エンジニアとして
この先生きのこるために

和田 卓人

Sep 10, 2020 @NTTコミュニケーションズ

和田卓人

Hatena

t-wada

t_wada

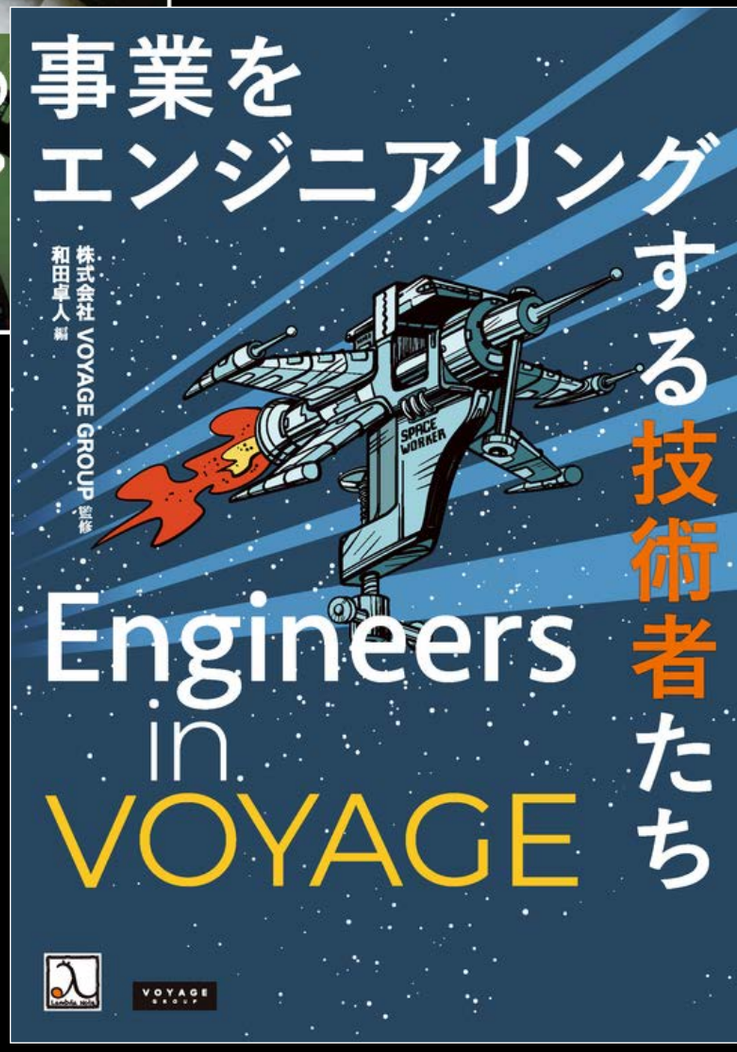
twada

手掛けた本たち


97
Collective Wisdom from the Experts
プログラマが
知るべき97のこと
97 Things Every Programmer Should Know
O'REILLY
オライリー・ジャパン


Test-Driven Development By Example
テスト駆動開発
Kent Beck 著
和田卓人 訳
テスト駆動開発（TDD）
原点から
本書は、Kent Beck, Test-Driven Development by Example
(Addison-Wesley, 2003)の新訳版です。


Avoiding the Pitfalls of Database Programming
SQL
アンチパターン
Bill Karwin 著
和田 卓人
和田 省二 監訳
児島 修 訳
O'REILLY
オライリー・ジャパン


事業を
エンジニアリングする技術者たち
株式会社 VOYAGE GROUP 監修
和田卓人 編
SPACE WORKER
Engineers
in
VOYAGE
VOYAGE GROUP

キャリア

- 大学在学中から設計とプログラミングのアルバイトを始める
- 卒業後プログラマとしてのキャリアを開始
- 電子政府のサブプロジェクト(数千人規模)でリードプログラマ
- XP のコーチとして 4 人のアジャイルチームに参加
- 講演、執筆、 OSS 活動を始める
- 現在は技術顧問業を行っている

よろしくお願いします

ソフトウエア開発力のさらなる強化へ
及川卓也氏・吉羽龍太郎氏・和田卓人氏がNTT Comの社外技術顧問に就任

カテゴリ Business/Technology

イノベーション　キャリアアップ　社内環境

97

プログラマが知るべき97のこと

97 Things Every Programmer Should Know

Kevlin Henney 編
和田 卓人 監修
夏目 大 訳

O'REILLY®
オライリー・ジャパン

The Pragmatic Programmer: From Journeyman to Master

新装版 達人プログラマー

職人から名匠への道

Andrew Hunt・David Thomas 共著
村上雅章 訳

The Pragmatic Programmer

Ohmsha

Takuto Wada
@t_wada

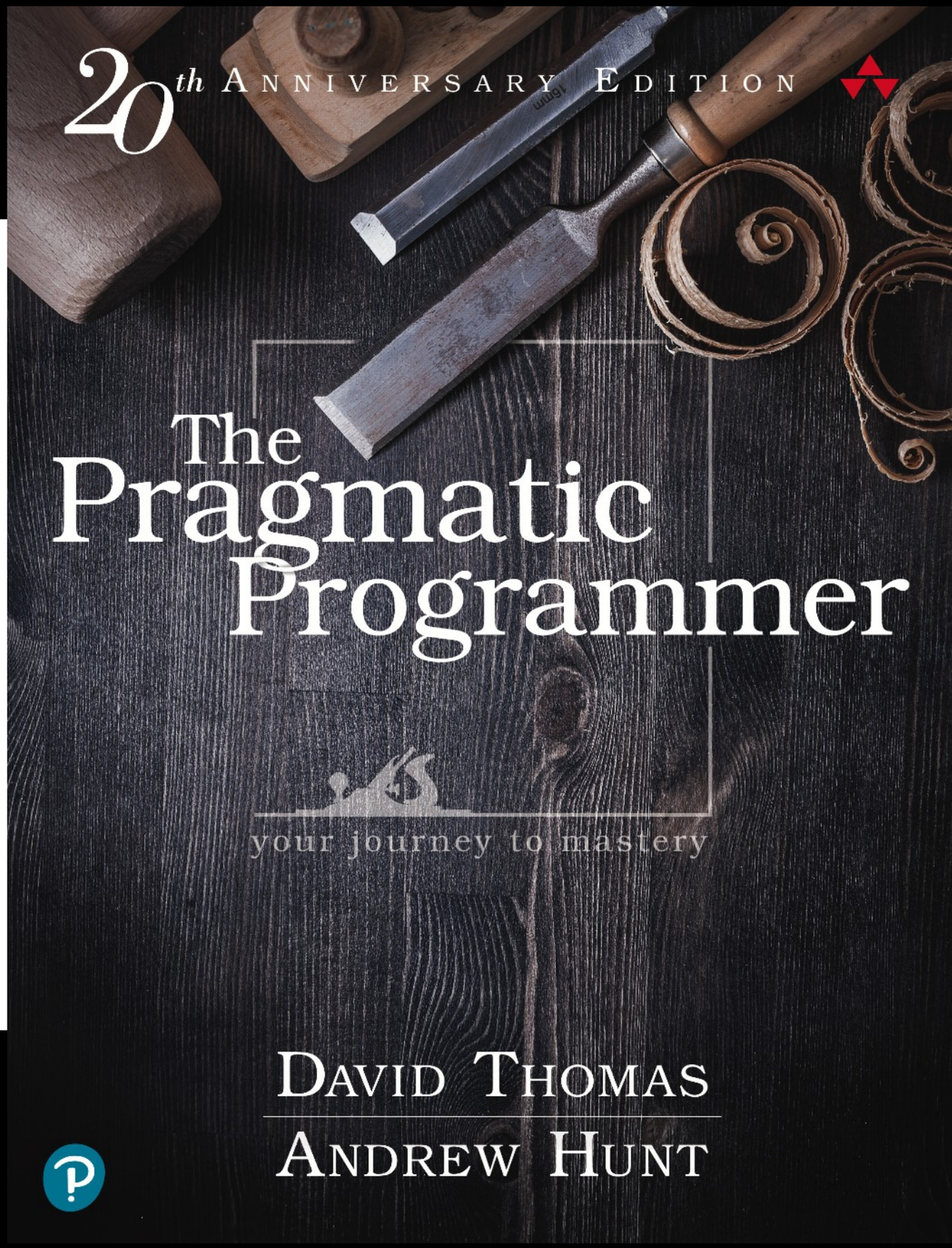
多くのプログラマの人生に影響を与えた『The Pragmatic Programmer』（邦訳『達人プログラマー』）が20年ぶりに改訂。1/3が新規追加の内容。既存部分もほとんどリライト。5月8日からbeta bookがPragmatic Bookshefで購入可 / “Coming Soon: The Pragmatic Programmer, 20th…” htn.to/XXURy7hXYF

Translate Tweet

10:04 AM - 7 May 2019

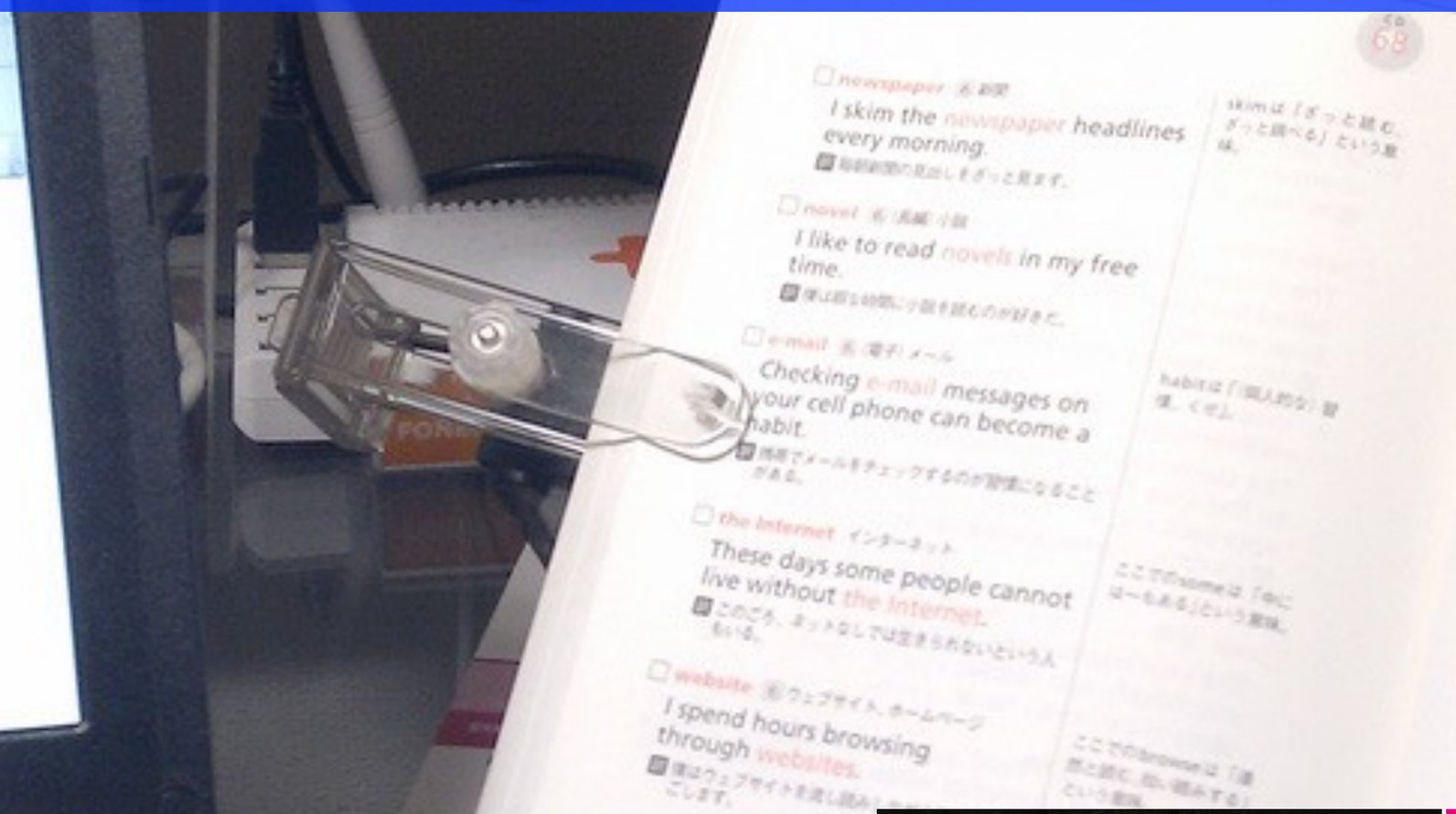
292 Retweets **621** Likes

1 292 621



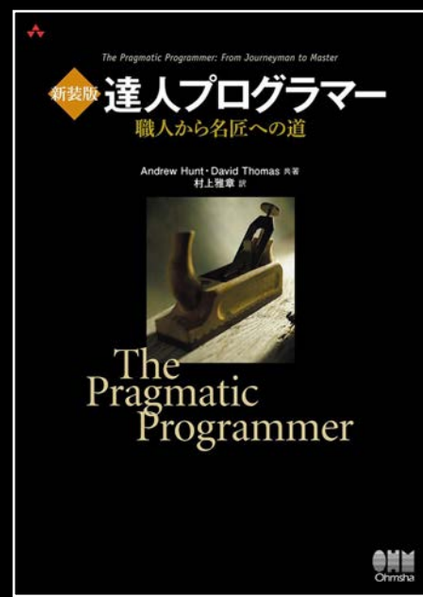
学び続ける姿勢


X200s
Insert
Home
PgUp
Delete
End
PgDn
Back space
Enter
Shift


68
newspaper 名 新聞
I skim the newspaper headlines every morning.
novel 名 長編小説
I like to read novels in my free time.
e-mail 名 電子メール
Checking e-mail messages on your cell phone can become a habit.
the Internet インターネット
These days some people cannot live without the Internet.
website 名 ウェブサイト、ホームページ
I spend hours browsing through websites.
Dialogue
Tom: Hi! I'm back!
Lisa : Oh, hi! I just came home, t
Tom: Did you check the answerin
Lisa : Yeah, but there were no ne
Tom: Turn on the TV, will you? I
weather forecast for tomorro
Lisa : There are no news programs
don't we watch the video we
until then?
Tom: Yeah, but I will check my e-ma
tomorrow's weather on the int
Lisa : OK. I'll watch TV for a while.
finished.


