

7. 開発担当者への声 ■ 優先度の強化や要望の強化の際も、その背景を含めて理解しよう。 アジャイル開発では、変更がより発生します。でも、その変更には理由があるはずです。変更の理由には、その背景を知ることが、POの判断を深く理解してきまします。 ■ 他のメンバーの作業状況も知り、チームの状況を把握しよう。 POとできるだけ頻りに意見を述べることが重要です。あなたが知らない、チーム全体の状況は積極的に把握していきましょう。ただし、それは時間をおく過ぎることは望まれないと思われます。毎朝のスタンディングミーティングの場などで、自分の報告・相談をできるだけ早く、他のメンバーの報告・相談も興味を持って聞いておきましょう。 ■ POの人物モデルが成長すれば、POの判断に近づけます。 スタートアップとの関係など、完全にPOと同一視点で判断することは困難ですが、POの人物モデルを通して理解が進むと、開発者がPOの判断をもっとよく予測して作業を進めることが可能になると思われます。 107	8.7 アジャイルの原則 #2 (2019/03/20 Version 1.0更新) 7.2 自らの状況を適切に伝えるコミュニケーション姿勢を持つ POは、正しい情報を持ってなければ、正しい判断はできません。あなたの情報を適切にPOもしくは関係者に伝える姿勢を持つ必要があります。では、適切に、というのはいかにいうことでしょうか？ ◆ 適切な相手 報告・相談は、適切な相手に届けましょう。作業のスクリーンや優先度の相談であればPOですが、ある製品の使い方であれば、製品ページに聞くべきかもしれません。技術的な問題であれば、メンバーの中に対応できる人がいるかをチームで全体に問いかけ呼びかけるべきです。 ◆ 適切なタイミング・態度 伝えたい内容や相手の状況とのバランスを理解して、タイミングをはかって報告・相談をしましょう。相手は何か急いでいる状態か否か、報告すべき内容は緊急性を伴うか否か、判断できないときは、短く相手に都合を聞くべきです。「その後、XXXの締め切りについて相談したいのですが、お時間大丈夫でしょうか？」さらに、作業が長くゆるる場合には、関係者のメンバー問題があるのか等、その作業への期待にあわせて報告の態度を問いかけましょう。すると、報告せず一人頑張った挙句、「できませんでした」という報告をするのではなく、途中で「苦戦中ですが」、状況を入れるべきです。 108	7. 開発担当者への声 ◆ 適切な態度 朝のミーティングなどでは、できるだけ頻りに結論を言うべきです。さらに相談が必要であれば、あらかじめ場を持って、より詳細に伝えるべきです。相手に質問を受けた場合には、相手の求める会話の態度を意識すべきです。「Yes」、「No」だけ返さないでいいや？ 判断根拠等を聞きたいのか？ 不明なら、大きな疑問から徐々に詳細に話すべきです。朝のミーティングでの報告が、昨日と内容が同じという報告は良いものか？です。同一問題を数回繰り返してはダメ。精度を細かくして、具体的な進捗を伝えなければ、進捗状況は伝わりません。 ■ コミュニケーションが難しい場合にはPOに相談しよう。 アジャイル開発では高度の深いコミュニケーションが必要で、かつ、他々にもコミュニケーション・スキルも向上することが期待されます。しかし、あなたがそれに耐えられないのであれば、POに相談しましょう。プロジェクト継続のタイミングをPOに判断してもらうことも一つの道です。 ■ あなた自身の人物モデルをPOとチームメンバーに持ってもらうようにしましょう。 POにも自らの人物モデルを正しく持ってもらわなければ、質の高いコミュニケーションには至りません。POとチームメンバーで書で示したあなたの人物モデルを正しく伝えるのに情報を提供すべきです。 109	8.7 アジャイルの原則 #2 (2019/03/20 Version 1.0更新) ■ POは現場の状況に興味を持っています。 POは作業進捗に対して、期待の期間とのズレがあった場合に、「時間がなかった理由は何？」とよく尋ねるでしょう。それは、POが現場の状況を深く理解するための質問です。「何か想定外の問題が発生しているのではありませんか？」と、同じ意図がもたれます。作業が上手く進まない原因は、遅延がPOに報告して、このままのアプローチで良いのかを判断していくべきです。POからの質問に対して、「一人悩んで、ずっと考えてました」という回答が出るケースは、アジャイル開発の進め方としては望ましい「状態」です。 110
--	--	--	--

