

VO 3

スクラムの紹介。

ソフトウェア開発などの「知的創造」においては、知識を発見し創造する能力が根本的な制約条件となります。

一方、会社組織は、ヒエラルキーとインセンティブに立脚した、一つ前の世代向きの設計となっています。

スクラムとは、知的創造及び共同発明に最適化されたものです。

そのため、組織の現行の方針や習慣、構造とは相反することになります。これらは簡単に変えられるものではないので、多くの組織では、現行の慣習を「スクラム」と呼び変えるだけにとどまっています。

こういった偽のスクラムに対して、皆さんが異議を呈する一助として本レッスンを作成しました。

VO 4

スクラムトレーニングシリーズ・モジュール1、「スクラム概論」へようこそ。

ここではスクラムの全体像をご紹介し、以降のモジュールで各トピックの内容を更に深めていきます。

私はマイケル・ジェームス、略してエム・ジェイです。様々な組織に対して、スクラムやそれに関連するアジャイルの実践について支援を行っています。

VO 5

このモジュールには3つのパートがあります。

モジュールの最後に、認定試験のスコア向上に実績のあるクイズがあります。こちらは、認定トレーニングを受講する際の必要条件となる場合もあります。

6ページのイラスト付きスクラムリファレンスカードもありますので、ぜひダウンロードしてください。

VO 6

スクラムは、新規プロダクト開発などの複雑な業務を取り扱うためのフレームワークです。

これは、製造や建設といった業種に適していた従来型のアプローチを置き換える手法です。では、どうして従来のやり方を変える必要があるのでしょうか。

VO 7

20世紀、大半の仕事では、イノベーションよりも業務を成し遂げることが重要視されていました。

人々は与えられた役割のもと、定められた型を使い、定められたやり方で、急速に変化することのない、定められた計画を遂行していました。

VO 8

およそ仕事というものは、予想できるものでした。なぜなら、私たちは、「何」を。。。成し遂げようとし、それを「どうやって」。。。成し遂げようとしているのかを、現代よりも深く理解していたからです。

予測可能な作業や繰り返し作業を遂行するためのアイデアを、フレデリック・テイラーやエイチ・エル・ガントが提唱し、当時の実業家たちがそれを活用しました。

VO 9

今日では、予測可能な繰り返し作業の多くは、機械を用いてもっと迅速に行われています。

世の中の動きも、より速くなりました。

競争力を高く保つためには、速やかに検証と適応を行い、目まぐるしく変化する環境に対応していく必要があります。

タイムボックスとフィードバックループを伴う透明性のあるフレームワークを用いれば、不確実性の高い状況を乗りこなそうとすると、力強い手助けになります。

これからの時代は、学習する組織だけが生き残れるのです。

VO 10

スクラムとは、仕事に求められるものは何か、そしてその仕事を成し遂げるにはどうすればよいかを学ぶためのフレームワークです。

カオスを枠組みに入れて、不確実性を最大限に活用しようという試みです。

ソフトウェアプロダクトの新規開発で主に用いられてきた手法ですが、その他の複雑な作業においても役立ちます。

VO 11

スクラムでは、フィードバックループを用いて。。。
開発中のプロダクト及び。。。
その開発プロセスについての検証と適応を促します。

VO 12

スクラムに関連するアジャイルムーブメントについては、こちらをご参照ください。

私たちが重視するのは次のことです。
プロセスやツールよりも、個人および対話を。
包括的なドキュメントよりも、動くソフトウェアを。
契約交渉よりも、顧客との協働を。
計画に従うことよりも、変化への対応を。

左の事柄に価値が無いというわけではありません。

私たちは、右の事柄に、より価値を置くということです。

VO 13

大きなリリースを最終段階で一度に行うのではなく、プロダクト開発サイクルの初期から、実際に動くフィーチャーを見たくなるものです。

理想的に言えば、インクリメンタルな改善を無限に重ねることができるはずです。

スプリント毎に、私たちは顧客ニーズに対する学びを深め、少しずつ、場合によっては一気に、方向性を変えていきます。

プロダクト開発は単発の「プロジェクト」ではなく、リリースを重ねながら常に進行していくものです。

スクラムではないものを見てみましょう。1970年、ウインストン・ロイスは論文を発表し、伝統的なガントチャートという概念のソフトウェア開発への導入を、初めて試みました。

