

The background of the slide is a blue-tinted photograph of several hands of different skin tones stacked on top of each other, symbolizing teamwork and collaboration. The hands are positioned in a way that they form a central point, with fingers pointing towards the center.

# 平成30年度 中小企業・小規模事業者に向けた 事例集アプリケーション開発に係る調査事業

# 目次

<b>1. 事業の概要</b>	<b>7</b>
1 – 1. 事業の目的について	7
1 – 2. 調査の内容について	8
(1) ユーザーにとって使いやすい事例の検索・閲覧のためのアプリ開発	8
1 – 3. 開発スケジュールについて	9
1 – 4. 体制について	10
(1) 体制図	10
(2) プロジェクト進捗管理・担当職員とのコミュニケーションについて	11
<b>2. システム概要</b>	<b>12</b>
2 – 1. 現状の課題と目指すべき状況	12
2 – 2. システム概要図・開発範囲	13
2 – 3. 機能一覧	14
<b>3. コンテストを活用したアプリケーション開発</b>	<b>19</b>
3 – 1. プロジェクト方針／要件検討	19
(1) プロジェクト方針について	19
(2) 要件検討について	20

# 目次

3 – 2. アプリケーション開発の進め方	21
3 – 3. UIデザインコンテスト実施報告	22
(1) コンテストの進め方	22
(2) コンテスト実施概要	23
(3) コンテスト参加者	24
(4) UIデザインコンテスト中間評価の目的と方法	25
(5) UIデザインコンテスト中間評価	28
(6) 中間評価結果のフィードバックについて	30
(7) UIデザインコンテスト最終評価	32
(8) コンテストの成果物	34
3 – 4. フロントエンド開発コンテスト実施報告	37
(1) フロントエンド開発コンテストの進行について	37
(2) コンテスト実施概要①	38
(3) コンテスト実施概要②	39
(4) コンテスト参加者	40



# 目次

3 – 5. バックエンド開発コンテスト実施報告	41
(1) バックエンド開発コンテストの進行について	41
(2) コンテスト実施概要①（コンテスト1回目）	42
(3) コンテスト実施概要②（コンテスト1回目）	43
(4) コンテスト参加者（コンテスト1回目）	44
(5) コンテスト実施概要①（コンテスト2回目）	45
(6) コンテスト実施概要②（コンテスト2回目）	46
(7) コンテスト参加者（コンテスト2回目）	47
3 – 6. インテグレーション開発コンテスト実施報告	48
(1) インテグレーション開発コンテストの進行について	48
(2) コンテスト実施概要①	49
(3) コンテスト実施概要②	50
(4) コンテスト参加者	51
3 – 7. 開発コンテストにおけるレビュープロセス	52
(1) 開発コンテストにおけるレビューシステムについて	52
(2) スコアカードの評価項目	53

# 目次

3 – 8. 開発コンテストで実施したプラクティス	54
(1) ベースコードの提供	54
(2) APIファーストの設計と開発	55
(3) 品質保持のための施策	56
(4) 自動ビルド・自動デプロイの構築	57
<b>4. コンテスト以外で実施した開発</b>	<b>58</b>
4 – 1. TC3で実施した開発タスク	58
(1) ユーザーインターフェースのローカライズ（日本語化）開発	58
(2) インフラ環境構築	59
(3) 結合テスト・品質管理	60
<b>5. 「中小企業支援プラットフォーム(仮称)」との連携</b>	<b>61</b>
5 – 1. 認証基盤との連携	61
5 – 2. 中小企業支援プラットフォーム（仮称）からの遷移	62

# 目次

<b>6. 総括</b>	<b>63</b>
6 – 1. 本調査における実績値について	63
(1) システムの開発言語、RDBMSとステップ数	63
(2) 開発フェーズ別の割合	64
(3) 不具合件数と発生原因	64
6 – 2. 本調査の評価	65
(1) 品質	65
(2) コスト（工数）	66
(3) 納期	67
(4) 効果的だった点	68
(5) 改善すべき点	69
6 – 3. 留意事項について	70
(1) Auth0の利用について	70
(2) 非機能要件について	71
<b>参考. Topcoderとは</b>	<b>72</b>
<b>参考. システム構成図</b>	<b>75</b>

# 1. 事業の概要

## 1-1. 事業の目的について

これまで中小企業庁をはじめ独立行政法人中小企業基盤整備機構等において、中小企業・小規模事業者（以下、「中小企業等」という。）や中小企業等支援事業者等の経営課題の解決に向けた参考となることを想定し、様々な事業に係る事例集を作成してきている。一方で、事例集に記載されている事例が当該企業の属性に応じて検索することが必ずしも容易ではない、あるいは事例集間での連携がなされていないなど、中小企業等や中小企業等支援機関等にとって必ずしも活用しやすい状況にあるとはいえない。そのため、当該事例集を電子化・データ化し一覧性を持たせると同時に、利用者の属性や関心に応じて例を検索・抽出できる適切な事例集の閲覧手法・媒体のあり方について調査を行い、電子化された事例集の閲覧・活用に当たっての課題とその解消方法を明らかにした上で、事例集の検索・閲覧が行いやすいアプリケーション（以下、「アプリ」という。）を試行的に開発し、事例集が中小企業等の経営課題の解決に一層寄与するものとなることが本事業の目的である。

当該アプリを設計するに当たっては、中小企業等の経営者等や中小企業等支援機関等における経営指導員等を主な利用者として想定しており、上記利用者が取り扱いやすいサービスデザインの実現に資するよう、アジャイル手法を用いる等、適切な手法を用いて設計する必要がある。

また、中小企業庁はワンストップで中小企業支援サービスを提供できるプラットフォーム「中小企業支援プラットフォーム(仮称)」の構築を検討しているところ、当該アプリと当該プラットフォームとの連携方法について検討を行うこととする。

# 1. 事業の概要

## 1 - 2. 調査の内容について

事業目的に関する先行事業や事例検索に係る現状と課題、中小企業等の経営者等や中小企業等支援機関等の想定される利用者の意見等を踏まえ、事例集の検索・活用を行うモデルとなるアプリを開発し、適切なアプリに求められる要件（アプリ利用者の利便性を考慮したデザイン、コンテンツ、機能等）のとりまとめを行うとともに、本開発プロセスを踏まえて、今後の中小企業等に関する事例集の更なる開発のあり方を提言する。

### （１）ユーザーにとって使いやすい事例の検索・閲覧のためのアプリ開発

ユーザーである中小企業経営者、中小企業支援機関等の職員、中小企業庁職員等が、中小企業庁等によって既に作成されている事例集に掲載されている事例を効率的に検索し、当該者の経営や経営相談対応等の参考として有効に活用されることを想定したアプリを開発する。

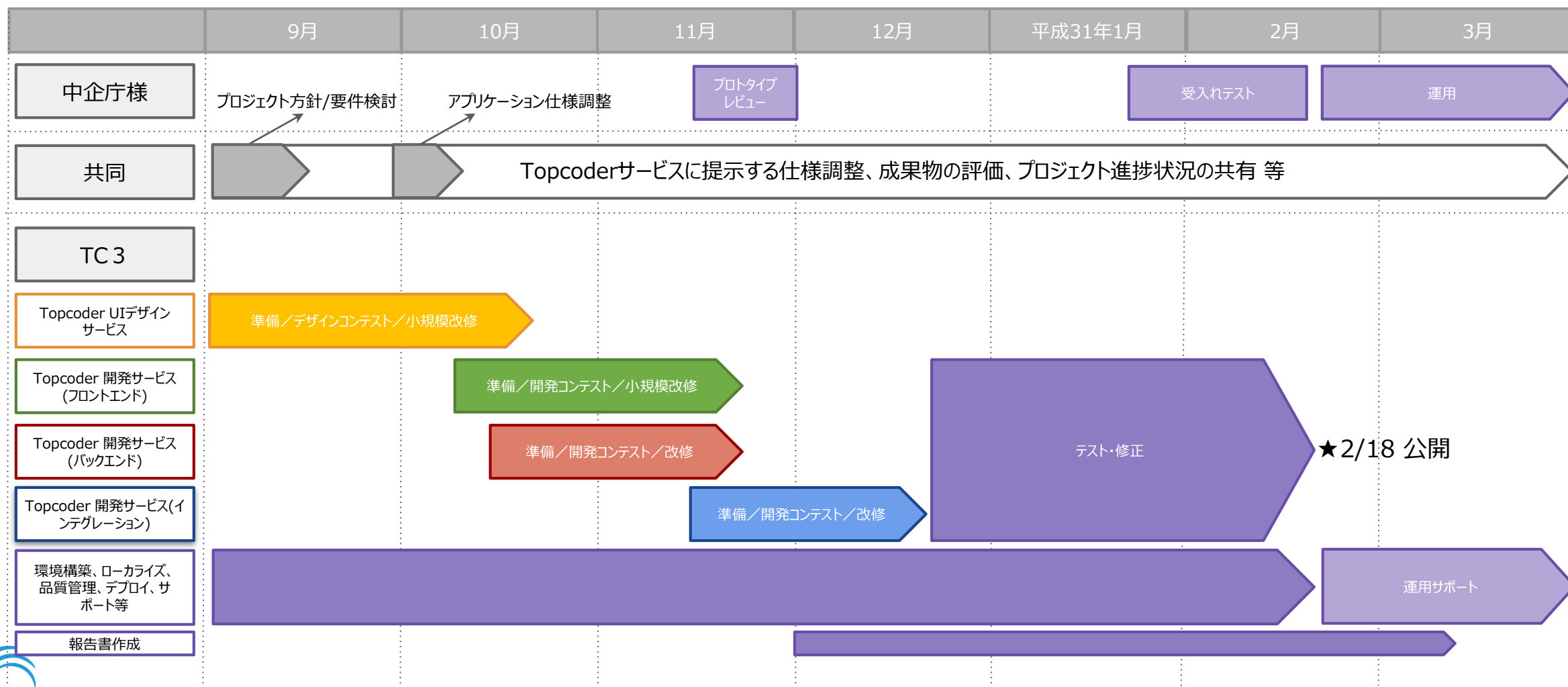
アプリを開発する過程で、ユーザーにとって使いやすいような適切なデザイン、コンテンツ、機能等を整理し、ユーザーインターフェースデザイン案、アプリ構成案を複数案提示の上、中小企業庁デジタルトランスフォーメーション室様と協議し、モデルとなるアプリの開発を行う。

アプリは、スマートフォン、タブレット、パーソナルコンピューター等において利用可能なウェブアプリケーションとし、レスポンスかつGoogle Chrome上で動くことを要件とする。



# 1. 事業の概要

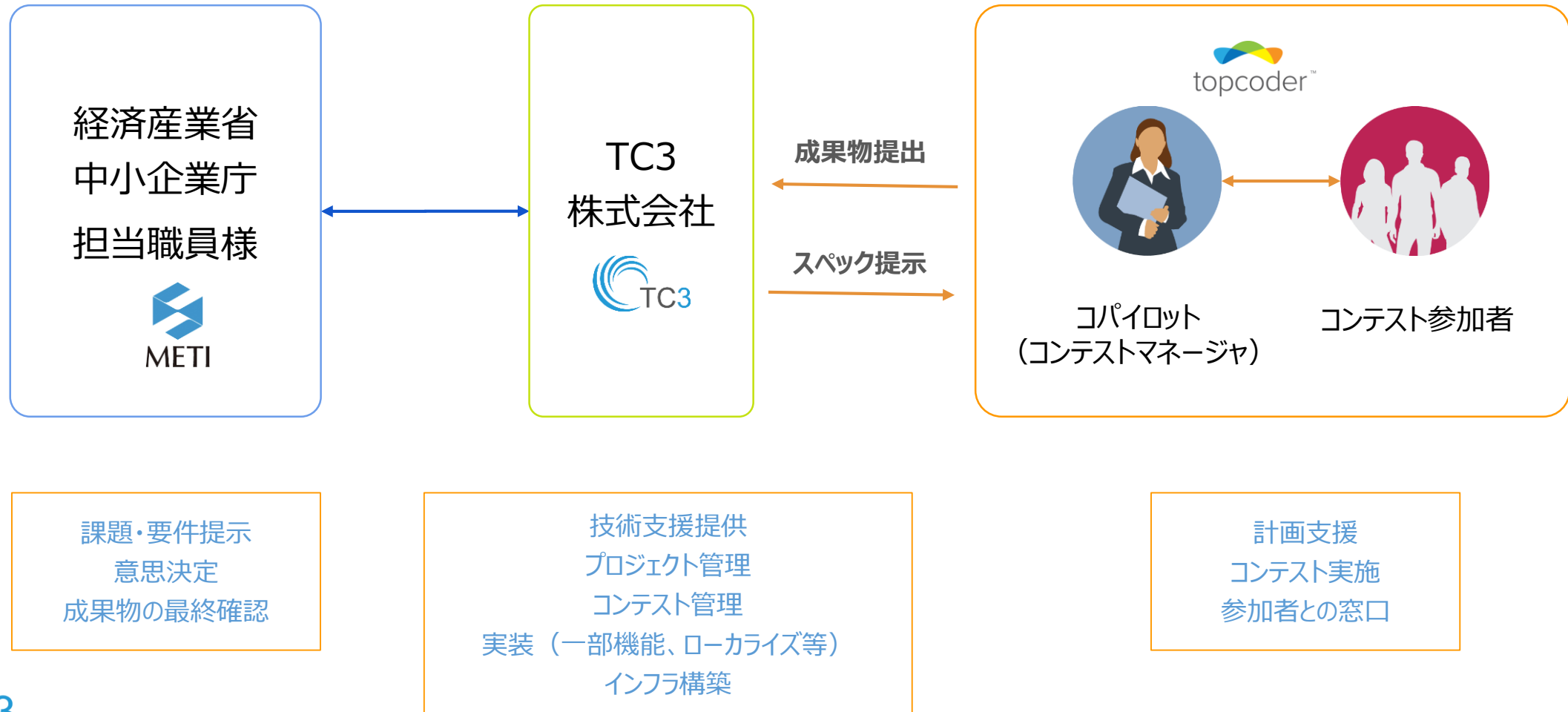
## 1 - 3. 開発スケジュールについて



# 1. 事業の概要

## 1 - 4. 体制について

### (1) 体制図



# 1. 事業の概要

## 1 - 4. 体制について

### (2) プロジェクト進捗管理・担当職員とのコミュニケーションについて

本開発におけるプロジェクト管理業務は、TC3にて担当した。

週次（開発フェーズ以降は隔週）で全体会議を開催し、スケジュールとタスクの状況確認を実施した。

#### Backlogの活用

対面の会議以外でも、担当職員様とのコミュニケーションを円滑に行うために、プロジェクト管理ツールのBacklog

（<https://backlog.com/ja/>）を導入した。これにより、以下のような様々な形態でのコミュニケーションをオンラインで行うことが可能になった。

- タスクの管理および関連するディスカッション
- 課題の管理および関連するディスカッション
- 情報共有および関連するディスカッション
- ファイルの共有および関連するディスカッション

## 2. システム概要

### 2-1. 現状の課題と目指すべき状況

本サービスは、ワンストップで中小企業支援サービスを提供できるプラットフォーム「中小企業支援プラットフォーム(仮称)」のひとつとして位置づけられ中小企業・小規模事業者や中小企業等支援事業者等の経営課題の解決のため各種事例の検索、閲覧を行うサービスである。尚、現状の事例集における課題と目指すべき状況は以下の通り。

#### ユーザにとっての現状と課題 (As Is)

自社に必要な事例集を見つけることができない  
(事例集が施策単位や表彰単位で複数サイトに点在)