7. 開発担当者への声 7.3 次にやるべき作業案をもって優先度を都度POと相談する ■ POの負担を下げることに貢献しよう。 ■ 技術者として、主体的な役割をこころえよう。 ■ コミュニケーションは、あなたの中のPOの人物モデルを確認し補正するタイミングだと思おう。 アジャイル開発では、作業の優先度に関わりがある場合には、POに相談する必要があります。PO自身の負担を下げて、あなたの中にあるPOの人物モデルを少し改善する姿勢を示すのであれば、あなたの考えや期待値と優先順位を同時に提示すべきです。あなたは常に受身ではなく、技術者として主体的であるべきなのです。たとえば「僕はA,B,Cの3つあります。XXXの理由で案Aが良いと思いますが、いかがですか？」と聞いてみてはどうでしょうか。決めるのはPOであっても、提案するのはあなたが良いのです。 ■ 次に、実施するべき作業が不明で、作業待ち状態に入ったのであれば、必ずPOに作業前の場で相談しよう。 アジャイル開発では、標準的契約のよう枠組みを使います。シンプルに言えば、結果に対して費用が支払われるのが製造業的契約、稼働時間に応じて費用が支払われるのが標準的契約です。つまり、POはあなたが仕事を持っていない「状態」を避けるのが標準的契約です。 111	8.7 アジャイルの原則 #2 (2019/03/20 Version 1.0更新) けたいのです。全ての作業が完了し、作業前の場が無くなったときや、他者の作業待ちに入ってから作業が無くなった場合には、必ずPOに次の作業前の場で依頼しましょう。普通は、チケットシステム上にチケットログとして残るため優先度の依頼になるはずです。それで良いものを確認して進めましょう。 ■ 仕事を完了する際にも、もし仕事をやる覚悟をただで、創的に実験期間が短くなる方法があれば、それもPOに伝えよう。 仕様は原則として守るべきなのですが、アジャイル開発ではその実質にかかる費用とのバランスを重視します。もし、若干仕様を変更するだけで、3ヶ月かかる実装が1週間で行われるのであれば、POには魅力的な提案に感じられるかもしれません。 ■ 仕事を完了する際にも、もし仕事をやる覚悟をただで、創的に実験期間が短くなる方法があれば、それもPOに伝えよう。 仕様は原則として守るべきなのですが、アジャイル開発ではその実質にかかる費用とのバランスを重視します。もし、若干仕様を変更するだけで、3ヶ月かかる実装が1週間で行われるのであれば、POには魅力的な提案に感じられるかもしれません。 112	7. 開発担当者への声 7.4 技術者として高い付加価値を提供する ■ 要求スキルの多様化に対応し、自分の付加価値を上げていきましょう。 「設計はできません」、「Javaでのコーディングしかできません」、「試験工程しかできません」などのように限定的な技術スキルのメンバーは、直ぐにアジャイル開発に対応することは難しいかもしれません。アジャイル開発では、基本設計は無しとしても、その他の工程は難しいかもしれませんが進捗が低下します。まずは、プロとして、技術処理技術について広く理解を深めていきましょう。 その昔、大量生産が重要だった時代に、工務局に専員が専務でられる流れ作業（ライン生産方式）が生まれました。その要員は常に同一工程を担当するため、後に、「単能工」と呼ばれることになりました。市場が成長し、ニーズが多様化してきた頃、多品種少量生産のモデルとして、一人または少数人数のチームで全工程を担当するセル生産方式が生まれました。彼らは全工程を担当できるため「多能工」と呼ばれるようになります。（参考：朝日新聞） さて、ソフトウェア製造業では、大量生産でしょうか？ それとも多品種少量生産でしょうか？ 当然、後者です。ソフトウェアの大量生産はコピーで済んでしまいます。皆さんは、多能工として、マルチな能力を発揮していなければいけません。もう、単一工程の専務専断ではないのです。 113	8.7 アジャイルの原則 #2 (2019/03/20 Version 1.0更新) ■ アジャイル開発について監督者を参考にしましょう。 本書はアジャイル開発を日本で実施するための大切な部分を特に技術強調して書いています。しかし、アジャイル開発の本場である米国のパラドックスを組合せて書きました。多数あります。それらにも興味を持って目を通してください。POの右腕として信頼される技術者を目標にしてください。 ■ 常に、新たな情報と技術情報を身につけ、プロジェクトの中見据的な方向についてもアドバイスしていきましょう。 首都圏であれば、各種の勉強会、ITの場などが企業持ちを超えて開催されています。積極的に参加することをお奨めします。 近年では、低コストなPOD(Print on Demand)サービスも普及してきました。業務に関連しない技術情報について、外部に発信することで理解を促すことも良いでしょう。 ただし、業務に関わる情報を外部に漏らすことは、禁止事項になります。 114