彼はこのような図を描きました。ご覧のように、いくつものフェーズに分かれています。

VO 15

1つのフェーズから次のフェーズへ、そしてまた次のフェーズへと繋がり、それぞれのフェーズで引き継ぎが生じます。

理論的には、分析作業は始めに全て完了させてしまい、次に設計に取りかかりこれも全て完了させ、そして次はコーディングに取りかかり全て完了させ、次にインテグレーションに移る、という調子で進みます。

これは物事がどのように進むかを示した素晴らしい理論でした。もしも我々が、初期段階に完璧な知識があり、進行中に全くミスを犯さなければ、その理論はうまく機能したことでしょう。

VO 16

あまり知られてはいませんが、ウィンストン・ロイス博士の論文にこんな一文があります。

「私はこの概念を信じているが、実践するとなるとリスクが高く、うまくいかないだろう」

VO 17

複雑な作業でこの理論がうまくいかないのは、初期段階に完璧な知識があるという前提が、ほぼあり得ないからです。

実際、始めたばかりの時点では、プロジェクトについての知識は少なく、時を経るにつれてだんだんと分かってくるものです。

残りのプロジェクト期間全体で、常に、今日が一番無知な日というわけです。

それなのに、計画駆動型アプローチでは、知識の少ない初期段階で最重要事項についての決断が求められます。

ウォーターフォール型プロジェクトでは、最終段階でようやく実際の作業が始まりますが、すでに過剰な時間がかかってしまっているうえに作業の質は低く、最終的に無秩序状態へと陥ってしまうこととなります。

VO 18

スクラムは、全てのフェーズをミキサーに放り込むようなものです。全ての材料が混ぜ合わさって全てのスプリントに分けられます。

VO 19

仕事を、特定の作業単位のフェーズに分割するのではなく、決まった期間のイテレーションに分割するのです。これをスプリントと呼びます。

どのスプリントも、分析、設計、実装、テスト、顧客ニーズの学習や将来の計画を組み合わせたものを含んでいます。

デプロイや出荷をより頻繁に行えるようになるでしょう。

VO 20

最初のスプリントが始まったときから、実際に動き、テスト済みで、潜在的には出荷可能なプロダクトの開発に取り組むのがスクラムチームです。たとえ小さくても開発は始めることができます。スクラムでは、これをインクリメントと呼びます。

そしてスプリントごとに、インクリメントを全員にデモンストレーションします。

的外れのプロダクトに触れてはじめて、顧客が自分の真のニーズを特定できる、ということがよく起こります。

短期間のイテレーションをこなし、顧客からより多くのフィードバックを得ることで、的を得たプロダクトに到達する可能性が高まります。

スプリント毎にプロダクトオーナーが実際に出荷する必要はありませんが、チームは、それが可能な状態を実現する役割を担っています。

VO 21

出荷可能なプロダクトを二週間毎に作り出すためには、クロスファンクショナルで自己組織化されたチーム内に、テスト、設計、業務要件の調整、コーディングといったスキルを有する人材を全て備えておく必要があります。

これで、最後まで待たされることなく最初のスプリントからテストを実行できます。

事前に設計を全て行ってしまうのではなく、少しの設計を毎スプリントで行い、継続的な設計を実現していきます。

技術的負債が生じないよう、継続的な再設計とリファクタリングを小規模に行っていくことになるでしょう。一つ一つのスプリントが、業務の全ての側面を組み合わせたものなのです。

