1. 開発言語の選択

Lambda関数構築時に、開発者の得意な言語をそれぞれの関数で選定したため、複数言語が混在したシステムとなった。

AWSのLambdaでは、複数の言語をサポートしているため、開発工程前にコーディング規約を定める必要がある。

⇒指摘にもある通り、なぜ言語を統一する必要があるのか述べないといけないが、ぱっと思いつかない。むしろ、複数の言語をサポートしていることを利点ととらえている記事もおおくあり、明確な理由がない限りはこの課題を避けたほうがよいと考える。

代替案

(1)機能がFunctionごとに分割されていることに起因する処理連携の難しさ。

実体験：

今回の検証では、アジャイル風の開発にのっとり、簡単な処理フローを検討し、各人がFunctionを作成したが、機能間の連携がうまくいかず、機能を修正したケースが多発した(例：引数の型が受け取り側と渡す側の間で異なり、処理が落ちた)

実体験から考えられるLambdaの課題：

機能の細分化、独立化が容易であること自体はLambdaのメリットであるが、変更容易性・拡張性を踏まえた設計をしないと、機能間の連携の複雑化を招く恐れがある。

⇒機能間の連携の複雑化は、ネットの記事でも触れられていたような気がしていて(すみませんすぐにはでてきませんでしたが・・・)、Lambdaの課題として定義してもよいかと考えました。