
アジャイル実務ガイド

Library of Congress Cataloging-in-Publication データが適用されています。

ISBN: 978-1-62825-423-5

発行者：

Project Management Institute, Inc.
14 Campus Boulevard
Newtown Square, Pennsylvania 19073-3299 USA
電話: +1 610-356-4600
ファックス: +1 610-356-4647
電子メール: customercare@pmi.org
インターネット: www.PMI.org

© Copyright 2017 Project Management Institute, Inc. All Rights reserved.

Project Management Institute, Inc. の出版内容は、多くの国々で認知されている米国知的財産法に保護された著作権を所有しています。PMIの出版内容を再発行したり、複製したりする場合は、弊社の許可を得る必要があります。詳細については <http://www.pmi.org/permissions> でご確認ください。

注文取引や価格については、Independent Publishers Group: へお問い合わせください。

独立出版社グループ
注文受付部門
814 North Franklin Street
Chicago, IL 60610 USA
電話: +1 800-888-4741
ファックス: +1 312- 337-5985
電子メール: orders@ipgbook.com (注文についてのみ)

その他のご質問は、PMI Book Service Centerまでお問い合わせください。

PMI Book Service Center
P.O. Box 932683, Atlanta, GA 31193-2683 USA
電話: 1-866-276-4764 (米国とカナダ) あるいは +1-770-280-4129 (米国、カナダ以外)
ファックス: +1-770-280-4113
電子メール: info@bookorders.pmi.org

Printed in the United States of America. 本書のいかなる部分であっても、電子的、手動、複写、記録、またはデータストレージ、データ取得システムなど、媒体や手段を問わず、発行者からの書面による許可なく複製または転送することは禁じられています。

この図書に使われている用紙は、米国情報規格協会の永久保存用紙質規格 (Z39.48-1984) に準拠しています。

PMI、PMIのロゴ、PMBOK、OPM3、PMP、CAPM、PgMP、PfMP、PMI-RMP、PMI-SP、PMI-ACP、PMI-PBA、PROJECT MANAGEMENT JOURNAL、PM NETWORK、PMI TODAY、PULSE OF THE PROFESSION、MAKING PROJECT MANAGEMENT INDISPENSABLE FOR BUSINESS RESULTS. のスローガンは、すべてProject Management Institute, Inc. の登録商標です。PMIの商標をすべて網羅した一覧についてはPMI 法務部までお問い合わせください。本書に見られるその他の商標、サービス商標、商品名、トレードドレス (製品に顕著なデザイン)、製品名、ロゴは、すべて該当する所有者に属するものです。ここに明示されていない権利もすべて保護されています。

SAFe は Scaled Agile, Inc. の登録商標です。

アジャイル・アライアンスおよびアジャイル・アライアンスのロゴは、Agile Alliance の商標です。

この実務ガイドは、Agile Alliance® との共同出資によるもので、Agile Alliance® のメンバーとの協同により作成されました。Agile Alliance® では、いかなるアジャイル方法論または認証も承認することはありません。

10 9 8 7 6 5 4 3 2 1

注意

プロジェクトマネジメント協会（PMI）の標準およびガイドラインに関する出版物は、本書も含め、ボランティアのコンセンサスに基づき、標準作成プロセスを通して作成されたものである。このプロセスは、ボランティアを募り、図書の対象となるトピックに関心をもつ人々の見解を求めるプロセスである。PMI はこのプロセスを管理し、コンセンサスを形成する上で、その公正さを保証するための規則を設定しているが、本書はPMIが執筆したものではなく、PMI は本標準およびガイドラインに関する出版物に含まれる情報の正確さ、完全さ、判断の健全性等に関して独自にテスト、評価、検証を行ったわけではない。

PMI は、その出版物、本書の適用、または本書への依拠の結果として直接間接に発生する個人への傷害、財産その他のあらゆる種類の損害について、それが特殊的、間接的、結果的、賠償的の如何に拘わらず、その責任を否認する。PMI は、本書に記載されているいかなる情報についても、それが明示的か暗示的かに拘わらず、その正確性または完全性について責任を認めず、いかなる保証も行わない。また、PMI は本書の情報が読者の特定の目的またはニーズを満たすことについて、その責任を認めず、保証もしない。また、本書の標準またはガイドによって、個々の製造者または納入者のプロダクトまたはサービスのパフォーマンスに関する保証の義務を負うものではない。

本書を出版および利用に供することにより、PMI はいかなる個人または法人のためあるいはこれに代わって、プロフェッショナルとしてのサービスやその他のサービスを行う義務を負うものではなく、またPMI は、いかなる個人または法人が第三者に対して負う義務の実行を引き受けるものでもない。本書を使用する者は自身独自の判断に依拠すべきであり、また、場合に応じて、特定の状況における合理的な判断を行うために有能なプロフェッショナルの助言を求めるべきである。本書に記載されるトピックに関する情報およびその他の標準などは、その他の情報源からも得ることができるので、ユーザーは本書に記載されていないその他の考察または情報をそれらの情報源に求めることができる。

PMI は、本書の内容に従うことを監視または強制する権限をもっておらず、またその義務も負わない。PMIは、安全性や健全性のために、プロダクト、デザイン、または導入に関して、認証、テストまたは検査を行わない。健全性や安全性に関連する情報への遵守についていかなる認証または記述が本書にあったとしても、PMI に帰属するものではなく、その責任はすべてその記述を認証した者または作成者に帰属する。

序文

Project Management Institute (PMI)と Agile Alliance® は、それぞれのコミュニティにおけるアジャイル・アプローチをより深く理解するためにこの実務ガイドを定めた。この実務ガイドのビジョンは、プロジェクト・チームに、より良い結果を出すためのツール、状況判断のためのガイドライン、および利用可能なアジャイルの技法と手法への理解を身に付けさせることである。

ソフトウェア開発を越えた様々な業界で、プロジェクト・チームはアジャイル・アプローチを使用している。両組織は、共通の言語、オープンな心、そしてプロダクトおよび成果物が市場投入される方法に柔軟に対応する意欲の必要性が拡大することを認識している。さらに、両組織は引渡しを成功させるためには複数の方法があることを理解している。広範囲なツール、技法、およびフレームワークがあり、チームは期待される成果を達成するためにそのプロジェクトや組織文化に適合する手法と実務慣行に選択肢がある。

『アジャイル実務ガイド』のコア委員会メンバーは様々な背景の出身者であり、様々な手法を使用している。委員会メンバーにはコンサルタントや、組織内部で仕事をする人もいる。全員が何年にもわたってアジャイル方式に携わってきた。

目次

1. はじめに	1
2. アジャイル入門	7
2.1 定義可能な作業と不確実性の高い作業	7
2.2 アジャイル宣言とマインドセット	8
2.3 リーンとカンバン方式	12
2.4 不確実性、リスクおよびライフサイクルの選択	13
3. ライフサイクルの選択	17
3.1 プロジェクト・ライフサイクルの特性	18
3.1.1 予測型ライフサイクルの特性	20
3.1.2 反復型ライフサイクルの特性	21
3.1.3 漸進型ライフサイクルの特性	22
3.1.4 アジャイル型ライフサイクルの特性	24
3.1.5 アジャイル適合性フィルター	25
3.1.6 ハイブリッド型ライフサイクルの特性	26
3.1.7 アジャイルと予測型を結合したアプローチ	27
3.1.8 いくつかのアジャイル構成要素を含む予測型アプローチ	28
3.1.9 いくつかの予測的構成要素を含むアジャイル・アプローチ	28
3.1.10 目的適合としてのハイブリッド型ライフサイクル	29
3.1.11 移行戦略としてのハイブリッド型ライフサイクル	30
3.2 アジャイル・アプローチの混合	31
3.3 テーラリングに影響するプロジェクト要因	32

4. アジャイルの実装：アジャイル環境の作成	33
4.1 アジャイルのマインドセットで開始.....	33
4.2 チームに力を与えるサーバント・リーダーシップ	33
4.2.1 サーバント・リーダーの責任.....	34
4.2.2 アジャイル環境におけるプロジェクト・マネジャーの役割.....	37
4.2.3 プロジェクト・マネジャーはサーバント・ リーダーシップを使用する	38
4.3 チーム構成.....	38
4.3.1 アジャイル・チーム.....	39
4.3.2 アジャイルにおける役割	40
4.3.3 汎化スペシャリスト.....	42
4.3.4 チームの構造	43
4.3.5 専任チーム・メンバー.....	44
4.3.6 チームの作業場所.....	46
4.3.7 組織内サイロ化の克服.....	47
5. アジャイルの実装: アジャイル環境での引渡し	49
5.1 プロジェクト憲章とチーム憲章	49
5.2 一般的なアジャイル実務慣行	50
5.2.1 レトロスペクティブ	50
5.2.2 バックログの準備.....	52
5.2.3 バックログの洗練.....	52
5.2.4 デイリー・スタンドアップ	53
5.2.5 デモンストレーションとレビュー.....	55
5.2.6 イテレーション・ベースのアジャイルの計画.....	55
5.2.7 チームの価値創出を助ける実務慣行の実施.....	56
5.2.8 イテレーションと増分が動くプロダクトの引渡しに役立つ方法.....	57
5.3 アジャイル・プロジェクトの課題のトラブルシューティング	57
5.4 アジャイル・プロジェクトにおける測定	60
5.4.1 アジャイル・チームが結果を測定する	61

6. プロジェクトのアジリティに関する組織的な考慮事項.....	71
6.1 組織のチェンジマネジメント	71
6.1.1 チェンジマネジメントの駆動要因.....	73
6.1.2 変革の準備	73
6.2 組織文化.....	75
6.2.1 失敗を恐れない環境づくり	75
6.2.2 文化の評価	75
6.3 調達と契約.....	77
6.4 実務慣行.....	79
6.5 複数チームの調整と依存関係（スケーリング）	80
6.5.1 フレームワーク	80
6.5.2 考慮事項.....	80
6.6 アジャイルとプロジェクトマネジメント・オフィス(PMO).....	81
6.6.1 アジャイルPMOは価値駆動	81
6.6.2 アジャイルPMOは招集指向	81
6.6.3 アジャイルPMOは学際的	82
6.7 組織構造.....	83
6.8 組織の発展.....	84
7. 行動の喚起.....	87
付録A1	
『PMBOK® ガイド』 マッピング	89
付録A2	
アジャイル宣言とのマッピング	97
付録A3	
アジャイルおよびリーン・フレームワークの概要.....	99
付属文書X1	
コントリビューターとレビューア	115

付属文書X2	
テラリングに影響を与える属性.....	119
付属文書X3	
アジャイル適合性フィルター・ツール.....	125
参照文献.....	139
参考文献一覧.....	141
用語集.....	149

表と図の一覧

図2-1.	アジャイル宣言 4つの価値.....	8
図2-2.	アジャイル宣言の背後にある12の原則.....	9
図2-3.	アジャイル宣言の価値、原則および共通実務慣行の関係.....	10
図2-4.	アジャイルは多数のアプローチを総称する用語.....	11
図2-5.	ステーシー複雑性モデルに触発された不確実性と 複雑性のモデル.....	14
図3-1.	ライフサイクルの連続性.....	19
図3-2.	予測型ライフサイクル.....	21
図3-3.	反復型ライフサイクル.....	21
図3-4.	増分がさまざまな大きさのライフサイクル.....	22
図3-5.	反復ベースおよびフロー・ ベースのアジャイル型ライフサイクル.....	24
図3-6.	アジャイル型開発と後続する予測型展開.....	27
図3-7.	アジャイルと予測型を同時に結合して使用するアプローチ.....	27
図3-8.	いくつかのアジャイル構成要素を含む予測型アプローチ.....	28
図3-9.	いくつかの予測的構成要素を含むアジャイル・アプローチ.....	28
図5-1.	残ストーリー・ポイントを示すバーンダウン・チャート.....	62
図5-2.	完了ストーリー・ポイントを示すバーンアップ・チャート.....	63
図5-3.	カンバンの例.....	65
図5-4.	フィーチャー図.....	67
図5-5.	プロダクト・バックログ・バーンアップ・チャート.....	68

図5-6.	アジャイル・コンテキストのアード・バリュー	69
図5-7.	完成したフィーチャーの累積フロー図.....	70
図6-1.	チェンジマネジメントとアジャイル・アプローチの関係	72
図6-2.	組織文化の評価例	76
図6-3.	初期にランク付けされた変更のバックログ	85
図6-4.	バックログとカンバンを使用した変更作業の体系化と追跡	86
図 A3-1.	適用範囲と詳細度でプロットしたアジャイル・アプローチ	100
図 A3-2.	仕掛り中の作業数制限を示すカンバンと、 作業の流れを最適化するプル・システム.....	105
図 A3-3.	クリスタル・ファミリー	106
図 A3-4.	フィーチャー駆動開発プロジェクトのライフサイクル.....	109
図 A3-5.	制約駆動型アジリティへのDSDMアプローチ	110
図 A3-6.	SoSチームに参加するスクラム・チームの代表者	112
図 X3-1.	アジャイル・アプローチの適合性モデル.....	127
図 X3-2.	アプローチへの賛同の評価.....	129
図 X3-3.	チームへの信頼の評価	130
図 X3-4.	チームの意思決定力の評価.....	130
図 X3-5.	チームのサイズの評価	131
図 X3-6.	経験レベルの評価	131
図 X3-7.	顧客やビジネス担当者へのアクセスの評価	132
図 X3-8.	変更の可能性の評価	132
図 X3-9.	プロダクトまたはサービスの重要性の評価	133
図 X3-10.	漸進型デリバリーの評価.....	133
図 X3-11.	適合性評価レーダー図.....	134
図 X3-12.	薬局プロジェクト	135
図 X3-13.	軍用メッセージング・システムの例.....	137

表1-1.	スコープ内の項目とスコープ外の項目	4
表3-1.	ライフサイクルの4つのカテゴリーの特性	18
表3-2.	適合性を改善するテーラリング・オプション	32
表4-1.	成功するアジャイル・チームの属性	40
表4-2.	アジャイル・チームにおける役割	41
表5-1.	アジャイルのペインポイントとトラブルシューティングの可能性	58
表A1-1.	プロジェクトマネジメント・プロセス群と知識エリアのマッピング	90
表A1-2.	『PMBOK®ガイド』知識エリアに定義されたアジャイルの適用	91
表A2-1.	『アジャイル実務ガイド』 で取り上げられているアジャイル宣言の価値	97
表A2-2.	アジャイル宣言の背後にある原則の実務ガイドとのマッピング	98
表A3-1.	スクラム・イベントと生成物	101
表A3-2.	エクストリーム・プログラミングの実務慣行	102
表A3-3.	カンバン方式の原則と特性の定義	104
表A3-4.	クリスタルのコア・バリューと共通プロパティ	107
表A3-5.	アジャイル統一プロセスの主要要素	111
表A3-6.	LeSSとスクラムの比較	113
表X2-1.	テーラリング・ガイドライン	121

1

はじめに

Welcome to the Agile Practice Guide! (『アジャイル実務ガイド』によろこそ!)このガイドは、Project Management Institute (PMI®) とAgile Alliance®がコラボレーションして開発された。この実務ガイドを開発したコア執筆チームのメンバーには両組織からのボランティアが含まれている。ボランティアは、多様な背景や信念、文化を有する現役の広範囲な実務者およびリーダーの当該分野専門知識を活用する。

この実務ガイドは、プロジェクトの計画および実施にアジャイル・アプローチを適応するプロジェクト・リーダーやチーム・メンバーに向けて、実践的なガイダンスを提供する。コア執筆チームは、予測型手法の使用への忠実な支持があることを認識しているが、アジャイルのマインドセット、価値および原則へとシフトすることへの情熱があることも認識している。一方で、この実務ガイドは、プロジェクト・アジリティへの実践的な手法を対象にしている。この実務ガイドは、予測型手法からアジャイル・アプローチへの道筋を提供している。事実、このふたつは異なるものとして取り扱われるが、両方で発生する計画法など類似の活動がある。

