

1 はじめに

1.1 本日の講演の趣旨

本日の講演の趣旨は、開発パートナーとの良好な関係を築く方法についてです。

開発パートナーとの付き合い方は、なかなか難しいです。具体的にいうと、以下のような局面が多いと思います。

- ・ 要件定義がはかどらない（なかなか決まらない、仕様を理解してもらえない）
- ・ 設計で矛盾が生じる（勝手に解釈する、矛盾あるまま進める）
- ・ 開発・テストで遅延する（不具合が解消されない、テストが停止する不具合が出る）
- ・ 稼働しても安定しない（バッチが朝までに終わらない、不正な処理が改修されない）

開発パートナーへの不安や不満は、常にあると思います。ただ、不安だ、不満だ、と漏らしても何も解決しません。

また、開発パートナー側も、お客様との良好な関係を構築したいと常に思っています。

不安や不満を減らし、開発パートナーとの良好な関係を構築するために、開発する側の観点から情報システム部門の方に留意してほしいことをご紹介します。

あわせて、アジャイル開発についてもご説明いたします。アジャイルの特徴・性質、特有のリスク・課題と低減・解決策もご紹介いたします。

2 請負と準委任

まず、今日の話は、請負契約と準委任契約から始めましょう。

請負契約は完成までの金額固定、準委任契約は完成までの金額が不定と思われるかたに、よく出会います。実は、請負契約も準委任契約も、金額は固定です。

請負契約は、〇〇が固定（＝決定している）です。結果、金額が固定になります。

準委任契約は、△△と□□が固定です。結果、金額が固定になります。

〇〇を固定＝最終決定できますか？ そうでないと、請負で金額が固定にはなりません。

以降、準委任、請負ごとに、開発パートナーとの良好な関係を築く方法を考えていきます。

3 準委任の場合の開発パートナーとの付き合い方

準委任は、工数と単価が固定です。依頼が明確で、依頼を受ける側が正しく理解して設計・開発をしてくれれば、無駄がありません。請負より安く仕上がる可能性が、あります。

一方で、依頼を間違えると、間違ったとおりに出来上がります。開発パートナーとの付き合い方の話の前に、準委任契約はアジャイルと親和性が高いので、ここでアジャイルの解説をします。

4 アジャイルの定義

一口にアジャイルといっても、その定義は曖昧です。まずは、今回の講演におけるアジャイルの言葉の定義を確認していきましょう。

4.1 アジャイルの手法

今日でいわれるアジャイルの手法のうち、特徴的なものを以下に紹介します。

・Scrum (スクラム) / Sprint (スプリント)

スクラム単位で、スプリントという工期を設定します。1スプリントは、たいてい、1~2週単位とします。

・インクリメンタルアプローチ

段階的に作りましょう、ということです。積み重ねれば理論上どんなに大きなシステムでも開発完了します。同時に、以下のような問題・性質を持ちます。

- 1) 積み重ね方を間違えると、手戻りが多発し終わらない
- 2) まとまった機能を一括リリースする場合は最後の部品が出来上がるまで待たされる

・タイムボックスアプローチ

期間と人数=工数固定という意味です。補足資料の「[タイムボックスアプローチ](#)」をご覧ください。期間と投入人数が固定されると、工数固定となります。この固定された工数の枠内で実現可能な作業を推進する、という考え方です。

・テスト駆動 (Test driven)