97
Collective Wisdom from the Experts
プログラマが知るべき97のこと
97 Things Every Programmer Should Know
O'REILLY
オライリー・ジャパン
Kevlin Henney 編
和田 卓人 監修
夏目 大 訳

“常にあなたの知識ポートフォリオに投資すること”



技術を学ぶのではなく、技術の学び方を学ぶ

Agenda

- 👉 学び方を学ぶ
- [来週] 現役技術者でいるために

👉 四半期毎に技術書を読む

手を動かして学ぶ

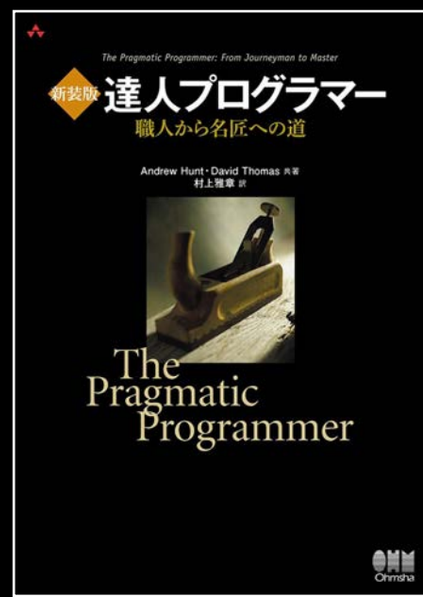
毎年少なくとも1つの言語を学習する

身の回りをプログラミング対象にする

アウトプットを行う

1. “四半期毎に

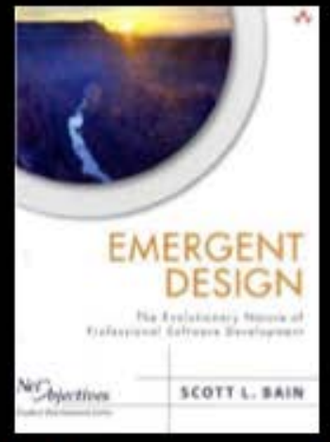
技術書を読む”




UNIXネットワークプログラミング
第2版 Vol.1
ネットワークAPI:ソケットとXTI
W.リチャード・スティーヴンス著 篠田陽一訳
UNIX NETWORK PROGRAMMING
Networking APIs : Sockets and XTI
Volume1, Second Edition


パターン、Wiki、XP
時を超えた創造の原則
デザインパターン
XP(エクストリームプログラミング)
Wiki

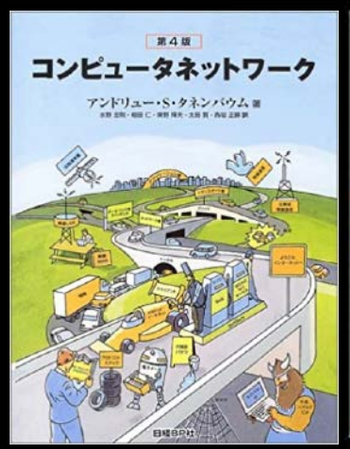

テスト駆動開発入門
TDD:TEST-DRIVEN DEVELOPMENT BY EXAMPLE
KENT BECK
ケント・ベック 著


EMERGENT DESIGN
The Evolutionary Nature of Professional Software Development
SCOTT L. BAIN


パターン指向リファクタリング入門
ソフトウェア設計を改善するための作法
デザインパターンを
取り入れて
プログラムを改善しよう


プレファクタリング


ソフトウェアアーキテクチャ
PATTERN-ORIENTED SOFTWARE ARCHITECTURE


第4版
コンピュータネットワーク
アンドリュー・S・タネンバウム著


アジャイルソフトウェア開発の奥義
第2版
オブジェクト指向開発の神髄と匠の技


Java言語で学ぶ
リファクタリング入門
結城 浩・
どうすれば、きれいで読みやすいコードになるのか?
Javaの実例を通して学ぶ
リファクタリング入門書の決定版


リファクタリング
プログラミングの
体質改善テクニック
マーチン・ファウラー 著


デザインパターン
改訂版


プログラムデザインのための
パターン言語


Patterns of Enterprise Application Architecture
エンタープライズアプリケーション
アーキテクチャパターン


TCP/IP
マスタリング
TCP/IP
入門編 第6版
井上直也・村山公保・竹下隆史
荒井 透・苅田幸雄 共著
Ohmsha


アジャイルソフトウェア達人の技
Clean
クリーンコード
Code
ASCII


達人プログラマー
The Pragmatic Programmer


The Pragmatic Programmers
達人プログラマー
ソフトウェア開発に不可欠な基礎知識
ASCII


プログラミング作法
THE PRACTICE OF PROGRAMMING
PROGRAMMING
ASCII


Stephen P. Berczuk,
Brad Appleton
Software Configuration Management Patterns:
パターンによるソフトウェア構成管理
いま、プロジェクトの現場は
混乱に陥っている!!
プロジェクトの失敗を防ぐためには必見!!


新装版
アンチパターン
ソフトウェア危篤患者の救出

TCP/IP
マスタリング
TCP/IP
SSL/TLS編
Eric Rescorla 著
Ohmsha

Code
Reading
コードは小説より面白い!

Code
Quality
コードクオリティ

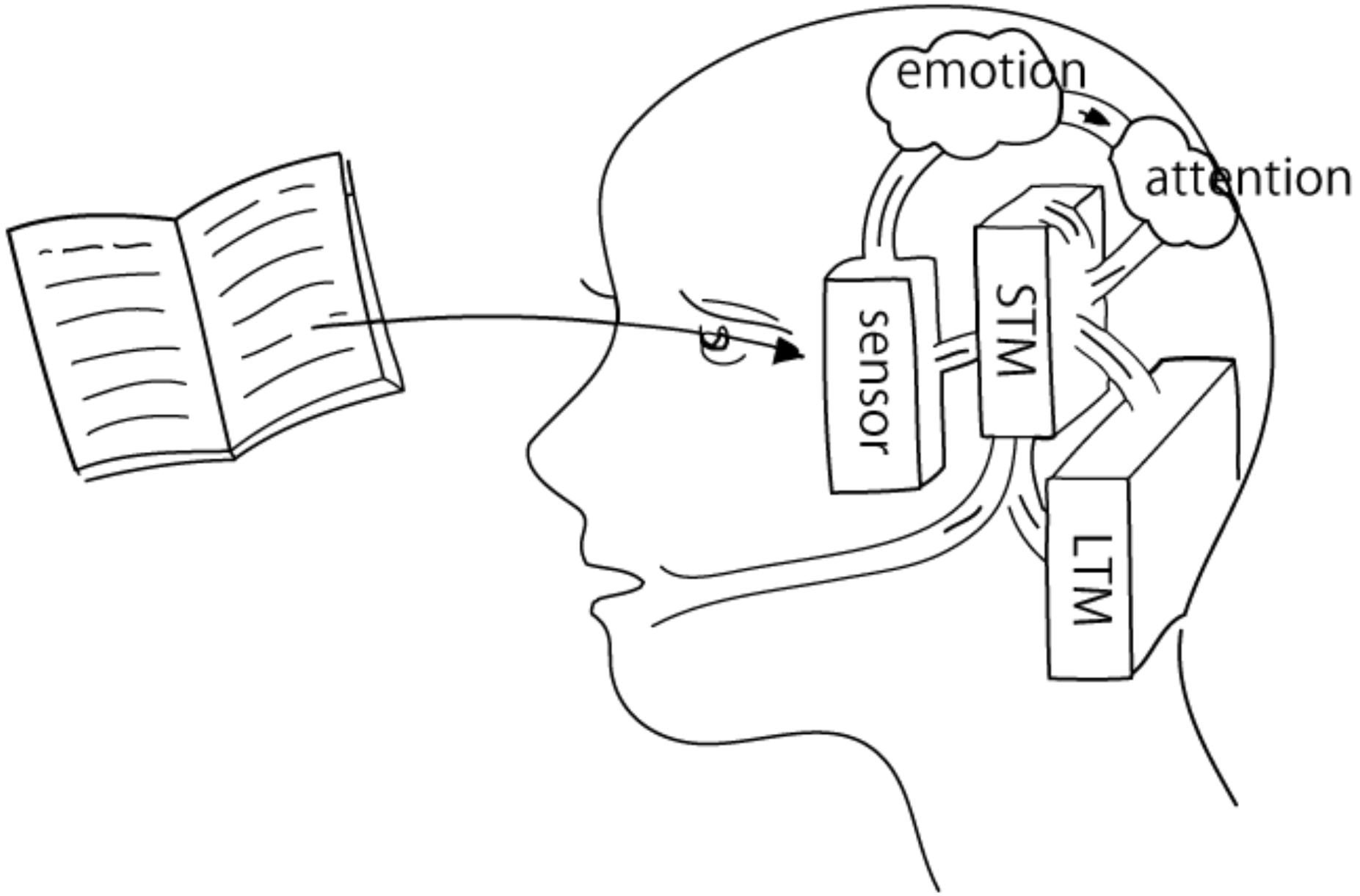
Beautiful
Code
ビューティフルコード

コードコンプリート
CODE
COMPLETE
第2版
完全なプログラミングを目指して
Microsoft Press

ケント・ベックの Smalltalk
ベストプラクティス・パターン
シンプル・デザインへの宝石箱

実践
パターン
PEARSON

学びの仕組み

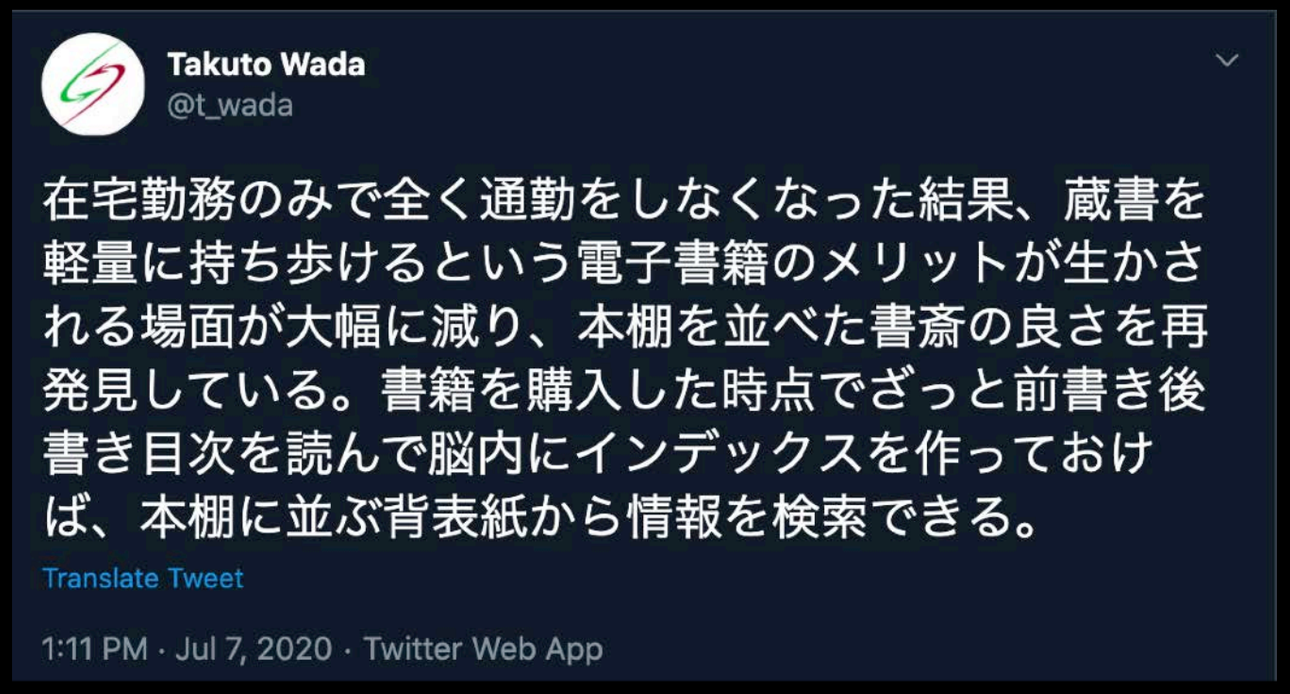


感覚記憶
0.5 ～ 2sec

短期記憶
15 ～ 30 sec

長期記憶
死ぬまで?

