤＹベンチャーズ、Carbide Ventures、等**

経営理念

**つくる人がもっとかがやけば、
世界はきっと豊かになる。**

ビジョン

**挑戦するエンジニアの
プラットフォームをつくる。**

Findy

サービス紹介

ファインディが展開するエンジニアプラットフォーム

Findy

Findy Freelance

Findy Team+

Findy Tools

Findy Conference

Findy AI+

Findy AI Career

Findy Insights

挑戦するエンジニアに最新のナレッジを

- 日本に居ながら海外カンファレンス体験を！
  - 開発生産性Conference / アーキテクチャConference / Data Engineering Summit / 内製開発Summit etc...
  - AI Engineering Summit Tokyo

Dr. Nicole Forsgren （開発生産性Conference 2024）


Kent Beck （開発生産性Conference 2025）

Gene Kim （開発生産性Conference 2025）

挑戦するエンジニアに最新のナレッジを

- 日本IT業界に起きている「いまの課題」へ切り込む

Findy

（参考）ソフトウェア開発現代史シリーズ https://speakerdeck.com/takabow

DATE	タイトル	概要	イベント
2024/8/8	モダンソフトウエアエンジニアリングとは（非公開）	モダンなソフトウェアエンジニアリングを導入することの具体的なメリットを把握する	お客様向け講演
2025/1/29	ソフトウェア開発現代史: 製造業とソフトウェアは本当に共存できていたのか？品質とスピードを問い直す	戦後日本の製造業とソフトウェアエンジニアリングの歴史を振り返り、世界と日本の進化になぜ差ついたのかを理解する	【ヤマハ発動機×SUBARU×三菱電機】今こそ考えたい「開発プロセスの品質視点」
2025/3/28	ソフトウェア開発現代史: なぜ日本のソフトウェア開発は「滝」なのか？製造業の成功体験とのギャップ	なぜ日本のソフトウェア開発は「ウォーターフォール」を受け入れ続けているのか？の謎に迫る	JaSST'25 Tokyo Day2
2025/4/15	ソフトウェア開発現代史: "LeanとDevOpsの科学"の「科学」とは何か？ - DORA Report 10年の変遷を追って -	DevOpsが未定着な日本の謎とDORA10年の歴史を振り返り、ソフトウェア開発マネージメントに持ち込まれた「科学的」とは何か？を解説	DevOpsDays Tokyo 2025 Day 1
2025/7/4	ソフトウェア開発現代史: 日本のソフトウェア工学が直面する、ハードウェアの進化に隠された「二重停滞」	CDの先駆者デイヴィッド・ファーリーは「ソフトウェア産業は学ぶことも進化することもなかなかできないで苦闘している」と指摘。この問題は日本においてさらに深刻です	開発生産性Conference 2025
2025/11/6	ソフトウェア開発現代史: 内製化はなぜ難しいのか – ウォーターフォールから生成AIまでに失われた土台	日本のIT業界では、多重請負構造を背景に、アジャイルやDevOpsといった“土台”を十分に築かないまま生成AI時代へ突入しました。そのため内製化の試みは、スケールや定着の壁に直面しやすい状況にあります	Qiita Conference 2025 Autumn
2025/11/7	ソフトウェア開発現代史: 選任QAの昔と今 –「私 vs あなた」から共に担う品質へ	かつてソフトウェア開発の現場では、「私（QA） vs あなた（開発）」という構図が当たり前でした。QAはゲートキーパーとして品質を守り、ときには開発者の背後に立ち作業を監視するような雰囲気すら存在していました	開発をリードする品質保証 - Findy Online Conference -
※本日※	ソフトウェア開発現代史: 55%が変化に備えていない現実 – AI支援型開発時代のReboot Japan	AI支援型開発時代を迎えた今、日本の多くの組織はその土台づくりに出遅れています。これはAI支援型開発の波に乗り遅れ、将来的に大きな生産性格差を生むリスクを意味します。	Agile Japan 2025

Findy

ソフトウェア開発現代史 -Beginning-

ソフトウェア関連の論文・文献

ICTの進化

西暦

主な出来事

1970 1976 1981 1987 1989/02/01 1994/10/31 1996/8 2001/5/18 2003/9/1 2005/10 2009/11/27 2011/7/16 2014/8/18 2017/6/22 2019/9/17 2021/3/15 2023/11/21 2025/6/25

1977 1979 1984 1988/3/1 1995/10/1 2000/12/1 2003/9/1 2005/10/7 2008/12/30 2010/10/12 2013/1/10 2016/10/6 2018/4/1 2019/9/30 2021/12/1 2023/11/28 2024/12/25

1978/5/1 1986 1989 1995 1999 2002 2004/11/16 2010/7/27 2012/3/15 2013/8/1 2018/3/27 2021/8/1 2022/12/17 2024/7/11

1975 1978 1979 1982 1990/10/10 1992/3/13 1996/4/1 2002/11/8 2004/7/16 2006/10/19 2010～ 2011/10/18 2014/4/22 2017/10/14 2018/11/22 2021/12/10 2024/9/13

パソコン通信 | 携帯電話・PHSの普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：
IBM System 360とVT100（1970～1980）

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始によるポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代iMac
（1998～）

F501i HYPER iモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、ブロードバンド時代に突入
（2001～）

初代iPhone発表
（2007/1/9）

初代iPad発表
（2010/4/3）

Samsung Galaxy S II
（2011/5/2）

Netflixが日本に上陸(2015/9/2)
※🇺🇸では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max
（2024）

50年前 40年前 30年前 20年前 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

1974年：🇺🇸「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：🇺🇸米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年～：🇺🇸米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：🇺🇸米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年～：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化

1970～1980年代：🌎多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：🇺🇸ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代～：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代～：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3～2023/5：🌎COVID-19

1970年代後半～1980年代末：🇯🇵日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：🇺🇸日本を含む諸外国へ「通商法スーパー301条」発動

2000/3～：🇺🇸ITバブル崩壊

2010年以降～：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979～：🇺🇸日本の製造業の高品質、ものづくりの強さを研究。

1991～：🇯🇵バブル崩壊「失われた○○年」

2001/2/11：🌎アジャイルソフトウェア開発宣言

2000年代後半～：🌎クラウドファースト・クラウドネイティブ時代に突入

2021～：🌎AI（GenAI）が前提の時代へ

2008/9～：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半～1990年代前半：🌎UNIX戦争

1985～1990：🇯🇵国家プロジェクト「Σ計画」が頓挫

2009：🇺🇸Flickrのエンジニアである John Allspaw と Paul Hammond が初めてDevOpsに繋がる伝説的な講演を行う

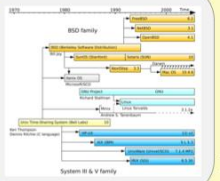
© The Deming Institute


“トリニトロン”
日本製品が欧米で人気


"ウォークマン"
1号機 TPS-L2
（1979）


Toyota Corolla
Liftback SR5 001
（1980～）


世界初のポータブル
CDプレーヤー
D-50（1984）



Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
（1996）


Bugzilla
リリース
（2000）

JIRA
Jira
リリース
（2002）


Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
（1995）


Subversionリリース
（2000）


CVS 誕生
（1990）

1990～2000：🌎第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

flickr
リリース
（2004/2）

aws
AWS
サービス開始
（2006）

Google Cloud
サービス開始
（2008）

Azure
サービス開始
（2010）

Selenium
（2004）

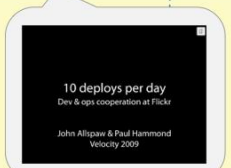
REDMINE
flexible project management
（2006）

trac
（2004）

git
（2005）

GitHub
リリース
（2008）

AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）



GitLab
（2011）


Jenkins 誕生
（2011）

2009～2020：🌎第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

GitHub Actions
（2018）

DATADOG
（2018）

Findy Team+
Team+
（2021/10）

Claude 1
（2023/3/14）

Cline
（2024/7）

GitHub Copilot
（2021/6/29）

Gemini 1
（2023/12/6）

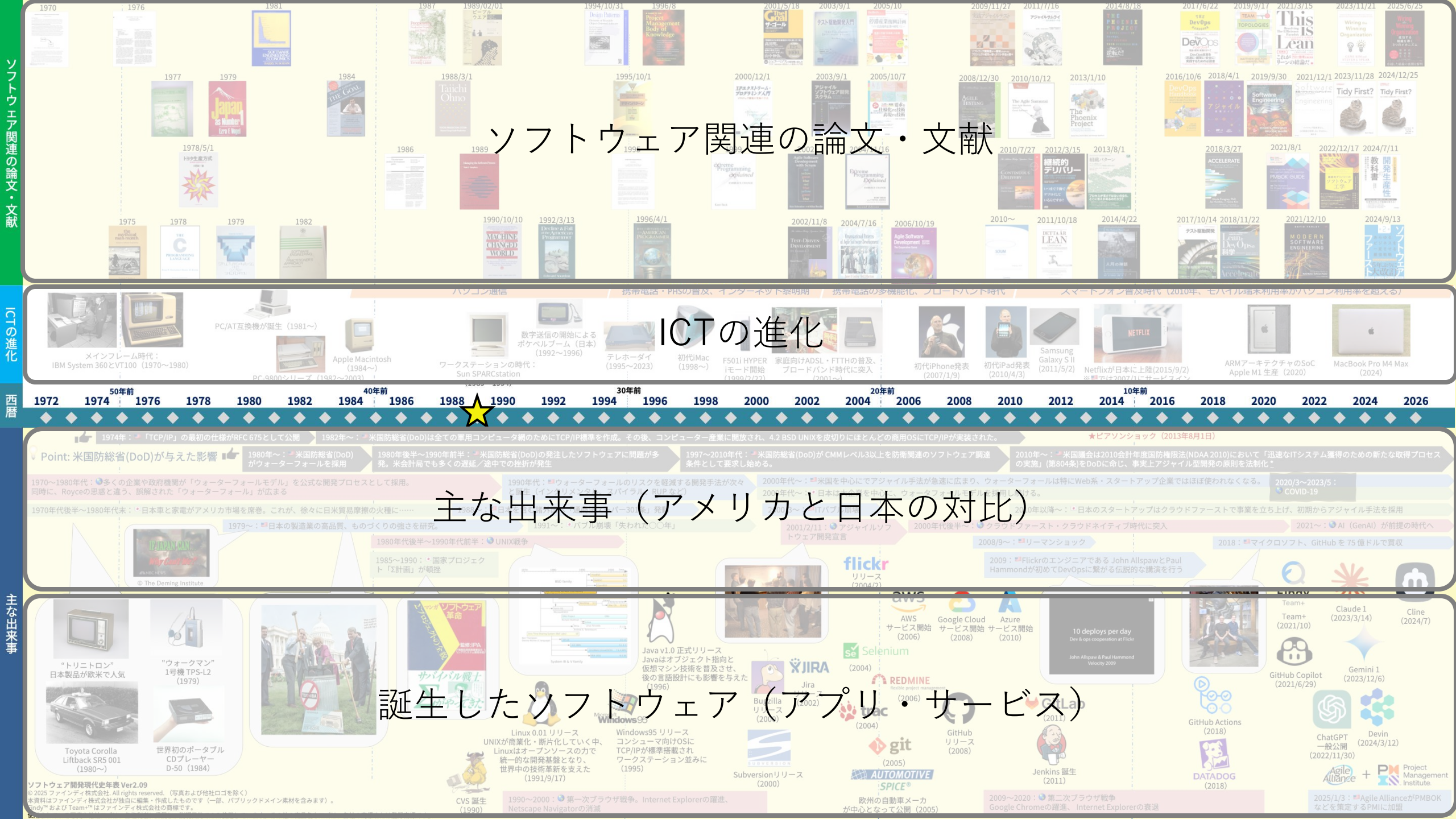
ChatGPT
一般公開
（2022/11/30）

Devin
（2024/3/12）

Agile Alliance + Project Management Institute

2025/1/3：🇺🇸Agile AllianceがPMBOKなどを策定するPMIに加盟

ソフトウェア開発現代史年表 Ver2.09


ソフトウェア関連の論文・文献
ICTの進化
主な出来事（アメリカと日本の対比）
誕生したソフトウェア（アプリ・サービス）
西暦
1972
1974
1976
1978
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022
2024
2026
50年前
40年前
30年前
20年前
10年前

ソフトウェア関連の論文・文献

ICTの進化

西暦

主な出来事

1970 1976 1981 1987 1989/02/01 1994/10/31 1996/8 2001/5/18 2003/9/1 2005/10 2009/11/27 2011/7/16 2014/8/18 2017/6/22 2019/9/17 2021/3/15 2023/11/21 2025/6/25

1977 1979 1984 1988/3/1 1995/10/1 2000/12/1 2003/9/1 2005/10/7 2008/12/30 2010/10/12 2013/1/10 2016/10/6 2018/4/1 2019/9/30 2021/12/1 2023/11/28 2024/12/25

1978/5/1 1986 1989 1995 1999 2002 2004/11/16 2010/7/27 2012/3/15 2013/8/1 2018/3/27 2021/8/1 2022/12/17 2024/7/11

1975 1978 1979 1982 1990/10/10 1992/3/13 1996/4/1 2002/11/8 2004/7/16 2006/10/19 2010～ 2011/10/18 2014/4/22 2017/10/14 2018/11/22 2021/12/10 2024/9/13

パソコン通信 | 携帯電話・PHSの普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：
IBM System 360とVT100（1970～1980）

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始によるポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代iMac
（1998～）

F501i HYPER iモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、ブロードバンド時代に突入
（2001～）

初代iPhone発表
（2007/1/9）

初代iPad発表
（2010/4/3）

Samsung Galaxy S II
（2011/5/2）

Netflixが日本に上陸(2015/9/2)
※🇺🇸では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max
（2024）

50年前 40年前 30年前 20年前 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

1974年：🇺🇸「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：🇺🇸米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年～：🇺🇸米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：🇺🇸米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年～：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化

1970～1980年代：🌎多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：🇺🇸ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代～：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代～：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3～2023/5：🌎COVID-19

1970年代後半～1980年代末：🇯🇵日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：🇺🇸日本を含む諸外国へ「通商法スーパー301条」発動

2000/3～：🇺🇸ITバブル崩壊

2010年以降～：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979～：🇺🇸日本の製造業の高品質、ものづくりの強さを研究。

1991～：🇯🇵バブル崩壊「失われた○○年」

2001/2/11：🌎アジャイルソフトウェア開発宣言

2000年代後半～：🌎クラウドファースト・クラウドネイティブ時代に突入

2021～：🌎AI（GenAI）が前提の時代へ

2008/9～：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半～1990年代前半：🌎UNIX戦争

1985～1990：🇯🇵国家プロジェクト「Σ計画」が頓挫

2009：🇺🇸Flickrのエンジニアである John Allspaw と Paul Hammond が初めてDevOpsに繋がる伝説的な講演を行う

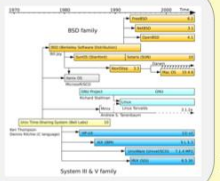
© The Deming Institute


“トリニトロン”
日本製品が欧米で人気


"ウォークマン"
1号機 TPS-L2
（1979）


Toyota Corolla
Liftback SR5 001
（1980～）


世界初のポータブル
CDプレーヤー
D-50（1984）



Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
（1996）


Bugzilla
リリース
（2000）

JIRA
Jira
リリース
（2002）


Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
（1995）


Subversionリリース
（2000）


CVS 誕生
（1990）

1990～2000：🌎第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

flickr
リリース
（2004/2）

aws
AWS
サービス開始
（2006）

Google Cloud
サービス開始
（2008）

Azure
サービス開始
（2010）

Selenium
（2004）

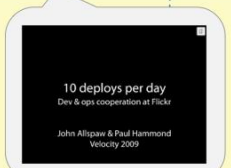
REDMINE
flexible project management
（2006）

trac
（2004）

git
（2005）

GitHub
リリース
（2008）

AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）



GitLab
（2011）


Jenkins 誕生
（2011）

2009～2020：🌎第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

GitHub Actions
（2018）

DATADOG
（2018）

Findy Team+
Team+
（2021/10）

Claude 1
（2023/3/14）

Cline
（2024/7）

GitHub Copilot
（2021/6/29）

Gemini 1
（2023/12/6）

ChatGPT
一般公開
（2022/11/30）

Devin
（2024/3/12）

Agile Alliance + Project Management Institute

2025/1/3：🇺🇸Agile AllianceがPMBOKなどを策定するPMIに加盟

ソフトウェア開発現代史年表 Ver2.09

西暦

50年前 / 40年前 / 30年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000

主な出来事

1974年：「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4

Point: 米国防総省(DoD)が与えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：米国
条件として要求し始める。

1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

200
200

1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

2

1979～：日本の製造業の高品質、ものづくりの強さを研究。

1991～：バブル崩壊「失われた○○年」

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

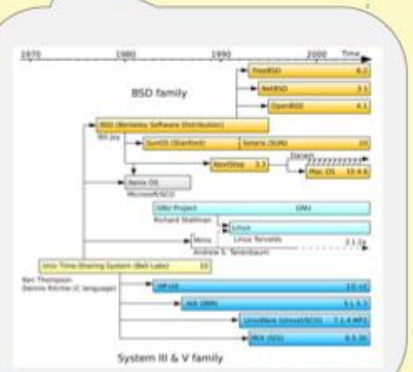
© The Deming Institute


“トリニトロン”
日本製品が欧米で人気


"ウォークマン”
1号機 TPS-L2
（1979）


Toyota Corolla
Liftback SR5 001
（1980～）


世界初のポータブル
CDプレーヤー
D-50（1984）




Java v1.0 正式リリース
Javaはオブジェクト指向と
仮想マシン技術を普及させ、
後の言語設計にも影響を与えた
（1996）


Bugzi
リリー
（200


Linux 0.01 リリース
UNIXが商業化・断片化していく中、
Linuxはオープンソースの力で
統一的な開発基盤となり、
世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSに
TCP/IPが標準搭載され
ワークステーション並みに
（1995）


CVS 誕生
（1990）

Subversion
（200

1990～2000：第一次ブラウザ戦争。Internet Explorerの躍進、
Netscape Navigatorの消滅

ソフトウェア開発現代史年表 Ver2.09

20年前　10年前

1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

ューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

1997～2010年代：米国防総省(DoD)が CMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年～：米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化 *

る開発手法が次々

2000年代～：米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代～：日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3～2023/5：COVID-19

2000/3～：ITバブル崩壊

2010年以降～：日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

2001/2/11：アジャイルソフトウェア開発宣言

2000年代後半～：クラウドファースト・クラウドネイティブ時代に突入

2021～：AI（GenAI）が前提の時代へ

2008/9～：リーマンショック

2018：マイクロソフト、GitHub を 75 億ドルで買収

2009：Flickrのエンジニアである John AllspawとPaul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

式リリース
ジェクト指向と
技術を普及させ、
計にも影響を与えた

flickr リリース (2004/2)

aws AWS サービス開始 (2006)

Google Cloud サービス開始 (2008)

Azure サービス開始 (2010)

Findy Team+ Team+ (2021/10)

Claude 1 (2023/3/14)

Cline (2024/7)

Bugzilla リリース (2000)

JIRA Jira リリース (2002)

Selenium (2004)

REDMINE flexible project management (2006)

GitHub Copilot (2021/6/29)

Gemini 1 (2023/12/6)

trac (2004)

GitHub リリース (2008)

GitLab (2011)

GitHub Actions (2018)

ChatGPT 一般公開 (2022/11/30)

Devin (2024/3/12)

SUBVERSION Subversionリリース (2000)

git (2005)

Jenkins 誕生 (2011)

DATADOG (2018)

AUTOMOTIVE SPICE® 欧州の自動車メーカが中心となって公開（2005）

Agile Alliance + Project Management Institute.

lorerの躍進、

2009～2020：第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

2025/1/3：Agile AllianceがPMBOKなどを策定するPMIに加盟

ソフトウェア関連の論文・文献

ICTの進化

西暦

1970

1976

1981

1987

1989/02/01

1994/10/31

1996/8

2000

1977

1979

1984

1988/3/1

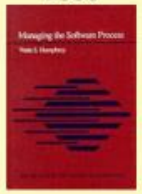
1995/10/1

2000/12/1

1978/5/1

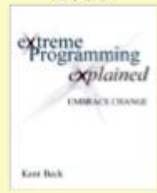
1986

1989

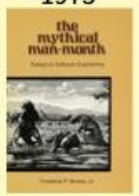


1995

1999



1975



1978

1979

1982

1990/10/10

1992/3/13

1996/4/1

パソコン通信

携帯電話・PHSの普及、インターネッ

メインフレーム時代：
IBM System 360とVT100（1970～1980）

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始による
ポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代iMac
（1998～）

F501i HYPER
iモード開始
（1999/2/22）

50年前　40年前　30年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000


2001/5/18
2003/9/1
2005/10
2009/11/27
2011/7/16
2014/8/18
2017/6/22
2019/9/17
2021/3/15
2023/11/21
2025/6/25
2000/12/1
2003/9/1
2005/10/7
2008/12/30
2010/10/12
2013/1/10
2016/10/6
2018/4/1
2019/9/30
2021/12/1
2023/11/28
2024/12/25
1999
2002
2004/11/16
2010/7/27
2012/3/15
2013/8/1
2018/3/27
2021/8/1
2022/12/17
2024/7/11
2002/11/8
2004/7/16
2006/10/19
2010〜
2011/10/18
2014/4/22
2017/10/14
2018/11/22
2021/12/10
2024/9/13
の普及、インターネット黎明期
携帯電話の多機能化、ブロードバンド時代
スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）
初代iMac
(1998〜)
F501i HYPER
iモード開始
(1999/2/22)
家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
(2001〜)
初代iPhone発表
(2007/1/9)
初代iPad発表
(2010/4/3)
Samsung
Galaxy S II
(2011/5/2)
NETFLIX
Netflixが日本に上陸(2015/9/2)
※🇺🇸では2007/1にサービスイン
ARMアーキテクチャのSoC
Apple M1 生産（2020）
MacBook Pro M4 Max
(2024)
20年前
10年前
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022
2024
2026

ソフトウェア関連の論文・文献

ICTの進化

西暦

主な出来事

1970 1976 1981 1987 1989/02/01 1994/10/31 1996/8 2001/5/18 2003/9/1 2005/10 2009/11/27 2011/7/16 2014/8/18 2017/6/22 2019/9/17 2021/3/15 2023/11/21 2025/6/25

1977 1979 1984 1988/3/1 1995/10/1 2000/12/1 2003/9/1 2005/10/7 2008/12/30 2010/10/12 2013/1/10 2016/10/6 2018/4/1 2019/9/30 2021/12/1 2023/11/28 2024/12/25

1978/5/1 1986 1989 1995 1999 2002 2004/11/16 2010/7/27 2012/3/15 2013/8/1 2018/3/27 2021/8/1 2022/12/17 2024/7/11

1975 1978 1979 1982 1990/10/10 1992/3/13 1996/4/1 2002/11/8 2004/7/16 2006/10/19 2010～ 2011/10/18 2014/4/22 2017/10/14 2018/11/22 2021/12/10 2024/9/13

パソコン通信 | 携帯電話・PHSの普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：
IBM System 360とVT100（1970～1980）

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始によるポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代iMac
（1998～）

F501i HYPER iモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、ブロードバンド時代に突入
（2001～）

初代iPhone発表
（2007/1/9）

初代iPad発表
（2010/4/3）

Samsung Galaxy S II
（2011/5/2）

Netflixが日本に上陸(2015/9/2)
※🇺🇸では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max
（2024）

50年前 40年前 30年前 20年前 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

1974年：🇺🇸「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：🇺🇸米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年～：🇺🇸米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：🇺🇸米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年～：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化

1970～1980年代：🌎多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：🇺🇸ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代～：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代～：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3～2023/5：🌎COVID-19

1970年代後半～1980年代末：🇯🇵日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：🇺🇸日本を含む諸外国へ「通商法スーパー301条」発動

2000/3～：🇺🇸ITバブル崩壊

2010年以降～：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979～：🇺🇸日本の製造業の高品質、ものづくりの強さを研究。

1991～：🇯🇵バブル崩壊「失われた○○年」

2001/2/11：🌎アジャイルソフトウェア開発宣言

2000年代後半～：🌎クラウドファースト・クラウドネイティブ時代に突入

2021～：🌎AI（GenAI）が前提の時代へ

2008/9～：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半～1990年代前半：🌎UNIX戦争

1985～1990：🇯🇵国家プロジェクト「Σ計画」が頓挫

2009：🇺🇸Flickrのエンジニアである John Allspaw と Paul Hammond が初めてDevOpsに繋がる伝説的な講演を行う

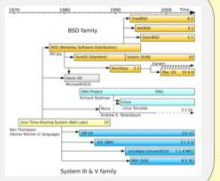
© The Deming Institute


“トリニトロン”
日本製品が欧米で人気


"ウォークマン"
1号機 TPS-L2
（1979）


Toyota Corolla
Liftback SR5 001
（1980～）


世界初のポータブル
CDプレーヤー
D-50（1984）



Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
（1996）


Bugzilla
リリース
（2000）

JIRA
Jira
リリース
（2002）


Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
（1995）


Subversionリリース
（2000）


CVS 誕生
（1990）

1990～2000：🌎第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

flickr
リリース
（2004/2）

aws
AWS
サービス開始
（2006）

Google Cloud
サービス開始
（2008）

Azure
サービス開始
（2010）

Selenium
（2004）

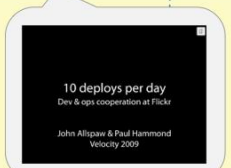
REDMINE
flexible project management
（2006）

trac
（2004）

git
（2005）

GitHub
リリース
（2008）

AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）



GitLab
（2011）


Jenkins 誕生
（2011）

2009～2020：🌎第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

GitHub Actions
（2018）

DATADOG
（2018）

Findy Team+
Team+
（2021/10）

Claude 1
（2023/3/14）

Cline
（2024/7）

GitHub Copilot
（2021/6/29）

Gemini 1
（2023/12/6）

ChatGPT
一般公開
（2022/11/30）

Devin
（2024/3/12）

Agile Alliance + Project Management Institute

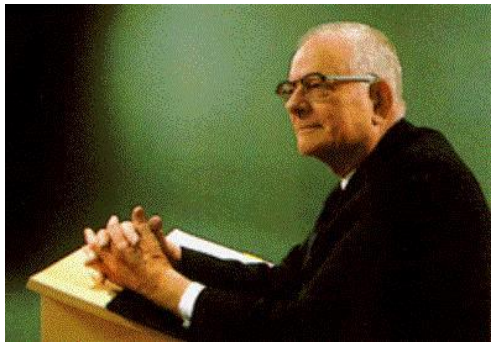
2025/1/3：🇺🇸Agile AllianceがPMBOKなどを策定するPMIに加盟

ソフトウェア開発現代史年表 Ver2.09

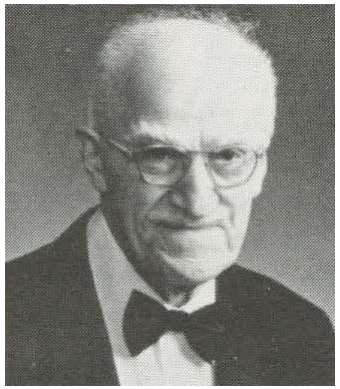
戦後、日本のものづくりは
「安かろう悪かろう」

ソフトウェア開発現代史 -Beginning-

- 戦後、日本のものづくりは「安かろう悪かろう」
- これを打破するため、先人たちは2人の人物に教えを請いました
  - 1950年 統計学の専門家であったW.E.Deming博士が日本科学技術連盟(JUSE)の招きにより初来日
    - PDCA＝デミング・サイクル
    - 日本の経営者に品質管理の考え方や統計的手法を伝える
    - 「品質の統計的管理8日間コース」セミナー
    - 「経営者のための品質管理講習会1日コース」セミナー
  - 1954年アメリカの品質管理コンサルタントであるJoseph M. Juran博士が初来日
    - パレートの法則
    - ジュランのトリロジー
      - 品質計画・品質管理・品質改善
    - TQM（Total Quality Management：全社的品質管理）の理論的基盤を築く
- これにより、日本製造業は飛躍的な品質向上を遂げます



W. Edwards Deming
（ウィリアム・エドワーズ・デミング、1900-1993）



Joseph M. Juran
（ジョセフ・M・ジュラン、1904-2008）

ソフトウェア開発現代史 -Beginning-

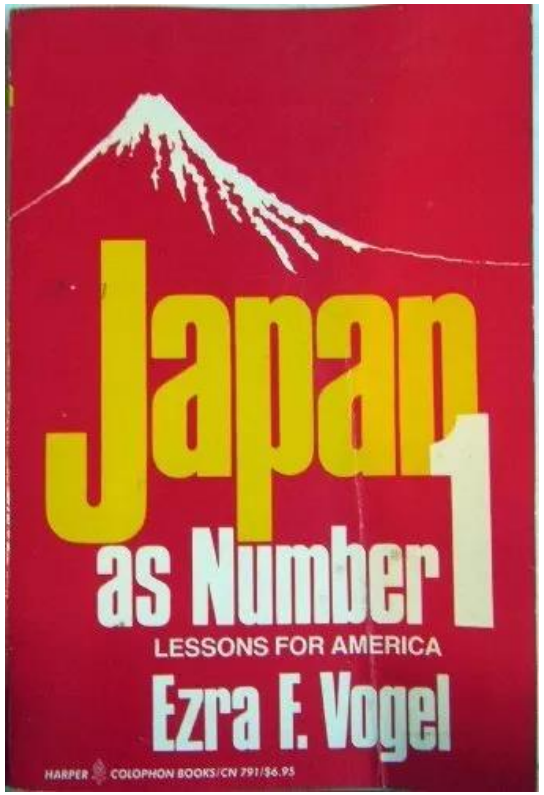
- 1960年頃、日本のものづくりは「品質」という武器を手に入れ躍進。日本製品が世界中で人気となる
- 1979年、米ハーバード大学のエズラ・ヴォーゲル教授が出版した『Japan as Number One: Lessons for America』がベストセラーに

ソニートリニトロンシリーズ
（1968～）

Toyota Corolla Liftback SR5 001
（1980～）

“ウォークマン” 1号機 TPS-L2
（1979）

世界初のポータブル
CDプレーヤー
D-50（1984）



Vogel, E. F. (1979). Japan as number one: Lessons for America. Cambridge, MA: Harvard University Press.