事例集の分量も多すぎて、参考になる情報がわからない、パートナーを探すことにも使えない

仮によい事例を見つけられたとしても、該当施策の活用方法（公募情報等）や相談先がわからない

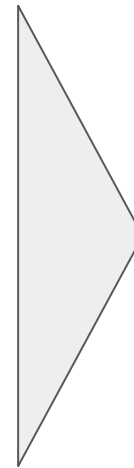
#### アプリ提供によって目指すべき状況 (To be)

個別に構築された事例集をとりまとめ、ニーズに応じた横断的検索が可能な仕組み

施策単位や企業名の検索だけでなく、地域性や企業規模、業種等、自社に近い事例を探ることが可能な仕組み

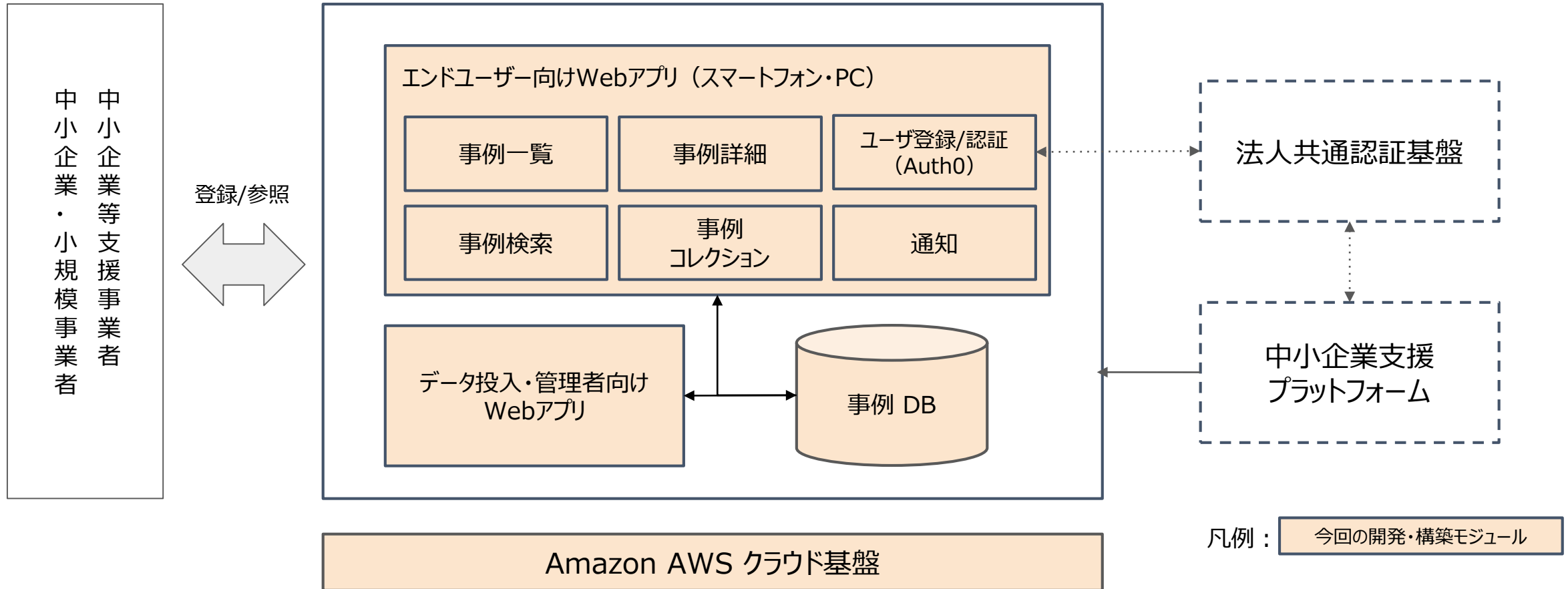
事例内で記載のある支援情報・支援機関にも簡単にアクセスできる仕組み

最終的には、省庁横断、自治体、独法等を巻き込むとともに、連携を目指す



## 2. システム概要

### 2-2. システム概要図・開発範囲





## 2. システム概要

### 2-3. 機能一覧

機能要件をブレイクダウンし、担当職員様との協議の上でそれぞれの内容の詳細化を行った。その結果導出された機能の一覧を以下に示す。

カテゴリ	機能		内容
事例検索アプリ	ホーム	新着事例	データ登録日より直近3件の新着事例を表示する “すべて見る”をクリックすると、事例一覧画面（登録日降順ソート表示）に遷移する
		人気の事例	人気順で上位の事例を3件表示する “すべて見る”をクリックすると、事例一覧画面（人気順ソート表示）に遷移する
		おすすめ事例	おすすめの事例をスコアの降順で表示する
		新着コレクション	データ登録日より直近4件の新着コレクションを表示する
	事例一覧	事例一覧	事例の一覧を6件表示する (1) 絞り込み 絞り込み用ドロップダウンよりカテゴリを選択することで一覧を絞り込む事ができる (2) ソート タブを選択することで一覧の並び順を切り替えることができる
		表示形式変更	下記の2つから表示形式を変更する (1) 3列タイル表示（デフォルト） (2) リスト表示

## 2. システム概要

カテゴリ	機能		内容
事例検索アプリ	事例一覧	事例サマリ情報	事例のサマリ情報を表示する
		もっと見る	「もっと見る」ボタンをクリックすることで、以降の事例が追加で表示される
	事例詳細	事例詳細情報	事例の詳細な情報を表示する 事例そのものの情報以外にも、関連する支援機関・活用した施策へのリンク情報を表示する
		閲覧数更新	事例詳細の閲覧回数を記録する
		いいね	いいねボタンをクリックすると、いいね数が 1 加算され、いいね済みとなる
		SNS投稿 (シェア)	事例ページをSNSに投稿（シェア）する 対応するSNSは、Twitter, Facebook
		コレクション登録	事例をコレクションに登録する
		類似事例表示	類似の事例を表示する
	事例検索	フリーワード検索	キーワードを指定して事例の各種情報を全文検索する キーワードを複数指定された場合は、AND条件として検索を行う
		地域検索(地域、都道府県)	地域、都道府県を選択して事例を検索する 事例企業の所在地情報、および、支援機関の所在地を対象とする
		項目選択形式での検索	項目を選択して事例を検索する。事例に対してあらかじめ設定されたタグを対象とする 項目は複数選択でき、その場合は OR 条件での検索となる

## 2. システム概要

カテゴリ	機能		内容
事例検索アプリ	検索結果	検索結果表示	事例検索画面で指定した条件に該当する事例を一覧表示する 結果一覧の事例をクリックすると、画面右側にその事例のサマリ情報を表示する 一覧下部の“もっと見る”ボタンをクリックすると、以降の結果の20件が追加で表示される
	コレクション	コレクション一覧	コレクションの一覧を表示する
		コレクション管理	コレクションに対し、作成／削除／変更操作を行うことができる
		事例一覧表示	コレクションに含まれる事例を登録日昇順で一覧表示する
		事例の削除	コレクションに登録されている事例を削除する
		コレクションの一括印刷	コレクションに含まれる事例を一括で印刷する 印刷用のページを別ウィンドウで表示し、ブラウザの印刷機能で印刷を行う
		コレクション内検索	コレクション内の事例に対して、検索を行う。検索はシンプルなキーワード検索とする
		SNS投稿（シェア）	コレクションページをSNSに投稿（シェア）する 対応するSNSは、Twitter, Facebook
		おすすめ事例	コレクション内の事例と類似した事例を表示する
	通知	未読通知数表示	ヘッダの通知アイコンに、未読数のインジケータを表示する
		通知一覧	お知らせの一覧を表示する

## 2. システム概要

カテゴリ	機能		内容
事例検索アプリ	ナビゲーションメニュー	ホーム	ホームを表示する
		事例一覧	事例一覧を表示する
		コレクション	コレクション一覧ページを表示する
		お知らせ	お知らせ一覧ページを表示する ユーザーログイン時は未読お知らせ数を表示する
ユーザーメニュー	ユーザー登録	ユーザー登録	ユーザー登録フォームを表示し、メールアドレス／パスワードで登録を行う
		外部アカウントで登録	外部サービスのアカウントで認証が行われた後、ユーザーが登録される 対応サービスは、法人共通認証基盤、Facebook、Twitter
		メールアドレス確認	ユーザー登録後、確認メールによりユーザー確認を行う ソーシャルアカウントの場合は、本人確認は行われしない。（SNS側で確認済みとする）
	ログイン	メールアドレス／パスワード入力	ログインフォームを表示する。登録メールアドレスとパスワードを入力して、認証される
		ソーシャルアカウント	ログインフォームの各種SNSアイコン押下によりSNSで認証を行う
	ログアウト		システムからログアウトされる

## 2. システム概要

カテゴリ	機能	内容
ユーザーメニュー	ユーザー情報変更	ユーザーの情報を変更する
管理者機能	事例情報マスタメンテナンス	事例情報を管理(作成/変更/削除)する
	企業情報マスタメンテナンス	企業情報を管理(登録/変更/削除)する
	お知らせマスタメンテナンス	お知らせ情報を管理(作成/変更/削除)する



## 3. コンテストを活用したアプリケーション開発

### 3-1. プロジェクト方針／要件検討

#### (1) プロジェクト方針について

本プロジェクトは以下の方針をもとに実施した。

プロジェクト全体	本プロジェクトは、プロジェクト全体や他モジュールの開発状況に応じて機動的なプロジェクト推進を可能とするため、アプリケーション開発をUIデザイン、フロントエンド開発、バックエンド開発、ローカライズ開発・品質管理・デプロイ等の各種別に分類し、各分類の開発を数日～2週間程度の短い期間を単位としたサイクルを回す形式とする。 また、必要に応じて中間成果物のフィードバック、コミュニケーションを実施し、機能の確認、修正に耐える方式で進めることとする。
要件検討、確認	必要最低限の検討とし、Topcoderサービスを利用することでUI/UXデザインを含めたアプリケーションの整理、具現化を実施していくこととする。
UI/UXデザイン及びアプリケーション開発	Topcoderサービスを利用し、コンテスト形式で行うこととする。これによりコミュニティから複数の成果物を得ることができ、より満足度の高いデザイン、品質の良いプログラムが採用可能となる。

※Topcoderシステムは、数日から2週間程度のコンテストを繰り返し実施することで、アジャイルでクイックなプロジェクトの進行を可能とすることから、今回の事業目的を達成するために適切であると考え、利用することとした。Topcoderについての詳細はP72「参考. Topcoderとは」参照。

# 3. コンテストを活用したアプリケーション開発

## 3-1. プロジェクト方針／要件検討

### (2) 要件検討について

以下の「要件検討」「SPEC準備」についての確認、検討を実施した。

要件検討	現行の事例集（PDF等）をベースに、事例画面に表示する項目、内容を整理する。 他システム共通の項目（データ）を洗い出し、データ構造、設定値を明確化する。 システム化するにあたり、重要視する機能における仕様を検討する。 （検索、コレクション等）		
	・事例分類などの検討 ・データ定義	・事例項目の洗い出し ・機能一覧	・その他
SPEC準備	デザインコンテスト仕様を検討、まとめる。		
	・ペルソナの定義 ・制約条件（画面サイズ等）	・サイトコンセプト検討 ・判定基準	・デザイン機能の定義 ・その他

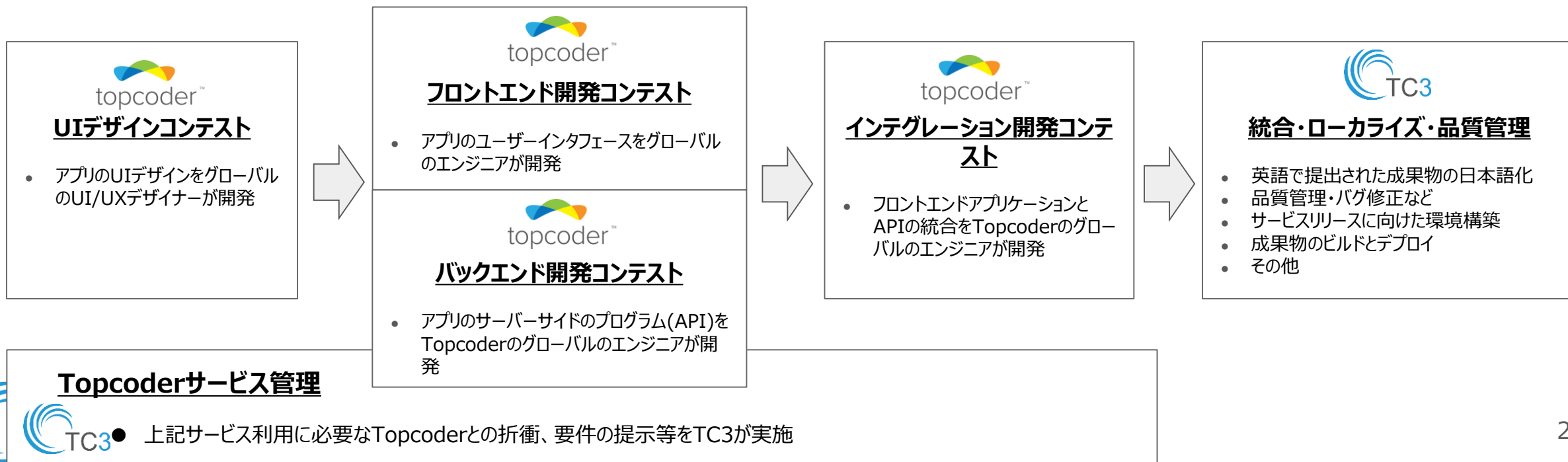
# 3. コンテストを活用したアプリケーション開発

## 3-2. アプリケーション開発の進め方

下記の流れに沿って、アプリケーション開発を遂行する。

- Topcoderコンテストにより、UIデザイン、フロントエンドアプリケーション、バックエンドAPIを短期間で高品質に開発する。
- TC3により、Topcoderサービスの管理、その成果物の統合／ローカライズ／品質管理を行う。
- TC3により、プロジェクト管理を行う。

### TC3プロジェクト全体管理

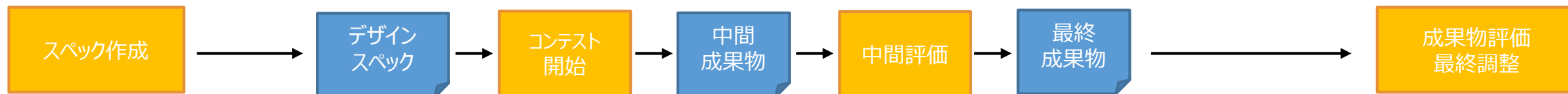


## 3. コンテストを活用したアプリケーション開発

### 3-3. UIデザインコンテスト実施報告

#### (1) コンテストの進め方

- デザインに求める要件を反映したコンテストスペックを作成し、コンテストを実施した。
- 中間チェックポイントを設け、中間レビューを実施した。
- 最終成果物に対する評価を実施し、最終成果物を獲得した。



#### スペック作成

- TC3とTopcoderコミュニティのコンテストマネージャにてコンテストへの要求事項（スペック）を作成
- 担当職員様がスペックをレビュー
- スペックに記載する主要項目
  - プロジェクト全体像
    - 誰に対してどのような価値を提供するものか
    - ワイヤーフレーム
    - 補足資料の説明
  - 評価基準
  - 成果物要件

#### コンテスト開始

- スペックが公開され、コンテストがスタート
- 参加者の質問に対して一次解答をコンテストマネージャ、もしくはTC3が実施（必要に応じて担当職員様にご相談）

#### 中間評価

- 通常、コンテスト開始から4～6日後に中間評価の時間を設定し、中間評価タイミングから48時間後までを目安にフィードバック提供を実施
- 方向性の修正
  - 成果物のサマリー・結論を確認
  - 指摘を各参加者へフィードバック