---	---	---	--

7. 開発担当者への声 7.5 相互にフィードバックし、双方が成長する ■ チームメンバーもPOが良いリーダーにならぬに留意しよう。 POが当初からアジャイル開発の成功経験が豊富な人務者であるとは限りません。もしもしたら、これまでウォーターフォール開発しか実施しておらず、初めてのアジャイル開発である可能性もあります。そうでなくても、人間は神様ではありません。完璧ではないのです。PO自身がアジャイル開発のPOとして上手に振舞えるように、適切な開発メンバーをフィードバックしておきましょう。特に、コミュニケーションの質が悪い、優先度の判断や実用性が全く理解できない、など、チームメンバー全体のマナーベータンに関する改善では、きちんと話し合ってから双方の改善点を明確にして対応していく必要があります。真に良いPOであれば、常にWin-Winの姿勢を持っているはずです。適宜に調整していきましょう。 115	8.7 アジャイルの原則 #2 (2019/03/20 Version 1.0更新) 7.6 アジャイルの翼に注意する アジャイル開発だからこそ、逆いような問題が起きます。注意しなければいけません。 ■ 誰かの見えると勘違いが増える。しかし、誰かを見ればは最後は無し、最も重要な部分で作業を進めよう。 アジャイル開発では動くのが早に認識できます。すると、具体的に細かくコメントが多数、出たり来たりします。しかし、それらに対して対応するのはおの、コメント対応の発生を見逃しましょう。細かくコメント対応をしないことでプロジェクトの前進のペースが一気に遅くなります。いつでもでも細かくコメントへの対応は最後でいいのです。もっとも重要な部分に集中していいのがアジャイル開発です。システムの大規模な重要部分でできる限り早く一回通して動くものを作り上げ、プロジェクト全体の目標を早くつづけることが大切なのです。そして、重要な部分・時間的余裕の少ない部分の部分を抽出して、そこを手で行っていくのがアジャイル開発なのです。 ■ アジャイル開発の初期には、アーキテクチャを固めることが重要であって、プロダクトコードの品質を目的にしないケースもあります。その場合、そのおかしコードは後で修正することもあります。 プロダクトコードに価値を置きすぎると、そのおかしコードを利用することが主観になって、見逃しの悪い設計になる場合があります。最初の一連の機能が開け 116	7. 開発担当者への声 たプログラムコードは、後々、捨てて、良いプログラムコードに置き換えること普通に行われなければいけません。もちろん、100%のタフタフタは費用対効果が良くなりませんが、維持管理のし易さやエラーの修正のしやすさを兼ねていなければいけません。新しいチームメンバーが入った際のプロジェクト理解用の作業としての活用もあってもいいです。 ■ POにリテラシーがあるため、開発メンバーに主体性が芽生えやすい場合には、指導者も人間に委ねてみようケースもあります。 諸君開発では受託開発としたことの責任が強く問われるが、アジャイル開発ではプロフェッショナルとしての専門意識の高揚は発生するものの、最終責任者はPOです。技術力を活かした主体性を発揮してPOに貢献することが期待されています。しかし、常にPOの指示に即応し続ける必要は無いという判断が必要である。マナーベータンが高くない場合には、そのままだけに人間としてスキルが固定化されるリスクがあります。そういう開発メンバーはアジャイル開発に向いていないと思います。 117	8.7 アジャイルの原則 #2 (2019/03/20 Version 1.0更新) 7.7 ウォーターフォールの悪癖は矯正する ■ 過剰な品質は要です。【オーバーフィット問題】 アジャイル開発ではおおよそ2週間に対する効果を重視します。自分が満足したくらいでいい。優先度の低いところの対応に時間をかけてはいけません。POは、そんな時間があるんだから。もしも別のところをやって欲しいと思います。自分の足元だけを見え満足してはいけません。時間がゆるゆるには進めないと優先度をPOに確保することが必要です。 ■ 悪い情報の提供をしないのは鉄則です。 「悪い情報」は、もちろん問題が明確になるまで報告するのとは違いますが、判断や、問題など、報告するまでに対策を出せ、とされるなら、対策が決まるまでは問題の報告は遅らせるという判断は駄目です。POには悪い「情報」をできるだけ早く報告し、それを含んだ視野の中で良い判断をしてもらいましょう。 ■ 自らの失敗や遅延を、ごまかしたり曖昧にして報告してはいけません。 正直で、透明性の高い「組織文化」を作ることにはあなたも協力していかねばいけません。失敗や遅延の大きさに応じて、チームメンバーの作業の優先度を変化させるのもアジャイル開発です。ごまかしで発生した不信感とは、その後のコミュニケーションの質を悪化させます。 118