スクラムチームは、フェーズ毎の引き継ぎにより作業するのではなく、相互に協働します。

VO 22

スクラムは、役割、イベント、ルール及びアーチファクトの構造を提供します。

では、その全体像を見ていきましょう。

VO 23

スクラムで定義される役割は、プロダクトオーナー、スクラム開発チーム、スクラムマスターの3つだけです。まず、プロダクトオーナーについて説明します。

VO 24

プロダクトオーナーとは、投資収益率(アール・オー・アイ)についての責任を持つただ1人の人間です。

主に、プロダクトバックログの優先順位を決定することを通じて、その影響力を行使します。

また、要求に関する問題が生じた場合、プロダクトオーナーが最終的な決定権を持ちます。

これは、プロダクトオーナーが要求事項の詳細を事前に全て与えるとか、各スプリントの始めに詳細を伝えるとかいう意味ではありません。

プロダクトオーナーが、こういった事柄の最終決定権をもつという意味です。

プロダクトオーナーには、プロダクト開発の裏にあるビジョンがなければなりません。

不確実性の高い状況では、詳細な道路地図はあまり役に立ちませんが、ビジョンを持つことは重要です。

VO 25

他の誰かがチームに何かを依頼したいときは、プロダクトオーナーを通す必要があります。

プロダクトオーナーは唯一、優先事項を決定する人間です。そして、どう行うかよりも何を行うかに着目して、ビジネス上の意思決定を行います。

VO 26

大規模スクラム(レス)では開発チームが複数になりますが、プロダクトオーナーは一人だけです。

VO 27

次に、スクラム開発チームです。

スクラム開発チームは、自己組織化したクロスファンクショナルな集まりであり、スプリントごとに出荷可能なプロダクトを開発する責任を負っています。

はじめは難しいものですが、現在、多くのチームがこの手法を学んでいます。あなたのチームも取り残されないよう習得していきましょう。

組織は「チーム」という言葉を軽く扱いがちです。

これまでの経験のなかで、全員が互いに信頼できる本物のチームにいたときを思い出してください。

そういうチームを、皆さんに作っていただきたいのです。

条件を整えると、チームワークは向上します。フルタイムのチームに適切な環境を提供し、効果的にレトロスペクティブを行っていくのです。

学び続ける組織の要素である、学び続けるチームへと成長できるかもしれません。

外部からチームにヒエラルキーや役職を押しつけられることはありません。リーダーシップが自然に生まれ、コントロールが人から人へと自然に流れていくに委ねるのです。

VO 28

スクラム開発チームは、9人を超えない小さなグループです。人間にとって、家族規模のグループは、何百万年も前から馴染みのある存在です。

大規模なグループが効果的に自己組織化するためには、小グループへの分割が必要です。

大規模スクラムでは、フルスタックフィーチャーチームが一つのプロダクトバックログから業務を選び、スプリント毎に成果を統合します。

VO 29

チームの協働が最も自然な形で生まれるのは、チームルームの中です。

VO 30

あなたの組織がグローバル展開していれば、連携や協調の無さには、既に苦しめられているのではないのでしょうか。スクラムでは、こういった問題は直ちに表面化します。

リモート勤務していたメンバーを一つの部屋に集めて、始めの1から2スプリントはずっと一緒に、その後は数ヶ月に一度この状態で作業することで、大きな進歩が生じるかもしれません。

情報共有ツールを使っても、それだけで組織変革が生じることはないでしょう。状況が悪くなることすらよくあります。

VO 31

スクラムマスターは、スクラムで一番誤解されている役割です。

例えば、私があなただのスクラムマスターだとしても、あなたはスクラム奴隷になるわけではありません。

むしろ逆です。スクラムマスターは、チームに対してマネジメント権限を持ちません。

つまりプロジェクトマネジャーやラインマネジャーは、定義上、スクラムマスターではないこととなります。

スクラムでは、あえてプロジェクトマネジャーの役割を設けていません。

VO 32

プロジェクトマネジメントの責任はプロダクトオーナーとチームが分かれて担い、スクラムマスターがファシリテーター役を担います。

VO 33

スクラムマスターは、中断や割り込みからチームを守り、チームが自然に自己組織化できる環境に導きます。

チームに影響する障害を取り除き、プロセスをファシリテートし、スクラムの実践方法を教え、質の良いエンジニアリングプラクティスやタイムボックスを推進し、ビジビリティを提供します。そして、マネジメント権力を行使せずに以上の全てを成し遂げるのです。