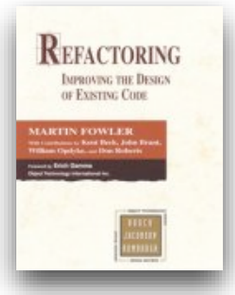
脳内インデックスを作る

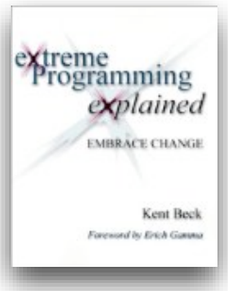


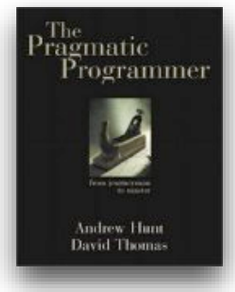
ピッカーを育てる ＝ 反復練習
何度も長期記憶から出し入れする

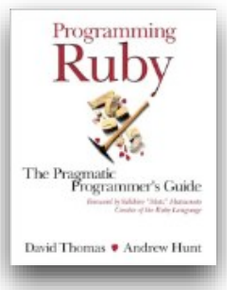
荷物を他の荷物とくっつける
連想記憶を育てる

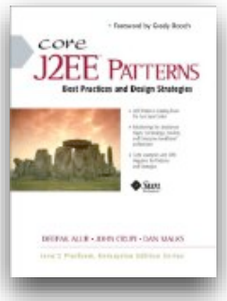
たとえば、時系列に並べる

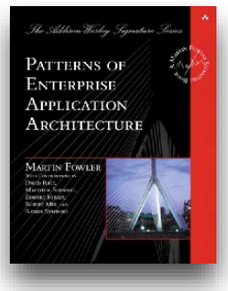


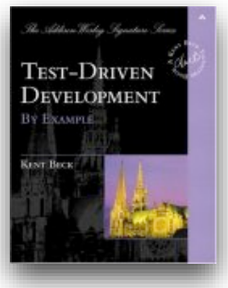


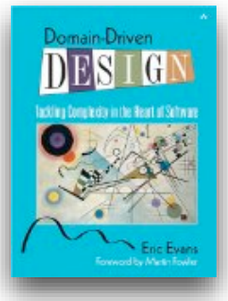


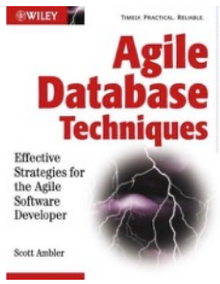


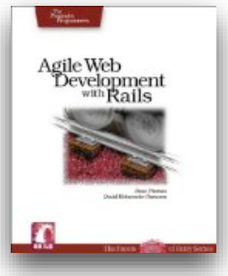


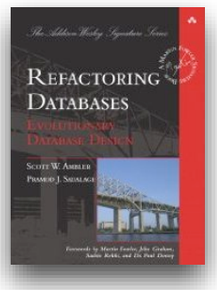


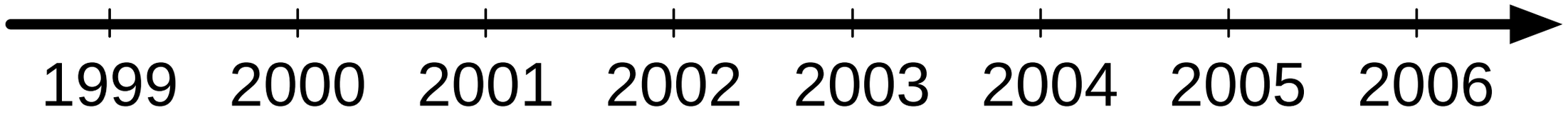












四半期毎に技術書を読む

👉 手を動かして学ぶ

毎年少なくとも1つの言語を学習する

身の回りをプログラミング対象にする

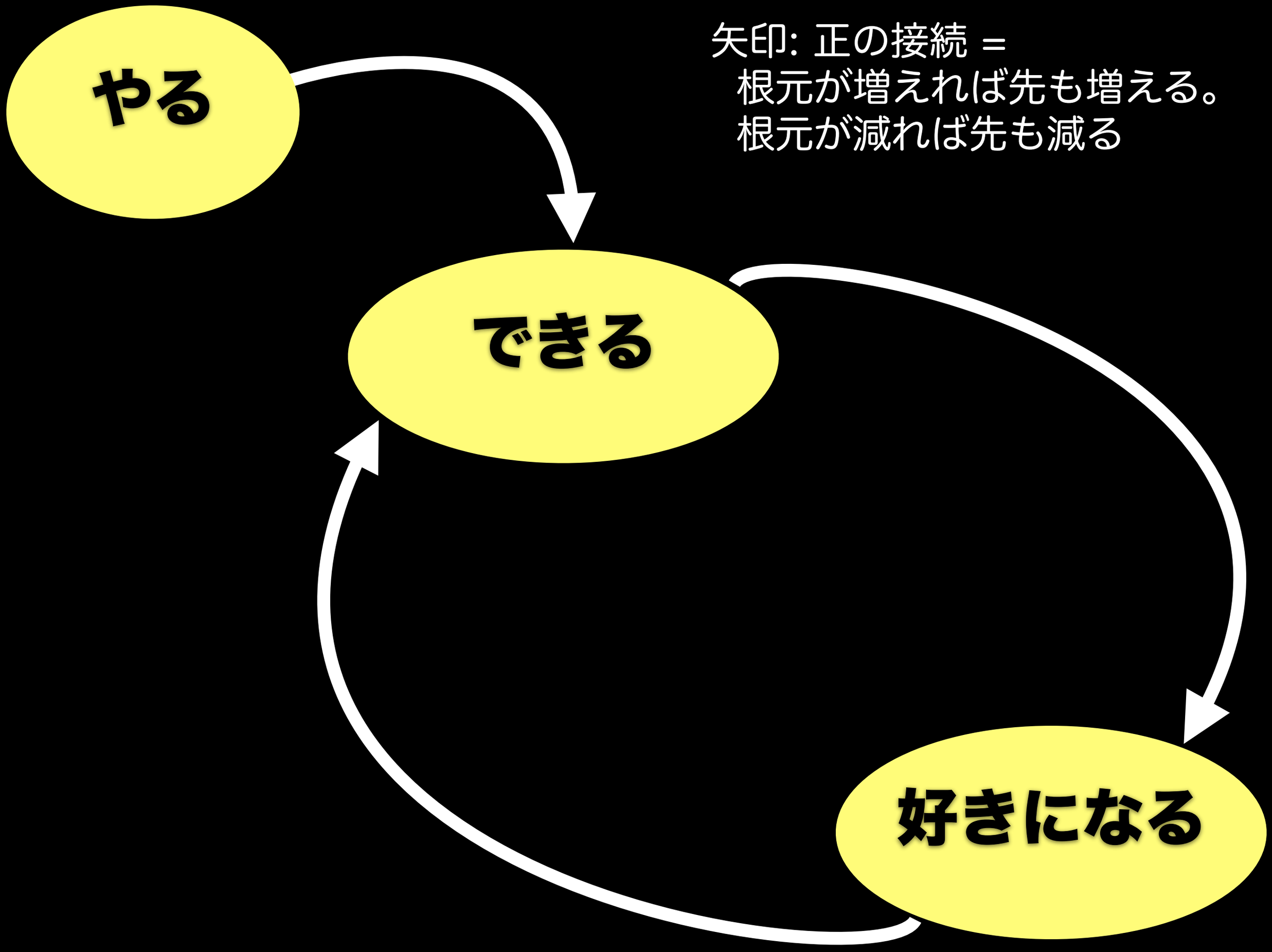
アウトプットを行う

2. “手を動かし

て学ぶ”


97
Collective Wisdom from the Experts
プログラマが
知るべき97のこと
97 Things Every Programmer Should Know
O'REILLY
オライリー・ジャパン
Kevlin Henney 編
和田 卓人 監修
夏目 大 訳

矢印: 正の接続 =
根元が増えれば先も増える。
根元が減れば先も減る



デールの円錐

I see and I forget.
I hear and I remember.
I do and I understand.
— Confucius

After 2 weeks, we tend to remember ...

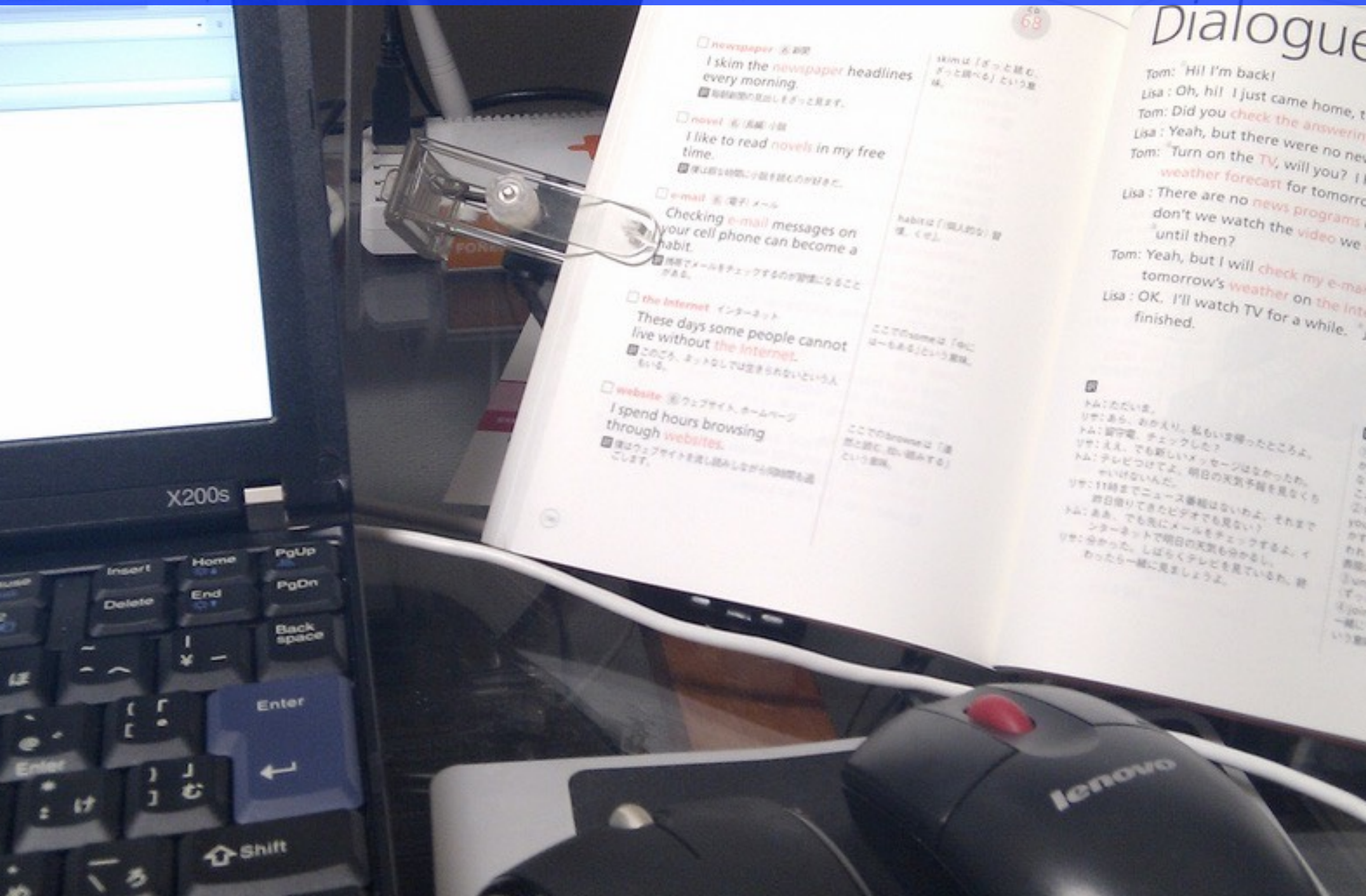
- 10% of what we READ
- 20% of what we HEAR
- 30% of what we SEE
- 50% of what we SEE & HEAR
- 70% of what we SAY
- 90% of what we SAY & DO