#### 成果物評価・最終調整

- 事前に提示した評価基準に沿ってTC3の協力の元、担当職員様が評価を行い順位を決定
- 成果物は想定アプリ画面の画像ファイル一式（Photoshop など）
- 成果物に対しての微調整や軽微な修正が可能（ファイナルフィックス）

# 3. コンテストを活用したアプリケーション開発

## 3 - 3. UIデザインコンテスト実施報告

### (2) コンテスト実施概要

#### コンテストページ

- ❖ 「METI - Case Studies Responsive Application Design Challenge」

#### コンテストスケジュール（実績）

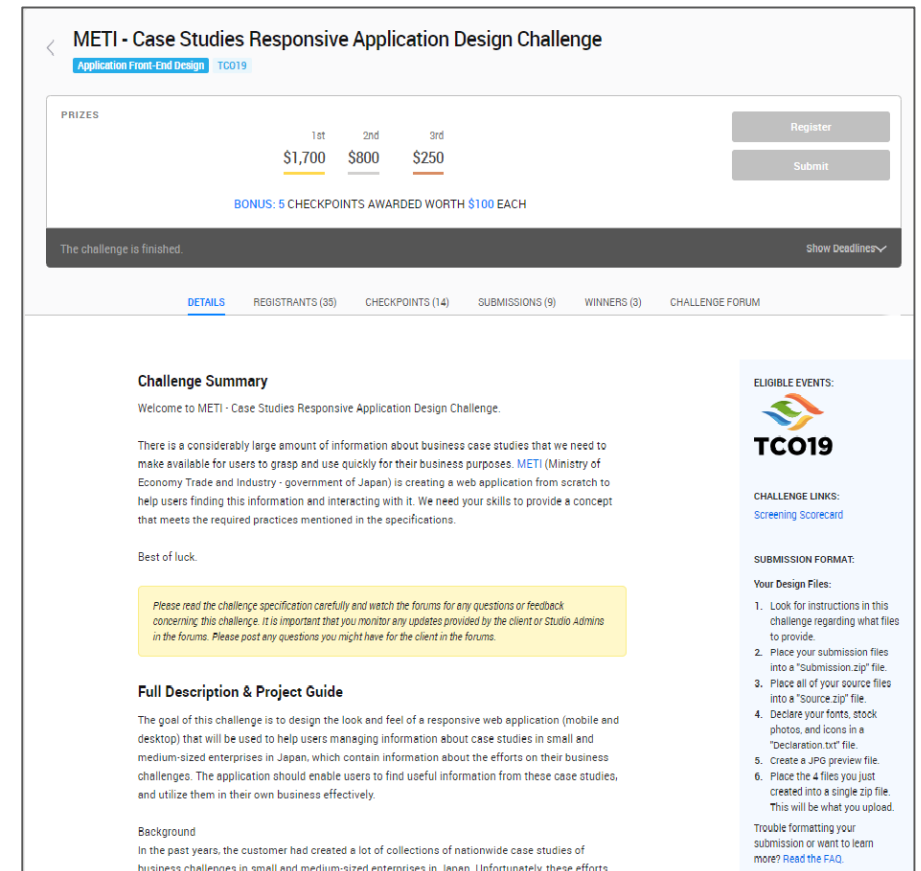
- ❖ 実施期間: 9/7 ~ 9/25（約14日間）

#### 登録者／成果物（実績）

- ❖ 登録: 35名
- ❖ 成果物: 9件

#### コンテスト対象画面

1. ログイン・ユーザー登録画面 (PC)
2. 事例一覧画面 (PC／スマートフォン)
3. 事例検索画面 (PC)
4. 新着事例表示画面 (PC)
5. 履歴表示画面 (PC)
6. 事例詳細表示画面 (PC／スマートフォン)
7. ブックマークした事例表示画面 (PC)
8. 事例新着画面 (PC)
9. ユーザー情報編集画面 (PC)

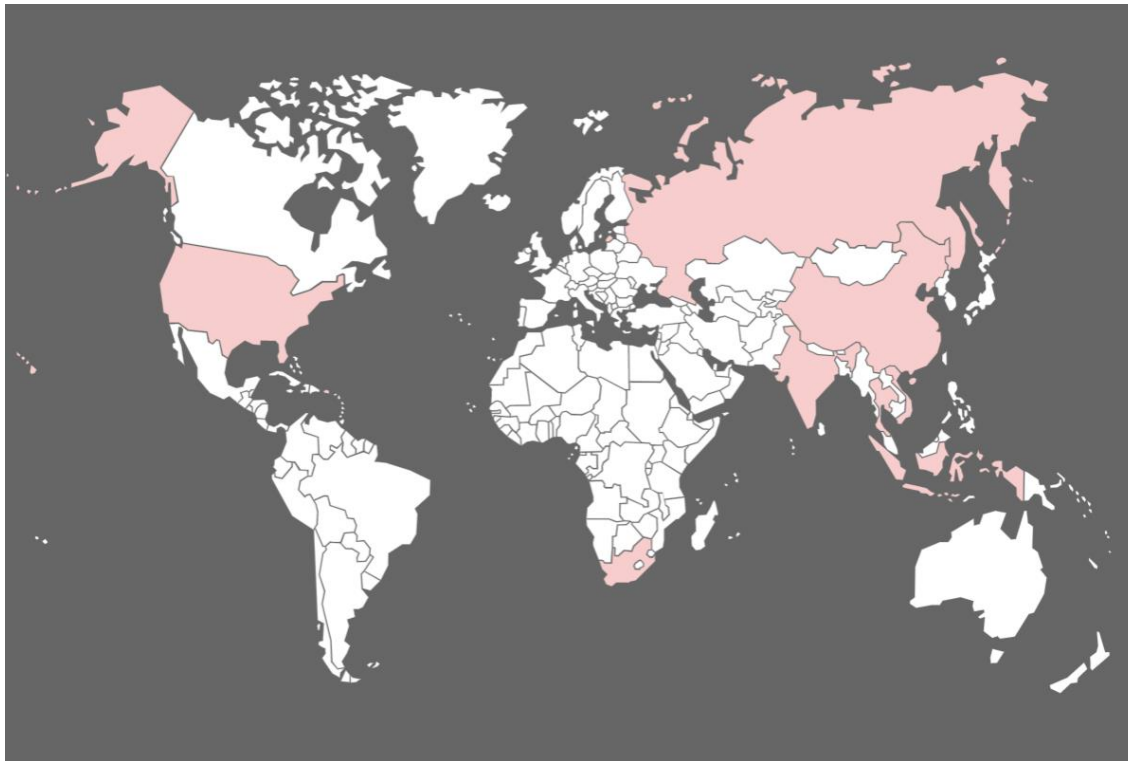




## 3. コンテストを活用したアプリケーション開発

### 3 - 3. UIデザインコンテスト実施報告

#### (3) コンテスト参加者



#### 【コンテスト参加者】

・35名

#### 【出身国（7ヶ国）】

・インド ・中華人民共和国  
・南アフリカ共和国 ・アメリカ合衆国  
・タイ王国 ・ベトナム社会主義共和国  
・ロシア連邦

#### 【成果物提出者】

・中間チェックポイント: 14名  
・最終: 9名

## 3. コンテストを活用したアプリケーション開発

### 3 - 3. UIデザインコンテスト実施報告

#### (4) UIデザインコンテスト中間評価の目的と方法

作成途中の成果物に対しフィードバックを行うことで、  
**最終成果物を期待に沿った内容、品質となるように調整すること**を目的とする。

##### 評価にて実施すべきこと

1. 全体的なフィードバック（共通的な指針）
2. 個別のフィードバック
  - 個別のフィードバックも全員が閲覧可能（個々の参加者とのプライベートなコミュニケーションは不可）。また、ネタばらしのようなフィードバックは不可。
3. 中間評価の勝者選出
  - 5名の勝者を選出する。勝者にはそれぞれ賞金が支払われる。
  - 中間評価の賞金は最終成果物を提出しないと受け取ることができない。最終成果物を出して欲しい（期待している）成果を選出するようにする。

## 3. コンテストを活用したアプリケーション開発

### 3-3. UIデザインコンテスト実施報告

- ・ **公正であること**

- スペックと異なる指針を提示することはできません。
- (特定の) 参加者の不利益となるフィードバックは不可。

- ・ **指針を示す**

- 良い点を示す
- 改善点を示す
  - こうしたら良くなる - より要望に沿ったものになるように促す。
  - 方向性が明らかにずれている - スペックが誤解されていることを指摘する。

- ・ 誉めて伸ばす

- 「ここが悪い」よりも「素晴らしいが、こうすると更に良くなる」という指摘の方が良い結果に繋がる。  
→ 参加者は「自分の成果物は評価が悪いかもしれない」と感じると中間評価後に離脱してしまうケースがあるため。

- ・ 指摘の目的を意識する

- 指針とは無関係な指摘は行わないようにする。(参加者が混乱してしまう可能性があるため)

- ・ 具体的な指摘を行う

- 指摘内容を具体的に示す方が、抽象的な指摘よりもより良い結果に繋がる。

## 3. コンテストを活用したアプリケーション開発

### 3-3. UIデザインコンテスト実施報告

中間評価では 3 Pass Review の実施を推奨。

Pass1 : 全体の把握

Pass2 : 個別の確認

Pass3 : 総括

#### Pass 1 : 全体の把握

提出物全体を見渡し、内容・品質の全体感を把握します。その後、「評価に値する」「評価に値しない」の 2 グループに分類。

※次の個別確認に進む前に、レビュアーが3件ずつ投票し対象を絞り込む。（最低5件以上になるようにする）

#### Pass 2 : 個別の確認

「評価に値する」グループの提出物について、それぞれの詳細を確認。（スペックを的確に捉えているか？）

「評価に値しない」グループの提出物について全体的な特徴を確認。（何故評価に値しないのか？）

#### Pass 3 : 総括

スペックに記載した要件の観点に照らし合わせ、提出物を確認する。

提出物の中で「優れていると感じるポイント」、「ずれていると感じるポイント」をあげる。

※絞り込んだ中から、更にレビュアーが3件ずつ投票し、勝者を決定する。

# 3. コンテストを活用したアプリケーション開発

## 3 - 3. UIデザインコンテスト実施報告

### (5) UIデザインコンテスト中間評価

- 中間評価会議の実施
  - 日時：2018年9月13日（木）14:00 - 16:30
  - 場所：経済産業省
  - 中間評価の主な目的
    - 成果物の軌道修正
      - 適切なフィードバックを行うことで、最終成果物の品質の向上をはかる
      - フィードバックは参加者全員に共有される
    - 中間チェックポイント時点での優秀成果物決定（5点）

会議風景





## 3. コンテストを活用したアプリケーション開発

### 3-3. UIデザインコンテスト実施報告

- 中間評価会議の実施内容

#	実施内容	実施者
1	提出物の印刷物（A3用紙）を壁に貼り出し	TC3
2	中間評価の目的と評価方法についてレビューへの説明	TC3
3	各提出物を見て、どのようなものがあがってきたのかを全体的に確認	レビューア
4	各レビューアが、個々の提出物に対して点数付けを行う。レビューアには決まった数の色分けされた付箋が配布され、印刷物に貼り付けて投票する形式とした。（評価内容は付箋へのコメントとして記載）	レビューア
5	各提出物について評価の内容を全員で共有。追加で良い点・改善点について意見を出し合う。	レビューア
6	休憩	—
7	付箋を集計し、上位5件の成果物を確定。	TC3
8	レビュー会議のクロージング	TC3
9	入賞者とレビューでのフィードバックを取りまとめコンテストマネージャに伝達	TC3

## 3. コンテストを活用したアプリケーション開発

### 3 – 3. UIデザインコンテスト実施報告

#### (6) 中間評価結果のフィードバックについて

- 全般的なフィードバック（抜粋）

#	フィードバック内容（日本語訳）
1	包括的なデザインとなっているようですが、情報アーキテクチャの観点でより改善、つまりさまざまなデータを魅力的でわかりやすい方法で表示する方法が求められています。
2	事例一覧は、もっと興味を引くような（且つ整理された）デザインが求められています。画像をうまく活用してみてください。
3	事例詳細は、もっと興味を引くようなデザインにできると思います。単に全ての情報を1ページに詰め込むだけでは良いアイデアとは言えないと思います。知りたい情報を探すのに苦労するようなものになってしまうでしょう。コンテンツをうまく分割することを検討してください。
4	フォントサイズに注意してください。ほとんどのデザインにおいて、年配の人が読むには文字が小さいように思えます。
5	検索機能のデザインについてはじっくり検討して頂きたいと思います。検索条件の指定の仕方についてのUXには、もっと良いアイデアを出してもらう必要があります。

## 3. コンテストを活用したアプリケーション開発

### 3-3. UIデザインコンテスト実施報告

- 各提出物への個別のフィードバック（抜粋）

提出物	デザイン	フィードバック内容（日本語訳）
261524	全体	<ul style="list-style-type: none"><li>良いスタートだと思います。</li><li>全体的にユーザーフレンドリーなデザインになっています。</li><li>色遣いが良いと思います。</li></ul>
	事例一覧	<ul style="list-style-type: none"><li>行の高さが不揃いなのが気になるので、揃えてほしいと思います。</li></ul>
	検索	<ul style="list-style-type: none"><li>もっと詳細がわかるようなデザインにしてください。</li></ul>
	事例詳細	<ul style="list-style-type: none"><li>どのようにして事例の情報を見るか、操作方法が直感的にわからなかった。</li><li>選択リストはどのように使うのでしょうか？</li></ul>
	ブックマーク	<ul style="list-style-type: none"><li>とても印象的なデザインだと思います。YouTubeのプレイリストのようで、操作になじみやすさがある。</li><li>フォルダ分けの仕方が見やすくて良いです。</li><li>もっと詳細がわかるようなデザインにしてください。</li></ul>

# 3. コンテストを活用したアプリケーション開発

## 3 - 3. UIデザインコンテスト実施報告

### (7) UIデザインコンテスト最終評価

- 開催概要
  - 日時：2018年9月25日（火）14:00 - 17:00
  - 場所：経済産業省
  - 最終評価の主な目的
    - 最終的な勝者を決定する
    - ダイレクトタスクを行うための変更点、修正点のヒアリング

会議風景



## 3. コンテストを活用したアプリケーション開発

### 3-3. UIデザインコンテスト実施報告

- 最終評価会議の実施内容

#	実施内容	実施者
1	提出物の印刷物（A3用紙）を壁に貼り出し	TC3
2	最終評価の目的と評価方法についてレビューへの説明	TC3
3	各提出物を見て、どのようなものがあがってきたのかを全体的に確認	レビュー
4	TC3より各提出物について概要を説明し、レビューより意見を集める	TC3、 レビュー
5	各レビューが、提出物に対して投票を行う。レビューには付箋が配布され、印刷物に貼り付けて投票する形式とした。（評価内容は付箋へのコメントとして記載）	レビュー
6	投票数により提出物に順位をつける 今回、1、2位と3、4位の提出物が同得票だったため、レビューの多数決で決定した	レビュー
7	レビュー会議のクロージング	TC3
8	入賞者をコンテストマネージャに伝達	TC3

