

ソフトウェアテスト 最初的一步 ～テスト設計技法を ワークで体験しながら学ぶ～

JaSST'25 Tokyo



本セッションの主催について

- WACATEというコミュニティが主催
 - ・ 企業、営利団体ではありません
 - ・ 転職、就職等のお話はありません
 - ・ 何かあれば実行委員まで



- この資料・お話する内容は、
個人の見解です。
- 所属する組織、団体の公式見解と
異なる場合があります。



1. 自己紹介
2. その前に
3. サマリー
4. なぜ？
5. いろいろなテスト設計技法
 - ・ 同値分割
 - ・ 境界値分析
 - ・ デシジョンテーブル
6. 今後の勉強に向けて・まとめ
7. WACATEの紹介

1. 自己紹介

簡単に自己紹介



- 風間 裕也（ブロッコリー）
- X(旧Twitter):@nihonbuson
- 所属
 - 株式会社10X 品質管理チーム
 - B-Testing（個人事業主）
 - WACATE（実行委員長）
 - JaSST Review（実行委員長）
 - ソフトウェアレビューを
エンジニアリングっぽく捉える会



2. ワークショップを始める前に

みなさんのことを少し教えてください



◆ どこから来ましたか？

- a. 東京都内
- b. 関東圏
- c. 東日本
- d. 西日本
- e. その他



IT系のワークショップの参加回数は？

- a. はじめて
- b. 数回
- c. 10回以上
- d. その他



WACATE(ワカテ)を知っていますか？

- a. 参加したことがある
- b. 参加を検討したことがある
- c. 名前を知っている
- d. はじめて聞いた
- e. その他



3. サマリー

本セッションで行うこと



目的：セッションでテスト設計技法の一部を説明する

1. なぜテスト設計技法を使うかを考えてもらう
2. いろいろなテスト設計技法があることを知る
3. ミニワークにより実際にやってみる





1. テスト設計技法をなぜ使うか？
2. テスト設計技法（一部）の説明・ワーク
（時間の関係からミニワーク）
3. 元資料の解説・今後の勉強に向けて・まとめ

4. なぜ？

まず、なぜテスト設計技法を使うのか考えてみましょう



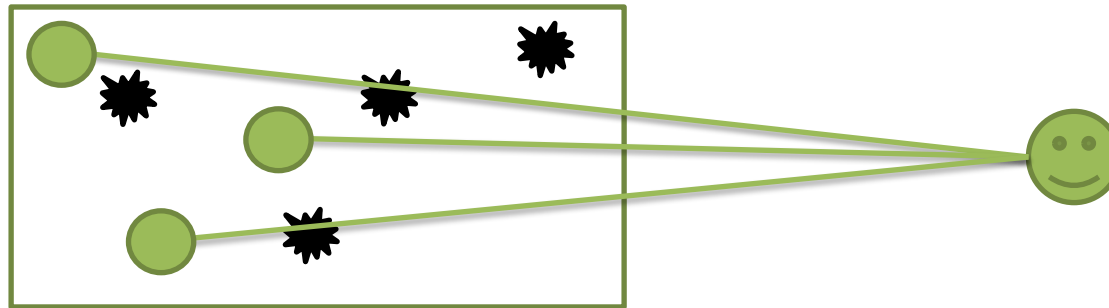
- テストに利用できる要員、機材、時間は有限
- 全ての入出力をテストするのは非現実的



現実的な手数で効率的・効果的に
バグを検出していく必要がある

Point

- テスト設計技法を使わない場合のあるある
 - たくさんテストをしてもなかなかバグが見つからない

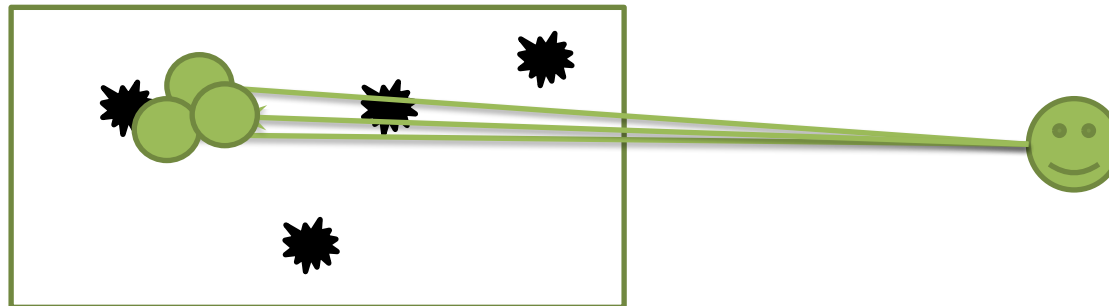


凡例

● テストした箇所

✱ バグがある箇所

- 同じところを何度もテストしている



現実的な手数で効率的・効果的に
バグを検出していく必要がある

- 現実的な手数で . . .

テストケースを減らして

- 効率的・効果的に . . .

バグがあるところを狙って

ダブらず広い範囲を

テスト設計技法の分類



● テスト実行
★ 見つかるバグ
☆ 見つからないバグ

補完



■ 効果: 多くのバグを検出できること

- どれだけバグを見逃さずに済むかが大事。左上の例だと、
 - バグ検出効果 = $4/6 = 67\%$
 - バグ検出効率 = $4/36 = 11\%$
- 理論的には、「全数テスト」は効果が最高。が、効率は低い。
- 同値分割法は、同値パーティションで全体を網羅するので効果的。

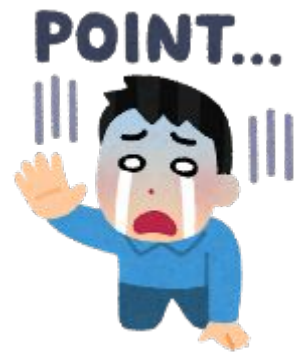
■ 効率: かけた時間に対して、検出できるバグが多いこと

- 重要なバグが出そうな場所を集中的に確認する。左下の例だと、
 - バグ検出効果 = $3/6 = 50\%$
 - バグ検出効率 = $3/13 = 23\%$
- テスト開始直後に欠陥を1つ見つければ効率は最高。が、効果は低い。
- 境界値分析は、バグしやすい値を叩くので効率的^(*1)。

技法単品ではなく、技法同士の違いを
考えてみると面白い。（と思う人もいる）

(*1)境界値分析は、同値分割の特性を併せ持つので、効果的でもある。

- テストケースを減らすということは、減らした分だけ確認しない箇所があるということ
- 適切にテスト設計技法を適用しないと必要なテストが漏れたり、無駄な箇所ばかりテストすることに . . .



5. いろいろなテスト設計技法

いろいろなテスト設計技法を知ろう



JSTQB/ISTQB

■仕様ベースの技法

同値分割法
境界値分析
デシジョンテーブル
原因結果グラフ法
状態遷移テスト
組み合わせテスト技法
ユースケーステスト
ユーザストーリーテスト
ドメイン分析

■欠陥ベースの技法

欠陥分類法

■経験ベースの技法

エラー推測
チェックリストベースドテスト
探索的テスト

■Structure-Based Testing

Introduction
Condition Testing
Decision Condition Testing
Modified Condition/Decision Coverage (MC/DC)
Testing

Multiple Condition Testing

Path Testing

API Testing

■Quality Characteristics for Technical Testing

Security Testing

Reliability Testing

Performance Testing

Resource Utilization

Maintainability Testing

Portability Testing

SQuBOK Guide

■経験および直観に基づいた技法

アドホックテスト

探索的テスト

■仕様に基づいた技法

ブラックボックステスト

同値分割

境界値分析／境界値テスト

デシジョンテーブルによるテスト

原因結果グラフによるテスト

状態遷移テスト

ランダムテスト

モデルベースドテスト

要因分析技法【富士通】

CFD技法

■コードに基づいた技法

ホワイトボックステスト

制御フローテスト

データフローテスト

トランザクションフローテスト

フォールトに基づいた技法

エラー推測テスト

ミューテーションテスト

■利用に基づいた技法

運用プロファイルによるテスト

ローカライゼーションテスト

ユーザー環境シミュレーションテスト

整合性確認テスト

■ソフトウェアの形態に基づいた技法

オブジェクト指向テスト

Webシステムのテスト

GUIテスト

サーバーサイドのテスト

データベーステスト

並行プログラムのテスト

プロトコル適格性テスト

実時間のテスト

モバイルアプリケーションのテスト

■組合せの技法

直行配列表(実験計画法)を用いたテスト

All-pair法を用いたテスト

■リスクに基づいた技法

テストマネジメントにおけるリスクベースドテスト

テスト設計におけるリスクベースドテスト

BS7925

EQUIVALENCE PARTITIONING
BOUNDARY VALUE ANALYSIS
STATE TRANSITION TESTING
CAUSE-EFFECT GRAPHING
SYNTAX TESTING
STATEMENT TESTING
BRANCH/DECISION TESTING
DATA FLOW TESTING
BRANCH CONDITION TESTING
BRANCH CONDITION COMBINATION TESTING
MODIFIED CONDITION DECISION TESTING
LCSAJ TESTING
RANDOM TESTING

ISO/IEC/IEEE 29119

■Specification-Based Test Design Techniques

Equivalence Partitioning
Classification Tree Method
Boundary Value Analysis
Syntax Testing
Combinatorial Test Design Techniques
Decision Table Testing
Cause-Effect Graphing
State Transition Testing
Scenario Testing
Random Testing

■Structure-Based Test Design Techniques

Statement Testing
Branch Testing
Decision Testing
Branch Condition Testing
Branch Condition Combination Testing
Modified Condition Decision Coverage (MCDC) Testing
Data Flow Testing