ソフトウェア開発現代史 -Beginning-

- 1960年頃、日本のものづくりは「品質」という武器を手に入れ躍進。日本製品が世界中で人気となる
- 1979年、米ハーバード大学のエズラ・ヴォーゲル教授が出版した『Japan as Number One: Lessons for America』がベストセラーに

ソニートリニトロンシリーズ
（1968～）

Toyota Corolla Liftback SR5 001
（1980～）

Vogel, E. F. (1979). Japan as number one: Lessons for America. Cambridge, MA: Harvard University Press.

品質管理と品質保証「品質」が日本の強い成功体験となる

“ウォークマン”1号機 TPS-L2
（1979）

世界初のポータブル
CDプレーヤー
D-50（1984）

ソフトウェア関連の論文・文献

ICTの進化

西暦

主な出来事

1970
1976
1981
1987
1989/02/01
1994/10/31
1996/8
2001/5/18
2003/9/1
2005/10
2009/11/27
2011/7/16
2014/8/18
2017/6/22
2019/9/17
2021/3/15
2023/11/21
2025/6/25
1977
1979
1984
1988/3/1
1995/10/1
2000/12/1
2003/9/1
2005/10/7
2008/12/30
2010/10/12
2013/1/10
2016/10/6
2018/4/1
2019/9/30
2021/12/1
2023/11/28
2024/12/25
1978/5/1
1986
1989
1995
1999
2002
2004/11/16
2010/7/27
2012/3/15
2013/8/1
2018/3/27
2021/8/1
2022/12/17
2024/7/11
1975
1978
1979
1982
1990/10/10
1992/3/13
1996/4/1
2002/11/8
2004/7/16
2006/10/19
2010～
2011/10/18
2014/4/22
2017/10/14
2018/11/22
2021/12/10
2024/9/13

パソコン通信
携帯電話・PHSの普及、インターネット黎明期
携帯電話の多機能化、ブロードバンド時代
スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：
IBM System 360とVT100（1970～1980）
PC/AT互換機が誕生（1981～）
PC-9800シリーズ（1982～2003）
Apple Macintosh
（1984～）
ワークステーションの時代：
Sun SPARCstation
（1989～1994）
数字送信の開始による
ポケベルブーム（日本）
（1992～1996）
テレホーダイ
（1995～2023）
初代Mac
（1998～）
F501i HYPER
iモード開始
（1999/2/22）
家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
（2001～）
初代Phone発表
（2007/1/9）
初代iPad発表
（2010/4/3）
Samsung
Galaxy S II
（2011/5/2）
Netflixが日本に上陸(2015/9/2)
※では2007/1にサービスイン
ARMアーキテクチャのSoC
Apple M1 生産（2020）
MacBook Pro M4 Max
（2024）

50年前
40年前
30年前
20年前
10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

1974年：「TCP/IP」の最初の仕様がRFC 675として公開
1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。
★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用
1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での拠出が発生
1997～2010年代：米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。
2010年～：米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化。

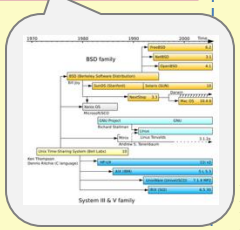
1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる
1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUPなど）
2000年代～：米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。
2000年代～：日本は大企業を中心に、ウォータフォールモデルを採用し続ける。
2020/3～2023/5：COVID-19

1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……
1988～：日本を含む諸外国へ「通商法スーパー301条」発動
2000/3～：ITバブル崩壊
2010年以降～：日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979～：日本の製造業の高品質、ものづくりの強さを研究。
1991～：バブル崩壊「失われた○○年」
2001/2/11：アジャイルソフトウェア開発宣言
2000年代後半～：クラウドファースト・クラウドネイティブ時代に突入
2021～：AI（GenAI）が前提の時代へ
2008/9～：リーマンショック
2018：マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半～1990年代前半：UNIX戦争
1985～1990：国家プロジェクト「Σ計画」が頓挫
2009：Flickrのエンジニアである John AllspawとPaul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

© The Deming Institute



Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
(1996)


Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
(1991/9/17)

Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
（1995）


CVS 誕生
(1990)


Bugzilla
リリース
(2000)

JIRA
Jira
リリース
(2002)

Subversionリリース
(2000)

flickr
リリース
(2004/2)

aws
AWS
サービス開始
(2006)

Google Cloud
サービス開始
(2008)

Azure
サービス開始
(2010)

Selenium
(2004)

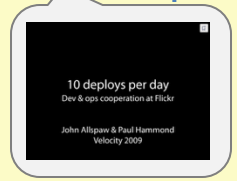
REDMINE
flexible project management
(2006)

trac
(2004)

git
(2005)

AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）

GitHub
リリース
(2008)



GitLab
(2011)


Jenkins 誕生
(2011)

GitHub Actions
(2018)

DATADOG
(2018)

Findy Team+
Team+
(2021/10)

Claude 1
(2023/3/14)

Cline
(2024/7)

GitHub Copilot
(2021/6/29)

Gemini 1
(2023/12/6)

ChatGPT
一般公開
(2022/11/30)

Devin
(2024/3/12)

Agile Alliance + Project Management Institute.

2009～2020：第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

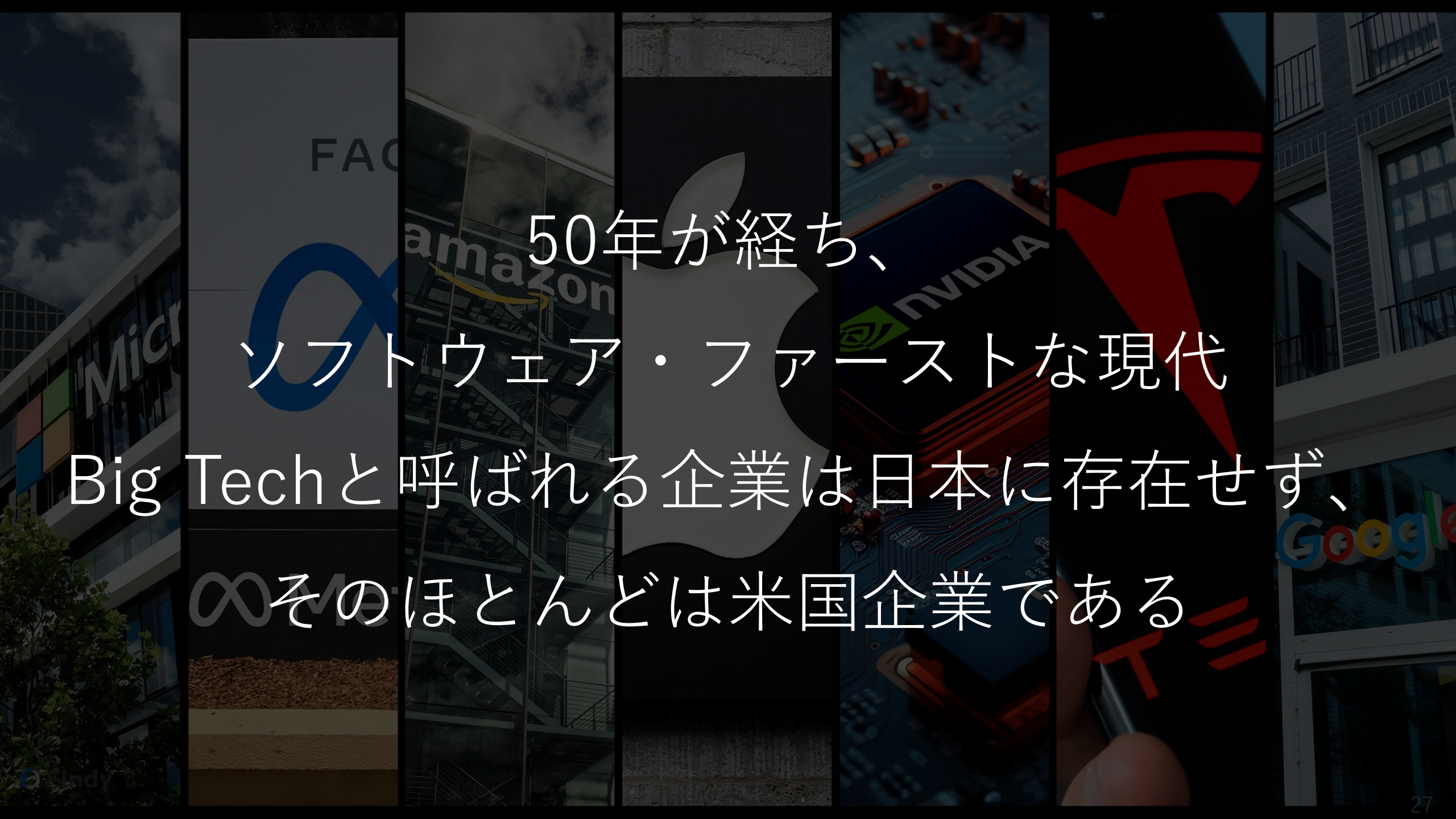
2025/1/3：Agile AllianceがPMBOKなどを策定するPMIに加盟

1990～2000：第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

ソフトウェア開発現代史年表 Ver2.09


戦後日本は愚直に学び、
品質を鍛え抜いて世界を驚かせた。
そして50年が経った・・・
ソフトウェア関連の論文・文献
ICTの進化
西暦
主な出来事
パソコン通信
携帯電話・PHSの普及、インターネット黎明期
携帯電話の多機能化、ブロードバンド時代
スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）
50年前
40年前
30年前
20年前
10年前
1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026
Point: 米国防総省(DoD)が与えた影響
1974年：「TCP/IP」の最初の仕様がRFC 675として公開
1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。
★ピアソンショック（2013年8月1日）
1980年～：米国防総省(DoD)がウォーターフォールを採用
1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生
1997～2010年代：米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。
2010年～：米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化
1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる
1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUPなど）
2000年代～：米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。
2000年代～：日本は大企業を中心に、ウォータフォールモデルを採用し続ける。
2020/3～2023/5：COVID-19
1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……
1988～：日本を含む諸外国へ「通商法スーパー301条」発動
2000/3～：ITバブル崩壊
2010年以降～：日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用
1979～：日本の製造業の高品質、ものづくりの強さを研究。
1991～：バブル崩壊「失われた○○年」
2001/2/11：アジャイルソフトウェア開発宣言
2000年代後半～：クラウドファースト・クラウドネイティブ時代に突入
2021～：AI（GenAI）が前提の時代へ
1980年代後半～1990年代前半：日米半導体摩擦
2008/9～：リーマンショック
2018：マイクロソフト、GitHub を 75 億ドルで買収
1985～1990：通産省の官民共同プロジェクト「Σ計画」が挫折
2009：Flickrのエンジニアである John Allspaw と Paul Hammond が初めてDevOpsに関する伝説的な講演を
© The Deming Institute
“トリニトロン”
日本製品が欧米で人気
"ウォークマン"
1号機 TPS-L2
(1979)
Toyota Corolla
Liftback SR5 001
(1980～)
世界初のポータブル
CDプレーヤー
D-50（1984）
Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
(1996)
Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
(1991/9/17)
Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
(1995)
Bugzilla
リリース
(2000)
Jira
リリース
(2002)
Subversionリリース
(2000)
CVS 誕生
(1990)
flickr
リリース
(2002)
AWS
サービス開始
(2006)
Google Cloud
サービス開始
(2008)
Azure
サービス開始
(2010)
Selenium
(2004)
REDMINE
(2006)
trac
(2004)
git
(2005)
AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）
GitHub
リリース
(2008)
GitLab
(2011)
Jenkins 誕生
(2011)
GitHub Actions
(2018)
DATADOG
(2018)
Findy Team+
Team+
(2021/10)
Claude 1
(2023/3/14)
Cline
(2024/7)
GitHub Copilot
(2021/6/29)
Gemini 1
(2023/12/6)
ChatGPT
一般公開
(2022/11/30)
Devin
(2024/3/12)
1990～2000：第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅
2009～2020：第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退
2025/1/3：Agile AllianceがPMBOKなどを策定するPMIに加盟
ソフトウェア開発現代史年表 Ver2.09
© 2025 ファインディ株式会社. All rights reserved.（写真および他社ロゴを除く）
本資料はファインディ株式会社が独自に編集・作成したものです（一部、パブリックドメイン素材を含みます）。
Findy® および Team+® はファインディ株式会社の商標です。
使用されている写真や他社ロゴは、各権利者に帰属し、説明目的でのみ使用しています。その他の商品名や口ゴは、各社の商標または登録商標です。


amazon
NVIDIA
Google


50年が経ち、
ソフトウェア・ファーストな現代
Big Techと呼ばれる企業は日本に存在せず、
そのほとんどは米国企業である

Findy

ターニングポイントはどこ？

1980年代後半

ソフトウェアプロジェクトは失敗ばかりだった

ソフトウェア関連の論文・文献

1970 1976 1981 1987 1989/02/01 1994/10/31 1996/8 2001/5/18 2003/9/1 2005/10 2009/11/27 2011/7/16 2014/8/18 2017/6/22 2019/9/17 2021/3/15 2023/11/21 2025/6/25

1977 1979 1984 1988/3/1 1995/10/1 2000/12/1 2003/9/1 2005/10/7 2008/12/30 2010/10/12 2013/1/10 2016/10/6 2018/4/1 2019/9/30 2021/12/1 2023/11/28 2024/12/25

1978/5/1 1986 1989 1995 1999 2002 2004/11/16 2010/7/27 2012/3/15 2013/8/1 2018/3/27 2021/8/1 2022/12/17 2024/7/11

1975 1978 1979 1982 1990/10/10 1992/3/13 1996/4/1 2002/11/8 2004/7/16 2006/10/19 2010～ 2011/10/18 2014/4/22 2017/10/14 2018/11/22 2021/12/10 2024/9/13

ICTの進化

パソコン通信 | 携帯電話・PHSの普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：IBM System 360とVT100（1970～1980）
PC/AT互換機が誕生（1981～）
PC-9800シリーズ（1982～2003）
Apple Macintosh（1984～）
ワークステーションの時代：Sun SPARCstation（1989～1994）
数字送信の開始によるポケベルブーム（日本）（1992～1996）
テレホーダイ（1995～2023）
初代Mac（1998～）
F501i HYPER iモード開始（1999/2/22）
家庭向けADSL・FTTHの普及、ブロードバンド時代に突入（2001～）
初代Phone発表（2007/1/9）
初代iPad発表（2010/4/3）
Samsung Galaxy S II（2011/5/2）
Netflixが日本に上陸(2015/9/2)※米では2007/1にサービスイン
ARMアーキテクチャのSoC Apple M1 生産（2020）
MacBook Pro M4 Max（2024）

西暦

50年前 40年前 30年前 20年前 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

主な出来事

1974年：米「TCP/IP」の最初の仕様がRFC 675として公開
1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。
★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響
1980年～：米国防総省(DoD)がウォーターフォールを採用
1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生
1997～2010年代：米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。
2010年～：米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化

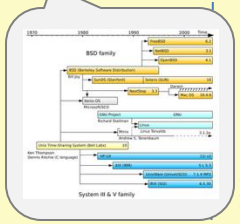
1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる
1990年代：米ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUPなど）
2000年代～：米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。
2000年代～：日本は大企業を中心に、ウォータフォールモデルを採用し続ける。
2020/3～2023/5：COVID-19

1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……
1988～：米日本を含む諸外国へ「通商法スーパー301条」発動
2000/3～：ITバブル崩壊
2010年以降～：日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979～：米日本の製造業の高品質、ものづくりの強さを研究。
1991～：日バブル崩壊「失われた○○年」
2001/2/11：アジャイルソフトウェア開発宣言
2000年代後半～：クラウドファースト・クラウドネイティブ時代に突入
2021～：AI（GenAI）が前提の時代へ
2008/9～：米リーマンショック
2018：米マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半～1990年代前半：UNIX戦争
1985～1990：日国家プロジェクト「Σ計画」が頓挫
2009：米Flickrのエンジニアである John Allspaw と Paul Hammond が初めてDevOpsに繋がる伝説的な講演を行う

© The Deming Institute



Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
(1996)

Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
(1991/9/17)

Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
(1995)

CVS 誕生
(1990)

Bugzilla
リリース
(2000)

Jira
リリース
(2002)

Subversionリリース
(2000)

flickr
リリース
(2004/2)

AWS
サービス開始
(2006)

Google Cloud
サービス開始
(2008)

Azure
サービス開始
(2010)

Selenium
(2004)

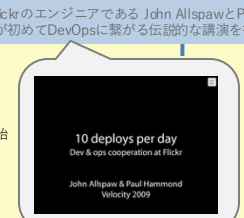
REDMINE
(2006)

trac
(2004)

git
(2005)

GitHub
リリース
(2008)

AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）



GitLab
(2011)

Jenkins 誕生
(2011)

GitHub Actions
(2018)

DATADOG
(2018)

Findy Team+
Team+
(2021/10)

Claude 1
(2023/3/14)

Cline
(2024/7)

GitHub Copilot
(2021/6/29)

Gemini 1
(2023/12/6)

ChatGPT
一般公開
(2022/11/30)

Devin
(2024/3/12)

Agile Alliance + Project Management Institute.

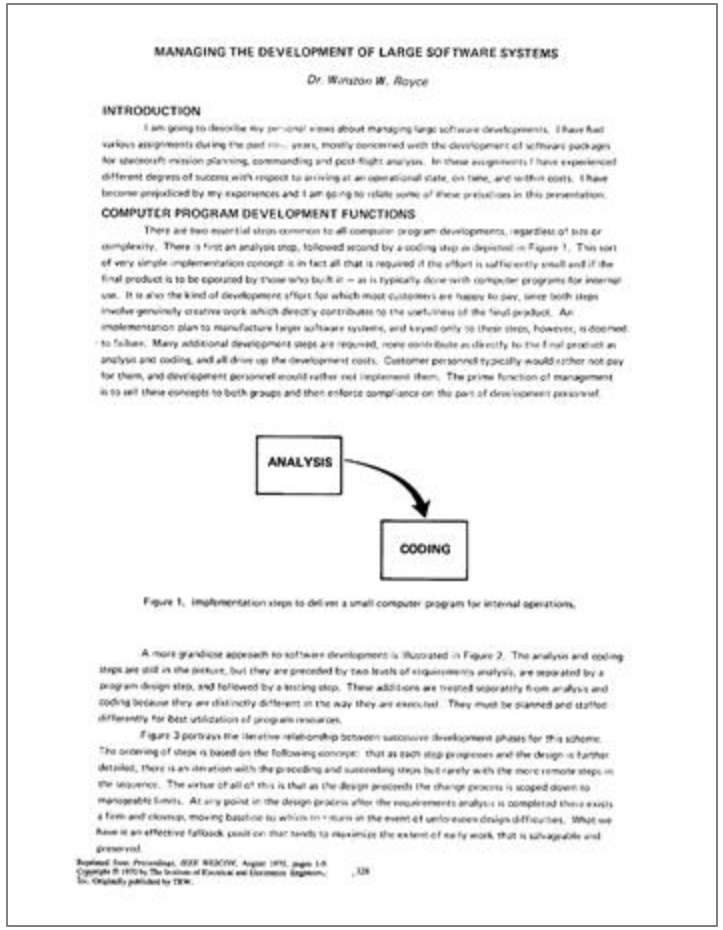
2009～2020：第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

1990～2000：第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

2025/1/3：Agile AllianceがPMBOKなどを策定するPMIに加盟

ウォーターフォール？

- Winston W. Royce による“Managing the Development of Large Software Systems(1970)”



MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS

Dr. Winston W. Royce

INTRODUCTION

COMPUTER PROGRAM DEVELOPMENT FUNCTIONS

ANALYSIS

CODING

Figure 1. Implementation steps to deliver a small computer program for internal operations.

https://dl.acm.org/doi/10.5555/41765.41801

Winston W. Royce
（ウィンストン・W・ロイス）

ウォーターフォール？

- Winston W. Royce による“Managing the Development of Large Software Systems(1970)”

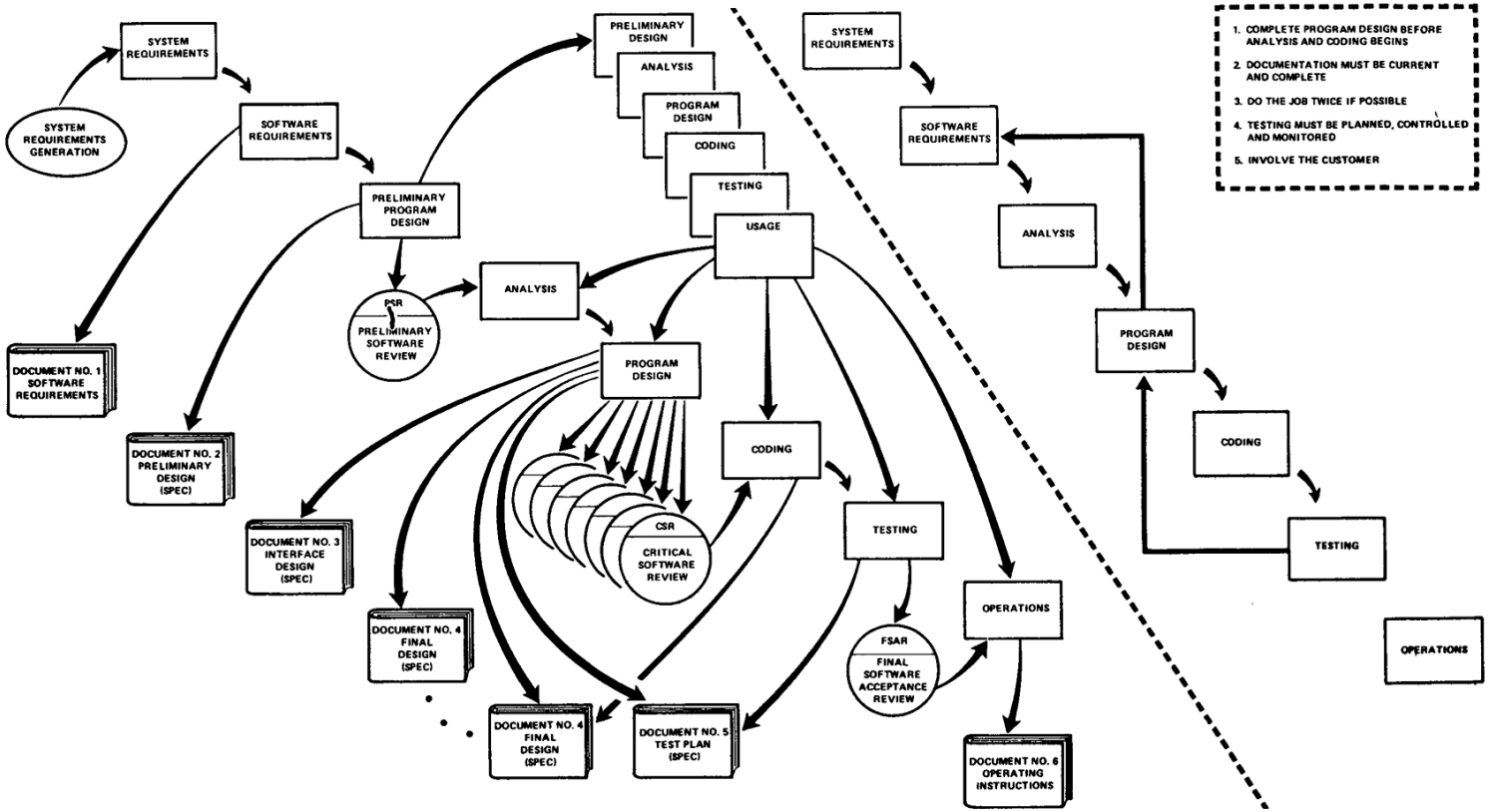


Figure 10. Summary

https://dl.acm.org/doi/10.5555/41765.41801

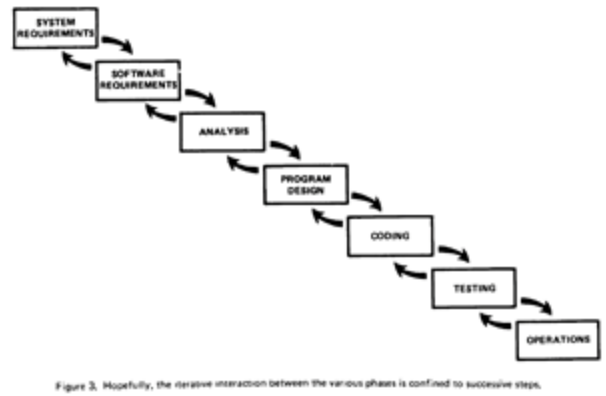


Figure 3. Hopefully, the iterative interaction between the various phases is confined to successive steps.

図3 顧客に引き渡すための大規模プログラムを開発するための実現ステップ

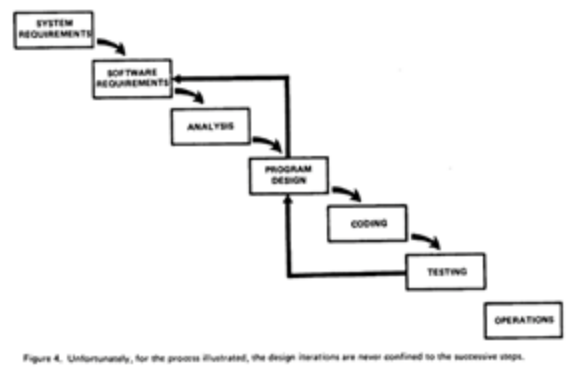


Figure 4. Unfortunately, for the process illustrated, the design iterations are never confined to the successive steps.

図4 反復（後戻り）は隣り合うステップに限定されない

「ウォーターフォール」と名付けた人物と広めた人物

SOFTWARE REQUIREMENTS: ARE THEY REALLY A PROBLEM?

T. E. Bell and T. A. Thayer

TRW Defense and Space Systems Group
Redondo Beach, California

SOFTWARE REQUIREMENTS: ARE THEY REALLY A PROBLEM?

T. E. Bell and T. A. Thayer

TRW Defense and Space Systems Group
Redondo Beach, California

Keywords and Phrases

Ballistic Missile Defense
Requirements
Requirements Problems
Software Engineering
Software Requirements
Software Requirements Engineering
Software Requirements Problems
SREM
SREP

Abstract

Do requirements arise naturally from an obvious need, or do they come about only through diligent effort -- and even then contain problems? Data on two very different types of software requirements were analyzed to determine what kinds of problems occur and whether these problems are important. The results are dramatic: software requirements are important, and their problems are surprisingly similar across projects. New software engineering techniques are clearly needed to improve both the development and statement of requirements.