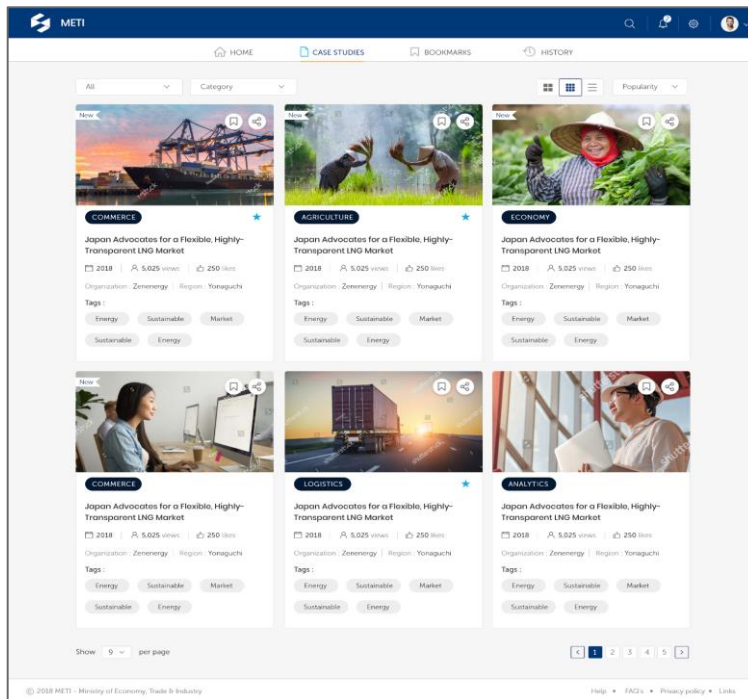
# 3. コンテストを活用したアプリケーション開発

## 3-3. UIデザインコンテスト実施報告

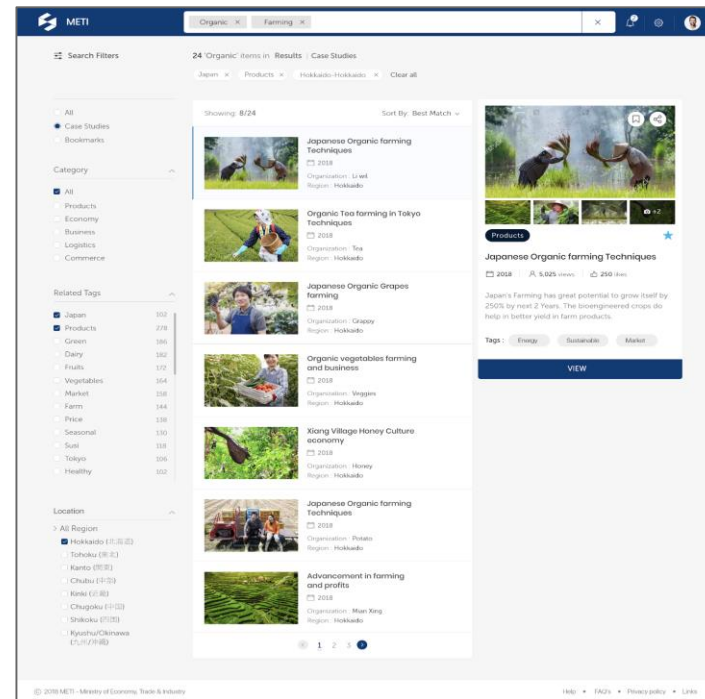
### (8) コンテストの成果物

- 1位のデザイン（一部抜粋）

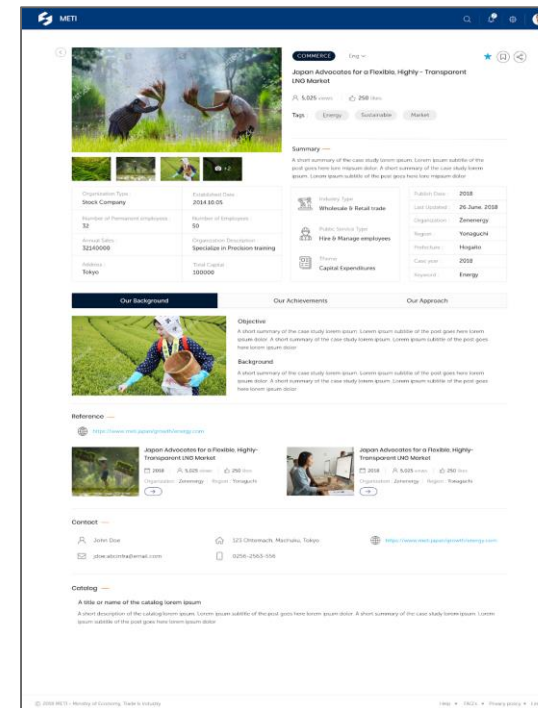
【事例一覧】



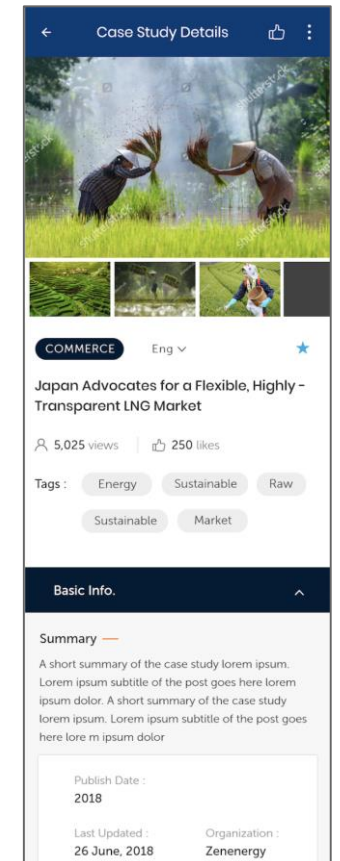
【検索結果】



【事例詳細】



【モバイル】



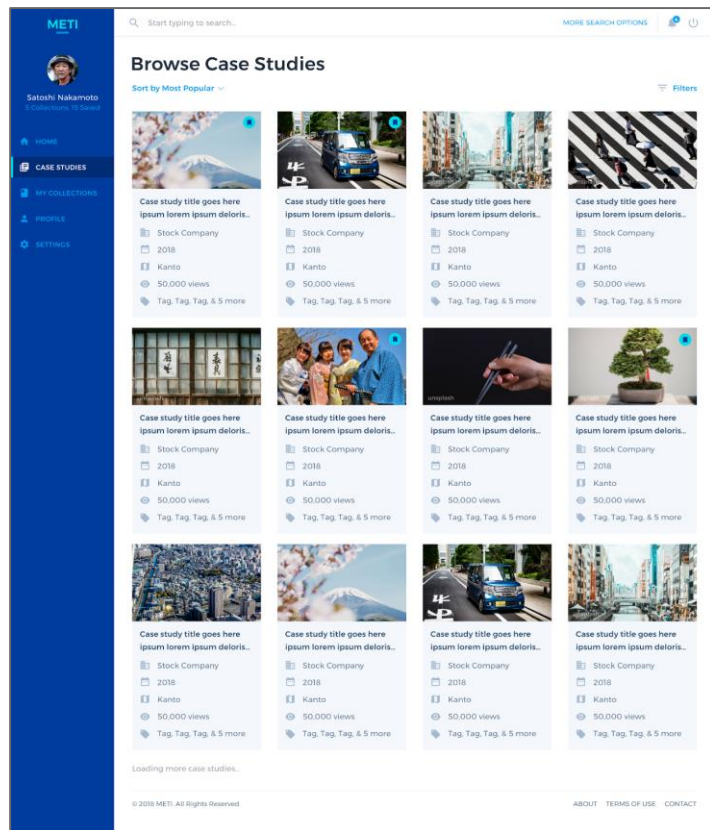


# 3. コンテストを活用したアプリケーション開発

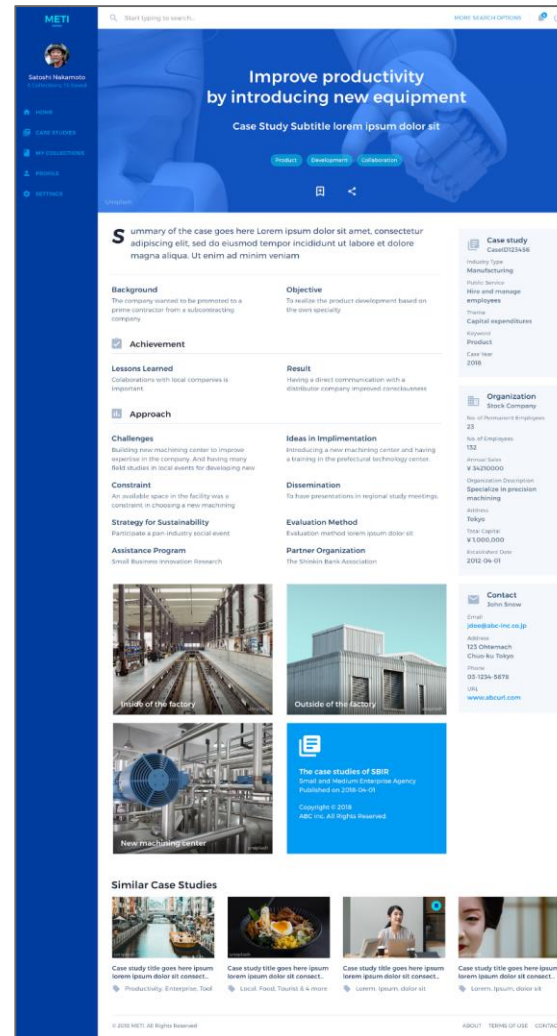
## 3-3. UIデザインコンテスト実施報告

- 2位のデザイン（一部抜粋）

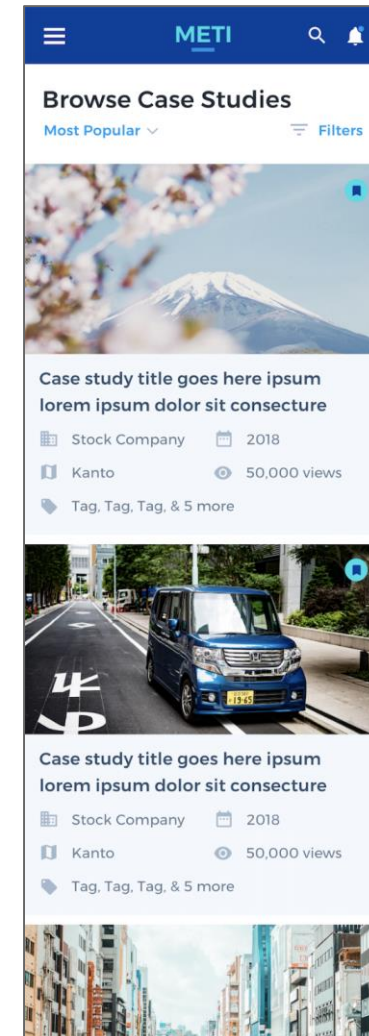
【事例一覧】



【トップページ】



【モバイル】





# 3. コンテストを活用したアプリケーション開発

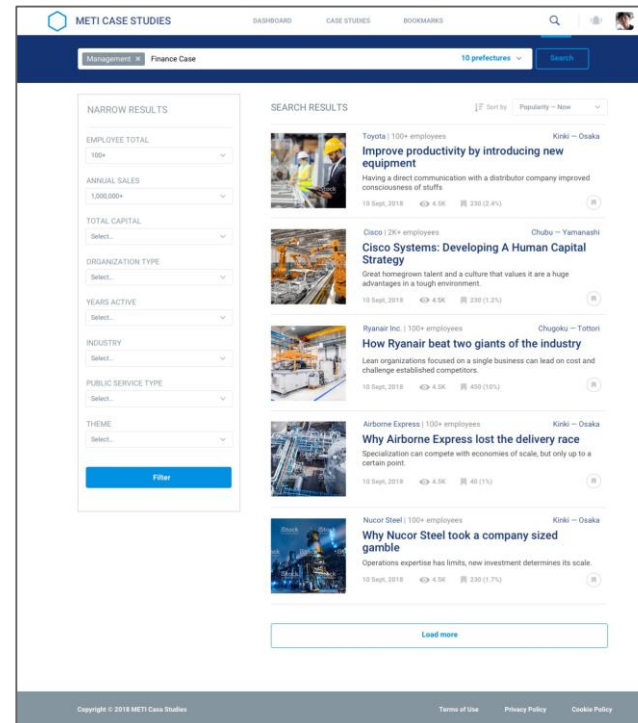
## 3-3. UIデザインコンテスト実施報告

- 3位のデザイン（一部抜粋）

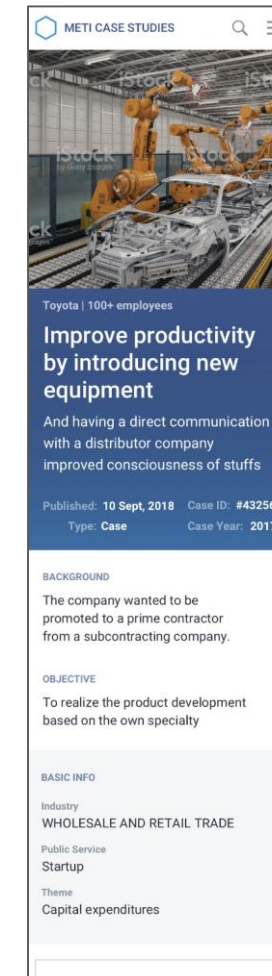
【トップページ】



【検索結果】



【モバイル】

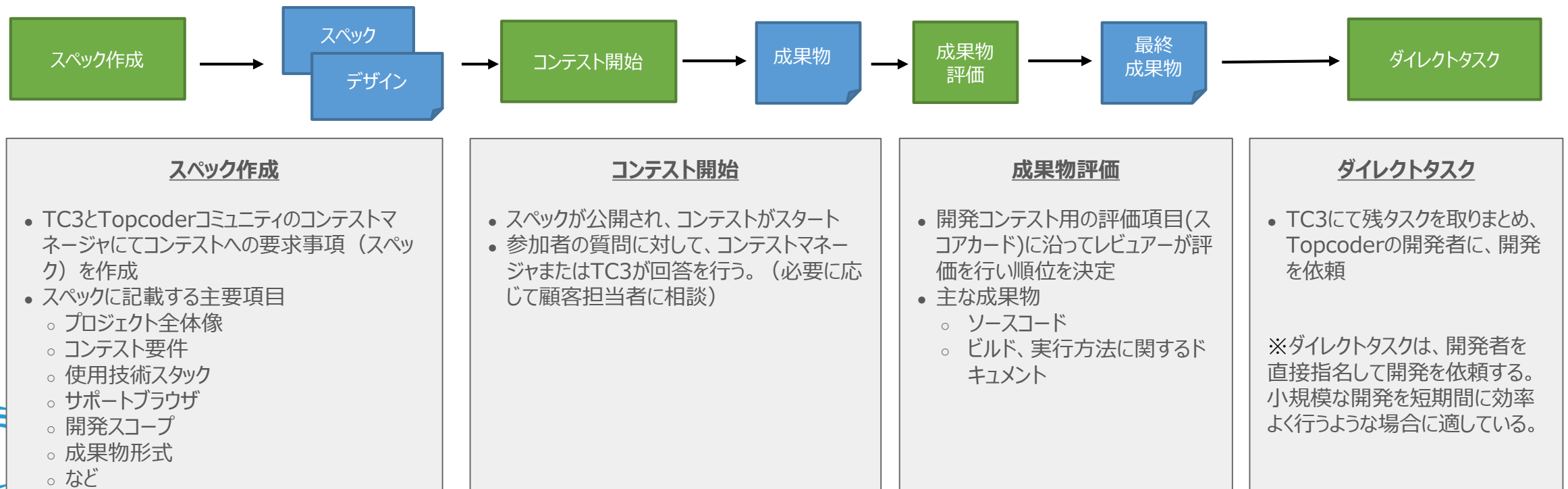


# 3. コンテストを活用したアプリケーション開発

## 3-4. フロントエンド開発コンテスト実施報告

### (1) フロントエンド開発コンテストの進行について

- アプリケーションのデザインおよび機能要件からコンテストスペックを作成し、コンテストを実施した。
- コンテスト開始後は、参加者との質疑応答を通して、開発中の問題点を解消した。
- コンテストにアサインされたレビュアーが成果物を評価し、上位の成果物を最終成果物として獲得した。
- スcopeに入らなかった機能、または必要な修正について、ダイレクトタスクによる開発を実施した。