■Experience-Based Test Design Techniques

Error Guessing

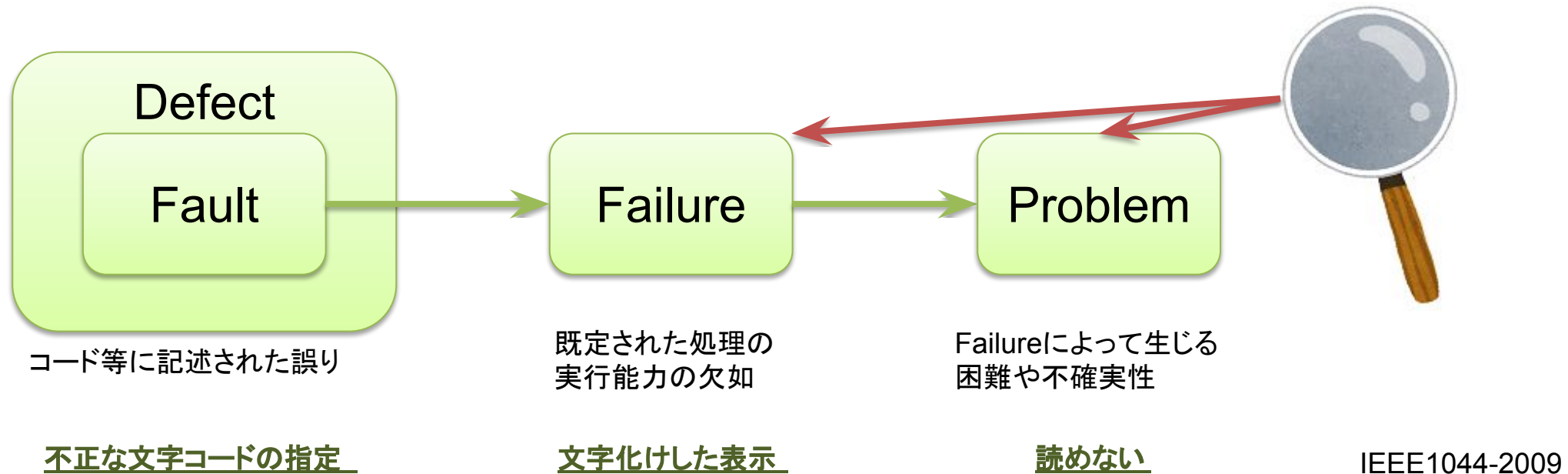
- 仕様ベースの技法から以下をご紹介します

- 同値分割
- 境界値分析
- デシジョンテーブル
- 状態遷移テスト
- ユースケーステスト
- 組み合わせテスト技法
- ドメイン分析

今回は
ココを紹介

今回は
省略

- 仕様ベースの技法では基本的に「Failure」や「Problem」を見つけますが、本セッション中は「バグ」という表現を使用しています。



- これから各技法の説明で例を提示します。
- 実在する施設・団体・商品等とは一切関係ございません。

同値分割

- 同値分割とは
 - 沢山ある入力値や出力値を【**同値パーティション**】に分けること
- 同値パーティションとは
 - システムによって**同じように扱われる**（と思われる）値のセットや範囲
- 同値パーティション内の値は、
全て同じように扱われるはずなので、
テストケースを減らすことができる
 - **同値パーティション毎に1つの値を確認すればOK**

例題

過去のJaSSTに参加したことがある場合、
リピート割が適用される。

このとき、以下のアンケート項目を同値分割しなさい。

7.過去にJaSSTに
は参加したことが
ありますか？ ✕

✓ ---いずれかひとつ選択してください---

JaSST 東京に参加したことがある。

他の地域の JaSST に参加したことがある。

JaSST 東京にも他の地域の JaSST にも参加したことがある。

今回が初めての参加である。

1. 有効パーティションと無効パーティションを見つける

有効パーティション

- JaSST東京参加済
- 他の地域JaSST参加済
- JaSST東京にも
他の地域JaSSTにも参加済

無効パーティション

- はじめての参加

2. 最もそのパーティションの特性を表す**代表値**を選ぶ 最頻値、平均値、中央値など...

有効パーティション

- **JaSST東京参加済**
- 他の地域JaSST参加済
- JaSST東京にも
他の地域JaSSTにも参加済

無効パーティション

- **はじめての参加**

3.期待結果を決定してテストケースにする

#	選択した項目（入力値）	リピート割（期待結果）
1	JaSST東京に参加したことがある。	適用される
2	今回が初めての参加である。	適用されない

- 同値分割の使いどころ
 - 基本的な動作を確認したいとき
 - 細かいテストの前にそもそもそれなりに動くのか
(**有効値**と**無効値**を1個ずつとか)
 - 異常系の細かいテストを全てパスした後で
普通の正常系でエラーになったらショック大きいですね！
 - ユーザー受入テストなど通常の使い方で問題がないか
 - ビルド後に想定外の影響がないかスモークテストとか
 - 他のテスト設計技法を適用する準備や合わせ技



- 同値分割の注意点
 - **漏れなく被りなく** 分割する必要あり
 - 「東京都」「関東圏」「東日本」の分割は不適切
 - 無効同値パーティションは**仕様に明示されない場合あり**
 - 暗黙の了解だったり、思考漏れだったり...
 - テストの目的が曖昧な場合、値だけで同じ同値パーティションか判断することはできない
 - どのような関係をテストしたいのか、も明らかにする必要がある



ミニワーク

～実際にやってみよう～



(公開資料では省略)

境界値分析

境界値分析（BVA）は、順序付けられた同値パーティションの境界上に存在する値を適切に処理していることをテストするために使用する。

引用： [ISTQBテスト技術者資格制度Advanced Level シラバス日本語版 テストアナリスト Version 3.1.1.J03](#)

境界値分析（BVA）は、**順序付け**られた
同値パーティションの**境界上**に存在する値を
適切に処理していることをテストするために使用する。

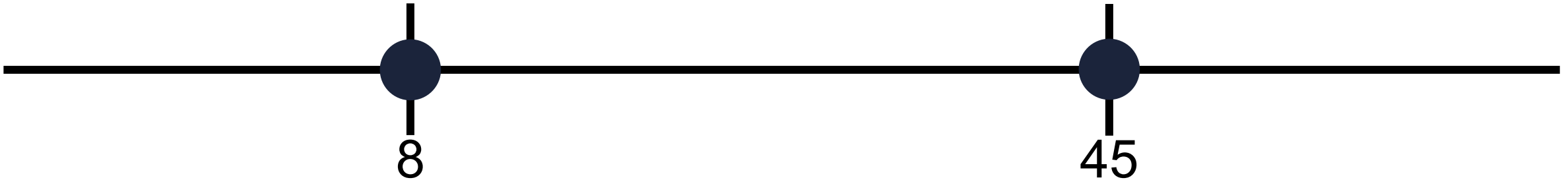
引用：[ISTQBテスト制度Advanced Level シラバス日本語版 テストアナリスト Version 3.1.1.J03](#)

まずは**同値分割**を
行って考える

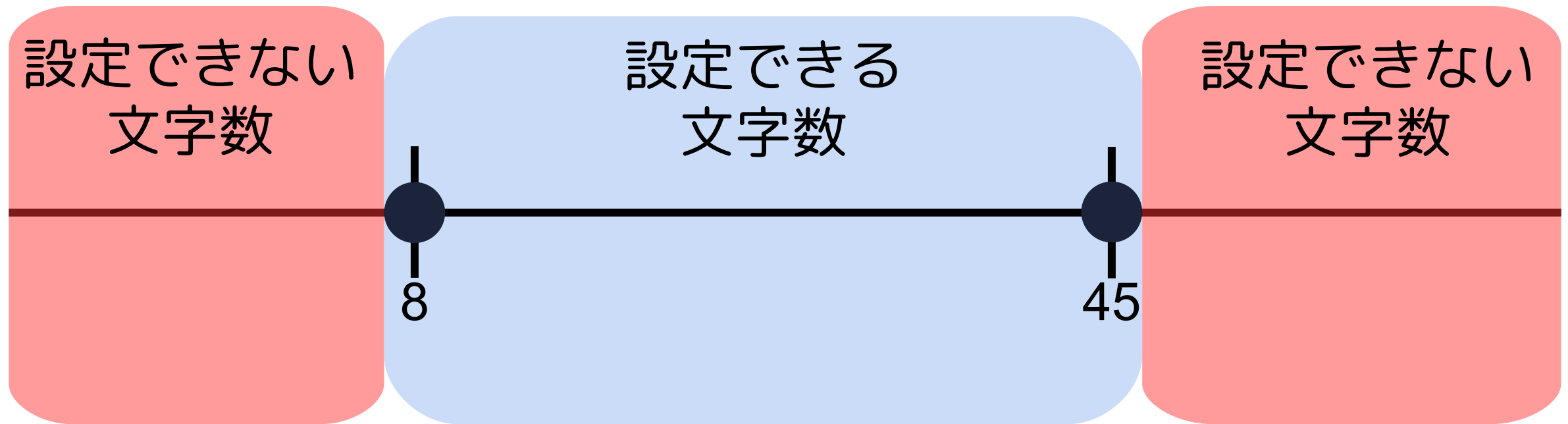
例題

8文字以上45文字以下のパスワードは設定できる

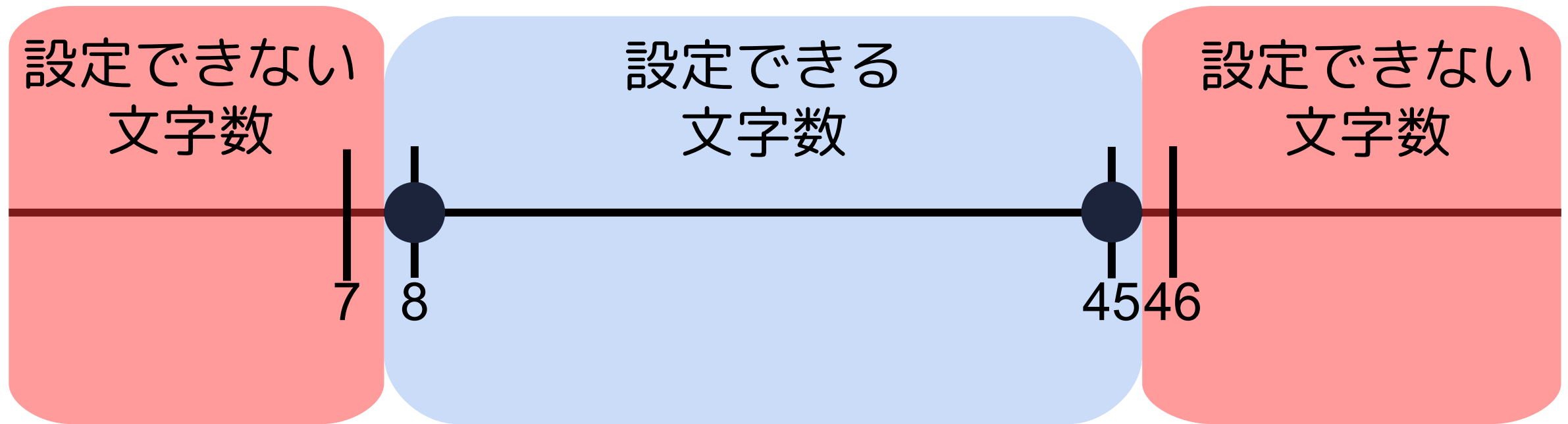
1. 数直線で表現する



2. 同値分割をする



3. 仕様で言及がない境界の値を記載する



「パスワードが8文字以上」の場合、
なぜ7と8をテストするのか？

```
if( x < 7 ){  
    return “入力したパスワードが短いです”;  
}  
...
```

