

# チーム開発実習レジュメ

2021/09/01

津田 雄吾

## 「分業」の大切さ

例えば1つのシステムを構築していく際に、時間に制限が無く、使える資源にも制限が無いのであれば、たった一人で開発していくのも難しくはありません。しかし現実には、多くの開発現場で期限が定められ、かつ使える資源にも限りがあります。

このような状況の中で、開発の効率化を図る一つの方法として、「分業」という仕組みがあります。分業とは、各人それぞれに役割を設け、複数人で業務を分担するということです。この、分業体制を取る人の集まりを「チーム」という単位で組織し、更に大きな分業体制の枠組みに加え、「プロジェクト」という単位で開発を進めていきます。

「チームによる開発が業務を効率化していく」と言えば、何となく良いように聞こえるかもしれませんが。しかしその本質的な所は少々複雑で、簡単に実現できるものではありません。

ここで知っておいて貰いたいのが、必ずしも人数が増えるだけで効率が増すという訳ではない、という事です。つまり、複数人で作業をするということはその分「連携力」が必要となり、連携が上手く取れないとかえって非効率な事態に陥ってしまうことになります。チーム開発と言っても、個々人のスキルや能力を活かせる環境が整備されていなければ、その分業体制は幻と言わざるを得ないでしょう。

問1 自分がチームに提供できる「価値」とは何か

問2 分業における「連携力」とは何か

## ソース管理

チームとしての連携力を高める手段の1つとして、「作成物の共有」があります。具体的には、バージョン管理システムを用いて、個人が作成したファイルをチーム全体で共有し、「誰が・いつ・どういった」変更点を加えたかをバージョンとして管理していくという方法です。

そして、チームでバージョン管理を使っていく中で、必ずと言っていいほど直面する事態の1つとしてあるのが、「変更点の競合」(コンフリクト)です。

例えば、あるテキストファイルをバージョン管理システムを使って編集していたとして、Aさんが編集した箇所と、Bさんが編集した箇所が被ってしまったとしたら、ファイルの最終変更記録はどのようになるのでしょうか。こういった場面を解決する為、「Git」などのバージョン管理システムでは、コンフリクトが発生した箇所を特定し、変更内容の振る舞いを編集者に決めてもらう方式を採用しています。

チームでバージョン管理を使っていく際、コンフリクト発生時の対応をよく相談し、あらかじめマニュアル化しておく事が望ましいです。

問3 「TeamRepository」という名前でリモートリポジトリを作成し、「A」「B」「C」という名前で新規作成したローカルフォルダにそれぞれクローンする。その後、「A」のローカルリポジトリ内で「memo」という名前でテキストファイルを作成し、コミット&プッシュ後、「B」「C」のローカルリポジトリでプルする。それぞれのローカルリポジトリはリモートリポジトリとどのように繋がっているか

問4 「A」のローカルリポジトリ内でテキストファイルを編集後、コミット&プッシュし、「B」のローカルリポジトリ内でも同様に編集を加えコミット&プッシュする。その時表示されるエラーはどのような内容か

問5 「B」のローカルリポジトリ内の変更内容を、「A」のローカルリポジトリの変更内容を使ってコンフリクトを適切にマージしなさい。その後、「C」のローカルリポジトリ内でも変更を加え、コミット&プッシュ後、変更内容を適切にマージしなさい

## プロジェクト進捗管理

チームの連携力を高めるもう一つの手段として、「プロジェクトの進捗管理」が挙げられます。進捗管理と一口に言っても、そこには様々な方法論があり、各開発工程の進捗状況が管理されています。

開発工程は主に、「要件定義」→「設計」→「実装」→「テスト」→「運用」という順序で進んでいきますが、要件定義から運用までを一本化して進める方式を「ウォーターフォール方式」といいます。これはちょうど、滝が上から下へと流れてくる様なイメージで、前工程に後戻りせず、一気に開発を推し進めていくという特徴があります。

それに対して、要件定義からテストまでの工程を1つの「サイクル」(繰り返し)として、機能ごとにそのサイクルで開発を進めていく方式を「アジャイル方式」といいます。これは、基本的に後戻りが難しいウォーターフォール方式と違って、機能別の開発となるため、実際の操作感等を確認しつつ、変更や修正が必要な箇所をあぶり出して開発を進めていけるという特徴があります。

また、進捗管理をするためには「チケット」という存在が欠かせません。チケットとは簡単に言うとチームの間で共有する「やる事リスト」(ToDoリスト)といえます。開発の進捗に応じて、チーム全体の現在の状況を正確に把握して、問題点を共有し解決していかなければなりません。チケット管理ツールなどを駆使し、進捗管理を推し進めていけば、想定外の事態への対処や他メンバーの業務内容の把握などがやりやすく感じると思います。

比較的短い期間での開発では、本格的な進捗管理ツールを使うほど規模も小さくなく、「動く」ものをできる限り高いクオリティで仕上げていくのが目標となるでしょう。その場合、私たちが意識しなければならないのは「変化への対応」です。包括的なドキュメントを盤石に仕上げるよりも、チームメンバー個々人との対話を重視し、常に変化に柔軟でいなければなりません。

問6 自分がとるべき「開発工程プロセス」とは何か