I. Introduction

Identifying the cause of a problem in a software system is often very easy -- if the cause is a problem in code. Typically, identified coding problems result in clearly incorrect answers or in abnormal terminations of the software. Similarly, problems in a software system caused by deficiencies in design are often easy to identify from unexpected software operation or from extreme difficulty in maintaining and modifying the system. Problems in the system caused by deficiencies in software requirements, on the other hand, are often not identified at all, or are thought to be caused by bad design or limitations of computing technology. If there are problems in developing requirements, however, the software system meeting those requirements will clearly not be effective in solving the basic need, even if the causes of the problems are incorrectly identified.

The Ballistic Missile Defense Advanced Technology Center (BMDATC) is sponsoring an integrated software development research program [1] to improve the techniques for developing correct, reliable BMD software. Reflecting the critical importance of requirements in the development process, the Software Requirements Engineering Program (SREP) has been undertaken as a part of this integrated program by TRW Defense and Space Systems Group* to examine and improve the quality of requirements.

One of the first efforts in SREP has been to characterize the problems with requirements so that techniques can be developed to improve the situation. Instead of pursuing philosophical discussions about what the problems might be, we have undertaken empirical studies to determine what the problems actually are. A limitation on the number of Ballistic Missile Defense (BMD) systems currently being developed (there is only one) has led us to examine both BMD and more common data processing systems to ensure that our results are characteristic of software requirements in general, rather than just one particular project.

This paper reports on initial results that have set much of the direction pursued in the Software Requirements Engineering Methodology [2], the Requirements Statement Language [3], and the Requirements Engineering and Validation System [4]. The initial efforts were oriented to determine the magnitude and characteristics of the problems, and to indicate what types of techniques could correct the problems. The empirical study of software problems is continuing in parallel with technology development so that the technology can be refined and tested for effectiveness in solving the identified problems.

II. What are Software Requirements?

One school of thought maintains that software requirements arise naturally, and that they are correct by definition. If these requirements merely state a basic need (e.g., "do payroll"), then that's all that is needed. On the other hand, if the requirements state each subroutine's detailed characteristics, then those are the required characteristics, and the implementer should not question them.

Adherents to this school of thought have grown fewer and fewer as the software industry has gathered experience with this approach to developing software. When every requirement ranging in detail from needs statements to subroutine specifications is considered in the same way, the resulting systems tend to be seriously deficient. If coding personnel are assigned the task of implementing a system with only a needs statement, the critical phase of software design will likely be skipped -- with disasterous results. On the other end of the scale, if detailed subroutine specifications are accepted without ever having been

*Under Contract DASG60-75-C-0022

Bell, T. E., & Thayer, T. A. (1976). Software requirements: Are they really a problem? Proceedings of the 2nd International Conference on Software Engineering (ICSE '76), 61–68. https://dl.acm.org/doi/10.5555/800253.807650

「ウォーターフォール」と名付けた人物と広めた人物

- T.E. BellとT.A. Thayerの1976年論文「Software Requirements: Are They Really a Problem?」では、Winston W. Royceの1970年論文を参照し、「開発活動のウォーターフォール（滝）」という表現を使用してソフトウェア開発プロセスを説明しました。
- BellとThayerは、Royceの1970年の論文を正確には理解していなかったようです（読んだの？というレベル）。Royceの論文にはBellとThayerのFigure 1に似た図が含まれていましたが、彼はそのモデルについて「上記の実装は失敗を招く」と記述しています。
- しかし、BellとThayerは要件の不備が設計や実装段階での失敗につながると述べ、「トップダウン型で進む厳密なプロセス」の必要性を主張しました。この主張が間接的にウォーターフォールの誤解を促進したのです。
- さらに、Barry Boehm氏が1981年の著書「Software Engineering Economics」でRoyce氏のモデルをウォーターフォール型として紹介したことで、この誤解が業界全体に定着しました。
- そして、彼らの図（Figure 1）は、USA DoD（米国国防総省）に受け入れられ、DOD-STD-2167A (1988) に組み込まれ、ミッションクリティカルなソフトウェア開発にデフォルトとして要求されるようになります。

Bell, T. E., & Thayer, T. A. (1976). Software requirements: Are they really a problem? Proceedings of the 2nd International Conference on Software Engineering (ICSE '76), 61–68. https://dl.acm.org/doi/10.5555/800253.807650

design. Each of these is at a lower level of detail than the former, so the system developers are led through a top-down process. The same top-down approach to a series of requirements statements is explained, without the specialized military jargon, in an excellent paper by Royce [5]; he introduced the concept of the "waterfall" of development activities. In this approach software is developed in the disciplined sequence of activities shown in Figure 1.

“一連の要求仕様書に対する同じトップダウンアプローチが、専門的な軍事用語を使わずに、Royce [5] の優れた論文で説明されている。彼は開発活動の「ウォーターフォール」という概念を導入した。このアプローチでは、ソフトウェアは図1に示される規律正しい活動の順序で開発される。”

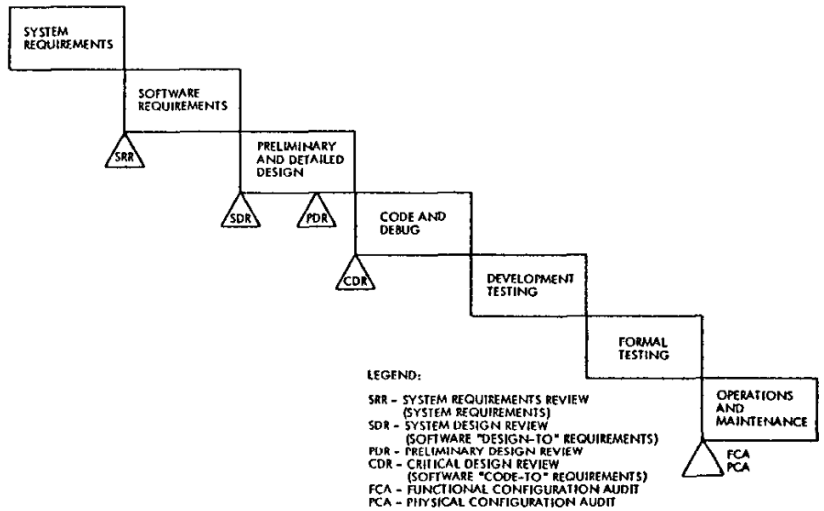


Figure 1. Development Phases of the System Development Cycle

ソフトウェア関連の論文・文献

1970

1976

1981

1987

1989/02/01

1994/10/31

1996/8

2000

1977

1979

1984

1988/3/1

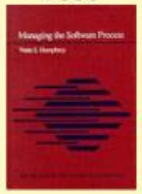
1995/10/1

2000/12/1

1978/5/1

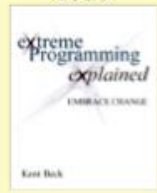
1986

1989

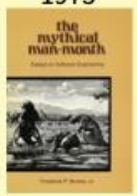


1995

1999



1975



1978

1979

1982

1990/10/10

1992/3/13

1996/4/1

ICTの進化

パソコン通信

携帯電話・PHSの普及、インターネッ

メインフレーム時代：
IBM System 360とVT100（1970～1980）

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始による
ポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代iMac
（1998～）

F501i HYPER
iモード開始
（1999/2/22）

西暦

50年前　40年前　30年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000

西暦

50年前 40年前 30年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000

主な出来事

1974年：「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4

Point: 米国防総省(DoD)が与えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：米国条件として要求し始める。

1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

200
200

1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

2

1979～：日本の製造業の高品質、ものづくりの強さを研究。

1991～：バブル崩壊「失われた○○年」

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

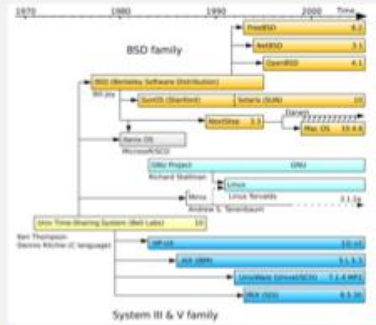
© The Deming Institute


“トリニトロン”
日本製品が欧米で人気


"ウォークマン"
1号機 TPS-L2
（1979）


Toyota Corolla
Liftback SR5 001
（1980～）


世界初のポータブル
CDプレーヤー
D-50（1984）




Java v1.0 正式リリース
Javaはオブジェクト指向と
仮想マシン技術を普及させ、
後の言語設計にも影響を与えた
（1996）


Linux 0.01 リリース
UNIXが商業化・断片化していく中、
Linuxはオープンソースの力で
統一的な開発基盤となり、
世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSに
TCP/IPが標準搭載され
ワークステーション並みに
（1995）


Bugzi
リリー
（200


Subversion
（200


CVS 誕生
（1990）

1990～2000：第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

ソフトウェア開発現代史年表 Ver2.09

西暦

50年前 40年前 30年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000

主な出来事

1974年：「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4

Point: 米国防総省(DoD)が与えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：米国
条件として要求し始める。

1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

200
200

1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

1979～：日本の製造業の高品質、ものづくりの強さを研究。

1991～：バブル崩壊「失われた○○年」

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

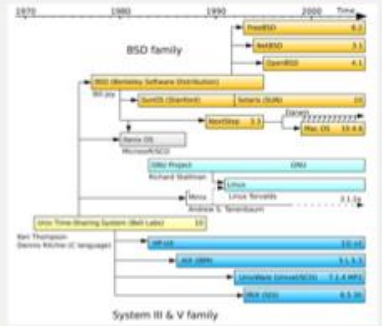
© The Deming Institute


“トリニトロン”
日本製品が欧米で人気


"ウォークマン"
1号機 TPS-L2
（1979）


Toyota Corolla
Liftback SR5 001
（1980～）


世界初のポータブル
CDプレーヤー
D-50（1984）




Java v1.0 正式リリース
Javaはオブジェクト指向と
仮想マシン技術を普及させ、
後の言語設計にも影響を与えた
（1996）


Bugzi
リリー
（200


Linux 0.01 リリース
UNIXが商業化・断片化していく中、
Linuxはオープンソースの力で
統一的な開発基盤となり、
世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSに
TCP/IPが標準搭載され
ワークステーション並みに
（1995）


Subversion
（200


CVS 誕生
（1990）

1990～2000：第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

ソフトウェア開発現代史年表 Ver2.09

西暦

50年前 40年前 30年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000

主な出来事

1974年：「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4

Point: 米国防総省(DoD)が与えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：米国
条件として要求し始める。

1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

200
200

1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

1979～：日本の製造業の高品質、ものづくりの強さを研究。

1991～：バブル崩壊「失われた○○年」

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

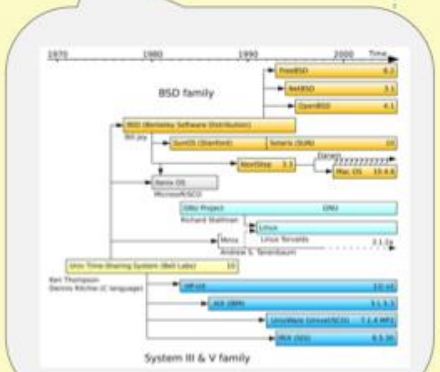
© The Deming Institute


“トリニトロン”
日本製品が欧米で人気


"ウォークマン”
1号機 TPS-L2
（1979）


Toyota Corolla
Liftback SR5 001
（1980～）


世界初のポータブル
CDプレーヤー
D-50（1984）




Java v1.0 正式リリース
Javaはオブジェクト指向と
仮想マシン技術を普及させ、
後の言語設計にも影響を与えた
（1996）


Bugzi
リリー
（200


Linux 0.01 リリース
UNIXが商業化・断片化していく中、
Linuxはオープンソースの力で
統一的な開発基盤となり、
世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSに
TCP/IPが標準搭載され
ワークステーション並みに
（1995）


Subversion
（200


CVS 誕生
（1990）

1990～2000：第一次ブラウザ戦争。Internet Explorerの躍進、
Netscape Navigatorの消滅

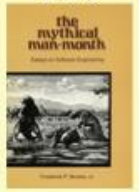
ソフトウェア開発現代史年表 Ver2.09

・文献

ICTの進化

西暦

主な出

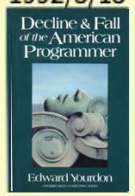
1975


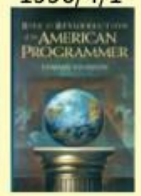
1978


1979


1982


1990/10/10


1992/3/13


1996/4/1


パソコン通信

携帯電話・PHSの普及、インターネッ



メインフレーム時代：
IBM System 360とVT100（1970～1980）


PC/AT互換機が誕生（1981～）


PC-9800シリーズ（1982～2003）


Apple Macintosh
（1984～）


ワークステーションの時代：
Sun SPARCstation
（1989～1994）


数字送信の開始による
ポケベルブーム（日本）
（1992～1996）


テレホーダイ
（1995～2023）


初代iMac
（1998～）


F501i HYPER
iモード開始
（1999/2/22）

50年前　40年前　30年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000

1974年：「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4

Point: 米国防総省(DoD)が与えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：米国
条件として要求し始める。

1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

200
200

1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

2

1979～：日本の製造業の高品質、ものづくりの強さを研究。

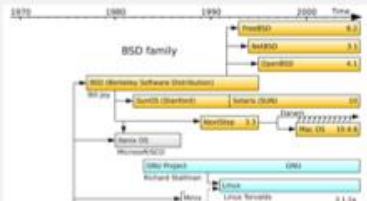
1991～：バブル崩壊「失われた○○年」

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

© The Deming Institute

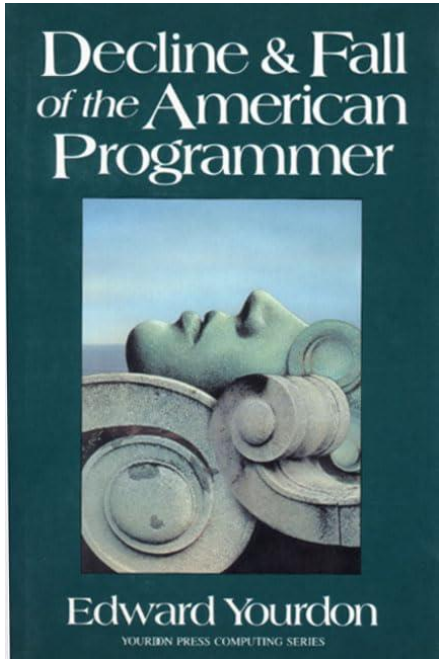



35年前に痛い目にあっている

- 1970〜1980年代：🌎多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる
- 1980年〜：US米国防総省(DoD)がウォーターフォールを採用
- 1980年後半〜1990年前半：US米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1992/3/13



Edward Nash Yourdon

"Decline & Fall of the American Programmer"
「アメリカン・プログラマー 没落の危機」

これはエドワード・ギボンの古典的名著「ローマ帝国衰亡史」（The History of the Decline and Fall of the Roman Empire, 1776-1788）にかけたタイトルと思われる。

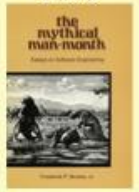
※ただ自慢したい写真

・文献

ICTの進化

西暦

主な出

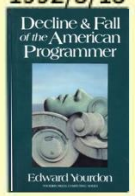
1975


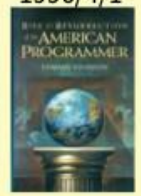
1978


1979


1982


1990/10/10


1992/3/13


1996/4/1


パソコン通信

携帯電話・PHSの普及、インターネッ



メインフレーム時代：
IBM System 360とVT100（1970～1980）


PC/AT互換機が誕生（1981～）


PC-9800シリーズ（1982～2003）


Apple Macintosh
（1984～）


ワークステーションの時代：
Sun SPARCstation
（1989～1994）


数字送信の開始による
ポケベルブーム（日本）
（1992～1996）


テレホーダイ
（1995～2023）


初代iMac
（1998～）


F501i HYPER
iモード開始
（1999/2/22）

50年前　40年前　30年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000

1974年：「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4

Point: 米国防総省(DoD)が与えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：米国
条件として要求し始める。

1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

200
200

1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

2

1979～：日本の製造業の高品質、ものづくりの強さを研究。

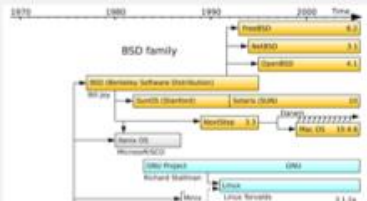
1991～：バブル崩壊「失われた○○年」

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

© The Deming Institute







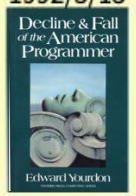
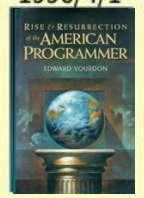
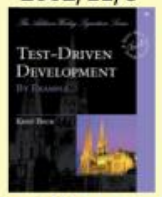
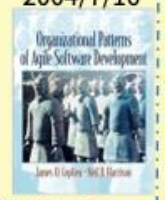


1978

1979

1982

1990/10/10

1992/3/13

1996/4/1

2002/11/8

2004/7/16

2006/10/19

パソコン通信
携帯電話・PHSの普及、インターネット黎明期
携帯電話の多機能化、ブロー

980）

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始による
ポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代iMac
（1998～）

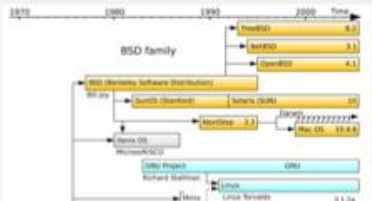
F501i HYPER
iモード開始
（1999/2/22）


家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
（2001～）

初代iPhon
（2007/
40年前
30年前
20年前
1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006
」の最初の仕様がRFC 675として公開
1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IP
えた影響
1980年～：米国防総省(DoD)がウォーターフォールを採用
1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生
1997～2010年代：米国防総省(DoD)が CMMレベル3以上を防衛関連のソフ条件として要求し始める。
関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
「ウォーターフォール」が広まる
1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）
2000年代～：米国を中心にでアジャイル手法が急速
2000年代～：日本は大企業を中心に、ウォータフォ
家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……
1988～：日本を含む諸外国へ「通商法スーパー301条」発動
2000/3～：ITバブル崩壊
1979～：日本の製造業の高品質、ものづくりの強さを研究。
1991～：バブル崩壊「失われた○○年」
2001/2/11：アジャイルソフトウェア開発宣言
2000年代
1980年代後半～1990年代前半：UNIX戦争
1985～1990：国家プロジェクト「Σ計画」が頓挫

ng Institute




マンガ ソフトウェア革命

BSD family

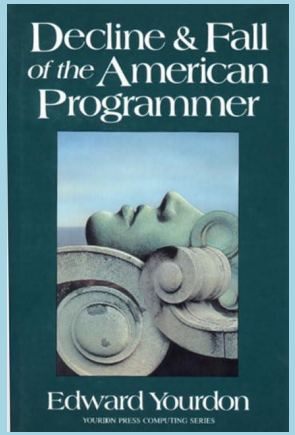

flickr
リリース
（2004/2）
aws
AWS

ところが、4年ほどで復活宣言

- 1970～1980年代：🌎多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる
- 1980年～：US米国防総省(DoD)がウォーターフォールを採用
- 1980年後半～1990年前半：US米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生
- 1990年代：USウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

ソフトウェアで復活を遂げたアメリカ

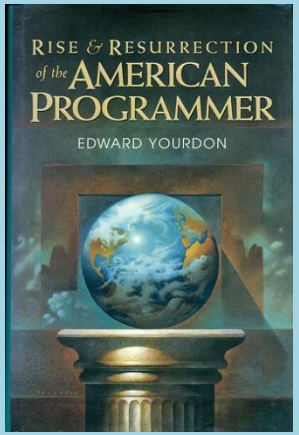
1992/3/13



アメリカン・プログラマー 没落の危機

僅か4年後

1996/4/1

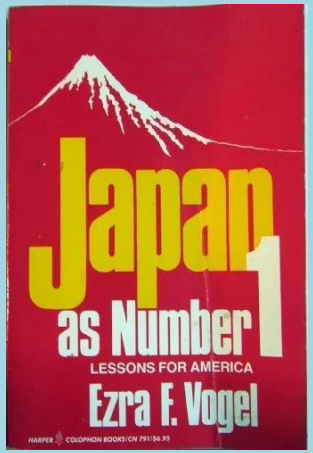


アメリカ人プログラマーの台頭と復活

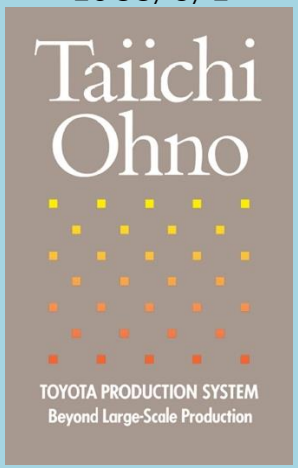
復活するアメリカ

研究しつくされる日本の製造業

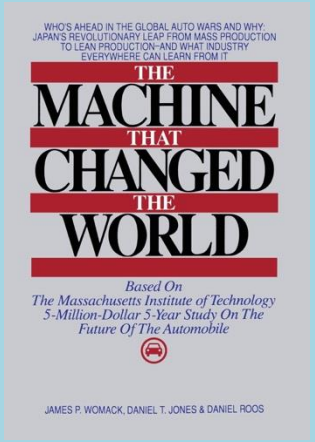
1979



1988/3/1

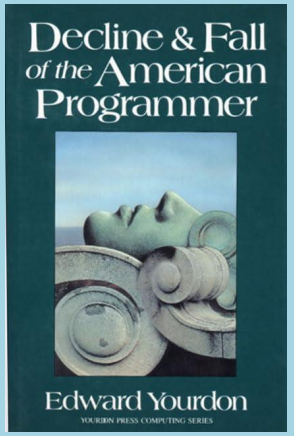


1990/10/10



ソフトウェアで復活を遂げたアメリカ

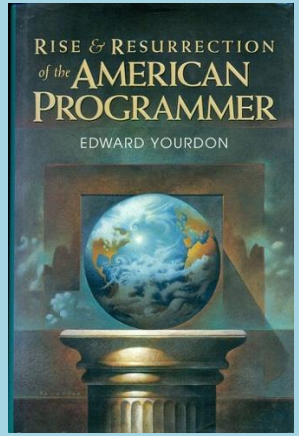
1992/3/13



アメリカン・プログラマー 没落の危機

僅か4年後

1996/4/1



アメリカ人プログラマーの台頭と復活

米国は、日本製造業の良いところを、ソフトウェア開発にも応用した

ソフトウェア関連の論文・文献

ICTの進化

西暦

主な出来事

1970 1976 1981 1977 1979 1984 1987 1989/02/01 1988/3/1 1994/10/31 1996/8 1995/10/1 1986 1995 1999 2001/5/18 2003/9/1 2005/10 2000/12/1 2003/9/1 2005/10/7 2002 2004/11/16 2009/11/27 2011/7/16 2014/8/18 2008/12/30 2010/10/12 2013/1/10 2010/7/27 2012/3/15 2013/8/1 2016/10/6 2018/4/1 2019/9/30 2021/12/1 2023/11/28 2024/12/25 2018/3/27 2021/8/1 2022/12/17 2024/7/11 2017/6/22 2019/9/17 2021/3/15 2023/11/21 2025/6/25 1975 1978 1979 1982 1990/10/4 1992/3/13 1996/4/1 2002/11/8 2004/7/16 2006/10/19 2010～ 2011/10/18 2014/4/22 2017/10/14 2018/11/22 2021/12/10 2024/9/13

パソコン通信 | 携帯電話・PHSの普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：IBM System 360とVT100（1970～1980）

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh（1984～）

ワークステーションの時代：Sun SPARCstation（1989～1994）

数字送信の開始によるポケベルブーム（日本）（1992～1996）

テレホーダイ（1995～2023）

初代iMac（1998～）

F501i HYPER iモード開始（1999/2/22）

家庭向けADSL・FTTHの普及、ブロードバンド時代に突入（2001～）

初代iPhone発表（2007/1/9）

初代iPad発表（2010/4/3）

Samsung Galaxy S II（2011/5/2）

Netflixが日本に上陸(2015/9/2)
※🇺🇸では2007/1にサービスイン

ARMアーキテクチャのSoC Apple M1 生産（2020）

MacBook Pro M4 Max（2024）

50年前 40年前 30年前 20年前 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

1974年：🇺🇸「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：🇺🇸米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年～：🇺🇸米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：🇺🇸米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年～：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化

1970～1980年代：🌎多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：🇺🇸ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代～：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代～：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3～2023/5：🌎COVID-19

1970年代後半～1980年代末：🇯🇵日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：🇺🇸日本を含む諸外国へ「通商法スーパー301条」発動

2000/3～：🇺🇸ITバブル崩壊

2010年以降～：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979～：🇺🇸日本の製造業の高品質、ものづくりの強さを研究。

1991～：🇯🇵バブル崩壊「失われた○○年」

2001/2/11：🌎アジャイルソフトウェア開発宣言

2000年代後半～：🌎クラウドファースト・クラウドネイティブ時代に突入

2021～：🌎AI（GenAI）が前提の時代へ

2008/9～：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半～1990年代前半：🌎UNIX戦争

1985～1990：🇯🇵国家プロジェクト「Σ計画」が頓挫

2009：🇺🇸Flickrのエンジニアである John Allspaw と Paul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

© The Deming Institute


“トリニトロン”
日本製品が欧米で人気


"ウォークマン"
1号機 TPS-L2
(1979)


Toyota Corolla
Liftback SR5 001
(1980～)


世界初のポータブル
CDプレーヤー
D-50（1984）

Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
(1996)


Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
(1991/9/17)


Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
(1995)


Bugzilla
リリース
(2000)


Jira
リリース
(2002)


Subversionリリース
(2000)


CVS 誕生
(1990)

flickr
リリース
(2004/2)

aws
AWS
サービス開始
(2006)

Google Cloud
サービス開始
(2008)

Azure
サービス開始
(2010)

Selenium
(2004)

REDMINE
flexible project management
(2006)

trac
(2004)

git
(2005)

AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）

GitHub
リリース
(2008)

GitLab
(2011)


Jenkins 誕生
(2011)

GitHub Actions
(2018)

DATADOG
(2018)

Findy Team+
Team+
(2021/10)

Claude 1
(2023/3/14)

Cline
(2024/7)

GitHub Copilot
(2021/6/29)

Gemini 1
(2023/12/6)

ChatGPT
一般公開
(2022/11/30)

Devin
(2024/3/12)

Agile Alliance + Project Management Institute.