「パスワードが8文字以上」の場合、
なぜ7と8をテストするのか？

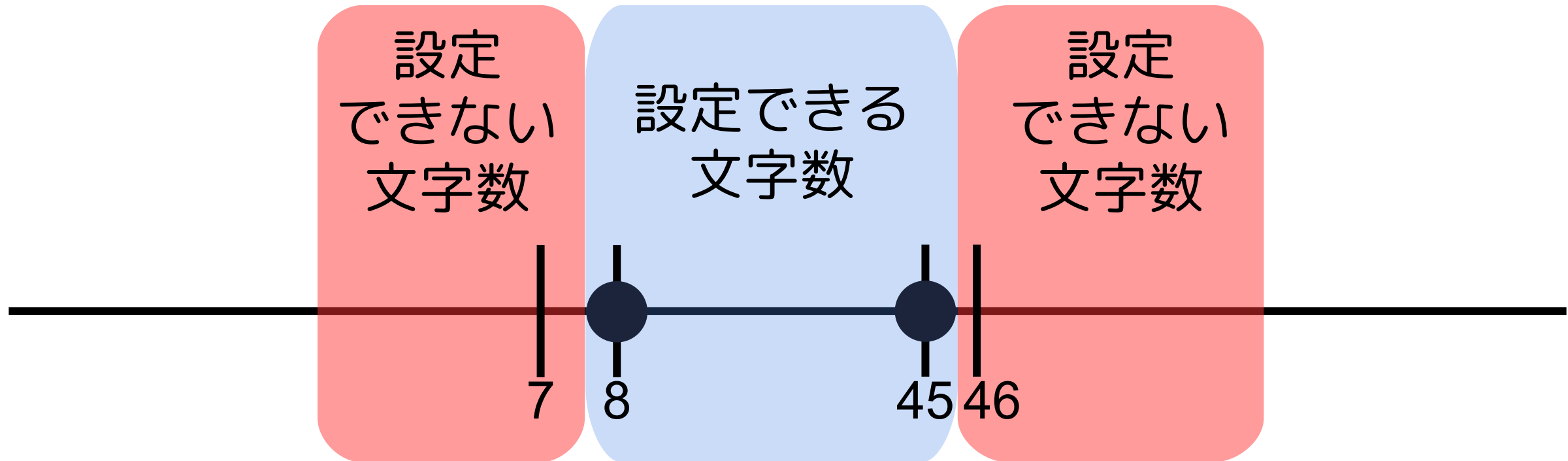
```
if( x < 7 ){  
    return “入力したパスワードが短いです”;  
}  
...
```

「パスワードが8文字以上」の場合、
なぜ7と8をテストするのか？

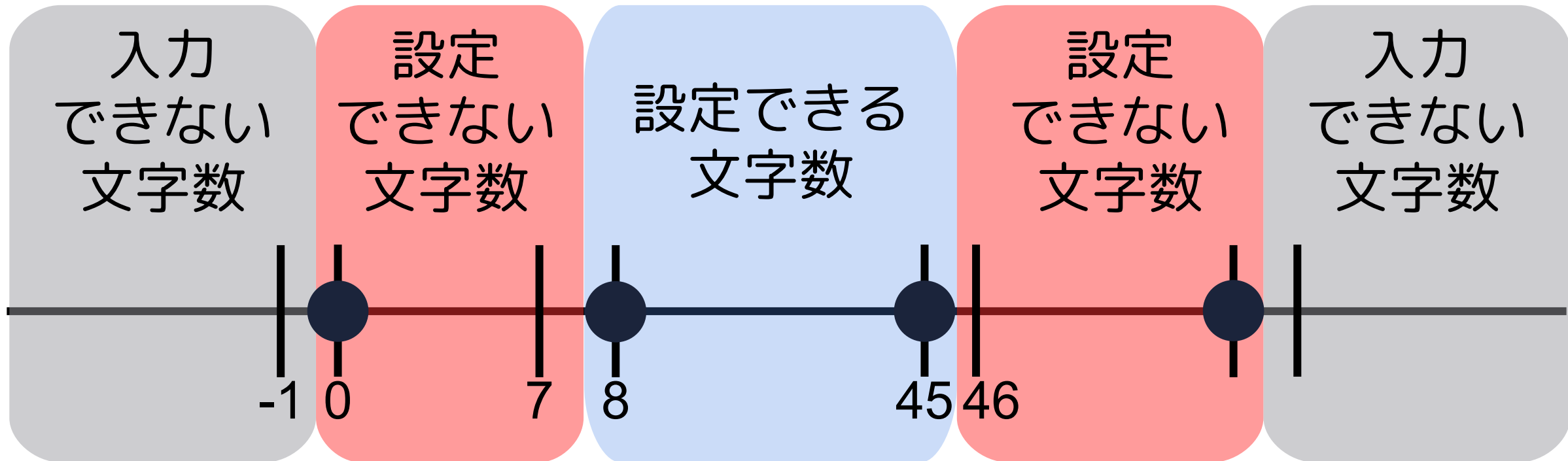
```
if( x <= 7 ){  
    return “入力したパスワードが短いです”;  
}  
...
```

不等号のミスによる不具合を発見できるのは、**7の時だけ！**

4. システム上の境界値も考える



4. システム上の境界値も考える

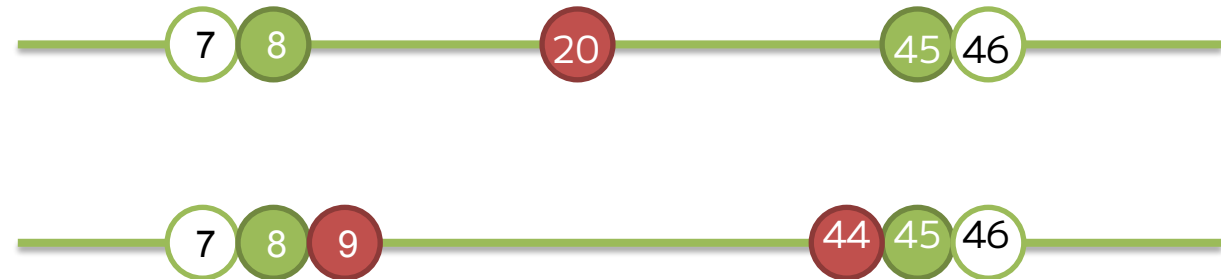


5. テストケースにする

#	文字数 (入力値)	設定可否 (期待結果)
1	-1文字	入力できない
2	0文字	設定できない
3	7文字	設定できない
4	8文字	設定できる

#	文字数 (入力値)	設定可否 (期待結果)
5	45文字	設定できる
6	46文字	設定できない
7	256文字	設定できない
8	257文字	入力できない

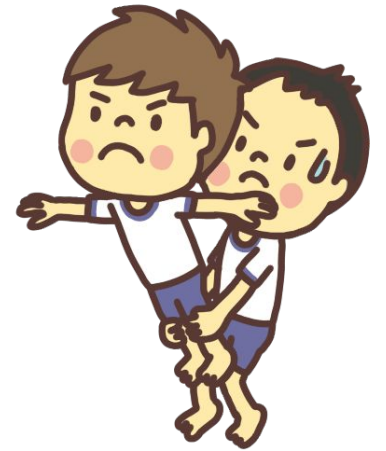
- 境界値分析の弱点
 - 「 $8 \leq x \leq 45$ 」を「 $x=8$ or $x=45$ 」と実装されると
バグを検出できない
 - 境界値は極端な値**。境界値ばかり攻めてると
普通のケースでバグを見逃すことに・・・
- 対策
 - 同値分割と組み合わせる
 - 3点の境界値をテストする



