

<div>1. 本社に2017年7月アジャイル開発</div> <div>る問題には、あなたが適切な教訓をしていく必要があります。</div> <div>■ 「アジャイル開発をやりたい」、と断固とした意思を伝えましょう。</div> <div>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よって、アジャイル開発は普通のソフトウェア開発になっています。大学時代の経験に、他社でアジャイル開発を推進して成果をおげているからかもしれません。あなたの会社がまだアジャイル開発を主流にできておらず、共に歩むあなたの覚悟と価値観があるものであれば、そのチャンスを逃がして、人材を伸ばし、組織を変革していきます。しかし、あなたがアジャイル開発に魅力を感じないのであれば、その人材が別の会社の特性を発見したとしても、その判断は受け入れてあげましょう。どちらががラビだったかは時代が決めてくれるはずです。</div> <div>■ 「アジャイル開発をやりたい」、と相談してくるPOや担当者をいいます。</div> <div>アジャイル開発は、ウォーターフォール開発に比べて発注側の組織とPOに大きな負担を求め、責任を負わせます。新しい事もたくさん学びたいと思います。自ら、全体の状況を使い、態度、最善を考え、責任をもつて優先度を選択し続けなければいけません。開発者とコミュニケーションを丁寧に取って高いモチベーションを維持して開発を続けていかなければいけません。発注側の担当者も発注先の開発担当者や対等に議論し、開発を進めていかなければいけません。さらに、・・・これらが認められたいと申告するPOや担当者が出たらどうすれば良いでしょう</div> <div>27</div>	<div>発注側の開発者 2.1 (2019年10月 10時間)</div> <div>か？ ウォーターフォール開発スタイルで開発できていた発注側が受けるスタイル、決まった作業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る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ので、あなたは選材適所と育成後のプランを考えなければいけません。</div>	<div>1. 本社に2017年7月アジャイル開発</div> <div>2.4 アジャイル開発の不都合を理解する</div> <div>■ アジャイル開発では動作するソフトウェアが早い段階で見えてきます。しかし、それが利用リソースでできるまでには、もっともっと時間がかかります。</div> <div>ウォーターフォール開発で納品されたソフトウェアは一定の品質基準を満足していることです。しかし、アジャイル開発では、品質確保より前にPoC(Proof of Concept:概念実証)としてリリースされ、使い勝手などの検証や使用を決定することになります。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を実施していくことになります。よって、「動いた」と「商用運用可能」の間には時間的な大きな時間差があります。動いたを見て、「すぐサービスを開始しよう」と、間違っただけの指示をしてはいけません。誤解しないように注意しましょう。</div> <div>■ ウォーターフォール開発の方が進化したケースもあります。</div> <div>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ャイル開発に向いているのか？ は考える必要があります。</div> <div>■ アジャイル開発組織以外の組織が重要な場合:</div> <div><ul style="list-style-type: none">経営幹部としてアジャイル開発に魅力を感じない場合。他社と競争する必要が強い場合。要求仕様が明確で、一度でもうろちる安く作ってほしい場合。人命や企業社会が懸かっている、新しいサービス開始は高品質の</div> <div>29</div>	<div>発注側の開発者 2.1 (2019年10月 10時間)</div> <div>■ 他社に開発権を渡さない場合 (ただし受注会社が存在する前提):</div> <div><ul style="list-style-type: none">どりあす予算内・期間内で完成することが重要である場合。自らの完成責任を回避しない場合。金でリスクを回避したい場合。</div> <div>■ 弊害:</div> <div><ul style="list-style-type: none">自社の価値観をアジャイル開発で変えることができる場合。開発責任を委ねられるPO人材がいなくなり、育成が難しい場合。</div> <div>■ コミュニケーションが上手でない場合 (コミュニケーションが苦手な人、ドキュメンテーションが苦手な人):</div> <div><ul style="list-style-type: none">ソフトウェア開発が大きいチームが10人を超えたら、適切なチーム分割で回避するケースありオフショア開発や自社への持ち帰り開発など、別科で開発しなければならぬ場合。