Passive

Active

Source: Edgar Dale (1969)

写経


newspaper 名 新聞
I skim the newspaper headlines every morning.
novel 名 長編小説
I like to read novels in my free time.
e-mail 名 電子メール
Checking e-mail messages on your cell phone can become a habit.
the Internet インターネット
These days some people cannot live without the Internet.
website 名 ウェブサイト、ホームページ
I spend hours browsing through websites.
Dialogue
Tom: Hi! I'm back!
Lisa : Oh, hi! I just came home, t
Tom: Did you check the answerin
Lisa : Yeah, but there were no ne
Tom: Turn on the TV, will you? I
weather forecast for tomorro
Lisa : There are no news programs
don't we watch the video we
until then?
Tom: Yeah, but I will check my e-ma
tomorrow's weather on the int
Lisa : OK. I'll watch TV for a while.
finished.
X200s
Insert
Home
PgUp
Delete
End
PgDn
Back space
Enter
Shift
lenovo

Takuto Wada
@t_wada

技術書の「写経」の方法。 1.ローカルで使えるSCMを用意 2.「ほんたった」などで対象の本を固定 3.ひたすらサンプルコードを写して実行 4.実行するたびにコミット(コミットログにページ番号を含める) 5.疑問点があったらコミットログや本に書き込む 6.章ごとにタグを打つ

Translate Tweet

5:06 PM · Feb 12, 2010 · Twitter Web Client

https://twitter.com/t_wada/status/9000231741

四半期毎に技術書を読む

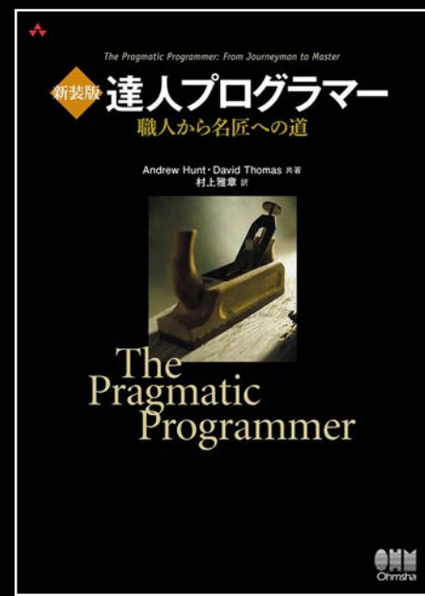
手を動かして学ぶ

👉 毎年少なくとも1つの言語を学習する

身の回りをプログラミング対象にする

アウトプットを行う

3. “毎年少なくとも一つの言語を学習する”



プログラミング言語は複数習得すべき

第二の言語には、是非とも、最初の言語とはパラダイムの違う言語を選ぶべきです。それはなぜかというと、パラダイムの違う言語を学ぶと、アルゴリズム、イディオム、パターンの実装について嫌でも考えるようになるからです。

同様のアルゴリズムを実装するにしても、色々なやりかたがあり得ることに気づきます。この体験が、プログラマの技術を大きく向上させます。

言語だけでなく文化も学ぶ

Andy HuntとDave Thomasは、多くの人に影響を与えた著書『達人プログラマー』の中で、「毎年、新たなプログラミング言語を1つは学ぶこと」と勧めています。私はそのアドバイスに従い、過去何年かの間に実際に数多くの言語を学んできました。

そして、その中で「**言語を学ぶというのは、ただ文法、構文を学ぶことではなく、その背景にある文化も学ぶこと**」という重要な教訓を得ました。

TECHNOLOGY RADAR

Techniques | Tools | Platforms | Languages & Frameworks

Search | About the Radar | Build your Radar | Subscribe

ADOPT

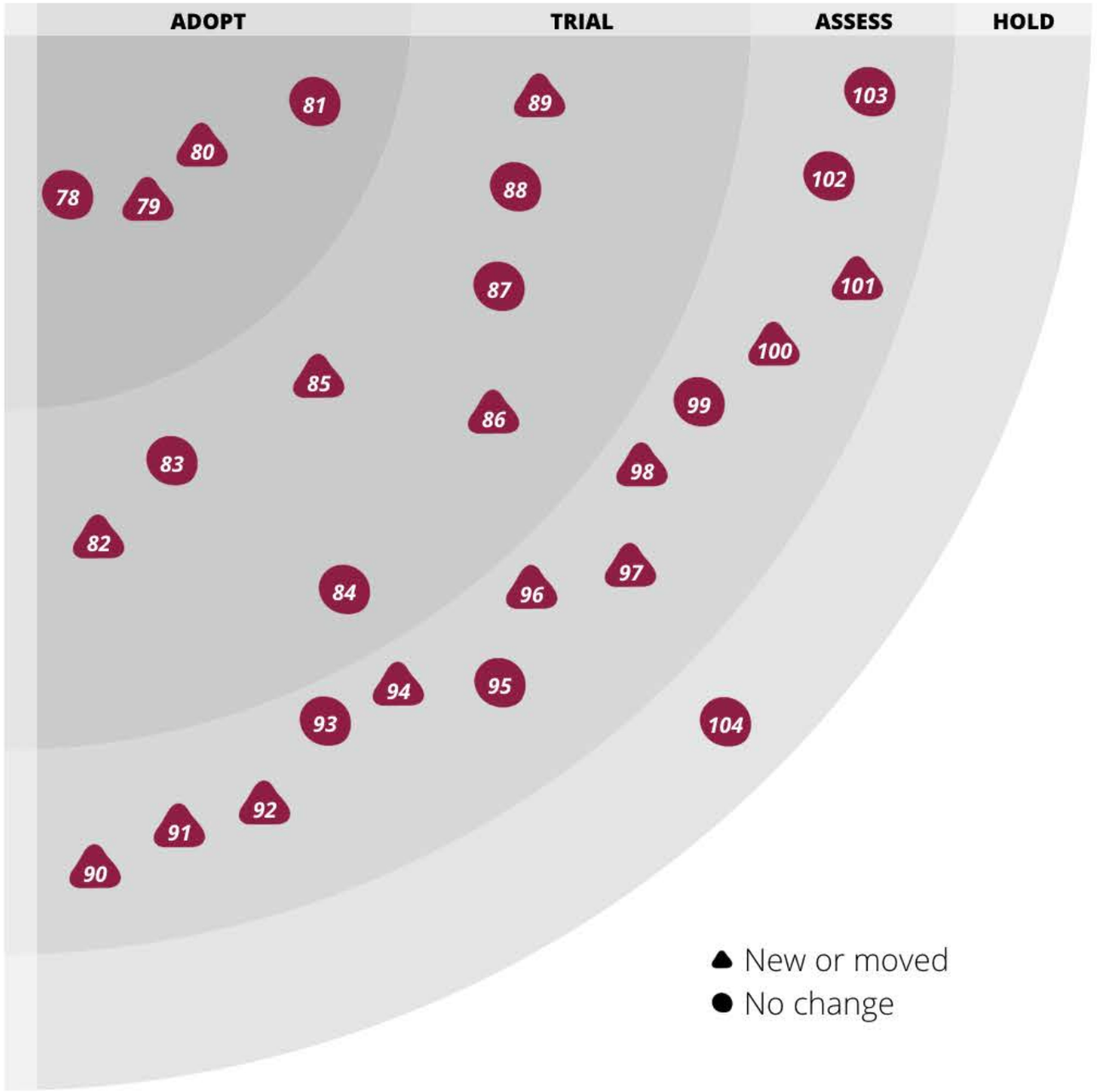
78. Ember.js
79. Python 3
80. ReactiveX
81. Redux

TRIAL

82. Avro new
83. Elixir
84. Enzyme
85. Hangfire new
86. Nightwatch new
87. Phoenix
88. Quick and Nimble
89. Vue.js

ASSESS

90. Angular 2 new
91. Caffe new
92. DeepLearning.scala new
93. ECMAScript 2017



Unable to find something you expected to see? Your item may

http://www.thoughtworks.com/radar/languages-and-frameworks

TECHNOLOGY RADAR

Techniques | Tools | Platforms | Languages & Frameworks

Search | About the Radar | Build your Radar | Subscribe

The information in our interactive Radar is currently only available in English. To get information in your native language, please download the PDF here.

ADOPT

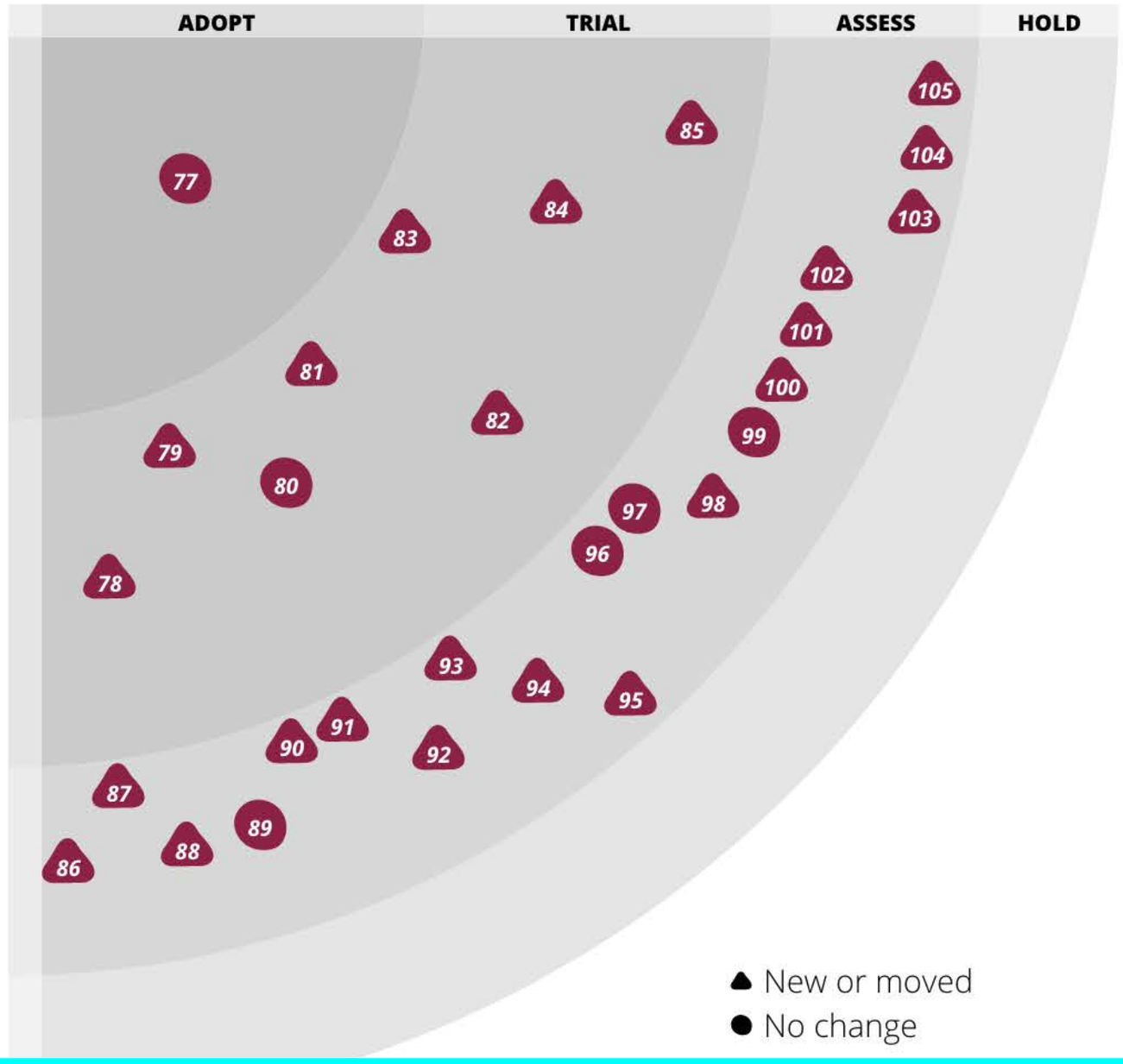
77. Python 3

TRIAL

78. Angular
79. AssertJ new
80. Avro
81. CSS Grid Layout new
82. CSS Modules new
83. Jest new
84. Kotlin
85. Spring Cloud

ASSESS

86. Android Architecture Components new
87. ARKit/ARCore new
88. Atlas and BeeHive new
89. Caffe



http://www.thoughtworks.com/radar/languages-and-frameworks

TECHNOLOGY RADAR

Techniques | Tools | Platforms | Languages & Frameworks

Search | About the Radar | Build your Radar | Subscribe

The information in our interactive Radar is currently only available in English. To get information in your native language, please download the PDF here.

