

# オフショア開発で挑む新規プロダクト開発 ～顧客価値を軸にしたワンチーム開発！～

製品企画開発本部（TSD）  
プロダクトエンジニアリング部  
経費モバイルユニット

須崎 慎一郎





01	自己紹介	-----	03
02	TeamSpiritとは？新規プロダクトとは？	-----	05
03	オフショア開発3つの課題	-----	09
04	オフショア開発の課題改善施策	-----	13
05	改善の取組み結果／今後へ向けて	-----	19

# 01

## 自己紹介

# 自己紹介

須崎 慎一郎

すさき しんいちろう

武器はメモを  
取ること、行動  
することです！



項目	内容
会社	株式会社チームスピリット
所属	製品企画開発本部(TSD) プロダクトエンジニアリング部 経費モバイルユニット
役職	QAエンジニア／エキスパート
主な仕事	<ul style="list-style-type: none"><li>経費精算の保守開発</li><li>新規プロダクト開発</li></ul>
その他活動	社内部活動「ワークアウト」部部長として 定期的にワークアウトを実施 Note：チムスピ★部活動 Note <a href="https://note.teamspirit.com/m/mbb841a64a962">https://note.teamspirit.com/m/mbb841a64a962</a>
個人での活動	X、ブログ、YouTube「 <b>SusakiWorks</b> 」で テスト・テスト自動化・業務効率化等の情報を発信中！



# 02

## TeamSpiritとは？ 新規プロダクトとは？



## TeamSpiritとは？

TeamSpiritは、勤怠管理・工数管理・経費精算の主な機能があり、働く人が毎日使うバックオフィス機能を提供し、そこから生まれる「ワークログ」の活用を通じて組織改善の一手をつくるクラウドサービスです。



小規模～中堅企業向け

 **TeamSpirit 勤怠** **TeamSpirit 工数** **TeamSpirit 経費**

従業員1,000名以上の大企業向け

 **TeamSpirit Enterprise**

# 新規プロダクト「勤怠設定ナビ」

## 勤怠設定ナビとは？

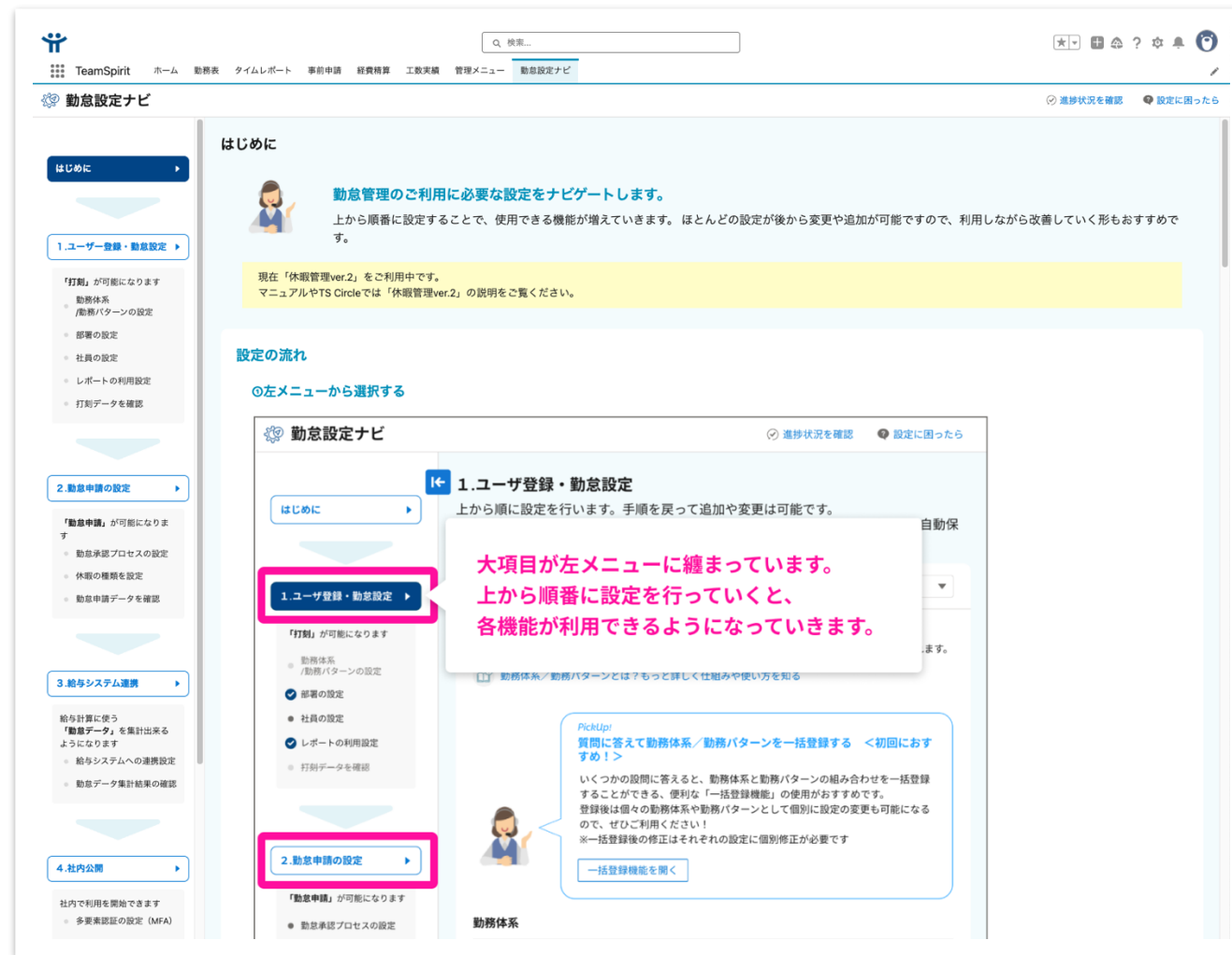
2024年夏に開発した、  
TeamSpiritの導入から運用まで  
**ステップ形式**で短時間で設定できる  
サービス。