2020～2023：🌎第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

1990～2000：🌎第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

2025/1/3：🇺🇸Agile AllianceがPMBOKなどを策定するPMIに加盟

ソフトウェア開発現代史年表 Ver2.09

1978

1979

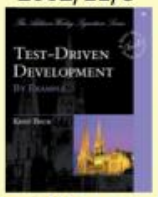
1982

1990/10/10

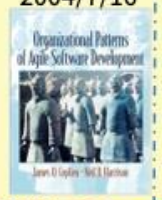
1992/3/13

1996/4/1

2002/11/8



2004/7/16



2006/10/19

パソコン通信

携帯電話・PHSの普及、インターネット黎明期

携帯電話の多機能化、ブロー

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始による
ポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代iMac
（1998～）

F501i HYPER
iモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
（2001～）

初代iPhon
（2007/

40年前

30年前

20年前

1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006

」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IP

えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：米国防総省(DoD)が CMMレベル3以上を防衛関連のソフ
条件として要求し始める。

関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代～：米国を中心にでアジャイル手法が急速

2000年代～：日本は大企業を中心に、ウォータフォ

家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

2000/3～：ITバブル崩壊

1979～：日本の製造業の高品質、ものづくりの強さを研究。

1991～：バブル崩壊「失われた○○年」

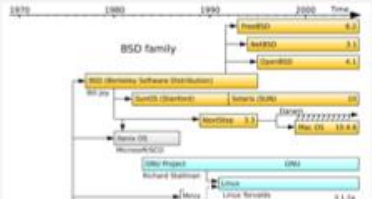
2001/2/11：アジャイルソフトウェア開発宣言

2000年代

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

ng Institute



flickr
リリース
(2004/2)

aws
AWS

ラショナル・ユニファイドプロセス（RUP）

- 1998年、ウォーターフォールの欠点を補うべく生まれたのがRational software社のRational Unified Process（RUP）
- 後にIBMに買収され統合（2003年）
- Rational software社にはUMLを生み出した“3アミーゴ”が所属していたことも有名
- RUPはソフトウェア開発のためのプロセスフレームワークであり、特に大規模で複雑なソフトウェアプロジェクトに適しています。RUPは、反復的（イテレーション）かつインクリメンタルな開発をサポートし、プロジェクトのリスクを早期に発見し、対応することを目的としています。

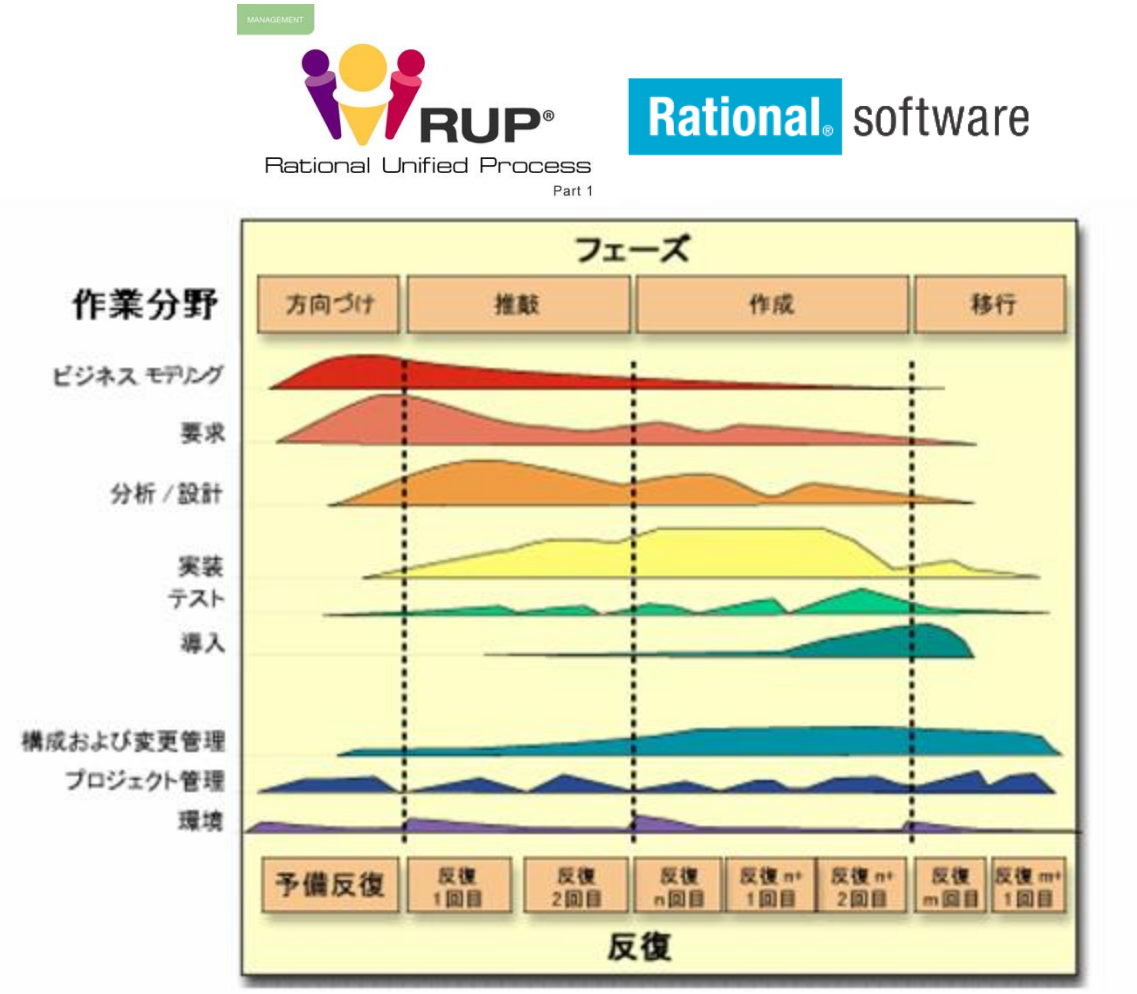


図 1　RUP の概要

ラショナル・ユニファイドプロセス（RUP）

- ウォーターフォールの提唱者 Winston Walker Royce の息子はWalker Royceは、IBMでRUPの発展の中心人物
  - 彼は（当時）IBMのRational部門のチーフソフトウェアエコノミストであり、「Software Project Management, A Unified Framework」の著者であり、IBM Rational Unified Process（RUP）における管理哲学においてボードメンバーであった。

Winston W. Royce
（ウィンストン・W・ロイス）

親子

米IBMソフトウェアグループ、
ラショナルブランドサービスバイスプレジデント（2003年当時）
Walker Royce氏（長男）
https://japan.cnet.com/article/20059853/

プロセスやツールよりも……

https://agilemanifesto.org/iso/en/manifesto.html

私たちは、実際にソフトウェアを開発し、他の人々がそれを実践するのを助けることによって、より良い開発方法を発見しています。
この作業を通じて、次のような価値を見出しました。

- プロセスやツールよりも**個人と対話**
- 包括的なドキュメントよりも**動作するソフトウェア**
- 契約交渉よりも**顧客との協力**
- 計画に従うことよりも**変化への対応**

つまり、左側の項目にも価値はありますが、私たちは右側の項目により価値を置きます。

※公式の日本語訳と違い、やわらかく訳し直したバージョンです

ラショナル・ユニファイドプロセス（RUP）

- 当時のIBMは、RUP（プロセス）を中核に添えてツールを売りたかった
  - **Rational Rose**: UMLベースのソフトウェア設計およびモデリングツール。
  - **Rational Software Architect (RSA)**: モデル駆動型開発のための包括的な設計およびモデリングツール。
  - **Rational RequisitePro**: 要件収集、管理、および追跡のためのツール。
  - **Rational DOORS**: 大規模なシステムやソフトウェアプロジェクトのための要件管理ツール。
  - **Rational ClearCase**: バージョン管理、構成管理、およびソフトウェアビルドのためのツール。
  - **Rational Synergy**: 構成管理および変更管理ツール。
  - **Rational ClearQuest**: バグ追跡および変更管理のためのツール。
  - **Rational Application Developer (RAD)**: JavaおよびWebアプリケーションの開発のための統合開発環境。
- RUPは、複雑さと導入コスト高さにより徐々に衰退する
  - RUPはその包括的な性質ゆえに非常に複雑であり、完全に導入するためには多くのトレーニングと管理が必要です。この複雑さが、多くの組織にとって導入のハードルとなりました。
  - RUPを効果的に導入・運用するためには、専用のツールやコンサルティングが必要であり、これが高いコストを伴いました。特に中小企業にとっては、コストの問題が大きな障害となりました。

1978

1979

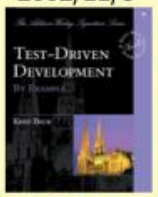
1982

1990/10/10

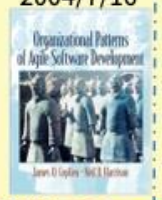
1992/3/13

1996/4/1

2002/11/8



2004/7/16



2006/10/19

パソコン通信

携帯電話・PHSの普及、インターネット黎明期

携帯電話の多機能化、ブロー

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始による
ポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代iMac
（1998～）

F501i HYPER
iモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
（2001～）

初代iPhon
（2007/

40年前

30年前

20年前

1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006

」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IP

えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：米国防総省(DoD)が CMMレベル3以上を防衛関連のソフ
条件として要求し始める。

関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代～：米国を中心にでアジャイル手法が急速

2000年代～：日本は大企業を中心に、ウォータフォ

家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

2000/3～：ITバブル崩壊

1979～：日本の製造業の高品質、ものづくりの強さを研究。

1991～：バブル崩壊「失われた○○年」

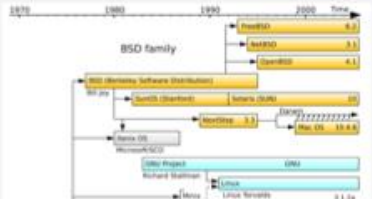
2001/2/11：アジャイルソフトウェア開発宣言

2000年代

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

ng Institute



flickr
リリース
(2004/2)

aws
AWS

ターニングポイント

US 米国で起きたこと（1990年代）

- 1990年代のアメリカでは、ウォーターフォール開発のリスクを軽減するための新たな開発手法が次々と登場した。
- 代表的なものに、
  - **インクリメンタルモデル**（小さく作って段階的に完成させる）
  - **スパイラルモデル**（Boehm, 1988／リスク駆動で反復的に進める）
  - **RUP（Rational Unified Process）**（統制された反復開発、ユースケース駆動）

  などがある。
- これらはいずれも、ウォーターフォールの欠点を補うための工学的進化であり、のちに登場するアジャイル開発の基盤となった。
- すなわち1990年代は、**「リスクを小さく刻む文化」への転換期**であったと言える。

ソフトウェア関連の論文・文献


1970

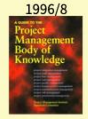

1976


1981


1987


1989/02/01


1994/10/31


1996/8


2001/5/18


2003/9/1


2005/10


2009/11/27


2011/7/16


2014/8/18


2017/6/22


2019/9/17


2021/3/15


2023/11/21


2025/6/25


1977


1979


1984


1988/3/1


1995/10/1


2000/12/1


2003/9/1


2005/10/7


2008/12/30


2010/10/12


2013/1/10


2016/10/6


2018/4/1


2019/9/30


2021/12/1


2023/11/28


2024/12/25


1978/5/1


1986


1989


1995


1999

2002

2004/11/16

2010/7/27

2012/3/15

2013/8/1

2018/3/27

2021/8/1

2022/12/17

2024/7/11

1975

1978

1979

1982

1990/10/10

1992/3/13

1996/4/1

2002/11/8

2004/7/16

2006/10/19

2010～

2011/10/18

2014/4/22

2017/10/14

2018/11/22

2021/12/10

2024/9/13

ICTの進化

パソコン通信

携帯電話・PHSの普及、インターネット黎明期

携帯電話の多機能化、ブロードバンド時代

スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：
IBM System 360とVT100（1970～1980）

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始による
ポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代iMac
（1998～）

F501i HYPER
iモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
（2001～）

初代iPhone発表
（2007/1/9）

初代iPad発表
（2010/4/3）

Samsung
Galaxy S II
（2011/5/2）

Netflixが日本に上陸(2015/9/2)
※では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max
（2024）

西暦

50年前 40年前 30年前 20年前 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

主な出来事

1974年：「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年～：米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化

1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代～：米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代～：日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3～2023/5：COVID-19

1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

2000/3～：ITバブル崩壊

2010年以降～：日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979～：日本の製造業の高品質、ものづくりの強さを研究。

1991～：バブル崩壊「失われた○○年」

2001/2/11：アジャイルソフトウェア開発宣言

2000年代後半～：クラウドファースト・クラウドネイティブ時代に突入

2021～：AI（GenAI）が前提の時代へ

2008/9～：リーマンショック

2018：マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

2009：Flickrのエンジニアである John AllspawとPaul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

© The Deming Institute

“トリニトロン”
日本製品が欧米で人気

"ウォークマン"
1号機 TPS-L2
（1979）

Toyota Corolla
Liftback SR5 001
（1980～）

世界初のポータブル
CDプレーヤー
D-50（1984）

Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
（1996）

Bugzilla
リリース
（2000）

Jira
リリース
（2002）

Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
（1995）

Subversionリリース
（2000）

CVS 誕生
（1990）

flickr
リリース
（2004/2）

AWS
サービス開始
（2006）

Google Cloud
サービス開始
（2008）

Azure
サービス開始
（2010）

Selenium
（2004）

REDMINE
（2006）

trac
（2004）

GitHub
リリース
（2008）

git
（2005）

AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）

GitLab
（2011）
Jenkins 誕生
（2011）

GitHub Actions
（2018）

DATADOG
（2018）

Findy
Team+
Team+
（2021/10）

Claude 1
（2023/3/14）

Cline
（2024/7）

GitHub Copilot
（2021/6/29）

Gemini 1
（2023/12/6）

ChatGPT
一般公開
（2022/11/30）

Devin
（2024/3/12）

1990～2000：第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

2009～2020：第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

2025/1/3：Agile AllianceがPMBOKなどを策定するPMIに加盟

ソフトウェア開発現代史年表 Ver2.09

ソフトウェア関連の論文・文献

1970 1976 1981 1987 1989/02/01 1994/10/31 1996/8 2001/5/18 2003/9 2005/10 2009/11/27 2011/7/16 2014/8/18 2017/6/22 2019/9/17 2021/3/15 2023/11/21 2025/6/25

1977 1979 1984 1988/3/1 1995/10/1 2000/12/1 2003/9/1 2005/10/7 2008/12/30 2010/10/12 2013/1/10 2016/10/6 2018/4/1 2019/9/30 2021/12/1 2023/11/28 2024/12/25

1978/5/1 1986 1989 1995 1999 2002 2004/11/16 2010/7/27 2012/3/15 2013/8/1 2018/3/27 2021/8/1 2022/12/17 2024/7/11

1975 1978 1979 1982 1990/10/10 1992/3/13 1996/4/1 2002/11/8 2004/7/16 2006/10/19 2010〜 2011/10/18 2014/4/22 2017/10/14 2018/11/22 2021/12/10 2024/9/13

ICTの進化

パソコン通信 | 携帯電話・PHSの普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：IBM System 360とVT100（1970〜1980）
PC/AT互換機が誕生（1981〜）
PC-9800シリーズ（1982〜2003）
Apple Macintosh（1984〜）
ワークステーションの時代：Sun SPARCstation（1989〜1994）
数字送信の開始によるポケベルブーム（日本）（1992〜1996）
テレホーダイ（1995〜2023）
初代iMac（1998〜）
F501i HYPERiモード開始（1999/2/22）
家庭向けADSL・FTTHの普及、ブロードバンド時代に突入（2001〜）
初代iPhone発表（2007/1/9）
初代iPad発表（2010/4/3）
Samsung Galaxy S II（2011/5/2）
Netflixが日本に上陸(2015/9/2)※🇺🇸では2007/1にサービスイン
ARMアーキテクチャのSoC Apple M1 生産（2020）
MacBook Pro M4 Max（2024）

西暦

50年前 | 40年前 | 30年前 | 20年前 | 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

主な出来事

1974年：🇺🇸「TCP/IP」の最初の仕様がRFC 675として公開
1982年〜：🇺🇸米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。
★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年〜：🇺🇸米国防総省(DoD)がウォーターフォールを採用
1980年後半〜1990年前半：🇺🇸米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生
1997〜2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。
2010年〜：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化

1970〜1980年代：🌎多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる
1990年代：🇺🇸ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）
2000年代〜：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。
2000年代〜：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。
2020/3〜2023/5：🌎COVID-19

1970年代後半〜1980年代末：🇯🇵日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……
1988〜：🇺🇸日本を含む諸外国へ「通商法スーパー301条」発動
2000/3〜：🇺🇸ITバブル崩壊
2010年以降〜：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979〜：🇺🇸日本の製造業の高品質、ものづくりの強さを研究。
1991〜：🇯🇵バブル崩壊「失われた○○年」
2001/2/11：🌎アジャイルソフトウェア開発宣言
2000年代後半〜：🌎クラウドファースト・クラウドネイティブ時代に突入
2021〜：🌎AI（GenAI）が前提の時代へ

1980年代後半〜1990年代前半：🌎UNIX戦争
2008/9〜：🇺🇸リーマンショック
2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

1985〜1990：🇯🇵国家プロジェクト「Σ計画」が頓挫
2009：🇺🇸Flickrのエンジニアである John Allspaw と Paul Hammond が初めてDevOpsに繋がる伝説的な講演を行う

© The Deming Institute


“トリニトロン”
日本製品が欧米で人気


"ウォークマン"
1号機 TPS-L2
(1979)


Toyota Corolla
Liftback SR5 001
(1980〜)


世界初のポータブル
CDプレーヤー
D-50（1984）


Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
(1996)


Bugzilla
リリース
(2000)


Jira
リリース
(2002)


Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
(1991/9/17)


Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
(1995)


Subversionリリース
(2000)

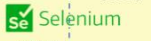

CVS 誕生
(1990)


flickr
リリース
(2004/2)

AWS
サービス開始
(2006)


Google Cloud
サービス開始
(2008)


Azure
サービス開始
(2010)


(2004)


(2006)


(2004)


GitHub
リリース
(2008)


(2005)


欧州の自動車メーカが中心となって公開（2005）


GitLab
(2011)


Jenkins 誕生
(2011)


GitHub Actions
(2018)


DATADOG
(2018)


Findy Team+
Team+
(2021/10)


Claude 1
(2023/3/14)


Cline
(2024/7)


GitHub Copilot
(2021/6/29)


Gemini 1
(2023/12/6)


ChatGPT
一般公開
(2022/11/30)


Devin
(2024/3/12)

2020年〜：🌎第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

1990〜2000：🌎第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

2025/1/3：🇺🇸Agile AllianceがPMBOKなどを策定するPMIに加盟

ソフトウェア開発現代史年表 Ver2.09

20年前にあった有識者からの警告

松原友夫氏

日経bizTech『日本のソフトウエア産業、衰退の真因』2005年10月

- 松原友夫氏（トム・デマルコ本の翻訳でおなじみ）による寄稿
  - https://xtech.nikkei.com/it/article/COLUMN/20070306/264055/
- 要約

① 背景と警告

1990年代、アメリカではエド・ヨードンがソフトウェア産業の危機を警告し、日本を模範的な存在として称賛しましたが、その後、アメリカはオブジェクト指向やアジャイルなどの新しい開発手法により復活を遂げました。

② 日本の遅れ

日本は、90年代においてメインフレームからクライアントサーバーへの移行に乗り遅れ、ソフトウェア開発における国際競争力を失いました。また、技術伝承の断絶や不十分なプロジェクト管理が原因で、開発の質が低下しました。

③ 産業構造の問題

日本のソフトウェア産業には多重下請け構造が蔓延し、派遣プログラマーに依存する状況が問題視されています。これにより、ソフトウェア開発の品質と効率が損なわれ、さらに低賃金の海外労働力に仕事が流れる懸念が高まっています。

④ 提言

日本のソフトウェア産業の再生には、「自立」が必要であると述べています。これは、ソフトウェア企業や技術者が自主的に技術と経営の自立を目指し、プロジェクトマネジメント力を高めることで達成されるべきです。また、政府やユーザー企業も、この自立を支援する役割を果たすべきであると提案しています。

（写真）http://www.kumikomi.net/archives/2005/02/04jasst.php

1978

1979

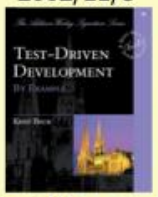
1982

1990/10/10

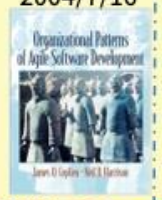
1992/3/13

1996/4/1

2002/11/8



2004/7/16



2006/10/19

パソコン通信

携帯電話・PHSの普及、インターネット黎明期

携帯電話の多機能化、ブロー

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始による
ポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代iMac
（1998～）

F501i HYPER
iモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
（2001～）

初代iPhon
（2007/

40年前

30年前

20年前

1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006

」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IP

えた影響

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：米国防総省(DoD)が CMMレベル3以上を防衛関連のソフ
条件として要求し始める。

関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代～：米国を中心にでアジャイル手法が急速

2000年代～：日本は大企業を中心に、ウォータフォ

家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

2000/3～：ITバブル崩壊

1979～：日本の製造業の高品質、ものづくりの強さを研究。

1991～：バブル崩壊「失われた○○年」

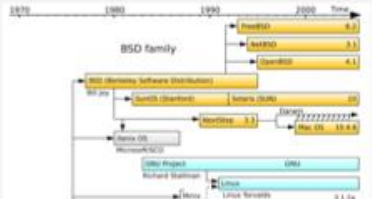
2001/2/11：アジャイルソフトウェア開発宣言

2000年代

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

ng Institute



flickr
リリース
(2004/2)

aws
AWS

の普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

初代iMac
(1998～)

F501i HYPER
iモード開始
(1999/2/22)

家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
(2001～)

初代iPhone発表
(2007/1/9)

初代iPad発表
(2010/4/3)

Samsung
Galaxy S II
(2011/5/2)

Netflixが日本に上陸(2015/9/2)
※🇺🇸では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max
(2024)

20年前　10年前

1998　2000　2002　2004　2006　2008　2010　2012　2014　2016　2018　2020　2022　2024　2026

ューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

1997～2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年～：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化 *_

る開発手法が次々
ど)

2000年代～：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。
2000年代～：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3～2023/5：
🌍COVID-19

2000/3～：🇺🇸ITバブル崩壊

2010年以降～：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

2001/2/11：🌍アジャイルソフトウェア開発宣言

2000年代後半～：🌍クラウドファースト・クラウドネイティブ時代に突入

2021～：🌍AI（GenAI）が前提の時代へ

2008/9～：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

2009：🇺🇸FlickrのエンジニアであるJohn AllspawとPaul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

flickr
リリース
(2004/2)

aws
AWS
サービス開始
(2006)

Google Cloud
サービス開始
(2008)

Azure
サービス開始
(2010)

Findy
Team+
Team+
(2021/10)

Claude 1
(2023/3/14)

Cline
(2024/7)

式リリース
ジェクト指向と
技術を普及させ、
計にも影響を与えた

Bugzilla

JIRA
Jira
リリース
(2002)

Selenium
(2004)

REDMINE
flexible project management
(2006)

GitHub Copilot
(2021/6/29)

Gemini 1
(2023/12/6)

GitLab

ソフトウェア関連の論文・文献

ICTの進化

西暦

主な出来事

1970 1976 1981 1987 1989/02/01 1994/10/31 1996/8 2001/5/18 2003/9/1 2005/10 2009/11/27 2011/7/16 2014/8/18 2017/6/22 2019/9/17 2021/3/15 2023/11/21 2025/6/25

1977 1979 1984 1988/3/1 1995/10/1 2000/12/1 2003/9/1 2005/10/7 2008/12/30 2010/10/12 2013/1/10 2016/10/6 2018/4/1 2019/9/30 2021/12/1 2023/11/28 2024/12/25

1978/5/1 1986 1989 1995 1999 2002 2004/11/16 2010/7/27 2012/3/15 2013/8/1 2018/3/27 2021/8/1 2022/12/17 2024/7/11

1975 1978 1979 1982 1990/10/10 1992/3/13 1996/4/1 2002/11/8 2004/7/16 2006/10/19 2010〜 2011/10/18 2014/4/22 2017/10/14 2018/11/22 2021/12/10 2024/9/13

パソコン通信 | 携帯電話・PHSの普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：
IBM System 360とVT100（1970〜1980）

PC/AT互換機が誕生（1981〜）

PC-9800シリーズ（1982〜2003）

Apple Macintosh
（1984〜）

ワークステーションの時代：
Sun SPARCstation
（1989〜1994）

数字送信の開始によるポケベルブーム（日本）
（1992〜1996）

テレホーダイ
（1995〜2023）

初代iMac
（1998〜）

F501i HYPERiモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、ブロードバンド時代に突入
（2001〜）

初代iPhone発表
（2007/1/9）

初代iPad発表
（2010/4/3）

Samsung Galaxy S II
（2011/5/2）

Netflixが日本に上陸(2015/9/2)
※🇺🇸では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max
（2024）

50年前 | 40年前 | 30年前 | 20年前 | 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

1974年：🇺🇸「TCP/IP」の最初の仕様がRFC 675として公開

1982年〜：🇺🇸米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年〜：🇺🇸米国防総省(DoD)がウォーターフォールを採用

1980年後半〜1990年前半：🇺🇸米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997〜2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年〜：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化

1970〜1980年代：🌎多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：🇺🇸ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代〜：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代〜：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3〜2023/5：🌎COVID-19

1970年代後半〜1980年代末：🇯🇵日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988〜：🇺🇸日本を含む諸外国へ「通商法スーパー301条」発動

2000/3〜：🇺🇸ITバブル崩壊

2010年以降〜：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979〜：🇺🇸日本の製造業の高品質、ものづくりの強さを研究。

1991〜：🇯🇵バブル崩壊「失われた○○年」

2001/2/11：🌎アジャイルソフトウェア開発宣言

2000年代後半〜：🌎クラウドファースト・クラウドネイティブ時代に突入

2021〜：🌎AI（GenAI）が前提の時代へ

2008/9〜：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半〜1990年代前半：🌎UNIX戦争

1985〜1990：🇯🇵国家プロジェクト「Σ計画」が頓挫

2009：🇺🇸Flickrのエンジニアである John Allspaw と Paul Hammond が初めてDevOpsに繋がる伝説的な講演を行う

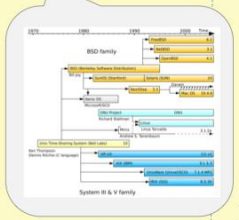

“トリニトロン”
日本製品が欧米で人気


"ウォークマン"
1号機 TPS-L2
（1979）


Toyota Corolla
Liftback SR5 001
（1980〜）


世界初のポータブル
CDプレーヤー
D-50（1984）

© The Deming Institute



Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
（1996）

Bugzilla
リリース
（2000）

Jira
リリース
（2002）

Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
（1995）

Subversionリリース
（2000）

CVS 誕生
（1990）

flickr
リリース
（2004/2）

AWS
サービス開始
（2006）

Google Cloud
サービス開始
（2008）

Azure
サービス開始
（2010）

Selenium
（2004）

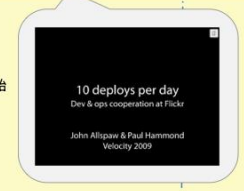
REDMINE
（2006）

trac
（2004）

git
（2005）

GitHub
リリース
（2008）

AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）



GitLab
（2011）

Jenkins 誕生
（2011）

GitHub Actions
（2018）

DATADOG
（2018）

Findy Team+
Team+
（2021/10）

Claude 1
（2023/3/14）

Cline
（2024/7）

GitHub Copilot
（2021/6/29）

Gemini 1
（2023/12/6）

ChatGPT
一般公開
（2022/11/30）

Devin
（2024/3/12）

1990〜2000：🌎第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

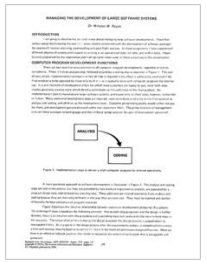
2009〜2020：🌎第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