# 3. コンテストを活用したアプリケーション開発

## 3 - 4. フロントエンド開発コンテスト実施報告

### (2) コンテスト実施概要①

#### フロントエンドアプリケーション

- ❖ ブラウザで動作するアプリケーション。主に HTML/CSS/Javascript で構成される。

#### UIプロトタイプ開発

- ❖ デザインからフロントエンドアプリケーションを開発する。コンテストスペックと補足資料を準備。
- ❖ ユーザーインターフェースにフォーカスし、バックエンドサービスとの連携は行わない。
  - データはダミーデータを使用する。ただし、アプリケーションコードにハードコードせずに、JSON形式のデータファイルから読み込む方式を指定。データフォーマットは、API仕様（Swaggerドキュメント・補足資料）で提供。
- ❖ 主要フレームワーク製品であるAngularを採用。

#### 開発スコープ

- ❖ 以下の8機能を対象とした。
  - ホーム画面
  - 事例一覧画面
  - 事例詳細画面
  - コレクション一覧画面
  - コレクション詳細画面
  - 検索BOX
  - 検索結果画面
  - 通知

# 3. コンテストを活用したアプリケーション開発

## 3 - 4. フロントエンド開発コンテスト実施報告

### (3) コンテスト実施概要②

#### コンテストページ

- ❖ 「METI - Case Studies Responsive Application Desktop & Tablet Prototype」

(<https://www.topcoder.com/challenges/30070715>)

#### コンテストスケジュール (実績)

- ❖ 実施期間: 10/24 ~ 11/3 (約9.5日間)
  - 開始 ~ 提出期限: 7日間
  - レビュー/アピール期間: 2.5日間

#### 登録者/成果物

- ❖ 登録: 52名
- ❖ 成果物: 1件
- ❖ 賞金獲得者: soso0574 (スコア: 91.83)

#### コンテスト運営 (コミュニティ)

- ❖ コパイロット (コンテストマネージャ): Wendell
- ❖ レビュアー: Wendell/iversonLv

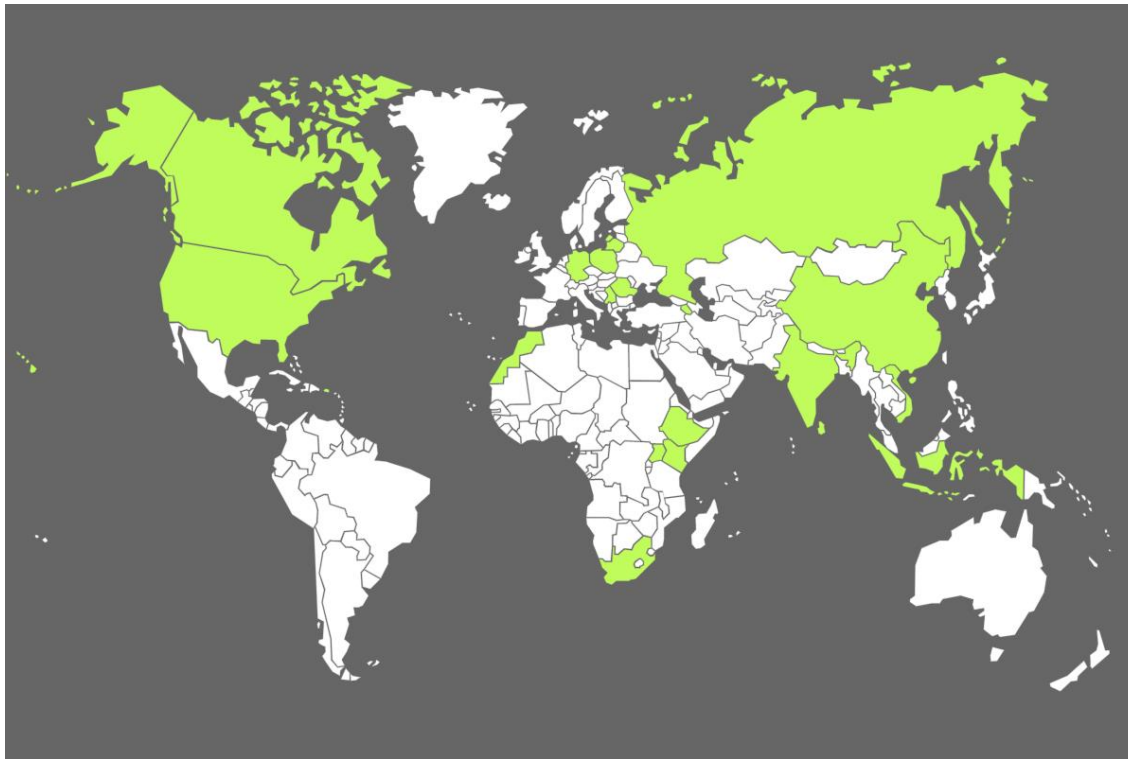
The screenshot shows the TopCoder challenge page for 'METI - Case Studies Responsive Application Desktop & Tablet Prototype'. The page is in English and includes the following sections:

- PRIZES:** A table showing prize amounts for 1st (\$1,600) and 2nd (\$800) places. Buttons for 'Register' and 'Submit' are visible.
- Challenge Overview:**
  - Project Background:** A paragraph explaining the challenge's purpose: to create a web application for the METI project using AngularJS 6 / HTML5 / CSS.
  - Challenge Objectives:** A paragraph stating the goal is to build the UI for the METI project using AngularJS 6 / HTML5 / CSS.
  - Technology Stack:** A list of technologies: AngularJS 6, HTML5, Javascript, and CSS.
  - API Integration:** A paragraph explaining that the API is not ready yet, and participants need to implement simple controllers to fetch data from JSON files.
  - Challenge Scope:** A paragraph stating that the screens listed below are in scope of the challenge.
  - General Requirements:** A list of requirements, including: 'It's very important that you check the supporting documents in the forum since they contain critical details to the implementation.', 'The data to be displayed must come from local JSON files and cannot be hardcoded.', 'Search option values should come from the spreadsheet we shared in the forum, do NOT just use some dummy values.', and 'The prototype must support localization using Angular localization best practices (for example: ...)'.
- ELIGIBLE EVENTS:** A section with the TCO19 logo and a 'REVIEW STYLE' section indicating 'Final Review: Community Review Board'.
- CHALLENGE LINKS:** A section with links for 'Review Scorecard' and 'Challenge Terms'.
- SHARE:** A section with social media sharing icons for Facebook, Twitter, LinkedIn, and YouTube.

## 3. コンテストを活用したアプリケーション開発

### 3-4. フロントエンド開発コンテスト実施報告

#### (4) コンテスト参加者



##### 【コンテスト参加者】

・52名

##### 【出身国（19ヶ国）】

・インド共和国 ・中華人民共和国 ・ルーマニア  
・セルビア共和国 ・ベトナム社会主義共和国  
・アメリカ合衆国 ・カナダ ・南アフリカ共和国  
・ロシア連邦 ・インドネシア共和国  
・スリランカ民主社会主義共和国 ・ケニア共和国  
・アルメニア共和国 ・ポーランド共和国  
・エチオピア連邦民主共和国 ・モロッコ王国  
・ウガンダ共和国 ・リトアニア共和国  
・ドイツ連邦共和国

##### 【成果提出者】

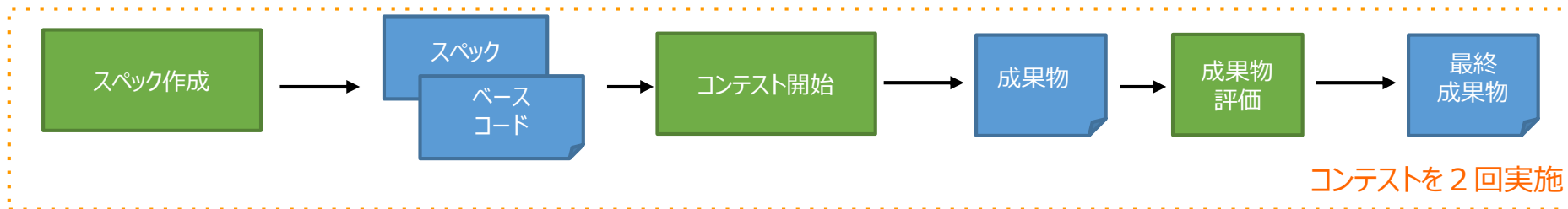
・1名

## 3. コンテストを活用したアプリケーション開発

### 3-5. バックエンド開発コンテスト実施報告

#### (1) バックエンド開発コンテストの進行について

- アプリケーションの機能要件からコンテストスペックを作成し、コンテストを実施した。
- コンテスト開始後は、参加者との質疑応答を通して、開発中の問題点を解消した。
- コンテストにアサインされたレビュアーが成果物を評価し、上位の成果物を最終成果物として獲得した。
- ダイレクトタスクの開発者を確保できなかったが、コンテストをもう一度実施することで対応した。



#### スペック作成

- TC3とTopcoderコミュニティのコンテストマネージャにてコンテストへの要求事項（スペック）を作成
- スペックに記載する主要項目
  - プロジェクト全体像
  - コンテスト要件
  - 使用技術スタック
  - 開発スコープ
  - 成果物形式
  - など

#### コンテスト開始

- スペックが公開され、コンテストがスタート
- 参加者の質問に対して、コンテストマネージャまたはTC3が回答を行う。（必要に応じて顧客担当者に相談）

#### 成果物評価

- 開発コンテスト用の評価項目（スコアカード）に沿ってレビュアーが評価を行い順位を決定
- 主な成果物
  - ソースコード
  - テストスクリプト
  - ビルド、実行方法に関するドキュメント



# 3. コンテストを活用したアプリケーション開発

## 3 – 5. バックエンド開発コンテスト実施報告

### (2) コンテスト実施概要① (コンテスト 1 回目)

#### バックエンドサービス (API)

- ❖ アプリケーションの動作に必要なデータ入出力機能を、REST APIで提供する。
- ❖ データベースまたは検索エンジンと連携する。また、ユーザー認証をベースにアクセスコントロールを行う。

#### バックエンドサービス開発

- ❖ コンテストスペック、データベーススキーマ、API仕様 (Swaggerドキュメント・補足資料) を準備
- ❖ Java/SpringBootベースでの実装。プロダクションコード・ユニットテストコードを開発。

#### 開発スコープ

- ❖ 以下の13 APIを対象とした。

➢ 事例取得	[GET]	/case_studies/{id}
➢ 事例Like	[POST]	/case_studies/{id}/like
➢ 閲覧履歴	[POST]	/case_studies/{id}/visit
➢ コレクション検索	[GET]	/collections?owner_id={owner_id}&sort=last_modified_at&order=desc
➢ コレクション取得	[GET]	/collections/{id}
➢ コレクション作成	[POST]	/collections
➢ コレクション更新	[PUT]	/collections/{id}
➢ コレクション削除	[DELETE]	/collections/{id}
➢ コレクション事例一覧取得	[GET]	/collections/{id}/items?sort=last_modified_at&order=desc
➢ コレクションへの事例追加	[POST]	/collections/{id}/items/add?item_id={item-id}
➢ コレクションへの事例削除	[DELETE]	/collections/{id}/items/{item-id}
➢ お知らせ未読数取得	[GET]	/notifications/status
➢ お知らせ一覧取得	[GET]	/notifications



# 3. コンテストを活用したアプリケーション開発

## 3 - 5. バックエンド開発コンテスト実施報告

### (3) コンテスト実施概要② (コンテスト 1 回目)

#### コンテストページ

- ❖ 「METI - Case Studies Responsive Application REST API」  
(<https://www.topcoder.com/challenges/30073051>)

#### コンテストスケジュール (実績)

- ❖ 実施期間: 10/27 ~ 11/4 (約7.5日間)
  - 開始 ~ 提出期限: 5日間
  - レビュー/アピール期間: 2.5日間

#### 登録者/成果物

- ❖ 登録: 59名
- ❖ 成果物: 5件
- ❖ 賞金獲得者
  - 1位 ngoctay (スコア: 96.75)
  - 2位 lf.coder040 (スコア: 90.5)

#### コンテスト運営 (コミュニティ)

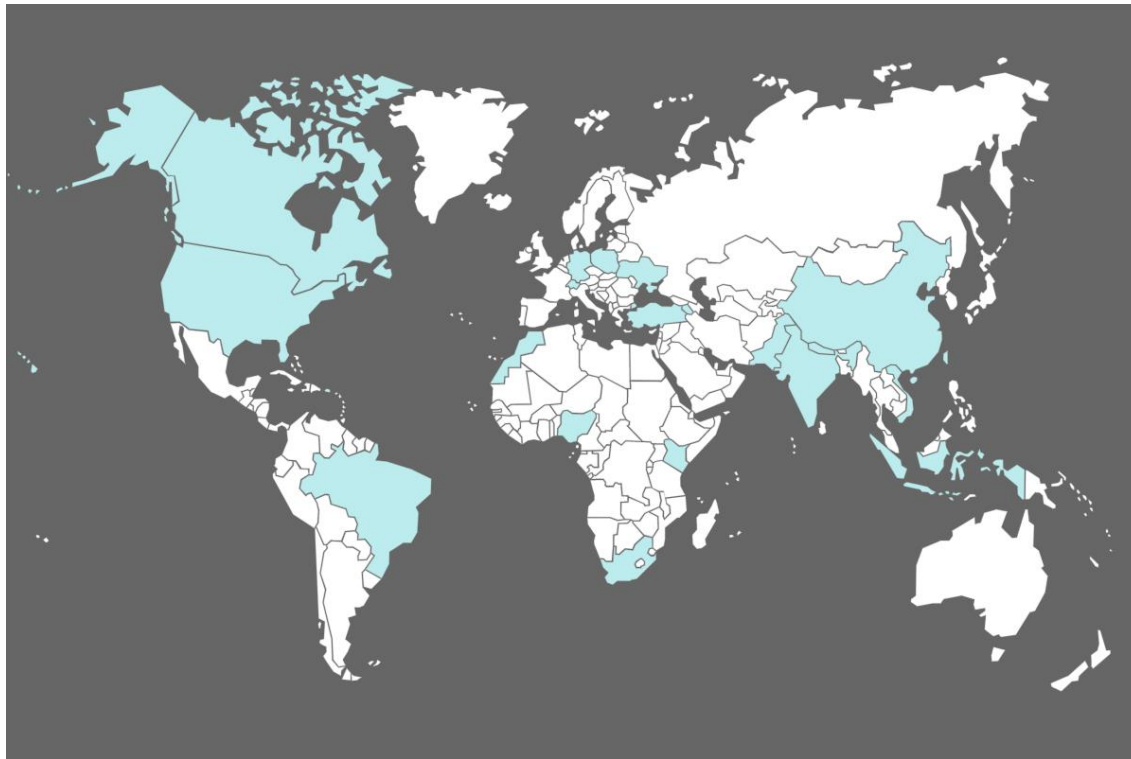
- ❖ コパイロット (コンテストマネージャ): Wendell
- ❖ レビュアー: Wendell/paolog

The screenshot shows the TopCoder challenge page for 'METI - Case Studies Responsive Application REST API'. The page is titled 'METI - Case Studies Responsive Application REST API' and includes a 'Code' tab with 'TCO19', 'Java', 'MySQL', 'REST', 'Spring', 'AWS', 'Microsoft Azure', and 'Linux' options. The 'PRIZES' section shows 1st place at \$1,200 and 2nd place at \$600. A 'Register' button and a 'Submit' button are visible. Below this, a timeline shows the challenge status: 'The challenge is finished.' and 'Hide Deadlines'. A table lists the schedule: Started (Oct 27, 22:03), Registration (Oct 29, 22:11), Submission (Nov 01, 22:04), Review (Nov 03, 17:23), and Winners (Nov 04, 12:05). The page also includes links for 'DETAILS', 'REGISTRANTS (59)', 'SUBMISSIONS (5)', 'WINNERS (2)', and 'CHALLENGE FORUM'. The 'Challenge Overview' section provides background information, objectives, technology stack (Java 9, Springboot, MySQL, Gradle), challenge scope, general requirements, and individual requirements. On the right, there are sections for 'ELIGIBLE EVENTS' (TCO19), 'REVIEW STYLE' (Final Review: Community Review Board), 'CHALLENGE LINKS' (Review Scorecard), and 'CHALLENGE TERMS' (Apprio NDA v2.0, Standard Terms for Topcoder Competitions v2.2). A 'SHARE' section with social media icons is also present.