## TeamSpiritの大きな課題

TeamSpiritの導入で、勤務体系や勤怠  
申請等の設定は、**初期コスト**がかかり、  
数年前から大きな課題の一つであった。

## 解決策として…

初期コストの課題を解消するため  
「勤怠設定ナビ」が考案された。



## 「勤怠設定ナビ」開発の概要

- 開発体制
  - 日本側：開発者、QA
  - オフショア側：開発者、ブリッジSE、QA
- 開発期間：約5ヶ月
- 開発プロセス：
  - 要件、設計：日本
  - 実装、ユニットテスト：オフショア
  - 受け入れテスト：日本



（日本メンバー）

開発から2ヶ月後・・・

プロダクト開発を進めていたが、2ヶ月あたりから「**3つの課題**」に直面した。



# 03

## オフショア開発での3つの課題

# オフショア開発で直面した課題

## 1.日本とオフショアで、コミュニケーションコストが爆増！

### 要因1：仕様に関する質疑応答の状況不明

仕様に関する質疑応答が各ツールに散乱し、回答状況を調べる時間がかかった。

(例)

- チャットツールメッセージ
- タスクチケットのコメント
- Webミーティングのメモ

そもそも、オフショアとの「**コミュニケーションルール**」を定めていなかった。

### 要因2：複雑な設計内容の伝達ミス

この開発体制で作成するプロダクトではこれまでより複雑な設計だった。

(例)

- 設定ON/OFF表示処理の違い
- 既存オブジェクトとのデータマッピング数が多く複雑

当初の開発体制では、複雑な部分を**うっかり伝え忘れてしまう**ことがあった。



- 日本側
- 日本メンバー(開発,QA)



質問を探すのが大変

オフショアブリッジSE  
(日本滞在)



情報不足のまま翻訳

- オフショア側
- 開発,QA

# オフショア開発で直面した課題

## 2.発注したものを作るだけの顧客価値不在の開発に…



### 要因：ゴール設定の誤り

要件を元に、  
「**プロダクトを作成すること**」  
のみをゴールにしていた。

(例)  
勤務体系の作成をステップで表示はできる。

ユーザー視点や使い勝手、  
**UIの見やすさ**といった本来の  
目的が置き去りになった。

タスクチケットの完了条件は満たしているが、ただ何か足りない…

デザイン資料、設計資料通りに  
作ってるけど、ユーザー視点からは  
使いやすいだろうか…

開発タスクに追われていて、考慮  
する余裕がない…



# オフショア開発で直面した課題

## 3.設計や開発のタスク・リスク見積り不足で、**開発スケジュールが遅延**...



### 要因1：既存機能への考慮不足

日本側の設計で、  
**既存機能に影響する範囲**の  
設計考慮が足りていなかった。

(例)