2025/1/3：🇺🇸Agile AllianceがPMBOKなどを策定するPMIに加盟

ソフトウェア開発現代史年表 Ver2.09

ハードウェアのとてつもない進化

メインフレーム時代
IBM System 360 at USDA

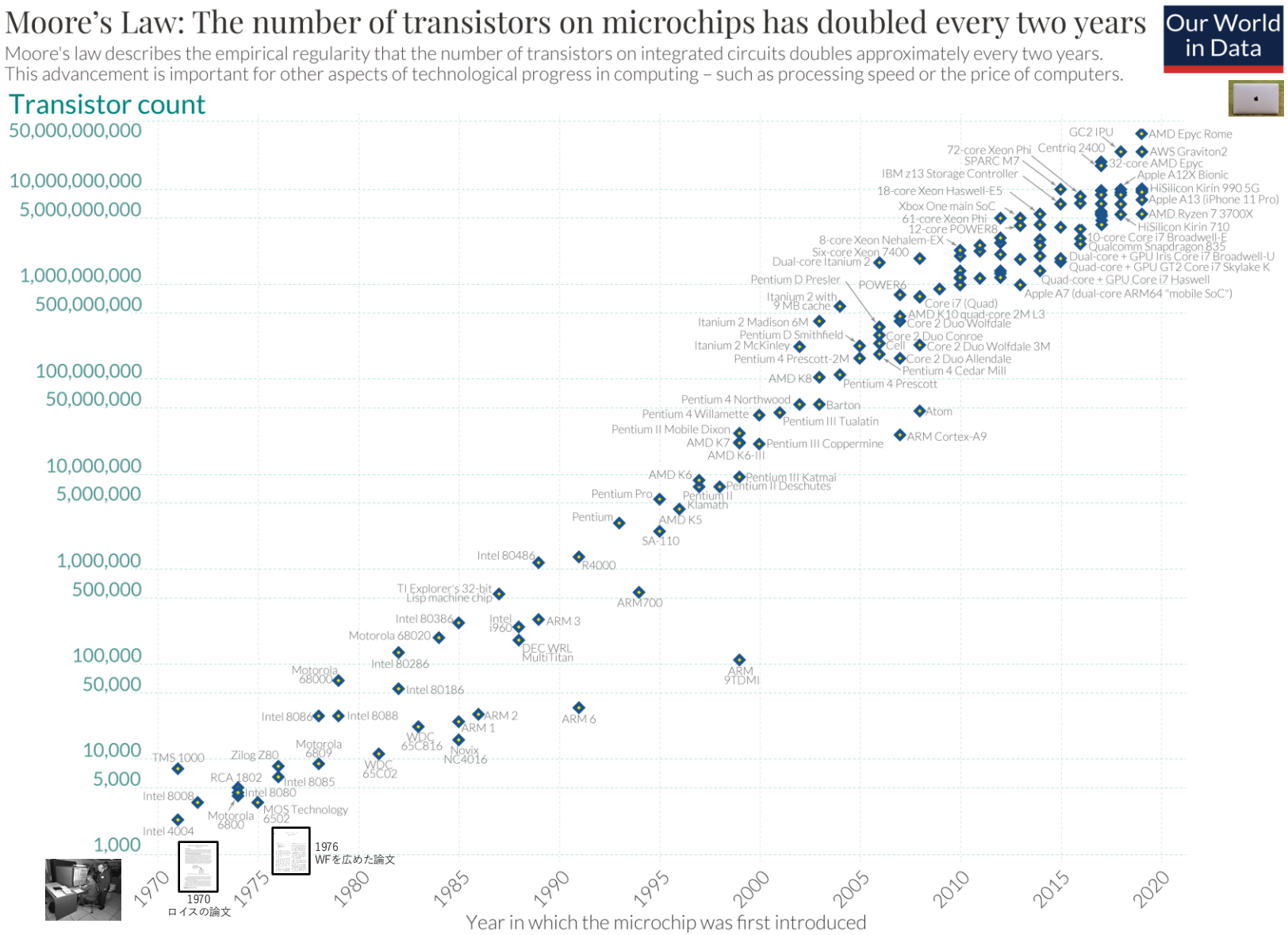


1970
ロイスの論文

指標	IBM System/360	MacBook Pro M4 Max（2024-2025）	比較結果
登場年	1964年	2024年11月	約60年の差
用途	メインフレーム、企業や政府の大規模データ処理	ポータブルコンピュータ、プロフェッショナル用途	-
処理速度	34,500命令/秒（Model 30）～16.6 MIPS（Model 91）	数千億命令/秒以上	約100,000 倍以上
CPU	8～64KB メモリのシステム	16コアCPU（12 Performanceコア + 4 Efficiencyコア）、最大4.5GHz	最新の3nmプロセス
GPU	なし（専用グラフィックス処理なし）	40コアGPU、ハードウェアレイトレーシング、Dynamic Caching	革新的グラフィックス処理
Neural Engine	なし	16コアNeural Engine（38兆回/秒の演算）	AI処理専用ハードウェア
メモリ	8KB～1MB（モデルにより異なる）	最大128GB ユニファイドメモリ メモリ帯域幅: 546GB/s	約128,000倍
ストレージ	数MB～数GB（磁気ドラム・磁気テープ）	最大8TB NVMe SSD	約8,000倍以上、超高速アクセス
サイズ	非常に大きく、専用のコンピュータルームが必要（数トン）	厚さ1.68cm、重さ2.16kg	持ち運び可能
消費電力	数キロワット～数十キロワット	約100W（最大時）	約1/100以下
コスト	$2,200,000～$12,500,000（当時の価格、モデルにより異なる）	$3,499～（基本構成）、最大構成で約$7,500+	大幅な低価格化と高性能化の両立
ディスプレイ	なし（コンソールパネルとランプのみ）	16.2インチ Liquid Retina XDR、3456×2234ピクセル SDR 1000nits、HDR 1600nits	高輝度・高解像度、統合ディスプレイ
カメラ	なし	12MP Center Stage カメラ （自動フレーミング機能付き）	ビデオ会議対応
接続性	専用チャネル、パンチカードリーダー、磁気テープ	Thunderbolt 5×3、HDMI 2.1、Wi-Fi 6E、Bluetooth 5.3、MagSafe 3	最大120Gb/s転送速度、無線・有線統合
バッテリー	なし（常時AC電源接続）	最大24時間のバッテリー駆動	完全なモバイル環境
AI機能	なし	Apple Intelligence統合、オンデバイスAI処理	プライバシー保護AI

ハードウェアのとてつもない進化

- トランジスタ数（ムーアの法則）の推移
  - オックスフォード大学系のデータサイトOur World in Dataでは、1970年代以降のマイクロプロセッサ上のトランジスタ数を示すグラフが公開されています
  - このグラフは縦軸を対数スケールでプロットしており、約2年ごとにチップ上のトランジスタ数が2倍になるムーアの法則が、50年以上にわたってほぼ一貫して成立してきたことを示しています



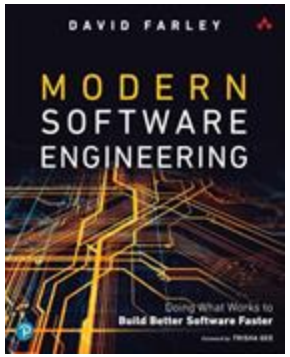
https://ourworldindata.org/moores-law#moore-s-law-has-held-true-for-more-than-half-a-century

Modern Software Engineering

私の印象では、ソフトウェア産業は学ぶことも進化することもなかなかできないで苦闘しているように見えます。
この相対的な停滞は、コードを実行するハードウェアのとてつもない進化によって見えなくされているのです。

David Farley（2021）

David Farley



引用：Farley, D.(2022), 長尾高弘 (訳). 継続的デリバリーのソフトウェア工学: もっと早く、もっと良いソフトウェアを作るための秘訣 (Nagao, T., Trans.). 日経BP社. (Original work published 2021) p.69

ソフトウェア関連の論文・文献

ICTの進化

パソコン通信 / 携帯電話・PHSの普及、インターネット黎明期 / 携帯電話の多機能化、ブロードバンド時代 / スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：
IBM System 360とVT100（1970〜1980）

PC/AT互換機が誕生（1981〜）

PC-9800シリーズ（1982〜2003）

Apple Macintosh
（1984〜）

ワークステーションの時代：
Sun SPARCstation
（1989〜1994）

数字送信の開始による
ポケベルブーム（日本）
（1992〜1996）

テレホーダイ
（1995〜2023）

初代iMac
（1998〜）

F501i HYPER
iモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
（2001〜）

初代iPhone発表
（2007/1/9）

初代iPad発表
（2010/4/3）

Samsung
Galaxy S II
（2011/5/2）

Netflixが日本に上陸(2015/9/2)
※🇺🇸では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max
（2024）

西暦

50年前 / 40年前 / 30年前 / 20年前 / 10年前

1972 | 1974 | 1976 | 1978 | 1980 | 1982 | 1984 | 1986 | 1988 | 1990 | 1992 | 1994 | 1996 | 1998 | 2000 | 2002 | 2004 | 2006 | 2008 | 2010 | 2012 | 2014 | 2016 | 2018 | 2020 | 2022 | 2024 | 2026

主な出来事

1974年：🇺🇸「TCP/IP」の最初の仕様がRFC 675として公開

1982年〜：🇺🇸米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年〜：🇺🇸米国防総省(DoD)がウォーターフォールを採用

1980年後半〜1990年前半：🇺🇸米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997〜2010年代：🇺🇸米国防総省(DoD)が CMM レベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年〜：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化 *

1970〜1980年代：🌏多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：🇺🇸ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代〜：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代〜：🇯🇵日本は大企業を中心に、ウォーターフォールモデルを採用し続ける。

2020/3〜2023/5：
🌏 COVID-19

1970年代後半〜1980年代末：🇯🇵日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988〜：🇺🇸日本を含む諸外国へ「通商法スーパー301条」発動

2000/3〜：🇺🇸ITバブル崩壊

2010年以降〜：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979〜：🇺🇸日本の製造業の高品質、ものづくりの強さを研究。

1991〜：🇯🇵バブル崩壊「失われた〇〇年」

2001/2/11：🌏アジャイルソフトウェア開発宣言

2000年代後半〜：🌏クラウドファースト・クラウドネイティブ時代に突入

2021〜：🌏AI（GenAI）が前提の時代へ

1980年代後半〜1990年代前半：🌏UNIX戦争

2008/9〜：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

1985〜1990：🇯🇵国家プロジェクト「Σ計画」が頓挫

2009：🇺🇸Flickrのエンジニアである John AllspawとPaul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

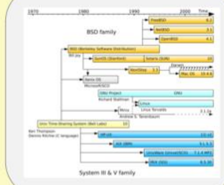
© The Deming Institute

“トリニトロン”
日本製品が欧米で人気

"ウォークマン"
1号機 TPS-L2
（1979）

Toyota Corolla
Liftback SR5 001
（1980〜）

世界初のポータブル
CDプレーヤー
D-50（1984）



CVS 誕生
（1990）

Linux 0.01 リリース
UNIXが商業化・断片化していく中、
Linuxはオープンソースの力で
統一的な開発基盤となり、
世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSに
TCP/IPが標準搭載され
ワークステーション並みに
（1995）

Java v1.0 正式リリース
Javaはオブジェクト指向と
仮想マシン技術を普及させ、
後の言語設計にも影響を与えた
（1996）

Bugzilla
リリース
（2000）

Jira
リリース
（2002）

Subversionリリース
（2000）

flickr
リリース
（2004/2）

Selenium
（2004）

REDMINE
（2006）

trac
（2004）

git
（2005）

AUTOMOTIVE SPICE®
欧州の自動車メーカ
が中心となって公開（2005）

aws
AWS
サービス開始
（2006）

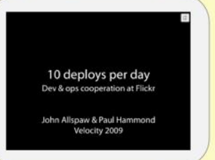
Google Cloud
サービス開始
（2008）

Azure
サービス開始
（2010）

GitHub
リリース
（2008）

GitLab
（2011）

Jenkins 誕生
（2011）



GitHub Actions
（2018）

DATADOG
（2018）

Findy Team+
Team+
（2021/10）

GitHub Copilot
（2021/6/29）

Claude 1
（2023/3/14）

Gemini 1
（2023/12/6）

Cline
（2024/7）

ChatGPT
一般公開
（2022/11/30）

Devin
（2024/3/12）

Agile Alliance + Project Management Institute

1990〜2000：🌏第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

2009〜2020：🌏第二次ブラウザ戦争
Google Chromeの躍進、Internet Explorerの衰退

2025/1/3：🇺🇸Agile AllianceがPMBOKなどを策定するPMIに加盟

ソフトウェア開発現代史年表 Ver2.09

の普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

初代iMac
(1998～)

F501i HYPER
iモード開始
(1999/2/22)

家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
(2001～)

初代iPhone発表
(2007/1/9)

初代iPad発表
(2010/4/3)

Samsung
Galaxy S II
(2011/5/2)

Netflixが日本に上陸(2015/9/2)
※🇺🇸では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max
(2024)

20年前　10年前

1998　2000　2002　2004　2006　2008　2010　2012　2014　2016　2018　2020　2022　2024　2026

ューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

1997～2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年～：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化 *_

る開発手法が次々
ど)

2000年代～：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。
2000年代～：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3～2023/5：
🌍COVID-19

2000/3～：🇺🇸ITバブル崩壊

2010年以降～：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

2001/2/11：🌍アジャイルソフトウェア開発宣言

2000年代後半～：🌍クラウドファースト・クラウドネイティブ時代に突入

2021～：🌍AI（GenAI）が前提の時代へ

2008/9～：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

2009：🇺🇸FlickrのエンジニアであるJohn AllspawとPaul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

flickr
リリース
(2004/2)

aws
AWS
サービス開始
(2006)

Google Cloud
サービス開始
(2008)

Azure
サービス開始
(2010)

Findy
Team+
Team+
(2021/10)

Claude 1
(2023/3/14)

Cline
(2024/7)

式リリース
ジェクト指向と
技術を普及させ、
計にも影響を与えた

Bugzilla

JIRA
Jira
リリース
(2002)

Selenium
(2004)

REDMINE
flexible project management
(2006)

GitHub Copilot
(2021/6/29)

Gemini 1
(2023/12/6)

GitLab

ターニングポイント

JP 日本はこの時期から立ち往生している……

- 一方、日本ではこの**過渡期の開発手法の発展**にほとんど取り組まなかった。
  理由はいくつか考えられる。

ロ 製造業モデルの成功体験

- 日本のソフトウェア開発は、製造業における工程管理や品質保証の発想をそのまま取り入れた。
- 本来のトヨタ生産方式は、現場での小さな試行と改善を重ねて品質を高める“学習の仕組み”であったが、ソフトウェア産業ではその思想が十分に継承されず、**形式的な標準化や事前設計の徹底だけが強調された。**
- 結果として、「最初に正しく設計すること」が理想とされる文化へと傾いていった。

ロ ベンダー構造（多重下請け）

- 要求定義と実装が分断されており、反復的に学習するサイクルを回すことが難しかった。
- RUPのように**開発プロセスを段階的に検証しながら体系的に進めるモデル**は、構造的に根付きにくかった。

ロ リスク文化の違い

- 「失敗を前提に改善する」よりも「最初から失敗しないように設計する」ことが重視された。
- スパイラルやインクリメンタルのように“試行錯誤を制度化する”発想は受け入れられにくかった。

- これらの停滞は、ハードウェアのとてつもない進化によって見えなくされてしまった。

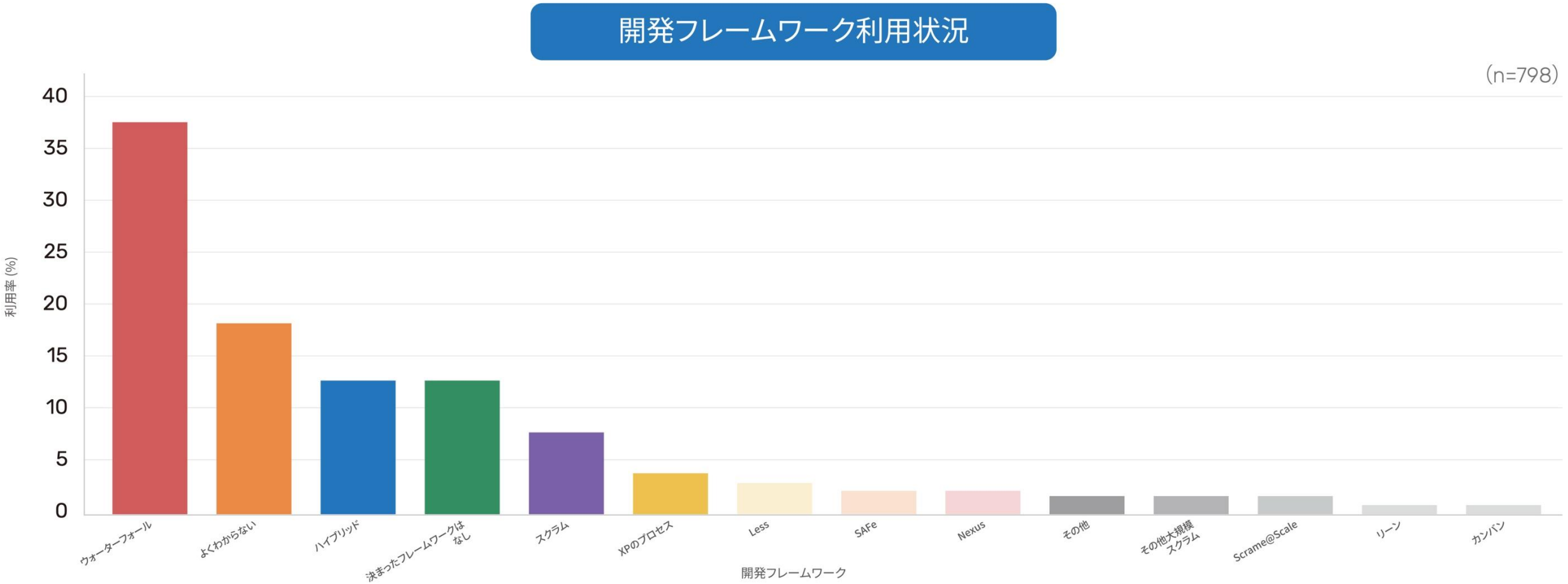
ソフトウェア開発における「開発生産性」に関する実態調査レポート

- 調査対象： ソフトウェア開発（組み込み開発を含む）に直接関わるエンジニア、プロダクトマネージャー、プロジェクトマネージャー、エンジニアリングマネージャー、開発責任者など
- 調査方法： インターネット調査
- 調査期間： 2025年4月2日(水)～2025年5月21日(水)
- 調査主体： ファインディ株式会社
- 実査委託先： GMOリサーチ&AI株式会社
- 有効回答数： 798名

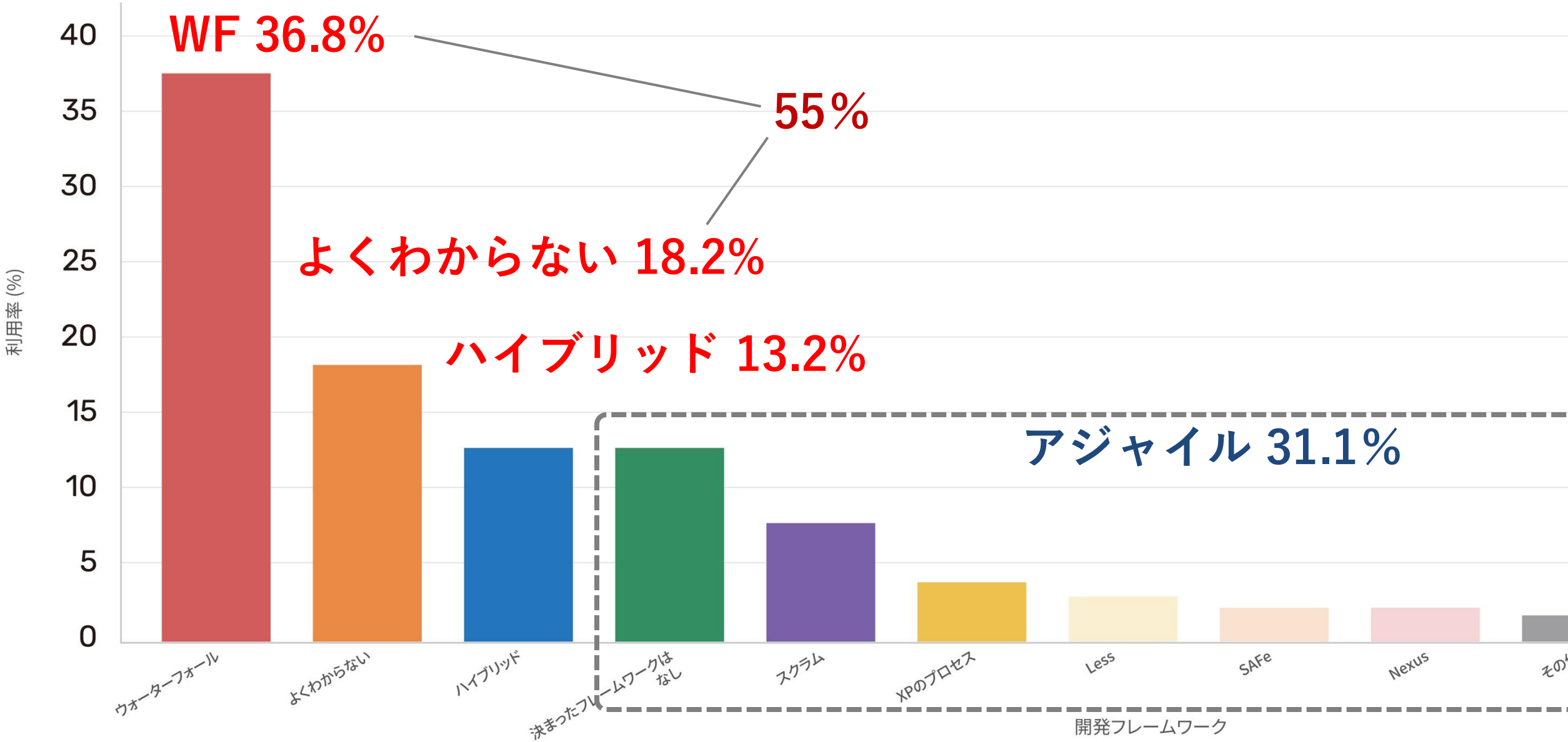
※本調査はファインディ株式会社の利用ユーザーに対する調査ではないことをご留意ください。

Findy

【問】あなたのチームで主に採用している開発フレームワーク（開発手法）を以下から選んでください。なお、XP（エクストリームプログラミング）のプラクティスはどのフレームワークでも活用可能であり、併用されていることを前提としています。（単一回答）



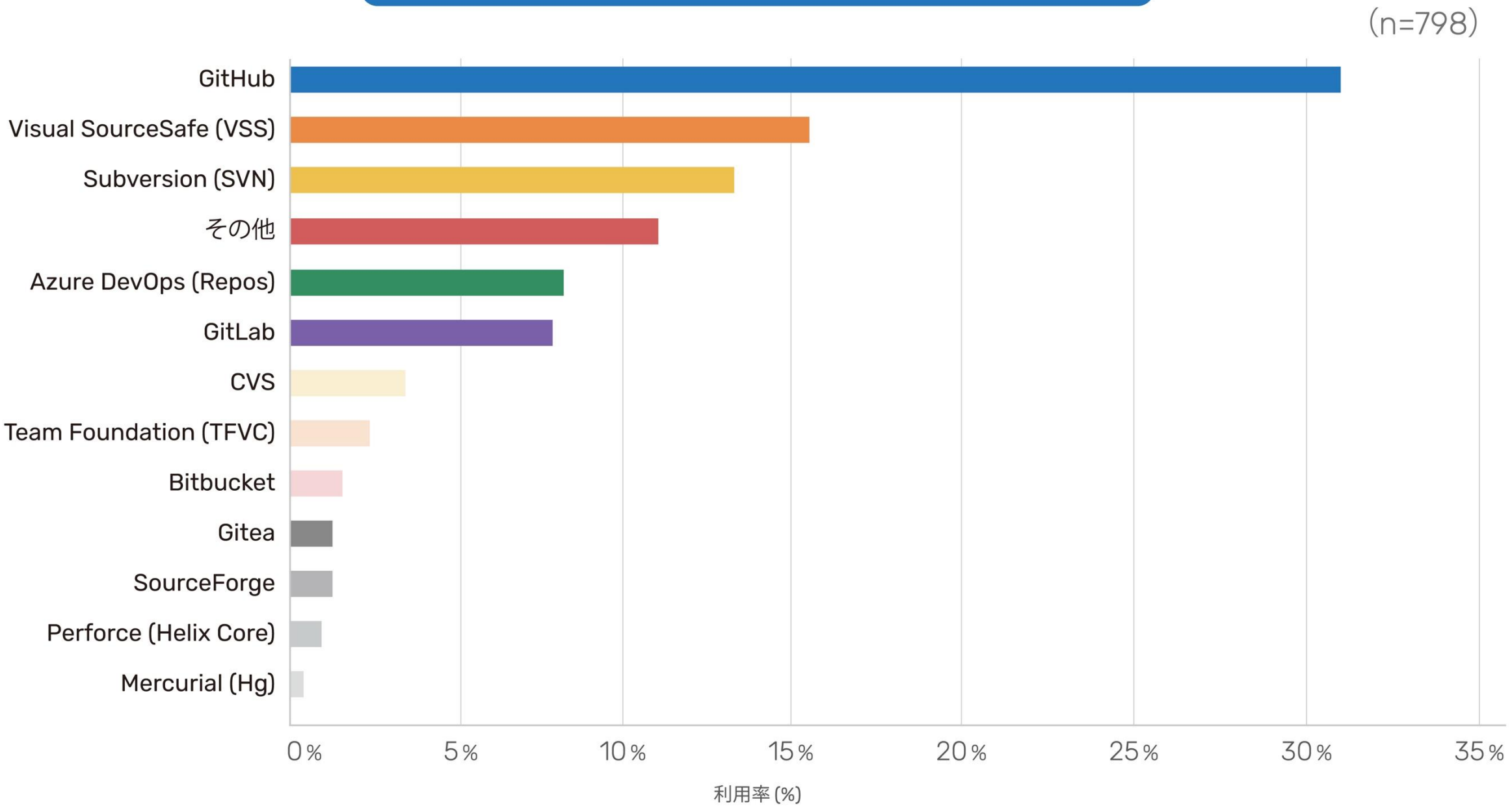
開発フレームワーク利用状況



開発フレームワーク利用状況

開発フレームワーク	回答者数	利用率	統計的有意性	分類
ウォーターフォール	294名	36.8%	p < 0.001	従来型手法
開発フレームワークはよくわからない	145名	18.2%	p < 0.001	組織課題
ウォーターフォールとアジャイルのハイブリッド	105名	13.2%	p < 0.001	混合手法
【アジャイル開発】決まったフレームワークはない	105名	13.2%	p < 0.001	アジャイル系
【アジャイル開発】スクラム	54名	6.8%	p < 0.001	アジャイル系
【アジャイル開発】XPのプロセス	30名	3.8%	p < 0.001	アジャイル系
【アジャイル開発】大規模スクラム：LeSS	19名	2.4%	p < 0.05	アジャイル系
【アジャイル開発】大規模スクラム：SAFe	10名	1.3%	p < 0.05	アジャイル系
【アジャイル開発】大規模スクラム：Nexus	10名	1.3%	p < 0.05	アジャイル系
【アジャイル開発】大規模スクラム：Scrum@Scale	6名	0.8%	n.s.	アジャイル系
【アジャイル開発】大規模スクラム：その他	6名	0.8%	n.s.	アジャイル系
リーン	3名	0.4%	n.s.	アジャイル系
カンバン	2名	0.3%	n.s.	アジャイル系
その他	8名	1.0%	n.s.	その他

ソースコード管理ツール利用状況



ソースコード管理ツール利用状況

ツール	利用率	サポート状況	最終更新	セキュリティリスク
Visual SourceSafe (VSS)	15.8%(126人)	サポート完全終了	2005年10月	極めて高い
CVS	3.6%(29人)	事実上の開発停止	2008年5月8日	高い
Subversion (SVN)	13.7%(109人)	限定的メンテナンス	2024年12月8日	中程度

- 従来型ツールの利用は「古い」以上に重大なリスクです。Visual SourceSafeは2005年に更新終了していますが、126人（15.8%）が依然使用していました。
- これらの利用者やSubversion（13.7%）、CVS（3.6%）の組織は、CopilotなどAI時代の支援ツールを活用できず、競争力格差拡大のリスクに直面しています。

従来型ツールの利用による技術的制約とは

■ API仕様の違いによる機能制限

- Visual SourceSafe: COM APIベースのアクセス方式（2005年設計のため、現代的なREST API仕様には非対応）
- Subversion: WebDAV/HTTP APIを実装（一部のJSON形式APIには対応しているが、現代的なWebhook機能は制限的）

→ 結果: **AIツールによるプログラム的なアクセスにおいて、一部機能が制限される場合があります**

■ メタデータ構造の違いによるAI学習への影響

- 従来型ツール: ファイル変更履歴を中心とした情報構造（コミット単位の情報は限定的）
- モダンツール: コミット意図、レビュー履歴、イシュー関連付けなどの豊富なコンテキスト情報を構造化

