



# ソフトウェアクラフトマンシップと プログラマーの誓い

2022年11月02日

ワイクル株式会社

角征典 (かどまさのり) @kdmsnr

[kado.masanori@waicrew.com](mailto:kado.masanori@waicrew.com)

# 自己紹介



▶ 角 征典 (@kdmsnr)

• 技術書の翻訳・執筆 →

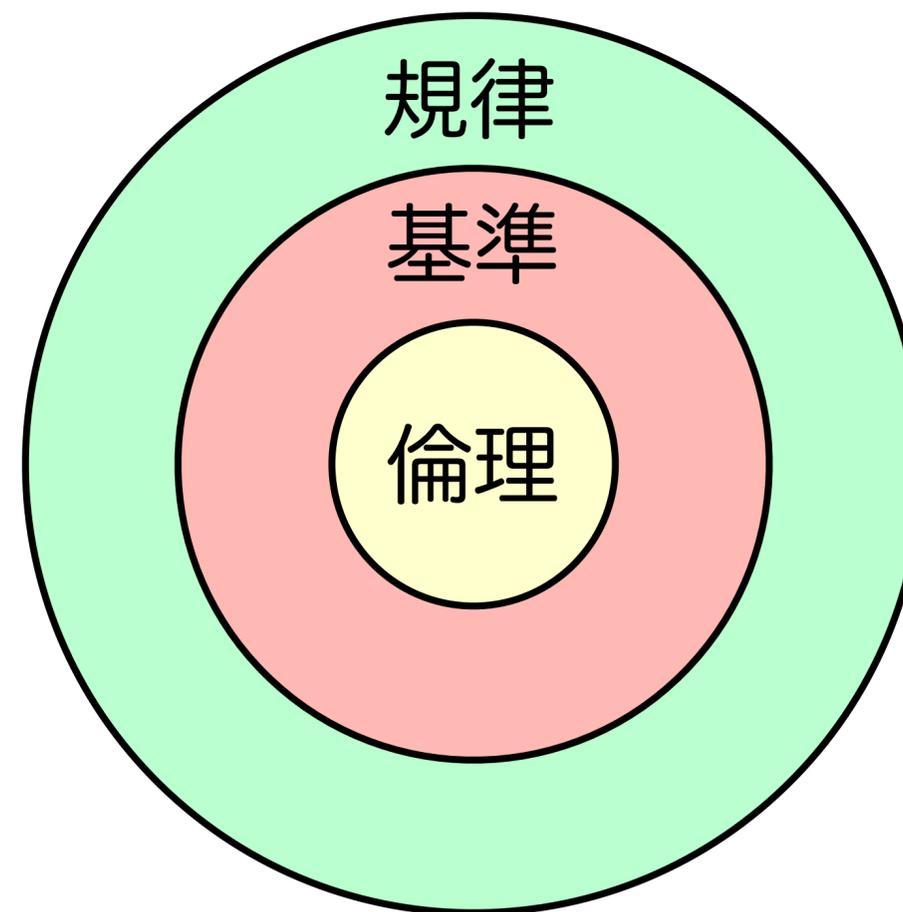
▶ ワイクル株式会社 代表取締役

• アジャイル開発／リーンスタータアップの導入支援

▶ 東京工業大学 環境・社会理工学院 特任講師

• エンジニアのためのデザイン思考





※同心円はClean Architectureのマネ（本には登場しない）

# きょうお話しすること

1. アンクル・ボブのこと
2. ソフトウェアクラフトマンシップ
3. プログラマーの誓い

# 1. アンクル・ボブのこと

# アングル・ボブ (1952~)



# アंकウル・ボブの歴史

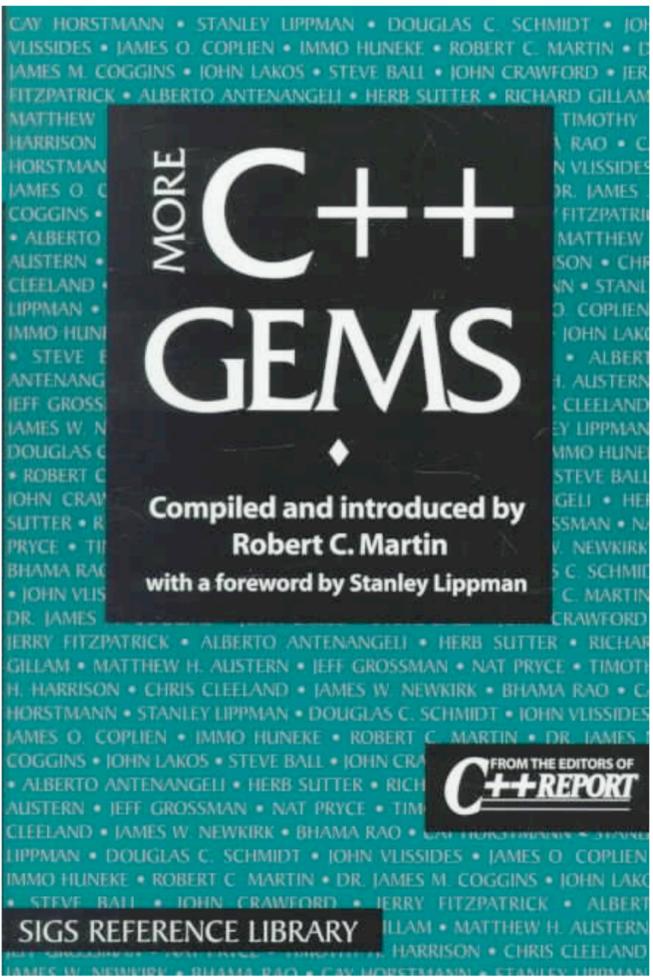
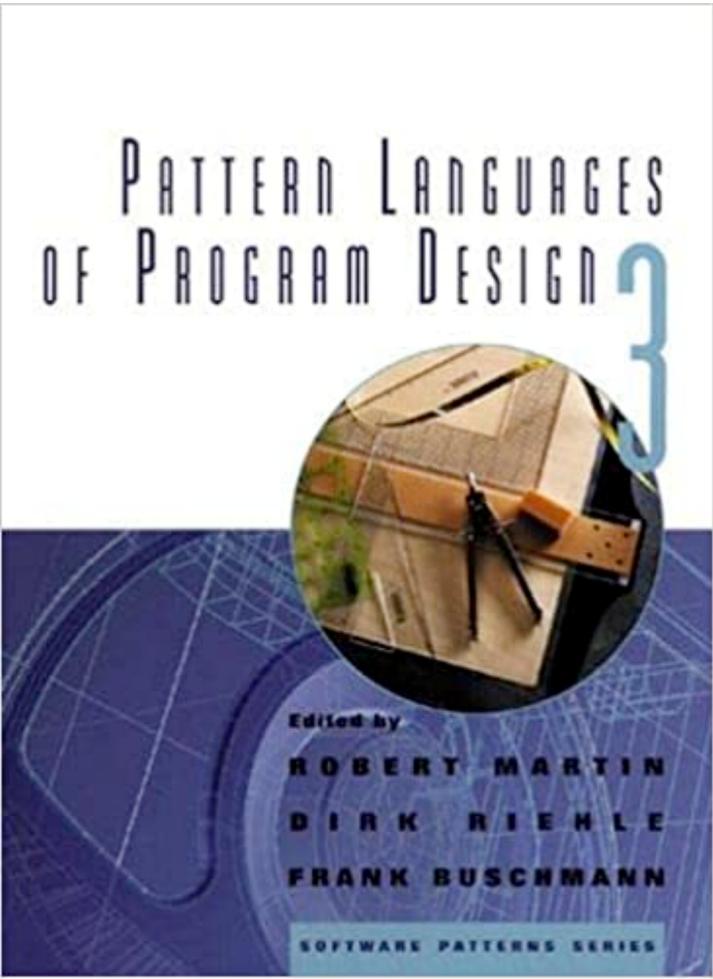
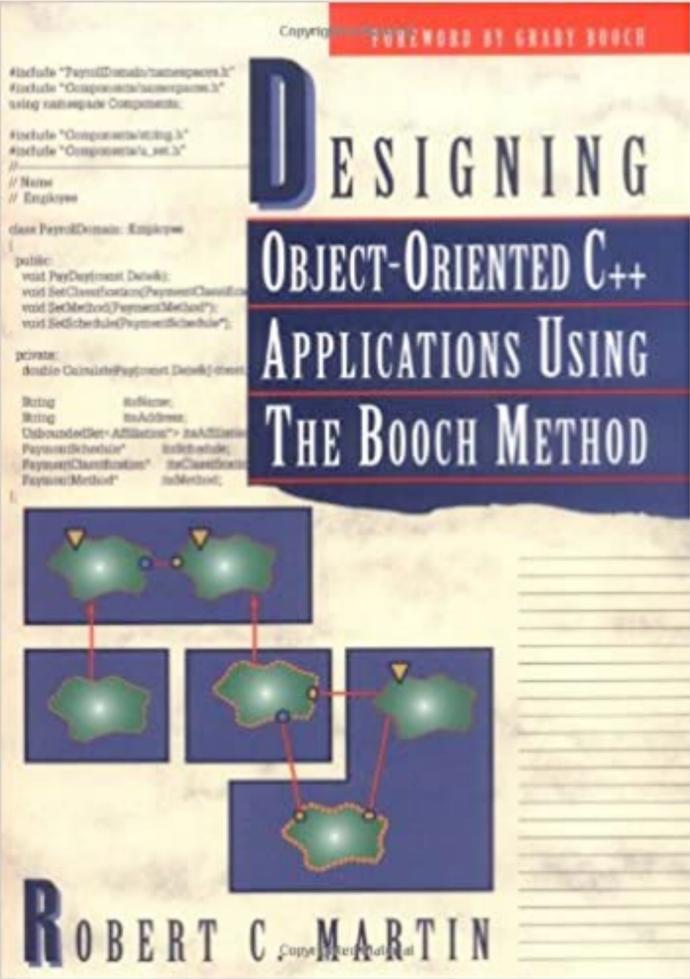
- ▶ 1952年生まれ
- ▶ 1970年：高校卒業後に就職
- ▶ 1972年(?)：結婚、解雇、再就職
- ▶ 1976～1988年：Teradyne社（大企業）
- ▶ 1986～1990年：スタートアップにジョイン
  - ・その後、倒産
  - ・この頃にC++とオブジェクト指向に出会う
  - ・「アंकウル・ボブ」と呼ばれる
  - 最初はイヤだったが、後に愛着がわく
- ▶ 1990～1991年：フリーランスでRational
  - ・ Grady BoochとUMLツール開発
- ▶ 1991～2010年：Object Mentor（起業）
  - ・ 受託開発、コンサルティング、教育
- ▶ 1996～1999年：『C++ Report』誌編集長



[https://en.wikipedia.org/wiki/C++\\_Report](https://en.wikipedia.org/wiki/C++_Report)

# その頃の著書

オブジェクト指向、パターン、C++



# アングル・ボブの歴史

- ▶ 1952年生まれ
- ▶ 1970年：高校卒業後に就職
- ▶ 1972年(?)：結婚、解雇、再就職
- ▶ 1976～1988年：Teradyne社（大企業）
- ▶ 1986～1990年：スタートアップにジョイン
  - ・その後、倒産
  - ・この頃にC++とオブジェクト指向に出会う
  - ・「アングル・ボブ」と呼ばれる
  - 最初はイヤだったが、後に愛着がわく
- ▶ 1990～1991年：フリーランスでRational
  - ・ Grady BoochとUMLツール開発
- ▶ 1991～2010年：Object Mentor（起業）
  - ・ 受託開発、コンサルティング、教育
- ▶ 1996～1999年：『C++ Report』誌編集長
- ▶ 1999年：Kent BeckとTDDに出会う

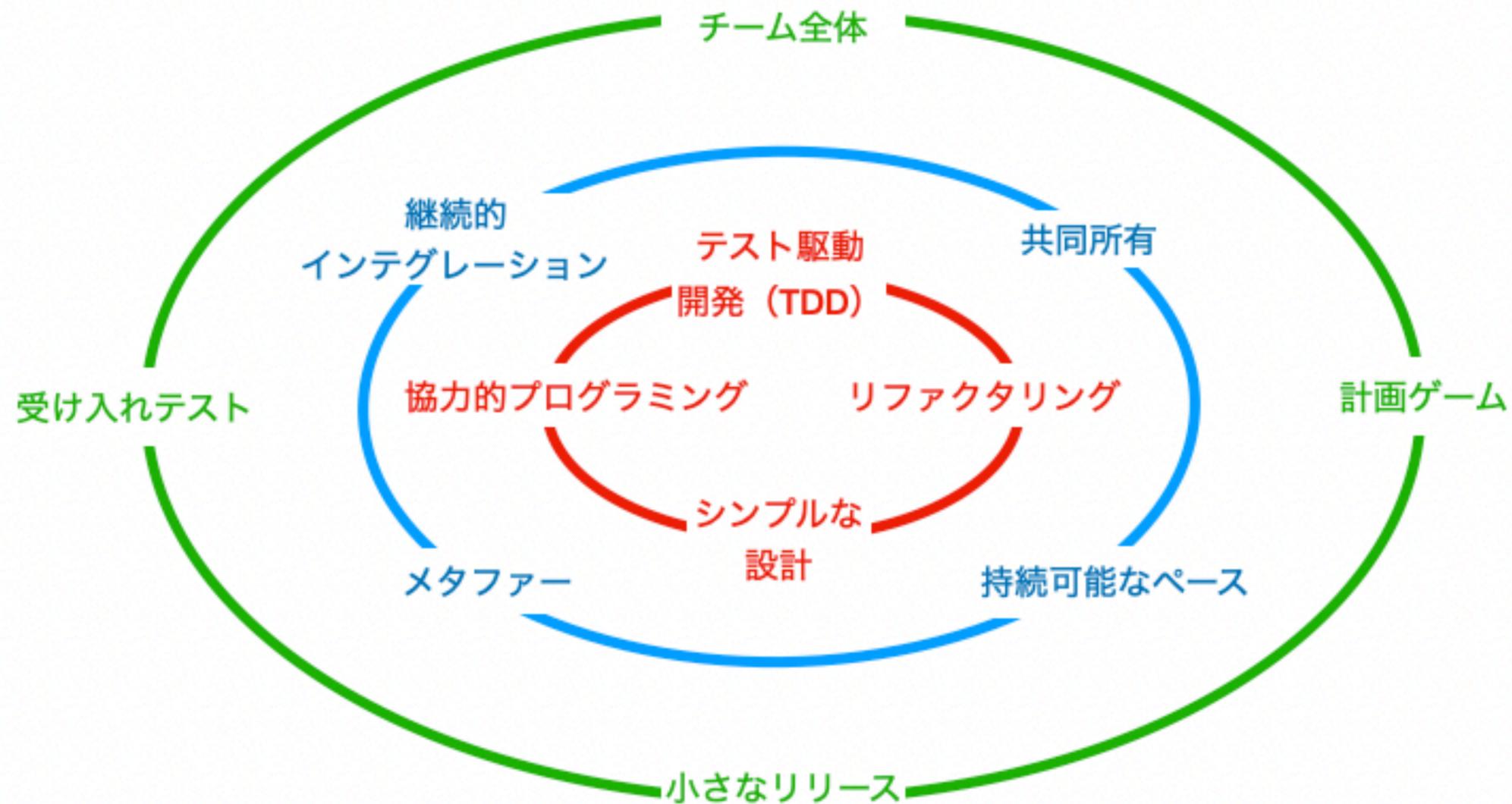
# はじめてプログラミングの「規律」を見た (1999)

