る問題には、あなたが適切な教済をしていく必要があります

■「アジャイル開発をやりたい」、と提案してきた担当者を活かしましょ

あなたの会社の若手の担当者がアジャイル開発を実施したいと提案してきたとき、 あなたはどうすべきでしょうか? 若手で情報処理スキルの高い人材にとって、7 アジャイル開発を推進して成果をあげているかもしれません。 あなたの会社がまた アジャイル開発を主流にできておらず、先に述べたかなたの覚悟と価値観があるの ファイル・回答を主席してきていかり、かしまでいたかなどの意味で簡単級のゆのか であれば、そのチャンを活めて、力林を伸ば、組織を変革していきましか。し かし、あなたがアジャイル関係に魅力を感じないのであれば、その人材があなたの 会社の将来性を見限ったとしても、その判断は受け入れてあげましょう。 どちらがガ ラバゴスだったのかは時代が決めてくれるはずです。

■「アジャイル開発をやりたくない」、と相談してくる PO や担当者もい ます。

負担を求め、責任を負わせます。 新らしい事もたくさん学ばないといけません。 自 ら、全体の状況を捉え、都度、最善を考え、責任をもって優先度を判断し続けなけれ ばいけません。開発者とはコミュニケーションを丁寧に行って高いモチベー を維持して開発を続けてもらわなければいりません。 発注側の担当者も発注先の 開発担当者と対等に議論し開発を進めていかなければいりません。 さらに、・・・ これらが耐えられないと申告する PO や担当者が出たらどうすれば良いでしょう

まった作法を繰り返すだけでそれっぽく進められた経験から抜け出すことができない 担当者もいるかもしれません。ウォーターフォール開発は全て消える訳ではないで

2.4 アジャイル開発の不都合を理解する

アジャイル開発では動作するソフトウェアが早い最階で見えてきます。しかし、それが簡用リリースできるまでには、もっともっと時間が かかります。

ることでしょう。 しかし、アジャイル開発では、品質確保より前に PoC (Proof o Concept.概念実証据)としてリリースされ、使い勝手などの検証や試用を実施すること になるでしょう。 それらのフィードバックを受けながら、その後に品質を高める作業 たななくにより。 くれらのパーティックを支げなから、その歌への質を向めの日本を を実施していくことになります。 よって、「動いす」と、「商用適用可能」の間には時 的な大きな時間遊があります。 動いすいを見て、「すぐサービスを開始しろ!」と、 間違った指示をしてはいけません。 説解しないように注意しましょう。

■ ウォーターフォール開発の方が適したケースもあります。

全ての場合にアジャイル開発が向いているとは限りません。 案件のタイプがアジ イル開発に合っているか? は考える必要があります

アジャイル開発価値より他の価値が重要な場合

- ◆ 他社と競争する必要が無い場合。 ◆ 要求仕様が固まっていて、誰でもいいから安く作ってほしい
- ◆ 人命や重要社会基盤であって、早いサービス開始よりも高品質の

他社に開発保険を求めたい場合(ただし受往会社が存在する情報):

- とりあえず予算内・期間内で完成する 自らの完成責任を回避したい場合。 ◆ 金でリスクを回避したい場合。

自社の価値観をアジャイル開発適合に変えることができない場合。 開発責任を委ねられるPO人材がいないし、育成もできない場合。

- **きく、ドキュス・トペースが強率的な場合):** サンドウェア開発規模が大きくチームが 10 人を超える場合。(適切) たチーム分割で回避するケースかり
- ればならない場合。(適切なツール使用で回避するケースあり)

■ そもそも何を作るかが決まっていない場合には、どんな開発スタイ

敗します。 製造請負契約であれば、たとえ見穂り仕様が・ワポ1枚であってもその 失敗責任は(その仕様で見穂)を出した)受注者が負いますが、アジャイル開発の場 合には自社で負うことになります。 失敗に対して、短絡的に「アジャイル開発は駄目

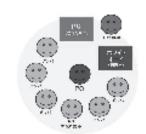
だ」、「ウォーターフォール開発に回帰せよ」と指示する前に、失敗の原因をきちんと 分析してください。情報からは POからも集めますが、全ての情報が得られない場合もあります。 あなたの情報収集能力と反省も問われるかもしれません。 まずは、 あなたゴールイメージの具体化に対して、あなた自身が PO にどう協力できていたの かを考えるのも一つの原因追求アプローチです。 きれいな PowerPoint 資料だけで

3. 発注側 PO の心得

3.1 プロダクトオーナ(PO)の責任と権限を理解し納得

■ ウォーターフォール開発は、発注者が「受身」でも一応完成します。

- ターフォール開発のほとんどは製造請負契約とセットです。 あ (生来のリオーアーノオール側側のなこれとは裏面面両外等ができていて)。あなに が帰注仕様者を離れなく書くことができて、かつ、予算内・期限内で受注してくれる 会社が見つかり製造諸負開発契約を禁べれば、あなたの勝ちです。 後はあなたが 不在であっても、納期には仕様を(最低限度に)満たす納品物が得られます。 仕様 が十分に書かれていれば、漢足できるソフトウェアの可能性だってあります。 仕様 解釈のズレの発生を避けるために、丁寧なレビューを実施することはあるとしても、あ なたは**常に受命**で構いませんでした。 開発会社から提出された進捗報告を眺め、 予定と実績の乖離確認を行い、もし遅延していれば遅延回復策を提案してもらって 味することだけだったかもしれません