コア執筆チームは、この実務ガイドの初版の開発を調整しマネジメントするために、アジャイルのマインドセットを使用した。技術と文化が変化するのに伴い、現在の手法は実務ガイドの将来の更新や改良として反映されるだろう。

コア・チームは、この実務ガイドには、典型的なPMI標準の文書スタイルではなく、形式張らないリラックスしたスタイルを採用した。ガイドには、キーポイントや概念をよりよく説明するための、ヒント、サイドバー、およびケーススタディなどの新しい要素が組み込まれている。チームは、これらの変更によってこの実務ガイドをより読みやすくユーザーフレンドリーにすることを意図している。

アジャイルが非ソフトウェア開発にも拡大しているため、この実務ガイドはコンピューター・ソフトウェア開発業界以外でもアジャイルが利用できるようにしている。製造業、教育、医療、その他の産業は、程度の違いはあるがアジャイル型になってきており、ソフトウェア以外の利用も本実務ガイドのスコープ内にある。

アジャイル型学習

教育業界は、ソフトウェア開発以外でアジャイル実務慣行を拡大するための最も重要で肥沃な領域である。世界中の中学校、高校、大学の教員たちは、学習の文化を創成するためにアジャイルを使用し始めている。アジャイル技法は、競合する事項の優先順位を決めることに焦点を合わせるために使用される。想像力を活用する対面での対話、意味のある学習、自律型チーム、そして漸進型学習や反復型学習はすべて、教室でもマインドセットを変更できるアジャイルの原則であり、教育目標を促進するものである (Briggs, 2014)。*

*Briggs, Sara. 「Agile Based Learning: What Is It and How Can It Change Education?」 (『アジャイル型学習とは何か、教育をどのように変化させるか』) *Opencolleges.edu.au* 2014年2月22日に <http://www.opencolleges.edu.au/informed/features/agile-based-learning-what-is-it-and-how-can-it-change-education/>.より出典

なぜ『アジャイル実務ガイド』か、なぜ今なのか？プロジェクト・チームは少なくとも過去数十年間にわたり様々な形態のアジャイルの技法と手法を使用してきた。アジャイル宣言^[1]は、アジャイルが相当な勢いで使用されるのに伴って、アジャイルの最終的な価値と原則を示した (第2.1節参照)。今日、プロジェクト・リーダーやチームは、技術の指数関数的進歩や、より即座な価値の創出を要求する顧客によって、自分たちが混乱した環境にいることを認識している。アジャイルの技法と手法は、破壊的技術を効果的にマネジメントする。さらに、アジャイルの第一原則は顧客満足度を最優先課題とすることであり、顧客を喜ばせるプロダクトやサービスの提供における鍵となる (第2.1節参照)。迅速かつ率直な顧客のフィードバック・ループは、ソーシャル・メディアの普及に伴って容易に利用可能である。したがって、競争力と妥当性を保つためには、組織はもはや内部に焦点を当てるのではなく、むしろ顧客の体験へと、外部へ焦点を当てる必要がある。

¹括弧内の数字は、本実務ガイド巻末に記載されている参考文献リストの数字を示している。

破壊的技術は、参入障壁を減少させることによって競争領域を急速に変化させている。より成熟した組織は高度に複雑化しており、潜在的に革新に時間がかかり、顧客に新しいソリューションを提供することにおいて遅れを取っている。これらの組織は、顧客のニーズに適合するプロダクトを迅速に生産することができるより小型の組織やスタートアップと自らが競合していることを知る。この変化のスピードは、競争力を維持し、既存の市場シェアを保持するために、大型組織にアジャイルのマインドセットを採用するよう推進し続ける。

『アジャイル実務ガイド』はプロジェクトに焦点を置き、プロジェクト・ライフサイクルの選択、アジャイルの実装、およびアジャイル・プロジェクトへの組織の考慮事項を取り扱う。組織のチェンジマネジメント (OCM)は、実務慣行の実装または変革にとって不可欠ではあるが、OCMは内部規約であるため、本実務ガイドのスコープ外である。OCMのガイダンスについては、『組織のチェンジマネジメント: 実務ガイド』[2]を参照。

本実務ガイドのスコープ内およびスコープ外にある項目は、表1-1に記載されている。

破壊的技術

破壊的技術は、クラウド・コンピューティングへの移行により特に可能になっている。世界中の企業は、従来型の市場へ参入するために、コンピュータ資源へ迅速かつ安価にアクセスするモデルを活用している。クラウド・コンピューティングでは必要とされる前払額は減少するが、使用期間または使用量に対して支払う方式に基いて、サブスクリプション・サービスを介して超過分を支払う。更新されたアプリケーション、インフラストラクチャー、およびプラットフォームは、反復的かつ漸進的方法でクラウドにリリースされ、技術の向上と進化する顧客要求に遅れないようにする。

表1-1.スコープ内の項目とスコープ外の項目

スコープ内	スコープ外
プロジェクト・レベルまたはチーム・レベルでのアジャイル手法の実装	組織全体を通じたアジャイルの実装、またはアジャイル・プログラムの策定
業界調査に掲載されているような最も一般的なアジャイル手法の範囲	ニッチな手法、企業固有の方法、または不完全なライフサイクル技法の範囲
アジャイル手法や実務慣行を選択する際に考慮すべき適合性要因	特定の手法と実務慣行の推薦や推奨
アジャイルの、『PMBOK®ガイド』プロセスと知識エリアへのマッピング	『PMBOK®ガイド』プロセスと知識エリアの変更または修正
ソフトウェア開発以外へのアジャイルの使用に関する議論	アジャイル手法におけるソフトウェア業界の影響の排除(ソフトウェア以外の多くの業界でアジャイルの使用が増えているが、本実践ガイドにはソフトウェアについての記述が含まれていることに注意)
プロジェクトや組織にアジャイルを導入する際のガイダンス、技法および手法	プロジェクトや組織でアジャイルを実装する方法についての具体的なステップ・バイ・ステップの手順
一般に受け入れられている用語の定義	新しい用語や定義

本実務ガイドは、急速な技術革新や複雑さに対処してチームの改善に専念してはいるが、予測型手法とアジャイル・アプローチとの間の厄介な中間的立場にあると自らを認識しているプロジェクト・チームを対象としている。本実務ガイドは、顧客の期待とニーズに応える事業価値を提示して、プロジェクトを成功させるために有用なガイダンスを提供する。

本実務ガイドは、次のように構成されている。

第2章 アジャイル入門 — 本章にはアジャイル宣言のマインドセット、価値、および原則が含まれている。それには、定義可能な作業と不確実性の高い作業の概念、およびリーン、カンバン方式およびアジャイル・アプローチの間の相関関係も含まれている。

第3章 ライフサイクルの選択 — 本章では、本実務ガイドで取り上げられている様々なライフサイクルを紹介する。本章では、適合性フィルター、テーラリングのガイドライン、および手法の一般的な組み合わせも取り扱う。

第4章 アジャイルの実装:アジャイル環境の作成 — 本章では、サーバント・リーダーシップやチームの構成などアジャイル環境を作成する際に考慮すべき不可欠な要素を議論する。

第5章 アジャイルの実装:アジャイル環境での提供 — 本章には、チームを編成する方法についての情報、およびチームが定期的に価値を創出するために使用することができる共通実務慣行についての情報が含まれている。それはチーム向けと、状況報告のための実証的な測定の例を提供する。

第6章 プロジェクトのアジリティに対する組織的な考慮事項 — 本章は、文化、適用能力、ビジネス慣行、およびPMOの役割などアジャイル・アプローチの使用に影響する組織的要因を探る。

第7章 行動の喚起 — 実施要請は、本実務ガイドの継続的な改善のためのインプットを要求する。

付録、付属文書、参考文献、参考文献一覧、および用語集は、追加の役立つ情報や定義を提供する。

- ◆ **付録** 本実務ガイドの本文に含めるには長すぎる必須情報を含む。
- ◆ **付属文書** 本実務ガイドの本文を補完する情報を含む。
- ◆ **参考文献** 本実務ガイドの中で引用されている標準およびその他の出版物の出典を記載する。
- ◆ **参考文献一覧** 本実務ガイドで扱うトピックについての詳細な情報を提供する追加出版物を章単位でリストする。
- ◆ **用語集** 本実務ガイドで使用されている用語とその定義のリストを記載する。

2

アジャイル入門

2.1 定義可能な作業と不確実性の高い作業

プロジェクト作業には、定義可能な作業から不確実性の高い作業までである。定義可能な作業のプロジェクトは、過去の同様なプロジェクトで成功してきた明確な手続きによって特徴づけられる。設計が完了している車や電気製品の製造、または住宅の建築は、定義可能な作業の例である。関与する生産領域とプロセスは通常よく理解されており、一般的には、実行の不確実性やリスクのレベルは低い。

新しい設計、問題解決、およびかつて行われたことのない作業は探索的である。これには、ソリューションを作成するために協力して問題を解決する当該分野専門家が必要になる。不確実性の高い作業に関わる人々の例には、ソフトウェア・システム・エンジニア、プロダクト・デザイナー、医師、教師、弁護士、問題解決エンジニアなどが含まれる。定義可能な多くの作業が自動化されるのに伴って、プロジェクト・チームは、本実務慣行ガイドに記述されている技法を必要とする不確実性の高い作業のプロジェクトに取り組んでいる。

不確実性の高いプロジェクトでは、変更、複雑さ、リスクの率が高い。これらの特性は、要求事項の大部分を最初に決定し、変更要求プロセスを通して変更をコントロールすることを目指す従来の予測型手法の問題点を提示できる。代わりに、アジャイル・アプローチは、短い周期での実現可能性を探求し、評価とフィードバックに基づいて迅速に適応するために作成された。

2.2 アジャイル宣言とマインドセット

ソフトウェア業界の思想的リーダーたちは、2001年に『アジャイルソフトウェア開発宣言』（図2-1参照）を公表することでアジャイル運動を正式なものとした。

私たちは、ソフトウェア開発の実践あるいは実践の手助けをする活動を通じて、よりよい開発方法を見つけだそうとしている。この活動を通して、私たちは以下の価値に至った。

プロセスやツールよりも**個人と対話を**、
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**顧客との協調を**、
計画に従うことよりも**変化への対応を**、

価値とする。すなわち、左記のことがらに価値があることを認めながらも、私たちは右記のことがらにより価値をおく。

© 2001, the Agile Manifesto authors

図2-1.アジャイル宣言 4つの価値

図2-2に表示のように、これらの価値に由来する12の明確な原則。

1. 顧客満足を最優先し、価値のあるソフトウェアを早く継続的に提供します。
2. 要求の変更はたとえ開発の後期であっても歓迎します。変化を味方につけることによって、お客様の競争力を引き上げます。
3. 動くソフトウェアを、2-3週間から2-3ヶ月というできるだけ短い時間間隔でリリースします。
4. ビジネス側の人と開発者は、プロジェクトを通して日々一緒に働かなければなりません
5. 意欲に満ちた人々を集めてプロジェクトを構成します。環境と支援を与え仕事が無事終わるまで彼らを信頼します。
6. 情報を伝えるもっとも効率的で効果的な方法はフェイス・トゥ・フェイスで話をする事です。
7. 動くソフトウェアこそが進捗の最も重要な尺度です。
8. アジャイル・プロセスは持続可能な開発を促進します。一定のペースを継続的に維持できるようにしなければなりません。
9. 技術的卓越性と優れた設計に対する不断の注意が機敏さを高めます。
10. シンプルさ（ムダなく作れる量を最大限にすること）が本質です。
11. 最良のアーキテクチャ・要求・設計は、自己組織的なチームから生み出されます。
12. チームがもっと効率を高めることができるかを定期的に振り返り、それに基づいて自分たちのやり方を最適に調整します。

図2-2.アジャイル宣言の背後にある12の原則

もともとソフトウェア業界から発したものであるが、これらの原則は、その後他の多くの業界にも広まった。

何がアジャイル・アプローチを構成するかは、マインドセット、価値、および原則の実施形態によって定義される。今日使用されている様々なアジャイル・アプローチは、アジャイルのマインドセット、価値および原則と共通の根源を持っている。この関係を図2-3に示す。

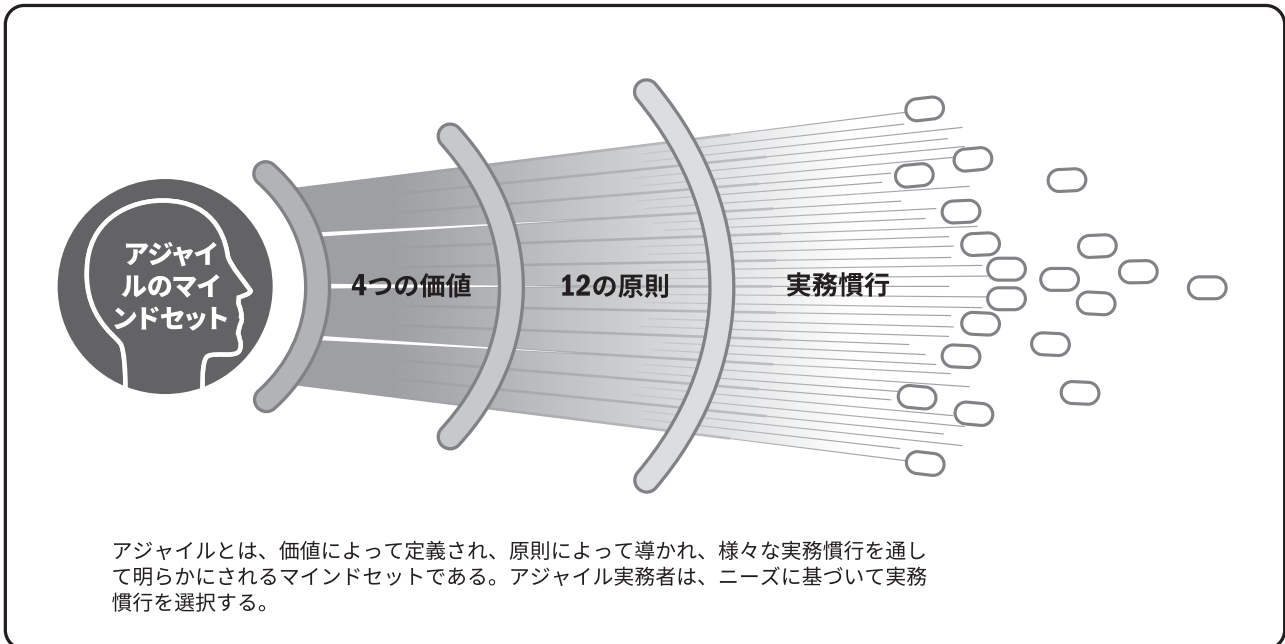


図2-3.アジャイル宣言の価値、原則および共通実務慣行の関係

図2-3に示されているように、Ahmed Sidky氏に触発されたこのモデルは、アジャイル宣言の価値で定義され、アジャイル宣言の原則に導かれ、さまざまな実務慣行によって可能になったマインドセットとして、アジャイルを明確に示している。宣言後は「アジャイル」という用語が普及されるようになった。一方で、今日プロジェクト・チームによって使用されているアジャイルのアプローチや技法がアジャイル宣言より数年、場合によっては数十年前から存在してきたことは注目に値する。

アジャイル・アプローチおよびアジャイル方式は、様々なフレームワークと方法を対象とする包括的用語である。図2-4は、アジャイルの位置付けを示し、アジャイル宣言の価値と原則を満たすあらゆる種類のアプローチ、技法、フレームワーク、方法、または実務慣行を総称する用語として図式化したものである。図2-4はまた、アジャイルとカンバン方式がリーンのサブセットであることを示している。なぜならば、それらが「価値に焦点を当てた」、「小さいバッチサイズ」、「無駄の排除」などのリーンの概念を共有するリーン思考の表した例だからである。