自然言語でも境界値分析を使う場合あり

例) 20人のクラスで**背の順**に並んだ時、
前から10人目までは組体操で上に登る人、
前から11人目以降は組体操で下で土台になる人

何らかのルールで**順序付け**ができれば、
数値でなくても境界値分析は適用可能



自然言語は境界が曖昧

「以上、以下」「から、まで」などの単語に注目

例

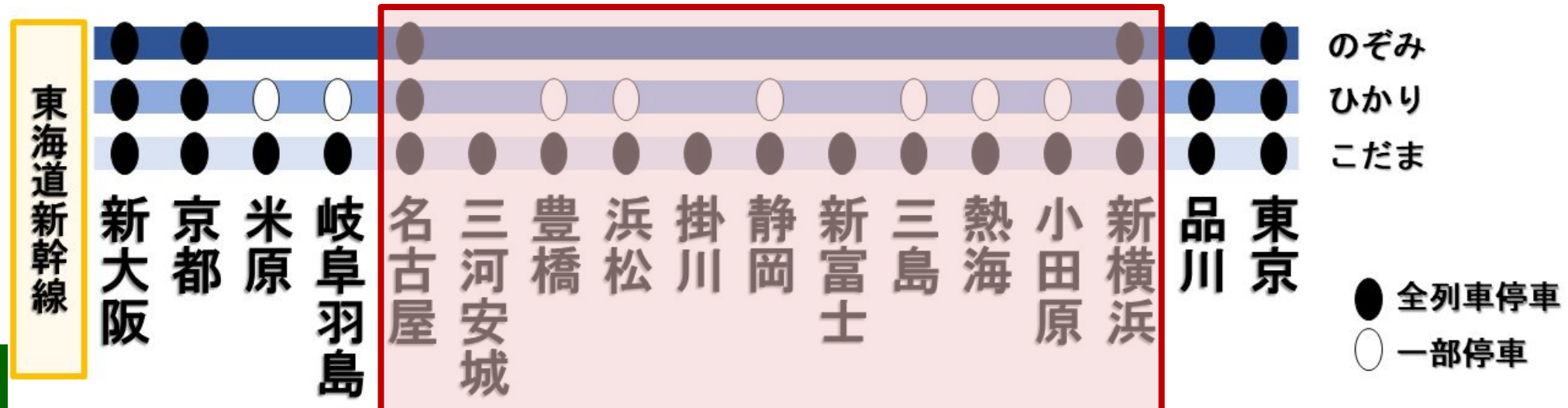
稼働時間...8:00から22:00まで

休止時間...22:00から8:00まで

8:00ちょうどや22:00ちょうどは稼働時間？休止時間？

例. 東海道新幹線で名古屋駅から**新横浜駅まで乗車した**

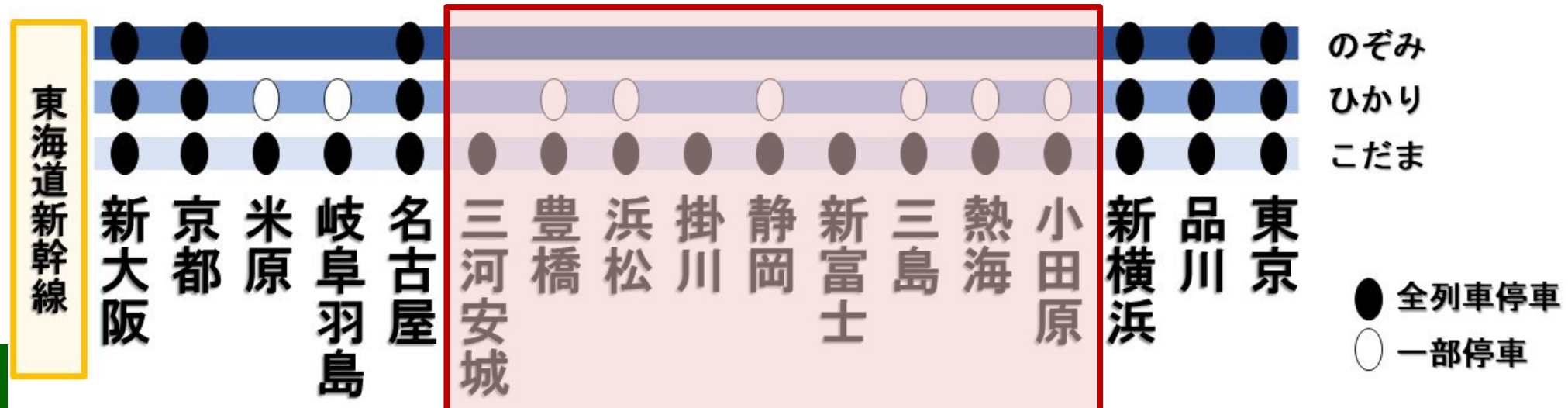
- 東海道新幹線で乗車していた駅は図の通り
- 境界値は「岐阜羽島と名古屋」「新横浜と品川」
- このときの「新横浜駅まで」は、**新横浜駅を含む**



例. 東海道新幹線の“のぞみ”は名古屋駅を出ると

新横浜駅まで停車しません

- “こだま”が止まって“のぞみ”が止まらない駅は図の通り
- 境界値は「名古屋と三河安城」「小田原と新横浜」
- このときの「新横浜駅まで」は、**新横浜駅を含まない**





ミニワーク

～実際にやってみよう～



(公開資料では省略)

- いかがでしたか？
 - ・ 思ったより手が動かなかった人もいるのでは？
- やさしいものでも、結構考えるところがあります
 - ・ 実際の仕事は、もっと難しい・複雑ですよ
 - ・ 解りやすい同値分割でも、実は怪しい場面が



デシジョンテーブル

- デシジョンテーブルとは
 - **複数の条件によって決まる動作**を表形式で整理する
 - ・ カタカナで書くと格好良いけど日本語（JIS X0125）だと『決定表』です。
 - 複数の条件の組み合わせの**不整合**や**考慮もれ**などバグがありそうなところを狙う設計技法です。
 - 複雑な**論理(ロジック)**を**網羅的**に確認します。

例題

とあるECサイトでは配送時に配送料がかかります。

- 合計10,000円以上購入すると無料
 - 通常荷物の場合は500円、大型荷物の場合は1500円
 - 大型荷物が1つでも含まれていたら大型荷物の配送料となる
 - 通常荷物は、配送地域によって値段が変わる
 - 四国・九州・北海道への配送料は+100円
 - 離島への配送料は+200円
 - 離島へ大型荷物を配送することはできない
- このときの配送料について考えてみよう。

1. 条件を洗い出す

- 購入額と同値パーティション
 - 10,000円以上、10,000円未満
- 荷物の種類の同値パーティション
 - 大型荷物を含む、大型荷物を含まない
- 配送先の同値パーティション
 - 本州、四国・九州・北海道、離島

1. 条件を洗い出す

- 購入額と同値パーティション
 - 10,000円以上、10,000円未満
- 荷物の種類の同値パーティション
 - 大型荷物を含む、大型荷物を含まない
- 配送先の同値パーティション
 - 本州、四国・九州・北海道、離島

配送先は
日本国内限定？

2.デシジョンテーブルの 条件部分に 書き出す

条件	購入額	10,000円以上
		10,000円未満
	大型荷物	含む
		含まない
	配送先	本州
		四国・九州・北海道
		離島

			1	2
条件	購入額	10,000円以上	○	
		10,000円未満		○
	大型荷物	含む	○	
		含まない		
	配送先	本州		
		四国・九州・北海道	○	
		離島		○

まずは10,000円以上で
大型荷物を四国に
配送する場合は...

次に10,000円未満で
通常荷物を離島に
配送する場合は...

			1	2
条件	購入額	10,000円以上	○	
		10,000円未満		○
	大型荷物	含む		
		含まない		
	配送先	本州		
		四国・九州・北海道		
		離島		

ちょっと待って！
そうじゃないんだ！

通常荷物を離島に
配送する場合は...

- 思いついた組み合わせで列を足していく
「足し算」の考え方は行わない
 - 思いつかなかった組み合わせが漏れてしまう
- 最初に全ての条件の組み合わせを**「掛け算」**で用意し、
テストできないものを削る**「引き算」**の考え方をを用いる

3.条件の掛け算をして、
全体のルール数を算出する

今回の場合、
購入額のルール数（2通り）×荷物種類のルール数（2通り）
×配送先のルール数（3通り）
＝12通り

つまり、12列分作成する

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上												
		10,000円未満												
	大型荷物	含む												
		含まない												
	配送先	本州												
		四国・九州・北海道												
		離島												

3.機械的にルールを埋めていく

全体の数（12通り）÷購入額のルール数（2通り） = **6**なので、

購入額の行は**6つつつ割り当てる**

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上	○	○	○	○	○	○						
		10,000円未満												
	大型荷物	含む												
		含まない												
	配送先	本州												
		四国・九州・北海道												
		離島												

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上	○	○	○	○	○	○						
		10,000円未満							○	○	○	○	○	○
	大型荷物	含む												
		含まない												
	配送先	本州												
		四国・九州・北海道												
		離島												

3.機械的にルールを埋めていく

購入額の条件1つあたりに割り当てた数（6つ）
÷大型荷物のルール数（2通り）= **3**なので、

大型荷物のルール数の行は**3つずつ割り当てる**

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上	○	○	○	○	○	○						
		10,000円未満							○	○	○	○	○	○
	大型荷物	含む	○	○	○									
		含まない				○	○	○						
	配送先	本州												
		四国・九州・北海道												
		離島												

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上	○	○	○	○	○	○						
		10,000円未満							○	○	○	○	○	○
	大型荷物	含む	○	○	○				○	○	○			
		含まない				○	○	○				○	○	○
	配送先	本州												
		四国・九州・北海道												
		離島												

3. 機械的にルールを埋めていく

大型荷物の条件1つあたりに割り当てた数（3つ）
÷ 配送先のルール数（3通り） = **1**なので、

配送先のルール数の行は**1つずつ割り当てる**

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上	○	○	○	○	○	○						
		10,000円未満							○	○	○	○	○	○
	大型荷物	含む	○	○	○				○	○	○			
		含まない				○	○	○				○	○	○
	配送先	本州	○											
		四国・九州・北海道		○										
		離島			○									

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上	○	○	○	○	○	○						
		10,000円未満							○	○	○	○	○	○
	大型荷物	含む	○	○	○				○	○	○			
		含まない				○	○	○				○	○	○
	配送先	本州	○			○			○			○		
		四国・九州・北海道		○			○			○			○	
		離島			○			○			○			○

4.ルール1列ずつに対して、動作を考える

例. 「購入額が10,000円以上」かつ「大型荷物を含む」
かつ「配送先が本州」の場合は0円

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上	☐	☐	☐	☐	☐	☐						
		10,000円未満							☐	☐	☐	☐	☐	☐
	大型荷物	含む	☐	☐	☐				☐	☐	☐			
		含まない				☐	☐	☐				☐	☐	☐
	配送先	本州	☐			☐			☐			☐		
		四国・九州・北海道		☐			☐			☐			☐	
		離島			☐			☐			☐			☐
動作	配送料	0円	X											
		500円												
		600円												
		700円												
		1,500円												

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上	○	○	○	○	○	○						
		10,000円未満							○	○	○	○	○	○
	大型荷物	含む	○	○	○				○	○	○			
		含まない				○	○	○				○	○	○
	配送先	本州	○			○			○			○		
		四国・九州・北海道		○			○			○			○	
		離島			○			○			○			○
動作	配送料	0円	X	X		X	X	X						
		500円										X		
		600円											X	
		700円												X
		1,500円							X	X				

