■ 日米独を理解せず、米国発のアジャイル開発翻訳本の内容をそのまま日本で実施してしまうと良いソフトウェアは作れません。

日本と米国のソフトウェア業界には、決して無視できない大きな差異が存在します。代表的な日本と米国の違いを見つめ直してみましょう。

- ❖ 日本で主流のソフトウェア開発は、**金額固定・納期固定・仕様固定で完成品を届ける請負開発契約**で、かつ、親和性の高いウォーターフォール開発スタイルで実施している。
- ❖ 日本ではソフトウェアは**「金で買う」**という価値観が主流であって、受注側に完成責任があると考えている。 会計・契約部門などは、仕様を満足すれば誰が作っても同じものができると思っているから、**「できるだけ安く買う」**ことが目標となっている。
- ❖ 日本において技術者の流動性は低く、経験とスキルが高い多くの開発者とプログラマはソフトウェア開発会社(SI 会社)に囲い込まれている。
- ❖ 日本の多くの開発者やプログラマは、コミュニケーションに不得意意識があり、面倒な作業だと思っている。

	日本のウォーターフォール	米国のアジャイル
契約制度	完成請負 (金額固定 納期固定 仕様固定)	稼動払い (人数×スキル単金×期間 ≒ Time&Material契約) ≒派遣
価値観	100%は要求書通、 精神させて、安く買う。 (建築アナロジー)	ユーザ企業からの応援に、常 最人材を集めて、 一緒に作る。 ≒内製
	失敗リスクを事前回避する のが正義。 高品質主義、100%主義、 計画主義。	速・チャレンジし、軌道修正す るのが正義。 品質はボハリ8トレードオフ、 グローバル水準(=低い)、 市場主義。
雇用制度	5社程が組込む人材中心、 発注者側には下請人は 少数。	流動性ある高スキル人材市 場から調達優先。 ≒派遣
ソフトウェア技術	発展途上国(技術輸入)	先進国
(ソフトウェア・コード)	外国語	プログラム＝母国語で書かれ たロジカルなドキュメント

■ 1-2 日本と米国の開発スタイルの違い

■ 1-3 ソフトウェア制造に対する価値観の争い

図 1-2, 図 1-3 に示すように日米にはソフトウェア開発の契約制度、ソフトウェア開発に対する価値観、プログラマなど技術者の雇用制度、ソフトウェア技術の成熟度など、重要な差異が存在します。 それらの差異を理解せず、米国発の翻訳本の主張を**そのまま現在の日本に適用してはいけません。**

【警告】 アジャイルソフトウェア開発宣言中の、『契約交渉よりも顧客との協調を』の部分は、『契約書別などなしで才何でも良い』と言っている訳ではありません。

アジャイルソフトウェア開発宣言(巻末の補足 1)にある『契約交渉よりも顧客との協調を、』という文言部分を、あまりにも素朴に受け止めて契約を気にしないことは大変危険です。

アジャイル開発において、意味のない見積りに時間をかけること、および、それをベースとした契約交渉に時間をかけることは大きな無駄です。 しかし、請負開発契約で進めるアジャイル開発は失敗です。 請負開発契約でアジャイル開発を実施してはいけません。 アジャイル開発で良いソフトウェアを作りたいのであれば準委任契約(技術者スキルと稼動量実績値で支払う契約)もしくは派遣契約を締結してアジャイル開発を実施しなければいけません。

■ 日米の違いを理解し、不足するものを整備・陶汰してアジャイル開発に望みましょう。

最初に**発注側組織がアジャイル開発に対応する組織に変革**しなければいけません。 そのためには次の 3 つの視点が重要です。 そして**経営幹部が危機感を持って推進**しなければそれらは実現できません。

【改革視点 1】 意思・価値観
【改革視点 2】 責任・制度
【改革視点 3】 PO 人材育成

これらを 2 章 幹部の心得と、3 章 PO の心得で具体的に考えていきます。

■ **アジャイル開発の推進の動機は、発注側にあります。**

アジャイル開発を日本で進める正しい順序は、以下になります。 受注側がアジャイル開発の実施を発注側の協力なしに推進することはあり得ません。

Step1-1 発注側・経営幹部がアジャイル開発を理解し、求め、推進する。
Step1-2 発注側・経営幹部がアジャイル開発可能な組織に変革する。
Step1-3 発注側・PO がアジャイル開発を理解し、納得し、主体的に推進する。
Step1-4 受注側・責任者が人材をそろえ準委任契約で受注する。
Step2 発注者は開発チームから信頼された仲間になる。
Step3 受注者全員が日々、全体と優先度を考え行動する。

WF 発注側　AG 発注側
WF 受注側　AG 受注側

2. 発注側 経営幹部の心得

まず発注側がアジャイル組織に変革すべし。
アジャイル開発のメリットを享受するのは発注側の経営者である。
ソフトウェア開発の現実を知り、日米の格差を知って、
発注側経営者は組織価値観を変革し、制度を適合させ、
PO 人材を育成し授護すべし。