アジャイルは、アプローチ、方式、実務慣行、技法あるいはフレームワークのいずれなのか。状況に応じてこれらのいずれか、またはすべてが当てはまる。本実務慣行ガイドでは、他の表現が明らかに適切な場合を除き、「アジャイル・アプローチ」と表現する。

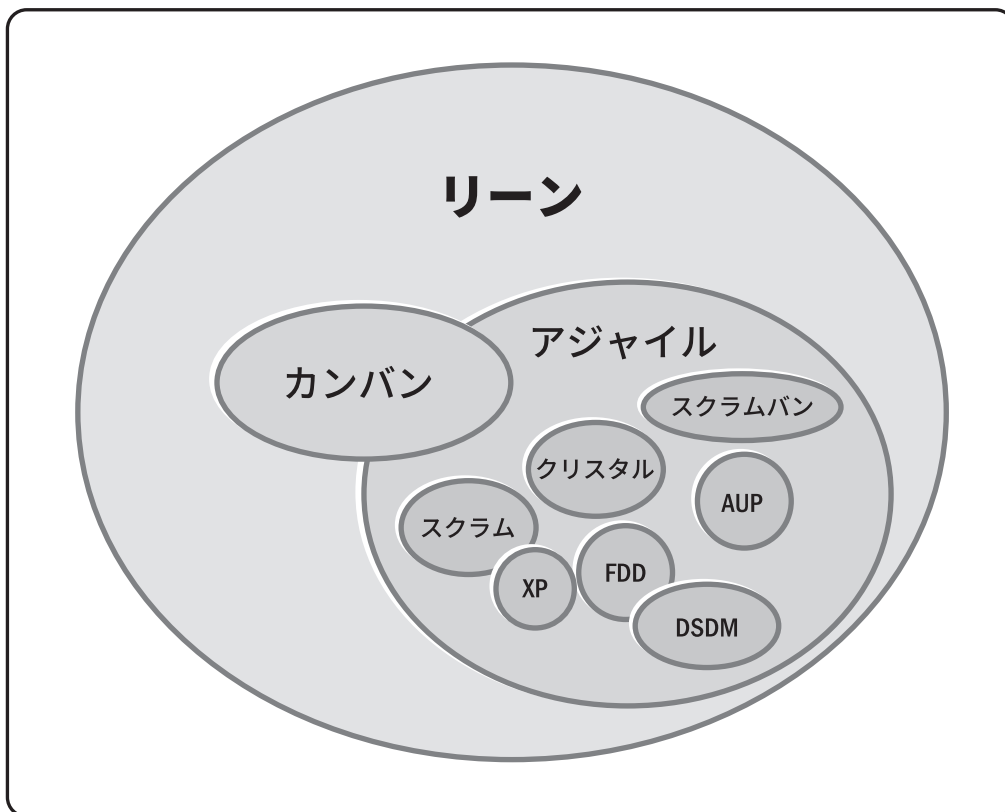


図2-4.アジャイルは多数のアプローチを総称する用語

一般に、アジャイルの価値と原則を満たすふたつの戦略がある。第一の戦略は、期待する結果を出すために意図的に設計され、実績が証明されている正式なアジャイル・アプローチを採用することである。そして、それを変更したりテラリングする前に、アジャイル・アプローチを学習し理解するために時間をかける。時期尚早で行き当たりばつりにテラリングすると、このアプローチの効果は最小限になり、ベネフィットも制限される。(テラリングの考慮事項については付属文書X2を参照。)

第二の戦略は、コア・バリューや原則の進展を実現するために、プロジェクトの状況に適した方法でプロジェクトの実務慣行を変更することである。タイムボックスを使用してフィーチャーを作成するか、または特定の技法を使って反復的にフィーチャーを洗練する。それが特定プロジェクトの状況で機能している場合、ひとつの大きなプロジェクトをいくつかのリリースに分割することを考慮する。プロジェクトの成功に役立つ変更を実施する。ただし、変更は組織の正式な実務慣行の一部である必要はない。最終ゴールは、アジャイルを行うこと自体ではなく、継続的に価値を顧客に提供し、よりよいビジネス成果を達成することにある。

2.3 リーンとカンバン方式

リーン、アジャイル、およびカンバン方式の関係について考察するひとつの方法は、アジャイルとカンバン方式がリーン思考に由来したものと捉えることである。言い換えると、リーン思考はアジャイルとカンバンの属性を共有する上位概念である。

この共有の継承された資産は非常に似ており、価値の提供、人間の尊重、ムダを最小化、透明性、変化への適応、継続的な改善に注力する。プロジェクト・チームは、時にはさまざまな方法を融合することが有用だと認識している。組織やチームにとって有益なものは何でも、どこに由来しているかを問わず行われるべきである。使用するアプローチが何かにかかわらず、最高の成果を上げることが目標である。

カンバン方式は、原初のリーン生産方式に触発され、とりわけ知的作業に使用される。それは2000年代半ばに、当時流行していたアジャイル方式への代替として出現した。

カンバン方式は、もともと「現状から始める」アプローチであるため、一部のアジャイル・アプローチと比べて規範的でも破壊的でもない。プロジェクト・チームは、カンバン方式を適用し始めることが比較的容易である。他のアジャイル・アプローチが必要もしくは適切と見られる場合は、それを適用することができる。カンバン方式に関する詳細は、付属文書A3「アジャイルおよびリーン・フレームワークの概要」を参照。



ケース

カンバン方式を取り巻く議論、そしてそれがリーンとアジャイルのいずれに帰属するかについての議論は多数あり、今後もあるだろう。それは、リーン生産方式およびその周辺で考えられたが、アジャイルの環境で広く使用されている。



2.4 不確実性、リスクおよびライフサイクルの選択

いくつかのプロジェクトには、プロジェクト要求事項と、現在の知識と技術を使用してその要求事項をどう満たすかについての多くの不確実性がある。これらの不確実性は、変更が高確率で発生したり、プロジェクトが複雑化したりする原因と成りうる。これらの特性を図2-5に図示する。

プロジェクトの不確実性が増加するにつれ、手直しのリスクと異なるアプローチを使用するニーズが同様に高まる。これらのリスクの影響を軽減するために、チームが不確実性の高いプロジェクトに取り組む際には、作業の小さな増分を介するライフサイクルを選択する。

チームは、小さい増分を使用するとき、作業を検証し、次にすることを変更することができる。チームが小さい増分を提供するとき、静的な仕様文書を使う場合より、迅速かつ正確に本当の顧客要求事項を理解することが可能になる。

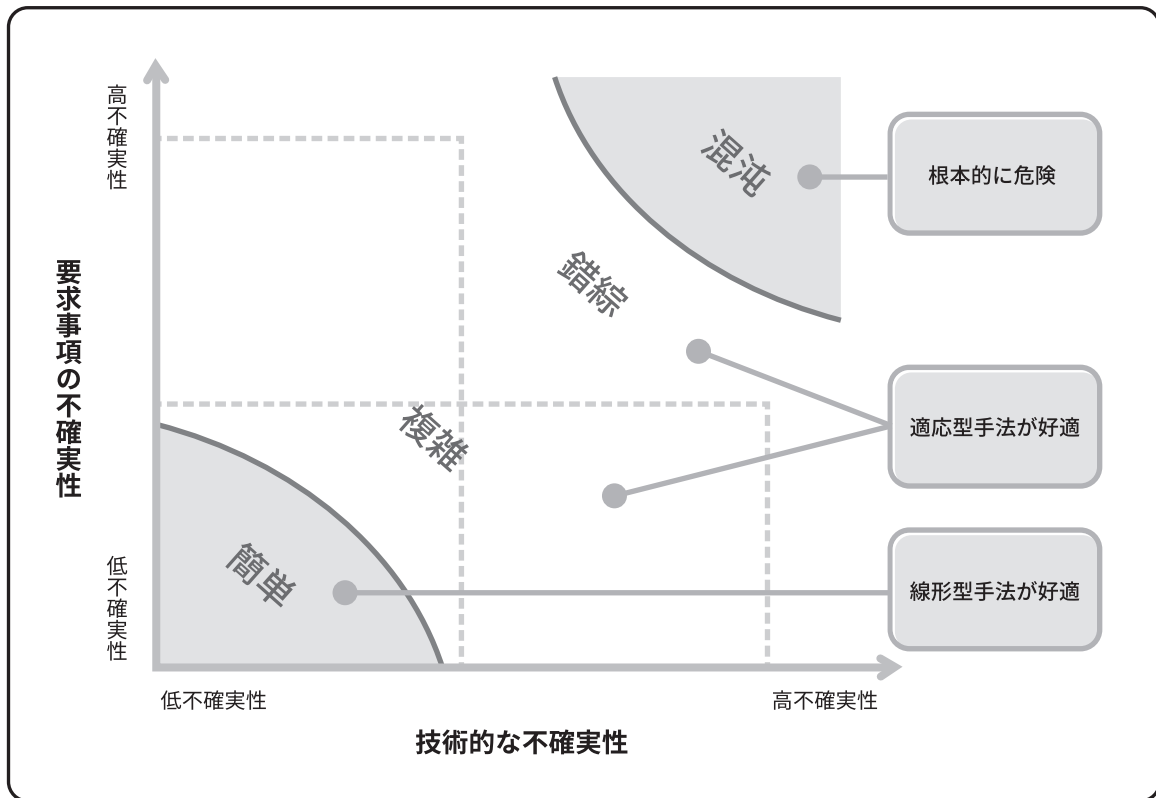


図2-5.ステーシー複雑性モデルに触発された不確実性と複雑性のモデル

チームは明確で安定した要求事項によってプロジェクトを計画しマネジメントすることができ、難なく技術的課題をクリアすることができる。しかし、プロジェクトの不確実性が増すにつれて、変更やムダな作業、手直しの可能性も同様に増加し、コストも時間もかかる。

一部のチームは、プロジェクト・ライフサイクルを展開して、反復型アプローチと漸進型アプローチを使用するようにした。多くのチームは、要求事項を反復的に探索し、漸進的により頻繁に納入するとき、変更への適用が容易であることを発見する。チームはフィードバックを得るので、これらの反復型および漸進型手法によってムダと手直しが減る。これらのアプローチが使用するものは次のとおりである。

- ◆ 非常に短いフィードバック・ループ
- ◆ プロセスの頻繁な適応
- ◆ 再優先順位付け
- ◆ 定期的に更新される計画
- ◆ 頻繁な引渡し

ヒント

簡単プロジェクト、複雑プロジェクト、錯綜プロジェクトとはどういう意味か。ボストン・ビッグ・ディグ建設プロジェクトのような大型プロジェクトを考えてみよう。表面的には、プロジェクトは高架道路を地下化するというかなり単純なものに見える。要求事項には高度な合意があった（図2-5のY軸を参照）。プロジェクトが開始するまで、プロジェクトをどのように進行するかについては確実性が低かった。そして、大型プロジェクトではよくあることだが、プロジェクトはその過程で予想外の驚きに遭遇した。

チームが中間成果物を提供する機会がほとんどないか、プロトタイピングの機会がほとんどないプロジェクトで作業する場合、チームは予測型ライフサイクルを使用してそれをマネジメントする可能性が高くなる。チームは発見したものに適応することができるが、アジャイル・アプローチを使用して要求事項の反復型の発見やフィードバックのための漸進的成果物をマネジメントすることはできない。

ビッグ・ディグ・プロジェクトは決して単純ではなかった。しかし、ステータシー複雑性モデルの左下部分から開始する多くのプロジェクトには、他のアプローチに移行する実際的手段がない。プロジェクトのライフサイクルに最適なアプローチを決定するために、要求事項と引渡し手段の両方でプロジェクトを評価する。

これらの反復型、漸進型、およびアジャイルのアプローチは、新しいツール、技法、マテリアル、またはアプリケーション領域が関わるプロジェクトに対してよく機能する。(第3章「ライフサイクルの選択」を参照)。それらはまた、次のようなプロジェクトに対してもよく機能する。

- ◆ 研究開発を必要とする
- ◆ 変更の確率が高い
- ◆ 不明確または未知の要求事項や、不確実性、またはリスクがある
- ◆ 最終ゴールを説明するのが難しい

小さい増分を開発してからそれをテストしレビューすることにより、チームは、短期間に低コストで不確実性を探索し、リスクを減らし、ビジネス価値の創出を最大化することができる。この不確実性は、適合性と要求事項（正しいプロダクトが作られているか）、技術的な実現可能性とパフォーマンス（このプロダクトはこの方法で開発できるか）、あるいはプロセスと人（これはチームが作業するのに効果的な方法か）に集中する。これら3つの特性（プロダクトの仕様、生産能力、およびプロセスの妥当性）のすべてに、一般的には高い不確実性要素がある。

しかし、反復型アプローチと漸進型アプローチは適用範囲が限られている。技術の不確実性と要求事項の不確実性の両方がとても高い（図2-5の右上）とき、プロジェクトは複雑から混沌へと移行する。プロジェクトが確実に可能になるためには、不確実性のひとつが含まれる必要がある。

3

ライフサイクルの選択

プロジェクトには多くの形態があり、さまざまな方法を取る。プロジェクト・チームは、その状況にとって最も成功する可能性の高いアプローチを選択するために利用できる特徴やオプションを認識している必要がある。

本実務慣行ガイドは、次のように定義される4つのタイプのライフサイクルに言及する。

- ◆ **予測型ライフサイクル** 大部分の計画を事前に実行してから、一回のパスで、すなわち逐次プロセスで実行する従来型の手法。
- ◆ **反復型ライフサイクル** 未完成の作業を改善し修正するために、作業のフィードバックできるようにするアプローチ。
- ◆ **漸進型ライフサイクル** 顧客が直ちに使用できる完成した成果物を提供するアプローチ。
- ◆ **アジャイル型ライフサイクル** 作業項目を絞り込んで頻繁に創出する反復型でもあり漸進型でもあるアプローチ。

非アジャイル・アプローチをどう呼ぶか

非アジャイル・アプローチを記述するために普遍的に使用される単一の用語はない。当初、本実務慣行ガイドでは、先行計画とその後の計画の実行への重点を記述するために計画駆動型という用語を使用した。このライフサイクルを記述するために、ウォーターフォールまたはシリアルという用語を好む人もいる。最終的には、それは『プロジェクトマネジメント知識体系ガイド（PMBOK®ガイド）』[3]と『PMBOK®ガイド 第5版 ソフトウェア拡張版』[4]で使用しているため、予測型という用語を確定した。

多くの組織は、このような極端な状況を経験するわけではなく、代わりに中間の状況となる。これは自然なことだが、ここでは両極端について話すことが必要である。「アジャイル型」が一端にあるとき、反対側を「予測型」と呼ぶ。

3.1 プロジェクト・ライフサイクルの特性

表3-1に、本実務慣行ガイドが取り扱うライフサイクルの4つのカテゴリーの特性を要約する。

表3-1.ライフサイクルの4つのカテゴリーの特性

特性				
手法	要求事項	活動	納品	目標
予測型	固定	プロジェクト全体で1回実行	1回の納品	コストのマネジメント
反復型	動的	是正されるまで反復	1回の納品	ソリューションの正しさ
漸進型	動的	増分毎に1回実行	頻繁で小さな納品	スピード
アジャイル型	動的	是正されるまで反復	頻繁で小さな納品	頻繁な納品とフィードバックを通じた顧客価値

すべてのプロジェクトにこれらの特性があって、いかなるプロジェクトも要求事項、引渡し、変更、およびゴールに対する考慮のないものはないことに注目する必要がある。プロジェクト固有の特性によってどのライフサイクルが最適かを決定する。