5.ルール上、あり得ない組み合わせ（禁則）を削除する

例. 「購入額が10,000円以上」かつ「大型荷物を含む」
かつ「配送先が離島」の場合は禁則

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上	○	○	○	○	○	○			○			
		10,000円未満			○				○	○	○	○	○	○
	大型荷物	含む	○	○	○				○	○	○			
		含まない			○	○	○				○	○	○	
	配送先	本州	○		○	○			○		○			
		四国・九州・北海道		○	○		○			○			○	
		離島			○			○			○			○
動作	配送料	0円	X	X	○	X	X	X			○			
		500円			○						○	X		
		600円			○						○		X	
		700円			○						○			X
		1,500円			○				X	X	○			

あり得ない
組み合わせを
削除した結果

			1	2	3	4	5	6	7	8	9	10
条件	購入額	10,000円以上	○	○	○	○	○					
		10,000円未満						○	○	○	○	○
	大型荷物	含む	○	○				○	○			
		含まない			○	○	○			○	○	○
	配送先	本州	○		○			○		○		
		四国・九州・北海道		○		○			○		○	
		離島					○					○
動作	配送料	0円	X	X	X	X	X					
		500円								X		
		600円									X	
		700円										X
		1,500円						X	X			

6.判定に無関係なルールを「ー」で明示して1列にまとめる (簡単化)

例「購入額が10,000円以上」かつ「大型荷物を含まない」は
配送先が「本州」「四国・九州・北海道」「離島」の
いずれの場合でも、配送料は「0円」
(配送先が判定に無関係)

購入額が
10,000円以上
かつ
大型荷物を
含まない
は、配送先が
配送料に無関係
(必ず0円)

			1	2	3	4	5	6	7	8	9	10
条件	購入額	10,000円以上	○	○	○	○	○					
		10,000円未満						○	○	○	○	○
	大型荷物	含む	○	○				○	○			
		含まない			○	○	○			○	○	○
	配送先	本州	○		○			○		○		
		四国・九州・北海道		○		○			○		○	
		離島					○					○
動作	配送料	0円	X	X	X	X	X					
		500円								X		
		600円									X	
		700円										X
		1,500円						X	X			

購入額が
10,000円以上
かつ
大型荷物を
含まない
は、配送先が
配送料に無関係
(必ず0円)

			1	2	3	4	5	6	7	8	9	10
条件	購入額	10,000円以上	○	○	○							
		10,000円未満						○	○	○	○	○
	大型荷物	含む	○	○				○	○			
		含まない			○					○	○	○
	配送先	本州	○		—			○		○		
		四国・九州・北海道		○	—				○		○	
		離島			—							○
動作	配送料	0円	X	X	X							
		500円								X		
		600円									X	
		700円										X
		1,500円						X	X			

購入額が
10,000円以上かつ
大型荷物を含む
は、配送先が
配送料に無関係
(必ず0円)

※離島は禁則のため
考慮しなくてOK

			1	2	3	4	5	6	7	8
条件	購入額	10,000円以上	○	○	○					
		10,000円未満				○	○	○	○	○
	大型荷物	含む	○	○		○	○			
		含まない			○			○	○	○
	配送先	本州	○		—	○		○		
		四国・九州・北海道		○	—		○		○	
		離島			—					○
動作	配送料	0円	X	X	X					
		500円						X		
		600円							X	
		700円								X
		1,500円				X	X			

購入額が
10,000円以上かつ
大型荷物を含む
は、配送先が
配送料に無関係
(必ず0円)

※離島は禁則のため
考慮しなくてOK

			1	2	3	4	5	6	7	8
条件	購入額	10,000円以上	○		○					
		10,000円未満				○	○	○	○	○
	大型荷物	含む	○			○	○			
		含まない			○			○	○	○
	配送先	本州	—		—	○		○		
		四国・九州・北海道	—		—		○		○	
		離島			—					○
動作	配送料	0円	X		X					
		500円						X		
		600円							X	
		700円								X
		1,500円				X	X			

購入額が
10,000円未満かつ
大型荷物を含む
は、配送先が
配送料に無関係
(必ず1,500円)

※離島は禁則のため
考慮しなくてOK

			1	2	3	4	5	6	7
条件	購入額	10,000円以上	○	○					
		10,000円未満			○	○	○	○	○
	大型荷物	含む	○		○	○			
		含まない		○			○	○	○
	配送先	本州	—	—	○		○		
		四国・九州・北海道	—	—		○		○	
		離島		—					○
動作	配送料	0円	X	X					
		500円					X		
		600円						X	
		700円							X
		1,500円			X	X			

購入額が
10,000円未満かつ
大型荷物を含む
は、配送先が
配送料に無関係
(必ず1,500円)

※離島は禁則のため
考慮しなくてOK

			1	2	3	4	5	6	7
条件	購入額	10,000円以上	○	○					
		10,000円未満			○		○	○	○
	大型荷物	含む	○		○				
		含まない		○			○	○	○
	配送先	本州	—	—	—		○		
		四国・九州・北海道	—	—	—			○	
		離島		—					○
動作	配送料	0円	X	X					
		500円					X		
		600円						X	
		700円							X
		1,500円			X				

購入額が
10,000円以上は、
大型荷物や配送先が
配送料に無関係
(必ず0円)

※大型荷物かつ離島は
禁則のため
考慮しなくてOK

			1	2	3	4	5	6
条件	購入額	10,000円以上	○	○				
		10,000円未満			○	○	○	○
	大型荷物	含む	○		○			
		含まない		○		○	○	○
	配送先	本州	—	—	—	○		
		四国・九州・北海道	—	—	—		○	
		離島		—				○
動作	配送料	0円	X	X				
		500円				X		
		600円					X	
		700円						X
		1,500円			X			

購入額が
10,000円以上は、
大型荷物や配送先が
配送料に無関係
(必ず0円)

※大型荷物かつ離島は
禁則のため
考慮しなくてOK

			1	2	3	4	5	6
条件	購入額	10,000円以上	○					
		10,000円未満			○	○	○	○
	大型荷物	含む	—		○			
		含まない	—			○	○	○
	配送先	本州	—		—	○		
		四国・九州・北海道	—		—		○	
		離島	—					○
動作	配送料	0円	X					
		500円				X		
		600円					X	
		700円						X
		1,500円			X			

デシジョンテーブルの使い方

1. 同値分割で条件を洗い出す
2. 最初に全ての組み合わせを用意する
3. テストできない組み合わせを削る（禁則の削除）
4. 動作の決定に無関係な条件をまとめる（簡単化）

全パターン...12通り
→禁則の削除...10通り
→単純化 ...5通り

			1	2	3	4	5
条件	購入額	10,000円以上	○				
		10,000円未満		○	○	○	○
	大型荷物	含む	—	○			
		含まない	—		○	○	○
	配送先	本州	—	—	○		
		四国・九州・北海道	—	—		○	
		離島	—				○
動作	配送料	0円	X				
		500円			X		
		600円				X	
		700円					X
		1,500円		X			

- 「削る」「まとめる」は慎重に！
 - **本当はテストできるのでは？**
 - 無関係と言える**根拠**は？
 - ケースを減らしたい気持ちが先行すると失敗するかも

			1	2	3	4	5	6	7	8	9	10	11	12
条件	購入額	10,000円以上	○	○	○	○	○	○						
		10,000円未満							○	○	○	○	○	○
	大型荷物	含む	○	○	○				○	○	○			
		含まない				○								○
	配送先	本州	○											
		四国・九州・北海道		○										
		離島			○									
動作	配送料	0円	X	X										
		500円												
		600円												
		700円												
		1,500円							X	X				X

大型荷物かつ離島は
間違って
設定できちゃうかも
しれないから
念のため
テストしよう

- **議論の土台**にするためにデシジョンテーブルを活用する
 - デシジョンテーブルで作成した結果が**絶対解ではない**
 - デシジョンテーブルを示すことで、
どのように削れば良いか議論できる
- 削減前のデシジョンテーブルも**残しておく**
 - 削った結果なのか、パターン漏れなのか判別したいため



ミニワーク

～実際にやってみよう～



(公開資料では省略)

省略

状態遷移テスト

省略

ユースケーステスト

省略

組合せテスト技法

省略

ドメイン分析

6. 今後の勉強に向けて・まとめ

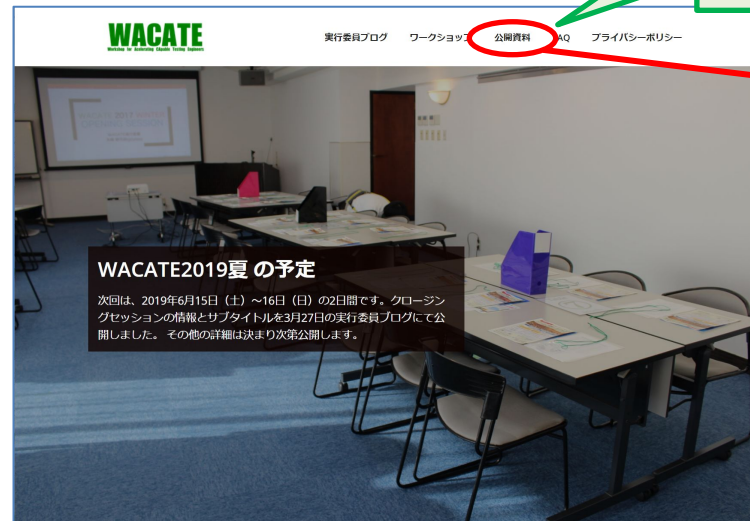
資料の解説



スライドにたどりつくには？

- A) ググる「テスト設計技法」＋「WACATE」
- B) 2次元バーコード
- C) WACATEサイトのリンク

WACATE
公開資料
のリンク

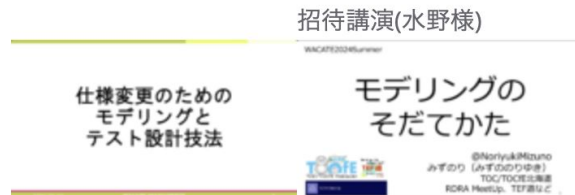


- WACATEの「テスト設計技法」に関連する公開資料

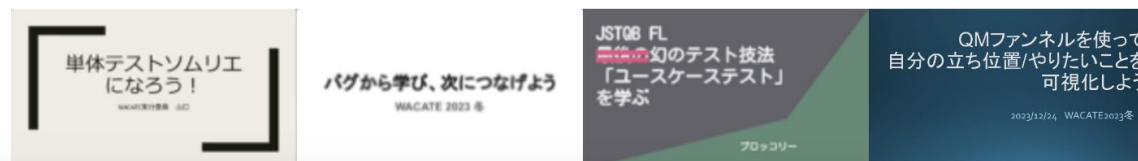
WACATE2024 冬 ～夢の中ではテストできてました～



WACATE2024 夏 ～テストケースには、必ず作った人の意図が存在する～



WACATE2023 冬 ～WACATE冬が今年もやってくる～



100セッション以上
資料公開中！

- なぜ、テスト設計技法を使うか
- テスト設計技法
 - いろいろとある
 - 一部を説明
 - 同値分割／境界値分析／デシジョンテーブル
- 実際に手を動かす
 - 同値分割と境界値分析のミニワークを実施
- 勉強用の資料



