

生産性向上に向けた私の働き方改革への取り組み

富士通エフ・アイ・ピー（株）

■ 執筆者Profile ■



陸 秋瑠

2017 年 富士通エフ・アイ・ピー（株）入社
システム技術推進部配属

2018 年 現在 第四サービスマネジメント部所属

■ 論文要旨 ■

筆者は OJT 期間中に、社内向けの教育管理システム構築プロジェクトに参画した。その中で 3 画面の詳細設計からシステムテストまでを担当した。しかし 1 画面目を完了した時点で、生産性が低いという課題に直面した。

このままの進め方ではスケジュールの遅延が発生し、納期遵守が厳しいため、生産性向上に向けた筆者の働き方の改革が必要と判断した。

そこで、生産性が低下する原因分析を行い、「一日の作業タスクが明確になっていない」、「作業時間を正確に把握できていない」、「一人で悩む時間が多い」という 3 つの課題が洗い出された。

それら課題の対策として、「タスクの優先度付け」、「ツールを用いた作業時間の管理」、「チャットツールを活用したチームでのナレッジ共有」を筆者の働き方改革として取り組んだ。その結果、当初目標の生産性を達成することができた。本稿では、直面した課題の原因分析から働き方の改革を成功させるプロセスを述べる。

■ 論文目次 ■

1. はじめに	《 3》
1. 1 当社の概要	
1. 2 アプリケーション技術者育成プログラム	
1. 3 参画プロジェクト	
2. 直面した課題	《 3》
3. 働き方改革への取り組み	《 4》
3. 1 振り返り	
3. 2 対策	
3. 2. 1 対策① 作業の優先度付け	
3. 2. 2 対策② 作業時間の管理	
3. 2. 3 対策③ ナレッジ共有	
4. まとめ	《 7》
4. 1 成果	
4. 2 今後の展望	

■ 図表一覧 ■

図1 優先順位表	《 5》
図2 一日のタスク・タイマー計測結果	《 6》
図3 各工程改革前後のレビュー指摘率比較グラフ	《 7》

1. はじめに

1. 1 当社の概要

当社は全国に展開されているデータセンターをベースに「IT アウトソーシング&クラウド」、「システムインテグレーション」、「プロフェッショナル」、「SaaS」の4つのサービスを提供している。システムの企画から設計、開発、運用、保守まで、ライフサイクル全般を支援し、お客様に安心・安全で、高品質かつ高コストパフォーマンスのサービスを提供している。

当社ではお客様のパートナーとして質の高いサービスを提供するために、常に高い技術力を継続追求している。その取り組みの一環として、若手SEのシステム開発技術の習得を目的としたアプリケーション技術者育成プログラム（以下育成プログラムという）が2015年から設立された。

1. 2 アプリケーション技術者育成プログラム

育成プログラムは一年間で構成される。前半はシステム開発言語のアルゴリズム、ASP.NET、データベースの基本的な仕組みなどを演習形式で学習する。後半は実際の開発プロジェクトに参画し、開発を実践するような育成活動である。また技術スキル以外は、プレゼンテーション能力を高める課題発表などに取り組んでいる。育成プログラムは二チーム制であり（JAVAチーム、.NETチーム）、配属されたのは.NETチームである。

1. 3 参画プロジェクト

社内向けシステムである教育管理システムの構築プロジェクトに参画した。

教育管理システムは社内の教育計画の申請を管理するWEBシステムである。従来部署ごとに管理していた教育計画を統合し、社内で一元管理及び、運用するためのシステムである。

開発期間は約7か月、システム規模は全19画面、10バッチである。開発メンバーは育成プログラムの新人5名、トレーナー2名、支援メンバー2名の合計9名から構成される。本プロジェクトで初めてシステム開発に携わり、システム詳細設計からシステムテストまで担当した。