最も強い影響力は「権力」ではないことがわかります。

VO 34

やがては、スクラムマスターはチームとプロダクトオーナーにそれほど注意を払わなくても良くなるでしょう。

「チームにスクラムを行わせる」ことをしないのが、優秀なスクラムマスターです。直接管理されるのではなく、既に学び協働する意欲を備えているのが優秀な開発者です。

組織のポリシーや構造に対する影響力を行使することによってチームがスクラムを行えるようになるのが、優秀なスクラムマスターです。

VO 35

スクラムマスターの役割は把握しづらいものです。更なる情報は「スクラムマスターチェックリスト」をご参照ください。スクラムマスターが調整する事項の実例リストです。

VO 37

スクラムでは3つのアーチファクトを定義しています。プロダクトバックログ、スプリントバックログ、インクリメントです。

VO 38

プロダクトバックログとは、顧客中心のフィーチャーを強制的に順位付けした一次元リストで、プロダクトオーナーがその順位を決定します。

行う可能性のあることは全てこのリストに含まれています。バックログになれば、存在しないとお考えください。

プロダクトバックログには誰もがアイテムを追加できます。それらに対してプロダクトオーナーは優先順位をつける責任を持ち、スクラムマスターはそれを可視化させる責任を持ちます。

しっかり作られたプロダクトバックログにはタスクが含まれません。あるのは、ユーザーストーリーやユースケースなどの形で表現された、しっかり作られたプロダクトバックログアイテム(ピー・ビー・アイ)だけです。

VO 39

スクラムにおいて、強制的に優先順位付けされたということは、最重要項目はただ一つであることを意味します。組織がこれを実践するのは、多くの場合躍進をもたらします。

VO 40

複数チームで一つのプロダクトを開発する場合にも、プロダクトバックログはただ一つだけです。

VO 41

スプリントバックログとは、現在のスプリントゴールを達成するために計画されたやるべき事項です。

スプリントバックログには期限があります。スプリントバックログは、スプリントのために抜き出されたプロダクトバックログのアイテム及びその実行計画を含みます。例えばスプリントタスクのリスト等です。

VO 42

複数チームの場合は、たとえ同一のプロダクトを開発している場合であっても、スプリントバックログは別々です。

VO 43

では、次にスクラムにおけるイベントの概要を説明しましょう。

VO 44

スクラムが定義づけをする5つのイベントのほかに、皆さんが有用だと考える6番目のイベントがあります。

スクラムが定義づける5つのイベントとは、スプリント、スプリントプランニング、デイリースクラム、スプリントレビュー、そしてスプリントレトロスペクティブです。

6番目のイベントには正式名称はありませんので、バックログリファインメントと呼ぶことにしましょう。

VO 45

2週間のスプリントにおいてこれらのイベントがどのように行われるか一例をご紹介します。私は個人的には、週末に働かずに済むようスプリントを金曜日に終えるのが好きです。

VO 46

スプリントプランニングでは、開発チームが当該スプリントで実行すべきアイテムを選びます。

VO 47

開発チームは、選ばれたアイテムをスプリントバックログに移し、どのように実行するかを計画するとともに、作業量が適切かを判断します。ここでは、一つのスプリントを計画します。

VO 48

大規模スクラムにおいては、複数の開発チームが、同一のプロダクトバックログから各チームのスプリントバックログにアイテムを移します。

コーディネーターの仲介なしで、複数チームが集まった一つのチームとして協働するよう計画を立てます。

VO 49

スプリントの実行期間中、開発チームは一日に一度、15分間のデイリースクラムで集まります。

一緒に仕事をしているのですから、もちろんずっと顔を合わせているのですが、この公式イベントでは、立ったまま、お互いに助け合うための方法を模索します。

チームは相互に対話します。スクラムマスター、プロダクトオーナーや上司などに対して報告するのではなく、チームと話し合うのです。

例えば、私がチームメンバー6人に話をするとします。しっかりと相手の目を見て、自分が昨日やったこと、今日やる予定のこと、作業が進まない理由、自分にとっての阻害要因について話をします。そして他のチームメンバーに会話のボールを渡し、今度は彼らがチームに対して話をします。