予測型ライフサイクルから始まって、反復型ライフサイクル、漸進型ライフサイクル、アジャイル型ライフサイクルに至る連続性を理解することで、プロジェクト・ライフサイクルの違いが明確になる。

『PMBOK®ガイド 第6版』の付属文書X3の図X3-1では、連続性は水平線上に表示されている。この表示は、一方の端から他方の端へのプロジェクト特性が変化していくことを強調している。図3-1では、2次元の正方形を用いて、連続性を視覚化している。

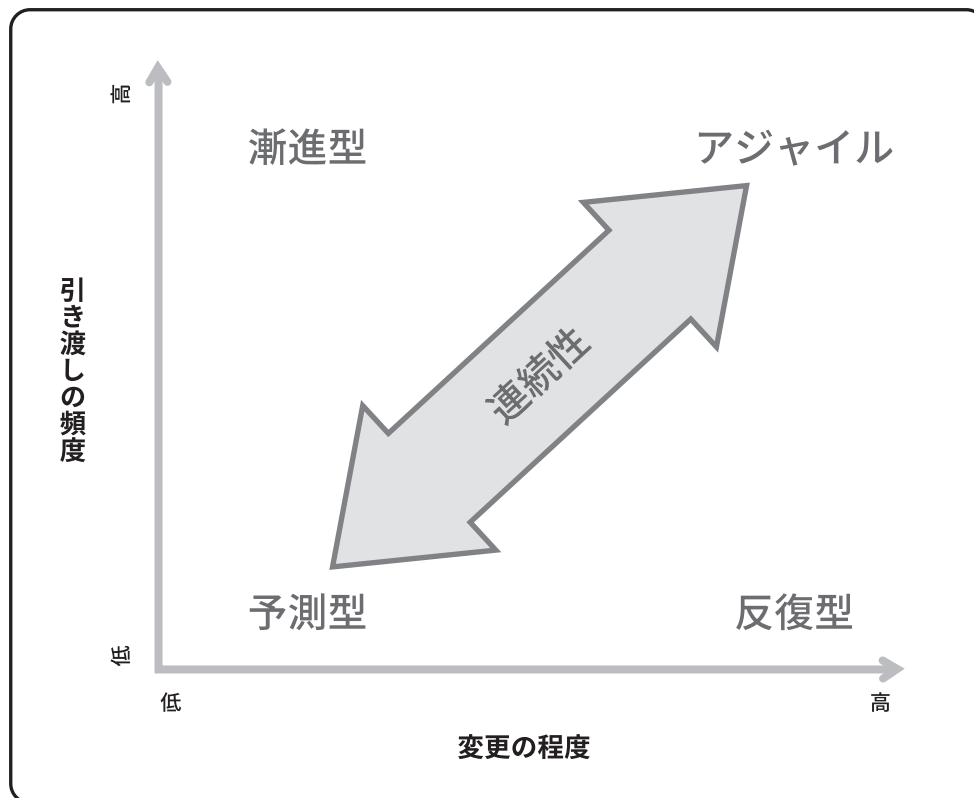


図3-1.ライフサイクルの連続性

すべてのプロジェクトに完璧なライフサイクルがあるわけではない。その代わりに、各プロジェクトは、その状況に応じた特性の最適なバランスを提供する位置を連続線上で見つける。具体的には、下記の通りである。

- ◆ **予測型ライフサイクル** 既知で証明済みのものを活用する。低減された不確実性と複雑さにより、チームは作業を予測可能な一連のグループに分けることができる。
- ◆ **反復型ライフサイクル** 部分的に完了した、または未完成の作業を改善し修正するために、作業のフィードバックできるようにする。
- ◆ **漸進型ライフサイクル** 顧客が直ちに使用できる完成した成果物を提供する。
- ◆ **アジャイル型ライフサイクル** 反復型と漸進型の両方の特性を活用する。チームがアジャイル・アプローチを使用するとき、完成した成果物を提供するために、プロダクトを反復させる。チームは早い段階でフィードバックを得て、顧客に認知、信頼、プロダクトのコントロールを提供する。チームは早めにリリースし、最も価値の高い作業を最初に提供できるので、プロジェクトは早期に投資収益を上げる可能性がある。

計画は常にそこにある

各ライフサイクルが計画の要素を共有することを覚えておくことが重要である。ライフサイクルを区別するものは、計画が完了しているかどうかではなく、むしろ計画がどの程度完了しているか、そしていつ完了するかである。

連続性の予測的終点で、計画は作業を駆動する。可能な限り多くの計画を前もって実行する。要求事項は可能な限り詳細に識別する。チームは、どの成果物をいつ納品し、いつ包括的な調達活動を行うことができるかを見積もる。

反復型アプローチでは、プロトタイプと実証も計画されているが、最初に作成された計画を修正したものをアウトプットとすることを想定している。未完成の作業の早期レビューは、将来のプロジェクト作業を知ることに関与する。

一方、漸進型施策はプロジェクト全体の連続したサブセットを納品するよう計画する。チームは、いくつかの連続した納品を事前に計画するか、または一度にひとつだけ計画することができる。納品は将来のプロジェクト作業を知らせる。

アジャイル型プロジェクトにおいても計画を立てる。主な違いは、頻繁な納品レビューからより多くの情報が入手可能になるのに伴い、チームが計画し再計画する点にある。プロジェクト・ライフサイクルのいかなる段階でも、プロジェクトには計画が必要となる。

3.1.1 予測型ライフサイクルの特性

予測型ライフサイクルでは、企業の要求事項、安定したチーム、低リスクに基づいた高い確実性を活用することが期待される。結果として、プロジェクト活動は、図3-2に示すように、しばしば連続的に実行される。

このアプローチを達成するために、チームには何をどのように納品するかを知るための詳細な計画が必要となる。これらのプロジェクトは、他の潜在的な変更（たとえば要求事項の変更や、プロジェクトのチーム・メンバーがチームの提供内容を変更するなど）が制限されている場合に成功する。チーム・リーダーは予測型プロジェクトへの変更を最小化しようとする。

チームがプロジェクトの初期段階で詳細な要求事項を作成し計画するとき、チーム・リーダーは制約条件を明示できる。チームはその制約条件を使ってリスクやコストをマネジメントできる。チームが詳細な計画に沿って進行する際に、彼らは、スコープ、スケジュール、あるいは予算に影響する可能性のある変更を監視・コントロールする。

部門効率性や連続した一連の作業実施を重視するため、予測型プロジェクトは通常、プロジェクトの終了時点まで事業価値を提供しない。予測型プロジェクトは、要求事項の変更や不一致に遭遇したり、技術的ソリューションがもはや簡単ではなくなったりすると、予期せぬコストを被ることになる。

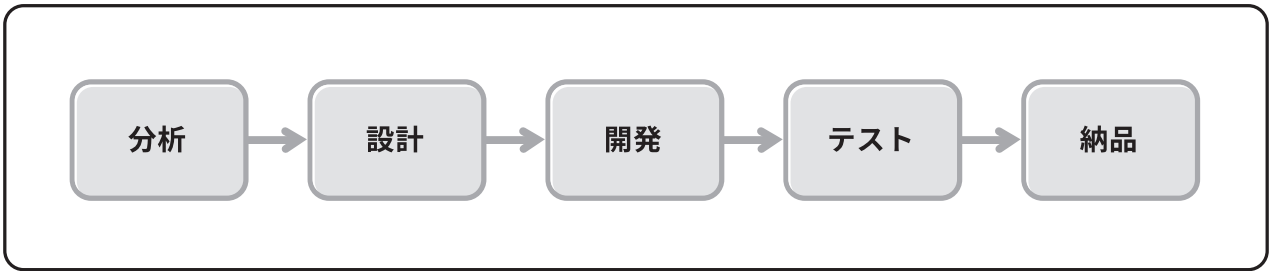


図3-2.予測型ライフサイクル

3.1.2 反復型ライフサイクルの特性

反復型ライフサイクルは、コンセプトの連続的プロトタイプや実証を介してプロダクトや所産を改善する。個々の新しいプロトタイプにより、新しいステークホルダーのフィードバックとチームの知見が得られる。そしてチームは、次のサイクルにひとつ以上のプロジェクト活動を繰り返すことにより、新しい情報を取り込む。チームは、指定されたイテレーションで数週間、タイムボックスを使用して知見を収集し、その知見に基づいて活動を手直しすることができる。このようにして、イテレーションはプロジェクトの不確実性を特定し、軽減するのに役立つ。

プロジェクトの複雑さが高い場合、プロジェクトが頻繁に変更される場合、あるいはスコープが目的とする最終プロダクトに対するステークホルダーの視点が異なる場合、プロジェクトは反復型ライフサイクルからベネフィットを得る。反復型ライフサイクルは、引渡しのスピードよりも学習することに最適化されているため、より長い時間がかかる。

図3-3に、単一プロダクトの引渡しのための反復型プロジェクト・ライフサイクルの要素を図示する。

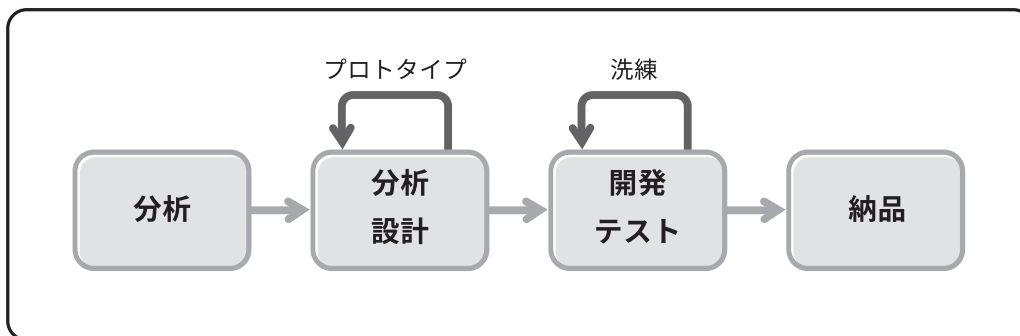


図3-3.反復型ライフサイクル

要求事項が毎日変更されるようなプロジェクトに関与したことがあり、「ビジネスが承認するプロトタイプを創出するときの要求事項がわかっている」と考えたことがあるか。ある場合には、これはアジャイル・アプローチが役に立つ可能性のあるプロジェクトである。プロトタイプは、各成果物に組み込むことができる要求事項に対するフィードバックと、それに対するより深い理解を促す。

3.1.3 漸進型ライフサイクルの特性

一部のプロジェクトは引渡しを最適化する。多くのビジネスや施策では、すべてが完了するのを待つ余裕がない。このような場合、顧客は全体的なソリューションの一部を受け取りたいと考えている。より小さな成果物のこの頻繁な引渡しは、漸進型ライフサイクルと呼ばれる(図3-4参照)。

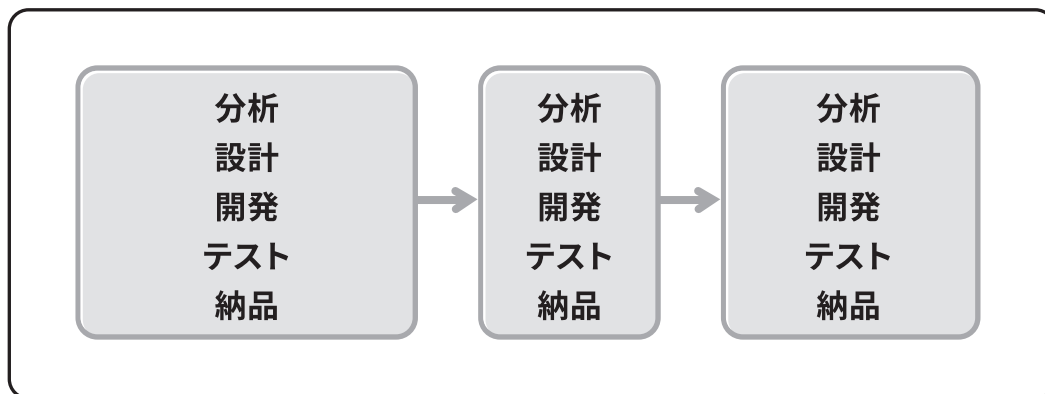


図3-4.増分がさまざまな大きさのライフサイクル

ヒント

新しいビジネス・サービスが実際にどう機能するか分からない。目的の成果を探索する評価基準を備えた概念実証 (PoC) を実施する。顧客のフィードバックに基づいて要求事項が変更されると思われる場合は、反復型アプローチを使用する。

漸進型ライフサイクルは、単一の最終プロダクトではなく、スポンサーや顧客により頻繁に価値を創出するために作業を最適化する。チームは作業を開始する前に最初の成果物を計画し、できるだけ早引渡しに取り掛かる。一部のアジャイル・プロジェクトは、プロジェクトの立上げから数日中に価値を創出する。その他のアジャイル・プロジェクトは、1週間から数週間の時間が掛る。

プロジェクトが継続するのに伴い、チームは初期のビジョンから逸脱する可能性がある。チームがより早く価値を創出するため、チームは逸脱をマネジメントすることができる。どれだけ変更や逸脱しているかよりも、顧客がプロジェクトの終了前に価値をより速やかに確実に得ることが重要である。

顧客に単一のフィーチャーまたは終了した作業結果を提供することは、漸進型アプローチの例である。

たとえば、建築業者は、建物の残作業を続ける前に、建物の完成した部屋やフロアを見せたい場合がある。その場合、建築業者は次のフロアに進む前に、そのフロア用の備品、塗料などでフロアを完成させることができる。顧客は、スタイルや色などの詳細を確認して承認することができ、さらに時間や資金をかける前に調整することができる。これにより、潜在的な手直しや顧客の不満が減る。

完全性と引渡しは主観的なものである。チームはプロトタイプに関するフィードバックを必要とし、一部の顧客に最小限の実行可能なプロダクト（MVP）を納品するように選択できる。顧客からのフィードバックは、最終的な完成したフィーチャーの後続の引き渡しに必要なものをチームが学習するのに役立つ。

アジャイル・チームは、重要な差別化要因として頻繁にビジネス価値を創出する。プロダクトが幅広いフィーチャーを追加し、対象となる消費者を拡大するとき、それは増分的に納品されると言える。

3.1.4 アジャイル型ライフサイクルの特性

アジャイル環境において、チームは要求事項が変更されることを予想する。反復型アプローチと漸進型アプローチは、プロジェクトの次の部分をよりよく計画するためのフィードバックを提供する。しかし、アジャイル・プロジェクトでは、漸進型引渡しによって隠された、または誤解された要求事項が明らかになる。図3-5は、プロジェクトが顧客のニーズに沿っており、必要に応じて適応させることができる漸進型作成を達成するための2つの可能な方法を図示したものである。