2.1 アジャイル開発は、あなたの望みを実現する手段

■ **ソフトウェア開発の競争力を上げるためにアジャイル開発を採用・推進するという意思を表明してください。**

アジャイル開発を推進するために、最初に重要なのは、あなた自身の意思です。

- ❖ 情報システム(ソフトウェアサービス)の開発力が自社の競争力の源泉だと思っていますか？
- ❖ 米国企業や国内ベンチャー企業等のソフトウェア開発力を脅威に感じており、

自社のソフトウェア開発力が低いことに危機感を持っていますか？

これらに同意できるでしょうか？ 「Yes」であって、現在ウォーターフォール開発スタイルを採用しているのであれば、アジャイル開発スタイルに変革するべきでしょう。

■ **あなた自身の危機感と覚悟を関係する組織の人々に伝え、あなたが変革をリードしていかなければいけません。**

■ **アジャイル開発においてソフトウェア制造の責任は、お金を出す発注側になります。**

■ **アジャイル開発の責任者はプロダクト・オーナ(PO)です。あなたは PO を育成し、授護しましょう。**

あなた自身の価値観(図 2-1)は、以下のいずれでしょうか？

【価値観 WF】 情報システムは金で買うものだ。 誰が作っても良いし安い方が良い。

【価値観 AG】 情報システムは自社が主導しソフトウェア技術者の協力を得て作り上げるものだ。 情報システムは市場の動向に合わせて継続して変更していくものだ。

アジャイル開発を推進するのであれば、**【価値観 AG】**に同意できると思いますが、その場合、**ソフトウェア制造の責任は自社側になることに抵抗が必要**です。 きっと、この変化に対して、これまで**【価値観 WF】**に慣れた組織や担当者からの抵抗があるでしょう。 お金を払っておいて自社で製造の責任を負うのでは損をしていると思う人もいるでしょう。 しかし、市場競争を含めた大局で見れば競争力の低いソフトウェアをお金と時間をかけて得ることの方が大きな無駄なのです。 あなた自身の危機感と覚悟をそれらの組織の人々に伝え、説得し、再教育し、変革をリードしていかなければいけません。

■ 2-1 社長(経営幹部)の価値観

2.2 組織をアジャイル開発に適合させる

あなたがアジャイル開発の推進を表明した後、あなたの組織がアジャイル開発を実施できるようになるためには、あなたはいくつかの変革をリードしなければいけません。

■ **価値観が浸透するまで繰り返し提示しましょう。**

アジャイル開発が必要になる背景を、あなたの組織(発注側の関係組織)の人々が理解しなければいけません。

- ◆ 情報システムの開発が企業の競争力の源泉です。
- ◆ ライバル(米国企業や国内ベンチャー企業等)はアジャイル開発によってその競争力を実現しています。
- ◆ ソフトウェアは物質いではなく、自らが優秀な技術者の協力を得て作り上げるものにしなければ、競争力のある真に良いソフトウェアは完成しません。
- ◆ アジャイル開発ではソフトウェアの開発責任と仕様コントロール権は発注側にあります。 見積り責任も完成責任も発注側にあります。

これらの現実、時代の変化に応じた価値観の変化を受け入れるには、おそらく抵抗もあるでしょう。 あらゆる人に、あなたの意思と価値観を示すことで解決していかなければいけません。

■ **契約書別(準委任契約)を整備しましょう。**

アジャイル開発を発注する場合の契約種別は、請負開発契約では失敗します。 準委任契約もしくは派遣契約にて実施すべきと開発組織および契約組織に宣言し、必要なら社内制度を整備しましょう。

- ❖ 抵抗例: 見積り競争が無いことで不安になる。
- ❖ 抵抗例: 準委任契約自体に抵抗感がある。お金を出して、発注側が責任を持つことに理解が示せない。
- ❖ 抵抗例: 未だにソフトウェア開発を建築とのアナロジーで捉えていて、時代の変化に着いていけない。

■ **契約人月単金の基準を見直し、優秀な人材を確保できるように柔軟性を持たせましょう。**

アジャイル開発では、できるだけモチベーションとスキルが高い技術者を確保し、コミュニケーション・コストを下げるために少数精鋭のチーム作りをするのが理想です。 契約の人月単金を下げる交渉をすれば、技術者の質が下がります。 スキルを正しく評価した適切な単金基準を用意しなければいけません。 実際に働いてもらいスキル貢献度に応じて金額を調整していける仕組みが最も合理的です。

- ❖ 抵抗例: 人月単金を競争で下げることをしないので不安になる。
- ❖ 抵抗例: 人月単金の基準を作れないので判断できない。
- ❖ 抵抗例: 従来の慣習に従いたい。

■ PO に与える責任に応じた権限を与えましょう。【権限・責任バランスの原則】

アジャイル開発では PO に開発完成責任が発生します。 PO には、その責任に見合った権限が与えられるように社内制度を整えましょう。 責任だけ大きくて権限が無い開発では失敗は目に見えています。

- ❖ 抵抗例: PO の役職が低すぎるから重要な権限は与えられない。
- ❖ 抵抗例: PO がまだ若すぎるから重要な権限は与えられない。
- ❖ 抵抗例: その権限を PO に与えると、上位職の役割存在意味が無くなるから与えられない。