ADOPT

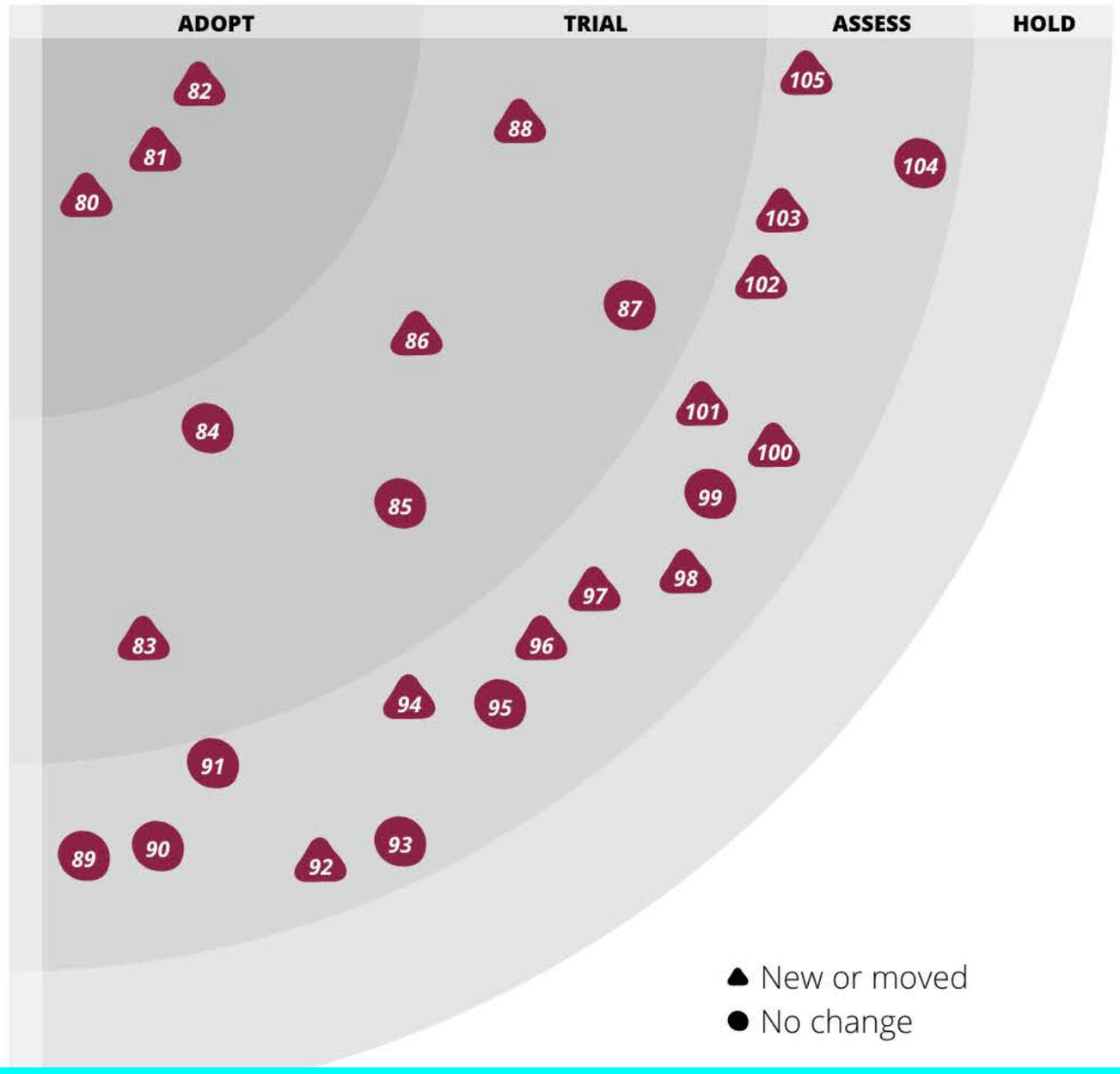
80. AssertJ
81. Enzyme
82. Kotlin

TRIAL

83. Apollo New
84. CSS Grid Layout
85. CSS Modules
86. Hyperledger Composer New
87. Keras
88. OpenZeppelin New

ASSESS

89. Android Architecture Components
90. Atlas and BeeHive
91. Clara rules
92. Flutter New



http://www.thoughtworks.com/radar/languages-and-frameworks

The information in our interactive Radar is currently only available in English. To get information in your native language, please download the PDF here.

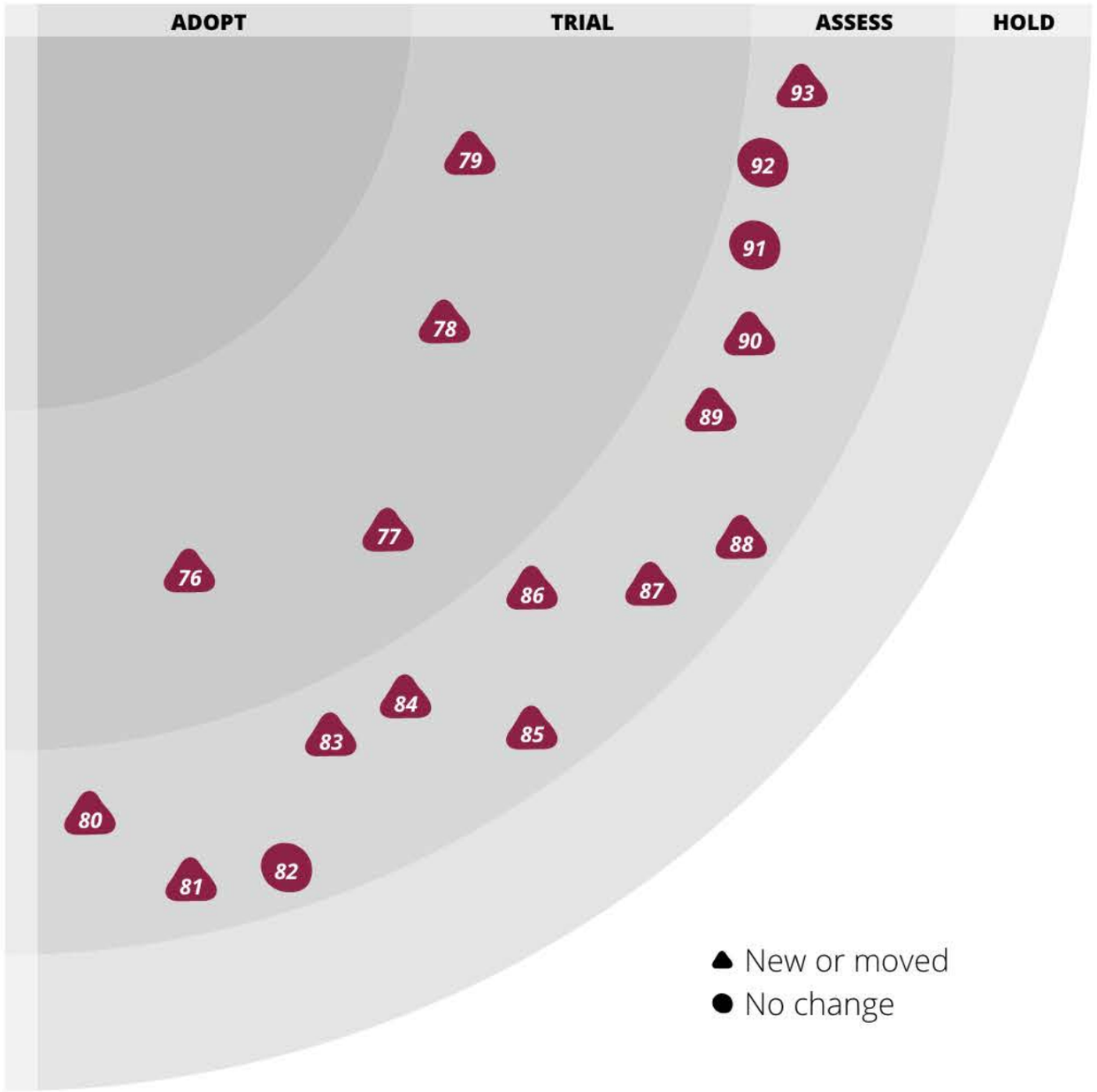
ADOPT

TRIAL

76. Jepsen
77. MMKV New
78. MockK New
79. TypeScript

ASSESS

80. Apache Beam New
81. Camunda New
82. Flutter
83. Ktor New
84. Nameko New
85. Polly.js New
86. PredictionIO New
87. Puppeteer New
88. Q# New
89. SAFE stack New
90. Spek New
91. troposphere
92. WebAssembly
93. WebFlux New



Unable to find something you expected to see?

Each edition of the radar features blips reflecting what we came across during the previous six months. We might have covered what you are looking for on a previous edition already. We sometimes cull

http://www.thoughtworks.com/radar/languages-and-frameworks

ADOPT

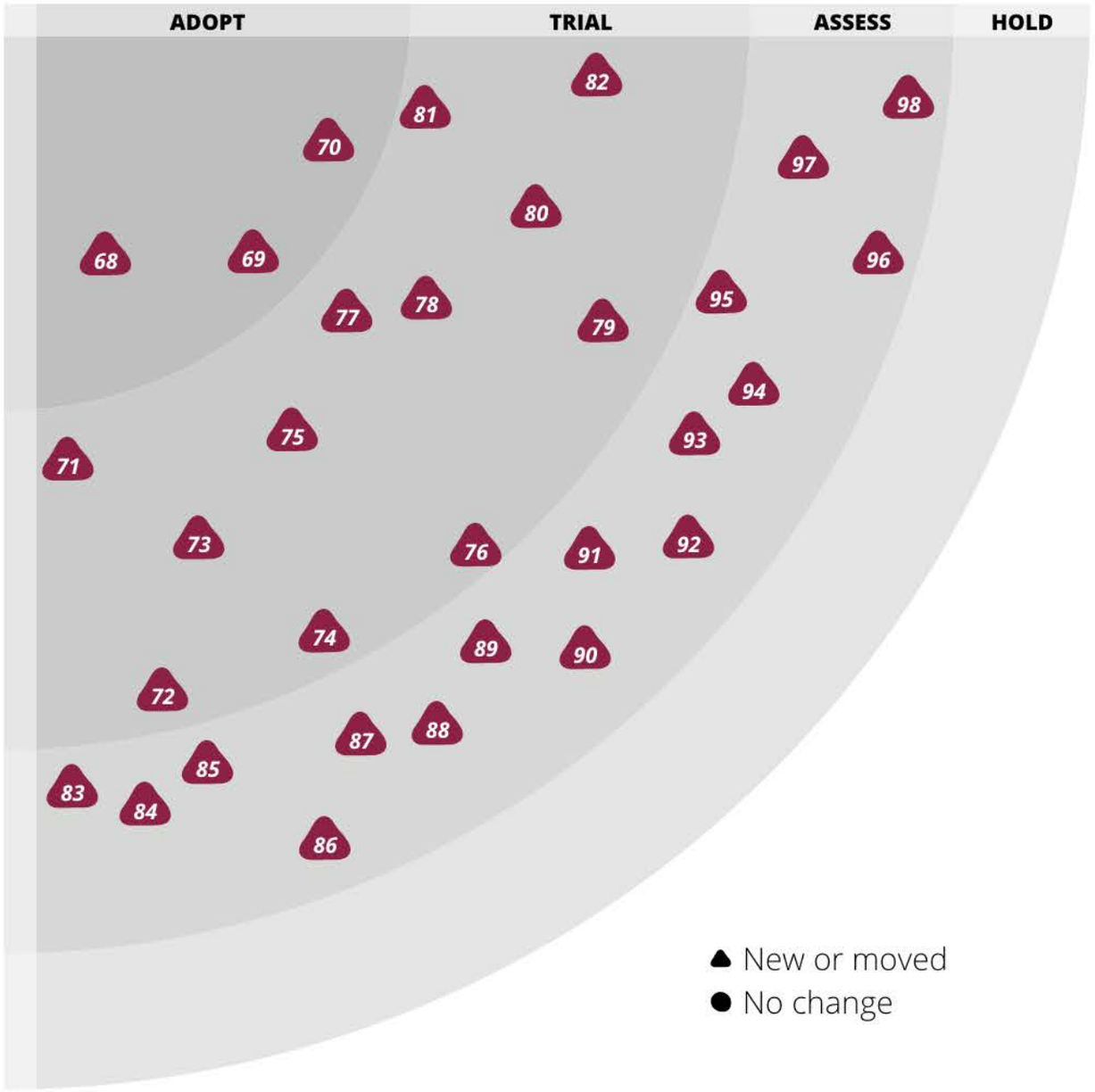
68. Apollo
69. MockK
70. TypeScript

TRIAL

71. Apache Beam
72. Formik
73. HiveRunner
74. joi
75. Ktor
76. Laconia
77. Puppeteer
78. Reactor
79. Resilience4j
80. Room
81. Rust
82. WebFlux

ASSESS

83. Aeron
84. Arrow
85. Chaos Toolkit
86. Dask
87. Embark
88. fastai
89. http4k
90. Immer
91. Karate



Unable to find something you expected to see?