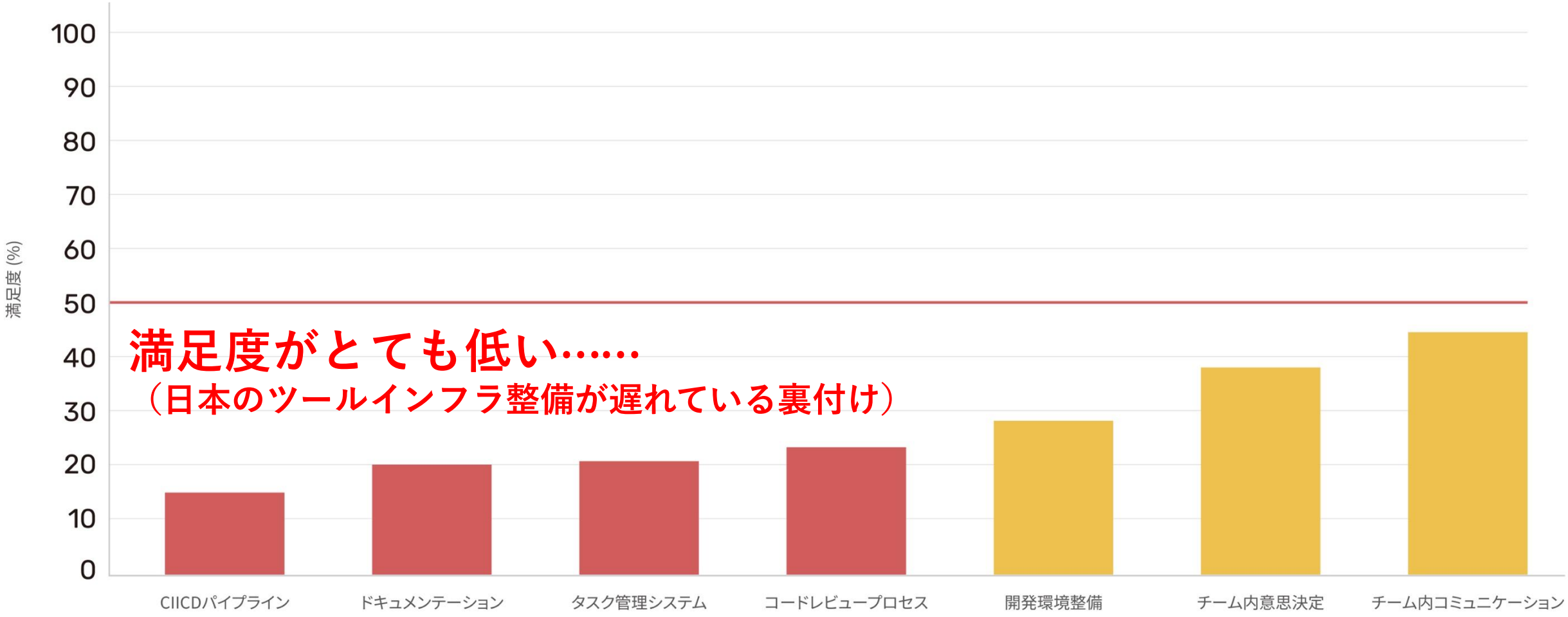
→ 結果: **AIによるコードコンテキストの理解において、活用できる情報量に制約が生じる場合があります**

■ 開発エディタとの統合における制約

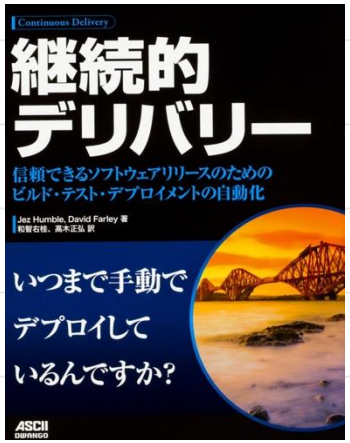
- VS Code、JetBrains IDEs: Git統合を前提とした拡張機能エコシステムが充実
- Cursor: Gitリポジトリ構造に最適化されたAI機能を提供

→ 結果: **従来型ツールでは最新の開発環境の一部機能を十分に活用できない場合があります**

開発環境・プロセス項目



開発環境・プロセス項目



CDは約15年前に登場していた技術だが……

（2010年、日本語訳は2012年）

満足度がとても低い……

（日本のツールインフラ整備が遅れている裏付け）

Findy

技術的発展の階層構造

土台のない現場に、生成AIムーブメントが来ていませんか？

これまでの
プログラミング・パラダイムシフト

※ソフトウェアと設計に対する考え方を根本から変える変化

ソフトウェア関連の論文・文献

ICTの進化

西暦

主な出来事


1970


1976


1981


1987


1989/02/01


1994/10/31


1996/8


2001/5/18


2003/9/1


2005/10


2009/11/27


2011/7/16


2014/8/18


2017/6/22


2019/9/17


2021/3/15


2023/11/21


2025/6/25


1977


1979


1984


1988/3/1


1995/10/1


2000/12/1


2003/9/1


2005/10/7


2008/12/30


2010/10/12


2013/1/10

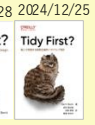

2016/10/6


2018/4/1


2019/9/30


2021/12/1


2023/11/28


2024/12/25


1978/5/1


1986


1989


1995


1999

2002

2004/11/16

2010/7/27

2012/3/15

2013/8/1

2018/3/27

2021/8/1

2022/12/17

2024/7/11

1975

1978

1979

1982

1990/10/10

1992/3/13

1996/4/1

2002/11/8

2004/7/16

2006/10/19

2010～

2011/10/18

2014/4/22

2017/10/14

2018/11/22

2021/12/10

2024/9/13

パソコン通信

携帯電話・PHSの普及、インターネット黎明期

携帯電話の多機能化、ブロードバンド時代

スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：
IBM System 360とVT100（1970～1980）

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh
（1984～）

ワークステーションの時代：
Sun SPARCstation
（1989～1994）

数字送信の開始による
ポケベルブーム（日本）
（1992～1996）

テレホーダイ
（1995～2023）

初代Mac
（1998～）

F501i HYPER
iモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
（2001～）

初代Phone発表
（2007/1/9）

初代iPad発表
（2010/4/3）

Samsung
Galaxy S II
（2011/5/2）

Netflixが日本に上陸(2015/9/2)
※では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max
（2024）

50年前　40年前　30年前　20年前　10年前

1972　1974　1976　1978　1980　1982　1984　1986　1988　1990　1992　1994　1996　1998　2000　2002　2004　2006　2008　2010　2012　2014　2016　2018　2020　2022　2024　2026

1974年：🇺🇸「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：🇺🇸米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年～：🇺🇸米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：🇺🇸米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年～：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化。

1970～1980年代：🌐多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：🇺🇸ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUPなど）

2000年代～：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代～：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3～2023/5：🌐COVID-19

1970年代後半～1980年代末：🇯🇵日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：🇺🇸日本を含む諸外国へ「通商法スーパー301条」発動

2000/3～：🇺🇸ITバブル崩壊

2010年以降～：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979～：🇺🇸日本の製造業の高品質、ものづくりの強さを研究。

1991～：🇯🇵バブル崩壊「失われた○○年」

2001/2/11：🌐アジャイルソフトウェア開発宣言

2000年代後半～：🌐クラウドファースト・クラウドネイティブ時代に突入

2021～：🌐AI（GenAI）が前提の時代へ

2008/9～：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半～1990年代前半：🌐UNIX戦争

1985～1990：🇯🇵国家プロジェクト「Σ計画」が頓挫

2009：🇺🇸Flickrのエンジニアである John Allspawと Paul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

“トリニトロン”
日本製品が欧米で人気

“ウォークマン”
1号機 TPS-L2
（1979）

Toyota Corolla
Liftback SR5 001
（1980～）

世界初のポータブル
CDプレーヤー
D-50（1984）

© The Deming Institute

Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
（1996）

Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
（1995）

Bugzilla
リリース
（2000）

Jira
リリース
（2002）

Subversionリリース
（2000）

CVS 誕生
（1990）

1990～2000：🌐第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

リリース
（2004/2）

AWS
サービス開始
（2006）

Google Cloud
サービス開始
（2008）

Azure
サービス開始
（2010）

（2004）

（2006）

（2004）

（2005）

欧州の自動車メーカが中心となって公開（2005）

GitHub
リリース
（2008）

（2011）

Jenkins 誕生
（2011）

GitHub Actions
（2018）

（2018）

2009～2020：🌐第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

Team+
（2021/10）

Claude 1
（2023/3/14）

Cline
（2024/7）

GitHub Copilot
（2021/6/29）

Gemini 1
（2023/12/6）

ChatGPT
一般公開
（2022/11/30）

Devin
（2024/3/12）

+

2025/1/3：🇺🇸Agile AllianceがPMBOKなどを策定するPMIに加盟

パラダイムシフト(1) - アセンブリ言語からC言語へ

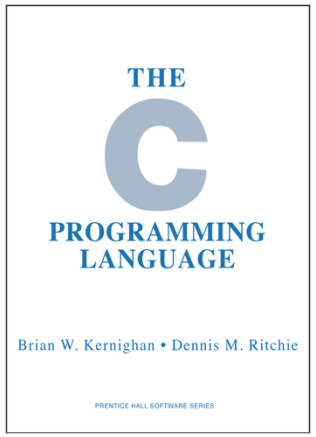
Z80 アッセンブリ

```
org 100h

msg:
    db 'Hello, World!$', 0

start:
    ld de, msg
    call print
    ret

print:
    ld c, 9
    int 21h
    ret
```

C言語

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

K&R版 C言語（1978）



1970年代の終わりから1980年代にかけて、プログラミングはCPUごとに命令体系が異なるアセンブリ言語から、構造化と移植性を備えたC言語へと進化した。
CはUNIXの開発を通じて普及し、ハードウェアに依存しないソフトウェア開発を実現。
この抽象化の進展が、後のオブジェクト指向やモダン言語の基盤となった。

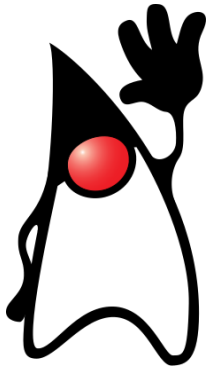
パラダイムシフト(2) - 手続き型からオブジェクト指向プログラミングへ

C言語

```c
#include <stdio.h>
#include <string.h>

// "クラス"構造体の定義
typedef struct {
    char message[50];
    void (*print)(struct HelloWorld*); // メソッドを指す関数ポインタ
} HelloWorld;

// メソッド関数の定義
void printMessage(HelloWorld *self) {
    printf("%s\n", self->message);
}

// "コンストラクタ"関数
void HelloWorld_init(HelloWorld *self, const char *msg) {
    strncpy(self->message, msg, sizeof(self->message) - 1);
    self->message[sizeof(self->message) - 1] = '\0'; // 文字列終端の確保
    self->print = printMessage;
}

int main() {
    // オブジェクトの生成と初期化
    HelloWorld hw;
    HelloWorld_init(&hw, "Hello, World!");

    // メソッドの呼び出し
    hw.print(&hw);

    return 0;
}
```

Java

```java
class HelloWorld {
    private String message;
    public HelloWorld(String message) {
        this.message = message;
    }
    public void print() {
        System.out.println(message);
    }
    public static void main(String[] args) {
        HelloWorld hw = new HelloWorld("Hello, World!");
        hw.print();
    }
}
```



1980年代後半～1990年代にかけて、ソフトウェア開発は「手順を組む」手続き型から、「現実世界をモデル化する」オブジェクト指向へと転換した。
Smalltalk（1970年代）や C++（1983年）で芽生えた思想が、Java（1996年リリース）の登場によりWeb時代の標準へ。
データと振る舞いを一体化し、再利用性・保守性・拡張性を重視する設計が主流となった。

ソフトウェア関連の論文・文献

ICTの進化

西暦

主な出来事

1970 1976 1981 1987 1989/02/01 1994/10/31 1996/8 2001/5/18 2003/9/1 2005/10 2009/11/27 2011/7/16 2014/8/18 2017/6/22 2019/9/17 2021/3/15 2023/11/21 2025/6/25

1977 1979 1984 1988/3/1 1995/10/1 2000/12/1 2003/9/1 2005/10/7 2008/12/30 2010/10/12 2013/1/10 2016/10/6 2018/4/1 2019/9/30 2021/12/1 2023/11/28 2024/12/25

1978/5/1 1986 1989 1995 1999 2002 2004/11/16 2010/7/27 2012/3/15 2013/8/1 2018/3/27 2021/8/1 2022/12/17 2024/7/11

1975 1978 1979 1982 1990/10/10 1992/3/13 1996/4/1 2002/11/8 2004/7/16 2006/10/19 2010〜 2011/10/18 2014/4/22 2017/10/14 2018/11/22 2021/12/10 2024/9/13

パソコン通信 | 携帯電話・PHSの普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：
IBM System 360とVT100（1970〜1980）

PC/AT互換機が誕生（1981〜）

PC-9800シリーズ（1982〜2003）

Apple Macintosh
（1984〜）

ワークステーションの時代：
Sun SPARCstation
（1989〜1994）

数字送信の開始による
ポケベルブーム（日本）
（1992〜1996）

テレホーダイ
（1995〜2023）

初代iMac
（1998〜）

F501i HYPER
iモード開始
（1999/2/22）

家庭向けADSL・FTTHの普及、
ブロードバンド時代に突入
（2001〜）

初代iPhone発表
（2007/1/9）

初代iPad発表
（2010/4/3）

Samsung
Galaxy S II
（2011/5/2）

Netflixが日本に上陸(2015/9/2)
※🇺🇸では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max
（2024）

50年前 40年前 30年前 20年前 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

1974年：🇺🇸「TCP/IP」の最初の仕様がRFC 675として公開

1982年〜：🇺🇸米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年〜：🇺🇸米国防総省(DoD)がウォーターフォールを採用

1980年後半〜1990年前半：🇺🇸米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997〜2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年〜：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化

1970〜1980年代：🌎多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：🇺🇸ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代〜：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代〜：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020/3〜2023/5：🌎COVID-19

1970年代後半〜1980年代末：🇯🇵日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988〜：🇺🇸日本を含む諸外国へ「通商法スーパー301条」発動

2000/3〜：🇺🇸ITバブル崩壊

2010年以降〜：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979〜：🇺🇸日本の製造業の高品質、ものづくりの強さを研究。

1991〜：🇯🇵バブル崩壊「失われた○○年」

2001/2/11：🌎アジャイルソフトウェア開発宣言

2000年代後半〜：🌎クラウドファースト・クラウドネイティブ時代に突入

2021〜：🌎AI（GenAI）が前提の時代へ

2008/9〜：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半〜1990年代前半：🌎UNIX戦争

1985〜1990：🇯🇵国家プロジェクト「Σ計画」が頓挫

2009：🇺🇸Flickrのエンジニアである John Allspaw と Paul Hammond が初めてDevOpsに繋がる伝説的な講演を行う

© The Deming Institute


“トリニトロン”
日本製品が欧米で人気


"ウォークマン"
1号機 TPS-L2
（1979）


Toyota Corolla
Liftback SR5 001
（1980〜）


世界初のポータブル
CDプレーヤー
D-50（1984）


Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
（1996）


Bugzilla
リリース
（2000）

JIRA
Jira
リリース
（2002）


Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
（1991/9/17）

Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
（1995）


Subversionリリース
（2000）


CVS 誕生
（1990）

flickr
リリース
（2004/2）

aws
AWS
サービス開始
（2006）

Google Cloud
サービス開始
（2008）

Azure
サービス開始
（2010）

Selenium
（2004）

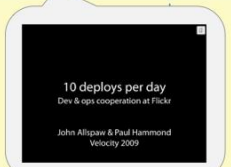
REDMINE
flexible project management
（2006）

trac
（2004）

git
（2005）

GitHub
リリース
（2008）

AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）



GitLab
（2011）


Jenkins 誕生
（2011）

GitHub Actions
（2018）

DATADOG
（2018）

Findy Team+
Team+
（2021/10）

Claude 1
（2023/3/14）

Cline
（2024/7）

GitHub Copilot
（2021/6/29）

Gemini 1
（2023/12/6）

ChatGPT
一般公開
（2022/11/30）

Devin
（2024/3/12）

Agile Alliance + Project Management Institute.

1990〜2000：🌎第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

2009〜2020：🌎第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

2025/1/3：🇺🇸Agile AllianceがPMBOKなどを策定するPMIに加盟

ソフトウェア開発現代史年表 Ver2.09

2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

Pが実装された。

★ピアソンショック（2013年8月1日）

トウェア調達

2010年～：米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化 *

速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

ォールモデルを採用し続ける。

2020/3～2023/5：COVID-19

2010年以降～：日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

代後半～：クラウドファースト・クラウドネイティブ時代に突入

2021～：AI（GenAI）が前提の時代へ

2008/9～：リーマンショック

2018：マイクロソフト、GitHub を 75 億ドルで買収

2009：FlickrのエンジニアであるJohn AllspawとPaul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

oogle Cloud
サービス開始
(2008)

Azure
サービス開始
(2010)

Findy Team+
Team+
(2021/10)

Claude 1
(2023/3/14)

Cline
(2024/7)

GitHub Copilot
(2021/6/29)

Gemini 1
(2023/12/6)

GitLab
(2011)

GitHub Actions
(2018)

GitHub
リリース
(2008)

ChatGPT
一般公開
(2022/11/30)

Devin
(2024/3/12)

Jenkins 誕生
(2011)

DATADOG
(2018)

Agile Alliance + Project Management Institute

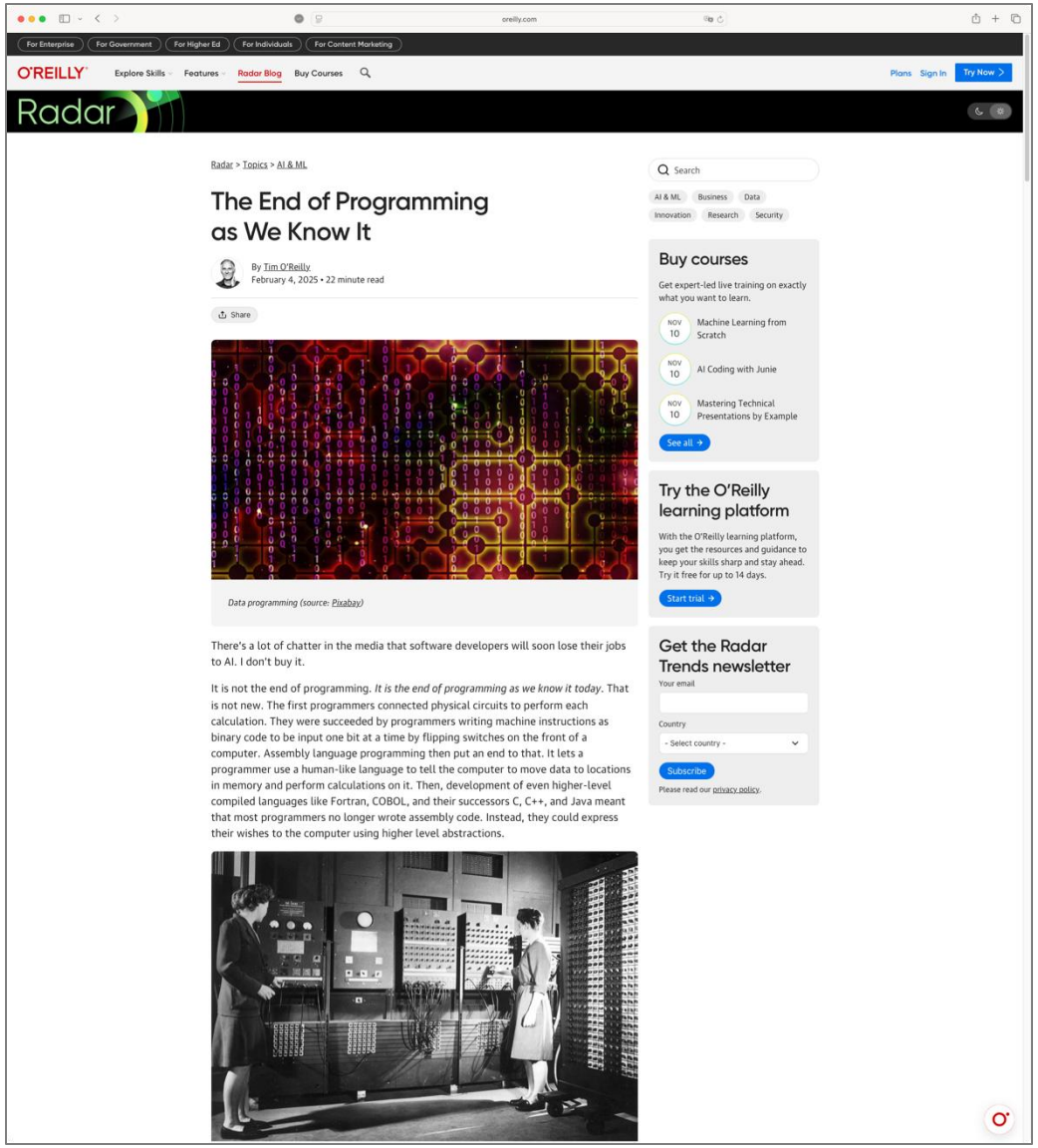
2009～2020：第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

2025/1/3：Agile AllianceがPMBOKなどを策定するPMIに加盟

現在、約30年ぶりの
プログラミング・パラダイムシフト

※ソフトウェアと設計に対する考え方を根本から変える変化

我々の知るプログラミングの終わり（ティム・オライリー / 2025年2月4日）



- メディアではAIによってソフトウェア開発者がまもなく仕事を失うという話が多く出回っています。私はそれを信じていません。
- これはプログラミングの終わりではありません。これは私たちが今日知っているようなプログラミングの終わりなのです。それは新しいことではありません。

2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

Pが実装された。

★ピアソンショック（2013年8月1日）

トウェア調達

2010年～：米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化 *

速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

ォールモデルを採用し続ける。

2020/3～2023/5：COVID-19

2010年以降～：日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

代後半～：クラウドファースト・クラウドネイティブ時代に突入

2021～：AI（GenAI）が前提の時代へ

2008/9～：リーマンショック

2018：マイクロソフト、GitHub を 75 億ドルで買収

2009：FlickrのエンジニアであるJohn AllspawとPaul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

oogle Cloud
サービス開始
(2008)

Azure
サービス開始
(2010)

Findy Team+
Team+
(2021/10)

Claude 1
(2023/3/14)

Cline
(2024/7)

GitHub Copilot
(2021/6/29)

Gemini 1
(2023/12/6)

GitHub
リリース
(2008)

GitLab
(2011)

GitHub Actions
(2018)

ChatGPT
一般公開
(2022/11/30)

Devin
(2024/3/12)

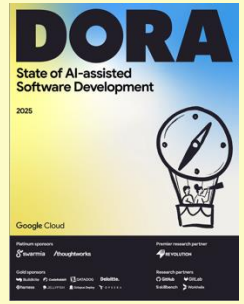
Jenkins 誕生
(2011)

DATADOG
(2018)

Agile Alliance + Project Management Institute

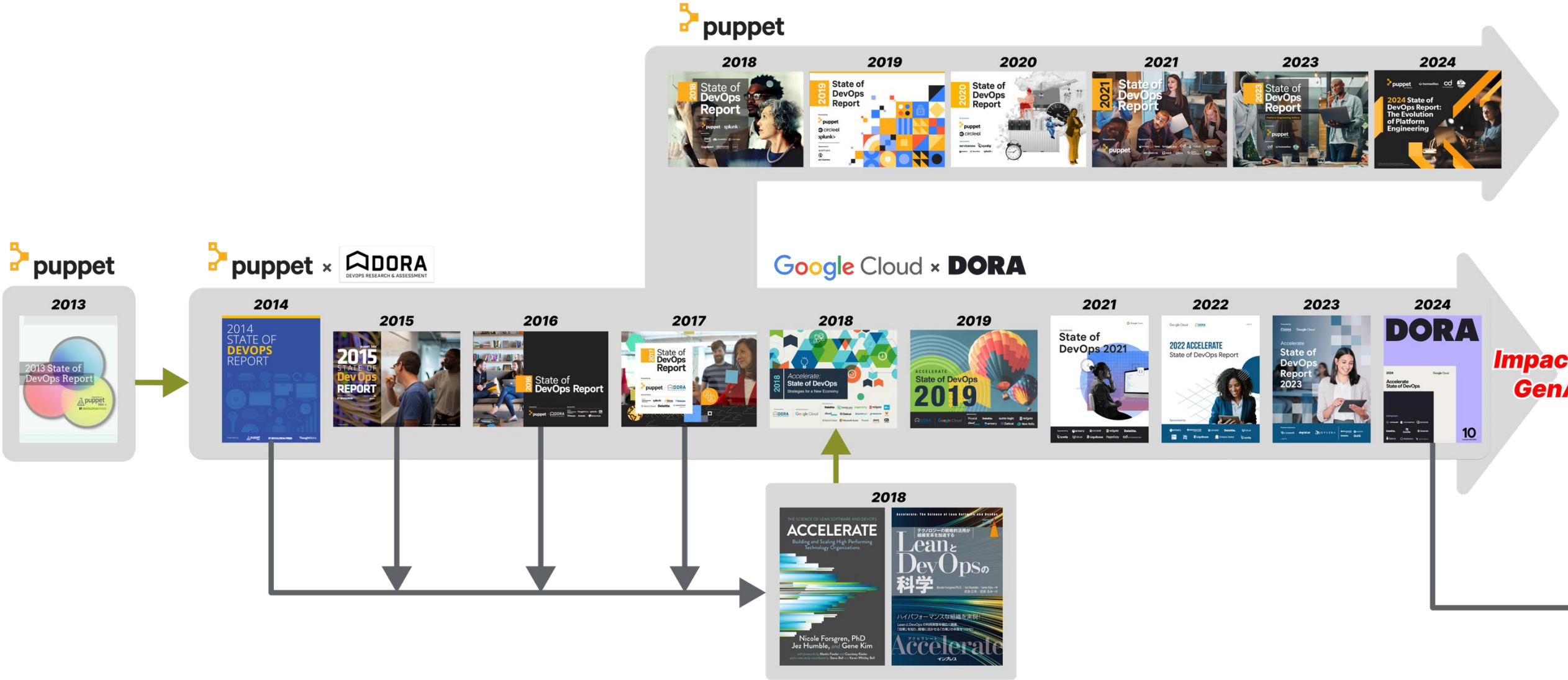
2009～2020：第二次ブラウザ戦争
Google Chromeの躍進、 Internet Explorerの衰退

2025/1/3：Agile AllianceがPMBOKなどを策定するPMIに加盟

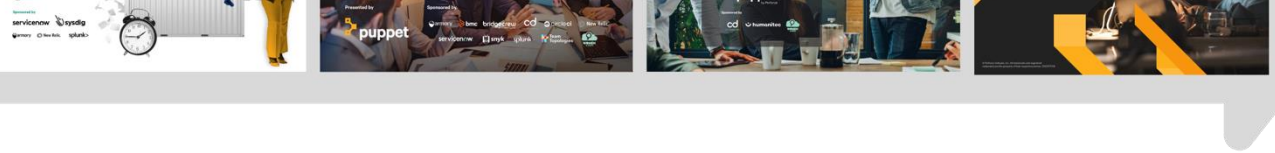


GitHub Copilot 誕生からすでに4年。

DORAアンケート回答者の 90% が、仕事で AI を使用していると回答。



Findy



DORA

2019

2021

2022

2023

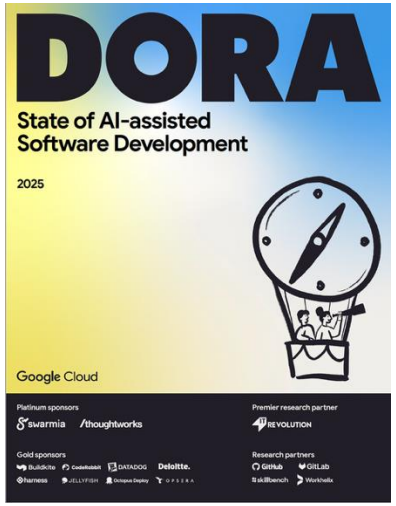
2024

Impact of GenAI

2025

2025

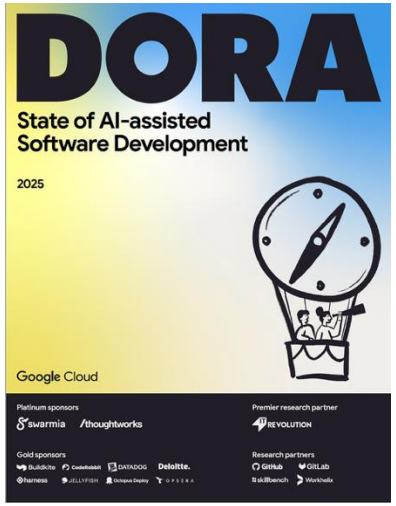
2025 DORA State of AI-assisted Software Development Report