7. WACATEの紹介

どのような雰囲気か知って欲しい



<https://wacate.jp>

1. WACATE ワカテと読みます
2. 活動
 - ・ 自主的なコミュニティ
 - ・ 1年に夏・冬の2回
 - ・ 1泊2日の合宿
3. 内容
 - ・ 主にソフトウェアテストの勉強会
 - ・ ワークショップがメイン
 - ・ 参加者間の交流も深める
4. ターゲット
 - ・ 若手でテストに興味がある人
 - ・ 特に年齢制限はない



ワークショップ向けの会場を利用

- セミナー＋ホテル



- なかなか聞けない話も
 - 成長につながる話
 - 参加者と距離の近さ



一つのテーマを掘り下げどっぷりつかると二日間

様々なトピックに触れ新しい発見を得ると二日間

夏

狭く深く



冬

広く浅く



20年近くの実績がある

2007年より開催

第1回目は上野の
「水月ホテル鷗外荘」
にて開催されました



写真はホテルホームページより

いろいろ扱ってきました

テスト海外動向

テスト分析(ゆもつよメソッド)

テスト分析・設計～テストケース作成までを一気通関で

CFD

人間関係

BOK

報告書

PFD

テスト計画

テスト教育

心・態度・習慣

テスト条件

テスト条件

テスト導出

コミュニケーション

バグ票

29119

TPI NEXT

Wモデル

データ分析

テスト設計

プレゼンテーション

テスト分析

テストケースの再利用

インシデントレポート

デシジョンテーブル

プロセス

品質

見積もり

同値分割・境界値分析

モチベーション

SaPID

組み合わせテスト

キャリアプラン

伝える

テストの基本

テストを語る

メトリクス

品質管理

英語

モデル検査

論文

技法

クラシフィケーション・ツリー

探索的テスト

リスクベースドテスト

インシデント分析

レビュー

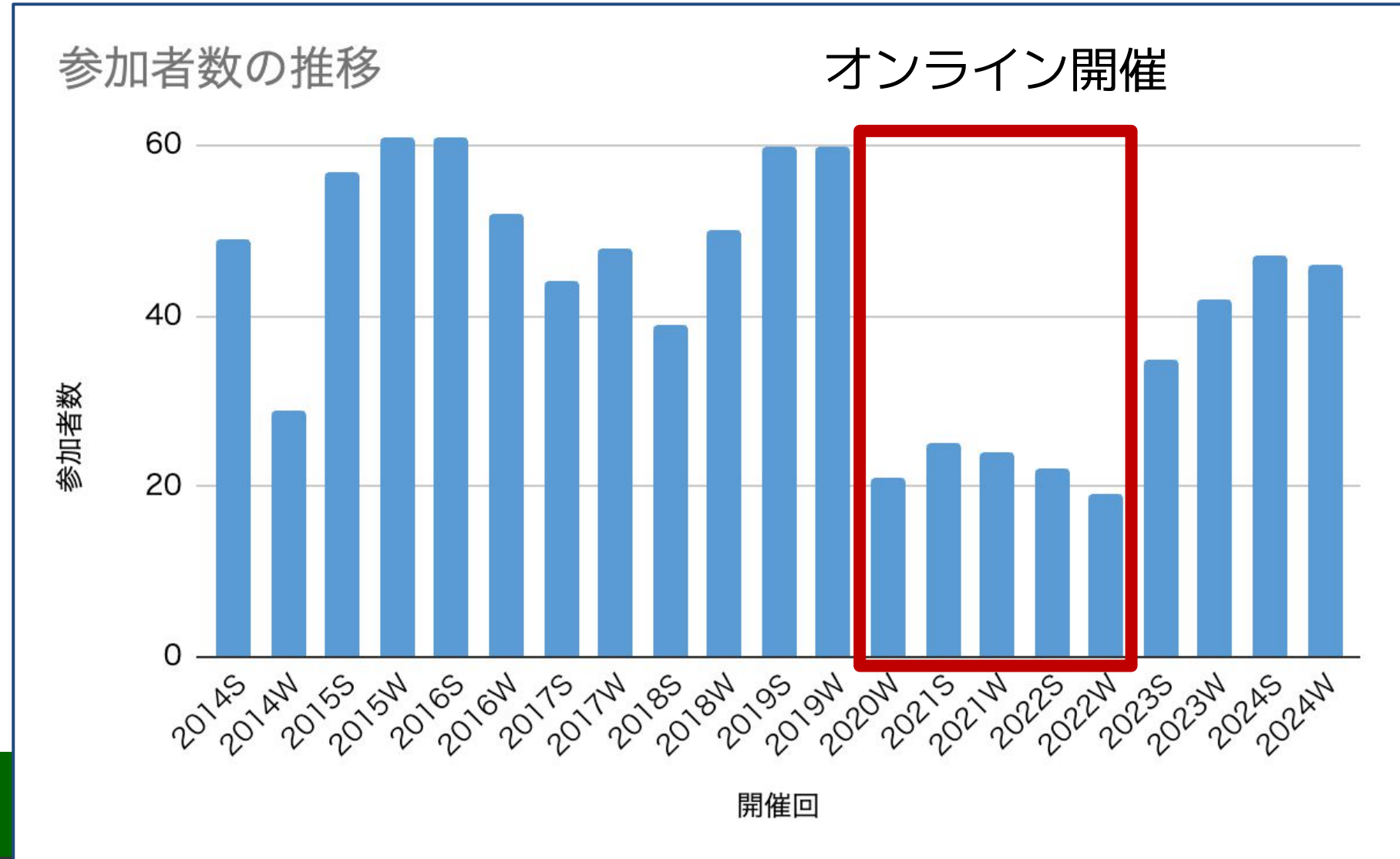
状態遷移テスト

ワールドカフェ

情報探索

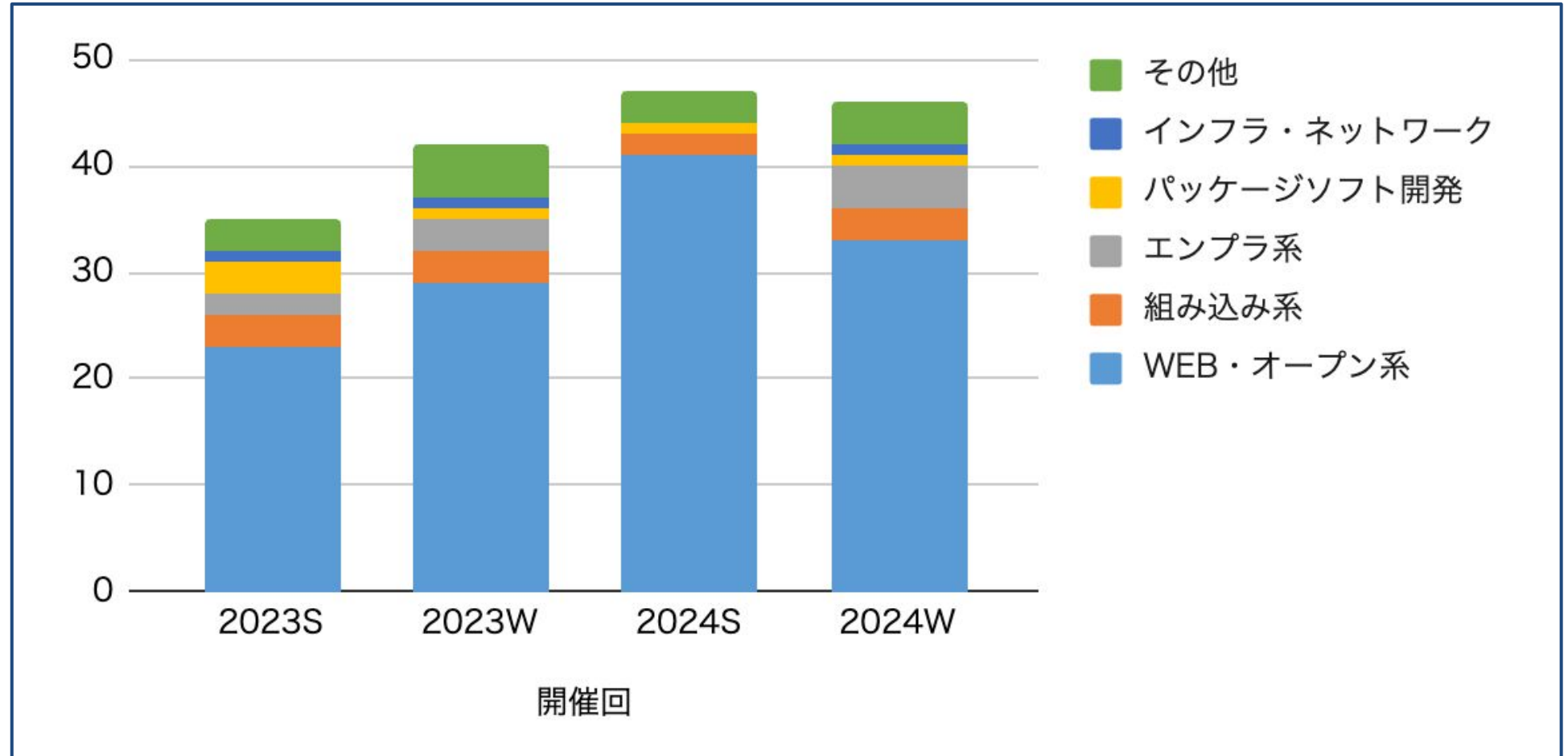
※大きさ、色に意図はありません

最近は
40名前後
の参加



多種多様な参加者(1)