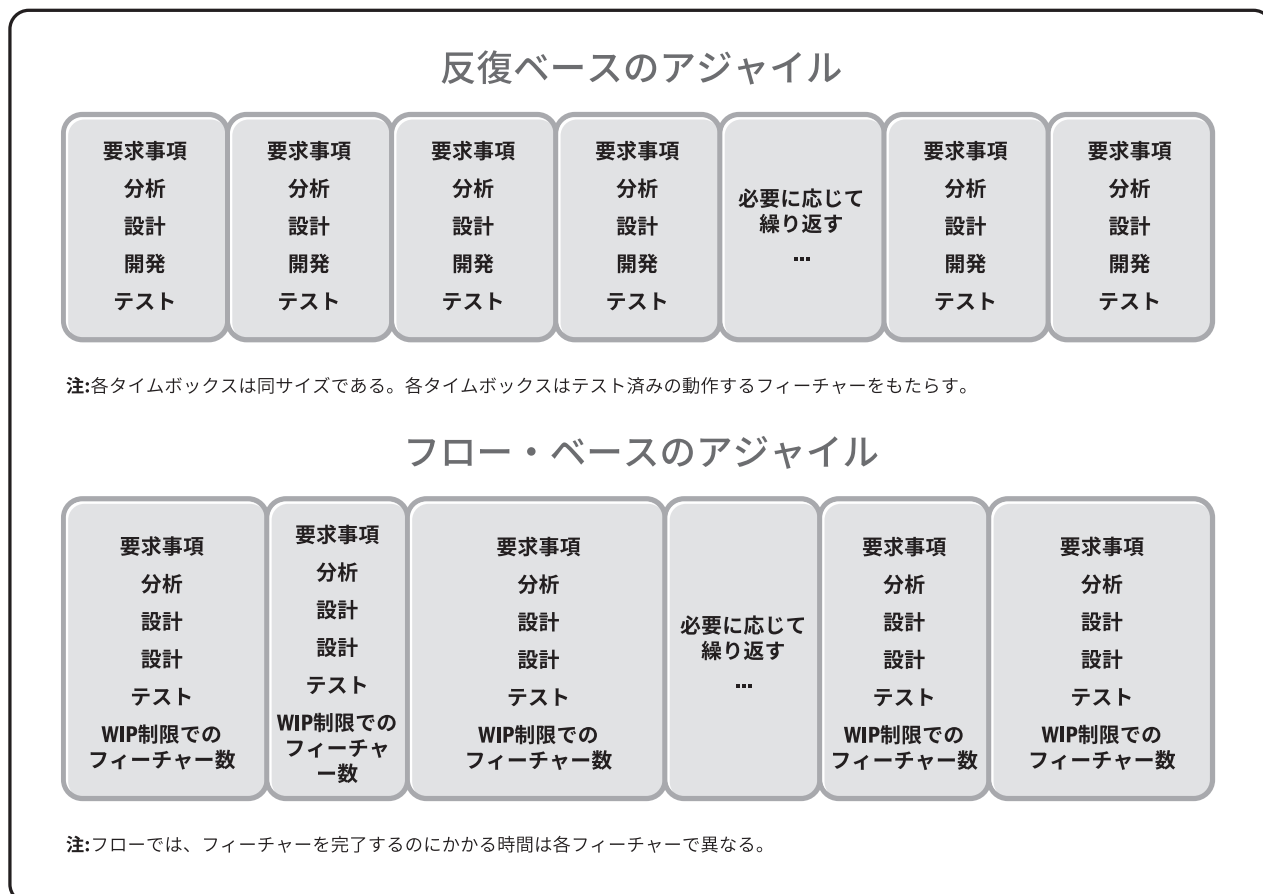


図3-5.反復ベースおよびフロー・ベースのアジャイル型ライフサイクル

反復ベースのアジャイルでは、チームは完成したフィーチャーを引渡しするのにイテレーション（同サイズのタイムボックス）で作業する。チームは最重要フィーチャーに取り組み、チームとして協力してそれを完了する。そしてチームは次の最も重要なフィーチャーに取り組み、それを完了させる。チームは一度にいくつかのフィーチャーに取り組むように決めるかもしれないが、イテレーション作業のすべてを一度に扱うわけではない（すなわち要求事項のすべてを扱い、そのあと分析のすべてが続くわけではない）。

フロー・ベースのアジャイルでは、チームは反復ベースのスケジュールではなく、作業を開始する遂行能力に基づいてバックログからフィーチャーを引き出す。チームは、タスクボード上の列を使用してワークフローを定義し、各列の進行中の作業をマネジメントする。各フィーチャーを完了するのにかかる時間はそれぞれ異なる。チームは、進行中の作業のサイズを小さく保ち、変更が必要な場合も手直しを減らせるように、課題を早期に特定する。計画とレビュー・ポイントを定義するイテレーションなしに、チームとビジネス・ステークホルダーは、計画、プロダクト・レビュー、およびレトロスペクティブ（振り返り）の最も適切なスケジュールを決定する。

アジャイル型ライフサイクルは、アジャイル宣言の原則を満たすものである。特に、価値のあるプロダクトを早期にかつ継続的に引渡しすることで、顧客満足度が向上する。さらに、機能的で価値を提供する漸進的な成果物が、進捗の主要な尺度となる。アジャイル型ライフサイクルは、高度な変化に対応し、より多くのプロジェクト価値を提供するために、反復型アプローチと漸進型アプローチの両方を組み合わせる。

3.1.5 アジャイル適合性フィルター

さまざまな評価モデルが存在し、アジャイル・アプローチを使用した場合の適合性やギャップを判断するのに役立つ。これらのモデルは、採用と適合性に関連するプロジェクトや組織の要素を評価し、適合性や潜在的风险領域を示すスコアを提供する。付属文書X3は、アジャイル適合性フィルターとして使用するための一般的な評価モデルを提供する。

ハイブリッド型ライフサイクル・プロジェクトの例

製薬会社で、開発プロセスの終了時に時間を要する米国食品医薬品局（FDA）の承認プロセスが必要なライフサイクル全体が図3-6のようになった。プロジェクト・チームはアジャイル方法で薬物試験を実施したが、FDA承認プロセスを実行するためには、外部グループに薬剤を提示しなければならなかった。コンサルタントは、より合理的なハイブリッド・アプローチを作成するために、FDA承認プロセス部分をアジャイル開発プロセスに統合するよう助力した。

この話を要約すると、FDA承認が開発プロセスの終了時に完了しているか、または変更後にFDA承認の必要があるため（これにはわずかな変更が加えられた後も含まれる）、開発プロセスは終了時点で別のフェーズとして残らなければならなかったということである。反復プロセスを使用した統合は失敗した。しかし、コンサルタントは、最終的なFDA承認プロセスを短縮した有用なクイック・スタート・ガイドとテスト・プロトコルを作成した。

3.1.6 ハイブリッド型ライフサイクルの特性

プロジェクト全体に対して単一のアプローチを使用する必要はない。プロジェクトはしばしば、特定のゴールを達成するために、異なるライフサイクルの要素を結合する。予測型、反復型、漸進型やアジャイル方式を組み合わせたものがハイブリッド・アプローチである。

図3-6は、ハイブリッド・モデルを形成するために結合したプロジェクト・タイプに対する基本的で単純なアプローチを示す。初期過程では、アジャイル型開発ライフサイクルを使用し、その後は予測型展開フェーズが続く。アジャイル・アプローチからベネフィットを受けるプロジェクトの開発部分に不確実性、複雑性、およびリスクがあって、その後、予測型方法で、おそらくは別のチームが引き受けるのが適切な、定義され、反復可能な展開フェーズが続くとき、このアプローチを使用することができる。このアプローチの一例は、新しいハイテク・プロダクトの開発と、それに続く数千人のユーザーへの展開とトレーニングである。



図3-6.アジャイル型開発と後続する予測型展開

3.1.7 アジャイルと予測型を結合したアプローチ

別のアプローチは、ライフサイクルを通してアジャイル・アプローチと予測型アプローチを結合して使用するものである。



図3-7.アジャイルと予測型を同時に結合して使用するアプローチ

図3-7では、同じプロジェクトにアジャイル・アプローチと予測型アプローチの結合が使用されている。おそらく、チームは漸進的にアジャイルに移行しており、短いイテレーション、毎日のスタンドアップ・ミーティング、およびレトロスペクティブのようなアプローチを使用しているが、先行見積り、作業の割当て、進捗状況の追跡など、プロジェクトの他の側面では、依然として予測型アプローチに従っている。

予測型アプローチとアジャイル・アプローチの両方を使用することは、よくあるシナリオである。それは明らかに、アジャイルのマインドセット、価値、および原則を完全に具現化しているわけではないため、このアプローチをアジャイルと呼ぶのは誤解を招く恐れがある。しかし、それは一種のハイブリッド・アプローチであるため、予測型と呼ぶのも正確ではない。

3.1.8 いくつかのアジャイル構成要素を含む予測型アプローチ

図3-8は、小さなアジャイル要素を含む予測型プロジェクトを示す。この例では、不確実性、複雑さ、またはスコープ・クリープの可能性があるプロジェクトの一部はアジャイル方法で取り組まれているが、プロジェクトの残りの部分は予測型アプローチでマネジメントされている。このアプローチの一例は、新しい構成要素を備えた設備を構築するエンジニアリング企業がある。

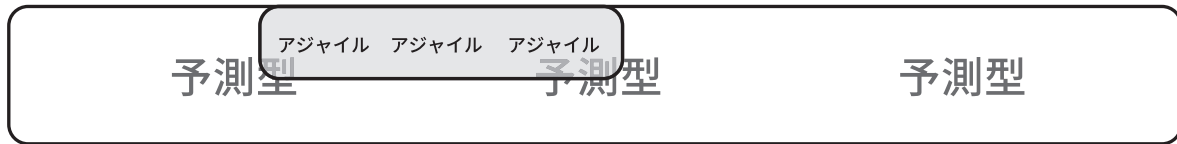


図3-8.いくつかのアジャイル構成要素を含む予測型アプローチ

組織がこれまでに実施した他の多くの設備プロジェクトと同様に、プロジェクトの大部分は定型作業であり予測可能であるが、このプロジェクトでは新しい屋根材を取り扱う。請負業者は、最適な設置方法を決定し、早期に課題が判明しても、問題を解決して実験と適応を通してプロセスを漸進的に改善するのに十分な時間が持てるよう、まずは地上で小規模な施工試験を計画する。

3.1.9 いくつかの予測的構成要素を含むアジャイル・アプローチ

図3-9はいくつかの予測的構成要素を含むアジャイル・アプローチを示す。このアプローチは、交渉の余地がない特定要素があるか、またはアジャイル・アプローチだけでは実行できない場合に使用される。例としては、協力的または漸進的な方法ではパートナーを組むことができない、またはパートナーを組む予定のない外部ベンダーによって開発された外部コンポーネントを統合することがある。そのコンポーネントが納品された後、一回の統合が必要となる。



図3-9.いくつかの予測的構成要素を含むアジャイル・アプローチ

ある政府機関に、信用保険アプリケーション開発プロジェクトがあった。この複数年プロジェクトは、老朽化した引受業務システムを、より反応が早いユーザー・インターフェイスを持つ新しい統合されたシステムに置き換えるというものだった。プロジェクトの大部分は、継続的な業務インプットを伴うアジャイル・アプローチを使用して実施された。

保険料率の計算方法は、経済協力開発機構（OECD）から200ページの仕様書として手渡された。この手順は、不明瞭（すなわちビジネスによる中間結果の確認）の可能性がほとんどなく、非常に明確に説明されており、計算手順を順番に処理する別のチームによりコード化されていた。このふたつのチームは、計算に必要なインプット変数と、アウトプット値の処理方法および表示方法については協力したが、それ以外では、計算チームは主に予測型方法で作業した。

計算チームの作業が完了したとき、保険料率計算からのアウトプットは、スクリーンおよびレポートに表示された。そしてビジネス・ユーザーは、情報の外観と使用に関するフィードバックを提供した。ふたつのチームは同時に実行したが、交流のニーズはほとんどなかった。彼らを物理的に接近させることで、開発の進捗状況を確認することが容易になったが、大部分は別々のサブプロジェクトだった。

3.1.10 目的適合としてのハイブリッド型ライフサイクル

プロジェクト・チームは、プロジェクト・リスクに基づいてハイブリッド型ライフサイクルを設計することがある。たとえば、キャンパスの建設プロジェクトには、改築したり建設したりする複数の建造物が含まれる。漸進型アプローチでは、いくつかの建造物を他の建造物よりも早期に完成することに資源を集中させ、投資対効果を加速させる。それぞれ個別の建造物を引き渡すことにより、その建造物に関する予測型ライフサイクルからベネフィットを得ることが十分によく知られている。

プロジェクトマネジメントのゴールは、現在の環境を前提として可能な限り最良の方法で事業価値を生み出すことである。その方法がアジャイルであるか予測型であるかは問題ではない。問うべき問題は次の通りである。「どうすれば最も成功できるか。」

チームが価値を生み出す際に、フィードバックは必要か。必要なら、増分が役立つ。アイデアを探索する際に、リスクをマネジメントする必要があるか。必要がある場合は、イテレーションまたはアジャイルが役立つ。

組織が中間価値を創出できない場合、アジャイル・アプローチは有用ではない可能性がある。それはかまわない。アジャイルのためのアジャイルはゴールではないのだから。重要なのは、プロジェクト、リスク、および文化のために機能するライフサイクル、またはライフサイクルの組合せを選択することである。

アジャイルとは、顧客ベースで頻繁に引渡しすることである。その引き渡しはチームへのフィードバックを作成する。チームはそのフィードバックを使用して、次の作業を計画し再計画する。

3.1.11 移行戦略としてのハイブリッド型ライフサイクル

多くのチームは、一晩中作業するアジャイル方法に切り替えることができない。アジャイル技法は、予測型環境になじんで成功してきた人にとっては、見た目も感じも異なっている。組織が大きくなり、可動部署が多くなればなるほど、移行に時間がかかる。そのため、漸進的な移行を計画する方が理にかなっている。

漸進的な移行には、チームやステークホルダー間の学習や適合性を改善するためにより反復的なアプローチを追加することが含まれる。その後、価値を高め、スポンサーへの投資対効果を上げるために、より多くの漸進的な技法を追加することを検討する。様々なアプローチのこの組合せは、ハイブリッド・アプローチと見なされる。

不確実性が中程度から低程度の、リスクの低いプロジェクトでこれらの新しい技法を試してみる。そして、組織がハイブリッド・アプローチで成功する場合は、さらに多くの技法を追加する必要がある、より複雑なプロジェクトで試してみる。これは、組織の状況と具体的なリスクへの漸進的なハイブリッド移行を、そしてチームが変化に適応して受け入れる準備ができているかどうかをテラリングする方法である。

3.2 アジャイル・アプローチの混合

アジャイル・チームがその実務慣行をひとつのアジャイル・アプローチに限定することはめったにない。各プロジェクトの背景には、チーム・メンバーのスキルと背景の多様な組み合わせ、開発中のプロダクトのさまざまなコンポーネント、そして作業が行われる環境の年数、規模、緊急度、複雑さ、および規制上の制約など、独自の特殊性がある。

アジャイル・フレームワークは、チームのためにカスタマイズされてはいない。チームは、定期的に価値を提供するために実務慣行をテラリングする必要がある。しばしばチームは、たとえ出発点として特定のフレームワークを使用する場合でも、独自の特別なアジャイルの融合を実践することがある。

融合アプローチ

アジャイル・フレームワークのテラリングの例として広く普及している最も一般的な融合のひとつには、スクラム・フレームワーク、カンバン方式、エクストリーム・プログラミング (XP) 方式の要素を調整して使用することが含まれる。スクラムは、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、スプリント・レトロスペクティブ・セッションを含む、プロダクト・バックログ、プロダクト・オーナー、スクラム・マスター、および機能横断開発チームの使用に関するガイダンスを提供する。カンバンは、作業の流れを視覚化し、障害を容易に可視化し、プロセス限界内の作業を調整することによってフローをマネジメントできるようにすることにより、チームがその有効性をさらに向上させるのに役立つ。さらに、ストーリー・カードの使用、継続的統合、リファクタリング、自動テスト、テスト駆動開発など、XPに基づくエンジニアリングの実務慣行は、アジャイル・チームの有効性をさらに高める。要約すると、これらのさまざまなアプローチの実務慣行の融合は、個々の孤立した構成要素よりも高いパフォーマンスの相乗効果をもたらす。

3.3 テーラリングに影響するプロジェクト要因

時々、プロジェクトの特性によっては、さらに適したアプローチをテーラリングする必要がある。表3-2は、考慮すべきプロジェクト要因とテーラリング・オプションを特定している。

