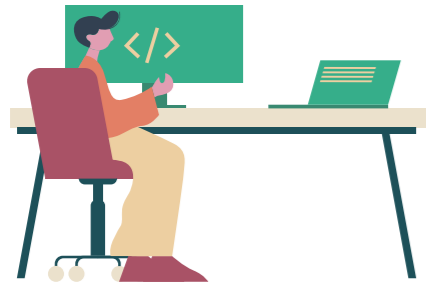


# スクラム開発とWebサービス開発技術の習得

## 目的

価値創造や研究よりも  
**既存の開発手法や思想の学習**に  
重きを置くこととした。



## 目標

実践的なプロダクト開発を通じて

- スクラム開発の理論や手法
- Webサービス開発に必要な技術
- パフォーマンス設計技術

を学習・習得する。

## 作成したプロダクト



週報\*を書く時間を  
削減したい学生を  
対象としたアプリケーション

\* PBL 活動中の学生に課される毎週の報告書



## スクラムの学習と実践

スクラム開発の経験者が少ない中、学習した理論を忠実に実施することでスクラムの根底にある経験主義\*  
という考えを実践的な形でプロダクト開発に取り入れることができた。

\* 事実に基づいた、経験に基づいた、そして証拠に基づいた方法で作業すること。過去の過ちや経験を学び、常に改善していくこと

## 技術習得のための取り組み

活動開始当時、Webサービス開発経験者が少なかったため開発技術自体の習得が必要だった。  
活動の中で各自が必要な技術をフロント・バックエンド領域双方の習得に向けて  
段階的に学習するようにした。

## 透明性

今なにをしているか  
どのような状況なのか  
分かること

### 大きく改善できた取り組み

**背景** 期限間近になって作業が完了しないことが発覚した。  
**課題** 作業者がいつまでに何を終わらせようとしているかわからない、他のチームメンバーが状況を把握できない。

### デイリースクラム\*でタスクの完了予定日を報告



進捗を把握しやすくなり、チーム全体で協力して  
タスクを終わらす体制をしくことができた。

\* 一日一回、開発チームが集まって進捗や問題の共有を行う場

## 検査

潜在的に望ましくない変化や問題がないかを確認し検知すること

**背景** コードレビューの修正に時間がかり、予定期間内にタスクが完了できなかった。  
**課題** コード実装後に初めてレビューを行うと手戻りが大きくなる。

### 設計レビューの導入

コード実装開始前に設計レビューを行う事で早期に  
問題に気づく事ができ、大きな手戻りをなくすことができた。

**背景** 前期で開発目標を達成できなかった。  
**課題** 現状の開発スピードで問題がないか明確ではない。

### プロダクトゴール\*達成までの計画をチームに共有

計画を明確にすることでチーム全体でどのようにタスクを  
進めていくかの認識が共有できた。

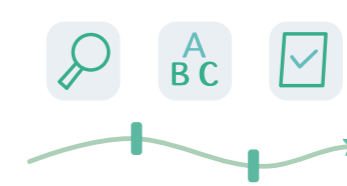
\* プロダクトの将来の状態を表す、スクラムチームの長期的な目標のこと

## スクラム3つの柱

## 適応

やり方を軌道修正したり  
よりよいやり方を  
探索・実践すること

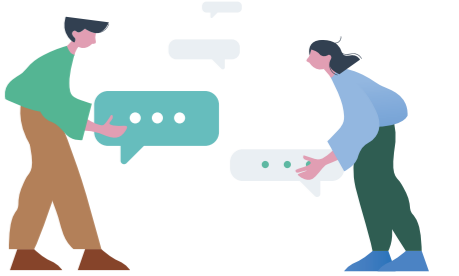
## 取り組みによって得られた効果



リリースを頻繁に行いつつ、プロダクトとしての  
品質を担保するため開発当初より CI/CD や  
自動テストなどを導入した。

- 開発速度の向上
- デグレートの防止

Web サービス開発の学習効率を高めるため  
成果物のピアレビューを行うようにした。



- コード品質の向上
- 他人のコードを読むことでの気づき

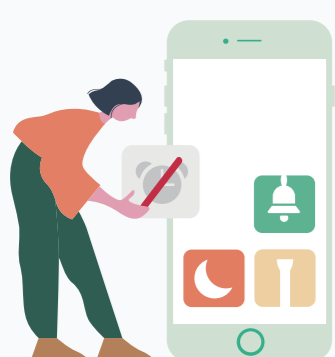
長時間の作業・調査が常態化していたため  
活動中盤よりペアプログラミングなど  
同期的な作業を積極的に行い  
協力して作業を終えるようにした。

- 知識の共有
- 作業効率の向上
- レビュー時間の短縮

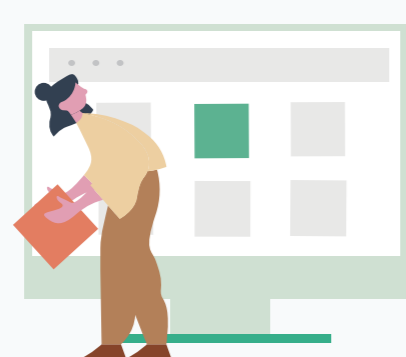
## パフォーマンス改善への取り組み

PBL 活動中盤で、新たな技術的チャレンジとしてパフォーマンス改善を目標に加えた。経験者が少ないため、計測から改善まで手探りの中で進めることになった。

パフォーマンスに関する調査を行い、改善目標を立てた。  
また計測結果やネット記事等を参考に  
以下の方法でパフォーマンス改善に取り組んだ。



不要なコードの削除



軽量ライブラリへの置き換え



監視ツールを用いて日々計測を行った

## 活動を通して学んだこと

- 闇雲なパフォーマンスチューニングは逆効果であり  
まずはボトルネックを明確にするための解析に時間をかける必要がある。
- パフォーマンスに最適化されたコードが開発者にとって開発しやすいとは限らない。
- コードだけでなく、アーキテクチャレベルの総合的な理解も求められる。