(適切なツール使用で回避するケースあり)</div> <div>■ そもそも「何を作るか」が定まっていない場合には、どんな開発スタイルを採用しても開発は成功しません。</div> <div>最終的に、(物理的)ソフトウェア(ボーリング)が決まっていなければ、開発は失敗します。製造業業界であれば、たとえ見解が違っても100%の失敗責任は自社の責任は(その仕様で開発)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。実際に現れて、結果的にアジャイル開発は数日</div> <div>30</div>
---	--	--	--

<div>1. 本社に2017年7月アジャイル開発</div> <div>だ、「ウォーターフォール開発に回せよ」と指示する前に、失敗の原因をちゃんと分析して欲しい。情報は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響を聞き取りたいかもしれません。まずは、あなたがイメージの具体化に対して、あなたが自身がPOにどう協力できているの、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、上手く動作するソフトウェアは作れません。</div>	<div>発注側の開発者 2.1 (2019年10月 10時間)</div> <div>WF 発注側 AG 発注側 WF 発注側 AG 発注側</div> <div>32</div>	<div>1. 本社に2017年7月アジャイル開発</div> <div>3. 発注側 POの心得</div> <div>3.1 プロダクトオーナー(PO)の責任と権限を理解し納得する</div> <div>■ 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。</div> <div>従来のウォーターフォール開発のほとんどは製造業発祥の会社です。あなたが発注仕様書を渡ればよく書くとでき、かつ、予算内・期間内で受注してくれる会社が見つかる製造業発祥の会社であれば、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。仕様が多分に書かれていけば、満足できるソフトウェアの可能性が高くなります。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必要です。あなたが「書かなくていい」で満足して、開発会社から提出された進捗報告を眺め、予定と実績の乖離を確認し、もし遅延していれば遅延原因を把握してもらって吟味するだけだとしてもいいかもしれません。</div>	<div>発注側の開発者 2.1 (2019年10月 10時間)</div> <div>※ たとえば、POアシスタントはアーキテクト</div> <div>■ 3-1 POはアジャイル開発の中心にいる</div> <div>34</div>
---	---	--	--

<div>1. 本社に2017年7月アジャイル開発</div> <div>■ アジャイル開発はPOが「主体的」でなければ完成しません。</div> <div>一方、アジャイル開発では、図3-1のようにPOが中心に位置します。あなたが仕様も実装も「積極的に関与」していくことになります。それを可能にするために製造業発祥のやり方(標準契約を締結している)です。あなたの経験に思い入れの作られるように指導できる代議士で、発注側あなたがソフトウェアを完成させる責任を負うのです。あなたの判断が失敗すれば、ソフトウェアは完成しないかもしれません。</div> <div>■ 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ん。</div> <div>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、基のわかるターゲットが相手で、質の高い開発チームを集め、良いチームを作り、適切な期間をかければ、それと同等でよいかもしれません。しかし、ゴールがはっきりせず、ターゲットが明確でなく、複数の人のメンバーを集めて、短い納期に合わせた開発だった。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か、現在満たしていない条件をどうやって満たしていくか、を責任者の視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避することを優先するから、POを受任しない。もしはアジャイル開発を採用しないのが宿命だとされます。また、以下の悪いが先行して主体的に行動できない場合にも辞退すべきかもしれません。</div>	<div>発注側の開発者 2.1 (2019年10月 10時間)</div> <div>◆ タイムラインにステークホルダーと調整して優先順位を決めることは困難なので避けたい。