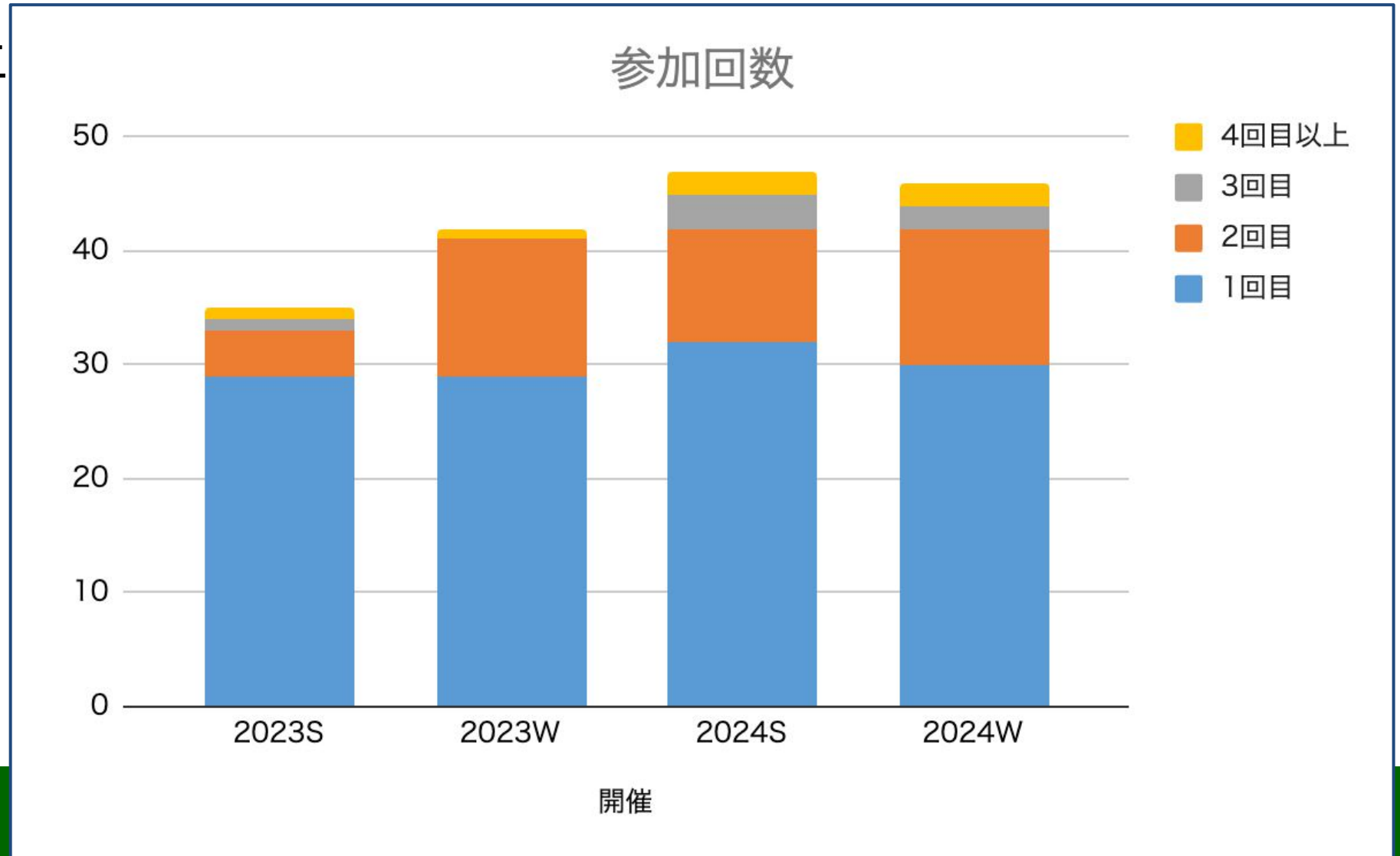
様々な業界 からの参加



多種多様な参加者(2)

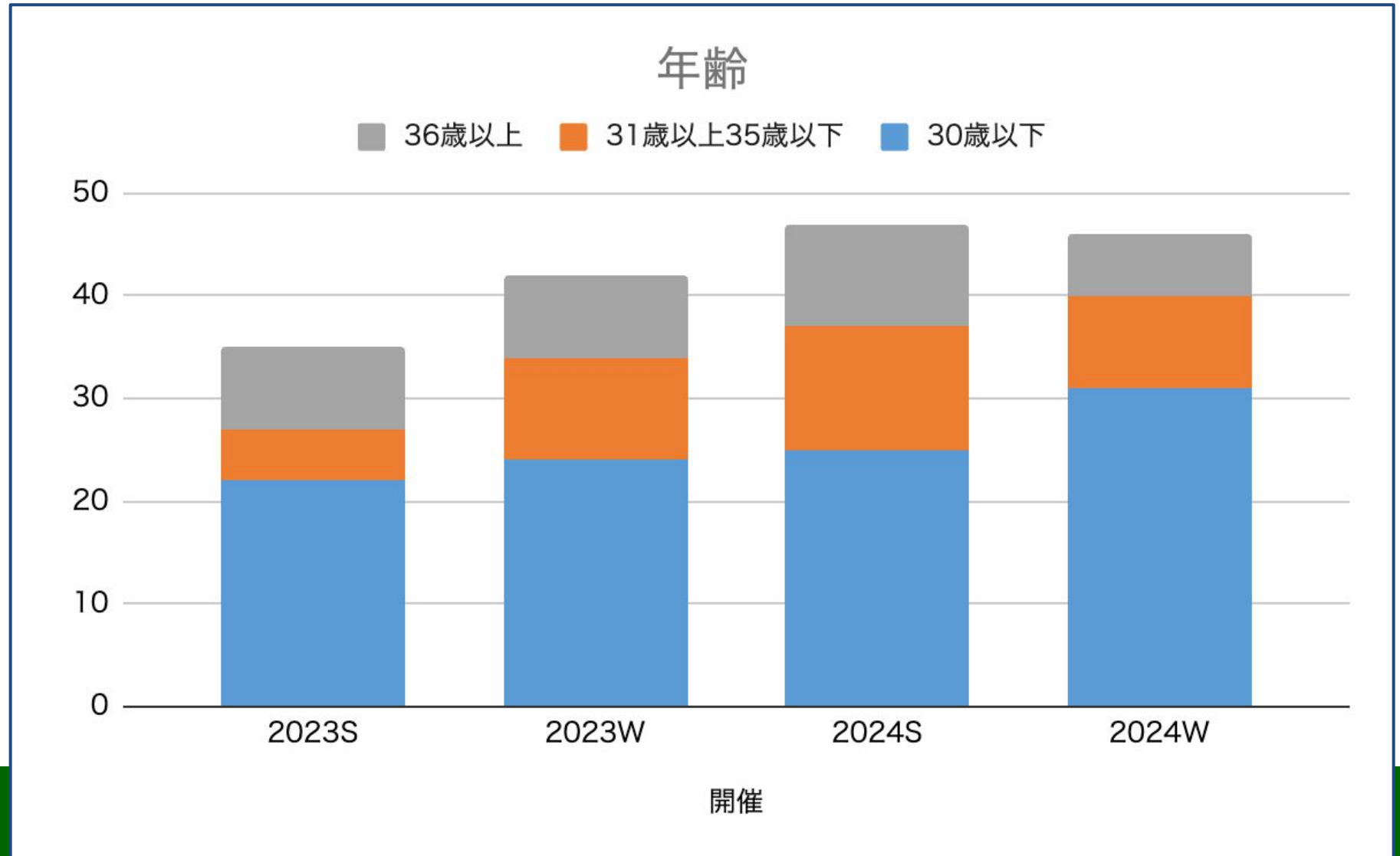
初参加が半数以上

リピーターも
増えてます



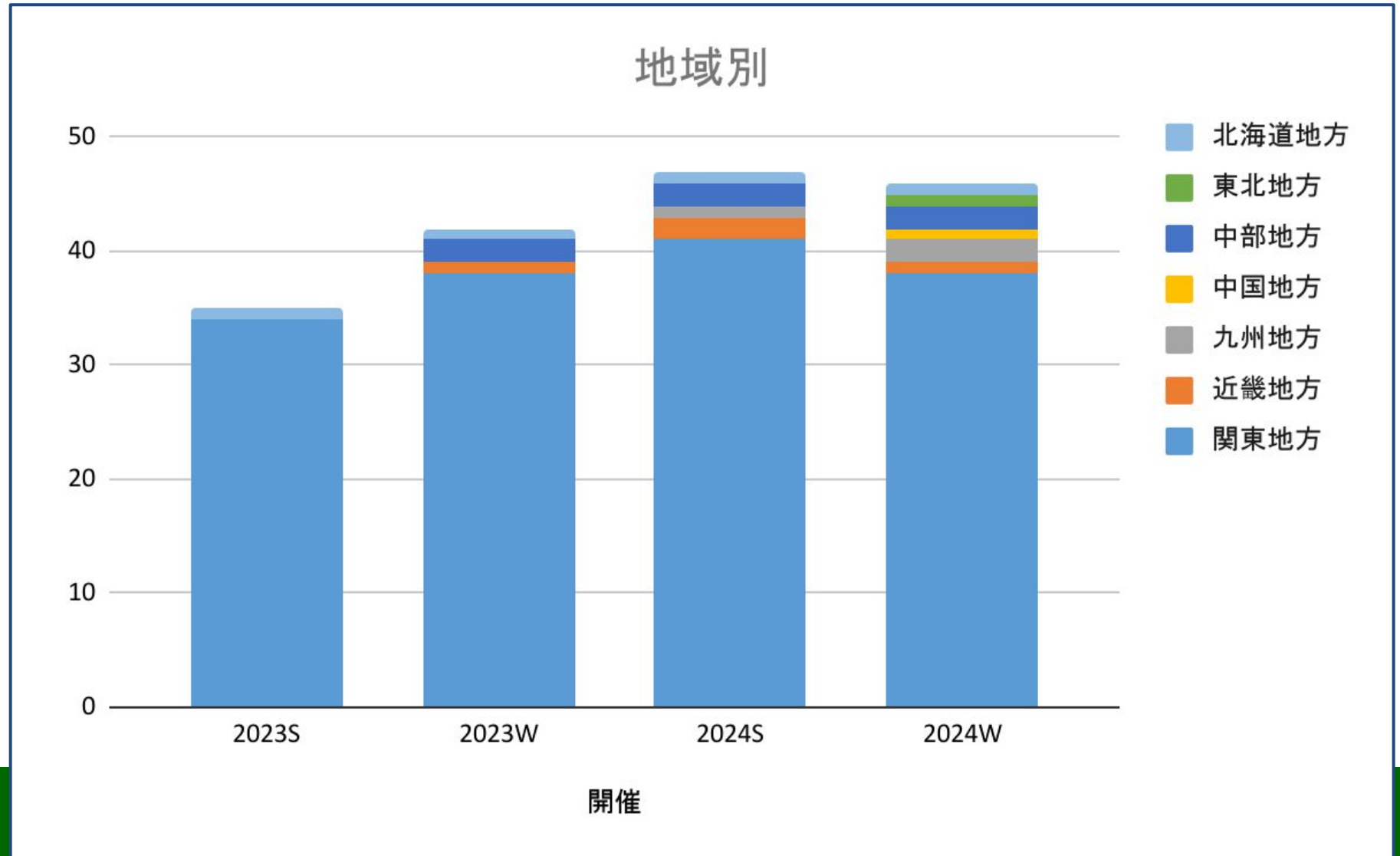
多種多様な参加者(3)