Each edition of the radar features blips reflecting what we came across during the previous six months. We might have covered what you are looking for on a previous radar already. We sometimes cull things just because there are too many to talk about. A blip might also be missing because the radar reflects our experience, it is not based on a comprehensive market analysis.

http://www.thoughtworks.com/radar/languages-and-frameworks

Languages & Frameworks

Adopt

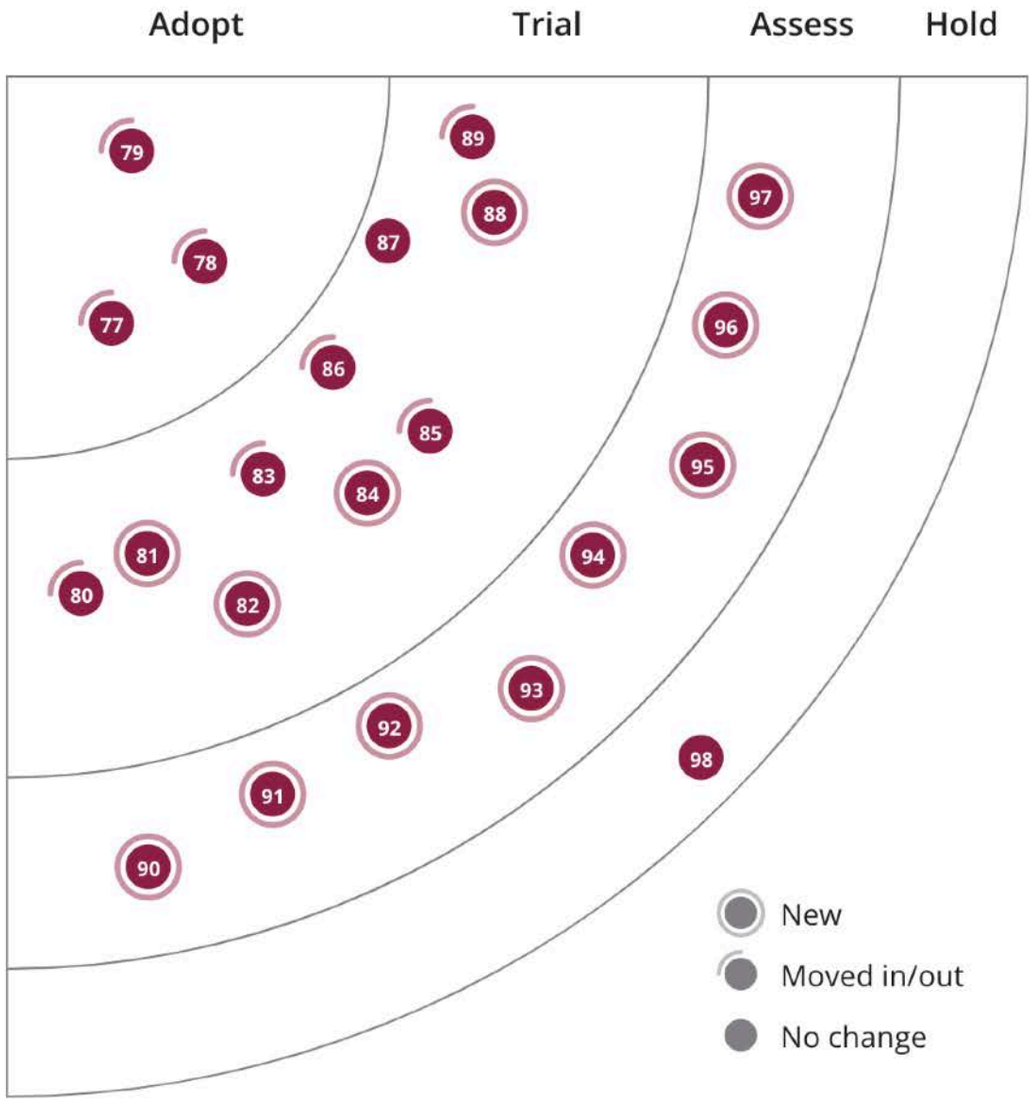
77. React Hooks
78. React Testing Library
79. Vue.js

Trial

80. CSS-in-JS
81. Exposed
82. GraphQL Inspector
83. Karate
84. Koin
85. NestJS
86. PyTorch
87. Rust
88. Sarama
89. SwiftUI

Assess

90. Clinic.js Bubbleprof
91. Deequ
92. ERNIE
93. MediaPipe
94. Tailwind CSS
95. Tamer
96. Wire



Unable to find something you expected to see?

Each edition of the radar features blips reflecting what we came across during the previous six months. We might have covered what you are looking for on a **previous radar** already. We sometimes cull things just because there are too many to talk about. A blip might also be missing because the radar reflects our experience, it is not based on a comprehensive market analysis.

http://www.thoughtworks.com/radar/languages-and-frameworks

技術者と英語

“英語ができるようになるというのは、
「 大きな図書館の鍵」を渡されるような
ものです。一人ひとりの人生にいろんな
可能性を与えてくれます”

—— 高松 珠子

http://blogos.com/article/85541/

四半期毎に技術書を読む

手を動かして学ぶ

毎年少なくとも1つの言語を学習する

👉 身の回りをプログラミング対象にする

アウトプットを行う

4. 身の回りを
プログラミング
対象にする

プログラマ向けの本の監修者は どうあるべきか

Kevlin Henney 編
和田 卓人 監修
夏目 大 訳

O'REILLY®
オライリー・ジャパン

プログラマらしく

怠惰、傲慢、短気

プレーンテキストを好む

すべてをバージョン管理する

すべてを自動化する

変化を抱擁する

プログラマらしく

原稿は markdown 形式

原文はスクレイピングして取得

git を使いバージョン管理

heroku に push してサイトに反映

監修差分は docdiff で表示

97 Things Every Programmer Should Know

EPUB で確認

管理者 | 全原稿 | 監訳中 | 監訳済 | 査読中 | 査読済 | 編集中 | 編集済

個々のエッセイのステータス管理

原文、翻訳原稿、監修原稿へのリンク

翻訳原稿と監修原稿の差分、レビュー後の差分

分別のある行動	Seb Rose	001	原文 翻訳 監訳 差分 N/A	編集済にする
関数型プログラミングを学ぶことの重要性	Edward Garson	002	原文 翻訳 監訳 差分 編集後差分	編集済にする
ユーザが何をするかを観察する（あなたはユーザではない）	Giles Colborne	003	原文 翻訳 監訳 差分 編集後差分	編集済にする
コーディング規約を自動化する	Filip van Laenen	004	原文 翻訳 監訳 差分 編集後差分	編集済にする
美はシンプルさに宿る	Jørn Ølmheim	005	原文 翻訳 監訳 差分 N/A	編集済にする
リファクタリングの際に注意すべきこと	Rajith Attapattu	006	原文 翻訳 監訳 差分 編集後差分	編集済にする
共有は慎重に	Udi Dahan	007	原文 翻訳 監訳 差分 編集後差分	編集済にする
ボーイスカウト・ルール	Robert C. Martin（アンクル・ボブ）	008	原文 翻訳 監訳 差分 編集後差分	編集済にする
他人よりまず自分を疑う	Allan Kelly	009	原文 翻訳 監訳 差分 N/A	編集済にする
ツールの選択は慎重に	Giovanni Asproni	010	原文 翻訳 監訳 差分 N/A	編集済にする
ドメインの言葉を使ったコード	Dan North	011	原文 翻訳 監訳 差分 編集後差分	編集済にする
コードは設計である	Ryan Brush	012	原文 翻訳 監訳 差分 編集後差分	編集済にする
コードレイアウトの重要性	Steve Freeman	013	原文 翻訳 監訳 差分 N/A	編集済にする
コードレビュー	Mattias Karlsson	014	原文 翻訳 監訳 差分 編集後差分	編集済にする
コードの論理的検証	Yechiel Kimchi	015	原文 翻訳 監訳 差分 N/A	編集済にする
コメントについてのコメント	Cal Evans	016	原文 翻訳 監訳 差分 N/A	編集済にする
コードに書けないことのみをコメントにする	Kevlin Henney	017	原文 翻訳 監訳 差分 編集後差分	編集済にする
学び続ける姿勢	Clint Shank	018	原文 翻訳 監訳 差分 N/A	編集済にする
「便利さ」は「利便性」ではない	Gregor Hohpe	019	原文 翻訳 監訳 差分 N/A	編集済にする
すばやくデプロイ、こまめにデプロイ	Steve Berczuk	020	原文 翻訳 監訳 差分 N/A	編集済にする
技術的例外とビジネス例外を明確に区別する	Dan Bergh Johnsson	021	原文 翻訳 監訳 差分 N/A	編集済にする
1万時間の訓練	Jon Jagger	022	原文 翻訳 監訳 差分 編集後差分	編集済にする
ドメイン特化言語	Michael Hunger	023	原文 翻訳 監訳 差分 N/A	編集済にする

見られて恥ずかしいデータは~~入力しない~~使わないこと
==========

Rod Begbie

夜遅くのことでした。~~その時、~~私はページレイアウトのテストのため、~~プレースホルダデータを~~サンプルデータを入力していました。

ユーザ名には、英国のパンクロックバンド、ザ・クラッシュのメンバーの名前を使いました。会社名には、同じく英国のパンクロックバンド、セックス・ピストルズの曲名を使いました。あとは~~ティッカーシンボルを~~ティッカーシンボル((監訳注: 株式市場で上場企業や商品を識別するため付けられる符丁 http://ja.wikipedia.org/wiki/ティッカーシンボル))を入れるだけです。そこで私は、卑猥な4文字の言葉を大文字で入れることにしたのです。

そう、皆さんご存知の、"F"で始まる言葉なんかを使ったわけです。

特に問題はないだろうと思っていました。自分や他のプログラマが面白がって見るだけのものだし、どうせ翌日には「本物」のデータソースに入れ替えることになっていたからです。ところが翌朝、プロジェクトマネージャが~~、~~一件の4文字言葉の表示されている画面のスクリーンショットをとり、あるプレゼンに使ってしまったのです。

プログラミング~~作業中には~~中の、~~どうしても、~~この種の「いたずら」~~をしたくなるものですが、そうするとプログラミング履歴に妙なデータが入り込んでしまうことになります~~や「武勇伝」は良くある話です。~~こういうのは戦争手記みたいなもので~~「まあ誰も見ないのだから」と油断していると、思いがけず多くの人の目に触れてしまうことがあるのです。

露呈の仕方は様々ですが、いずれにしろ珍しいことではありません。~~また~~しかし、関わったプログラマ個人や、開発チーム、あるいは~~会社全体にとって命取りになってしまうこともあります~~会社全体にとっては命取りになりかねません。どんなパターンがあり得るか、例をいくつかあげておきましょう。

* ~~ステータスミーティング中~~進捗会議中、まだ機能が実装されていないボタンを顧客がクリックしてしまう。すると「二度とクリックすんじゃねーぞ、バーカ！」というメッセージが表示される。
* レガシーシステムの保守を担当するプログラマが、エラーダイアログの追加を指示される。そこで彼は、既存のログ機能の出力を利用しようと考える。しかし、元々は裏で~~動いていたログ機能で、~~動いていて出力がユーザの目に~~触れることはなかったものである。しかし、その改造によって~~触れることはなかったログ機能だったために、保守作業によって「おい、バットマン、データベースのクソ野郎がヘマをしやがったぜ！」といったメッセージが画面に表示され、ユーザから見えるようになってしまう。
* 誰かが、テスト用の管理インタフェースと、製品版の管理インタフェースを混同してしまい、ふざけたデータを入力してしまう。その結果、オンラインストアの顧客が、「等身大ビル・ゲイツ型マッサージロボット - 価格100万ドル」といった商品が売られているのを目にすることになる。