表3-2.適合性を改善するテーラリング・オプション

プロジェクト要因	テーラリング・オプション
要求パターン: 安定的または散発的	多くのチームは、(定期的なタイムボックスの形で) カデンツを使用することが、新しい作業のデモ、レトロスペクティブ、取り込み役に役立つことを知る。さらに、一部のチームでは、より多くの作業を受け入れるために一層の柔軟性が必要となる。チームは両方でベストを得るために、カデンツでフロー・ベースのアジャイルを使用することができる。
チームの経験レベルによって必要とされるプロセス改善の割合	より頻繁にレトロスペクティブを行い、改善点を選択する。
作業フローが様々な遅れや障害によってしばしば中断される	カンバンを使用し、フローを改善するために作業プロセスのさまざまなエリアの限界を検証することにより、作業の可視化を検討する。
プロダクト増分の品質が悪い	さまざまなテスト駆動型開発の実務慣行の使用を検討する。この間違い防止の原則により、欠陥が放置されにくくなる。
プロダクトの開発に複数のチームが必要	混乱を最小限に抑えながら、ひとつのアジャイル・チームから複数にスケールアップするには、まずアジャイル・プログラムのマネージメントまたは正式なスケリングのフレームワークについて学ぶ。そして、プロジェクトの状況に合った手法を作成する。
プロジェクトのチーム・メンバーがアジャイル手法の経験不足	アジャイルのマインドセットと原則の基礎についてチーム・メンバーのトレーニングを始めることを検討する。チームがスクラムやカンバンなどの特定手法を使用するよう決定した場合は、チーム・メンバーが使用法を学ぶことができるように、その手法のワークショップを提供する。

テーラリングに影響を及ぼす要因についての追加ガイダンスは、「テーラリングに影響する属性」に関する付属文書X2を参照。

4

アジャイルの実装：アジャイル環境の作成

4.1 アジャイルのマインドセットで開始

アジャイル・アプローチを使ってプロジェクトをマネジメントするには、プロジェクト・チームがアジャイルのマインドセットを採用する必要がある。つぎの質問への答えは、実装戦略を立てるのに役立つ。

- ◆ プロジェクト・チームはアジャイル・アプローチでどう行動できるか。
- ◆ チームは何を迅速に引き渡し、次の引渡しサイクルに役立つどのような早期のフィードバックを得ることができるか。
- ◆ チームは透明な方法でどう行動できるか。
- ◆ 優先度の高い項目に注力するためには、どのような作業を回避できるか。
- ◆ サーバント・リーダーシップ手法はチームのゴールの達成にどのように役立つか。

4.2 チームに力を与えるサーバント・リーダーシップ

アジャイル・アプローチは、チームに力を与える方法としてサーバント・リーダーシップに重点を置く。サーバント・リーダーシップとは、可能な限り最高のチーム・パフォーマンスを実現するために、チーム・メンバーのニーズと開発を理解し対処することに注力し、チームへのサービスを通じた主導を行う実務慣行である。

サーバント・リーダーの役割は、チームがアジャイルを発見し定義するのを促進することである。サーバント・リーダーはアジャイルを実践し広める。サーバント・リーダー手法のプロジェクトは次の順序で機能する。

- ◆ **目的** プロジェクトのゴールに向けて参画し融合できるように、チームと協力して「理由」または目的を定義する。チーム全体がペルソナ・レベルではなくプロジェクト・レベルで最適化される。
- ◆ **人** 目的が確立されると、チームを鼓舞して全員が成功できるような環境を作成する。各チーム・メンバーに、プロジェクト作業全体にわたって貢献するよう要請する。
- ◆ **プロセス** 「完璧な」アジャイル・プロセスに従うことを計画するのではなく、むしろ結果を出すことにこだわる。完成した価値を機能横断チームが創出し、プロダクトおよびプロセスに反映させるとき、チームはアジャイルだと言える。チームがそのプロセスを何と呼ぶかは問題ではない。

サーバント・リーダーシップの次の特性は、プロジェクト・リーダーがよりアジャイルになることを可能にし、チームの成功を促進する。

- ◆ 自己認識の促進
- ◆ 傾聴
- ◆ チーム・メンバーへの奉仕
- ◆ 人の成長の支援
- ◆ コーチング対コントロール
- ◆ 安心、敬意、信頼の促進
- ◆ 人の気力と知性の促進

サーバント・リーダーシップは、アジャイルに固有というわけではない。しかし、それを実践した後は、サーバント・リーダーは、通常、サーバント・リーダーシップがアジャイルのマインドセットや価値にどのように統合されるかを見ることができる。

リーダーがそのサーバント・リーダーシップや促進するスキルを開発すると、アジャイルになる可能性が高くなる。その結果、サーバント・リーダーは、チームが協働してより速く価値を創出することを支援できる。

成功するアジャイル・チームは成長のマインドセットを受け入れ、そこで新しいスキルを学ぶことができると確信する。チームとサーバント・リーダーが共に学べると確信するとき、すべての人がより能力を発揮する。

4.2.1 サーバント・リーダーの責任

サーバント・リーダーは、チーム内と組織全体間のコミュニケーションを確立し連携を図るために関係をマネジメントする。これらの関係はリーダーが組織をナビゲートしてチームを支援するのに役立つ。この種の支援は、障害を取り除くのに役立ち、チームがプロセスを合理化することを促進する。サーバント・リーダーはアジャイルを理解し、アジャイルの特定手法を実践するため、チームのニーズを満たすことを支援できる。

4.2.1.1 サーバント・リーダーはファシリテートする

プロジェクト・マネジャーがサーバント・リーダーとして行動するとき、「調整のマネジメント」から「コラボレーションの促進」に重点を移す。ファシリテーターは、誰もがベストの思考と仕事ができるよう支援する。ファシリテーターは、チームの参加、理解、アウトプットの責任の共有を奨励する。ファシリテーターは、チームが受入れ可能なソリューションを作成することを支援する。

サーバント・リーダーは、チーム内およびチーム間のコラボレーションと対話を促進する。例えば、サーバント・リーダーは、チーム内およびチーム間でのボトルネックを顕在化させ伝達するのを支援する。こうしてチームはこのボトルネックを解決する。

さらに、ファシリテーターは、対話のある会議、非公式の会話、知識の共有を通してコラボレーションを促進する。サーバント・リーダーは、他の人が責任を負わなければならないような決定を下すことでこれを行うのではなく、公平な橋渡し役や指導者になることによって行う。

4.2.1.2 サーバント・リーダーは組織の障害を取り除く

アジャイル宣言の第一の価値は、プロセスやツールよりも個人や相互作用である。サーバント・リーダーは、チームや組織のアジリティを妨げているプロセスを綿密に検討し合理化するよりも、より完全な責任を負う。例えば、ある部署が詳細な文書化を必要とする場合、サーバント・リーダーの役割は、その部署と協力して必要な文書をレビューし、アジャイルな成果物がそれらの要求事項をどう満たしているかの理解を共有し、必要な文書の量を評価することである。それによってチームは、文書作成にではなく貴重なプロダクトを提供することに多くの時間を費やすことができる。

サーバント・リーダーは、ボトルネックの原因となり、チームや組織のアジリティを妨げている時間を要する他のプロセスにも目を向ける必要がある。対処の必要があるプロセスや部門の例には、財務部門、変更管理委員会、または監査が含まれる。サーバント・リーダーは、他の人と連携し協力して、アジャイル・チームやリーダーをサポートするためにプロセスをレビューするよう促す。例えば、チームにとって、作業プロダクトを2週間に一度引き渡しして待ち行列に入れるだけであることと、長いリリース・プロセスのためリリースに6週間以上かかることもあることとどちらがいいだろうか。あまりに多くの組織が、貴重なプロダクトやサービスをチームが迅速に引き渡すことを妨げる「ボトルネック」となるプロセスを抱えている。サーバント・リーダーは、これらの組織的な障害を変更または除去して引渡しチームを支援する能力を持っている。

人間関係のスキル 対 技術的スキル

サーバント・リーダーシップに加えて、チーム・メンバーは、技術的スキルだけでなく、その人間関係のスキルと感情的知能のスキルにも重点を置く。チームの全員が、より多くの施策、インテグリティ、感情的知能、誠実さ、コラボレーション、謙虚さ、そしてチーム全体がよく連携できるように様々な方法でコミュニケーションをとる意欲を示すよう努力する。

チームは、プロジェクトの方向性の変更や、技術的なプロダクトの変更に適切に対応するために、これらのスキルを必要とする。誰もが仕事に、そして互いに適応できる時、チーム全体が成功する可能性はより高くなる。

4.2.1.3 サーバント・リーダーは他の人の貢献のため下準備をする

アジャイルでは、チームがその作業プロセスと作業プロダクトをマネジメントする。自己管理および自己組織化は、組織とプロジェクトにサービスを提供し支援している人全員に適用される。サーバント・リーダーは、チーム、プロジェクトおよび組織のニーズを満たすために努力する。サーバント・リーダーは、チーム・スペースのために施設の調整をしたり、チームが一度にひとつのプロジェクトに集中できるように上層部と調整したり、チームとストーリーを開発するためにプロダクト・オーナーと調整したりする。一部のサーバント・リーダーは、監査と調整して法的規制環境で必要とされるプロセスを洗練したり、財務部門と連携して組織を漸進型予算編成に移行したりする。

サーバント・リーダーは、チームが最高の仕事が遂行できるよう下準備に注力する。サーバント・リーダーは、プロジェクトに影響を与え、組織が今とは違った考え方ができるようになることを奨励する。

4.2.1.4 サーバント・リーダーの責任を考慮する

サーバント・リーダーには、可能性のある多くのタイトルをつけることができるが、最も重要なことは何をするかである。サーバント・リーダーが有する責任の例を次に示す。

- ◆ アジャイルである理由と方法についてステークホルダーを教育する。優先順位付け、権限を持ったチームのより大きな説明責任と生産性、より頻繁なレビューによる品質向上などに基づく、事業価値のベネフィットを説明する。
- ◆ 指導、励まし、支援を通してチームをサポートする。チーム・メンバーのトレーニングとキャリア開発を主導する。矛盾する表現ではあるが「チームの背後にいてチームを主導する」とは、チーム・メンバーを育成する上でのリーダーの役割を語っている。支援や励まし、プロフェッショナル育成を通して、チーム・メンバーは自信を得、より大きな役割を果たし、組織内のより高い地位で貢献する。サーバント・リーダーの重要な役割は、たとえ彼らがチームからいなくなったとしても、現在の役割を介して、そして役割を越えてチーム・メンバーを育成し成長させることである。
- ◆ リスクの定量的分析のようなプロジェクトマネジメントの技術的アクティビティでチームを助ける。チーム・メンバーは、役割や機能の知識や経験をもっていないことがある。技術を持っていたり、トレーニングを受けたことがあるサーバント・リーダーは、チーム・メンバーにトレーニングを実施するか、これらのアクティビティを引き受けることでチームを支援できる。
- ◆ チームの成功を祝い、外部グループとの活動の構築を支援し橋渡しする。コラボレーションの強化に対する感謝と善意の上昇スパイラルを創出する。

4.2.2 アジャイル環境におけるプロジェクト・マネジャーの役割

多くのアジャイルなフレームワークや手法はプロジェクト・マネジャーの役割に言及していないため、アジャイル・プロジェクトにおけるプロジェクト・マネジャーの役割は不明なところがある。一部のアジャイル実務者は、自律型チームがプロジェクト・マネジャーの責任を引き受けるため、プロジェクト・マネジャーの役割は必要ないと考える。しかし、実用的なアジャイル実務者や組織は、プロジェクト・マネジャーが多くの状況で重要な役割を担えることを認識している。これまでのプロジェクト・マネジャーとの決定的な違いは、役割と責任が幾分異なって見えることである。

ヒント

プロジェクト・マネジャーの価値は、その地位にあるのではなく、他のすべての人をより良くする能力にある。

4.2.3 プロジェクト・マネジャーはサーバント・リーダーシップを使用する

『PMBOK®ガイド 第6版』は、プロジェクト・マネジャーを、「プロジェクト目標を達成することに責任をもつチームをリードするために、母体組織が任命する人物」と定義している。

多くのプロジェクト・マネジャーは、プロジェクトを中心となって調整し、チームの状況を追跡し、同じ組織の他部署に示すことに慣れている。この手法は、プロジェクトがサイロ化された機能に要素分解されたときには問題がなかった。

しかし、変更の多いプロジェクトには、ひとりでマネジメントできる以上の複雑さがある。その代わりに、機能横断チームが自身の作業を調整し、ビジネス担当者（プロダクト・オーナー）と協働する。

アジャイル・プロジェクトにおいては、プロジェクト・マネジャーは中心であることからチームと上層部に役立つことにシフトする。アジャイル環境では、プロジェクト・マネジャーはサーバント・リーダーであり、支援の必要な人を指導し、チームのコラボレーションを強化し、ステークホルダーのニーズを調整することに重点をおくことに変更する。サーバント・リーダーとして、プロジェクト・マネジャーは、作業を遂行するための知識を持っている人への責任の委譲を奨励する。

4.3 チーム構成

アジャイル宣言の価値と原則の両方の根底にある教義は、個人と相互作用の重要性である。アジャイルは、価値のフローを最適化し、人々の「活用方法」よりもむしろ顧客への迅速なフィーチャーの引渡しを強調している。

ヒント

意欲に満ちた人々を集めてプロジェクトを構成する。彼らに必要な環境と支援を与え、信頼して仕事をさせる。

チームが価値のフローを最適化する方法について考えるとき、次の利点に分かる。

- ◆ 人が協力しあう可能性が高い。
- ◆ チームは価値のある作業を優先して完了する。
- ◆ 同時に複数の作業を行うわけではなく、コンテキストを再確立する必要がないため、チームが無駄にする時間がずっと少なくなる。

4.3.1 アジャイル・チーム

アジャイル・チームは、フィードバックが得られるように、迅速なプロダクト開発に注力する。実際、最も効果的なアジャイル・チームの規模は、3人から9人となる傾向がある。理想的には、アジャイル・チームはチーム・スペース内の同一場所に配置される。チーム・メンバーはチームに100%専念する。アジャイルは、次の期間のスコープで誰がその作業を実行するかをチーム・メンバーが決定するという自己管理チームを奨励する。アジャイル・チームは、サーバント・リーダーシップの下で目標を達成する。リーダーは、作業に対するチームのアプローチを支援する。

機能横断のアジャイル・チームは、機能的なプロダクト増分を頻繁に生成する。それは、チームが共同で作業を行い、作業を完了するのに必要なスキルのすべてを備えているからである。

アジャイル・アプローチにかかわらず、チームが仕掛中の作業数を制限すればするほど、メンバーが協働して作業を迅速に処理する可能性が高くなる。成功しているアジャイル・チーム・メンバーは、協働作業の代わりにミニ・ウォーターフォールのわなに陥らないように、様々な方法（ペアリングやスウォーミング、モビングなど）で協働する。ミニ・ウォーターフォールは、チームが所定期間に**すべての**要求事項を対処し、設計の**すべて**を実行しようとし、そして**すべて**を開発しようとするときに発生する。このシナリオでは、開発時点または開発後のテスト時点で、それはもはや有効ではないことをチームが認識する。この場合、チームは要求事項の**すべて**に対処することで時間を無駄にした。逆に、チーム・メンバーが全面的に少数のフィーチャーを生成するために協働するとき、前進してより小さな完了したフィーチャーを引き渡すのに伴って学習する。

アジャイル・プロジェクトは、チーム内およびチーム間での協働を高めるプロジェクト・チーム構造からベネフィットを得る。表4-1は、協働的なチーム・メンバーが生産性を高め、革新的な問題解決を促進する方法を示す。