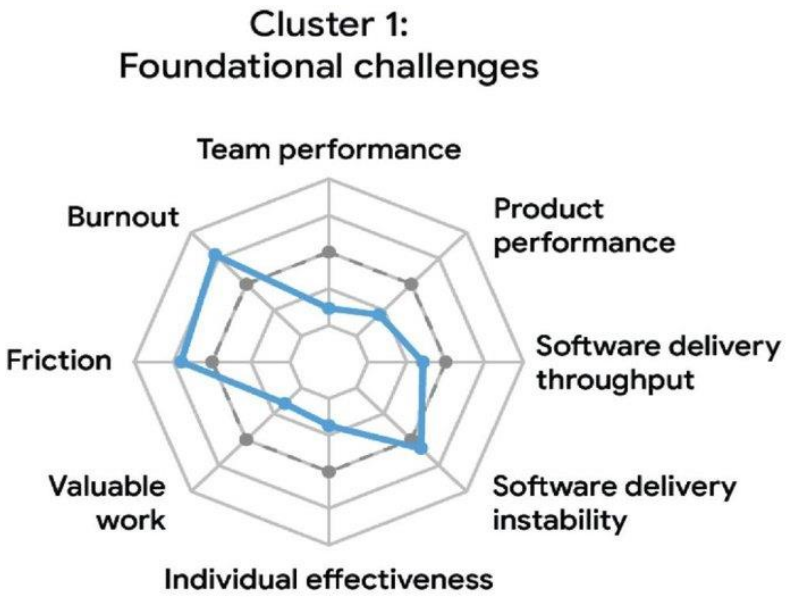


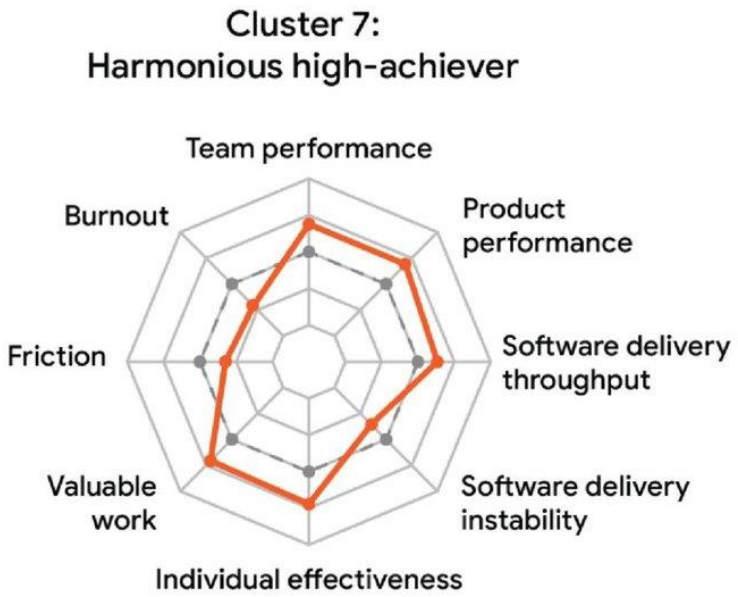
- **重要なポイント: AIは増幅器（amplifier）**
  - DORAの調査には、100時間を超える定性データと、世界中の約5,000人の技術専門家からのアンケート回答が含まれています。
  - ソフトウェア開発におけるAIの主な役割は、増幅器としての役割です。AIは、高パフォーマンス組織の強みと、苦戦している組織の機能不全を拡大します。
  - AI がソフトウェア開発を加速するものの、その加速によって下流で弱点が露呈する可能性があります。
  - 強力な自動テスト、成熟したバージョン管理プラクティス、迅速なフィードバック ループなどの堅牢な制御システムがない場合、変更量の増加は不安定につながります。
  - フィードバック ループが高速な疎結合アーキテクチャで作業するチームは成果を上げます。
  - 一方、密結合システムと遅いプロセスに制約されるチームはほとんど、またはまったくメリットがありません。

2025 DORA State of AI-assisted Software Development Report

- **クラスター分析によって、7つのチームのアーキタイプが判明**
  - **クラスター1「基本的な課題」グループは、**サバイバル モードに陥っており、プロセスと環境に大きなギャップがあるため、パフォーマンスが低く、システムの安定性（変化しにくさ）が高く、燃え尽き症候群や摩擦のレベルが高い状態。
  - **クラスター7「調和のとれた高パフォーマンス」は、**チームのウェルビーイング、プロダクトの成果、ソフトウェア デリバリーなど、複数の分野で優れた成果を上げている。







Findy

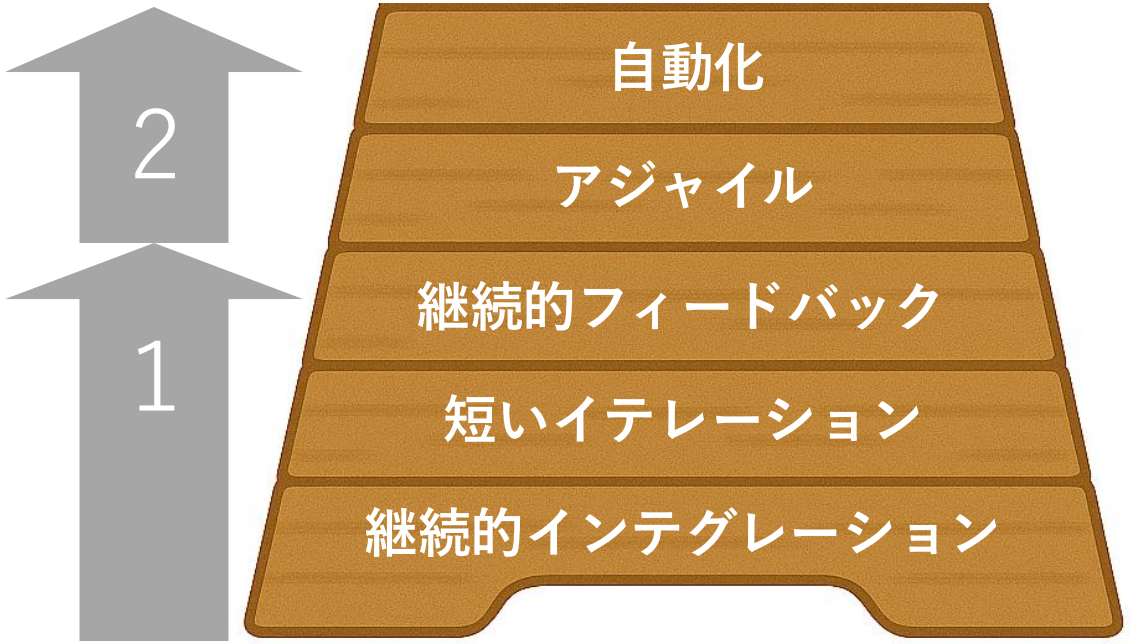
技術的発展の階層構造

技術的発展の階層構造

- **foundation 1: WFのリスク軽減手法**（1990年代～）
  - インクリメンタル、スパイラル、RUP
  - 継続的インテグレーション(CI)
  - 短いイテレーション、継続的フィードバック

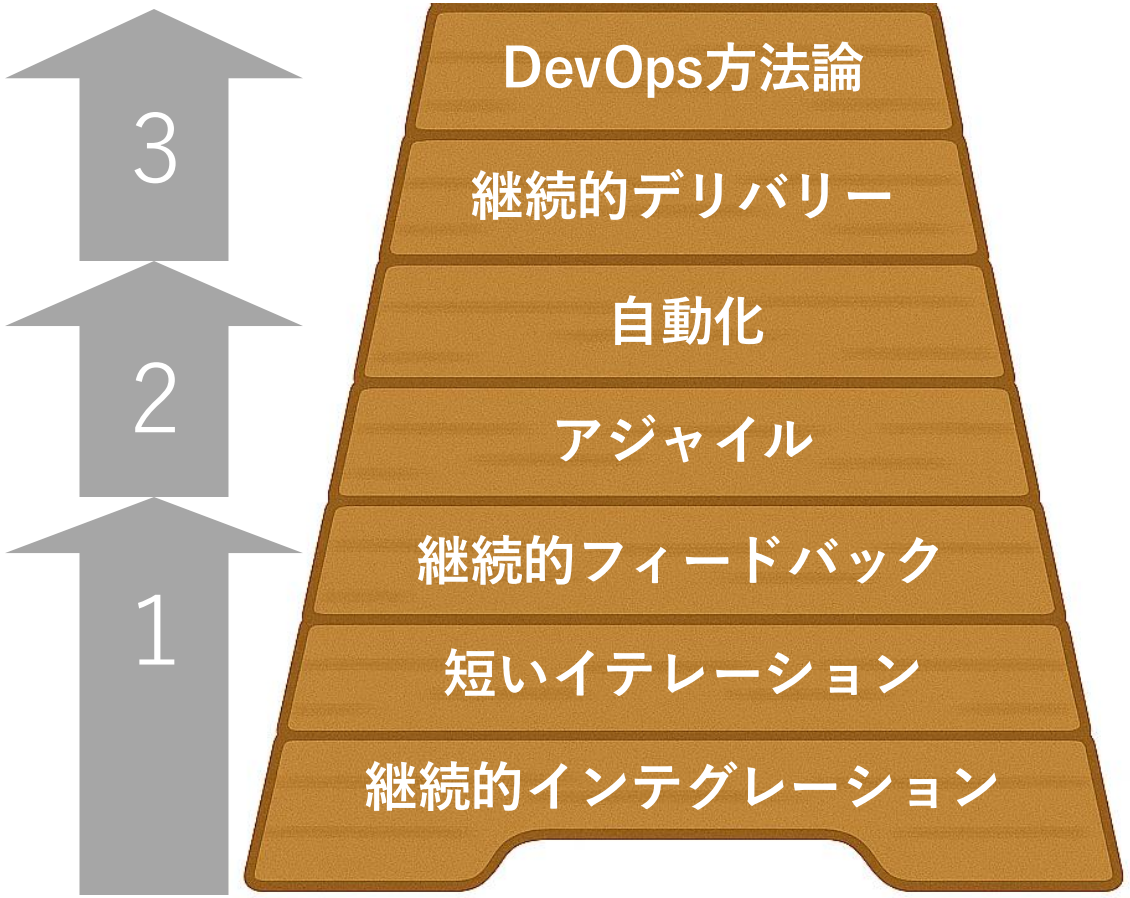
技術的発展の階層構造

- **foundation 2: アジャイル**（2000年代～）
  - クリスタル、Extreme Programming、Scrum
  - アジャイルソフトウェア開発宣言（Agile Manifesto）
- **foundation 1: WFのリスク軽減手法**（1990年代～）
  - インクリメンタル、スパイラル、RUP
  - 継続的インテグレーション(CI)
  - 短いイテレーション、継続的フィードバック



技術的発展の階層構造

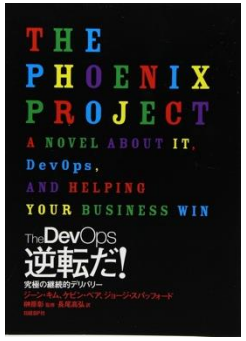
- **foundation 3: DevOps方法論**（2010年代～）
  - **アジャイルを土台に**、開発(Dev)と運用(Ops)の協力
  - 自動化を中心とした継続的デリバリー(CD)
  - フロー、フィードバック、**継続的学習と実験**
- **foundation 2: アジャイル**（2000年代～）
  - クリスタル、Extreme Programming、Scrum
  - アジャイルソフトウェア開発宣言（Agile Manifesto）
- **foundation 1: WFのリスク軽減手法**（1990年代～）
  - インクリメンタル、スパイラル、RUP
  - 継続的インテグレーション(CI)
  - 短いイテレーション、継続的フィードバック

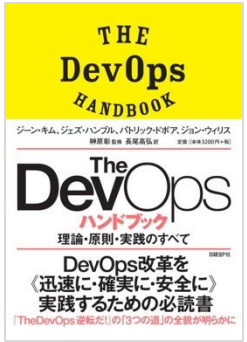


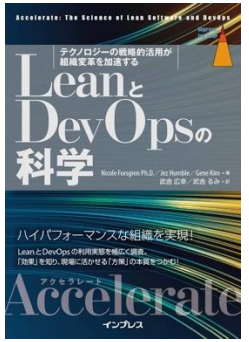
2010年代中盤：ジーン・キムとDevOpsの理論体系化

Gene Kim 氏（開発生産性Conference 2025）

2010年代中盤、DevOpsムーブメントの中心人物として台頭したのがジーン・キム（Gene Kim）です。
それまで現場レベルの実践知として広がっていたDevOpsの考え方を、組織全体で取り組むべき体系的な方法論として確立した功績は計り知れません。
彼の著作は技術書の枠を超え、多くの組織でDevOps変革の指南書となっています。

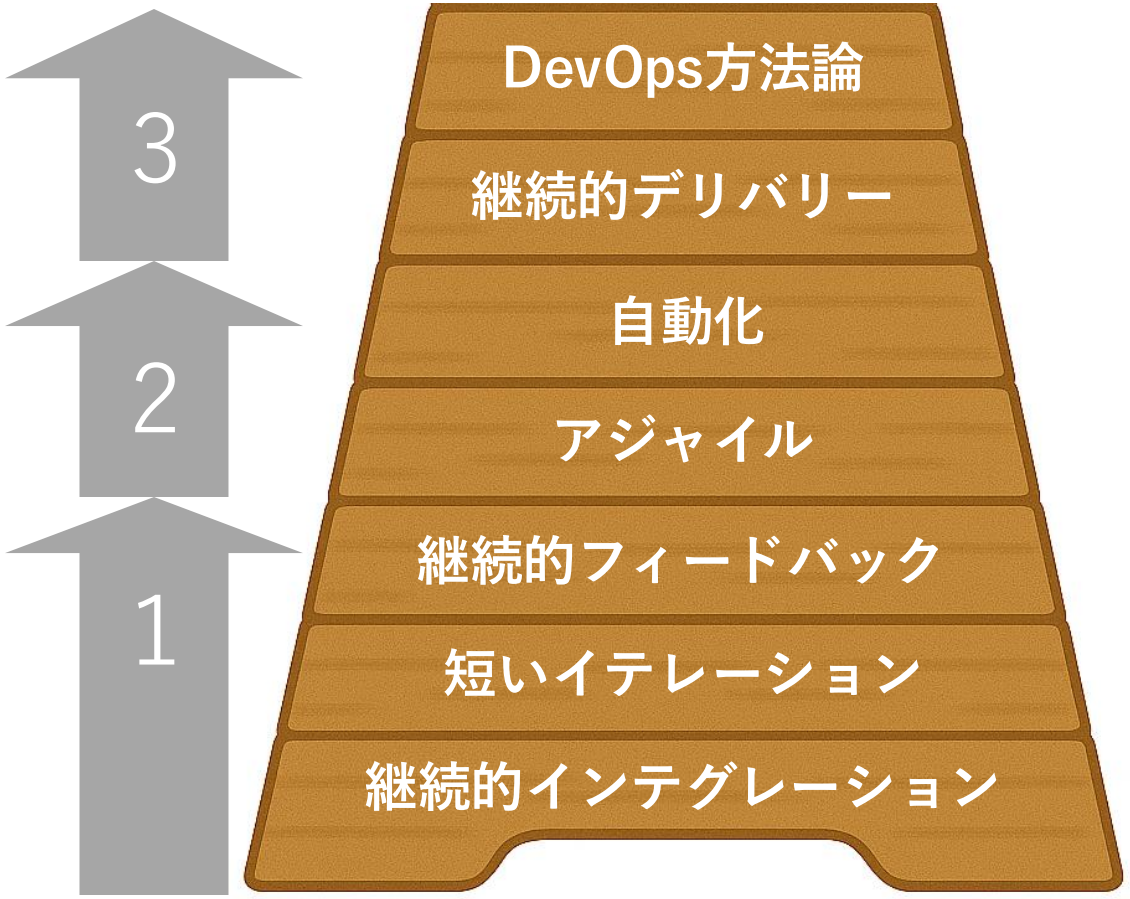






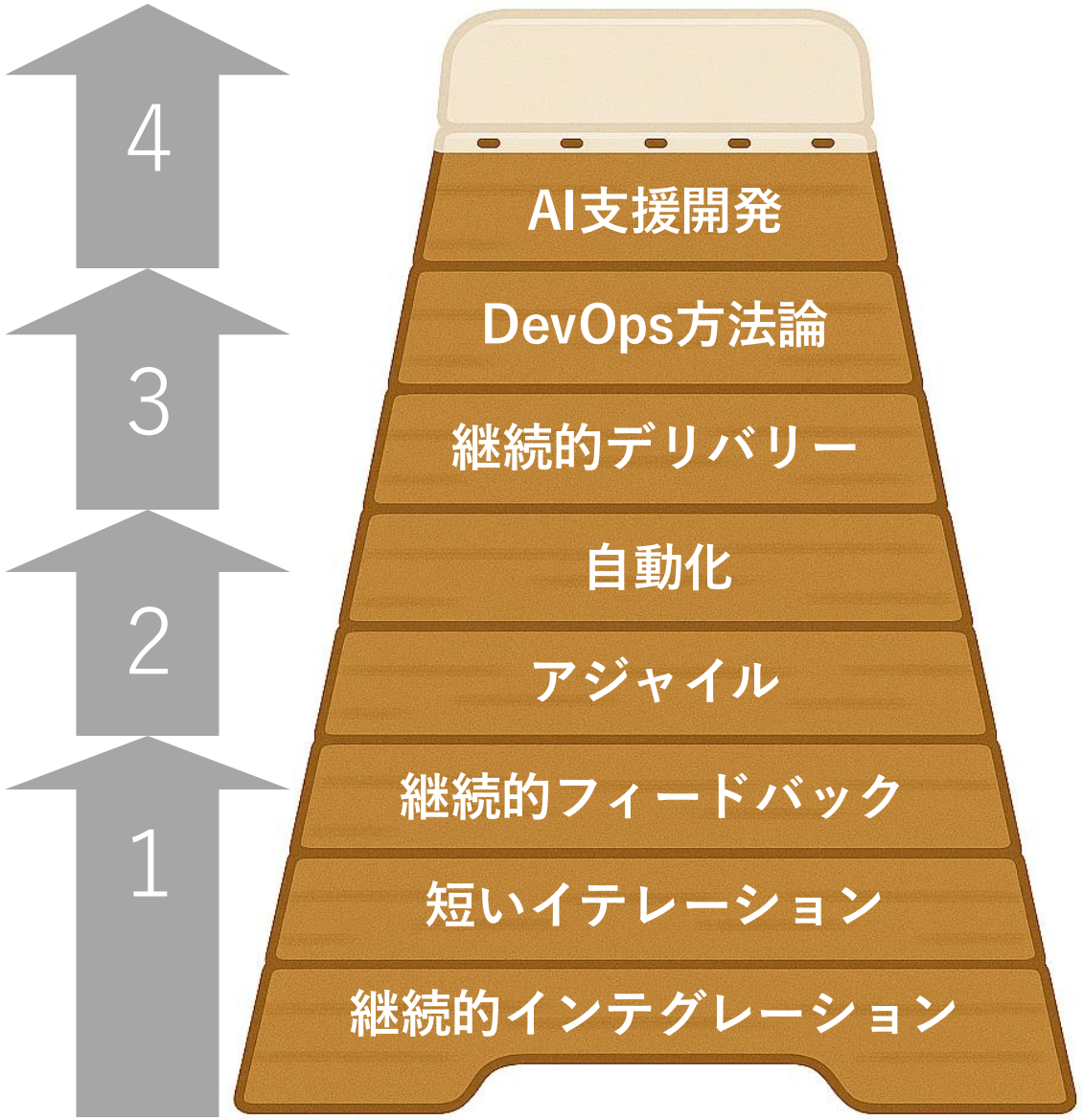
技術的発展の階層構造

- **foundation 3: DevOps方法論**（2010年代～）
  - **アジャイルを土台に**、開発(Dev)と運用(Ops)の協力
  - 自動化を中心とした継続的デリバリー(CD)
  - フロー、フィードバック、**継続的学習と実験**
- **foundation 2: アジャイル**（2000年代～）
  - クリスタル、Extreme Programming、Scrum
  - アジャイルソフトウェア開発宣言（Agile Manifesto）
- **foundation 1: WFのリスク軽減手法**（1990年代～）
  - インクリメンタル、スパイラル、RUP
  - 継続的インテグレーション(CI)
  - 短いイテレーション、継続的フィードバック



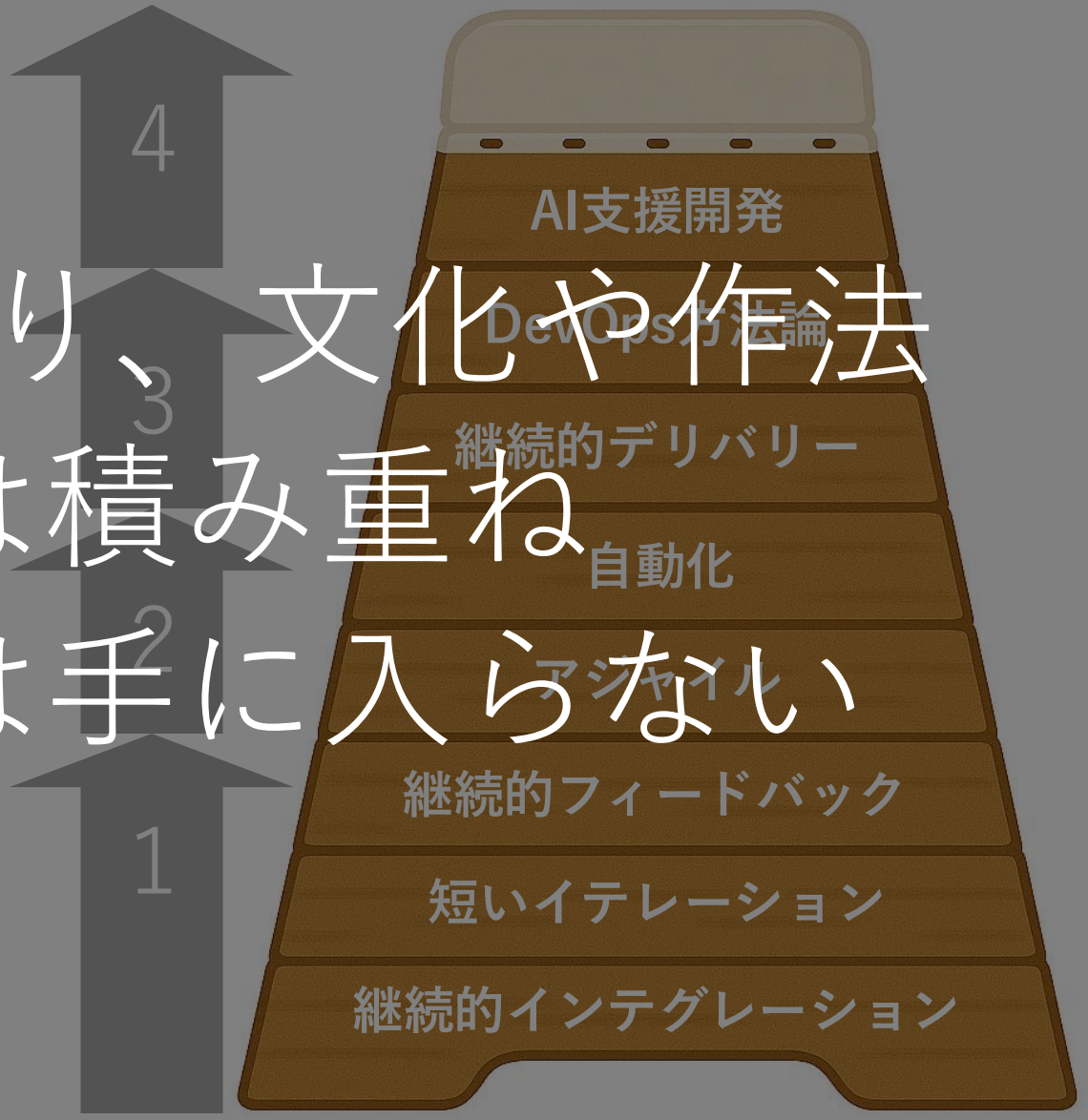
技術的発展の階層構造

- **foundation 4: AI支援開発**（2020年代〜）
  - 人間とAIの協働で、ソフトウェアを効率的に創る
  - DevOpsの土台があって初めて効果を発揮
- **foundation 3: DevOps方法論**（2010年代〜）
  - **アジャイルを土台に**、開発(Dev)と運用(Ops)の協力
  - 自動化を中心とした継続的デリバリー(CD)
  - フロー、フィードバック、**継続的学習と実験**
- **foundation 2: アジャイル**（2000年代〜）
  - クリスタル、Extreme Programming、Scrum
  - アジャイルソフトウェア開発宣言（Agile Manifesto）
- **foundation 1: WFのリスク軽減手法**（1990年代〜）
  - インクリメンタル、スパイラル、RUP
  - 継続的インテグレーション(CI)
  - 短いイテレーション、継続的フィードバック



技術的発展の階層構造

- **foundation 4: AI支援開発**（2020年代〜）
  - 人間とAIの協働で、ソフトウェアを効率的に創る
  - DevOpsの土台があって初めて効果を発揮
- **foundation 3: DevOps方法論**（2010年代〜）
  - **アジャイルを土台に**、開発(Dev)と運用(Ops)の協力
  - 自動化を中心とした継続的デリバリー(CD)
  - フロー、フィードバック、**継続的学習と実験**
- **foundation 2: アジャイル**（2000年代〜）
  - クリスタル、Extreme Programming、Scrum
  - アジャイルソフトウェア開発宣言（Agile Manifesto）
- **foundation 1: WFのリスク軽減手法**（1990年代〜）
  - インクリメンタル、スパイラル、RUP
  - 継続的インテグレーション(CI)
  - 短いイテレーション、継続的フィードバック



技術というより、文化や作法
これらは積み重ね
一足飛びには手に入らない

Findy

ソフトウェア関連の論文・文献

ICTの進化

西暦

主な出来事

Manufacturing Industry in Japan

The New New Product Development Game (1986)

SCRUM Development Process（1995）

XP Explained : Embrace Change（1999）

The Scrum Guide（2010～）

アジャイルソフトウェア開発宣言（2001）

Agile

DevOps（2009～）

GenAI（2021～）

50年前 40年前 30年前 20年前 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

メインフレーム時代：
IBM System 360とVT100（1970～1980）

PC/AT互換機が誕生（1981～）

PC-9800シリーズ（1982～2003）

Apple Macintosh（1984）

ワークステーションの時代：
Sun SPARCstation（1989～1994）

数字送信の開始によるポケベルブーム（日本）（1992～1996）

テレホーダイ（1995～2023）

初代iPhone発表（2007/1/9）

初代iPad発表（2010/4/3）

Samsung Galaxy S II（2011/5/2）

Netflixが日本に上陸(2015/9/2)
※では2007/1にサービスイン

ARMアーキテクチャのSoC
Apple M1 生産（2020）

MacBook Pro M4 Max（2024）

1974年：「TCP/IP」の最初の仕様がRFC 675として公開

1982年～：米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業…

1980年～：米国防総省(DoD)がウォーターフォールを採用

1980年後半～1990年前半：米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997～2…年代：米国防総省(DoD)がCMMレベル3以上を…衛関連のソフトウェア調達条件として要求し始める。

2010年～：米国議会は2010会計年度国防権限法(NDAA 2010)において…速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化…

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1970～1980年代：多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。
同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代：米国を始めとしてアジャイル手法が急速に広まり、ウォーターフォールモデルを採用し続ける。

1970年代後半～1980年代末：日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988～：日本を含む諸外国へ「通商法スーパー301条」発動

2000/3～：ITバブル崩壊

2008/9～：リーマンショック

2010年以降：…本のスタートアップはクラウドファーストで事業を立ち上げ、初めからアジャイル手法を採用

1979～：日本の製造業の高品質、ものづくりの強さを研究。

1991～：バブル崩壊「失われた○○年」

2009：Flickrのエンジニアである John Allspaw と Paul Hammond が初めて Dev…

2020～2023/5：…19

2023～：AI（GenAI）が前提の時代へ

2018：マイク…ソフ…GitHub を 75 億ドルで買収

1980年代後半～1990年代前半：UNIX戦争

1985～1990：国家プロジェクト「Σ計画」が頓挫

© The Deming Institute

“トリニトロン”
日本製品が欧米で人気

"ウォークマン"
1号機 TPS-L2
(1979)

Toyota Corolla
Liftback SR5 001
(1980～)

世界初のポータブル
CDプレーヤー
D-50（1984）

Java v1.0 正式リリース
Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた
(1996)

Linux 0.01 リリース
UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた
(1991/9/17)

Windows95 リリース
コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに
(1995)

Bugzilla
リリース
(2000)

Jira
リリース
(2002)

Subversionリリース
(2000)

CVS 誕生
(1990)

Selenium
(2004)