VO 50

大規模スクラムにおいては、各開発チームはデイリースクラムを行うほか、他の開発チームと協働するための様々なコラボレーションのパターンを実行します。

VO 51

調整の責任はすべて、自己組織化した開発チームにあります。「リリース・トレイン・エンジニア」や、マネジメント役が割り当てた調整役などは存在しません。

スクラムマスターは自らマネジメントを行えるチームの能力を引き出します。ですから、良いスクラムマスターであれば、決して実際の調整業務を行うことはありません。

VO 52

スクラムマスターは、チームがトランク上の継続的インテグレーション、テスト駆動開発、その他の現代的な開発実践技術を学べるように支援を行います。

VO 53

スプリントレビューでは、プロダクトを検査し適応することを目的とします。ここでは、スプリントの成果を検査し、今後の計画、つまりプロダクトバックログを、検査から学んだことに適応させます。

チームは、潜在的に出荷可能なプロダクトの増分のデモンストレーションを、興味を抱いてくれている相手、理想的には顧客やエンドユーザーに対して行います。

ステークホルダーからフィードバックを得ることで、今後の開発の検証と適応を行うことができます。

「約束通りのものを作ってくれたけど、実際に見てみると欲しいものとは違うことに気づいたよ。間違ったプロダクトを見るまでは気づくはずもなかった。」というフィードバックもよくあります。

どうやら人は、自分が本当に欲しいものを理解するには、実際に動くソフトウェアを試す必要があるようです。

大規模スクラムにおいては、複数チームにより統合された一つのプロダクトについて、スプリントレビューを一回行います。

VO 54

定期的なスプリントレビューでは、プロダクト及び明らかになった要求事項についてフィードバックを提供します。

VO 55

各スプリントの最後に、チームでスプリントレトロスペクティブを行い、自らのプロセスについて検証と適応を行います。

直近のイテレーションでの協働の仕方について検証及び適応を行います。

主に、うまくいったこと、さらに改善できること、学んだこと、困惑していることについて話します。話しながら、お互いにフィードバックを行います。

具体的な実践方法については、モジュール6で説明しています。最終的にチームが自分たちの開発プロセスにオーナーシップを持つようになること。これが、全ての鍵となります。

VO 56

定期的なスプリントレトロスペクティブでは、チームがプロダクト開発に用いているプロセスや手法についてのフィードバックループがもたらされます。

思い出していただきたいのは、スクラムとは、プロダクト及びそれを開発する私たちが用いるプロセスについて学習することを目的としたフレームワークであるということです。

VO 57

大規模スクラムにおいても、チームはそれぞれのレトロスペクティブを実践します。

その後、包括的なレトロスペクティブをチーム相互、およびプロダクトオーナー、スクラムマスター、もしあればマネジャーとの間で行います。

包括的なレトロスペクティブでは、複数のチームに影響を及ぼす、システム的かつ組織的課題を探します。

VO 58

バックログリファインメントにおいては、チームとプロダクトオーナーが集まって、プロダクトバックログ上で次に来るアイテム、つまり次の数回のスプリントで行うであろうアイテムをいくつか取り上げて、少し先のことを検討します。

そして、それらアイテムを明確化し、一つのスプリント内で収まる様子をイメージできるように、プロダクトバックログ上の巨大なアイテム、またはエピックを、プロダクトバックログ上の小さなアイテム、例えばユーザーストーリーへと分解し、優先順位について話し合います。

以上の作業はスプリントプランニングでも行えますが、経験を積む中で、別の日に別のイベントで行う方が好ましいと気づきました。

VO 59

スクラムの概要は以上です。

スクラムについてももう少し詳しく知りたい方は、スクラムリファレンスカード及びスクラムマスターチェックリストをダウンロードしてください。

更に深く学びたいという方は、当社のスクラム研修にご参加いただくか、他のモジュールで更に踏み込んだ解説をご覧ください。

スクラムに関するコーチングが必要な場合はどうぞ当社へお電話ください。また、本トレーニングモジュール動画に対するご意見ご感想もお待ちしております。