「好事門を出でず、悪事千里を走る」というのは古くから言われることです。今の時代なら、なおさらそうでしょう。誰かの「あら」が見つかれば、その噂は、Digg、Twitter、Flib-Flarbなどにより、あっという間に世界中に広まってしまいます。担当者が寝ている間に、タイムゾーンの違う国に広まってしまえば、何も手を打つことはできないでしょう。

~~ソースコードの中でさえ、まったく安心とは言えません。2004年にはWindows 2000のソースコードのTAR書庫がファイル共有ネットワークに流出しています。その時は、ソースコードに卑猥な言葉、ふざけた言葉が使われていないか、grepで嬉々として調べている人間が大勢いました*（実を言えば、その時に発見された // TERRIBLE HORRIBLE NO GOOD VERY BAD HACK = 「実に実にひどい、まったく良いところのない、最悪のハッキングだ」というコメントを私は気に入ってしまい、以来、何度か使ってしまっています・・・）。~~

~~要するには~~ソースコードの中でさえ、~~コードに何かテキストを入力する~~まったく安心とは言えません。2004年にはWindows 2000のソースコードのTAR書庫がファイル共有ネットワークに流出しています。その時は~~一一コメントであれ、あるいはログ~~ソースコードに卑猥な言葉、~~ダイアログ~~ふざけた言葉が使われていないか、~~テストデータであれ一一常に~~grepで嬉々として調べている人間が大勢いました((http://www.kuro5hin.org/story/2004/2/15/71552/7795))（実を言えば、その時に発見された // TERRIBLE HORRIBLE NO GOOD VERY BAD HACK = 「~~これがもし公になったとして問題にならないか」と自問せよ~~実に実にひどい、~~ということです。そうすれば~~まったく良いところのない、~~突然~~最悪のハッキングだ」というコメントを私は気に入ってしまい、~~卑猥な言葉が大写しになり~~以来、~~その場にいる全員が顔を赤らめるというような事態は防げます~~何度か使ってしまっています・・・）。

~~[* http://www.kuro5hin.org/story/2004/2/15/71552/7795](http://www.kuro5hin.org/story/2004/2/15/71552/7795)~~ 要するには、コードに何かテキストを入力する時は一一コメントであれ、あるいはログ、ダイアログ、テストデータであれ一一常に「これがもし公になったとして問題にならないか」と自問せよ、ということです。そうすれば、突然、卑猥な言葉が大写しになり、その場にいる全員が赤面するというような事態は防げます。

最近つくったもの

子育てや教育関係の

LINE Bot

Amazon Alexa Skill

など

四半期毎に技術書を読む

手を動かして学ぶ

毎年少なくとも1つの言語を学習する

身の回りをプログラミング対象にする

👉 アウトプットを行う

5. アウトプットを行う

デールの円錐

I see and I forget.
I hear and I remember.
I do and I understand.
— Confucius

After 2 weeks, we tend to remember ...

- 10% of what we READ
- 20% of what we HEAR
- 30% of what we SEE
- 50% of what we SEE & HEAR
- 70% of what we SAY
- 90% of what we SAY & DO

Passive

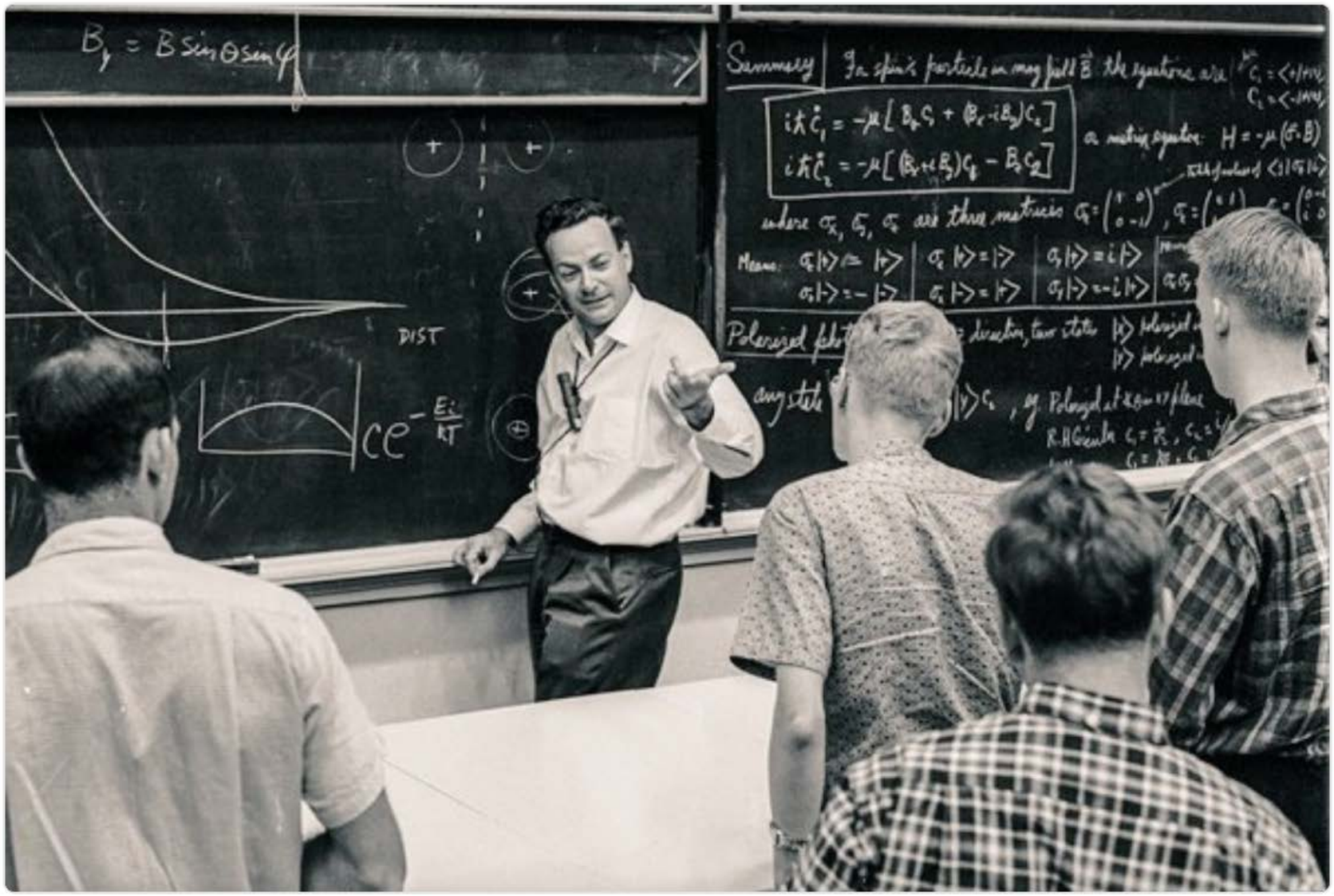
Active

Source: Edgar Dale (1969)

Richard Feynman
@ProfFeynman

Follow

If you want to master something, teach it.



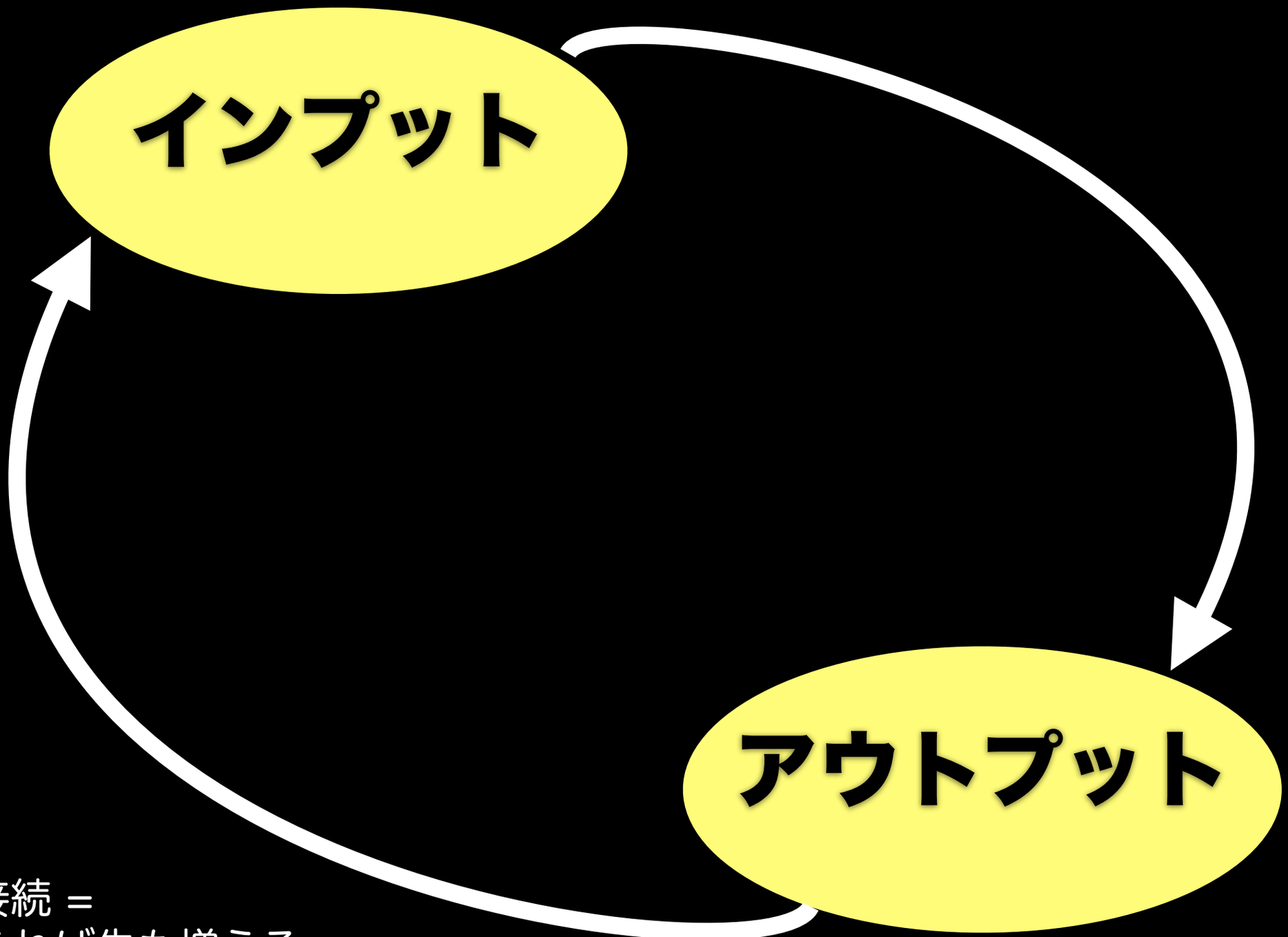
2:15 AM - 6 Apr 2018

1,808 Retweets **4,299** Likes

35 1.8K 4.3K

https://twitter.com/ProfFeynman/status/981943455508148225

正のフィードバックループ



矢印: 正の接続 =
　根元が増えれば先も増える。
　根元が減れば先も減る

量は質に転化する

http://d.hatena.ne.jp/KZR/20080808/p1

blog を書く

2013-12-25 tdd

■ 不具合にテストを書いて立ち向かう

テストを行っている品質保証チームや、実際にシステムを使っているお客様から不具合が報告されたとき、あなたはどう思いますか？ 悲しんだり、恥ずかしいと思い、不具合修正にすぐに着手したいと気がはやるのが人情というものです。しかし、焦っているときに行う作業はしばしば視野が狭く、一つの不具合修正が三つの新たな不具合を生んでしまうようなことになりがちです。

テスト駆動開発（TDD : Test Driven Development）は、プログラマが自分の不安を克服し、自分が書くコードに自信を持ちながら一歩一歩進んでいくための手法です。不具合の発生は、端的に言えばこれまでの「自信」を揺らがせる事態です。テスト駆動開発者は不具合にどう立ち向かうのでしょうか？

やはりテストを書いて立ち向かってゆくのです。私はテスト駆動開発を数年間実践してきた中で、心がけているひとつの「掟」があります。それは「**不具合の修正時には必ず先に不具合を再現する自動テストを書いてから修正する**」というものです。これはもちろん私の発案ではなく、 XP（eXtreme Programming） や TDD の先達から学び、それを実践するうちに私にも身についてきたものです。例えば『テスト駆動開発入門』では、次のように記されています。

> *欠陥が報告されたときに最初にすべきことは何か。→ 失敗する最小のテストを作成し、実行した後に修正する。*
>
> *回帰テストは、完全な先見があれば、最初のコーディング時に作成していたテストである。回帰テストを作成するたびに、どうすれば、最初にそのテストを作成できたかを考えよう。*
>
> *―― 『テスト駆動開発入門』 p.136 回帰テスト*