</div> <div>◆ 現場の偏見に付き合っても、態度、方針や優先順位について批判していくのは困難なので避けたい。</div> <div>◆ たくさんのレビュー依頼を減らす、知力・体力に限界を感じる。</div> <div>◆ あなたの自身を開発するものゴールに納得できない。</div> <div>◆ 開発するものがイメージできない。何を開発すればいいのかわからない。</div> <div>■ 責任の大きさに伴う権限が与えられていない場合、POになるのを止めておきましょう。</div> <div>ステークホルダーとの調整の困難な情報や、納品物を検証する実務的な情報などを、自ら持っていることが重要で、権限自体を持っていなくても、権限を何となく可及の範囲で確保できるなら大丈夫かもしれません。しかし、たとえば全スチームホルダーと調整する権限が無い担当者がPOになる場合には、失敗するリスクは非常に高くなるでしょう。権限範囲がない場合には、自らの判断に照らし、その権限範囲を調整しなければ、責任ある判断ができないでしょう。POになる前に経営幹部に必要な権限は要求するべきです。</div> <div>■ POはアジャイル開発のSPでです。自らの権限を維持しましょう。</div> <div>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。</div> <div>36</div>	<div>1. 本社に2017年7月アジャイル開発</div> <div>■ 複数のPOがチームを構成することも可能です。ただし、簡単ではありません。</div> <div>POが1名だけの体制にはプロフェッショナルの可用性(継続性)として問題があります。そこで2名-3名でPOがチームを構成することがあります。しかし、容易ではありません。</div> <div>このとき注意すべきは、各PO間の権限のズレ、判断のズレを最小にすることです。複数の判断が発生するときにアジャイル開発は混乱してしまいます。開発メンバーにも主体性が生まれるようになります。POチームとして判断基準と優先順位を合わせ、開発メンバーを信頼することになるなければいけません。</div>	<div>発注側の開発者 2.1 (2019年10月 10時間)</div> <div>3.2 発注側幹部がアジャイル開発の価値観に納得していない時</div> <div>1章で述べたように、アジャイル開発は発注側の意思が前提である正しいスタイルです。しかし、不幸にも逆転することがあるでしょう。なぜ、逆転になるのでしょうか。</div> <div>■ PO自身がアジャイル開発を実施したいと思うのは、開発のスピードアップやコスト削減を求めているからです。</div> <div>PO自身がアジャイル開発を望むときは、どんな時でしょうか？ 一時的、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でいていないでしょうか。もう一つは、ウォーターフォール開発の変更対応の遅さやバードットの無さ、納品物の費用対効果の悪さ等に不満に感じるときかもしれません。ウォ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、見積もりのコストを含んだ高いもの、開発期間がリスクを見えにくい期間になります。また、当初の仕様で十分過ぎた点がある点がある、納品されたソフトウェアと期待とのズレが非常に大きくなるケースがあります。自分でもチームを作ってから前向きに作った方が、安く良いものが作れると思うとアジャイル開発を実施したいと思うのが価値観なので、そのほか、競合社の進め方を知った時にもあるかもしれません。</div>
---	---	---	--

<div>1. 本社に2017年7月アジャイル開発</div> <div>■ 社外と競争する必要がある価値であれば、組織の幹部がアジャイル開発のガッツを求めます。</div> <div>アジャイル開発の進化への対応の妨げは、インターネット上のサービスで競争する企業が増えることを受けます。しかし、世の中には様々な価値観の組織が多岐にわたります。年度内に与えられた予算の中で、確信に裏打ちされた重要な場合が多岐にわたります。そんな組織にアジャイル開発はそもそも適合しません。</div> <div>■ 社外と競争し、競争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識を授けよう。</div> <div>システム開発の競争力が上がると、それは幹部にとっても企業の競争力を左右する重要な要素です。幹部がもしアジャイル開発を知らなければ、その手法の価値を把握できません。さらに、既成概念に固まり、制度や標準を合わせて進める組織も合理的で管理を待っています。</div> <div>■ 企業の幹部が価値を理解できなかったら、競合他社の成果を示して、気づきを得よう。