--	--	--	---

7. 開発担当者への声 ■ ドキュメント化を最初にするのは時間の無駄です。Face to Faceで議論し、ドキュメントは議論を促進・共有するものと考えよう。 方法を議論する際に、検討資料を作成して案を3つ並べて、各種の観点で評価して選定していく、そのスタイルをPOが望んでいてよいのか？ 望んでいないのか？ あるいは、あなたが「その方が議論は進めるのでいい、こうではないですか？」、「いや、ここはこうではないか？」という議論をホワイトボードの前で、ササカサ決める部分も多くなるのではないのでしょうか。重要な選択については、決定した方法と決定のポイントやキーマン化することが期待されますが、コミュニケーションのための大量のドキュメント作成は止めます。ウォーターフォール開発では受託者によって判断と議論は責任問題に陥るために重要でした。しかし、アジャイル開発では優先度のPOが責任を持ちます。判断や決定は1回ではなく、訂正可能なのです。 ■ ソースコードのレビュー単位は小さくしましょう。 ソースコードのレビューに出す際は、ソースコードをレビューする人の負担も軽減し読みやすさを考え、小さい単位にしましょう。自己満足で優先した巨大なソースコードをレビューに出す、相手の必要を無視して、方式のズレがあった場合でも大きな無駄作業となります。やむを得ず巨大なソースコードをレビューに出す場合には、F2Fで適切な解説を行い、レビュー進捗を改良しましょう。アジャイル開発ではソースコードもコミュニケーションの一部として大切なことになります。 119	8.7 アジャイルの原則 #2 (2019/03/20 Version 1.0更新) ■ 作主・主観・前提主観を優先してはいけません。 作主は、原則、アジャイル開発においては判断のための一つの情報にしか過ぎません。ウォーターフォールで慣れた作法・実装のやりかたであっても、その場で本当に重要なものを改めて考え直して欲しい。編成が大切であります。 ■ 絶対的な工程の区切りはあきません。バランスです。 ウォーターフォール開発では、原則として工程に拘りません。そのため工程完了の判定の実施などをする開発もあるでしょう。しかし、アジャイル開発では、その区切りは曖昧です。大枠での工程、自分自身が何が工程なのか、などは意識するために工程名は使いますが、決して工程「区切り」ではなく、工程名はコミュニケーションの用語だと思えます。 ■ 作業の順序は重要です。「上も並んでいる順にやります」は考えていないことかもしれません。 諸君開発では、結局 100%やりきることが求められます。その時は、どの順番で実施するのかわかりやすくなるのが重要でした。しかし、アジャイル開発では優先度の高さを、捨てる事が発生します。たとえば、モバイル開発で、暫定的に作っておいた部分や、設計が滞った部分などです。それらの部分に対してUT 100%を最初から目指すことは無意味になります。 120	7. 開発担当者への声 ■ 作ったソースコードは捨てても構いません。先行しすぎた単体試験は無駄になります。 「手戻りが発生します」、「既製のソースコードが無駄になります」は、ウォーターフォール開発では工程が増加するため大きな問題でしたが、アジャイル開発では優先するの次第で、捨てる事が発生します。たとえば、モバイル開発で、暫定的に作っておいた部分や、設計が滞った部分などです。それらの部分に対してUT 100%を最初から目指すことは無意味になります。 121	8.7 アジャイルの原則 #2 (2019/03/20 Version 1.0更新) ■ 作ったソースコードは捨てても構いません。先行しすぎた単体試験は無駄になります。 「手戻りが発生します」、「既製のソースコードが無駄になります」は、ウォーターフォール開発では工程が増加するため大きな問題でしたが、アジャイル開発では優先するの次第で、捨てる事が発生します。たとえば、モバイル開発で、暫定的に作っておいた部分や、設計が滞った部分などです。それらの部分に対してUT 100%を最初から目指すことは無意味になります。 122
--	--	---	--