Commence of the second

図 3-1 PO はアジャイル開発の中心

■ アジャイル開発は PO が「主体的」でなければ完成しません。

様も実装も**装置がに指揮**していくことになります。 それを可能にするために製造請 負契約ではなく推委任契約を締結しているのです。 あなたの期待に近いものが作 れるように指揮できる代償として、発注側のあなたがソフトウェアを完成させる責任を

■ PO がソフトウェア完成の責任を負うことを納得しなければいけませ

ソフトウェアを完成させる責任は、明確なゴールイメージがあり、話のわかるステー クホルダが相手で、質の高い開発火バを集め、良いチームを作り、適切な期間をかければ、それほど困難ではないかもしれません。 しかし、ゴールがはっきりせず、ス テーケホルダが信頼できず、頭数だけのメンバを集めて、短い納期に間に合わせる 開発だったら、誰がやっても失敗するでしょう。 PO になる前に、成功のための条件 をどの程度満たしているが、現在満たされていない条件をどうやって満たしているか。 を責任者の根点で考えるべきでしょう。 そして経営幹部に不足する条件を満たすた めの支援を要求しなければいけません。 もし、あなたの価値観がリスクと責任を回 遊することを優先するから、POを受諾しかい、アジャイル開発を採用しかいのが割合 逃すべきかもしれません。

- ◆ タイムリーにステークホルグと調整して優先期位を決めることをは面倒な
- ◆ 現場の悩みに付き合って、都度、方針や優先順位について判断していく
- たくさんのレビュー依頼を裁くなど、知力・体力に限界を感じる。
- あなた自身が開発するもののゴールに納得できない。
- 開発するものが全くイメージできない。何を開発すればいいのかわから

■ 責任の大きさに伴う権限が与えられていない場合、PO になるのを

いがとの調整の実効的な権限や、納品物を検収する実効的な権限など を、自ら持っていることが重要です。 権限自体を持っていなくても、権限を持った上 lとの連携作業で進められるなら大丈夫がもしれません。 しかし、たとえば全くステ ・クホルダと調整する権限が無い担当者がPOになる場合には、矢敗するリスクは乳 常に高くなるでしょう。 検収権限がない場合には、自らの判断の前に都度、その検 収権限者と調整しなければ、責任ある判断ができないでしょう。 PO になる前に経針

■ PO はアジャイル開発の SPoP です。 自らの健康を維持しましょう。

PO はアジャイル開発の中心です。 PO が不在になればアジャイル開発が停止す るリスクがある SPoF (Single point of Failure、単一故障点)です。 健康維持には十分 注意しなければいけません

■ 複数名が PO テームを構成することも可能です。ただし、簡単では

しかし、1 名だけで実施する体制にはプロ あります。 そこで2~3名で PO がチームを組むことが考えられます。 しかし、容易

す。 複数の船頭が発生すると逆にアジャイル開発は混乱してしまいます。 開発メ バにも主体性が生まれなくなります。 PO チームとして開発メンバから信頼される

3.2 発注側幹部がアジャイル開発の価値観に納得し

1章で述べたように、アジャイル開発は発注側幹部の意思から始まるのが正し xタイルです。しかし、不幸にも逆順となることが良くあるでしょう。 なぜ、逆順にな るのでしょうか、

■ PO 自身がアジャイル開発を実施したいと望むのは、開発のスピー ドアップやコスト低減を求めたときです。

PO 自身がアジャイル開発を望むとさとは、どんな時でしょうか? 一つは、情報処理スキルが高い人材がアジャイル開発を体験し、そのスピード感が忘れられないとさ ではないでしょうか。もう一つは、ウォーターフォール開発の変更対応の耐さやスピ

ーターフォール開発では受注側が全てのリスクを負うため、見積額はリスク料金を含 んだ高いもの 間発線間もリスクを見た長い機関にかります。 また 当初の仕様で さくなるケースもあるのです。 自分でチームを作って自ら指揮して作った方が、安く 良いものが作れると思うことがアジャイル開発を実施したいと思う強い動機となるでし よう。そのほか、競合社の進め方を知った時ということもあるかもしれません

■ 社外と競争する必要がない組織であれば、組織の幹部がアジャイ ル開発のメリットを求めません。

るような企業が最大限に享受します。しかし、世の中には異なる価値観の組織も多 くあります。 年度内に与えられた予算の中で、確実に買う、ことが重要な場合だって あるはずです。そんな組織にアジャイル開発はそもそも適合しませ、