</div> <div>インターネット市場でサービス競争をする会社の幹部に、競争力を向上させる手法がもし魅力に感じないのであれば、その会社は真の競争をしないのかもしれない。もしは幹部にはいつと競争する他の価値観があるかもしれません。そんな中で、あなたがたが完成責任というリスクを負うのは、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難しく、小さな成功</div>	<div>発注側の開発者 2.1 (2019年10月 10時間)</div> <div>を積み重ねて実績を積み上げていきました。また、契約としては製造業を使って技術者を集め、内製するのだからやりやすいでしょう。結果として、競合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ょう。その逆をアジャイルに比べていいたいです。また、競合企業がアジャイル開発で成功している事例によっては情報として伝えていきたいと思います。あとは、使命感を持って開発する力があるのも、もっと力を発揮できる環境を築くことも、幸運を祈ります。</div> <div>■ POはアジャイル開発のSPでです。自らの権限を維持しましょう。</div> <div>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。</div> <div>36</div>	<div>1. 本社に2017年7月アジャイル開発</div> <div>3.3 ゴールに対して、価値のある部分をまず動かして全体の目的を見つける</div> <div>■ ゴールを決めるのはPOの責任です。</div> <div>アジャイル開発では、開発メンバーにゴールを理解してもらって協力してもらう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、安易に開発をはじめてはいけません。ゴールを明確にするためには、プログラム開発以外にも重要なことあるかもしれません。一旦、開発が始められてしまうと、それだけで開発を安心させてしまうかもしれません。手段を目的しては駄目です。何かがやっていると感の演出は駄目です。開発のゴールに明確化する責任などの重みがかかるのがアジャイル開発です。</div> <div>■ ゴールまでの計画を構築する責任も、発注側のPOです。</div> <div>アジャイル開発で、ゴールを決め、ゴールに向かうの全体スケジュールを計画し、それに必要な体制環境と期間を決めるのもPOです。当然、ウォーターフォール開発と同様に精度高く短時間で開発する方法は存在しません。アジャイル開発において、期間・体制・見えないものに決定責任を負う最大の手法は、重要かつ価値の解決を小さく頻りに目標へつづると価値を出す、ということです。解決すべき重要な部分と解決しきれない部分の両方があります。</div> <div>◆ プログラムの主要機能の実現方法の目処がついていない</div> <div>◆ 期待される性能を満足できない見込みが全く無い</div> <div>41</div>	<div>発注側の開発者 2.1 (2019年10月 10時間)</div> <div>◆ プログラムの根幹として利用するライブラリ/ソフトウェア/クラウド機能/OS(Open Source Software)の理解が十分ではない</div> <div>◆ 他システムとの連携方法の理解が十分ではない</div> <div>◆ 期待するユーザー/クライアントの実用性の目処が立っていない</div> <div>◆ プログラムの中心となるアルゴリズムの目処が立っていない</div> <div>◆ プログラムのアーキテクチャが定まっていない</div> <div>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据えて度々(KKD法)で見て見聞かすようなやり方があります。最初はその部分の解決に目処をつけよう。そして、その結果を受けて見聞の精度が向上するのを、チームの皆と議論しよう。次に重要な部分があって、それを解決しなければいけな全体の流れが不透明でいなくなる。それは、中絶となる機能と短期間で見聞してみよう。それによって、どこに問題があるかが見えてくるはずなんです。いくつかの重要な部分について解決の目処が立ったら、後はその経験とこれまでのチームメンバーの経験を含めて見聞ができるタイミングが来るでしょう。そして、次に、残りの部分に目処を定めて、皆と皆で納得できる見聞の計画の完成となります。見聞った計画を実行して途中で新たな問題が出てきたら、計画は簡単に見直ししよう。</div> <div>■ 基本設計を構築するための初期プログラムコードは、無理に強要しないことを推奨いたします。</div> <div>基本設計を明確にするために、一旦、解決するものを組み立ててみることを。そして、ドキュメントを積み重ねて設計の精度を上げるよりも、簡単なものになります。</div> <div>42</div>
---	--	---	--