表4-1.成功するアジャイル・チームの属性

属性	ゴール
専任者	<ul style="list-style-type: none"> 注力と生産性の向上 10人未満の小さなチーム
機能横断チーム・メンバー	<ul style="list-style-type: none"> 頻繁な開発と引渡し 独立したチームとして完成した価値を創出する すべての作業活動を統合して完了した作業を引き渡す チーム内やプロダクト・オーナーなど他の人からのフィードバックを提供する
コロケーションまたは場所の課題をマネジメントする能力	<ul style="list-style-type: none"> より良いコミュニケーション 改善されたチーム・ダイナミクス 知識の共有 学習コストの削減 協働への約束ができる
ジェネラリストとスペシャリストの混合チーム	<ul style="list-style-type: none"> スペシャリストは専門知識を提供し、ジェネラリストは誰が何をするかについて柔軟性を提供する チームはそれぞれの専門能力を持ち寄り、注力する専門知識に加えて複数のスキルに渡る広範な経験を備えた汎化スペシャリスト集団になる
安定した作業環境	<ul style="list-style-type: none"> 引渡しのための相互依存 作業への合意された手法 簡素化されたチームのコスト計算（ラン・レート） 知的財産の保管と拡大

4.3.2 アジャイルにおける役割

アジャイルでは、次のような3つの共通の役割が使用される。

- ◆ 機能横断チーム・メンバー
- ◆ プロダクト・オーナー
- ◆ チーム・ファシリテーター

表4-2にこれらのチームの役割を記述する。

表4-2.アジャイル・チームにおける役割

役割	説明
機能横断チーム・メンバー	<p>機能横断チームは、作業対象プロダクトの生産に必要なすべてのスキルを備えたチーム・メンバーから構成される。ソフトウェア開発では、機能横断チームは一般的には設計者、開発者、テスター、およびその他必要な役割の担当者から成る。機能横断開発チームは、通常のカデッツでリリース可能なプロダクトを潜在的に提供できる専門家で構成される。機能横断チームは、完成した作業を可能な限り短時間に、高品質で、外部依存関係なしに提供できるため、重要である。</p>
プロダクト・オーナー	<p>プロダクト・オーナーは、プロダクトの方向付けに責任を負う。プロダクト・オーナーは、その事業価値に基づいて作業をランク付けする。プロダクト・オーナーは、プロダクトのフィードバックを提供し、開発および提供されるべき機能の次の部分に方向性を設定することにより、チームと日々協働する。その作業は小さく、しばしば1枚のインデックスカードに記載される程度に小さいことを意味する。</p> <p>プロダクト・オーナーは、ステークホルダー、顧客、およびチームと協力してプロダクトの方向性を定義する。一般的には、プロダクト・オーナーはビジネス上の背景を持ち、意思決定のために深い当該分野専門知識を持ち込む。場合によっては、プロダクト・オーナーは、アーキテクトのような深い専門知識や、プロダクト・マネジャーなど顧客に関する深い専門知識を持つ人に支援を求めることもある。プロダクト・オーナーは、チーム全体の作業の流れを体系化しマネジメントする方法についてのトレーニングを必要とする。</p> <p>アジャイルでは、プロダクト・オーナーはチームとともにチームのバックログを作成する。バックログは、チームがムダを出さずに最高の価値を創出する方法を理解するのに役立つ。</p> <p>アジャイル・チームにとって不可欠な成功要因は、強力なプロダクト・オーナーシップである。アジャイル・チームが、顧客にとっての最高の価値に注意を払わなければ、評価されていない、つまりあまり価値のないフィーチャーを作成してしまい、そのため取組みをムダにすることがある。</p>
チーム・ファシリテーター	<p>アジャイル・チームに通常見られる第3の役割は、チーム・ファシリテーター、すなわちサーバント・リーダーという役割である。この役割は、プロジェクト・マネジャー、スクラム・マスター、プロジェクト・チーム・リード、チーム・コーチ、またはチーム・ファシリテーターと呼ばれることがある。</p> <p>すべてのアジャイル・チームは、チームにサーバント・リーダーシップを必要とする。人は、ファシリテーション、コーチング、障害の除去というサーバント・リーダーシップ・スキルを身につけるために時間を必要とする。</p> <p>多くの組織は、当初内部のコーチング機能がまだ完全には開発されていないとき、外部のアジャイル・コーチを招聘する。</p> <p>外部コーチには経験の優位性はあるが、クライアント組織での人間関係が脆弱という欠点がある。一方、内部コーチは組織内で強い関係を築いているものの、自身を非常に効果的にしうる幅広い経験が不足している傾向がある。</p>

「I字型とT字型」

一部の人はひとつの領域に深い専門性を備えているが、その領域以外にはほとんど貢献しない。このような人は、アジャイル・コミュニティでは「I字型」として知られている。これは「I」文字のように深さはあるが幅がないからである。これとは対照的に、「T字型」はひとつの分野における専門知識を、関連分野ではあまり開発されていないスキルと優れたコラボレーション・スキルで補完する。たとえば、プロダクトのある分野をテストし、プロダクトの異なる分野を開発することができる人は、T字型とみなされる。

T字型の人は、定義され、認識されている専門性と主要な役割を備えているが、必要に応じていつでもどこでも他の人を助けるためのスキル、多用途性、コラボレーションの適性を持っている。このコラボレーションによって、ひとりしかその仕事ができない場合に起こる、引継ぎと制約条件が軽減される。

4.3.3 汎化スペシャリスト

アジャイル・チームは機能横断型ではあるが、しばしばそのようには始まらない。しかし、多くの成功したアジャイル・チームは、汎化スペシャリスト、すなわち「T字型」人材から構成されている。

このことは、チーム・メンバーはひとつの専門分野ではなく、注力される専門技能とともに複数のスキルにわたる幅広い経験を兼ね備えていることを意味する。アジャイル・チーム・メンバーは、協働して速やかに作業を達成するための緊密なコラボレーションと自己組織化によって、そのような特性を開発するために努力するが、それは日常的にお互いが助けあうことを義務付けている。

個人のスループットは関係ない。個人のスループットに焦点を当てるのが残りのチームにとってボトルネックとなる場合は有害となる。ゴールは、チームが完成した作業の引渡しを最適化してフィードバックを得ることである。

顧客が、優れた品質の迅速なフィーチャーの引渡しのような優れた結果を望む場合、チーム資源の効率を最大化するためにチームを専門的な役割を持った人だけで構成することはできない。チーム目標はフロー効率であり、チーム全体のスループットを最適化することである。小さいバッチ・サイズは、チームとして一緒に働くことを促進する。プロダクト・オーナーの仕事は、チームが最高の価値の作業に確実に取り組めるようにすることである。

4.3.4 チームの構造

多くの業界でチームはアジャイルの原則と実務慣行を採用してきた。メンバーを機能横断チームに編成して、作業プロダクトを反復的に開発する。



ケース

この実務慣行ガイドを書くために集められたコア・チームの背景はさまざまで、PMIを代表する人もいれば、アジャイル・アライアンスを代表する人もいた。彼らは自己組織化チームであり、作業を完了するために少しずつの増分で作業した。PMIは当該分野専門家グループを集めて作業を検査した。これによってチームはフィードバックを組み込むことができ、開発されたプロダクトを改善することができた。しかし、コア・チーム・メンバーの時間がこの試みに100%費やされたわけではなかったため、典型的なアジャイル・チームを代表するとは言えなかった。



一部の組織ではコロケーションされた機能横断チームを編成することができる一方で、そうでない組織もある。すべてのチーム・メンバーをコロケーションする代わりに、一部の組織ではチームは分散もしくは分裂される。分散したチームは、複数の場所に機能横断チームがいる。分裂したチームは、各チーム・メンバーがオフィス内や自宅など、個々が完全に異なる場所で作業する。これらの配置は、コミュニケーション費用の増加により理想的とは言えないものの、依然として機能しうる。

ある米国の大手金融機関に、米国東海岸とインド各地を拠点とするチーム・メンバーからなる複数のチームが関わるプログラムがあった。チームが最初にスタートした時、ひとつの大きな分散されたチーム（UX、アナリスト、開発者、テスター）だった。そこでは、友好的に作業引き継ぎをするためにチーム・メンバー全体で作業時間の一部が重複している「フォロー・ザ・サン」²開発の実務慣行を行った。チーム・メンバーは毎日一緒にスタンドアップ・ミーティングを行い、すべてのチーム・メンバーが参加できるようにウェブカメラを使用した。米国での主要な役割（アナリスト、プロダクト・オーナー、UXデザイナー、開発リーダーなど）は、インドに拠点を置くチーム・メンバーからの質問に早期に答え、障害の解決を支援した。

プロダクトが大きくなって、資金が増えるにつれ、5つの小さなチームに分けることに決めた。これを行うため、彼らはさまざまな場所にコロケーションし分散したチームを編成することに決めた。これらの各場所に、開発者とテスターから成る、機能横断のコロケーションされたチームを編成することを決定した。

また、米国内の2か所にアナリストのコア・メンバーが配置した。彼らは米国ベースのプロダクト・マネジャーとプロダクト・オーナーと協働し、その後各チームと協働した。プログラム全体としてプロダクト・レビューを実施した場所に組織化されたチームがいたが、他の活動のほとんどは、各チームに最適なものに基づいてチーム・レベルで実施され、自己組織化を可能にした。

4.3.5 専任チーム・メンバー

チーム・メンバーの時間がチームに100% 割り当てられていないとき何か起こるか。この状態は理想的ではないが、残念ながら避けることはできない。

誰かの25%か50%の時間しかチームに割り当てない場合に発生する重要な問題は、彼らがマルチタスクやタスク切替えを行うことである。マルチタスクは、チームの作業のスループットを低下させるため、チームが予測した引渡しの実施に影響がでる。

ヒント

マルチタスクは、チーム・メンバーが作業の切り替えや他の人の作業完了待ちなどで時間を無駄にするため、チーム全体の進捗が遅れる。メンバーが100% チームに専念している場合、チームは最大限迅速なスループットとなる。

²フォロー・ザ・サンの開発プロセスでは、プロダクト開発を加速するために、毎日の終わりにひとつのサイトから次のサイト、多くの場合地域が離れたサイトへ作業が引き継がれた。

タスクを切り替えるとき、20%から40%の生産性損失の可能性がある。損失は、タスクの数とともに指数関数的に増加する。

ふたつのプロジェクト間でマルチタスクするとき、それぞれのプロジェクトに50%が使えるわけではない。タスクの切替えのために、それぞれのプロジェクトにおいて20%から40%の損失が発生する。

マルチタスクをするとき、間違いを犯す可能性が高くなる。タスクの切替えにより記憶領域が消費されるため、マルチタスク時には人はタスクの背景を覚えにくくなる。

チームの全員がひとつのプロジェクトに100%割り当てられているときは、チームとして継続的に協働し、全員の作業をより効果的にすることができる。



ケース

この実務慣行ガイドを開発しているコア・チーム・メンバーは、彼らの100%の能力をチームの取組みに割り当てることができないので、そのスループットは、フルタイムでプロジェクトに専念できる場合よりも相当に低下する。しかし、たとえ分裂しその達成能力の一部で作業したとしても協働作業は経済的には実行可能だが、コロケーションし最大生産能力で注力することは実現できない。このため、チームは分裂を潜在的なリスクとして特定した。チームは、コラボレーション・ツールを使用して作業の進捗状況を追跡と監視し、適宜個人の能力に基づいて割り当てを調整した。



アジャイル環境のチーム、特にプロジェクト資源マネジメント知識エリアのプロセスに関する詳細は、「プロジェクトマネジメント・プロセス群と知識エリアのマッピング」の表A1-2を参照。

ヒント

必要とするすべての役割を、すべてのチームが持つわけではない。例えば、一部のチームはデータベース管理者やリサーチ・アナリストの支援を必要とする。チームが一時的にスペシャリストを任命するとき、誰もが同じ期待を持っていることを確認することが重要である。このスペシャリストはいつまでチームに100%割り当てられているだろうか。コミットメントのレベルを明確にしてチームが引き渡しできるようにするため、全員（スペシャリストとチーム）への期待値を設定する。兼任は、プロジェクトにリスクをもたらす。

4.3.6 チームの作業場所

チームは作業を協働し、チームとしてその状態を理解し、協力するための作業場所を必要とする。アジャイル・チームによっては、全員がひとつの部屋で作業することもある。スタンドアップ・ミーティングや図表のためにチームの作業場所があり、パーティション・スペースやオフィスで独自の作業をするチームもある。

企業はオープンで協働的な職場環境に移行しているが、思考や仕事が中断されない時間を必要とするチーム・メンバーのための静かなスペースを作り出す必要もある。したがって、企業は、個人が邪魔されることなく働くことができる静かなエリアまたは私的空間と、共有エリアを調整するようにオフィスを設計している（「Caves (洞窟)」や「コモン」と呼ばれることもある）。

チーム内に地理的に分散されたメンバーがいるとき、職場のどのメンバーがバーチャルで、どの程度が同じ場所にいるかを定める。ドキュメント共有、ビデオ会議、その他のバーチャルなコラボレーション・ツールなどのテクノロジーは、人々がリモートで協働するのに役立つ。

地理的に分散したチームは、バーチャルな作業空間を必要とする。さらに、定期的にチームを一ヶ所に集めることで、チームが信頼関係を築き、一緒に働く方法を学ぶように配慮する。

分裂したチームのコミュニケーションをマネジメントするのに考慮すべきいくつかの技法には、フィッシュボウル・ウィンドウとリモート・ペアリングがある。

- ◆ チームが分裂されているさまざまな場所の間で長期間のビデオ会議のリンクを設定することによって、フィッシュボウル・ウィンドウを作成する。メンバーは作業の開始時点でリンクを開き、終了時にリンクを閉じる。この方法で、メンバーはお互いを見て自発的に関わることで、地理的分離時に発生するコラボレーション・ラグを減らすことができる。
- ◆ 音声およびビデオ・リンクを含む、画面を共有するためのバーチャル会議ツールを使用することにより、リモート・ペアリングを設定する。タイム・ゾーンの違いが考慮されている限り、これは対面式ペアリングとほぼ同程度に効果的である。

ヒント

さまざまな部門からの異なるスキルを持つ人を集めてチームを編成する。マネジャーやリーダーをアジャイルのマインドセットについて教育し、アジャイル変革の早い段階で参画させる。

4.3.7 組織内サイロ化の克服

アジャイル・チームを形成する際の最善の出発点は、すべてのチーム・メンバーが同等の声をもち、傾聴され、配慮されることを確実にするため、基本的な信頼と安全な職場環境を構築することである。これにより、アジャイルのマインドセットが構築されるとともに、基本的な成功要因となり、他のすべての課題とリスクを軽減することとなる。

しばしば、サイロ化した組織は、機能横断型のアジャイル・チームを形成する上で障害となる。機能横断チームを編成するために必要なチーム・メンバーは、一般的には異なるマネジャーに報告し、マネジャーはメンバーのパフォーマンスを測定するために異なるメトリックスを使う。マネジャーは、資源効率性ではなくフロー効率性（およびチーム・ベースのメトリックス）に注力する必要がある。

組織のサイロ化を克服するために、これらのチーム・メンバーのさまざまなマネジャーと協力し、必要となる個人を機能横断チームに専念させる。これはチームの相乗効果を創出するだけでなく、人を活用することが開発中のプロジェクトやプロダクトをどのように最適化するか確認できる。

チームについての詳しい情報は、「付属文書X2 テーラリングに影響を及ぼす属性」を参照。

トピック

アジャイル・プロジェクトのリーダーとしては、第一に、機能横断的でしかも100%ひとつのチームに専念するチームをいかに編成するかに注力する。毎日仕事をして一緒にコミュニケーションを取るために開発者やテスターなど主要チーム・メンバーを得るだけであったとしても、それはアジリティへと向かう正しい方向の一歩となる。