36歳以上の
参加者も
一定数います



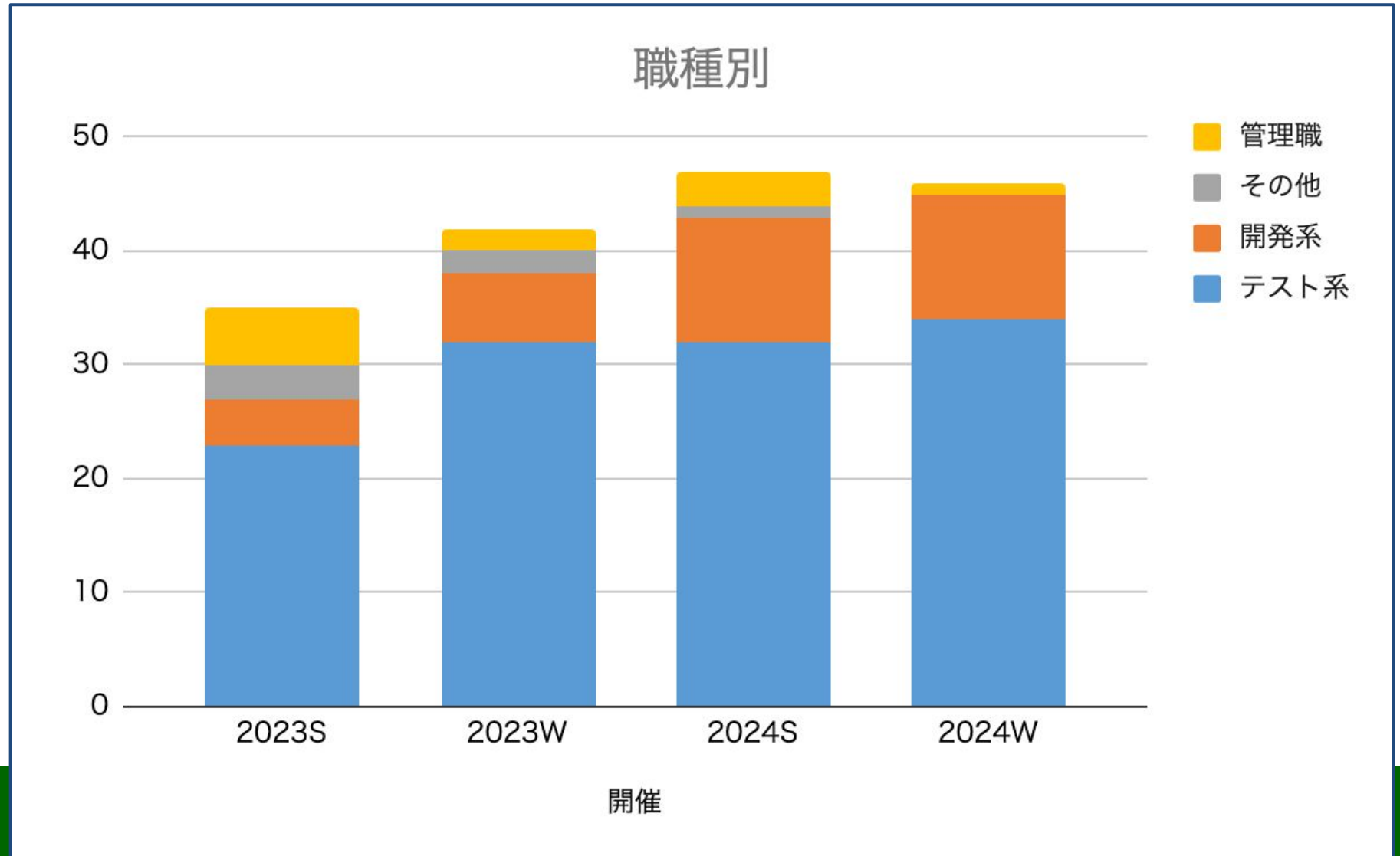
多種多様な参加者(4)

関東以外からの
参加者も



多種多様な参加者(5)

開発系の参加者
も増えてます



2日間のスケジュール

- ◆ 学び・経験・交流もあるテスト漬けの2日間
- ◆ 前泊も可能（終電で来る方も）

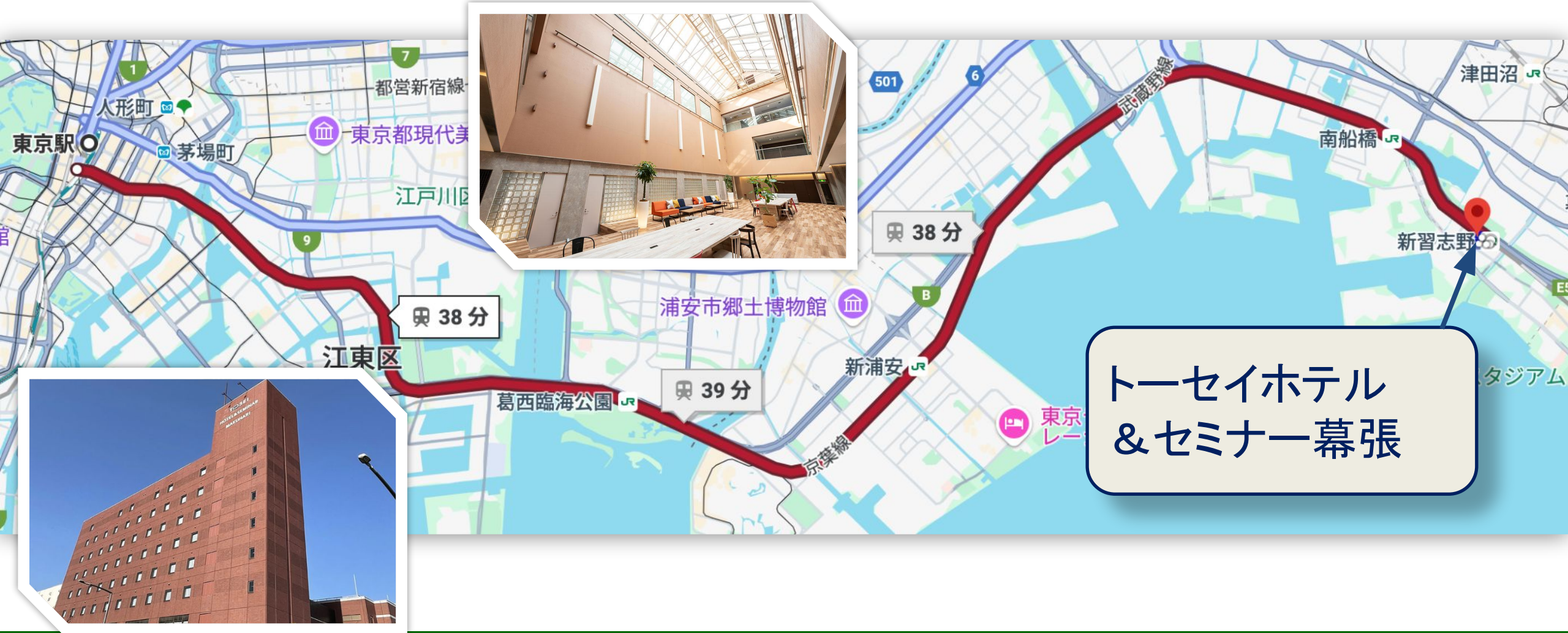


1日目		2日目	
		7:00	朝食
		9:00	セッション
10:00	セッション		
12:10	昼食	12:10	昼食
13:10	セッション	13:10	セッション
17:20	休憩：自室	16:40	解散
19:00	夕食		
	休憩		
21:20	分科会		

開催概要

テーマ	未定
サブタイトル	未定
主催	WACATE実行委員会
日時	2025年 6月28日(土) 9:30 受付開始 2025年 6月29日(日) 17:00頃 終了予定 (1泊2日 4食付き)
会場	トーセイホテル&セミナー幕張 JR京葉線 新習志野駅 徒歩約2分
合宿費	3万円前後の予定(宿泊費、4食の食事代含む) 35歳以下は数千円安くなる予定
申し込み方法	WACATEサイト上より

都内からある程度近い



- テストの勉強を最近始めた
 - 業務の経験が短い
 - 広い視野を持ちたい
 - 若手世代同士で交流したい
 - ベテランとも交流したい
 - エンジニアとして元気を得たい
-
- 周りの方にも紹介いただけると幸いです
 - 同期、後輩、先輩、部下、上司 . . .

- 他のセミナーでは得られないものがたくさん

WACATE

Workshop for Accelerating CApable Testing Engineers