設定をOFFに変更したが、関連する既存  
項目がOFFにすべきである。

本来機能テスト以前に検出する不  
具合が受入テストまで検出できな  
かった。

### 要因2：オフショア担当箇所の見積もり乖離

オフショア担当の実装や事前調査  
は、オフショア見積もった。

(例)

エラーバリエーションについて既存機能  
から調査する

既存機能への考慮不足の影響で、  
**見積と実際の工数が大きく乖離**し  
た。

# 04

## オフショア開発の課題改善施策

# オフショア開発の課題改善の取組み

## 各課題の改善取組み



コミュニケーション  
コスト爆増



質疑応答や情報の一元化

顧客価値不在の開発



価値を見い出す  
ミーティングを導入

開発スケジュール遅延



進捗をこまめに共有

日本とオフショアのワンチーム化



# オフショア開発の課題改善の取組み

## コミュニケーションコスト爆増への改善

### Q&A イメージ



### 改善策：質疑応答や情報の一元化

「**Q&A**」**シート**の導入及び、  
Q&Aにフォーカスした  
ミーティングを実施

チャットツール、タスクチ  
ケット、ミーティングでの口  
頭確認など、仕様に関する質  
問のやり取りを「**Q&A**」  
シートに**一元管理**

質疑応答時間と  
開発時間を分け  
て確保！

■新機能画面「A」作成 QandA

No.	質問日	カテゴリ	概要	質問内容	回答	備考
1	2024/4/1	デザイン	最新のデザインページの場所	最新のデザインページの場所はどこですか？	以下URLとなります。	
2	2024/4/5	機能実装	保存処理の実装	スプリント1では保存実装まで行いますか？	はい、保存処理実装をお願いします。	
3	2024/4/12	仕様確認	設計資料とデザイン資料の相違	設定項目1に対して、以下のように相違しています。 どちらが正しいですか？ ○設計資料 ・テスト12345  ○デザイン資料 ・テストABCDE	設計資料が正しいです。 デザイン資料の内容が古い情報でした。 申し分けありません。	
4	2024/4/20	機能実装	削除処理の実装	スプリント1では削除処理は実装しますか？	スプリント1では実装しません。 スプリント3で実装予定です。	
			ダイアログ文言更新	ダイアログの文言更新の範囲を教えてください。	ダイアログの文言更新は、以下範囲です。 ・機能A ・機能B ・機能C	