2.3 PO 人材を育成し、授護する

■ **PO 人材への期待は、一般的に言われるビジネスのリーダシップと同一です。**

PO 人材への期待を表明して、PO として活躍できる人材を増やしていきましょう。
では、PO 人材への期待は何でしょうか？

【アジャイル開発の PO 人材】
PO 能力 ＝ ビジネス・リーダシップ ＋ 情報処理スキル

アジャイル開発の進め方は経営者のビジネス・スタイルに近いものになりますから、ある程度の情報処理スキルを持っていれば、後は、一般的に言われるビジネスのリーダシップの能力と同になります。 詳細は他書籍に譲りますが、たとえば以下のような期待があります。

- ◆ 視野が広く現実を見ろ力がある。(競合状況、ステークホルダ都合、サービスと市場動向、エンドユーザ心理、開発現場の実態と課題など)
- ◆ 費用対効果など、ビジネスセンスがあることも重要。
- ◆ バランスの良い判断ができる。 優先度判断力があり的確。 判断理由(ビジョン)を言えることが望ましい。
- ◆ 間違った場合にも素直に謝罪し、訂正できる力がある。
- ◆ 責任を完遂するために、考え続ける習慣がある。

- ◆ 良いソフトウェアを目指すモチベーションがある。
- ◆ 難題には自らが先頭で立ち向かう姿勢がある。
- ◆ 人格面で信頼されている。
- ◆ コミュニケーション力がある。
- ◆ メンバを育成する。

自組織内に PO がいなければどうすれば良いでしょうか？ 自らの後継者候補をまずは PO にして育てていくこと、一つの方法でしょう。

情報処理スキルとビジネス・リーダシップのどちらか一方を選ぶのであれば、ビジネス・リーダシップを選び、適切な技術面のアドバイザを同伴させます。

POは、Win-Winに向けて、
最適なバランスをとり続ける存在

■ 2-2 PO はバランスを維持する存在

■ **PO に対する古い価値観の組織からの抵抗に、あなたが授護しましょう。**

組織が、アジャイル開発に対応できるまでの間、PO は組織の各所から様々な抵抗を受けることになります。

- ❖ 抵抗例: 新しい価値観とこれまでの社内規定との整合性が取れない。
- ❖ 抵抗例: 新しい仕組みを導入するには、これまでとは異なる法制度の適用を学び考えて、新たな規定を作らなければいけないので面倒である。
- ❖ 抵抗例: そもそも新しいことが嫌である。 前例がない。

このような前例主義者や無思考主義者と PO を戦わせてはいけません。 時間の無駄です。 **PO は、社内の抵抗と戦うよりも、良いソフトウェア開発に集中してもらうべきです。** 情報システムの開発が会社の競争力を左右するのであれば、それらの組織の課題をできるだけ PO から切り離し、あなた自身が解消していかなければいけません。 再び、あなた自身がその場に立って覚悟と価値観を伝えることからはじめましょう。

■ **PO の失敗に対する適切なセーフガードを用意しましょう。**

アジャイル開発ではソフトウェア開発の完成責任が PO にあります。 そして、ソフトウェア開発は極めて複雑で難しい作業なので、失敗することもあります。 一方、ウォーターフォール開発であれば、失敗の責任を発注会社に転嫁することができます。 良いソフトウェア開発を作りたいと強く思わなければ、**責任の重いアジャイル開発を**

逃げ、責任が軽く自分が楽できるウォーターフォール開発を選択することも非推奨できないでしょう。 よって、アジャイル開発を PO が選択していくためには、アジャイル開発が上手く行かなかった場合でも、きちんと救済できる制度が必要なのです。

ところで先にも述べましたが、PO はビジネス・リーダと同様な判断を行います。 つまり、ステークホルダの事情の変化、競合企業による新サービスリリースなどの影響をダイレクトに受けるでしょう。 それらに起因するビジネス・リーダの方針変更を失敗と呼ぶのでしょうか？ アジャイル開発でも一緒です。 上手く進まない原因をきちんと分析し、サービスリリース時期の延伸や開発中止などの判断を適切に行わなければいけません。

- ❖ 失敗原因例: ステークホルダの計画を前提に進めていたが、ステークホルダの方針が変わってしまった。 予測不能だった。
- ❖ 失敗原因例: 競合企業が新たなサービスを開始し、現在開発しているシステムの価値が大きく低下しまった。 予測不能だった。
- ❖ 失敗原因例: 開発システムの根幹となる技術の困難さが具体的に判明してきた。 実際に試して判明した。 試すまでわからないことだった。

■ **PO の失敗に対しては、即応することでリスクを削減しましょう。**

開発の途中で上記のような状況が判明したらその時点でウォーターフォール開発では難しい方針変更を、アジャイル開発では翌日からでもすることが可能なのです。 もし、PO からのそれらの悪い状況のエスカレーションが無く手遅れとなった場合には、その PO はアジャイル開発の本当の進め方を理解していないか、社内の情報伝達に問題があった可能性があります。

開発メンバから PO が信頼されなかった場合など、開発現場のリーダシップに関わ