本稿では、システム開発期間中に発生した課題への対応を通じて生産性向上のための自身の働き方改革について論ずる。

2. 直面した課題

教育管理システムを開発する際、各メンバーにいくつかの画面機能が割り当てられた。最初に担当したのはログイン画面である。富士通では、システム機能の規模を表す尺度としてFunctionScale値（以降FS値という）が用いられている。ログイン画面のFS値は225であり、全機能の中でFS値は最も低く、簡単な画面である。

実装を始める前、画面がシンプルなので、簡単に実装できそうな印象であった。しかし、1画面目が終わった際に、チーム内の目標生産性400FS/人月に対して、筆者の生産性は139.7FS/人月、チームの中で最も低いことが分かった。

次に担当することになっている教育資格抽出画面の機能構築は、ログイン画面と比べて機能も多くFS値が890FSと難易度がかなり高い。

1画面目を作成した際の生産性だとスケジュールに遅延が発生し、納期遵守が厳しい状況だった。これまでの作業の進め方には問題があるのではないかと考え、働き方改革が必要だと判断した。

3. 働き方改革への取り組み

3.1 振り返り

まず生産性が低かった原因分析を行った。

洗い出してみると、作業タスクが明確になっていない、作業進捗を正確に把握できていない、一人で悩む時間が多いという3つの原因が挙げられる。

(1) 一日の作業タスクが明確になっていない

自分の作業タスクを整理できていなかった。

例を挙げると、プログラムのソースレビューをした際に、詳しくない機能について調べた。他の使い方が気になり、レビューに関係のないところまで深く調べていた。しかし途中で「どこに向かって走ってるんだっけ？」と自己学習するのか、レビューするのかわからなくなることが多かった。

何をいつまでにやるという計画ができておらず、行き当たりばったりで仕事をしていた。

(2) 作業進捗を正確に把握できていない

スケジュールに進捗を記録していたが、正確な値を記録できていなかった。

進捗管理は当初、個人ごとに管理し、それをもとにチームの朝会で進捗を共有していた。1画面の機能構築が完了した時点で、チームのお互いの進捗を把握するために一つの管理ファイルにまとめることになった。

その際、チームメンバーと比べ、進捗の漏れや更新忘れが多いことに気づいた。思い返すと、一日の終わりにまとめてスケジュールを更新したり、翌日に更新をしていた。そのため記憶があいまいで更新内容が適当になってしまっていた。作業時間が正確に把握できず、スケジュールに遅れが発生しても気づくことができていなかった。

(3) 一人で悩む時間が多い

分からないことは質問せず、一人で解決しようとしていた。

開発期間中にチームメンバーが忙しいため、質問すると迷惑するのではないかと考え、質問することができなかった。その悩みをチームの振り返りの際に、共有すると他のメンバーも同じような悩みを抱えており、お互いに遠慮しあって質問しにくいという状況であることがわかった。

以上が効率を下げる原因と考えられる。いままでの作業の進め方の課題を認識することができた。

3.2 対策

振り返りで抽出した3つの原因に対して、それぞれ対策の検討、試行を行った。

3. 2. 1 対策① 作業の優先度付け

対策①は、仕事を効率よく進めるために毎日の作業に優先度をつけることである。タスクを実行に移す時に、緊急度と重要度による4つの分類から優先度を決める。優先順位表を図1に示す。