これまでの「パターン」や「ノウハウ」とは違う！



# プログラマーの「規律」

⇒ エクストリームプログラミング（特にTDD）





YOSHIKI SHIBATA

merpay

マイクロサービスの開発とテストファースト/テスト駆動開発

merpay

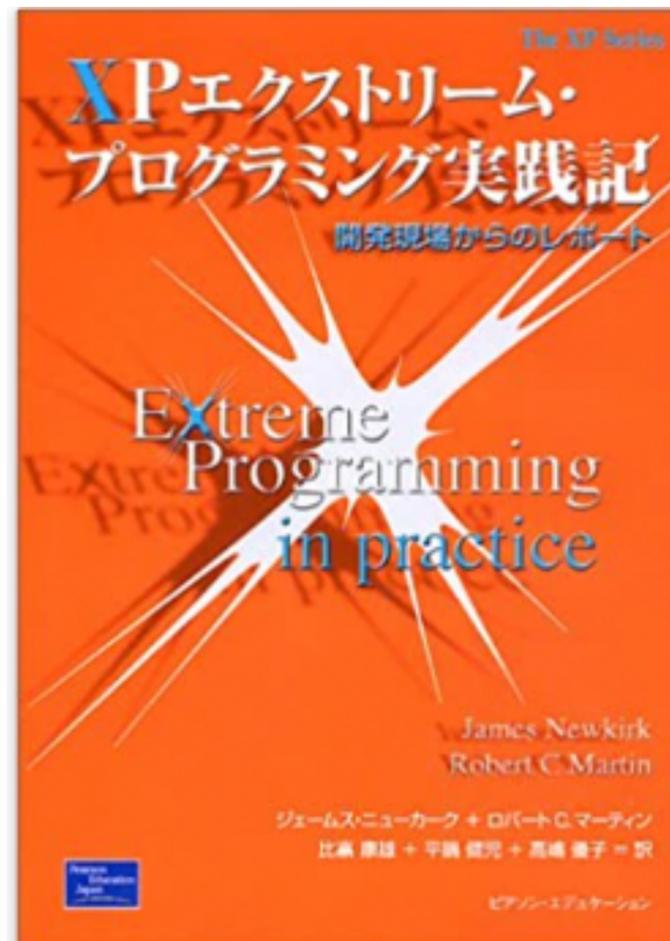
ここまでのまとめ

1980~1990 年代のテストは  
手作業で行い、かつ目視で  
確認するのが主流だった

<https://www.youtube.com/watch?v=Tr8vmTg2NVA>

# その頃の著書

エクストリーム・プログラミング



# アンクル・ボブの歴史

- ▶ 1952年生まれ
- ▶ 1970年：高校卒業後に就職
- ▶ 1972年(?)：結婚、解雇、再就職
- ▶ 1976～1988年：Teradyne社（大企業）
- ▶ 1986～1990年：スタートアップにジョイン
  - ・その後、倒産
  - ・この頃にC++とオブジェクト指向に出会う
- ▶ 1990～1991年：フリーランスでRational
  - ・ Grady BoochとUMLツール開発
- ▶ 1991～2010年：Object Mentor（起業）
  - ・ 受託開発、コンサルティング、教育
- ▶ 1996～1999年：『C++ Report』誌編集長
- ▶ 1999年：Kent BeckとTDDに出会う
- ▶ 2001年：アジャイルマニフェストの起案

# アジャイルソフトウェア開発宣言 (2001)



私たちは、ソフトウェア開発の実践  
あるいは実践を手助けをする活動を通じて、  
よりよい開発方法を見つけだそうとしている。  
この活動を通して、私たちは以下の価値に至った。

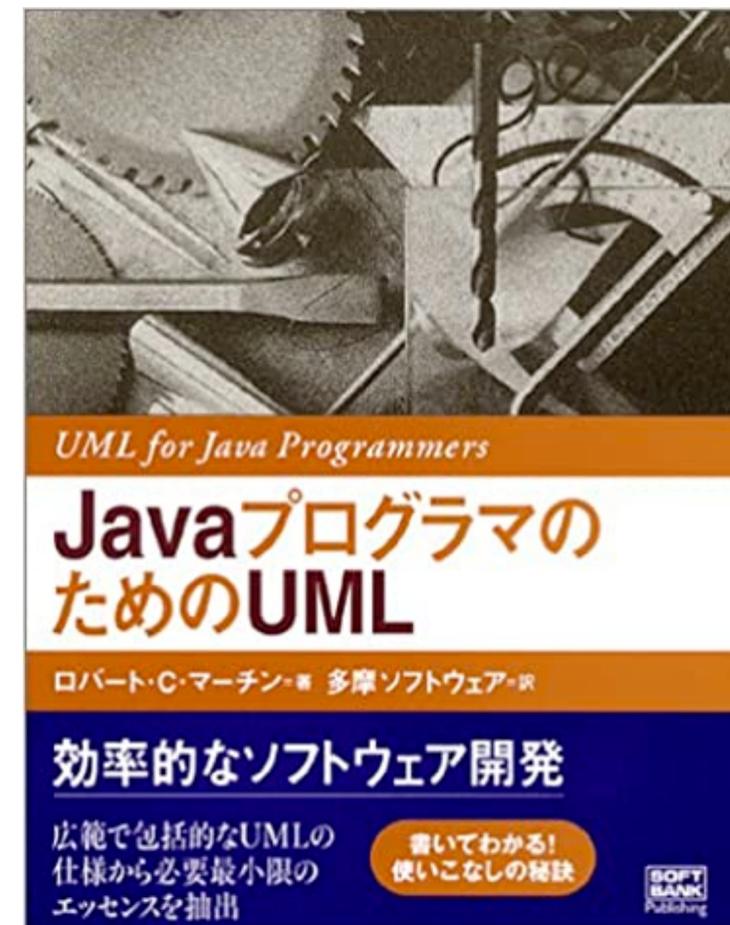
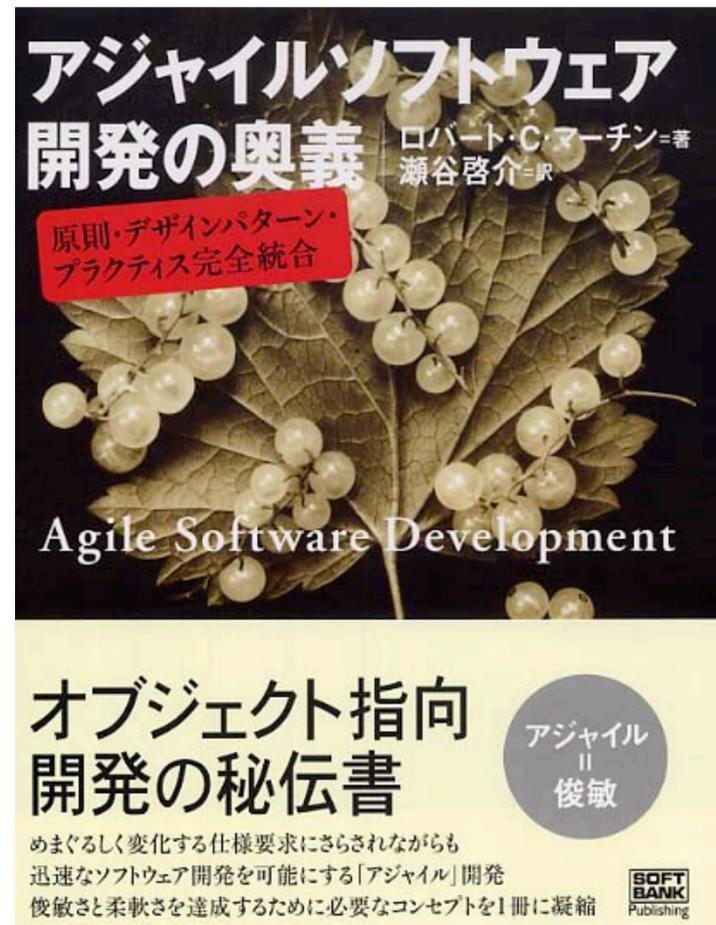
プロセスやツールよりも個人と対話を、  
包括的なドキュメントよりも動くソフトウェアを、  
契約交渉よりも顧客との協調を、  
計画に従うことよりも変化への対応を、

価値とする。すなわち、左記のことがらに価値があることを  
認めながらも、私たちは右記のことがらにより価値をおく。

<https://agilemanifesto.org/iso/ja/manifesto.html>

# その頃の著書

アジャイル、オブジェクト指向、デザインパターン、UML



# へロへロスクラム (2009)

## FlaccidScrum

29 January 2009



**Martin Fowler**

- ◆ AGILE
- ◆ AGILE ADOPTION
- ◆ BAD THINGS

There's a mess I've heard about with quite a few projects recently. It works out like this:

- They want to use an agile process, and pick Scrum
- They adopt the Scrum practices, and maybe even the principles
- After a while progress is slow because the code base is a mess

<https://martinfowler.com/bliki/FlaccidScrum.html>

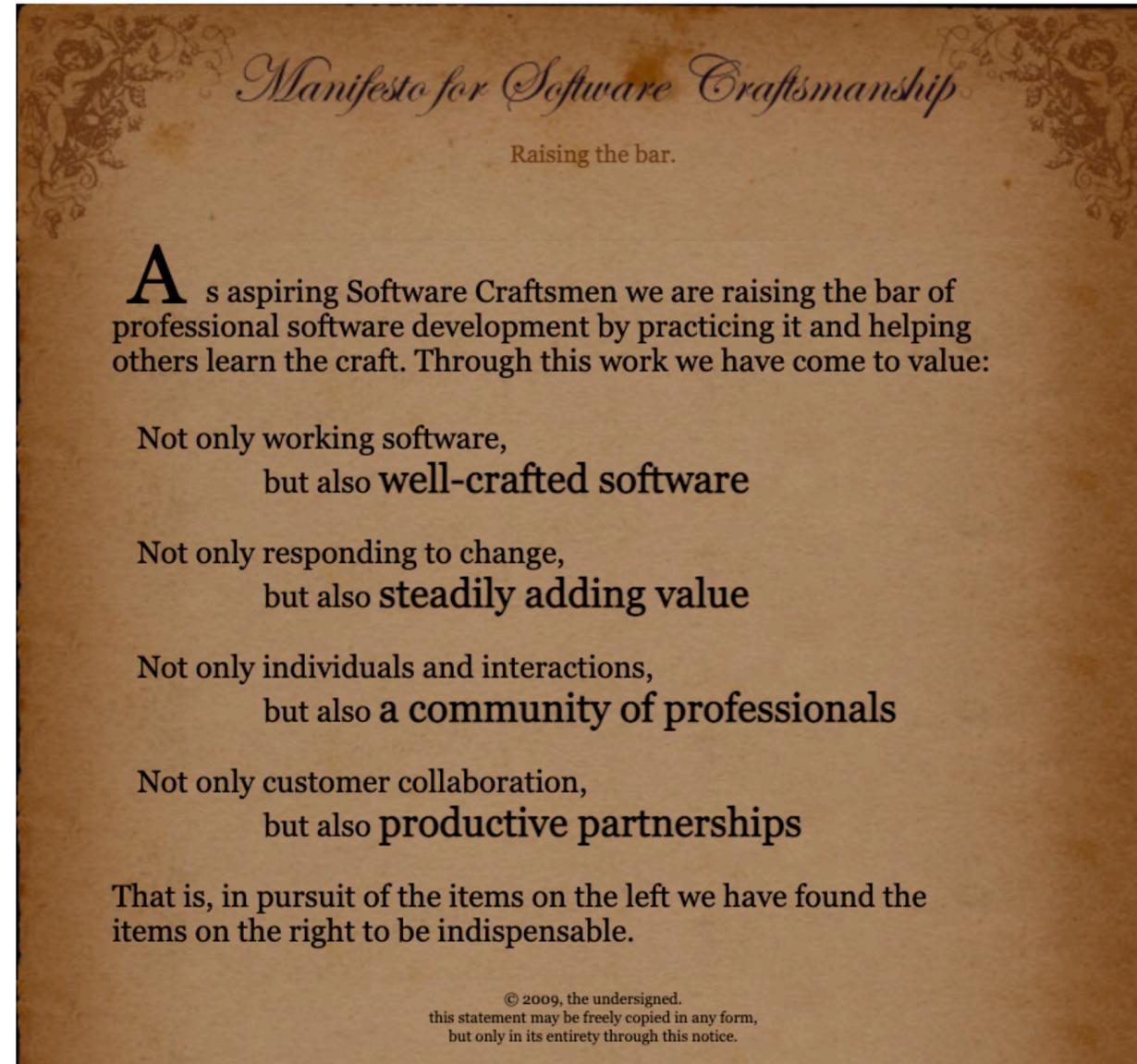
日本語訳：<https://bliki-ja.github.io/FlaccidScrum/>

# アジャイルの二日酔い

- ▶ 多くの組織では、アジャイルとスクラムが同義語になっている。
- ▶ アジャイルコーチは、テクニカルプラクティスをコーチできるほどの技術スキルを持っていない。エンジニアリングについて話すこともほとんどない。
- ▶ アジャイルと開発者はお互いに離れようとしている。

『Clean Agile』 Bob C. Martin, Sandro Mancuso

# ソフトウェアクラフトマンシップ宣言 (2009)



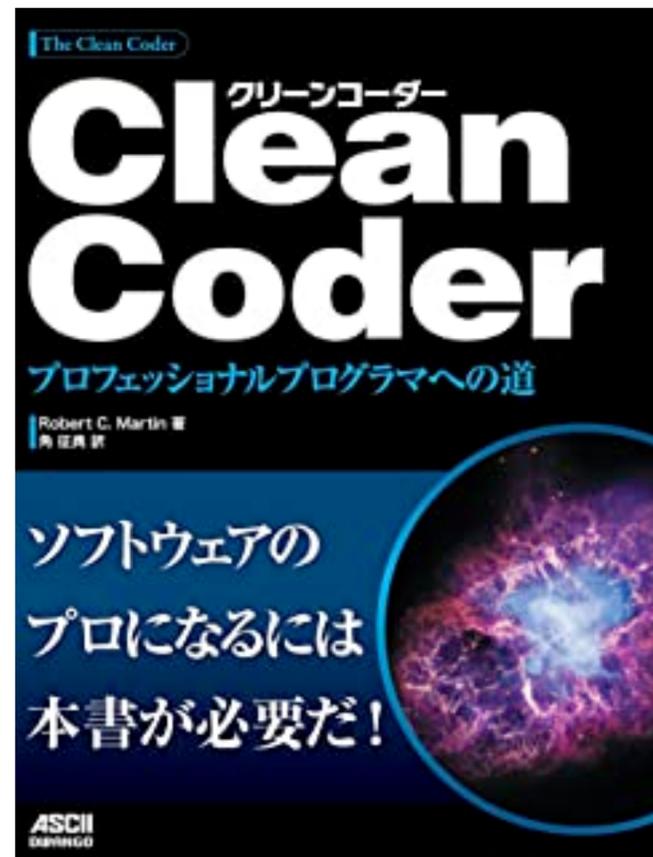
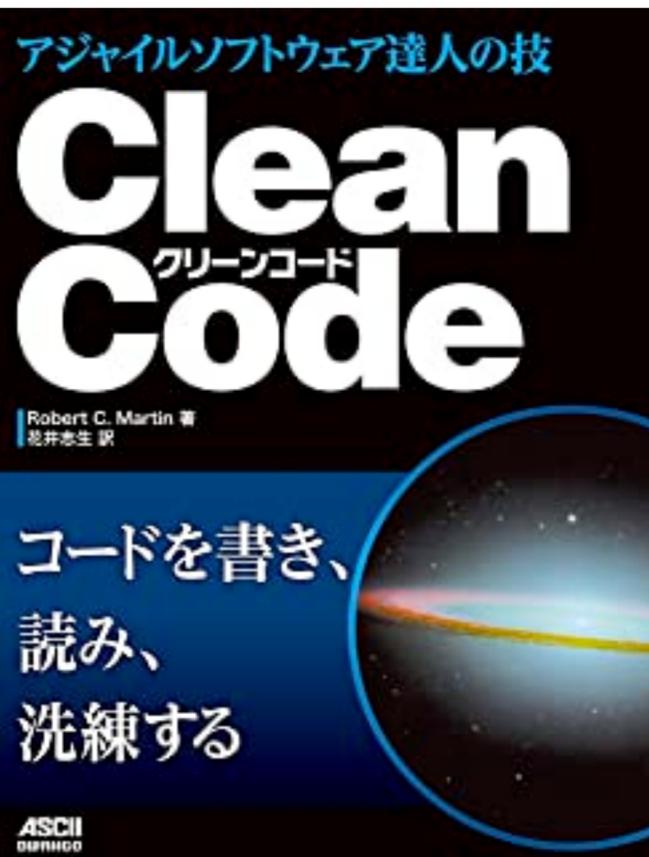
私たちは意欲的なソフトウェアクラフトマンとして、ソフトウェアクラフトマンシップの実践あるいは専門技術の学習の手助けをする活動を通じて、プロとしてのソフトウェア開発の水準を引き上げようとしている。この活動を通して、私たちは以下の価値に至った。