テストを中心に、作業を組み立てようということです。最終出口であるテストから始めて、テストで終わるといえます。

5 アジャイルのメリット

ここまでアジャイルを説明してきたところで、アジャイルのメリットを考えてみましょう。

5.1 速い

ものづくりが、速いです。Waterfallより、はるかに速いことは間違いありません。体感できます。

5.2 成果物最低限

アジャイルでは、成果物最低限というのがウリです。

5.3 楽しい

おそらく、アジャイル最大のメリット。それは、「作って楽しい」です。

すくなくとも、アジャイルを採用すれば、いま流行の働き方改革になります。また、開発者の職場満足度は上がるでしょう。

6 アジャイルのデメリット

アジャイルのメリットをみてきました。対局にある、アジャイルの主要なデメリットを見てみましょう。

6.1 品質不定：デグレ不安

アジャイルは、開発が速いです。速い分、テストが不十分の場合が多いです。

【重要】アジャイルのデメリット・不安は、デグレード（機能劣化）です。

6.2 要件未定：なにができるか分からない

アジャイルは、スプリント終了時点で、なにが完成するか保証がないです。

6.3 期日未定：いつ出来上がるか分からない

期日についても、同様です。スプリントに期日はありますが、その期日が到来した時点でできたものが、完成物です。タイムボックスアプローチゆえ、とも言えます。

7 準委任における開発パートナーとの関係性

ここで話を開発パートナーとの関係に戻しましょう。準委任契約において開発パートナーと良好な関係を築くには、以下を考慮するとよいのではないのでしょうか。

7.1 依頼内容・作業インプットを明確にする

発注側が依頼内容を明確化し、作業に必要なインプットをそろえるほうが、結果が出ます。

アジャイルの説明で紹介した「タイムボックスアプローチ」を思い出してください。準委任契約も、タイムボックスアプローチです。タイムボックス終了時に、できるだけ多く、できるだけ使い物になる成果を出せるよう、チームとして相互協力できると、良好な関係が築けます。

7.2 優先順位を明確にする

依頼した内容について、重要度・優先度も開発パートナーと共有しましょう。アジャイルで、スクラム（小チーム）・スプリント（1-2 週間）という説明をしました。準委任契約も似たような進め方が向いています。

7.3 期待するアウトプットを事前に決めておく

開発パートナーから受け取る成果物のアウトプットイメージを事前に決めておくといいです。

成果物の詳細度、決定しておきたいこと、明確化すべき箇所などを事前に定義しておき、開発パートナーから提示される納入物を検品する際のインプットとします。

アジャイルを紹介したときに、テスト駆動（Test Driven）という考え方を紹介しました。準委任の作業においても、最後の結果を想定して作業を進めるのがよいと思います。

上記 3 点を、準委任契約において開発パートナーと良好な関係を築くポイントとして挙げました。ただ、これら 3 点は、情報システム部門だけでは、なしえないです。特に要件定義段階や、ユーザー受け入れテストなどの局面では、システムを実際に使うユーザーの協力が必要です。

社内ユーザーの協力を取り付ける役割を、開発パートナーは情報システム部門に望んでいます。

8 請負における開発パートナーとの関係性

準委任の次に、請負を前提に開発パートナーとの良好な関係を考えてみましょう。

8.1 なんでも請負にしない

請負は、仕様固定です。以下のような作業を無理に請負にしないよう注意しましょう。

- ・ 要件定義（仕様を決める取り組みで、仕様を固定できません）
- ・ データ移行（何がでてるかわからないので、仕様を固定できません）
- ・ ユーザー研修（ユーザーの IT リテラシー、やる気に依存します）

8.2 Change bank 構想

Change bank（直訳：変更銀行）という考え方を紹介させてください。[補足資料の「Change bank 構想」](#)をご覧ください。

要件定義 = 仕様決定は、完璧にはなりません。人間がやることです。大なり小なり、必ず、抜け漏れ・矛盾点があります。抜け漏れ・矛盾点をなくすために要件定義期間を倍にするよりは、ある程度で見切りをつけるほうが、システム完成までの期間が短くなり、費用も少なくすみます。そう考えると、発注者側・受注者側両方とも、仕様には抜け漏れ・矛盾がある前提で仕事をすることになります。なので、抜け漏れ・矛盾を見越して、期間と工数 = 予算の裏打ちをしておく必要があります。その裏打ちを、Change Bank といいます。

8.3 捨てるものを決めておく

優先順位の低い機能を捨てる準備をしておいてください。

要件定義 = 仕様書を発注者側が完璧に揃えましたか？

システム開発 PJ を頓挫させるのと、少し欠けても完遂するのとどちらがよいですか。

捨てる機能を、発注者側で事前に決めておき、リスクに備えておいていただくと、開発プロジェクトが円滑に運営できることが多いです。

ご清聴、ありがとうございました。