REDMINE
(2006)

trac
(2004)

git
(2005)

AUTOMOTIVE SPICE®
欧州の自動車メーカが中心となって公開（2005）

AWS サービス開始 (2006)

Google Cloud サービス開始 (2008)

Azure サービス開始 (2010)

GitHub リリース (2008)

GitHub Actions (2018)

Jenkins 誕生 (2011)

DATADOG (2019)

Findy Team+ (2021/10)

Claude 1 (2023/3/14)

Cline (2024/7)

Gemini 1 (2023/12/6)

ChatGPT 一般公開 (2022/11/30)

Devin (2024/3/12)

1990～2000：第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

2009～2020：第一次ブラウザ戦争 Google Chromeの躍進…

2025/1/3：…Agile Alliance…PMBOKなどを策定するPMIに加盟

ソフトウェア関連の論文・文献

ICTの進化

西暦

主な出来事

Waterfall（1976〜）

1970 1976 1981 1987 1989/02/01 1994/10/31 1996/8 2001/5/18 2003/9/1 2005/10 2009/11/27 2011/7/16 2014/8/18 2017/6/22 2019/9/17 2021/3/15 2023/11/21 2025/6/25

1977 1979 1984 1988/3/1 1995/10/1 2000/12/1 2003/9/1 2005/10/7 2008/12/30 2010/10/12 2013/1/10 2016/10/6 2018/4/1 2019/9/30 2021/12/1 2023/11/28 2024/12/25

1978/5/1 1986 1989 1995 1999 2002 2004/11/16 2010/7/27 2012/3/15 2013/8/1 2018/3/27 2021/8/1 2022/12/17 2024/7/11

1975 1978 1979 1982 1990/10/10 1992/3/13 1996/4/1 2002/11/8 2004/7/16 2006/10/19 2010〜 2011/10/18 2014/4/22 2017/10/14 2018/11/22 2021/12/10 2024/9/13

パソコン通信 | 携帯電話・PHSの普及、インターネット黎明期 | 携帯電話の多機能化、ブロードバンド時代 | スマートフォン普及時代（2010年、モバイル端末利用率がパソコン利用率を超える）

メインフレーム時代：IBM System 360とVT100（1970〜1980）

PC/AT互換機が誕生（1981〜）

PC-9800シリーズ（1982〜2003）

Apple Macintosh（1984〜）

ワークステーションの時代：Sun SPARCstation（1989〜1994）

数字送信の開始によるポケベルブーム（日本）（1992〜1996）

テレホーダイ（1995〜2023）

初代iMac（1998〜）

F501i HYPERiモード開始（1999/2/22）

家庭向けADSL・FTTHの普及、ブロードバンド時代に突入（2001〜）

初代iPhone発表（2007/1/9）

初代iPad発表（2010/4/3）

Samsung Galaxy S II（2011/5/2）

Netflixが日本に上陸(2015/9/2)※🇺🇸では2007/1にサービスイン

ARMアーキテクチャのSoC Apple M1 生産（2020）

MacBook Pro M4 Max（2024）

50年前 | 40年前 | 30年前 | 20年前 | 10年前

1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018 2020 2022 2024 2026

1974年：🇺🇸「TCP/IP」の最初の仕様がRFC 675として公開

1982年〜：🇺🇸米国防総省(DoD)は全ての軍用コンピュータ網のためにTCP/IP標準を作成。その後、コンピューター産業に開放され、4.2 BSD UNIXを皮切りにほとんどの商用OSにTCP/IPが実装された。

★ピアソンショック（2013年8月1日）

Point: 米国防総省(DoD)が与えた影響

1980年〜：🇺🇸米国防総省(DoD)がウォーターフォールを採用

1980年後半〜1990年前半：🇺🇸米国防総省(DoD)の発注したソフトウェアに問題が多発。米会計局でも多くの遅延／途中での挫折が発生

1997〜2010年代：🇺🇸米国防総省(DoD)がCMMレベル3以上を防衛関連のソフトウェア調達条件として要求し始める。

2010年〜：🇺🇸米国議会は2010会計年度国防権限法(NDAA 2010)において「迅速なITシステム獲得のための新たな取得プロセスの実施」(第804条)をDoDに命じ、事実上アジャイル型開発の原則を法制化

1970〜1980年代：🌎多くの企業や政府機関が「ウォーターフォールモデル」を公式な開発プロセスとして採用。同時に、Royceの思惑と違う、誤解された「ウォーターフォール」が広まる

1990年代：🇺🇸ウォーターフォールのリスクを軽減する開発手法が次々と誕生（インクリメンタル、スパイラル、RUP など）

2000年代〜：🇺🇸米国を中心にでアジャイル手法が急速に広まり、ウォーターフォールは特にWeb系・スタートアップ企業ではほぼ使われなくなる。

2000年代〜：🇯🇵日本は大企業を中心に、ウォータフォールモデルを採用し続ける。

2020〜2023/5：🌎COVID-19

1970年代後半〜1980年代末：🇯🇵日本車と家電がアメリカ市場を席巻。これが、徐々に日米貿易摩擦の火種に……

1988〜：🇺🇸日本を含む諸外国へ「通商法スーパー301条」発動

2000/3〜：🇺🇸ITバブル崩壊

2010年以降〜：🇯🇵日本のスタートアップはクラウドファーストで事業を立ち上げ、初期からアジャイル手法を採用

1979〜：🇺🇸日本の製造業の高品質、ものづくりの強さを研究。

1991〜：🇯🇵バブル崩壊「失われた○○年」

2001/2/11：🌎アジャイルソフトウェア開発宣言

2000年代後半〜：🌎クラウドファースト・クラウドネイティブ時代に突入

2021〜：🌎AI（GenAI）が前提の時代へ

2008/9〜：🇺🇸リーマンショック

2018：🇺🇸マイクロソフト、GitHub を 75 億ドルで買収

1980年代後半〜1990年代前半：🌎UNIX戦争

1985〜1990：🇯🇵国家プロジェクト「Σ計画」が頓挫

2009：🇺🇸Flickrのエンジニアである John AllspawとPaul Hammondが初めてDevOpsに繋がる伝説的な講演を行う

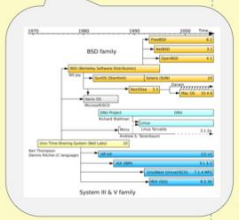

“トリニトロン”日本製品が欧米で人気


"ウォークマン"1号機 TPS-L2（1979）


Toyota Corolla Liftback SR5 001（1980〜）


世界初のポータブルCDプレーヤー D-50（1984）

© The Deming Institute



Java v1.0 正式リリース Javaはオブジェクト指向と仮想マシン技術を普及させ、後の言語設計にも影響を与えた（1996）

Linux 0.01 リリース UNIXが商業化・断片化していく中、Linuxはオープンソースの力で統一的な開発基盤となり、世界中の技術革新を支えた（1991/9/17）

Windows95 リリース コンシューマ向けOSにTCP/IPが標準搭載されワークステーション並みに（1995）

Bugzilla リリース（2000）

Jira リリース（2002）

Subversionリリース（2000）

CVS 誕生（1990）

flickr リリース（2004/2）

AWS サービス開始（2006）

Google Cloud サービス開始（2008）

Azure サービス開始（2010）

Selenium（2004）

REDMINE（2006）

trac（2004）

git（2005）

GitHub リリース（2008）

AUTOMOTIVE SPICE®

欧州の自動車メーカが中心となって公開（2005）

GitLab（2011）

Jenkins 誕生（2011）

GitHub Actions（2018）

DATADOG（2018）

GenAI

Findy Team+（2021/10）

GitHub Copilot（2021/6/29）

Gemini 1（2023/12/6）

ChatGPT 一般公開（2022/11/30）

Devin（2024/3/12）

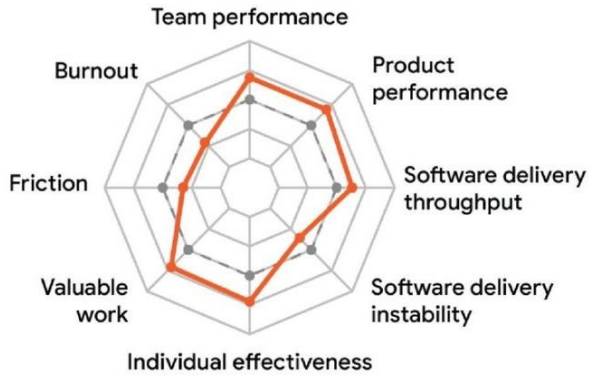
Agile Alliance + Project Management Institute.

2025/2〜：🌎Agile AllianceがPMBOKなどを策定するPMIに加盟

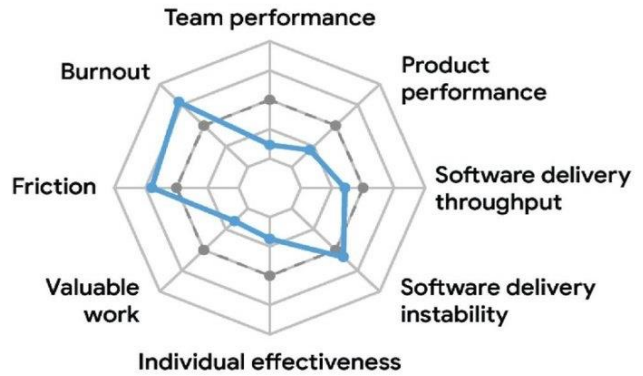
1990〜2000：🌎第一次ブラウザ戦争。Internet Explorerの躍進、Netscape Navigatorの消滅

2009〜2020：🌎第二次ブラウザ戦争 Google Chromeの躍進、 Internet Explorerの衰退

Cluster 7:
Harmonious high-achiever



Cluster 1:
Foundational challenges



Findy

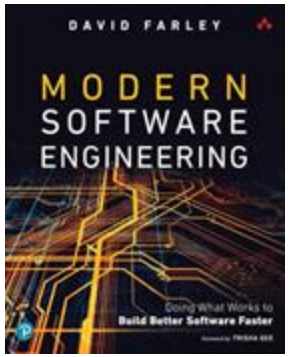
AIによるソフトウェア開発は、
皆さんの組織で
何を「増幅」させていますか。

Modern Software Engineering

間違った考え方をなかなか捨てられない理由のひとつは、ソフトウェア開発のパフォーマンス（能力、業績）を効果的に計測できていないことにあります

David Farley（2021）

David Farley



引用：Farley, D.(2022), 長尾高弘 (訳). 継続的デリバリーのソフトウェア工学: もっと早く、もっと良いソフトウェアを作るための秘訣 (Nagao, T., Trans.). 日経BP社. (Original work published 2021) p.70

測定は必要。だが、制御のためではない

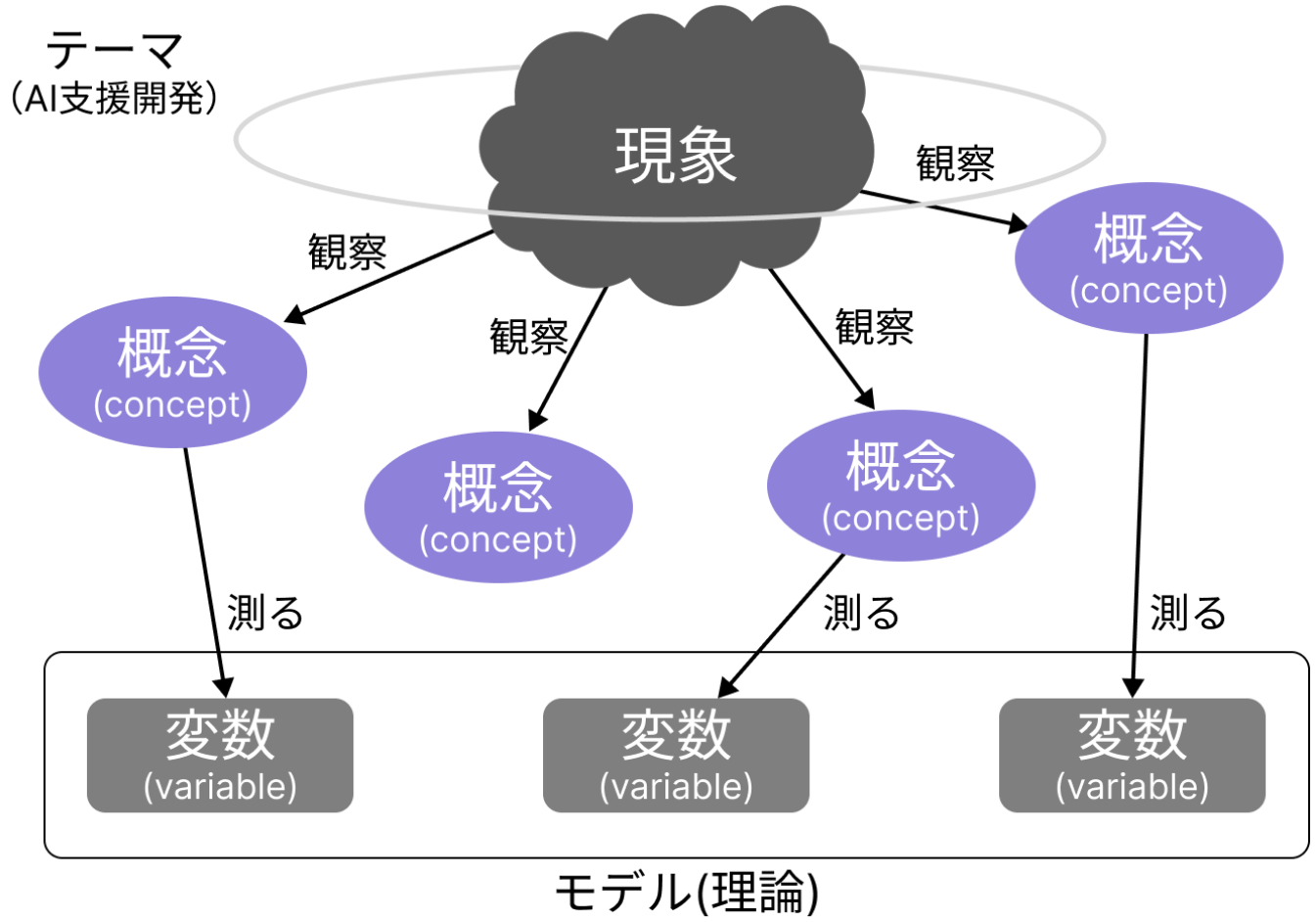
“私が指標に懐疑的なことを言うと誰かがこう返しました。
「あなたはただ何も測りたくないだけでしょう」
それは100万％違います。
私は自分のソフトウェア開発プロセスを測定しています。
開発を始めてからずっとです。
そして非常に価値があると思っています。
自分がやっていることを数値化して分析して解釈できるのですから。”

Kent Beck

Kent Beck （開発生産性Conference 2025）

重要なのは測定に基づく「科学的リサーチ」

- 科学的リサーチとは、現象を体系的に調査し、新しい発見や理論を導き出す手法です。観察から始まり、概念化、**測定可能な変数**への置き換え、モデル化という段階を経て、客観的なデータに基づいた結論を導きます。



向後, 千春. (2016). 18歳からの「大人の学び」基礎講座: 学ぶ, 書く, リサーチする, 生きる. 北大路書房. 図3-3に筆者が加筆

【2024 DORA Report】 4つのソフトウェアデリバリーのパフォーマンスレベル

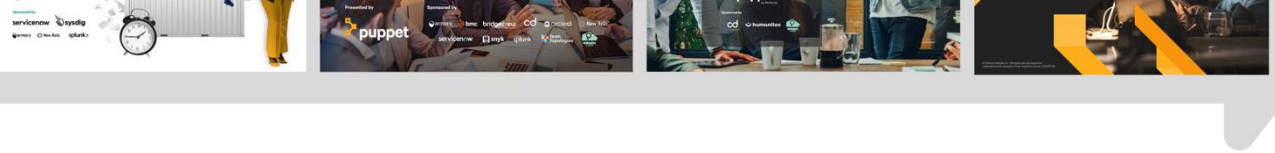
DevOpsによる
ソフトウェア開発の現状

- **ソフトウェアデリバリーパフォーマンス**
  1. **ソフトウェアデリバリースループット**
     (Software delivery throughput)
  2. **ソフトウェアデリバリーの不安定性**
     (Software delivery instability)

1. **ソフトウェアデリバリースループット**
   a. **変更のリードタイム**
      (Change lead time)
   b. **デプロイ頻度**
      (Deployment frequency)
   c. **失敗したデプロイメントの復旧時間**(Failed deployment recovery time)
2. **ソフトウェアデリバリーの不安定性**
   a. **変更失敗率**
      (Change failure rate)
   b. **リワーク率**
      (Rework rate)

パフォーマンスレベル	変更のリードタイム	デプロイの頻度	変更時の障害率	デプロイ失敗時の復旧までの時間	回答者の割合*
エリート	1日未満	必要に応じて(1日に複数回のデプロイ)	5%	1時間未満	19% (18〜20%)
高	1日から1週間の間	1日1回から週1回の間	20%	1日未満	22% (21〜23%)
中	1週間から1か月の間	週1回から月1回の間	10%	1日未満	35% (33〜36%)
低	1か月から6か月の間	月1回から6か月に1回の間	40%	1週間から1か月の間	25% (23〜26%)

* 89% 不確実性区間



DORA

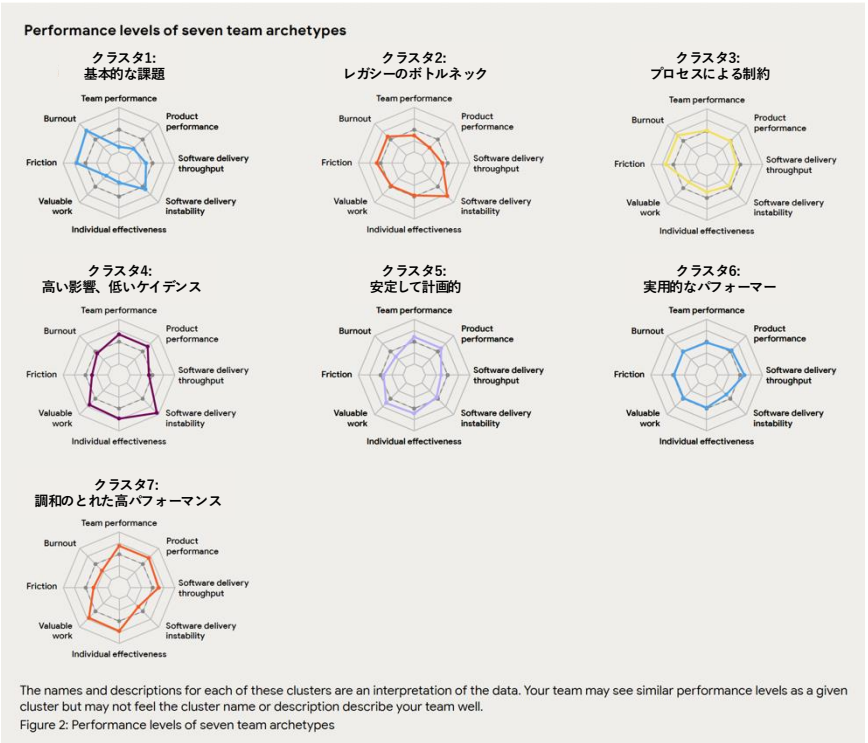
Impact of GenAI

【2025 DORA Report】7つのチームアーキタイプのパフォーマンスレベル

AI支援による
ソフトウェア開発の現状

- **ソフトウェアデリバリーパフォーマンス**
  1. **ソフトウェアデリバリースループット**
     (Software delivery throughput)
  2. **ソフトウェアデリバリーの不安定性**
     (Software delivery instability)
- **AIの導入**（AI adoption）
- **プロセスと実践**
  （Processes and practices）
- **個人の特性**（Individual traits）
- **環境および組織の特性**
  （Environmental and organizational traits）
- **サービスの特性**（Service traits）
- **組織のパフォーマンス**
  （Organizational performance）
- **チームのパフォーマンス**
  （Team performance）
- **プロダクトのパフォーマンス**
  （Product performance）
- **個人の成果**（Individual outcomes）

1. **ソフトウェアデリバリースループット**
   （Software delivery throughput）
   a. **変更のリードタイム**
      (Change lead time)
   b. **デプロイ頻度**
      (Deployment frequency)
   c. **失敗したデプロイメントの復旧時間**
      (Failed deployment recovery time)
2. **ソフトウェアデリバリーの不安定性**
   （Software delivery instability）
   a. **変更失敗率** (Change failure rate)
   b. **リワーク率** (Rework rate)
3. **チームパフォーマンス**
   （Team performance）
4. **プロダクトパフォーマンス**
   （Product performance）
5. **個人の有効性**
   （Individual effectiveness）
6. **価値ある仕事**（Valuable work）
7. **摩擦**（Friction）
8. **燃え尽き症候群**（Burnout）



The names and descriptions for each of these clusters are an interpretation of the data. Your team may see similar performance levels as a given cluster but may not feel the cluster name or description describe your team well.
Figure 2: Performance levels of seven team archetypes

Findy

おわりに

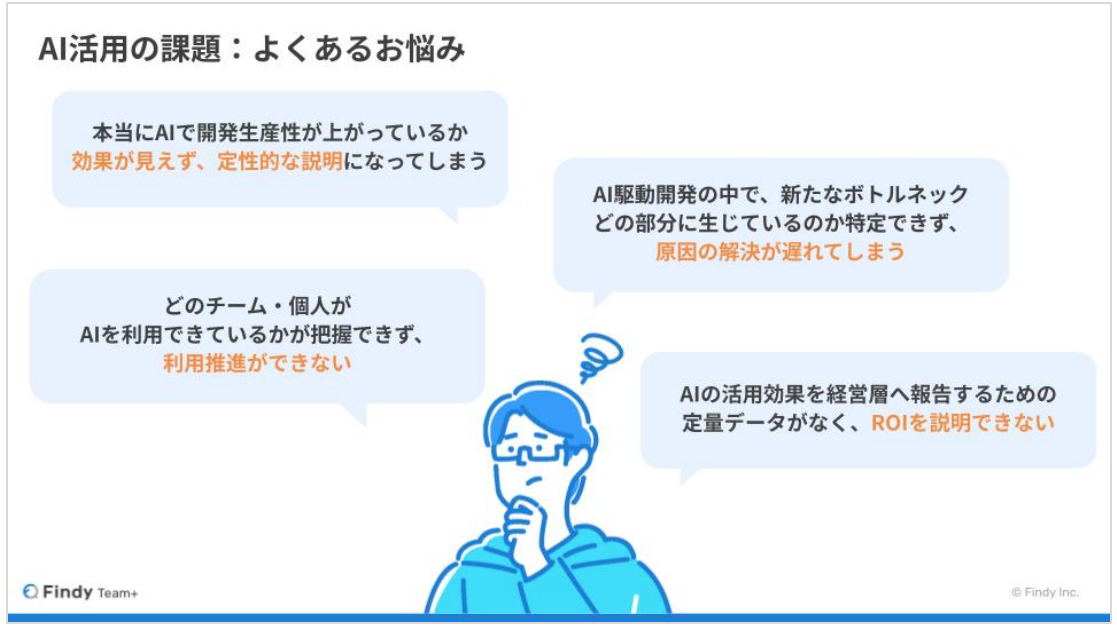
失われた土台を取り戻すために

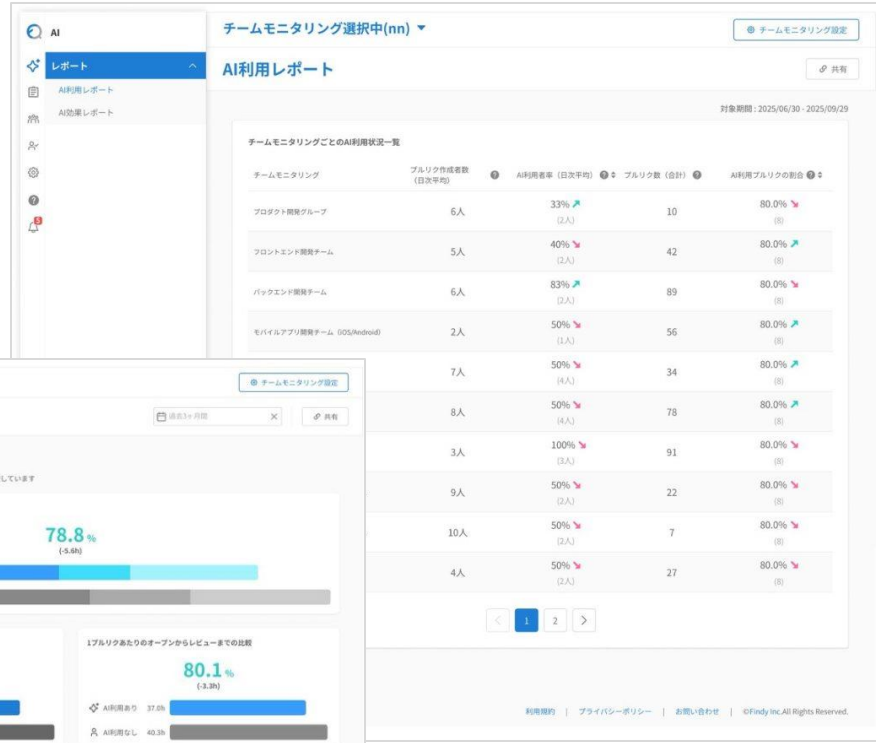
- 生成AIは、あらゆるコードを一瞬で生み出すようになりました。
  しかし、それが「動く」ことと「正しく動く」ことは、同じではありません。
- AIは最適解を提示しますが、**意図を理解しているわけではない。**
  だからこそ、私たちは問い直す必要があります。
  このコードは、何を前提に動いているのか。
  この結果は、再現性があるのか。
  この品質を、誰が説明できるのか。
- 品質保証の使命は変わりません。
  不確実な生成物に、確かさを与えること。
  そのために私たちは、設計を理解し、リスクを見抜き、根拠を示す。
- 技術が自動で進化しても、品質は自動では生まれません。
  “考えて確かめる”という人の営みこそ、これからの時代の土台です。

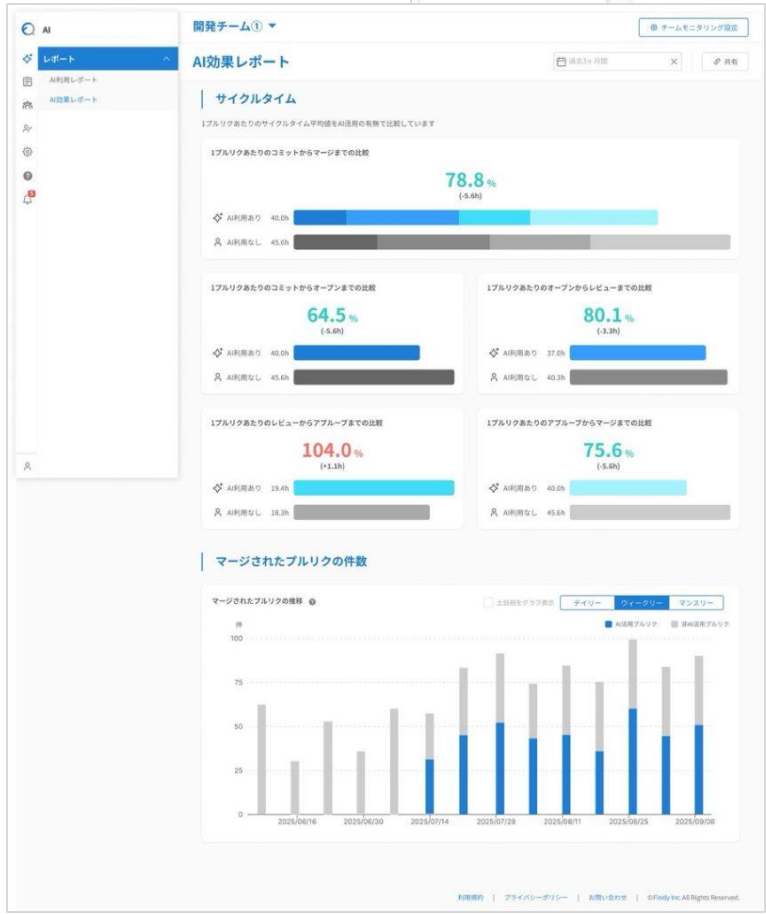
Findy

お知らせ

ブースに来てね







Findy

告知

connpass

TECH PLAY

Findy