動くソフトウェアだけでなく、精巧に作られたソフトウェアも変化への対応だけでなく、着実な価値の付加も個人との対話だけでなく、専門家のコミュニティも顧客との協調だけでなく、生産的なパートナーシップも

すなわち、左記のことからを追求するなかで、右記のことからも不可欠であることがわかった。

<https://manifesto.softwarecraftsmanship.org/>

# クリーン・シリーズ (2008~)



# おまけ：「クリーン」とは？



<https://agile-douga.tv/products/an-interview-with-uncle-bob-on-clean-agile>

# クラフトマンシップと裂け目 (2011)

## CraftmanshipAndTheCrevasse

19 January 2011



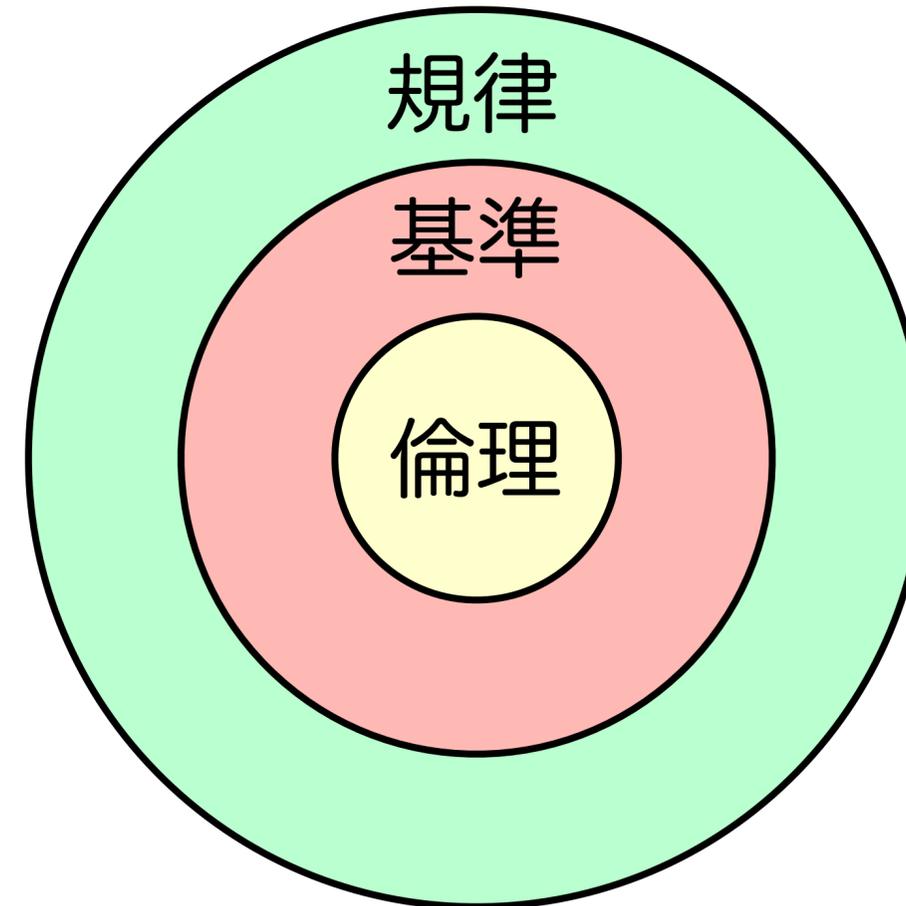
**Martin Fowler**

- ◆ AGILE
- ◆ EXTREME PROGRAMMING
- ◆ PROCESS THEORY

Daniel Terhorst-North's recent blog post on software craftsmanship has unleashed a lot of blog discussions (which I summarize below, if you're interested). There's a lot in there, but one of his themes particularly resonated with me, hence this post.

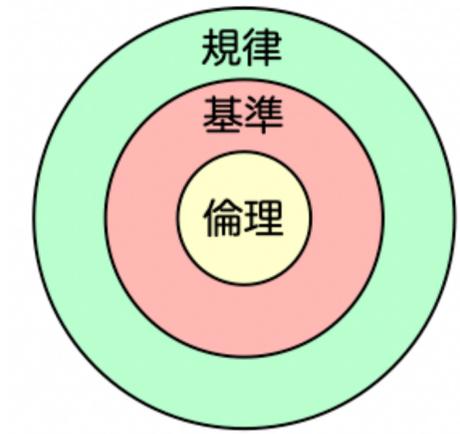
<https://martinfowler.com/bliki/CraftmanshipAndTheCrevasse.html>  
日本語訳：<https://bliki-ja.github.io/CraftmanshipAndTheCrevasse/>

# それでも、クラフトマンシップ (2回目)



※同心円はClean Architectureのマネ (本には登場しない)

# 規律、基準、倫理



## ▶ 規律

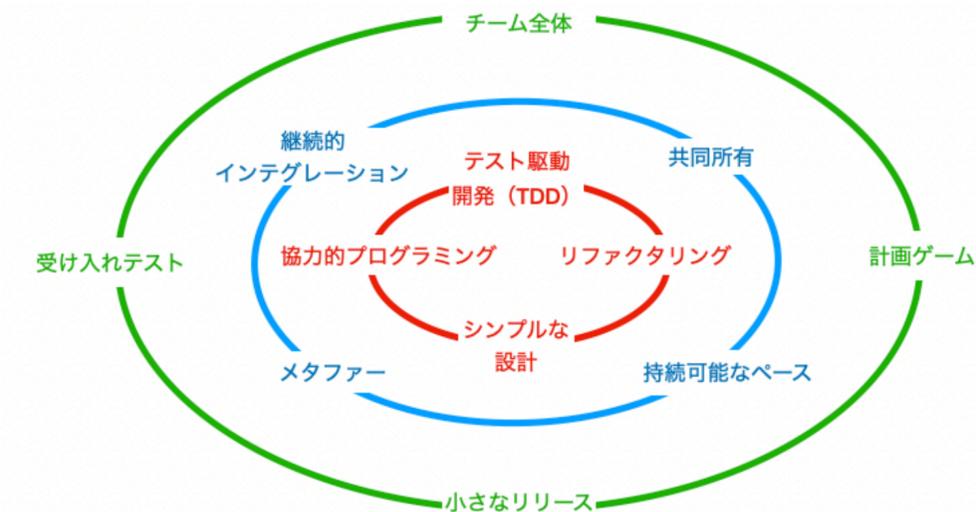
- テスト駆動開発、リファクタリング、シンプルな設計、協力的プログラミング（ペア + モブ）、受け入れテスト

## ▶ 基準

- 生産性、品質、勇気

## ▶ 倫理

- プログラマーの誓い（十戒）
  - 3つのテーマ（有害、誠実、チームワーク）



# なお、最後の本になる (らしい)

ETHICS OF BEING A PROGRAMMER

goto;  
book club



UNCLE BOB  
Author of "Clean Code"  
and many more books

ALLEN HOLUB  
Author of "Holub on Patterns"  
and many more books

Clean Code  
A Handbook of Agile Software Craftsmanship  
Robert C. Martin

たぶん私の最後の本になるね。  
Clojureの本なら書くかもしれないけど

SUBSCRIBE

<https://www.youtube.com/watch?v=QnmRpHFoYLk>

## 2. ソフトウェアクラフトマンシップ

# クラフトマンシップの定義

- ▶ 「クラフトマン」とは、特定の分野に関する高度なスキルを持ち、物事を成し遂げる人である。道具や業界に精通しており、仕事に誇りを持ち、仕事に対する尊厳とプロ意識を持って行動できると信頼されている人である。



『Clean Craftsmanship』 Bob C. Martin

# クラフトマンシップの定義（その他）

- ▶ ソフトウェア開発者のコーディングスキルを重視するアプローチ。開発者の説明責任よりも金銭的な問題を優先させるなど、ソフトウェア業界の悪弊に対するソフトウェア開発者の反応である。（Wikipedia）
- ▶ ソフトウェアクラフトマンシップは、熟練への長い旅です。ソフトウェア開発者が自らのキャリアに責任を持ち、常に新しいツールやテクニックを学び、自分を高めていくことを選択するマインドセットです。ソフトウェアクラフトマンシップは、ソフトウェア開発に責任、プロフェッショナリズム、プラグマティズム、そしてプライドを取り戻すものです。（Sandro Mancuso 『The Software Craftsman』）