## 3. コンテストを活用したアプリケーション開発

### 3-5. バックエンド開発コンテスト実施報告

#### (4) コンテスト参加者（コンテスト1回目）



#### 【コンテスト参加者】

・59名

#### 【出身国（20ヶ国）】

・ドイツ連邦共和国 ・モロッコ王国 ・インド共和国  
・中華人民共和国 ・パキスタン\_イスラム共和国  
・南アフリカ共和国 ・アメリカ合衆国 ・ケニア共和国  
・ナイジェリア連邦共和国 ・ベトナム社会主義共和国  
・ウクライナ ・トルコ共和国 ・インドネシア共和国  
・スイス連邦 ・アルメニア共和国 ・ポーランド共和国  
・ブラジル連邦共和国 ・中華民国  
・ネパール連邦民主共和国 ・カナダ

#### 【成果物提出者】

・5名

# 3. コンテストを活用したアプリケーション開発

## 3 – 5. バックエンド開発コンテスト実施報告

### (5) コンテスト実施概要①（コンテスト2回目）

#### バックエンドサービス（API）

- ❖ アプリケーションの動作に必要なデータ入出力機能を、REST APIで提供する。
- ❖ データベースや検索エンジンと連携する。また、ユーザー認証をベースにアクセスコントロールを行う。

#### バックエンドサービス開発

- ❖ コンテストスペック、データベーススキーマ、API仕様（Swaggerドキュメント・補足資料）を準備
- ❖ Java/SpringBootベースでの実装。プロダクションコード・ユニットテストコードを開発。

#### 開発スコープ

- ❖ 以下の13 APIを対象とした。

➤ 事例更新		[PUT]	/case_studies/{id}
➤ 事例削除		[DELETE]	/case_studies/{id}
➤ 画像作成		[POST]	/case_studies/{id}/images
➤ 画像更新		[PUT]	/case_studies/{id}/images/{image-id}
➤ 画像削除		[DELETE]	/case_studies/{id}/images/{image-id}
➤ お知らせ更新	[PUT]		/notifications/{id}
➤ お知らせ削除	[DELETE]		/notifications/{id}
➤ 組織取得		[GET]	/organizations?field
➤ 組織更新		[PUT]	/organizations/{id}
➤ 組織削除		[DELETE]	/organizations/{id}
➤ カタログ取得	[GET]		/catalogs?field
➤ カタログ更新	[PUT]		/catalogs/{id}
➤ カタログ削除	[DELETE]		/catalogs/{id}

# 3. コンテストを活用したアプリケーション開発

## 3-5. バックエンド開発コンテスト実施報告

### (6) コンテスト実施概要② (コンテスト2回目)

#### コンテストページ

- ❖ 「METI - Case Studies Responsive Application REST API #2」  
(<https://www.topcoder.com/challenges/30074181>)

#### コンテストスケジュール (実績)

- ❖ 実施期間: 11/11 ~ 11/20 (約8.5日間)
  - 開始 ~ 提出期限: 5日間
  - レビュー/アピール期間: 3.5日間

#### 登録者/成果物

- ❖ 登録: 61名
- ❖ 成果物: 6件
- ❖ 賞金獲得者
  - 1位 If.coder040 (スコア: 90.17)
  - 2位 freegod (スコア: 86.17)

#### コンテスト運営 (コミュニティ)

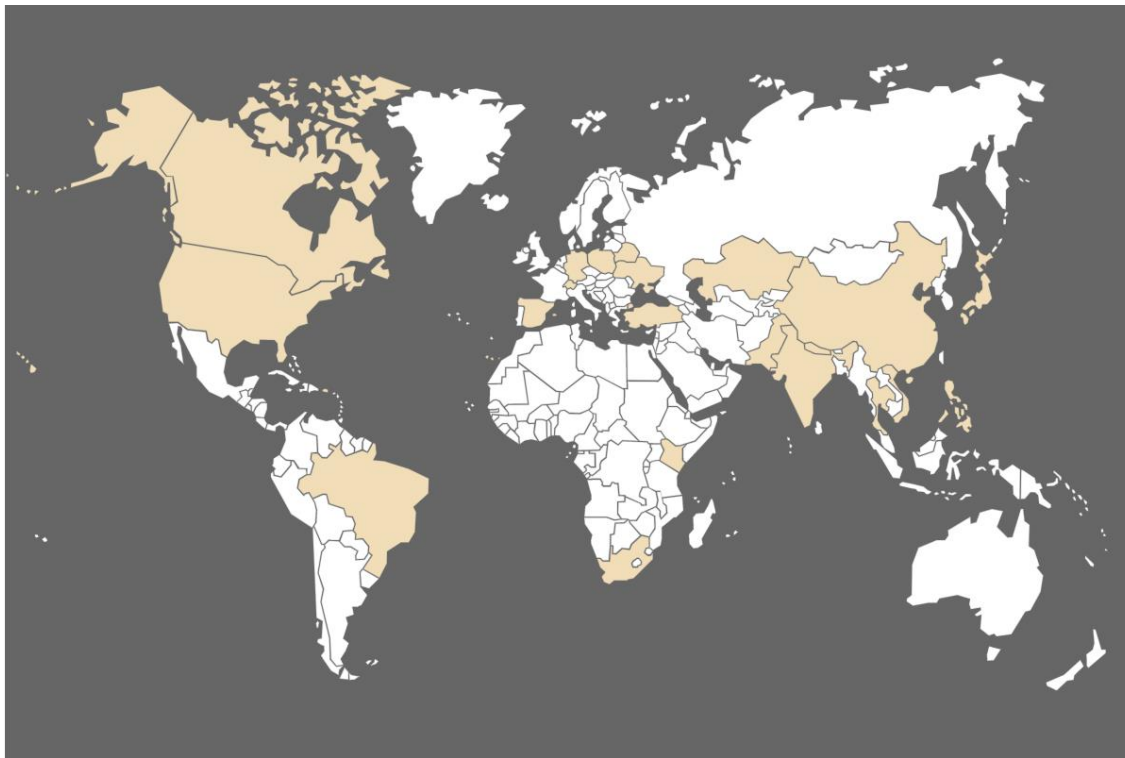
- ❖ コパイロット (コンテストマネージャ): Wendell
- ❖ レビュアー: Wendell/onsky

The screenshot shows the TopCoder challenge page for 'METI - Case Studies Responsive Application REST API #2'. The page includes a header with the challenge title and a navigation bar with links for 'Code', 'TC19', 'Java', 'Spring', 'MySQL', 'REST', 'AWS', 'Linux', and 'Microsoft Azure'. Below the header, there is a 'PRIZES' section showing '1st \$1,000' and '2nd \$500'. A 'Register' button and a 'Submit' button are visible. A table shows the challenge timeline: 'Started Nov 11, 23:03', 'Registration Nov 13, 23:05', 'Submission Nov 16, 23:10', 'Review Nov 19, 00:15', and 'Winners Nov 20, 10:35'. The 'Challenge Overview' section includes 'Project Background', 'Challenge Objectives', 'Technology Stack' (Java 9, Springboot, MySQL, Gradle), 'Challenge Scope', and 'General Requirements'. On the right side, there is a sidebar with 'ELIGIBLE EVENTS: TCO19', 'REVIEW STYLE: Final Review: Community Review Board', 'Approval: User Sign-Off', 'CHALLENGE LINKS: Review Scorecard', 'CHALLENGE TERMS: Apporio NDA v2.0, Standard Terms for Topcoder Competitions v2.2', and 'SHARE:' with social media icons.

## 3. コンテストを活用したアプリケーション開発

### 3-5. バックエンド開発コンテスト実施報告

#### (7) コンテスト参加者（コンテスト2回目）



#### 【コンテスト参加者】

・61名

#### 【出身国（21ヶ国）】

・中華人民共和国 ・インド共和国  
・パキスタン\_イスラム共和国 ・ポーランド共和国  
・ドイツ連邦共和国 ・アメリカ合衆国 ・ケニア共和国  
・トルコ共和国 ・スペイン王国 ・ウクライナ  
・フィリピン共和国 ・南アフリカ共和国  
・日本 ・ベトナム社会主義共和国  
・ネパール連邦民主共和国 ・ベラルーシ共和国  
・タイ王国 ・カザフスタン共和国 ・スイス連邦  
・ブラジル連邦共和国 ・カナダ

#### 【成果物提出者】

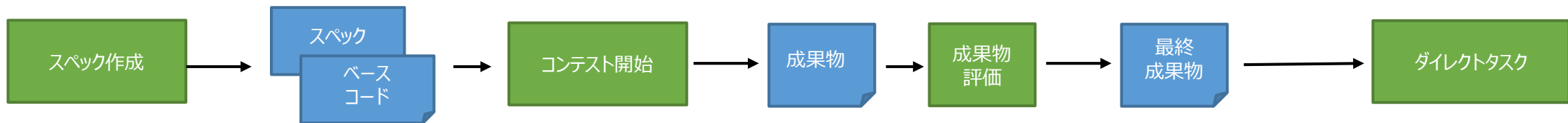
・6名

# 3. コンテストを活用したアプリケーション開発

## 3-6. インテグレーション開発コンテスト実施報告

### (1) インテグレーション開発コンテストの進行について

- アプリケーションの機能仕様および詳細設計からコンテストスペックを作成し、コンテストを実施した。
- コンテスト開始後は、参加者との質疑応答を通して、開発中の問題点を解消した。
- コンテストにアサインされたレビュアーが成果物を評価し、上位の成果物を最終成果物として獲得した。
- スcopeに入らなかった機能、または必要な修正について、ダイレクトタスクによる開発を実施した。



#### スペック作成

- TC3とTopcoderコミュニティのコンテストマネージャにてコンテストへの要求事項（スペック）を作成
- スペックに記載する主要項目
  - プロジェクト全体像
  - コンテスト要件
  - 使用技術スタック
  - サポートブラウザ
  - 開発スコープ
  - 成果物形式
  - など

#### コンテスト開始

- スペックが公開され、コンテストがスタート
- 参加者の質問に対して、コンテストマネージャまたはTC3が回答を行う。（必要に応じて顧客担当者に相談）

#### 成果物評価

- 開発コンテスト用の評価項目（スコアカード）に沿ってレビュアーが評価を行い順位を決定
- 主な成果物
  - ソースコード
  - テストコード
  - ビルド、実行方法に関するドキュメント

#### ダイレクトタスク

- TC3にて残タスクを取りまとめ、Topcoderの開発者に、開発を依頼

※ダイレクトタスクは、開発者を直接指名して開発を依頼する。小規模な開発を短期間に効率よく行うような場合に適している。

# 3. コンテストを活用したアプリケーション開発

## 3 - 6. インテグレーション開発コンテスト実施報告

### (2) コンテスト実施概要①

#### インテグレーション

- ❖ フロントエンドアプリケーションとバックエンドAPIの連携部

#### インテグレーション開発

- ❖ フロントエンドにおける主な開発
  - API呼び出しによるデータ入出力の実装、ユーザーインタフェースの調整
- ❖ バックエンドにおける主な開発
  - フロントエンドとの結合にて発生したAPIロジックの調整など
- ❖ コンテストスペック、補足資料、API仕様（Swaggerドキュメント・補足資料）を準備

#### 開発スコープ

- ❖ 以下の9機能を対象とした。
  - ヘッダー
  - ナビゲーションメニュー
  - ホーム画面
  - 事例一覧画面
  - 事例詳細画面
  - コレクション一覧画面
  - コレクション詳細画面
  - 通知
  - 各種ダイアログ



# 3. コンテストを活用したアプリケーション開発

## 3-6. インテグレーション開発コンテスト実施報告

### (3) コンテスト実施概要②

#### コンテストページ

- ❖ 「METI - Integration Challenge #1」  
(<https://www.topcoder.com/challenges/30075162>)

#### コンテストスケジュール (実績)

- ❖ 実施期間: 11/26 ~ 12/6 (約10日間)
  - 開始 ~ 提出期限: 5日間
  - レビュー/アピール期間: 5日間

#### 登録者/成果物

- ❖ 登録: 76名
- ❖ 成果物: 1件
- ❖ 賞金獲得者
  - 1位 anonymousjaggu (スコア: 84.33)

#### コンテスト運営 (コミュニティ)

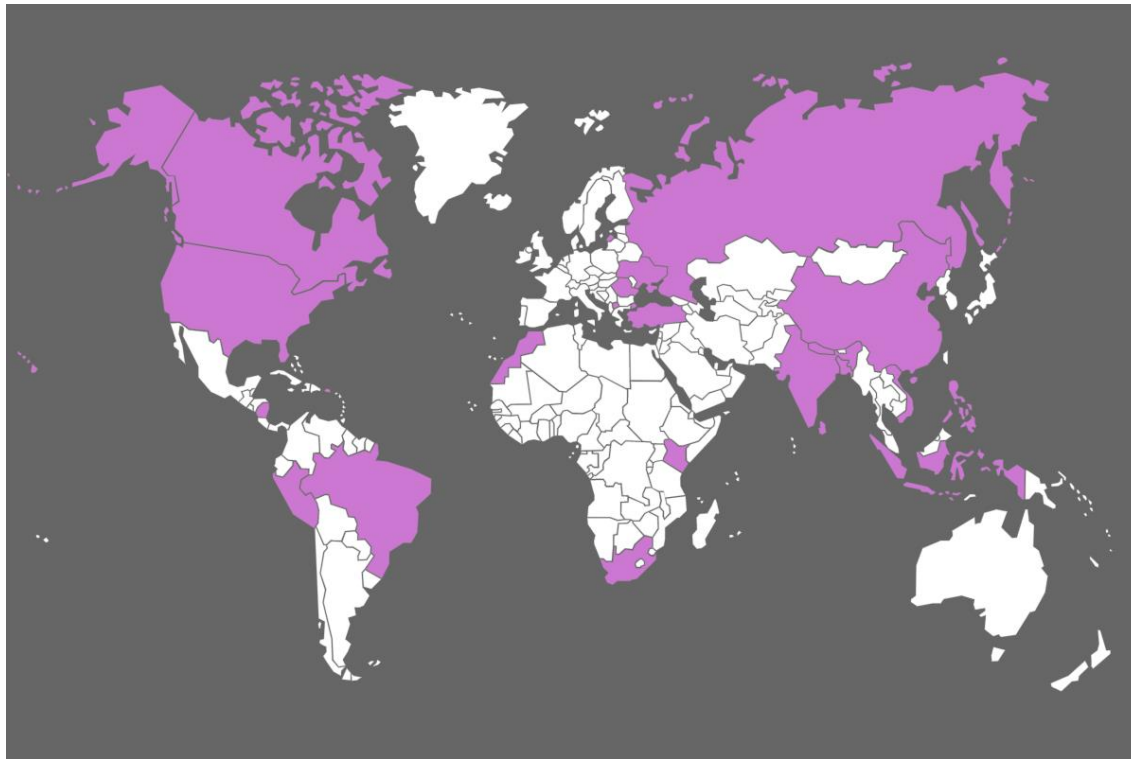
- ❖ コパイロット (コンテストマネージャ): Wendell
- ❖ レビュアー: Wendell

The screenshot shows the TopCoder challenge page for 'METI - Case Studies Responsive Application REST API #2'. The page includes a header with the challenge title and a navigation bar with links for 'Code', 'TC19', 'Java', 'Spring', 'MySQL', 'REST', 'AWS', 'Linux', and 'Microsoft Azure'. Below the header, there is a 'PRIZES' section showing '1st \$1,000' and '2nd \$500'. A 'Register' button and a 'Submit' button are also visible. A table shows the challenge timeline: 'Started Nov 11, 23:03', 'Registration Nov 13, 23:05', 'Submission Nov 16, 23:10', 'Review Nov 19, 00:15', and 'Winners Nov 20, 10:35'. The 'Challenge Overview' section includes 'Project Background', 'Challenge Objectives', 'Technology Stack' (Java 9, Springboot, MySQL, Gradle), 'Challenge Scope', 'General Requirements', and 'Challenge Links'. The 'Eligible Events' section shows 'TCO19' and 'Review Style: Final Review: Community Review Board'. The 'Challenge Terms' section mentions 'Apprio NDA v2.0' and 'Standard Terms for Topcoder Competitions v2.2'. The 'Share' section includes social media icons for Facebook, Twitter, LinkedIn, and YouTube.