■ 社外と競争し情報システムが競争力の源泉となる企業なら、幹部に アジャイル開発という技の知恵を授けましょう。

発の競争力が上がること、それは幹部にとっても企業の競争力を左右 する重要な情報のはずです。 幹部がもレアジャイル開発を知らないのであればそ り予法の情報を提供すべきです。 さらに、深く理解してもらい、制度の整備等も合

が価値を理解できなかったら、競合他社の成功を示し て、気づきを待ちましょう。

競争する会社の幹部に、競争力を向上させる手法がもし魅力的に響かないのであ れば、その会社は真の競争をしていないのかもしれません。 もしくは幹部にはもっ と優先する他の価値観があるのかもしれません。 そんな中で、あなただけが完成責 任というリスクを負うか否かは、あなたの自身の判断です。 少なくとも、最初から大き なアジャイル開発を実施することは避け、小さな成功を積み重ねて実績を積んでい さましょう。また、契約としては派遣契約を使って技術者を集め、内製するのがわか りやすいでしょう。 結果として、類似の他のウォーターフォール開発と比べると良い ものが作れるでした。その悪をアールしていてことも負いでした。 また、酸合金 業等がアジャイル個発で成功している事例については情報として伝えて行きました。 あけは、使命感を持って企業を変える努力をするもとし、もっと力を発揮できる環境を 探すもよし、幸運を祈るだけです。

-ルに対して、価値のある部分をまず動かして 3.3 ⊐-全体の目処をつける

■ ゴールを決めるのは PO の責任です。

アジャイル開発では、開発メンバにゴールを理解してもらって協力してもらわなけ ればいけません。 万が一、ゴールを PO が正しく理解できていないのであれば、安 まに関金をはしては、寸音せん。ゴールを明確にするためには、ブロクスと、 数に関金をはしては、寸音せん。ゴールを明確にするためには、ブロクスと 以外にもっと重要なことがあるかもしれません。 一旦、開発が動き始めてしまと、そ れだけで開題を安心させてしまからしれません。 手段を目的化しては駄目です。 何かやっている成じの溶出も駄目です。間巻の前にゴールを用降化する部をかと

■ ゴールまでの計画を見積る責任も、発注側のPOです。

それに必要な体制規模と期間を決めるのも PO です。 当然、ウォーターフォールと 四様に精度高く知時間で見積もる方法は存在しません。 アジャイル開発におい 期間も体制も見稽もれないのに完成者任を果たす最大の手法は、**意要な部分の個**

- 期待される性能を満足できるか見通しが全く無い。

- プログラムの核として利用するバッケージ・ソフトウェア/クラウド機能/ OSS (Open Source Software) の理解が十分では 他システムとの連携方法の理解が十分ではない
- ・ 期待するユーデインタフェースの実現方法の目処が立っていない。

解決すべき重要な部分の見通しが不明な時点で、それを解決せずに見積もっても 度胸(KKD法のD)で見積もったとしか言いようがありません。 最初にその部分の解 決に日処をつけましょう。 そして、その結果を受けて見積りの構度が向上するか、 チームの皆と議論してみましょう。 次に重要な部分があって、そこを解決しなければ やはり全体が連続が下透明からしたません。そのとさは、そこを解決していきましょう。 重要な部分が明確でない場合には、中後となる機能と短期間で簡易に軟作してみましょう。 それによって、どこに難しきがあるのが見えてくるはすです。 いく つかの重要な部分について解決の日処が付くと、後はそれらの経験とこれまでのチ ムメンバの経験を合わせて見積りができるタイミングがいつか訪れるでしょう。 そ かります。 見積もった計画を実行していく中で新たな課題が出てきたら、計画は 素早く見直しましょう。

いことを原則としましょう。

基本設計を明確にするために、一旦、動作するものを組み上げてみることをする: とは、ドキュメントを眺め続けて設計の精度を上げるよりも効率的なものになります。

しかし、注意点があります。 基本設計を決める際に作ったプログラムコードは再利 用可能でしょうか? 使える場合もありますが、無理に使わず捨てるものだと思いましょう。 もったいない病、ライン単金病、干疫日的化病等に負けてはいけません。 現 代のシステム開発では、プログラムコードよりも良い設計の方が圧倒的に価値が高い のです。 優先度判断ができない姿勢、二兎を迫う姿勢は避けましょう。 同様に、基 本設計を試行錯誤するプログラムコードに UT を実施する等のパランスの悪さも避け なければいけません

■ 未確定部分があれば、課題を分離して、仮置きして全体を進めるこ

アルゴリズムや詳細方式が決まっていない部分があれば その部分に対して簡素 ハー・フィー・マッキ個の及ぶできる。いっぱい。同の方があれば、その部分に対して開業 で交換可能なインタフェースを用意して、一旦、最も単純・原始的なアルゴリズム・詳 細力式を使うことにして、全体を進めましょう。 全体を通すことで、さらに重要な問題 が無いかを探っていさましょう。