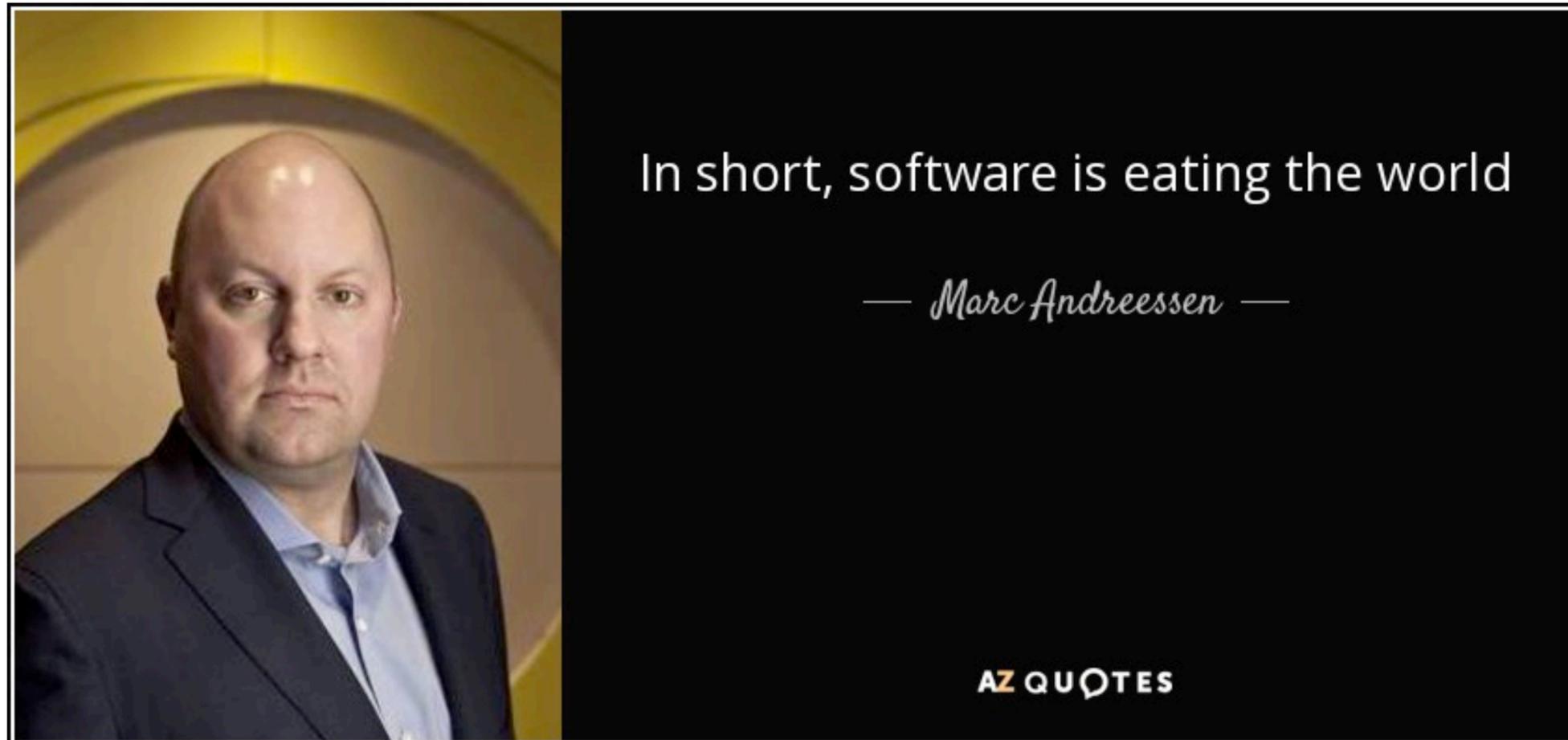
👉 「個人」としてのプロ意識

# ソフトウェアが世界を支配する時代 (2011)

社会はまだそのことを理解していない。

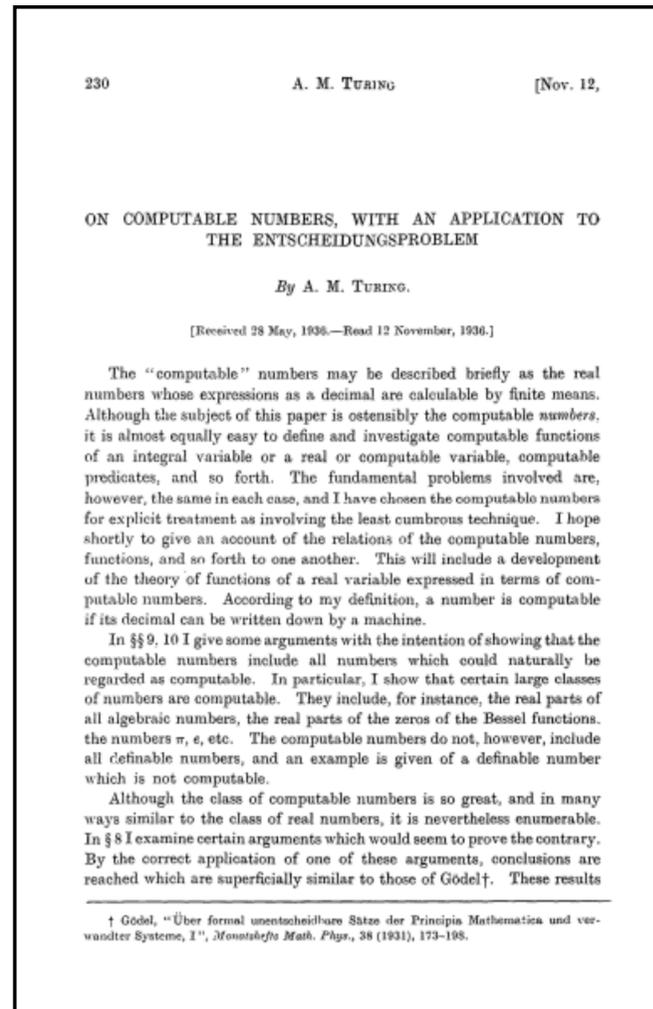
我々プログラマーもそのことを理解していない。

『Clean Craftsmanship』 Bob C. Martin

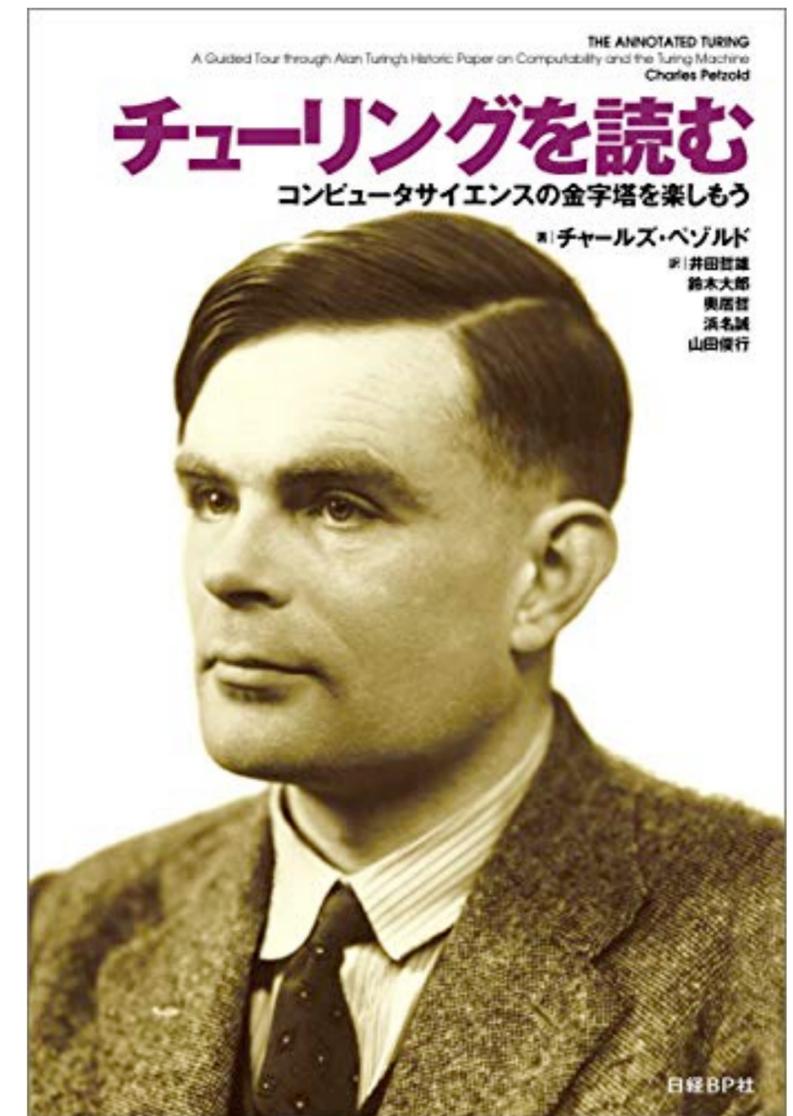
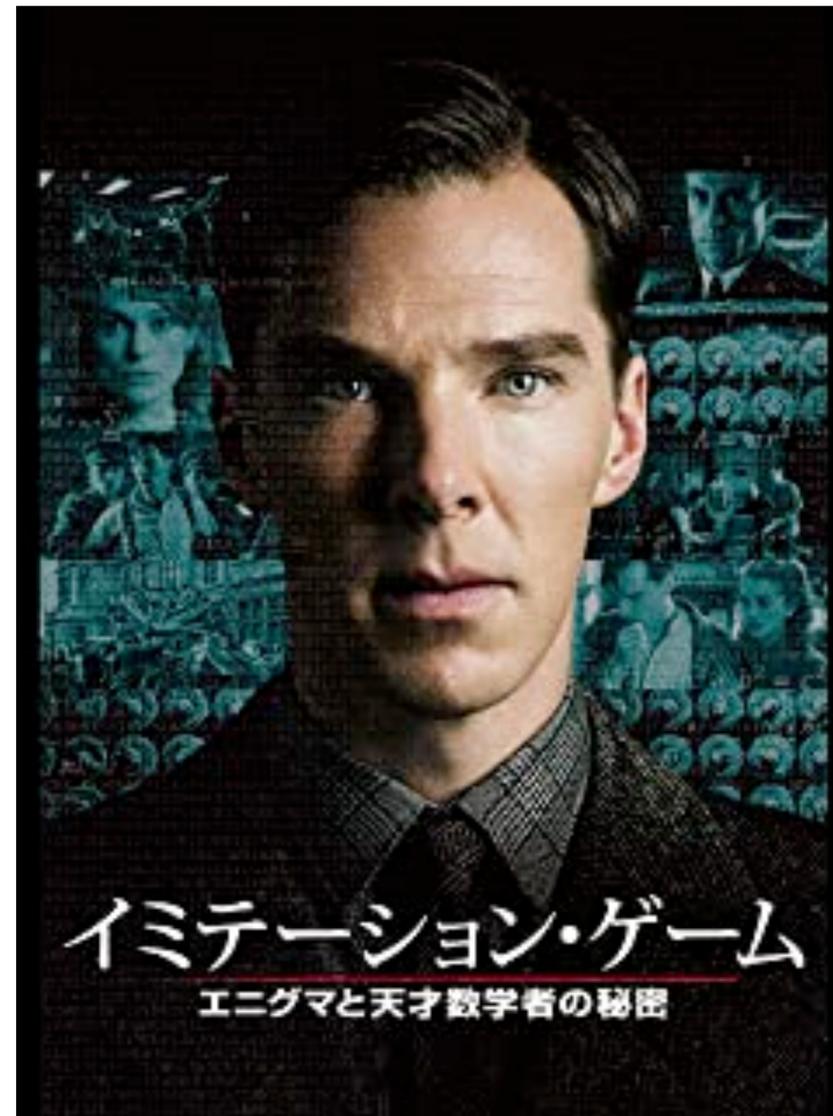


# 「ソフトウェア」の誕生

決定問題 (1936)、ACE (1945)



「計算可能数とその決定問題への応用」 (1936)

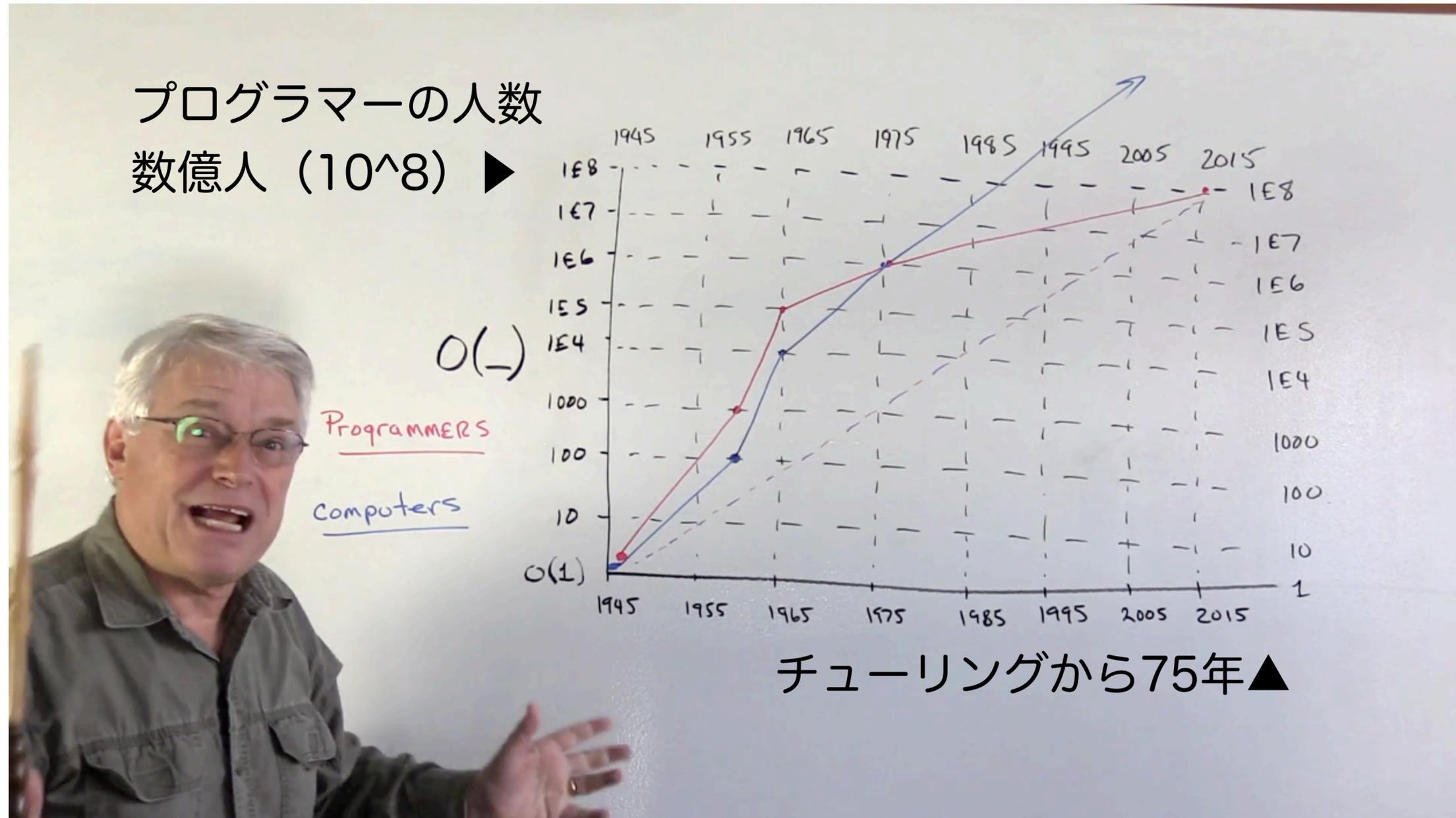


これからは（問題を）計算可能な形にする  
**能力のある数学者**がかなりの人数必要になるはずだ。  
困難となるのは、我々が何をしているのかを  
見失わないために、適切な**規律**を維持することである。

## —アラン・チューリング

A.M. Turing's ACE Report of 1946 and Other Papers - Vol. 10,  
"In the Charles Babbage Institute Reprint Series for the History of Computing", (B.E. Carpenter, B.W. Doran, eds.).  
The MIT Press, 1986.

# 能力や規律を持たないプログラマーの増加



$$\log_2 10^8 \approx 27$$
$$75 \div 27 \approx 2.8$$



約3年で  
プログラマーが倍!?



いつも半数が

# 駆け出しエンジニア

Clean Coder (Clean Coders Video Series) by Robert C. Martin



SYSTEM FAILURE

その後、またビジョンが変わった。1999年の映画『マトリックス』では、主人公はプログラマーであり、世界の「救世主」だった。実際、彼の神々しいパワーは「コード」を読み解く能力に由来するものだった。『Clean Craftsmanship』 Bob C. Martin

# 世界で最も影響力のある100人 (2013)

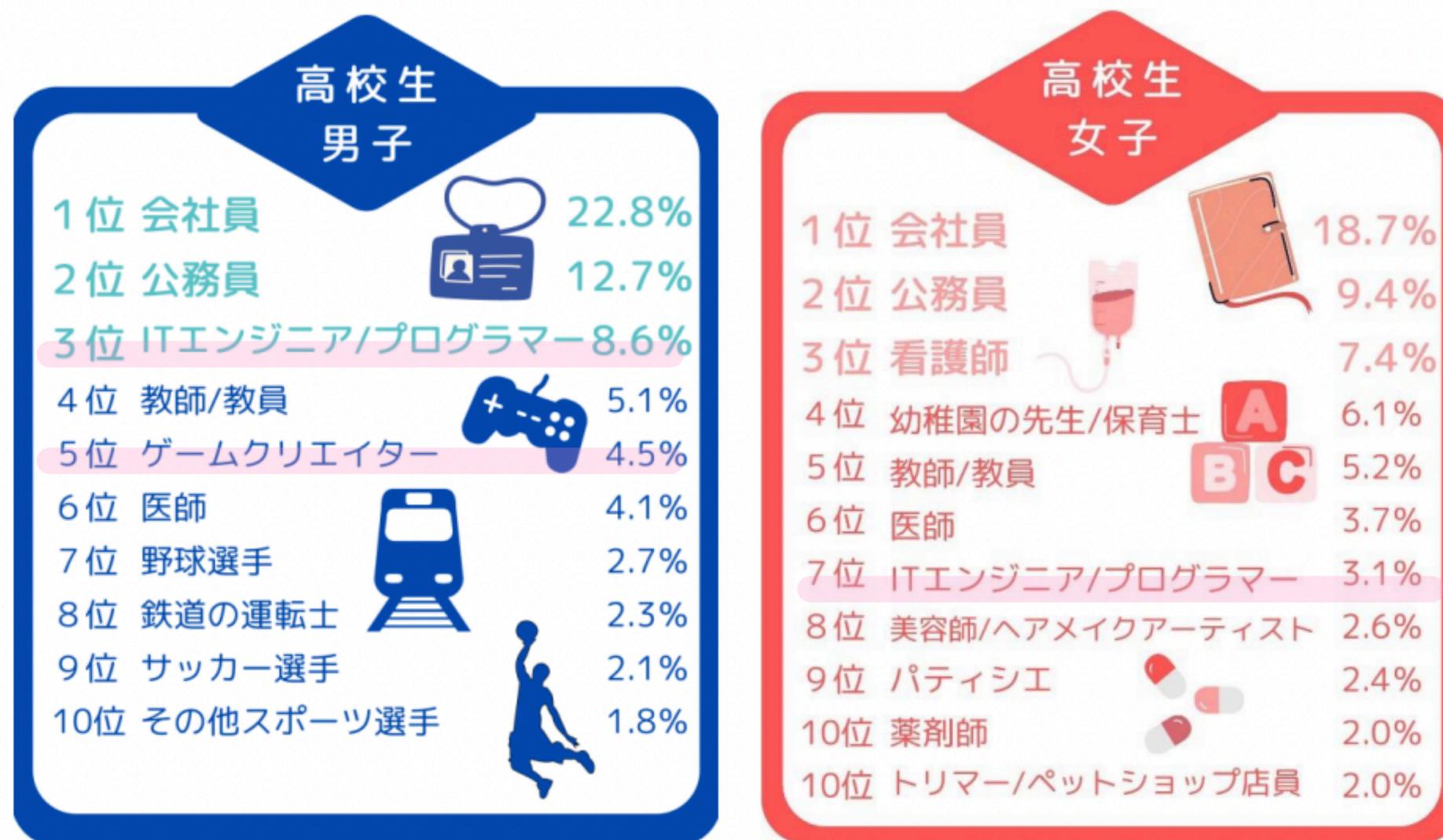
Notch and Jeb (Minecraft)



<https://time100.time.com/2013/04/18/time-100/slide/markus-persson-and-jens-bergensten/>

補足：他にも リーナス（2003）、パンジー創業者（2005）、ヴィタリック・ブテリン（2021）など

# 高校生の「大人になったらなりたいもの」 (2022)

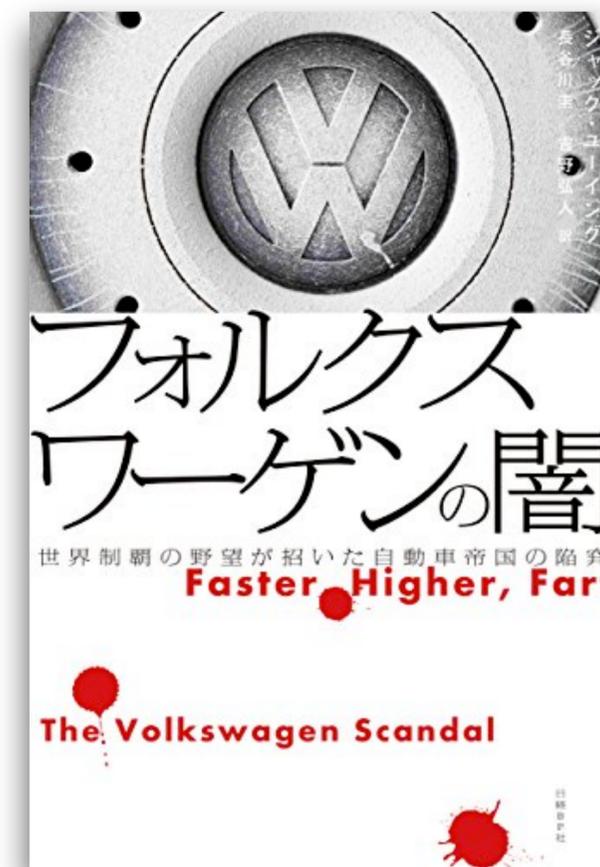


“リモートワークの普及が加速したことで、  
働く場所を選ばない自由な働き方やワークライフバランスについて意識が高まっている”