## 3. コンテストを活用したアプリケーション開発

### 3-6. インテグレーション開発コンテスト実施報告

#### (4) コンテスト参加者



#### 【コンテスト参加者】

・76名

#### 【出身国（21ヶ国）】

・インド共和国 ・ブラジル連邦共和国  
・ネパール連邦民主共和国 ・南アフリカ共和国  
・中華人民共和国 ・ペルー共和国  
・スリランカ民主社会主義共和国 ・アメリカ合衆国  
・ウクライナ ・カナダ ・トルコ共和国 ・ロシア連邦  
・インドネシア共和国 ・ケニア共和国  
・ベトナム社会主義共和国  
・バングラデシュ人民共和国 ・モロッコ王国  
・フィリピン共和国 ・ニカラグア共和国 ・ルーマニア  
・マケドニア共和国

#### 【成果物】

・1件

## 3. コンテストを活用したアプリケーション開発

### 3-7. 開発コンテストにおけるレビュープロセス

#### (1) 開発コンテストにおけるレビューシステムについて

開発コンテストでは、コンテストにレビュアーがアサインされ、評価プロセスは全てコンテスト内で実施される。各提出物は、コンテストスペックの内容と客観的評価基準に基づいてスコアが付けられ、スコアの高い提出物を出した参加者が勝者となる。

- 開発コンテストごとにレビュアーが1名～2名アサインされる。
- レビュアーは、評価基準（スコアカード）に沿って提出物をレビューを行いそれぞれにスコアをつける。
- レビュープロセスは2段階で行われる。レビュアーから最初のスコアが出ると、参加者はレビュー結果について抗議（アピール）を行うことができる。
- アピール内容に基づいてレビュアーがスコアの修正を行い、最終的なスコアが確定される。（レビュアーが複数の場合は平均点となる）
- 入賞には、一定以上のスコアを満たす必要がある。これにより、低品質な提出物が入賞してしまうことが防止される。

※Topcoderコミュニティにおいて、当該分野で実績を積んだ開発者が、レビュアーとしてアサインされる。

## 3. コンテストを活用したアプリケーション開発

### 3-7. 開発コンテストにおけるレビュープロセス

#### (2) スコアカードの評価項目

Section 1.1	Contest Specification Requirements
Question 1.1.1	Have all major specification requirements been met?
Question 1.1.2	Have all minor specification requirements been met?
Section 1.2	Code Best Practices & Technical Requirements
Question 1.2.1	Does the submission follow standard coding best practices?
Question 1.2.2	Does the submission include an appropriate amount of comments?
Question 1.2.3	Has obsolete or unnecessary code been cleaned up?
Question 1.2.4	Has reasonable consideration been given to security?
Question 1.2.5	Has reasonable consideration been given to performance?
Section 1.3	Deployment Guide
Question 1.3.1	Merely following the instructions in the Deployment Guide, you are able to successfully deploy the application?
Question 1.3.2	Has the submission maintained the Deployment Docs if they previously existed?
Question 1.3.3	Is there some means of verification of submission provided?

## 3. コンテストを活用したアプリケーション開発

### 3-8. 開発コンテストで実施したプラクティス

#### (1) ベースコードの提供

開発コンテストでは、コンテストという性質上、開発者の間で具体的なコードが共有されることは少なく、実装の詳細での種々の決定は各開発者に委ねることになる。各提出物に大きな差異が出てしまったり、或いは品質にむらが発生するのを防止するために、アーキテクチャを最小限にカバーする動作可能なコードを用意し、開発者に提供した。

これにより、アーキテクチャの共有、成果物に求めるものの伝達をスムーズに行うことができた。

また、ベースコードをビルドして動作させるためのビルド用スクリプトを提供した。これにより、開発者がアプリケーションコードの開発に集中できるようにした。また、ビルドプロセスが統一されることで、レビュー時間の短縮、インテグレーションの効率化において効果が得られた。

バックエンド開発のベースコードには、以下のようなものを含めた。

- サンプルAPI実装（共通のロジックを含む最低限のAPI実装）
- ユニットテスト
- ビルドスクリプト（ビルドプロセスを全て実行できるスクリプト）
  - コードのビルド
  - 自動テストの実行
  - 静的コード解析の実行

## 3. コンテストを活用したアプリケーション開発

### 3 - 8. 開発コンテストで実施したプラクティス

#### (2) APIファーストの設計と開発

APIファーストの方針に則って、まずREST APIの設計を行い、その後、APIをベースに各コンポーネントの開発を進めていった。APIを先に規定することで、以下のことが実現された。

- バックエンドコンテストの開発者はAPIの実装だけに専念することができ、フロントエンドの開発者はAPIの実装がなくてもUIの開発だけに専念することができた。コンテスト参加者が自分の得意分野だけに注力することで効率よく高品質な成果物を得ることができた。
- バックエンドの開発とフロントエンドの開発を並行して実施することが可能になり、工期を短縮できた。

APIの設計書には、Swagger (OpenAPI 3.0) を利用した。Swaggerの利用によって以下のような利点があった。

- APIの物理的な詳細 (URL、パラメタ、項目名、項目のデータ型など) を効率よく記述することができた。
- Topcoderのコミュニティにて多くの開発者に親しまれており、形式について特別な説明無しに利用してもらうことができた。
- ツールの利用により、開発者が劇的に開発効率を高めることができた (テストデータの作成やスタブの生成などの自動化)

## 3. コンテストを活用したアプリケーション開発

### 3 – 8. 開発コンテストで実施したプラクティス

#### (3) 品質保持のための施策

開発コンテストのレビューフェーズは、詳細なテストを網羅的に実施するには期間が短い。そのような中でも成果物の品質を保つために、開発工程にて以下のような施策を行うことで、品質を素早く検証できるようにした。

##### 自動テストの開発

- ユニットテストフレームワーク (JUnit等) を使用した自動テストの開発をコンテストの成果物に含めた。
- 自動テストが無い、または失敗するテストを含む成果物はスコアが落とされるため、スクリーニングに寄与する。
- 自動テストにより、後続の追加開発および修正において、コードのデグレードの発生の防止に役立った。特に、オリジナルのコードを開発したメンバーとは別のメンバーが開発するような場合に効果があった。

##### 静的コード解析ツールの導入

- [SpotBugs](#) を導入し、自動的に潜在的なバグの検査を行った。
- ビルドプロセスに組み込むことで、違反を含むソースコードが検出されると、ビルド自体が失敗するようにした。
- ソースコードに違反を含む成果物はスコアが落とされるため、スクリーニングに寄与する。



## 3. コンテストを活用したアプリケーション開発

### 3-8. 開発コンテストで実施したプラクティス

#### (4) 自動ビルド・自動デプロイの構築

ソースコードの更新に伴い自動化されたビルドプロセスを継続的に実行される仕組みを用意した。また、主要な変更については実行環境に自動でデプロイされるようにした。これにより、以下のような効果を得ることができた。

- ビルドプロセスには、（単なるモジュールのビルドだけでなく）自動テストや静的コード解析チェックが実行される、欠陥の発生をすぐに検出し、欠陥がデプロイに混入することを防いだ。
- デプロイ手順を自動化することで、デプロイのコストを大幅に削減し、成果物や修正を頻繁にデプロイして、実環境でテスト・レビューすることができた。

Topcoderのようなコミュニティによる開発を行う場合、ソースコードのオーナーを固定することは難しく、あるソースコードを複数の開発者が次々と修正していくことも頻繁に起こり得る。これらの仕組みにより、そのような開発においても、コードのデグレードやバグの混入を防ぐのに、非常に効果的であった。

※本開発においては、これらを実現するために [CircleCI](#) を利用した

## 4. コンテスト以外で実施した開発

### 4-1. TC3で実施した開発タスク

本開発において、以下のような観点によりコンテストで実施しない方が良いと判断した開発タスクは、TC3内部で実施した。

- 実施に必要なリソース（開発環境やツール・サービス等）を、多数の参加者に公平に提供できない作業
- 日本語（英語以外の言語）をベースとした作業
- コンテスト形式が非効率になってしまう作業（テスト等）

以下に、代表的なタスクを挙げる。

#### （1）ユーザーインターフェースのローカライズ（日本語化）開発

コンテストの成果物は英語がベースになるため、日本語化を行う必要があった。

Topcoderのようなグローバルなコミュニティで開発を行うにあたっては、ほとんどの場合、日本語（英語以外の言語）をベースにしての開発は難しい。日本人または日本語を十分に扱えるメンバーを探すことも可能ではあったが、リスク要因であるため、今回は内部で日本語化を行った。

※日本語化作業のオーバーヘッドを削減する取り組みとして、フロントエンド開発コンテストにおいては、ローカライゼーションライブラリを使用を義務付けた。これにより、日本語化に伴うソースコード変更を最小限にすることができ、また、言語をスイッチしての開発を可能にした。

## 4. コンテスト以外で実施した開発

### 4 – 1. TC3で実施した開発タスク

#### (2) インフラ環境構築

##### AWS環境構築

- 本開発におけるアプリケーション動作環境をAWS（Amazon Web Services）上に構築した。

##### 認証基盤

- 独自認証基盤、および各認証システム（法人共通認証基盤、各種SNS）とのインテグレーション基盤としてAuth0を使用した。  
Auth0を本開発用にカスタマイズを行った。

数多くの開発者が自由に参加できるコンテストでは、有償のリソースを使用する開発には、通常、コストの管理が必要になりオーバーヘッドになる。また、成果物に差が出にくい、コミュニティメンバーの興味を引くことができるか不透明である、等の理由により、本開発では内部で環境構築を行った。

# 4. コンテスト以外で実施した開発

## 4 – 1. TC3で実施した開発タスク

### (3) 結合テスト・品質管理

- インテグレーション開発終了後に、テスト環境にデプロイされたアプリケーションのテストを実施した。
- テスト手法
  - 予め定義したテストケースに沿ってマニュアルで実施した。
  - 数サイクル実施し、段階的に不具合を修正した。
- デバイスのバリエーション
  - PCブラウザ（Google Chrome / Internet Explorer 11）
  - 携帯、タブレット（それぞれ、エミュレータ／実機）
- テストケースサマリ

機能カテゴリ	テストケース数	機能カテゴリ	テストケース数
共通	19	コレクション詳細	46
ログイン/ユーザー登録	42	コレクション一覧	99
ユーザー情報編集	24	検索	181
ホーム	42	検索結果	182
事例詳細	99	通知	7
事例一覧	94	合計	825

## 5. 「中小企業支援プラットフォーム(仮称)」との連携

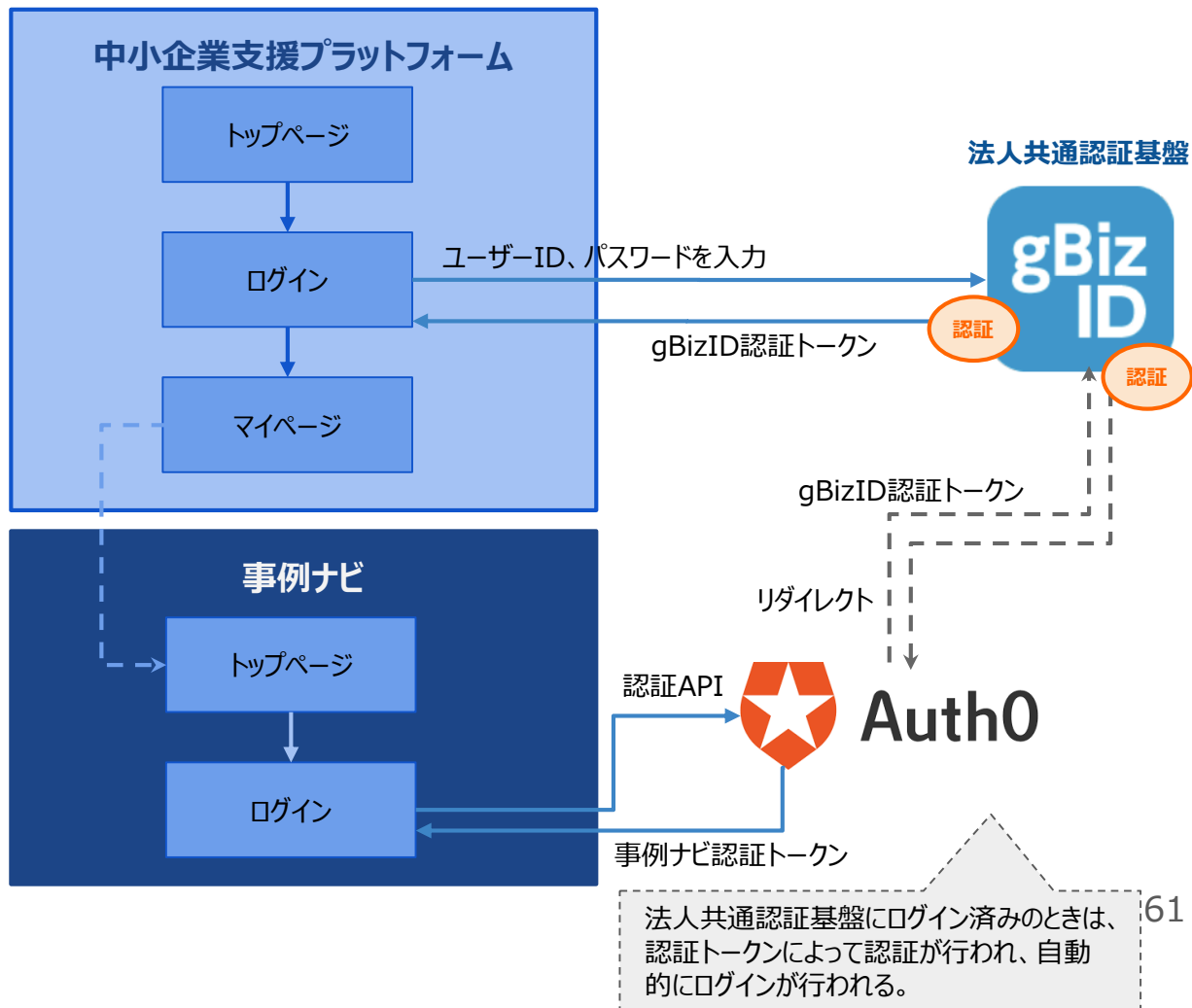
### 5-1. 認証基盤との連携

#### 法人共通認証基盤との認証連携

事例検索アプリケーション（「事例ナビ」）は、OpenIDConnectに対応した法人共通認証基盤によるSSO（シングルサインオン）をサポートする。

本開発においては、Auth0 (<https://auth0.com/>) を拡張して、法人共通認証基盤をIDプロバイダーとして透過的に利用することで、これを実現した。

