

Clean ArchitectureとAtomic Designを導入することによる開発体験の変化

22011376

柳萬 真伸

目的

今学期は設計手法を意識して開発に取り組んだ。具体的に使用したClean ArchitectureとAtomic Designについて得られたメリットとそこから得た学びを述べる。

Clean Architectureを使用したことによる開発体験の変化

Clean Architectureを端的に表すと不安定なモノに依存しないための設計手法である。ここでの依存とはソフトウェア設計で使われる専門用語で、あるモノがなければ存在できない関係のことである。依存されている数が多いモノほど、変更することによる影響は大規模になる。例えば、パソコンのマウスを買い替えても今まで通りパソコンを使用することができる。これは、パソコンがマウスに依存しない作りとなっているからである。もし、パソコンがマウスに依存していれば、マウスを変える度にパソコンを変える必要が出てくる。従って、変更する可能性が高いモノには依存すべきではない。この設計手法を使用することにより、例えばログイン機能の実装では、認証に使用するデータの種類と認証の方法を別々に決めることによって、認証データの受け取り方を柔軟にすることができた。

Atomic Designを使用することによる開発体験の変化

Atomic Designとは端的に表すとフロントエンドにおける各コンポーネントの責任を単一にするための設計手法である。ここでの責任を単一にするとは、ソフトウェア設計で使われる専門用語で仕事を一つしか行わないことである。一つのデザインを単一の責任となるように分解することにより、あるものを変更することによる影響を最小限にすることができる。例えば、スマホのクレジット機能を使用していると、スマホのアップデートをしてしまうとクレジット機能が使えなくなる心配が生まれる。しかし、スマホとクレジットカードを分けて持っておくとその心配は生まれない。確かに、この設計手法を使用することにより、柔軟にUIを変更できるプロダクトとなった。

今後の展望

以上で述べた二つの設計手法から学んだことは不安定なものに依存してはいけない、責任を複数持たないといけないということである。これらの要素は一つの設計手法に依存しない普遍的な概念なので、今後今回使用した設計手法とは違う設計手法で開発する際もこの概念を取り入れたいと考える。