出典：第一生命／第33回「大人になったらなりたいもの」調査結果

# 規律を持たなかったことによる問題

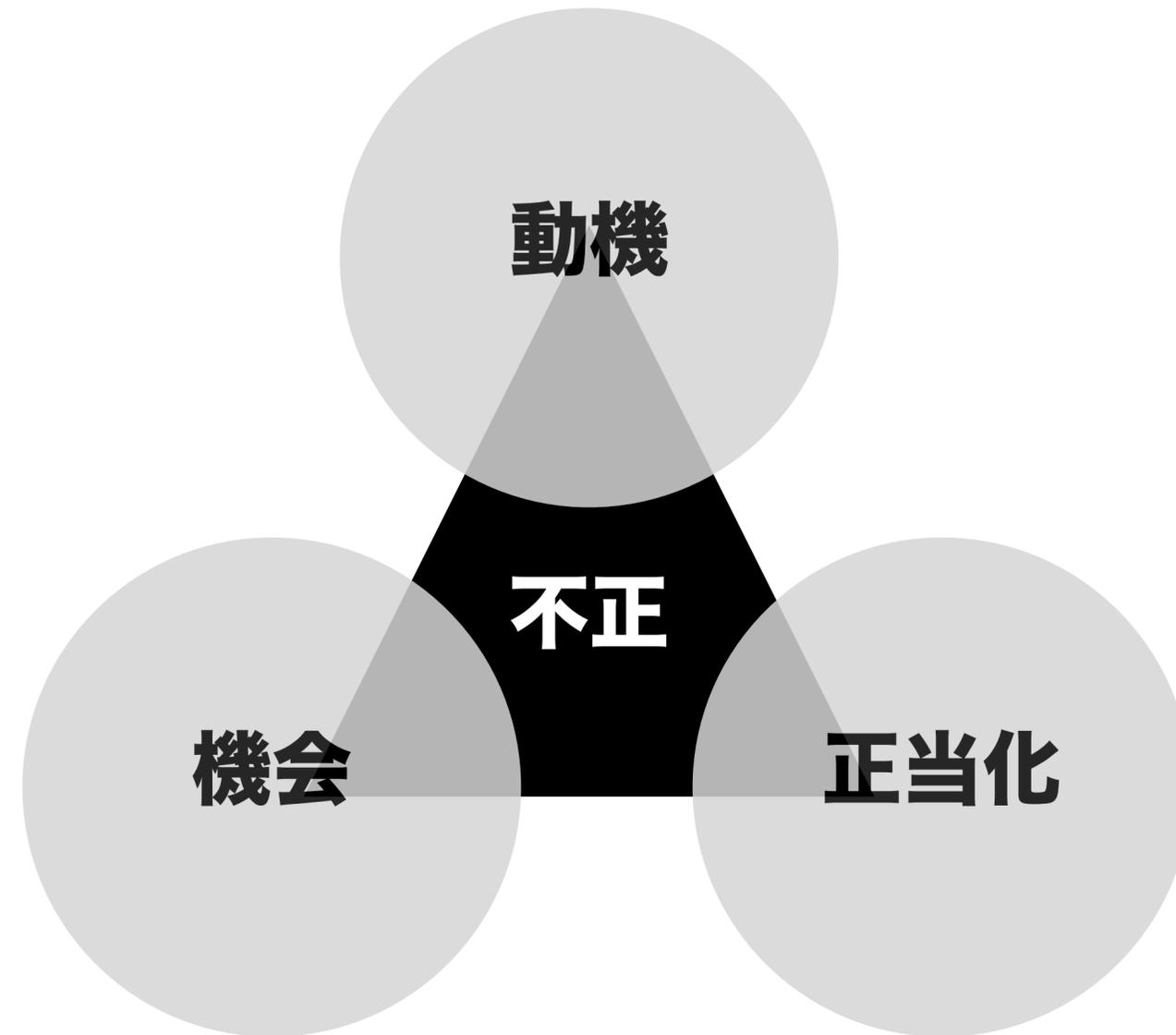
- ▶ VW社が競合他社に比べて簡素な排ガス対策装置で対応できていたことに、かねて疑問を抱いていたからだ。疑問の答えは、優れた技術にあったわけではなく、「いかさま」だった。排ガス対策装置の作動を弱める「デフィート・ソフト」を使い、公式試験のときにだけ、「ジェッタ」などに搭載する排気量2.0Lのディーゼルエンジンの排ガス成分が規制値内に収まるようにした。通常の走行時は、規制値を大幅に超える有害な排ガス物質を垂れ流す。



<https://xtech.nikkei.com/dm/atcl/news/16/080808697/>

# クレッシーの「不正のトライアングル」

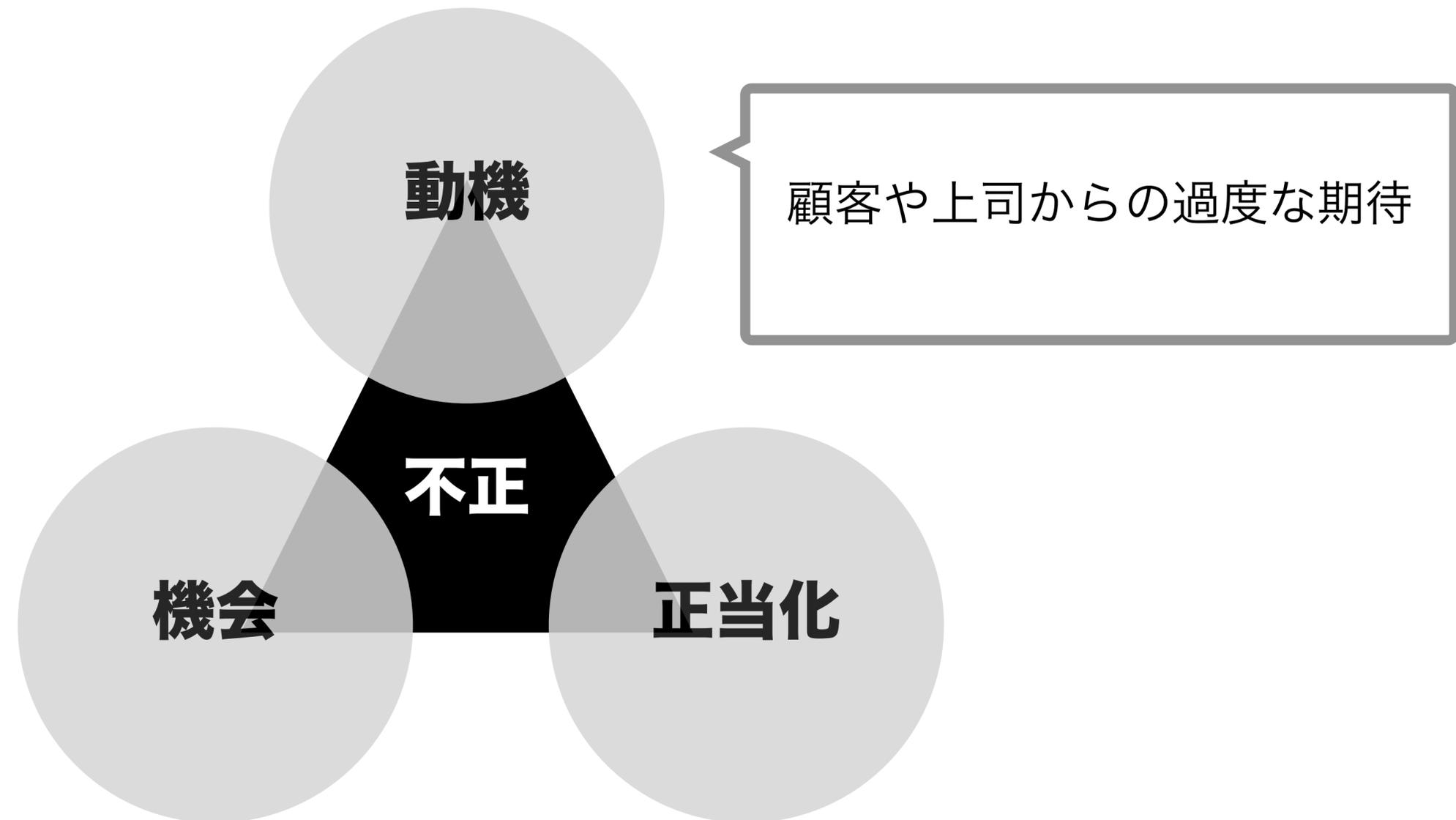
神戸製鋼品質不正問題(\*)の分析例



飯野 大介, 小松原 明哲, 2BI-2 不正のトライアングルによる組織的不正の評価について,  
人間工学, 2019, 55 巻, Supplement 号, p. 2BI-2, 公開日 2019/07/31, Online ISSN 1884-2844, Print ISSN 0549-4974,

# クレッシーの「不正のトライアングル」

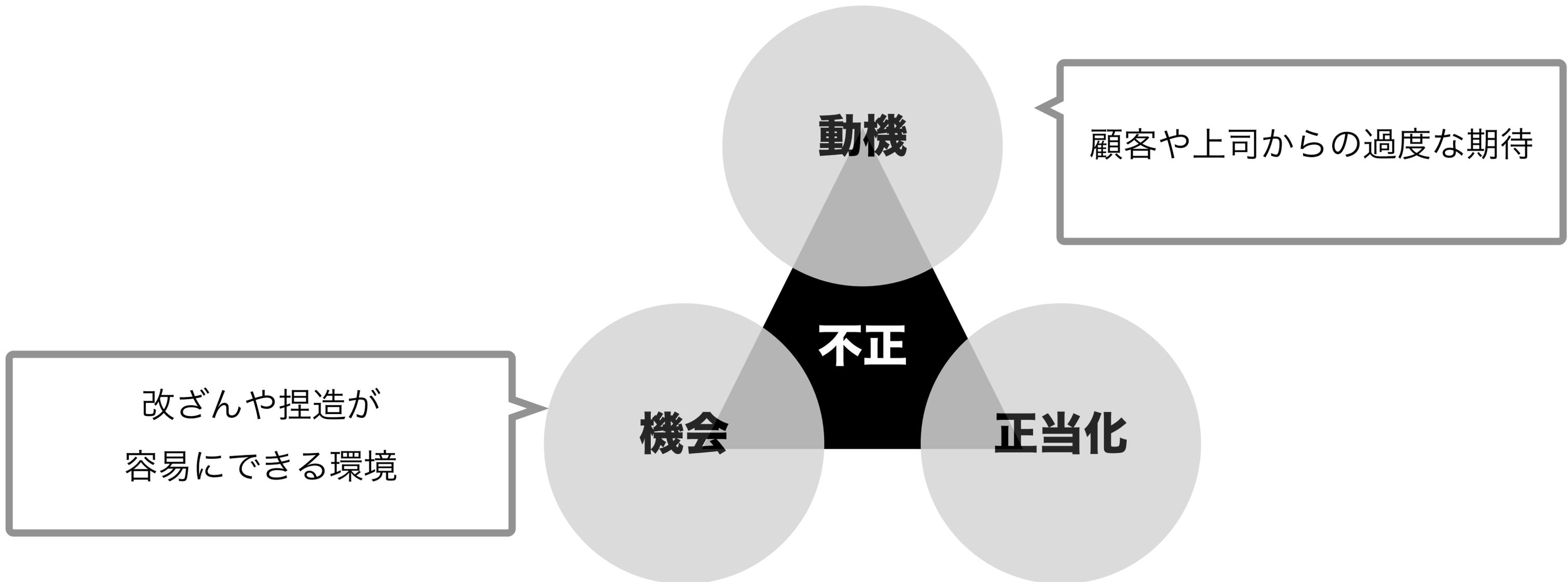
## 神戸製鋼品質不正問題(\*)の分析例



飯野 大介, 小松原 明哲, 2BI-2 不正のトライアングルによる組織的不正の評価について,  
人間工学, 2019, 55 巻, Supplement 号, p. 2BI-2, 公開日 2019/07/31, Online ISSN 1884-2844, Print ISSN 0549-4974,

# クレッシーの「不正のトライアングル」

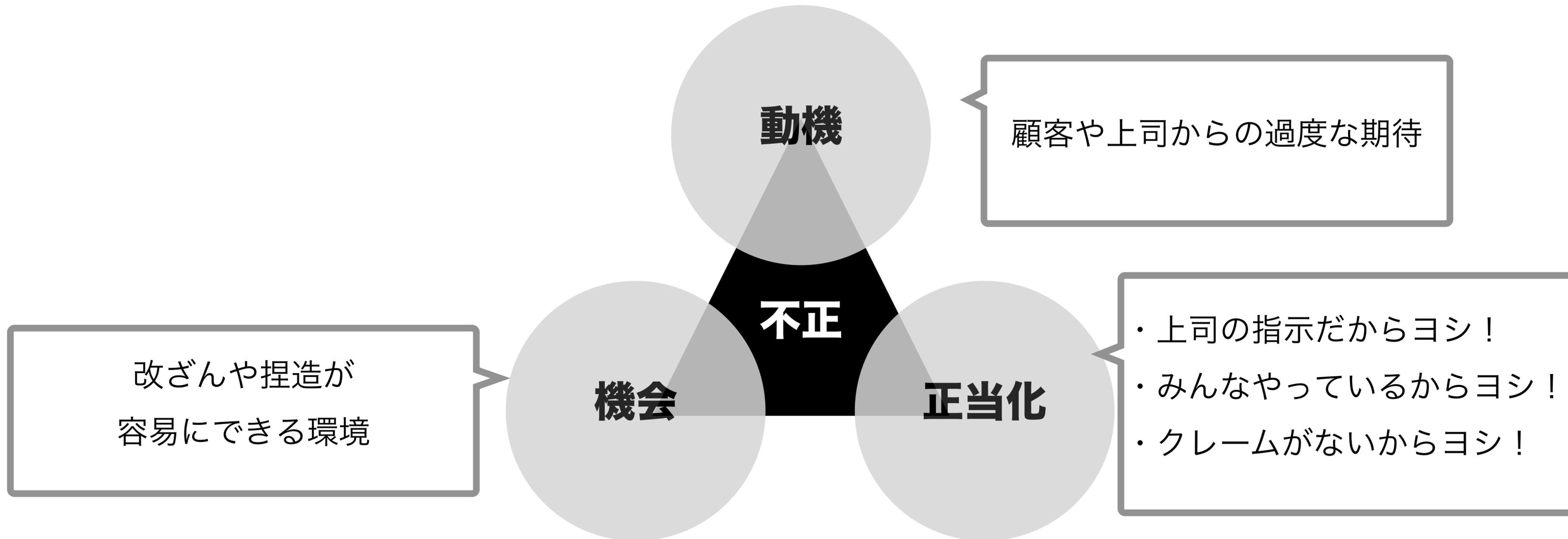
## 神戸製鋼品質不正問題(\*)の分析例



飯野 大介, 小松原 明哲, 2BI-2 不正のトライアングルによる組織的不正の評価について,  
人間工学, 2019, 55 巻, Supplement 号, p. 2BI-2, 公開日 2019/07/31, Online ISSN 1884-2844, Print ISSN 0549-4974,

# クレッシーの「不正のトライアングル」

## 神戸製鋼品質不正問題(\*)の分析例



飯野 大介, 小松原 明哲, 2BI-2 不正のトライアングルによる組織的不正の評価について,  
人間工学, 2019, 55 巻, Supplement 号, p. 2BI-2, 公開日 2019/07/31, Online ISSN 1884-2844, Print ISSN 0549-4974,



Ethan Mollick ✓

@emollick



Kant was right?

The slippery slope of ethical lapses is very real, this paper shows. People who are given the opportunity to engage in small ethical compromises are twice as likely to escalate to major unethical lapses. Holding to absolute ethical standards helps prevent a slide

## The Slippery Slope: How Small Ethical Transgressions Pave the Way for Larger Future Transgressions

David T. Welsh  
University of Washington

Lisa D. Ordóñez  
University of Arizona

Deirdre G. Snyder and Michael S. Christian  
University of North Carolina at Chapel Hill

Many recent corporate scandals have been described as resulting from a slippery slope in which a series of small infractions gradually increased over time (e.g., McLean & Elkind, 2003). However, behavioral ethics research has rarely considered how unethical behavior unfolds over time. In this study, we draw on theories of self-regulation to examine whether individuals engage in a slippery slope of increasingly unethical behavior. First, we extend Bandura's (1991, 1999) social-cognitive theory by demonstrating how the mechanism of moral disengagement can reduce ethicality over a series of gradually increasing indiscretions. Second, we draw from recent research connecting regulatory focus theory and behavioral ethics (Gino & Margolis, 2011) to demonstrate that inducing a prevention focus moderates this mediated relationship by reducing one's propensity to slide down the slippery slope. We find support for the developed model across 4 multiround studies.