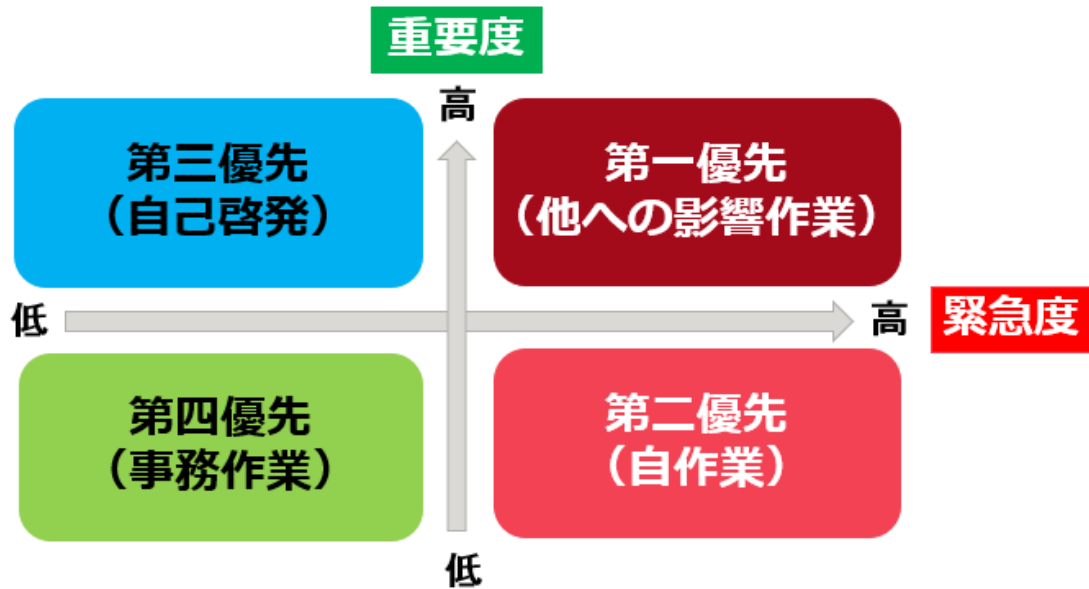


図1 優先順位表

第一優先は他のメンバーの進捗に関係する作業である。チームメンバーの設計書レビューなど優先度が高いタスクが当てはまる。第二優先は第一優先に対して、自分だけで完結できる作業である。自分が担当している画面機能構築のプログラミングなどが当てはまる。第三優先は自己啓発、第四優先は事務作業である。

優先順位表をもとに、自分のタスクの優先度を決めて作業の計画を立てた。

ある日のタスクは以下のように優先順位を設定した。

- ① メンバーAさんが書いたプログラムのソースレビュー
- ② 教育資格抽出画面のプログラミング
- ③ プログラミングで分からないことをまとめて学習
- ④ 作業表の記入

改革前は一日のタスクは明確にしてない状態だった。改革後はこのように、朝に一日の仕事を明確に分類し、納期も考慮して優先度を設定する。そうすることによって、作業順序を迷う時間が減少した。

その結果、行きあたりばったりで作業することがなくなり、計画的に作業を進めることができた。

3. 2. 2 対策② 作業時間の管理

対策②は、一日のトータル作業時間を把握するためにタイマーを使って作業時間を計る

ようにしたことである。活用していたのが Google Chrome のタスク・タイマーというアプリケーションである。タスク名を記入し、作業時間を計測できるという機能である。また目標時間設定すれば、進捗状況がリアルタイムで表示される。

図2のようにタスクごとに作業時間を計った。

タスク	経過時間	目標時間	進行状況
EP050レビュー後の修正	1:19:44	1:30:00	<div><div></div></div>
Loginエビデンス修正	1:46:38	2:00:00	<div><div></div></div>
EP050のSQL文作成	1:56:09	4:00:00	<div><div></div></div>
日常業務	0:30:00	0:30:00	<div><div></div></div>
合計時間	5:32:31	8:00:00	<div><div></div></div>

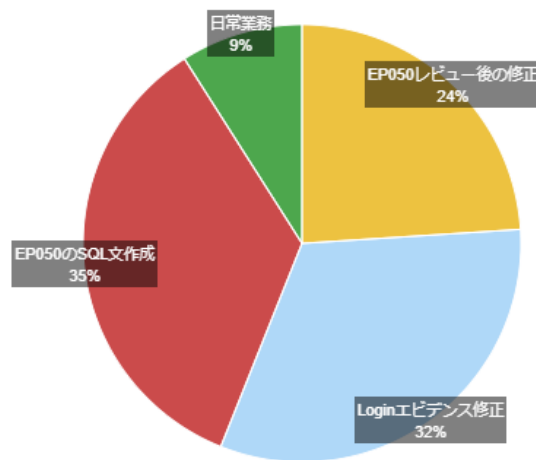


図2 一日のタスク・タイマー計測結果

改革前はスケジュールの記入漏れ及び、作業記録が正確でないことが多く発生していた。それに対して改革後はタイマーを使うことによって一日をどのように使ったのかを確認することができるようになり、スケジュールに正確な実績の入力ができるようになった。