「事例ナビ」のログイン画面には、法人共通認証基盤SSO用のボタンが用意されており、クリックするとOAuthベースの認証フローが開始される。既に法人共通認証基盤にログインされていれば、自動で事例検索アプリケーションにもログインされる。ログインされていなければ、法人共通認証基盤のログイン画面が表示される。法人共通認証基盤にログインすることで事例検索アプリケーションへのログインが行われる。



## 5. 「中小企業支援プラットフォーム(仮称)」との連携

### 5-2. 中小企業支援プラットフォーム（仮称）からの遷移

#### ■メニュー構成に応じた事例一覧画面の表示



事例ナビ

## 6. 総括

### 6-1. 本調査における実績値について

本調査の実績における主な統計値を以下に示す。

#### (1) システムの開発言語、RDBMSとステップ数

#		言語	ファイル数	ステップ数※
1	フロントエンド	TypeScript/JavaScript	59	3,846
2		Sass(CSS)	33	7,683
3		HTML	31	3,069
4	バックエンド	Java	82	5,718
5		Java (ユニットテスト)	16	3,752
6	DBMS	SQL (MySQL)	1	590
合計				24,658

※ステップ数の計測には、cloc (<https://github.com/AlDanial/cloc>) を使用した



## 6. 総括

### 6-1. 本調査における実績値について

#### (2) 開発フェーズ別の割合

本調査実績から設計、製作、テストの各開発フェーズ別における工期の割合は以下の通りとなった。  
(3フェーズの合計を100とし、単体テストは製作に含めることとする)

設計	製作	テスト
22%	62%	16%

#### (3) 不具合件数と発生原因

本調査における、結合テスト時における不具合発生件数及び発生原因は以下の通りとなった。  
尚、テスト実施のためデータ整備、環境設定に起因するものは、本件数には含まないものとする。

#		件数	割合
1	設計漏れ	9件	29%
2	製作漏れ	22件	71%
合計		31件	

## 6. 総括

### 6-2. 本調査の評価

本調査の実績を、「ソフトウェア開発データ白書2018-2019」を参考に総括する。参照する統計データは、各値の「中央値」を参考とするものとする。（「平均値」は、極端に大きな値の影響を受けやすいため）

#### （1）品質

- P60「結合テスト・品質管理（テストケースサマリ）」及びP64「不具合件数と発生原因」により、結合テスト時のテストケース密度は、33.46件/KSLOCとなり、統計データ（35.70件/KSLOC）とほぼ同等。また、検出バグ密度は、P64「不具合件数と発生原因」により、1.257件/KSLOCとなり、統計データ（1.600件/KSLOC）を下回る結果となったがテストケースは各機能を網羅されているため、品質の問題はないと分析する。
- 本調査では、フロントエンド開発、バックエンド開発及び不具合改修等、製作のほとんどをTopcoderを利用している。各開発は、コンテスト形式で参加者を募って実施し、提出された中から上位の成果物を採用する方法とした。各成果物の採点は、仕様要件が満たされているか否かだけでなく、セキュリティ、パフォーマンス、コーディング標準などを客観的かつ総合的に判定されるため結果、品質の良い成果物を得ることができた。
- UI/UXデザインタスクについてもコンテスト形式でデザイン案を募り、中間評価、修正を経て最終デザイン案により順位を決定した。複数のデザイン案による比較、検討を行うことで、よりユーザエクスペリエンスの高いデザインを選択することができた。

工程	P25	中央値	P75
結合テスト（テストケース）	17.85	35.70	64.27
結合テスト検出バグ数	0.943	1.600	2.626

[件/SLOC]

「ソフトウェア開発データ白書2018-2019」より

## 6. 総括

### 6-2. 本調査の評価

#### (2) コスト（工数）

- Topcoderを利用しているプロジェクトの特性上、ウォーターフォール型との単純な比較はできない。そのため、製作工期に絞って生産性を評価することとする。
- コンテスト形式での実施のため開発コンテストの期間及び単体テスト、不具合改修期間を製作時の工数として換算する。
- P60「システムの開発言語、RDBMSとステップ数」及び上記工数より、製作工期の生産性は、約16SLOC/人時となり統計データ（14.5SLOC/人時）を上回る結果となった。
- 要件定義、設計工期（主に設計書作成及びレビュー）における追加要因を加味すると、全体工数としてはウォーターフォール型に比べ少ない工数で対応できたと推察する。

[SLOC/人時]

工程	P25	中央値	P75
製作	9.9	14.5	24.8

「ソフトウェア開発データ白書2018-2019」より

## 6. 総括

### 6-2. 本調査の評価

#### (3) 納期

- P64「開発フェーズ別の割合」により、設計工期、製作工期、テスト工期の内訳比率は、20：60：20となり、ウォーターフォール型の開発に比べ、製作工期の比率が他の工期に比べて非常に多い結果となった。これは、設計工期における設計作業、設計書の作成、レビューにかかる工期を最低限とし、その分、製作工期において中間成果物による動作確認、フィードバックを繰り返すことで機能確認を実施することとした。
- 各機能の製作が終了した時点で結合テスト以降はウォーターフォール型と同様に実施した。（品質向上のため）
- その結果、仕様の変更、追加や設計工期に起因する不具合の発生があったものの、全体のスケジュール（納期）に影響のない範囲での対応を完了することができた。
- 更に、「法人共通認証基盤」との認証連携にも対応できたため、今回の開発手法には一定の効果があったと考察する。  
（今回のスコープは、「法人共通認証基盤」との連携方法の検討まで）

# 6. 総括

## 6-2. 本調査の評価

本調査を通じて効果的だった点と改善すべき点を以下に挙げる。

### (4) 効果的だった点

- コンテストのメリット
  - 複数の成果物から良いものを選択できる点。
  - 次点の成果物の良い点を取り込むなど、他のデザイナー・開発者の成果を取り入れることができる点。
- 設計書ベースではなく成果物ベースでの確認・レビュー
  - 前者の方式に比べ情報量が豊富なため、意思決定を素早く行うことができた。
  - 製造の手戻りのコストは製造工程の効率化により低減したため、スケジュール遅延を引き起こすことはなかった。
- 製作工程の効率化
  - 開発コンテスト開催時にベースコードを提供したことで、アーキテクチャやビルド手順が統一された。
  - APIファーストの設計・Swaggerを活用したことで、開発の効率化がはかれた。
  - ユニットテストの義務化・静的コード解析チェックの導入により品質が向上した。
  - 継続的なビルドとデプロイの導入により、成果物の評価を効率的に行うことができた。
- Backlogの活用
  - 高いセキュリティを保ちながら大容量のファイルを簡単に共有できる点。
  - タスクや課題をチケット化することにより状況の共有、進捗が管理できる点。

## 6. 総括

### 6-2. 本調査の評価

#### (5) 改善すべき点

- 提出物の少なかったコンテストがあった
  - 2つの開発コンテストで成果物が1つだけだった。
  - スcopeが大き過ぎたため脱落者が出てしまった可能性が考えられる。良い成果物を得るには、高めのゴール設定を行うことが重要であるが、高すぎた場合は調整を行う必要がある。
  - Topcoderのイベントとスケジュールが重なり、参加メンバーが積極的でなかった可能性が考えられる。なるべくイベント開催期を避けるか、リスクを前提としたスケジュールを組んでおく必要がある。
- テスト・品質管理における改善点
  - 自動テストの実施は非常に効果があったが、プラクティスを完全にカバーできたわけではない。
  - フロントエンドの開発では、ソースコードの安定度合いが見切れなかったため、自動テストは開発を行わなかった。現状はソースコードが十分安定しているため、自動テストの開発は十分可能である。
  - ユニットテストのカバレッジ率については、レビュアーの目視確認で行った。自動化してビルドプロセスに組み込むことで、更に効率的に品質向上が目論める部分であると言える。
  - 結合テストはマニュアルで実施したためリソースを要した。そのため回帰テストを頻繁に実施できなかった。自動化が可能なケースについては自動化を行うことでより品質向上を強化できる。

## 6. 総括

### 6-3. 留意事項について

以下に、今後事例集アプリを更に開発する上で留意すべき点をまとめておく。

#### (1) Auth0の利用について

本開発では、ユーザー認証とその周辺機能を効率的に実現するために Auth0 を利用した。

- ユーザー登録・メールアドレス検証（メール送信含む）
- ログイン画面・認証
- パスワード変更画面・パスワード変更変更フロー（メール送信含む）
- 法人共通認証基盤アカウントでのログイン
- SNS（Facebook、Twitter）アカウントでのログイン

Auth0の利用範囲を独自実装に置き換える場合は、これらの項目の再実装を検討する必要がある。

また、Auth0の利用プランにより、登録・管理できるユーザー数が異なるため、前提とするユーザー数およびAuth0使用によるコストの見積りを再度実施する必要がある。



## 6. 総括

### 6-3. 留意事項について

#### (2) 非機能要件について

本開発においては、目的がプロトタイプ開発であり、また限られたユーザーへの公開という前提であるため、通常のシステム開発で必要となる要件を限定的にカバーした。

更に本格的な運用を計画するにあたっては、本開発でカバーしていない要件（主に非機能要件）の検討を要する。

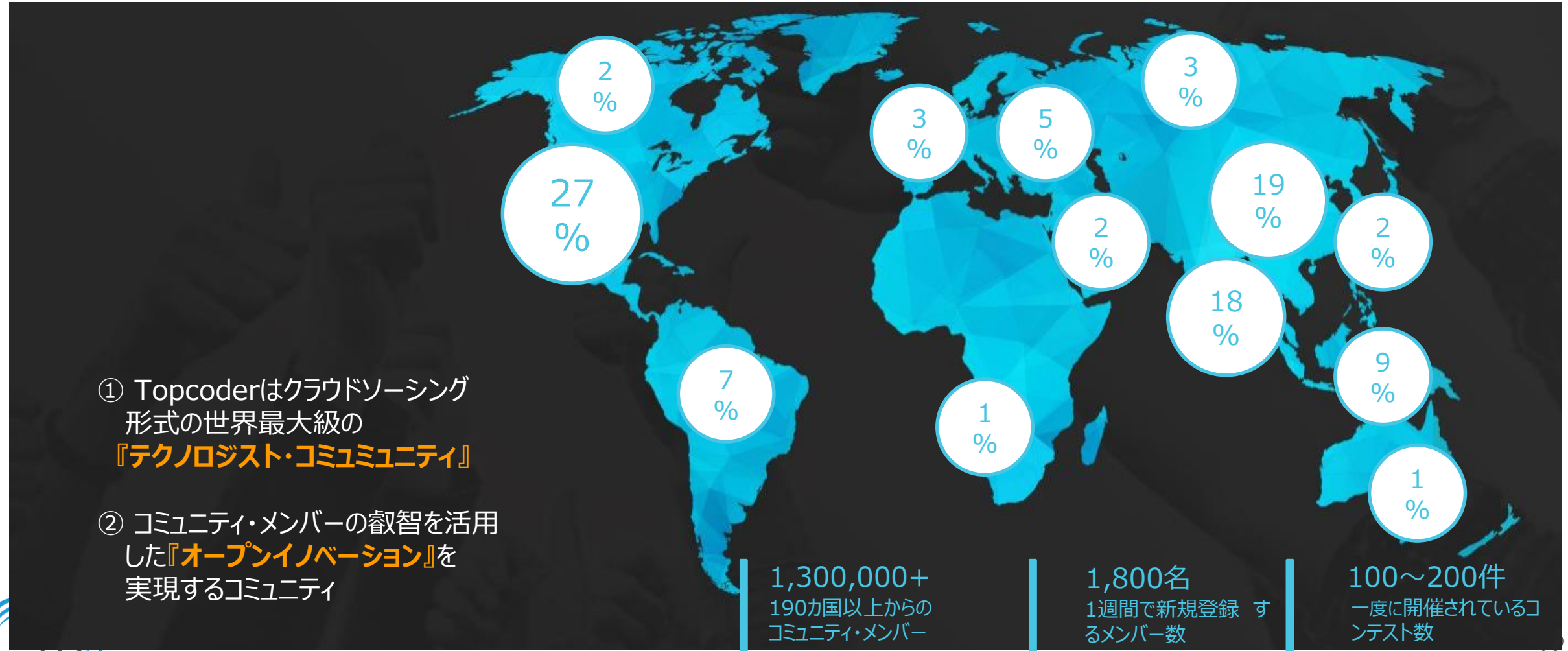
##### 主要検討項目

- SLA
- 運用に関する要件（および設計と実装）
- サーバーのキャパシティ計画（再検討）
- 上記項目を前提とした負荷試験
- ドキュメンテーション（ユーザーマニュアル、運用手順書、など）
- 保守体制
- など

※本調査においては、事例データ約350件で負荷試験を実施した。今後追加されるデータ量を想定した負荷試験を行う必要がある。

## 参考. Topcoderとは

- ・世界で130万人以上のコミュニティ・メンバーを擁する世界最大級の「テクノロジスト・コミュニティ」。
- ・グローバル企業、国家機関のデジタル・トランスフォーメーションを実現する先進技術分野の R&Dパートナーとして機能。



## 参考. Topcoderとは

- ・競技プログラミング・コンテストをその出自とする独自のメソドロジーで質の高い成果物を開発するコミュニティと仕組みを持つ。
- ・数日から 2週間程度のクイックなコンテストを繰り返し実施することによって、アジャイルでクイックなプロジェクト進行を可能にしている。



### 沿革

- ・ 2001年 : 競技プログラミングのコミュニティとして設立
- ・ 2003年 : 初めての商用サービス開発プロジェクトを実施
- ・ 2016年 : TC3株式会社が日本市場でのサービス提供を開始

### Topcoder設立の背景

- ・ 設立者のJack Hughesは元々システム開発会社を運営。優秀な人材の確保に苦労。人月ビジネスに疑問を感じていた。
- ・ 稼働時間に対する支払いではなく 成果物に対する支払いを基本思想とする。

### 特長

- ・ 「コンテスト」と呼ぶ競争形式を採用。
- ・ コンテスト形式により、優れた成果物のみを採用。
- ・ 登録者数はグローバルで 130万人以上。日本人は約2万人でデータサイエンス分野に多い。

### コンテスト

- ・ 数日から 2週間程度に課題を切り分けた実行単位。
- ・ メンバーが一人で単独で行える大きさに設定。
- ・ 個々のメンバーの得意領域へのフォーカスと敗北時の損失最小化を目的とする。

## 参考. Topcoderとは

- ・アプリ開発にて活用する Topcoderサービスの概要を下記に記す。
- ・各サービス共に、絞り込んだ技術領域をテーマとして開発が実施され、グローバルから当該領域のスペシャリストを集める事ができる仕組みであり、最終的に提示された複数の成果物を更に比較・評価する事によって品質が高く、仕様に最も合致した成果物を選択する事ができる。

コンテスト種別	UIデザインサービス	フロントエンド開発サービス	バックエンド開発サービス
プロジェクト全体期間	1ヶ月	1ヶ月	
コンテスト期間	1～2週間	1～2週間	
コンテスト登録者・参加者目安	20～50名	20～50名	
最終提示成果物数目安	5～10件	2～3件	
成果物評価方法	主観評価 (お客様が主に評価)	客観評価 (Topcoderが主に評価)	
成果物の形態	Photoshop, Illustrator, Sketch等のデータ	HTML, CSS, JavaScript等の 動作可能な実行ファイル	

# 参考. システム構成図