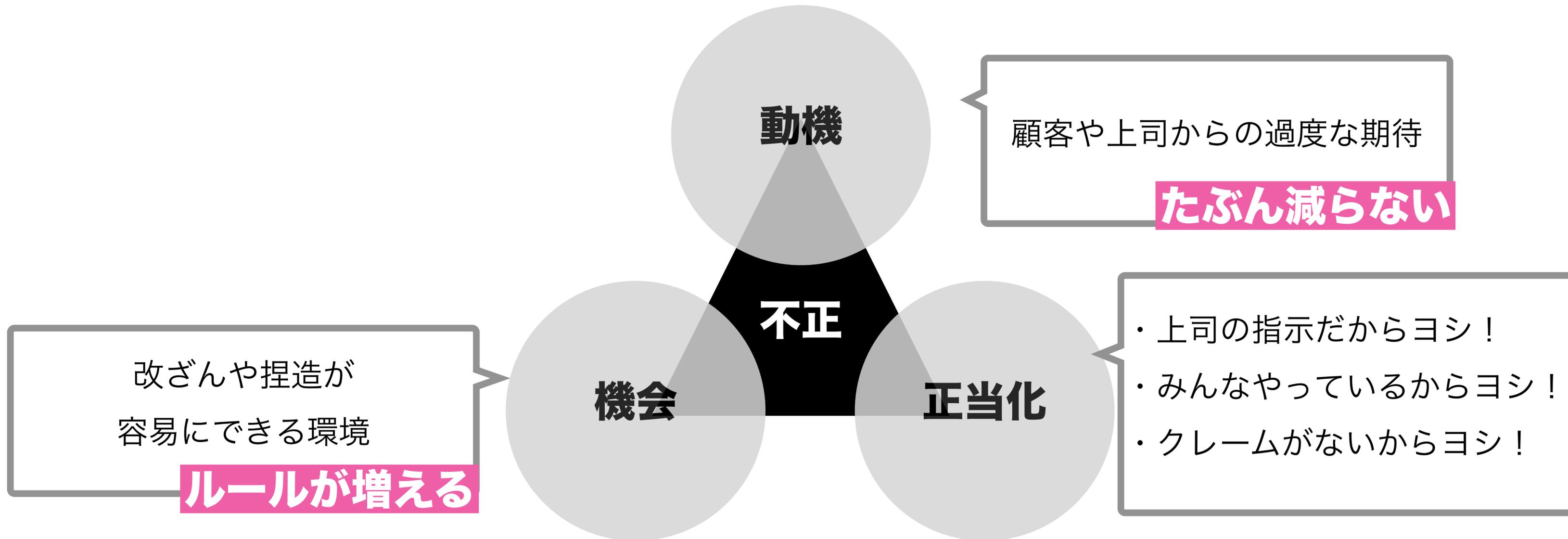
*Keywords:* behavioral ethics, self-regulation, moral disengagement, regulatory focus

“倫理的な「滑り坂論法」は、現実に存在する。小さな倫理的妥協をする機会を与えられた人は、大きな倫理的過ちへとエスカレートする可能性が2倍もある。絶対的な倫理観を持つことが、滑り防止につながる。”

<https://twitter.com/emollick/status/1582954628076871681> (via <https://twitter.com/hiyori13/status/1583004042120220675>)

# クレッシーの「不正のトライアングル」

## 神戸製鋼品質不正問題(\*)の分析例

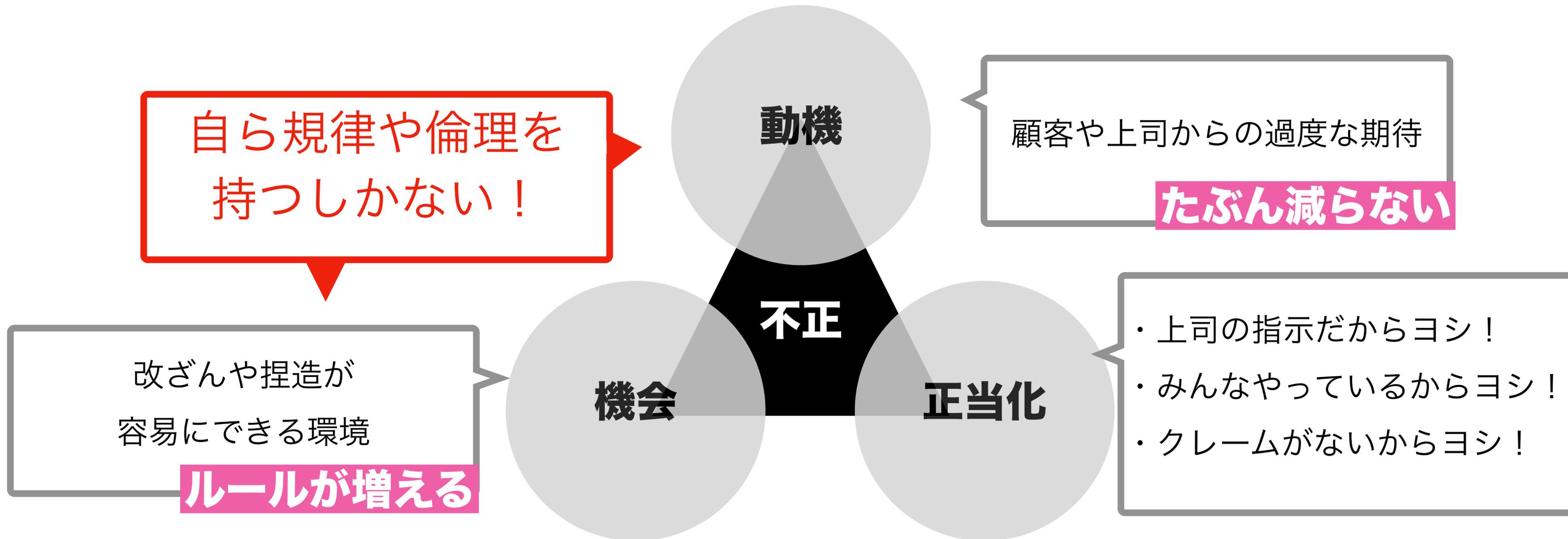


飯野 大介, 小松原 明哲, 2BI-2 不正のトライアングルによる組織的不正の評価について,

人間工学, 2019, 55 巻, Supplement 号, p. 2BI-2, 公開日 2019/07/31, Online ISSN 1884-2844, Print ISSN 0549-4974,

# クレッシーの「不正のトライアングル」

## 神戸製鋼品質不正問題(\*)の分析例



飯野 大介, 小松原 明哲, 2BI-2 不正のトライアングルによる組織的不正の評価について,

人間工学, 2019, 55 巻, Supplement 号, p. 2BI-2, 公開日 2019/07/31, Online ISSN 1884-2844, Print ISSN 0549-4974,

# 倫理を持つエンジニアが必要



## エシカル・エンジニア

ー 技術の外側へのエクスペローラー ー

帝京大学工学部客員教授  
**本澤養樹** Yasuki Motozawa

1982年東京工業大学工学部生産機械工学科卒業, 2008年博士(医学)(獨協医科大学). 1982年(株)本田技術研究所, 2014年より東京都市大学工学部客員教授, 2015年より滋賀医科大学社会医学講座客員教授, 2017年(株)本田技術研究所退職, 2017年より帝京大学工学部客員教授. 2009年より2017年まで自動車技術会研究倫理委員会委員(前身の委員会含む), 2016年より自動車工学基礎講座講師, 2019年より技術者のための倫理育成検討ワーキンググループリーダー.



会誌自動車技術4月号「エシカル・エンジニアー技術の外側へのエクスペローラー」

<https://www.jsae.or.jp/ikusei/rinri.php>

# 倫理を持つエンジニアの証 「鉄の指輪」



“鉄の指輪は、カナダで教育を受けた多くのエンジニアが、その職業に伴う義務や倫理を象徴し思い出させるものとして身につけている指輪である”

# クリーンコーダーの証 「グリーンバンド」

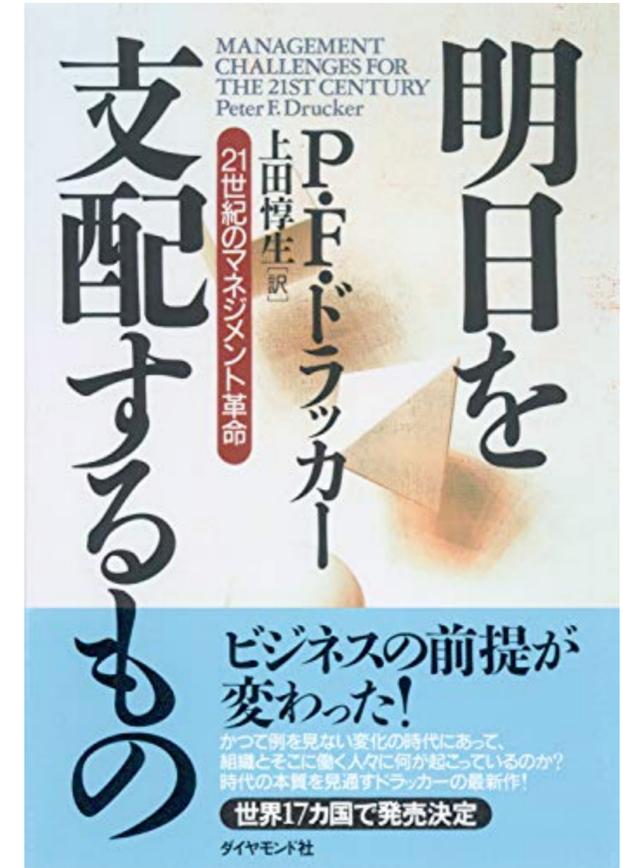
テストがパスした「グリーン」を意味する



<http://butunclebob.com/ArticleS.UncleBob.GreenWristBand>

# ミラー・テストで判断する

- ▶ 倫理についての原則はただ一つだけである。  
判断の方法は簡単である。  
それをミラー・テストという。  
倫理の問題とは、  
朝、髭を剃るとき、あるいは口紅を塗るとき、  
どのような顔を見たいかというだけの問題である。



# ミラー・テストで判断する

▶ 仕事から家に帰ったとき、鏡に映った自分を見て

「今日はいい仕事をした」と言うだろうか？

それとも、すぐにシャワーを浴びるだろうか？



プログラマーが自分で正しさを決めるだけでは不十分である。これからは、規律、基準、倫理が必要になるはずだ。我々は、それらをプログラマーである自分たちで定義すべきなのか、それとも知らない誰かに強制されるべきなのかを決めなければならない。

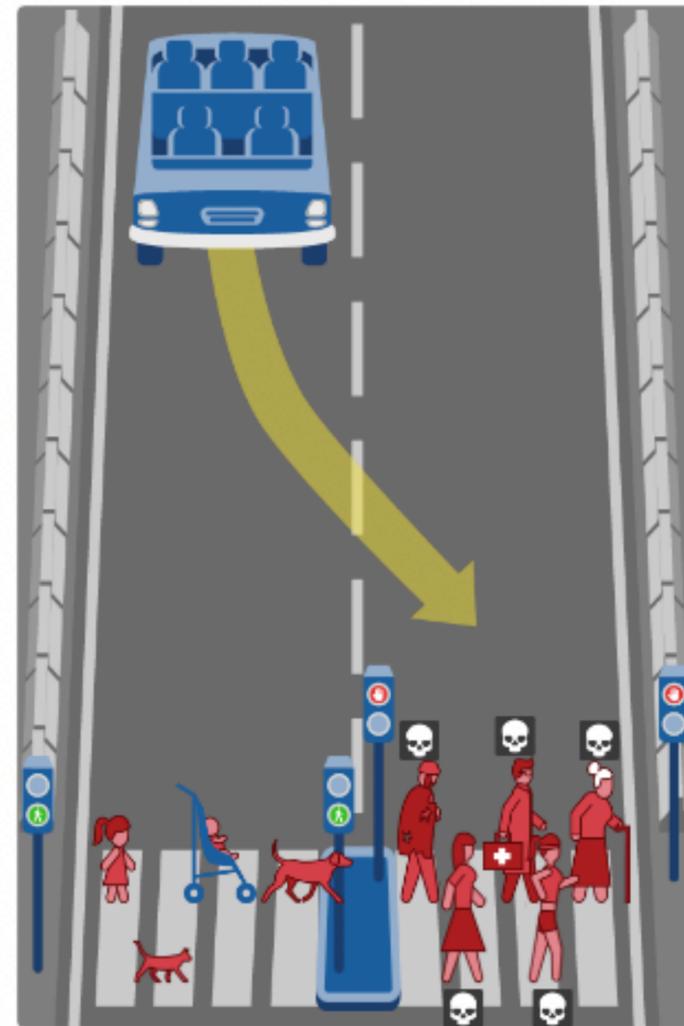
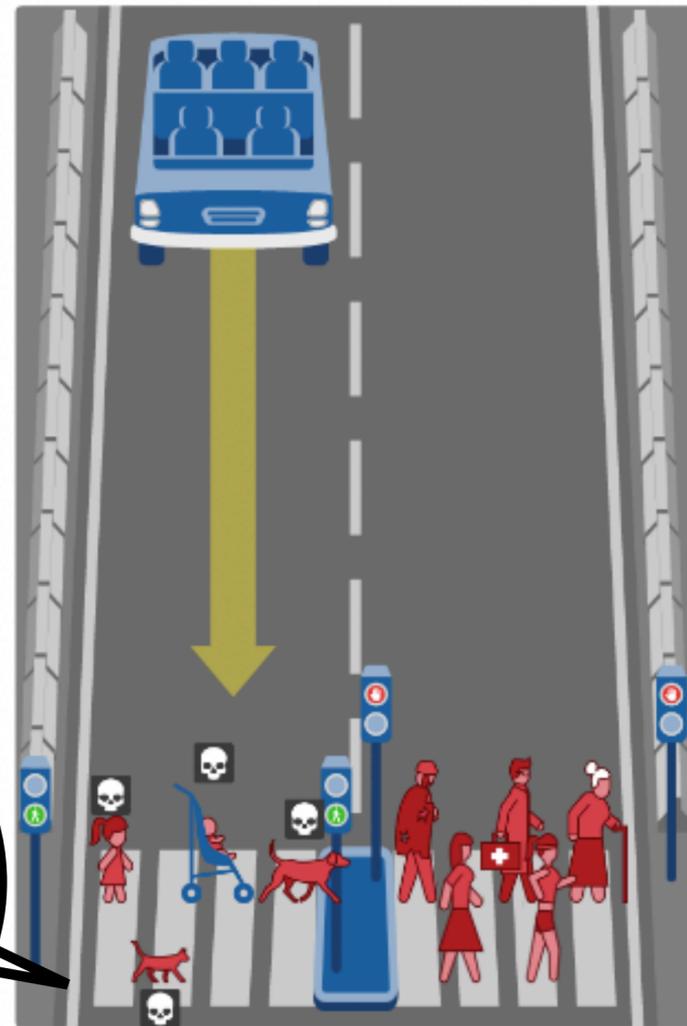
👉 「集団」 としてのプロ意識

# 3. プログラマーの誓い

**「プロ意識」は重要**  
**...とはいえ、難しい💧💧**

# あたらしい倫理的な問題

自動運転車はどうすべき？ (by MITメディアラボ)



<https://wired.jp/2019/01/02/moral-machine/>  
<https://www.moralmachine.net>

# 改めて「3つの正義」を考えてみる

## ▶ 幸福の最大化（帰結主義）：ビジネス

- ・ > 価値があればええやる、動けばええやる（結果重視）

## ▶ 自由の尊重（義務論）：アジャイル

- ・ > みんなが自律的に動けばええやる（プロセス重視）

## ▶ 美徳の促進（徳倫理学）：クラフトマンシップ

- ・ > より良いエンジニアになるべきやる（性質重視）
- ・ > アリストテレスを読み



# ニコマコス倫理学

**NHKテキスト**  
2022年 5月 ●EPL  
月曜日—午後 10:25—10:50  
(西・北地区)—午後 05:30—05:55  
金曜日—午後 01:05—01:30

100分  
de  
名著

人生の究極目的を問う

アリストテレス  
*Aristotle's*

ニコマコス  
*Ethica Nicomachea*

倫理学

山本芳久  
*Yamamoto Yukihisa*

二千年の時を超え、  
私たちが照らす不朽の古典。

人はいかに生きるべきか？  
単純にして劇的な問いを徹底的に考え、  
「幸福」を明確に定義してみせた西洋哲学史の金字塔、  
古代ギリシア最高の知性による慧眼を、学問に解きほぐす。

ファスト教養  
10分で答えが欲しい人たち

レジー  
regis

教養=ビジネスの  
役に立つ!?