# オフショア開発の課題改善の取組み



## 顧客価値不在の開発への改善

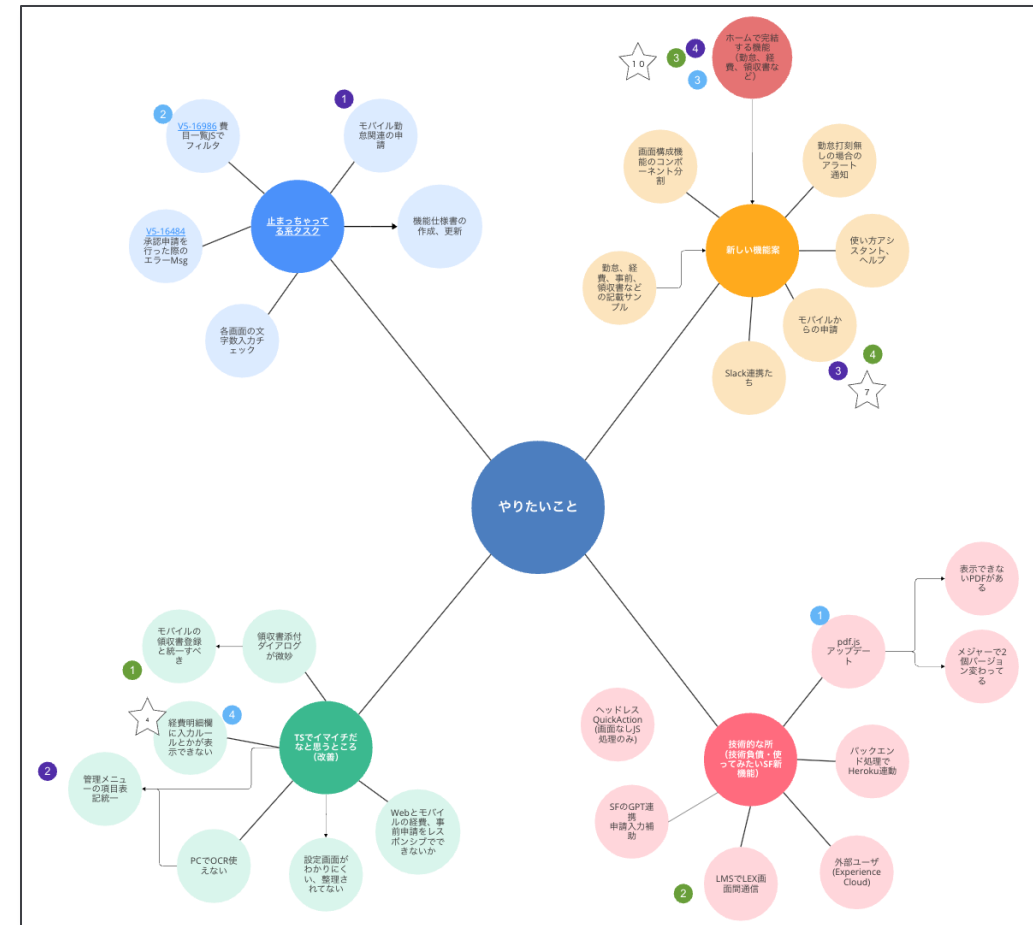
### 改善策：価値を見出すミーティングを導入

日本のみで実施予定だった、開発現場のボトムアップからより良い**プロダクト作成に関する意見交換会**にオフショアメンバーも巻き込む

(例)

・「勤怠設定ナビ」の開発や実装にあたって、エンドユーザが操作しやすくするUIの提案

## イメージ



# オフショア開発の課題改善の取組み

## 開発スケジュール遅延に対する改善



基本を大事に！

### 進捗をこまめに共有

日本・オフショアメンバーで  
「夕会」を実施し、  
**日々の進捗を明確に共有**

(例)

- ・ 完了日がいつか？
- ・ 躓いている部分はどこか？

### 確認事項 例

・ 工数は明確に反映したか？	✓
・ 完了日に変わりはないか？	✓
・ タスクチケットの更新は最新か？	✓
・ 困っていることはないか？	✓
・ 相談したいことはないか？	✓



# オフショア開発の課題改善の取組み



## 日本とオフショアのワンチーム化

### モブ設計ミーティング導入

「**モブ設計ミーティング**」を導入し、開発者とQAを含めてフロント・バックエンド設計を実施、仕様や要件の抜け漏れを防止

(例)

「勤怠設定ナビ」の勤務体系作成シーンで、各勤務体系の初期値やエラーメッセージを設計

### 日本とオフショアの垣根を超えたタスク分担

開発プロセスを日本とオフショアで分断せず、**手の空いたメンバーが協力してタスクを消化**する。

(例)

オフショアで実施予定だったシステムテストを日本側で実施する。  
日本側が実施予定だった機能テストをオフショアメンバーが実施する。

# 05

## 改善の取組み結果／今後へ向けて

# 改善の取り組み結果

1

## 質疑応答や情報の一元化

質問回答に専念する時間、作業をする時間をキッチリ分けることができた。  
スイッチングコストを抑えて、コミュニケーションコストは許容範囲内に収まった。

2

## 価値を見い出すミーティングを導入

オフショアから出た意見をもとに、UIの改善をすることができた。

3

## 進捗をこまめに共有

遅延を食い止め、プロダクトリリースすることができた。

4

## 日本とオフショアのワンチーム化

お互いに協力し合う中で、上記の①～③を推進しやすくなった。



## 今後へ向けて

改善に取り組んだことにより、オフショアか否かに関わらず発生する「設計の情報不足」という課題が浮き彫りになってきた。  
この課題に対しても、改善を検討する。

今回の改善は、オフショアか否かに関わらず黎明期のスクラムチームにも転用できる可能性がある。

**標準化**を検討する。