これにより、作業進捗を把握しやすくなり、作業調整ができるようになった。

その結果、毎日スケジュールの更新への意識が強くなり、スケジュールの記入漏れの発生頻度が削減できた。

3. 2. 3 対策③ ナレッジ共有

対策③は、自分で10分考えたり、調べたりしても解決できない場合、お互いに質問することをチーム内で決めた。

なぜならば、10分調べても分からない場合は、自分の力で解決できないことが多い。そして、調べ続けて分からないと集中力が低下するため、その場合は質問したほうが効率的であると考えられる。

質問する場合は Lync を用いた。Lync とは社内で利用するチャットツールである。チームでグループを作成し、質問がある場合はそこに送るようにした。Lync を活用した理由として、相手の都合のよい時に確認できるため、割り込みを削減できるというメリットがあ

るからである。

また質問だけではなく、実装中に起きたエラーとその解決方法を一つの共有ファイルにまとめて、チームメンバーが参照できるようにした。

ナレッジ共有することによって、悩む時間を大幅に減少できた。その結果、効率的に仕事を進めることができた。更にチームメンバーにエラーの解決方法を伝えることによって、自分の知識の整理をすることができ、更に理解度を深めることができた。

4. まとめ

4. 1 成果

3つの対策に取り組んだ結果、生産性及び、品質が向上でき、働き方改革が実現できた。

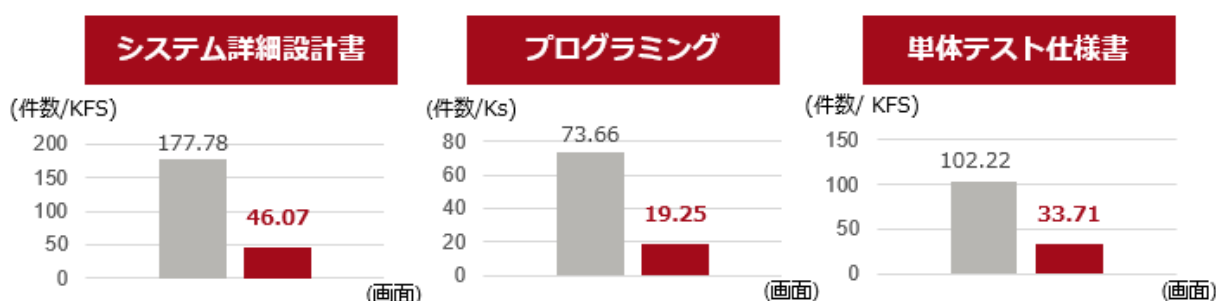


図3 各工程改革前後のレビュー指摘率比較グラフ

図3ではログイン画面と2画面目の機能構築のシステム詳細設計書、プログラミング実装、単体テスト仕様書のレビュー指摘率を比較したものである。2画面目はログイン画面よりはそれぞれのレビュー指摘率を約65%以上削減することができた。そして改革前と比較すると各工程の生産性は最初の139.7 FS/人月から511.8 FS/人月に上がり、2.7倍向上することができた。またチーム内の目標の生産性400 FS/人月も達成できた。

4. 2 今後の展望

現在は既存のシステムのアプリケーションの保守をする業務に参画している。今後も3つの対策を活用して、生産性を高めていきたい。

今回の取り組みでチームの目標生産性を達成することができた。しかし、これは新人向けの設定値であり、標準よりは低い値となっている。今後は標準の生産性を達成できるように取り組みたい。それを達成するために、技術力の向上はもちろんのこと、必要なのは作業効率の改善意識をもつことだと考える。日々、PDCAサイクルをまわし、業務効率化及び、品質向上への自分なりの改革を継続していきたい。