不具合修正時のテストは、次のような手順で行います。

01. 手元で不具合を再現させる

“情報発信、blog，発表，公開などは、数学の(未解決問題の)証明ではなく、料理のようなもの”

執筆する


Webアプリケーション開発のためのプログラミング技術情報誌
WEB+DB PRESS
[ウェブDBプレス]
ムービーもWeb配信
vol.35
2006
特集1
実演! プログラマのための設計技法
テスト駆動開発
特集2 "速く・美しく・効率的に"開発するための
Ajaxの定石
特集3 [定番・新定番] 早ミミ情報!
PostgreSQL
負荷対策ツール
特別企画 [速習] Java, PHP, C言語
コードリーディング
ビギナー必見!
Javaの地図帳
▶Railsと日本語
▶Smartyプラグインでらくらく開発
▶JBoss SeamによるEJB開発


Test-Driven Development By Example
テスト駆動開発
Kent Beck 著
和田卓人 訳
テスト駆動開発(TDD)を
原点から学ぶ
本書は、Kent Beck, Test-Driven Development by Example
(Addison-Wesley, 2003)の新訳版です。
Ohmsha

技術同人誌市場の登場

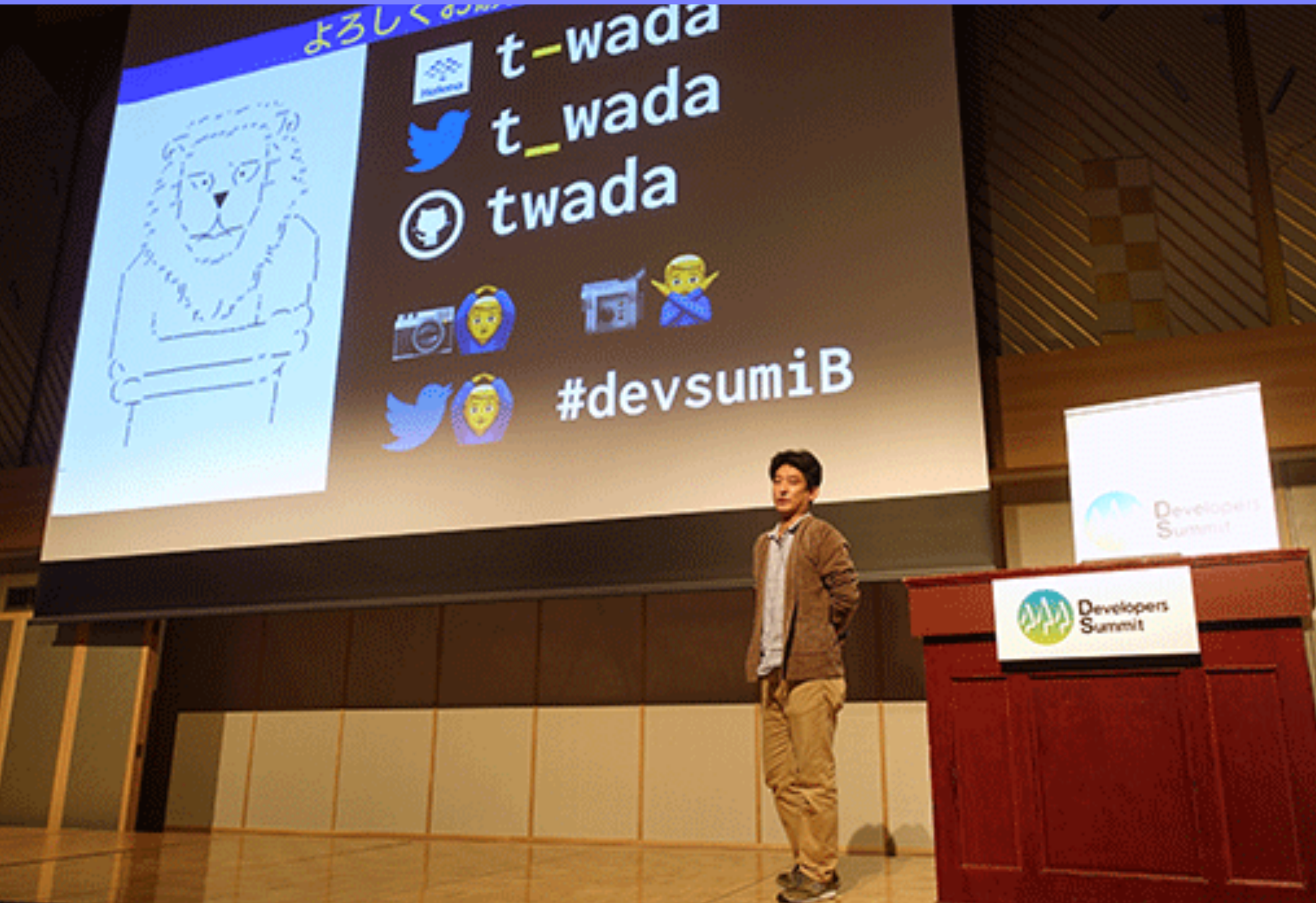
イベント情報　サークル参加要項とガイド　マーケットのFAQ　スポンサー募集要項

Illustration by shatiko.

技術書オンリーイベント 第9回 技術に出会えるオンラインイベントに参加しよう！

https://techbookfest.org/event/tbf09

講演する(まずは仲間内のLTから)

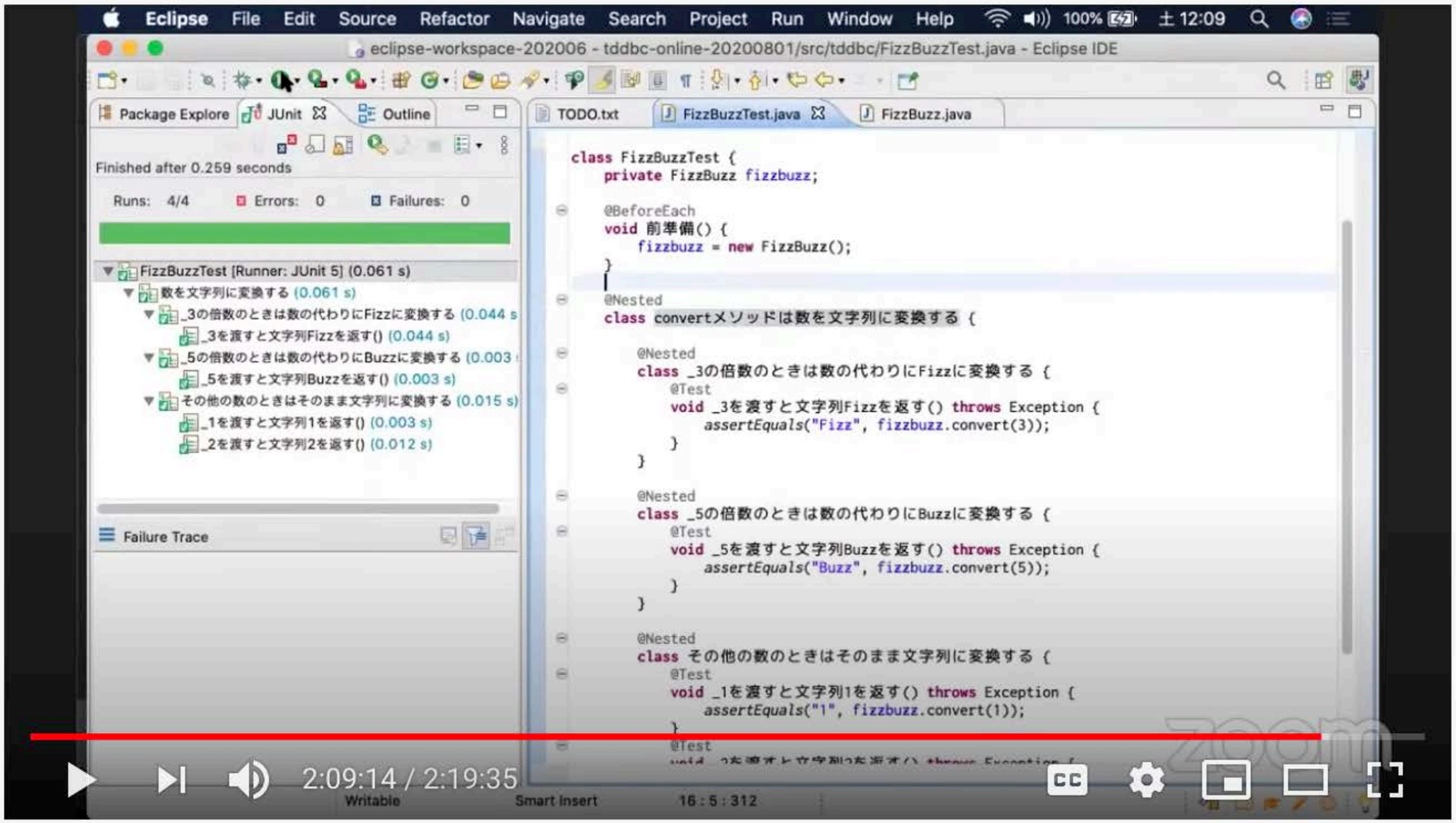


ライブコーディングの効果

動画配信

YouTube JP

Search



TDD Boot Camp 2020 Online #1 基調講演/ライブコーディング

13,577 views • Streamed live on Jul 31, 2020　482　2　SHARE　SAVE

TDDBC-Online
493 subscribers

SUBSCRIBE

https://www.youtube.com/watch?v=Q-FJ3XmFIT8

コードを公開する

Takuto Wada
twada

Block or report user

Towersquest Inc.
Tokyo, Japan
http://tako3.com/http://github…

Organizations

Pinned repositories

power-assert-js/power-assert
Power Assert in JavaScript. Provides descriptive assertion messages through standard assert interface. No API is the best API.
JavaScript ★ 1.7k 35

unassert-js/unassert
Encourages programming with assertions by providing tools to compile them away.
JavaScript ★ 118 3

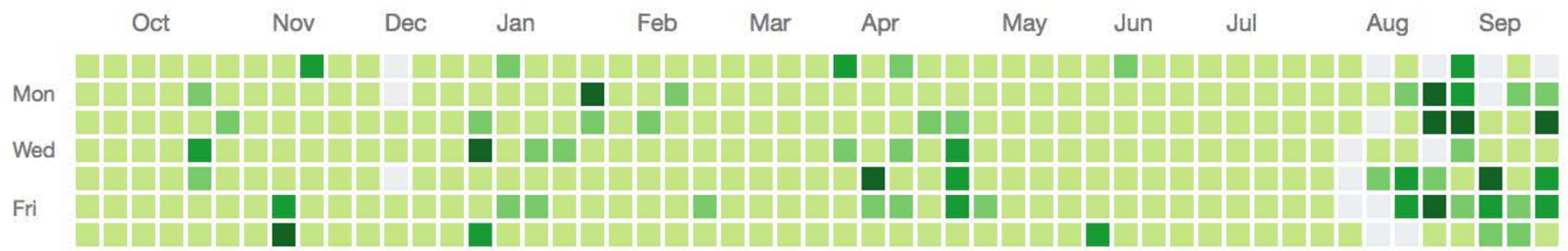
licensify
Browserify plugin to prepend license header to your bundle
JavaScript ★ 142 9

qunit-tap
A TAP Output Producer Plugin for QUnit
JavaScript ★ 73 14

type-name
Just a reasonable `typeof`
JavaScript ★ 65 4

coverlay.el
Test coverage overlay for Emacs
Emacs Lisp ★ 47 6

1,695 contributions in the last year



power-assert

```
assert(typeof item.id === 'strong')
       |      |    |  |
       |      |    |  false
       |      |    "foo"
       |      Item{id:"foo",name:"bar"}
       "string"
```

```
--- [string] 'strong'
+++ [string] typeof item.id
@@ -1,6 +1,6 @@
 str
-o
+i
 ng
```

https://github.com/power-assert-js/power-assert

アウトプットのチャネル

- 書く
  - フロー型メディア（Twitter）
  - ストック型メディア（blog, Qiita 等）
  - 技術同人誌
  - 雑誌記事（Web媒体, 紙媒体）
  - 書籍（共著, 単著, 翻訳, 監訳）
- 話す
  - 講演（社内勉強会, 社外LT, 社外講演）
    - 特にライブコーディング
  - 動画配信
- GitHubでコード公開

ご清聴ありがとうございました

四半期毎に技術書を読む

手を動かして学ぶ

毎年少なくとも1つの言語を学習する

身の回りをプログラミング対象にする

アウトプットを行う