最上の人になるために  
古典文学を読む

本書の内容はYouTubeでチェック  
お金を掛けない勉強はムズ

集英社新書

# アリストテレスの徳（卓越性）

## ▶ 思考の徳（知的徳）

- ・ エピステーメー、テクネ、**フロネーシス**など

## ▶ 性格の徳（倫理的徳）

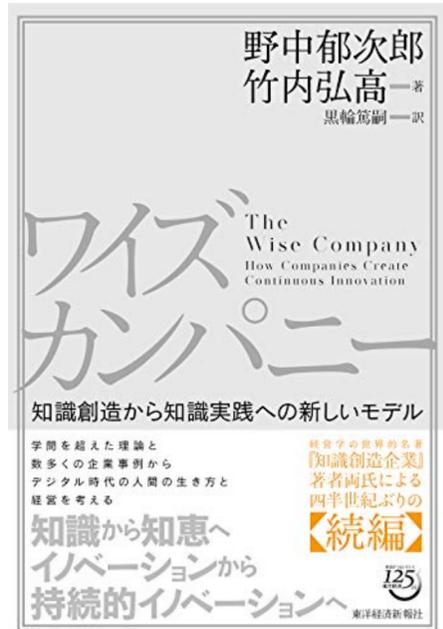
- ・ 勇気、節制、友愛、正義など



# 「実践知」としてのフロネシス

エピステーメーが「なぜを知る」知識、  
テクネーが「いかにを知る」知識だとすれば、  
フロネシスは「何をなすべきかを知る」知識といえる。

—野中郁次郎・竹内弘高『ワイズカンパニー』



それぞれ、科学的知識、工学的知識、実践的知識と言い換えてもいいかも？

# 徳（卓越性）を高めるには？

- ▶ "上手に家を建てることから人はすぐれた建築家になり、下手に建てることから劣悪な建築家になる"
- ▶ "要するに一言でいえば、同じような活動の反復から、人の性格の状態が生まれるのである"

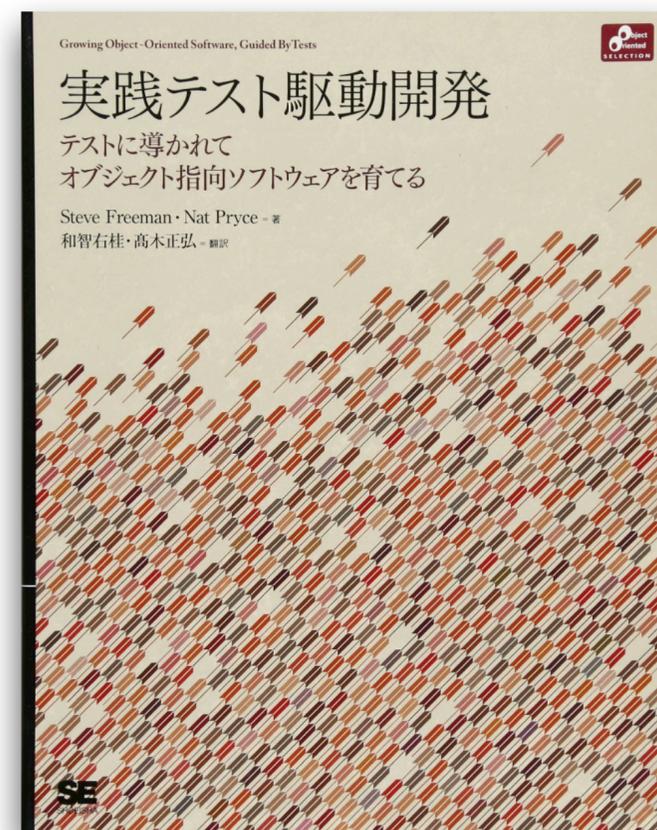


# TDDの“習慣”がプロを作る

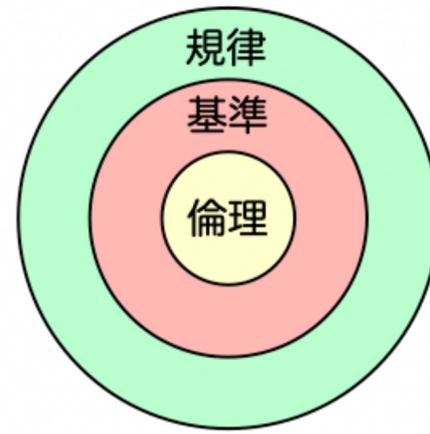
TDDを実践していなければ、プロのソフトウェア開発者になれない。

私は本気だ。というより、それが真実になりつつある。

『Clean Craftsmanship』 Bob C. Martin



# プログラマーの徳（卓越性）



## ▶ 規律

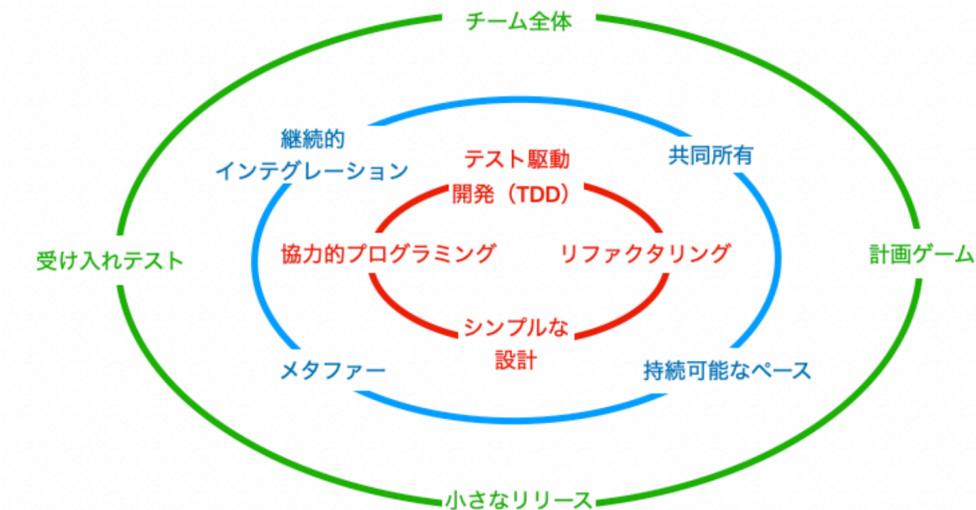
- テスト駆動開発、リファクタリング、シンプルな設計、協力的プログラミング（ペア + モブ）、受け入れテスト

## ▶ 基準

- 生産性、品質、勇気

## ▶ 倫理

- プログラマーの誓い（十戒）
  - 3つのテーマ（有害、誠実、チームワーク）

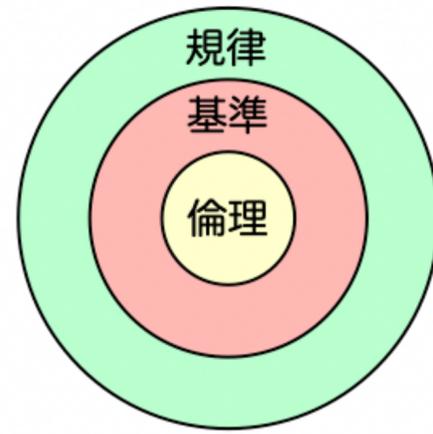


# 余談：TDDは万能ではない

- ▶ Jim Coplien：ボトムアップで優れたアーキテクチャは考えられない
  - <https://www.infoq.com/interviews/coplien-martin-tdd/>
  - <https://rbc-us.com/documents/Why-Most-Unit-Testing-is-Waste.pdf>
- ▶ Rich Hickey：テストはセーフティネットにすぎない
  - <https://www.infoq.com/presentations/Simple-Made-Easy/>
- ▶ DHH：テスト容易性のために、システムを分断する設計にする必要はない
  - <https://dhh.dk/2014/tdd-is-dead-long-live-testing.html>
- ▶ Kent Beck：グリーンテストは好きだが（複雑なシステムでは）幻想で自己満足でもある
  - <https://youtu.be/l3f9Vxxs1S4?t=691>

<http://blog.cleancoder.com/uncle-bob/2014/05/02/ProfessionalismAndTDD.html>

# プログラマーの徳（卓越性）



## ▶ 規律

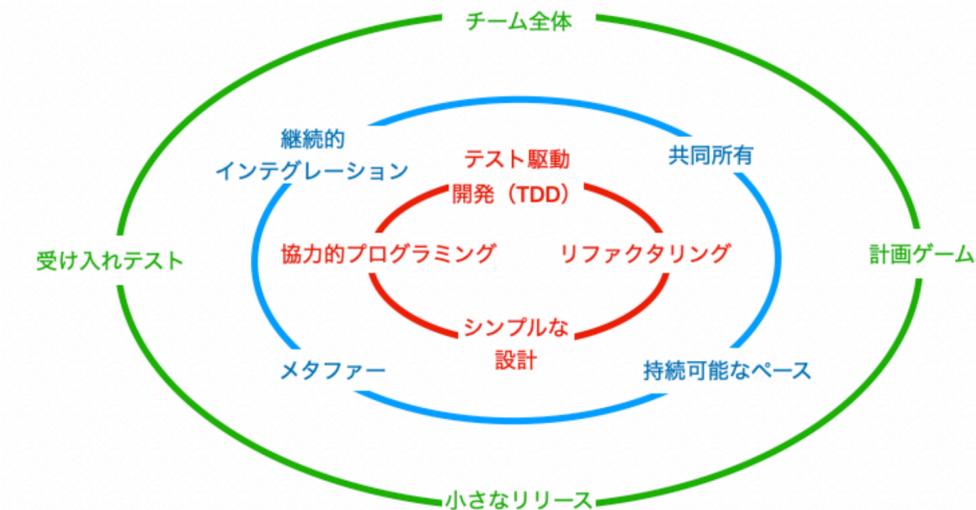
- テスト駆動開発、リファクタリング、シンプルな設計、協力的プログラミング（ペア + モブ）、受け入れテスト

## ▶ 基準

- 生産性、品質、勇気

## ▶ 倫理

- プログラマーの誓い（十戒）
- 3つのテーマ（有害、誠実、チームワーク）



# 医師の倫理 「ヒポクラテスの誓い」

表1 ヒポクラテスの誓い

医神アポロン、アスクレピオス、ヒュギエイア、パナケイア、およびすべての男神・女神たちの御照覧をおおぎ、つぎの誓いと師弟契約書の履行を、私は自分の能力と判断の及ぶかぎり全うすることを誓います。

この術を私に授けていただいた先生に対するときは、両親に対すると同様にし、共同生活者となり、何かが必要であれば私のものを分け、また先生の子息たちは兄弟同様に扱い、彼らが学習することを望むならば、報酬も師弟契約書もとることなく教えます。また医師の心得、講義そのほかすべての学習事項を伝授する対象は、私の息子と、先生の息子と、医師の掟てに従い師弟誓約書を書き誓いを立てた門下生に限ることにし、彼ら以外の誰にも伝授はいたしません。

養生治療を施すに当たっては、能力と判断の及ぶ限り患者の利益になることを考え、危害を加えたり不正を行う目的で治療することはいたしません。

また求められても、致死薬を与えることはせず、そういう助言も致しません。同様に婦人に対し墮胎用のペッサリーを与えることもいたしません。私の生活と術ともに清浄かつ敬虔に守りとおします。

結石の患者に対しては、決して切開手術は行わず、それを専門の業とする人に任せます。

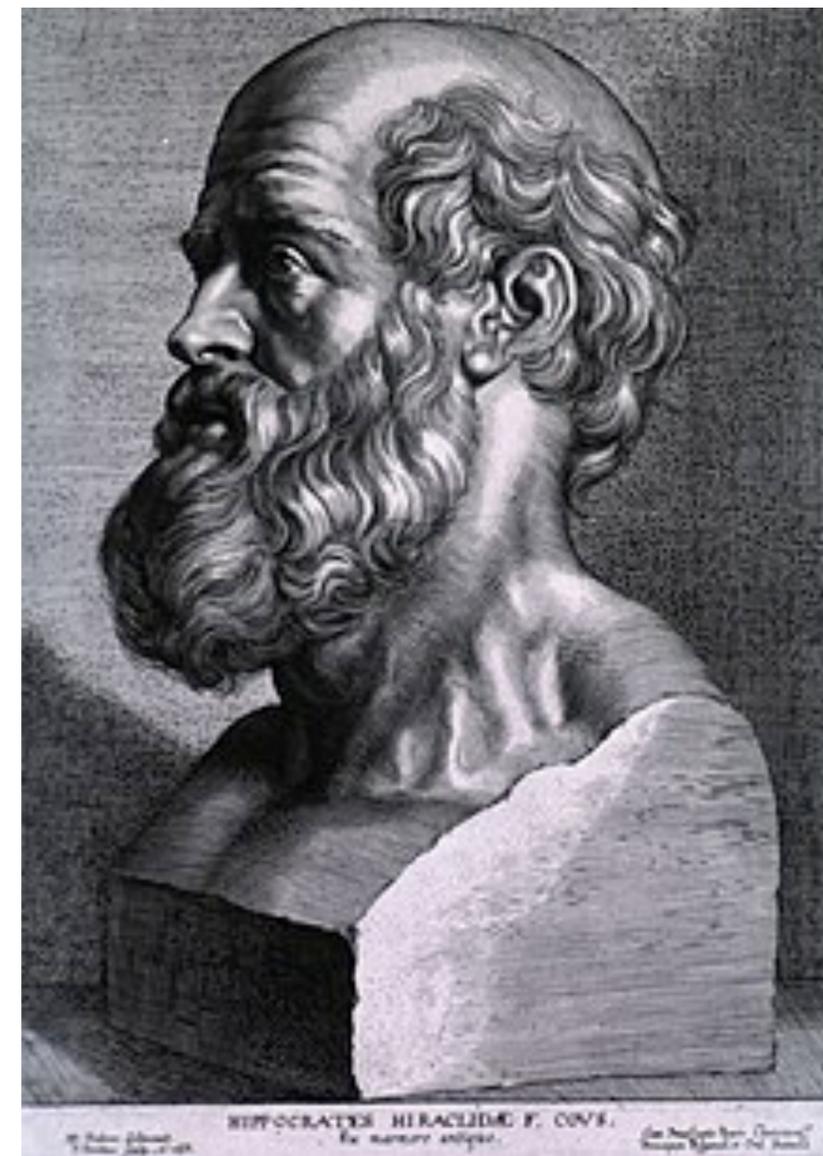
また、どの家には行って行くにせよ、すべては患者の利益になることを考え、どんな意図的不正も害悪も加えません。とくに、男と女、自由人と奴隷のいかなをとわず、彼らの肉体に対して情欲をみだすことはいたしません。

治療の時、または治療しないときも、人々の生活に関して見聞きすることで、およそ口外すべきでないものは、それを秘密事項と考え、口を閉ざすことに致します。

以上の誓いを私が全うしこれを犯すことがないならば、すべての人々から永く名声を博し、生活と術のうえでの実りが得られますように。しかし誓いから道を踏み外し偽誓などをするのであれば、逆の報いをうけますように。

(大槻マミ太郎訳：誓い、小川鼎三編、ヒポクラテス全集、第1巻、エンタプライズ、東京、1985;580-582より引用)

[https://www.med.or.jp/doctor/rinri/i\\_rinri/a06.html](https://www.med.or.jp/doctor/rinri/i_rinri/a06.html)

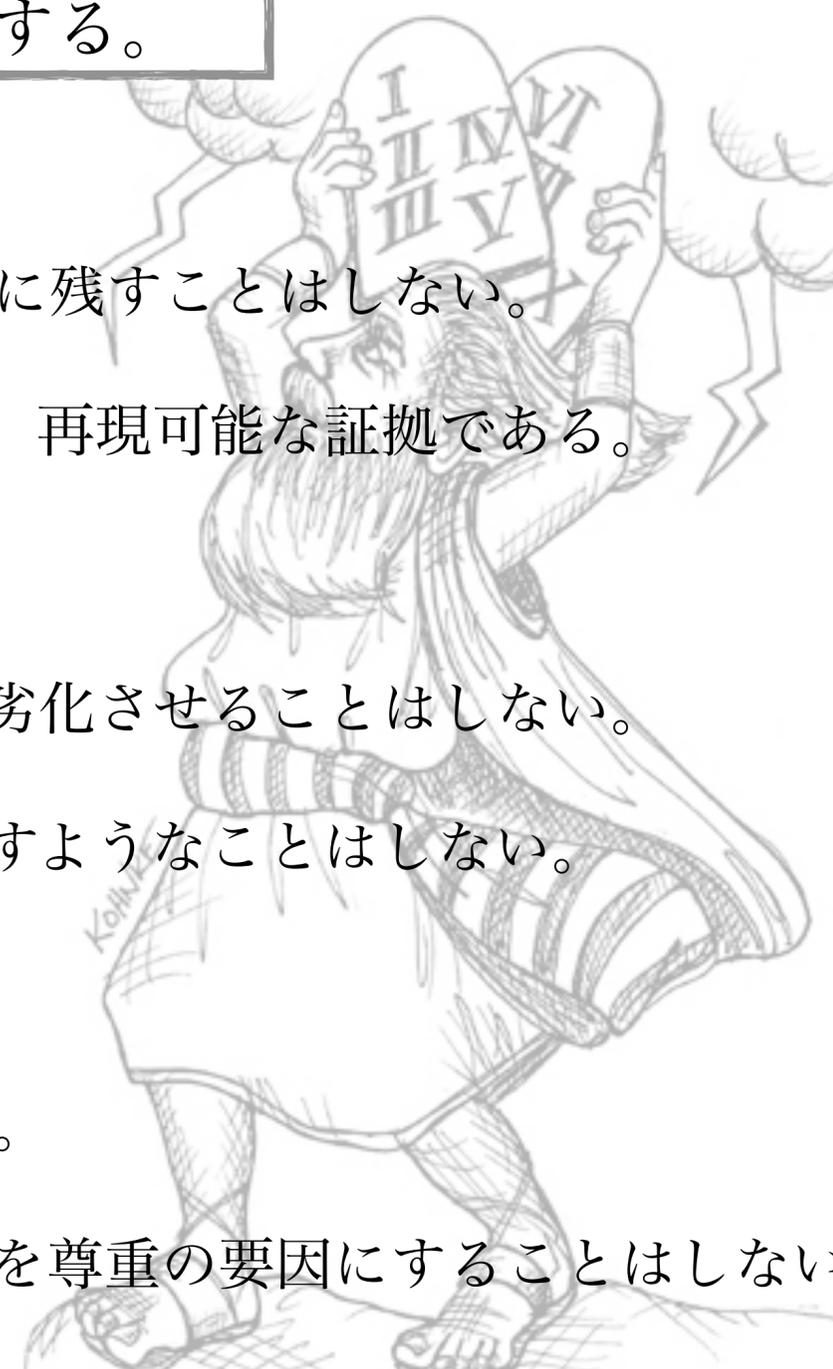


出典：Wikipedia（CC表示-継承 4.0）

# プログラマーの誓い

コンピュータープログラマーの職業の名誉を守り、維持するために、私は自分の能力と判断の限りにおいて、以下のことを約束する。

1. 私は、有害なコードを作らない。
2. 私が作るコードは、常に私の最高傑作である。振る舞いや構造に欠陥のあるコードを故意に残すことはしない。
3. 私は、コードが正常に動作する証拠をリリースごとに用意する。それは、迅速で、確実に、再現可能な証拠である。
4. 私は、誰かの進捗を妨げないように、小さく何度もリリースする。
5. 私は、あらゆる機会において、恐れることなく執拗に私の作品を改善する。決して作品を劣化させることはしない。
6. 私は、私や誰かの生産性を高めるために、できる限りのことをする。決して生産性を落とすようなことはしない。
7. 私は、他の人が私をカバーできるように、私が他の人をカバーできるように努める。
8. 私は、規模と精度の両方を正直に見積もる。合理的な確実性がないときには約束をしない。
9. 私は、仲間のプログラマーの倫理、基準、規律、スキルを尊重する。その他の属性や特性を尊重の要因にすることはしない。
10. 私は、私の技術の学習と向上を怠らない。



**が、似たものがないわけではない**

# ソフトウェア・エンジニアリングのための 倫理ならびに専門職実務綱領 (ACM/IEEE-CS)

## ▶ The Software Engineering Code of Ethics and Professional Practice

- <https://ethics.acm.org/code-of-ethics/software-engineering-code/>

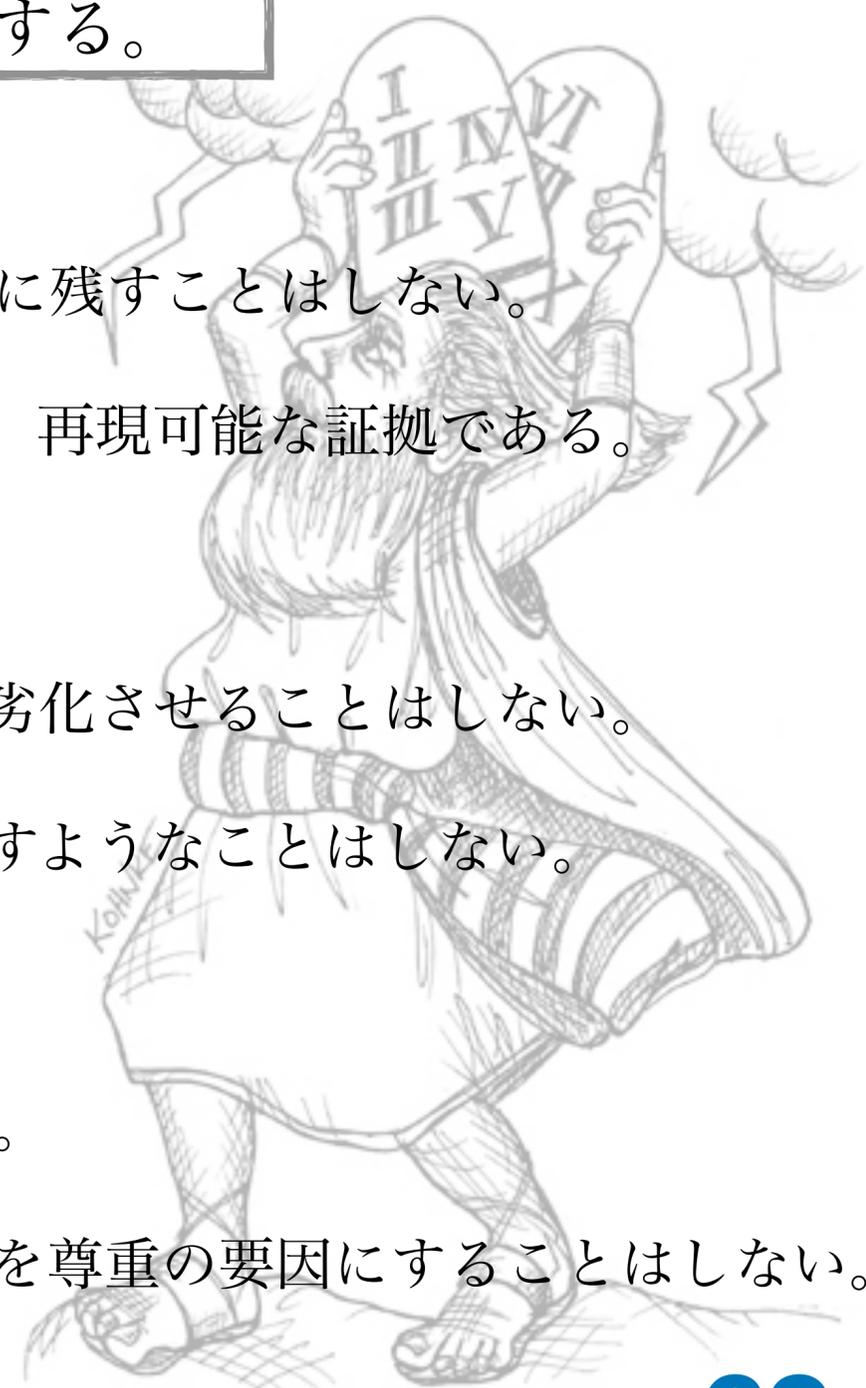
(日の丸アイコンをクリックすると日本語のPDFが閲覧可能)

- 1 公共性-ソフトウェア・エンジニアは公共の利益と調和するよう行動すべきである。
- 2 顧客ならびに雇用者-ソフトウェア・エンジニアは、公共の利益と調和しながら、顧客と雇用者の最高の利益を実現するよう行動すべきである。
- 3 製品-ソフトウェア・エンジニアは、その製品と、製品に関する変更が、専門家として可能な限り最高の基準に合致していることを確保すべきである。
- 4 判断-ソフトウェア・エンジニアは専門家としての判断において誠実さと独立性を維持すべきである。
- 5 管理-ソフトウェア・エンジニアリングの管理者とリーダーは、ソフトウェア開発の管理ならびにソフトウェア・メンテナンスの管理に対する倫理的アプローチに賛同し、それを推し進めるべきである。
- 6 専門職-ソフトウェア・エンジニアは公共の利益と調和するよう、その専門職の倫理性と評判を高めていくべきである。
- 7 職業上の同僚-ソフトウェア・エンジニアは他のソフトウェア・エンジニアに対して公正で協力的であるべきである。
- 8 自己の向上-ソフトウェア・エンジニアは自己の専門職実務に関する生涯続く学習に参加し、かつその専門職実務に対する倫理的アプローチを推し進めるべきである。

# 「クリーン」プログラマーの誓い

コンピュータープログラマーの職業の名誉を守り、維持するために、私は自分の能力と判断の限りにおいて、以下のことを約束する。

1. 私は、有害なコードを作らない。
2. 私が作るコードは、常に私の最高傑作である。振る舞いや構造に欠陥のあるコードを故意に残すことはしない。
3. 私は、コードが正常に動作する証拠をリリースごとに用意する。それは、迅速で、確実に、再現可能な証拠である。
4. 私は、誰かの進捗を妨げないように、小さく何度もリリースする。
5. 私は、あらゆる機会において、恐れることなく執拗に私の作品を改善する。決して作品を劣化させることはしない。
6. 私は、私や誰かの生産性を高めるために、できる限りのことをする。決して生産性を落とすようなことはしない。
7. 私は、他の人が私をカバーできるように、私が他の人をカバーできるように努める。
8. 私は、規模と精度の両方を正直に見積もる。合理的な確実性がないときには約束をしない。
9. 私は、仲間のプログラマーの倫理、基準、規律、スキルを尊重する。その他の属性や特性を尊重の要因にすることはしない。
10. 私は、私の技術の学習と向上を怠らない。



有害 🍆

# 約束1：有害なコードを作らない

- ▶ フォルクスワーゲンのプログラマーが破ったのはこのルールだ。
- ▶ 彼らのソフトウェアは雇用主（フォルクスワーゲン）に利益をもたらしたかもしれない。だが、社会一般には害をもたらした。
- ▶ 我々プログラマーは、そのようなことを決してやってはならない。

『Clean Craftsmanship』 Bob C. Martin

# 約束2：欠陥のあるコードを故意に残さない

- ▶ コードを動かすことはゴールではない
- ▶ 振る舞いは緊急なだけ。構造こそが重要。
  - → アイゼンハワーのマトリックス
- ▶ コードの構造に責任を持つのはプログラマーである
- ▶ 構造を変更できてこそ「ソフト」ウェアである

重要 緊急	重要 緊急ではない	
重要ではない 緊急	重要ではない 緊急ではない	

図 12-1 アイゼンハワーの決定マトリックス

『Clean Craftsmanship』 Bob C. Martin

# 約束3：コードが正常に動作する証拠を用意する

- ▶ コードを数学的に証明することはできない（あるいは難しい）
- ▶ でも、実証的な証拠なら残すことができる
- ▶ それは、テストコードである

『Clean Craftsmanship』 Bob C. Martin

誠実 🙏

# 約束4：小さく何度もリリースする

- ▶ 継続的にデプロイしろ

『Clean Craftsmanship』 Bob C. Martin

# 約束5：執拗に作品を改善する

- ▶ 約束5には「作品」という言葉が使われている。プログラマーが作成するのはコードだけではない。設計、ドキュメント、スケジュール、計画なども作成する。これらも継続的に改善すべき「作品」である。
- ▶ どうするのか？ 作品に対して「親切な行為」をすればいい。
- ▶ ボーイスカウトの規則：  
「チェックインするコードはチェックアウトしたときよりも美しく」

『Clean Craftsmanship』 Bob C. Martin

# 約束6：生産性を高める

- ▶ 速く進みたければ、うまく進むしかない
  - 自動化しよう
  - 集中力を高めよう
  - 時間管理をうまくやろう（ポモドーロテクニックおすすめ）

『Clean Craftsmanship』 Bob C. Martin

# チームワーク

# 約束7：他の人が私をカバーできるように、 私が他の人をカバーできるように努める。

- ▶ チーム全体に知識を広げよう。
- ▶ 知識を広げる最善の方法は、一緒に（ペアかモブで）働くことである。
- ▶ 車を運転しているときに他のドライバーを怒鳴りつけたことはないだろうか？  
これは「フロントガラス効果」と呼ばれる。フロントガラスを挟むと、他人のことをバカ、マヌケ、敵だと見なすようになる。
- ▶ フロントガラス効果を回避するには、1年に数回は物理的な部屋に集まる。

『Clean Craftsmanship』 Bob C. Martin

# 約束8：正直に見積もる

- ▶ 見積りは確率分布（幅）なので、以下のように考えてみてはどうか？
  - 平均値： $(2 \times \text{最有力} + (\text{最良} + \text{最悪}) / 2) / 3$
  - シグマ： $(\text{最良} - \text{最悪}) / 6$
- ▶ 無理なものは無理と言おう
- ▶ 「とりあえずやってみます」には気を付けよう

『Clean Craftsmanship』 Bob C. Martin

# 約束9：仲間を尊重する

- ▶ そのまま！
- ▶ 人を属性で判断しない（性別、人種、宗教など）
- ▶ 「規律、基準、倫理」を持っているかどうかで判断すべき
  - ・ ただし、能力や規律を持たない人が入ってきてても歓迎してあげましょう
  - ・ むしろ、そういう人たちを育てる責任が私たちにはあります

『Clean Craftsmanship』 Bob C. Martin

# 約束10：技術の学習と向上を怠らない

- ▶ プログラマーは学習をやめてはならない。学ぶべき領域は事実上無限だ。我々の業界は過去何十年かけて急速に変化している。これからもその変化はしばらく続くだろう。あなたはそれに追いつく必要がある。
- ▶ 書籍やブログを読み続けよう。ビデオを観続けよう。カンファレンスやユーザーグループに参加し続けよう。研修に行き続けよう。学習を続けよう。
- ▶ 過去の偉大な作品に注目しよう。1960年代、1970年代、1980年代に書かれた書籍は、素晴らしい洞察と情報があふれている。

『Clean Craftsmanship』 Bob C. Martin

まとめ

# まとめ

- ▶ アンクル・ボブはいろんな経験をしているおじいちゃん（70歳）
  - ・ 70歳で何かを伝えようとしてくれるプログラマーは少なくない？
- ▶ クラフトマンシップはプロ意識（個人 → 集団）
  - ・ まずは、家に帰って鏡で自分の顔を見てみよう
- ▶ 徳（卓越性）は日々の習慣や誓いを守ることから生じる
  - ・ 有害なことをせず、誠実な態度を示し、チームワークを重視しよう

# 読んでみてください！