5

アジャイルの実装： アジャイル環境での引渡し

5.1 プロジェクト憲章とチーム憲章

プロジェクト・チームが、なぜこのプロジェクトが重要か、チームがどこに向かっているのか、そしてプロジェクトの目標は何かを把握できるように、すべてのプロジェクトにはプロジェクト憲章が必要である。しかし、プロジェクト憲章そのものはチームにとって十分ではないかもしれない。アジャイル・チームには、チームの規範と、一緒に働く方法への理解が必要となる。その場合、チームにはチーム憲章が必要となる。

憲章の作成プロセスは、チームが協働し、プロジェクトを中心として融合する方法を学ぶのに役立つ。

少なくとも、アジャイル・プロジェクトでは、チームはプロジェクトのビジョンや目的、そして明確な一連の作業の合意を必要とする。アジャイル・プロジェクト憲章は、次の質問に対応する。

- ◆ なぜこのプロジェクトをするのか。それはプロジェクトのビジョンである。
- ◆ 誰が、どのようなベネフィットを得るか。それはプロジェクトのビジョンやプロジェクトの目的の一部である。
- ◆ このプロジェクトにとって「完了」とは何を意味するか。それはプロジェクトのリリース基準である。
- ◆ どのように協働するのか。それは作業フローで説明する。

サーバント・リーダーは憲章の作成プロセスをファシリテートする。チームは協力することで融合でき、プロジェクト憲章は作業を開始するための優れた方法となる。さらに、チーム・メンバーは、どのように協働できるかを理解するためにコラボレーションを望むかもしれない。

チームが協働する方法を理解する限り、憲章の作成のための正式なプロセスを必要としない。一部のチームは、チーム憲章の作成プロセスからベネフィットを受ける。チーム・メンバーが社会契約の基礎として使用するための憲章の作成時のアイデアがいくつかある。

- ◆ 持続可能なペースやコアタイムなど、チームの価値
- ◆ 「Ready(準備完了)」とは、チームが作業に携わることができるようになること、「Done(完了)」とはチームが一貫して完全性を判断できること、タイムボックスの尊重、または仕掛り中の作業数 (WIP) の制限の使用などの作業合意
- ◆ 会議では人が話しているときに遮らないなどの基本原則
- ◆ チームが会議時間をどのように使うかなどグループ規範

サーバント・リーダーはチームと共に、その他について決定する。

チームの社会契約（チーム憲章）とは、チーム・メンバーが互いにどのように対話するかであることを忘れてはいけない。チーム憲章のゴールは、チーム・メンバーがチームとして最高の能力を発揮できるアジャイル環境を作成することである。

5.2 一般的なアジャイル実務慣行

第5.2.1項から第5.2.8項は、最も一般的なアジャイル・プロジェクトの実務慣行について記述する。

5.2.1 レトロスペクティブ

最も重要な実務慣行は、レトロスペクティブである。なぜならば、それによってチームがそのプロセスを学び、改善し、適応させることができるからである。

レトロスペクティブは、チームがプロダクトとそのプロセスに関して以前の作業から学ぶ上で役立つ。アジャイル宣言の背後にある原則のひとつは次の通りである。「チームがもっと効率を高めることができるかを定期的に振り返り、それに基づいて自分たちのやり方を最適に調整します。」

多くのチームはイテレーション（特に2週間のイテレーション）を使用するが、それはイテレーションの最後でデモとレトロスペクティブを行うからである。しかし、チームはレトロスペクティブを行うためにイテレーションを必要とするわけではない。チーム・メンバーは、下記のような重要な時点でレトロスペクティブを実施することを決める。

- ◆ チームが何かのリリースや出荷を完了するとき。それはとても大きな増分である必要はない。どんなに小さくても、リリースであればよい。
- ◆ 前回のレトロスペクティブから数週間以上経過したとき。
- ◆ チームが行き詰まり、完成した作業がチーム内で留まっているとき。
- ◆ チームが他のマイルストーンに到達したとき。

チームは、暫定的なレトロスペクティブまたはプロジェクト終了時のレトロスペクティブのいずれかから、十分な時間をつかって学ぶことでベネフィットを得る。チームは、作業プロダクトと作業プロセスについて学ぶ必要がある。例えば、一部のチームは作業を完了するのに問題がある。チームが十分な時間を計画に入れる場合、データを収集し、処理し、検証として後で何を行うかを決定するために、レトロスペクティブを構成できる。

第一に、レトロスペクティブは責めたり非難したりするためのものではなく、チームが以前の作業から学び小さな改善を行うための時間である。

レトロスペクティブは、定性的データ（人の感情）と定量的データ（測定値）を調べ、そのデータを使って根本原因を見つけ、対策を立て、アクションプランを策定する。プロジェクト・チームは、障害を取り除くために結局多くのアクション項目に至ることがある。

今後のイテレーションまたは作業期間の改善に取り組むためにチームの能力に照らして、アクション項目の数を制限することを検討するとよい。一度に多くのものを改善しようとしてそのいずれも完了しないことは、より少ないアイテムを完了することを計画しそれらのすべてを正常に完了するよりも、はるかに好ましくない。そして時間があれば、チームはリストにある次の改善項目に取り組むことができる。チームが改善項目を選択した場合、成果をどう測定するかを決める。そして、次のイテレーションに、成果を測定して、各改善が成功したか失敗したかを確認する。

チームのファシリテーターは、各改善項目の重要度のランクを付けようとして指導する。改善項目がチームによってランク付けされると、チームは次のイテレーションのために適切な項目数を選択する（または、フロー・ベースの場合はフローに作業を追加する）。

5.2.2 バックログの準備

バックログは、チームのためにストーリー形式で提示された、すべての作業を順序付けたリストである。作業が開始される前にプロジェクト全体のすべてのストーリーを作成する必要はない。最初のリリースをおおまかに理解し、次のイテレーションのために十分な項目を理解するだけでよい。

プロダクト・オーナー（またはその分野のプロダクト・マネジャーとすべての関連プロダクト・オーナーから構成されるプロダクト・オーナー価値創造チーム）は、時間と共に期待される成果物の順序を示すためにプロダクトのロードマップを作成することができる。プロダクト・オーナーは、チームが生成するものに基づいてロードマップを再計画する。（ロードマップの例については「付録文書X3 アジャイル適合性フィルター・ツール」を参照。）

5.2.3 バックログの洗練

イテレーション・ベースのアジャイルでは、プロダクト・オーナーはしばしばイテレーションの途中で、ひとつ以上のセッション中に次のイテレーションのストーリーを準備するためにチームと協力する。これらの会議の目的は、ストーリーが何か、相対的にどの程度大きいかを理解できるように、ストーリーをチームが十分に洗練することである。

洗練作業の期間の合意はない。次のような連続性がある。

- ◆ フロー・ベースのアジャイルに対するジャストインタイムの洗練。チームは、次のカードを To-do(実行前)の列から取り出して議論する。
- ◆ 多くのイテレーション・ベースのアジャイル・チームは、2週間のイテレーションの途中でタイムボックス化された1時間の議論をする。(チームは、十分頻繁なフィードバックを提供するイテレーションの所要期間を決定する。)
- ◆ イテレーション・ベースのアジャイル・チームのための複数回の洗練議論。プロダクト、プロダクト領域、または問題のドメインを体験するのが初めてのとき、チームはこれを行う。

ヒント

インパクト・マッピングを使ってプロダクトが適合しているかを確認する。通常の場合では、プロダクト・オーナーがこの作業をリードする。サーバント・リーダーは、プロジェクトに役立つ方法として必要な会議をファシリテートすることができる。

洗練会議では、プロダクト・オーナーがストーリーのアイデアをチームに提示し、チームはストーリーの潜在的な課題や問題について学ぶことができる。プロダクト・オーナーが依存関係を把握していない場合、チームにリスクを理解させるためにフィーチャーをスパイクするよう要請することができる。

プロダクト・オーナーがバックログの準備および洗練会議を実行する方法はたくさんあるが、例えば次のような方法がある。

- ◆ チームが開発者、テスター、およびビジネス・アナリストかプロダクト・オーナーの3人組として議論し、ストーリーを書くことを奨励する。
- ◆ チームに全体的なストーリーの概念を提示する。チームは議論し、それを必要なだけ多くのストーリーに分けて洗練する。
- ◆ チームと協力してストーリーを調査し書くためのさまざまな方法を見つける。さらに、各ストーリーを十分に小さくすることで、完成した作業の安定したフローをチームが作成することを確実にする。最低限1日に1回はストーリーを完成できるようになることを考慮する。

チームはしばしば、1週間に1時間以上は費やすことなく、次のバッチ作業のためにストーリーを洗練するという目標を持っている。チームは、作業の計画にではなく、作業の実行に費やす時間を最大化したい。チームが週に1時間以上ストーリーの洗練に費やす必要がある場合、プロダクト・オーナーが準備をし過ぎている、またはチームには作業を評価し洗練するために必要な重要スキルが不足している可能性がある。

5.2.4 デイリー・スタンドアップ

チームはスタンドアップを使用して互いにコミットし、問題を明らかにし、作業がスムーズにチームの中を流れるようにする。

15分以上の長いスタンドアップをタイムボックス化しない。チームは何らかの形でカンバンまたはタスク・ボードを「行き来き」することができ、チームの誰もがスタンドアップをファシリテートすることができる。

イテレーション・ベースのアジャイルでは、全員が次の質問に答える。

- ◆ 前回のスタンドアップ以来、私は何を完了したか。
- ◆ 今から次のスタンドアップまでの間に、私は何を完了する予定か。
- ◆ 私の障害（またはリスクや問題）は何か。

このような質問に回答することで、チームが自己組織化し、前日まで、およびイテレーション中にコミットした作業を完了することにお互いに説明責任を持つようになる。

フロー・ベースのアジャイルはチームのスループットに重点を置いた、異なるアプローチでスタンドアップを行う。チームはボードを右から左に評価する。質問は次の通りである。

- ◆ この作業を進めるために我々は何をする必要があるか。
- ◆ ボードにないものに取り組んでいる人はいるか。
- ◆ チームとして何を完了させる必要があるか。
- ◆ 作業の流れにボトルネックやブロッカーがあるか。

典型的な落とし穴のひとつは、スタンドアップが状況確認会議になることである。従来、予測型環境で働いてきたチームは、状況を報告するのに慣れているため、この落とし穴に陥る傾向がある。

スタンドアップでよく見られる別の落とし穴は、問題が明らかになるとチームがその問題を解決し始めることである。スタンドアップは、問題を解決するためではなく、問題があることを認識するためのものである。課題をパーキング・ロット（保留したものを置くスペース）に追加して、スタンドアップ直後に別の会議を作成し、そこで問題を解決する。

チームは独自のスタンドアップを実行する。スタンドアップをうまく実行すれば、それは非常に有益である。また、チームの作業の性質が強いコラボレーションを要求する場合においても非常に有益である。チームがいつスタンドアップが必要か、またいつスタンドアップを効果的に使用できるかを意識して決定すること。

ヒント

チーム・メンバーがプロジェクト・マネージャーやリーダーの代わりにスタンドアップをファシリテートし、状況確認会議にならずに、チームが自己組織化してお互いにコミットするために使用することを推奨する。

5.2.5 デモンストレーションとレビュー

通常、チームがユーザー・ストーリーの形でフィーチャーを完成させると、その作業プロダクトを定期的にデモする。プロダクト・オーナーは、デモを見てストーリーを受託するか拒否する。

イテレーション・ベースのアジャイルでは、チームはイテレーションの最後に完了したすべての作業項目をデモする。フロー・ベースのアジャイルでは然るべきとき、通常は十分なフィーチャーがまとまったとき、チームは完了した作業をデモする。プロダクト・オーナーを含むチームは、プロダクトのフィードバック時期を決定するための意見を必要とする。

一般的なガイドラインとして、少なくとも2週間に1回はチームが持っている作業プロダクトをデモする。それはほとんどのチームにとって十分な頻度であるため、チーム・メンバーは誤った方向に向かうのを防ぐフィードバックを得られる。それはまた、希望する頻度、必要な頻度でプロダクトを完成させるために、チームがムダのないプロダクト開発を保持できるフィードバックの頻度である。

プロジェクトをアジャイルにするものの基本的な部分は、作業中プロダクトの頻繁な引渡しである。デモやリリースをしないチームは迅速に学習することができず、アジャイル技法を採用しているとは言えない。チームは、頻繁な引渡しを可能にするために追加のコーチングを要求することがある。

5.2.6 イテレーション・ベースのアジャイルの計画

各チームの能力はさまざまである。各プロダクト・オーナーが作成する典型的なストーリー・サイズは異なる。チームは、ストーリー・サイズを考慮して、1回のイテレーションで完成するストーリーがチーム能力よりも多くなるようにはコミットしようとはしない。

人が作業できない場合（休日、休暇、または次の作業に参加できないような事項）、プロダクト・オーナーは、チームの能力が低下したことを理解する。チームは、前回の期間に完了したのと同じ量の作業を終了することはできない。チームの能力が低下した場合、チームは、その能力に見合った作業のみを計画する。

チームは完了可能なものを見積もる。それが能力の尺度である（例については第4.10節を参照）。チームは、予想外のことを知ることはできないため、何を引渡しできるかを100%確実に予測することはできない。プロダクト・オーナーがストーリーを小さくし、チームが完成したプロダクトの形で進捗を見るとき、チームは将来何をすることができるかを学ぶ。

アジャイル・チームは、一度にすべてを計画するわけではない。そうではなく、アジャイル・チームは、継続的なサイクルで計画を少し立て、引き渡し、学習し、そして少し多く再計画する。

ヒント

落とし穴があることを示して、チームがスタンドアップを改善する方法を発見を助けること

5.2.7 チームの価値創出を助ける実務慣行の実施

チームが品質に注意を払わなければ、どんなものでも速やかにリリースすることはできない。

エクストリーム・プログラミングに由来する次の技術的実務慣行は、チームが最高のスピードで引き渡すのに役立つ。

- ◆ **継続的統合** プロダクトに関係なく、作業を全体に頻繁に組み込み、その後プロダクト全体が意図したとおりに動作するかどうかを確認するために再テストする。
- ◆ **全レベルでのテスト** エンド・ツー・エンド情報に対してはシステム・テスト、ビルディング・ブロックに対しては単体テストを行う。それぞれのテストにおいて、統合したテストを行う必要があるか、またそれをどこで行うかを理解する。作業プロダクトが良好かどうかを調べるスモーク・テストは有用である。チームは、回帰テストをいつ、どれを実行するかを決定することは、プロダクト開発パフォーマンスとプロダクト品質を維持するのに役立つことを見出した。アジャイル・チームは自動テストを好むため、予定された引き渡しの実現と維持が可能になる。
- ◆ **受入れテスト駆動型開発(ATDD)** ATDDでは、チーム全体が集まり、作業プロダクトの受入基準について議論する。そしてチームはテストを作成することで、チームは基準を満たす十分なコードと自動テストを書くことができる。非ソフトウェアのプロジェクトでは、チームがひとかたまりの価値を完成させたときそのテスト方法を検討する。
- ◆ **テスト駆動開発(TDD)と振舞い駆動型開発(BDD)** プロダクトの記述や作成を行う前に自動テストを書くことは、実際にプロダクトを設計し、間違いがないことを保証するのに役立つ。非ソフトウェアのプロジェクトでは、チームの設計を「試運転」する方法を検討する。ハードウェアやメカニカルプロジェクトではしばしば、そのデザインの間テストのためにシミュレーションを使用する。
- ◆ **スパイク (タイムボックス化された研究や検証)** スパイクは学習に役立ち、見積り、受入基準の定義、プロダクトを通じたユーザー行動フローの理解などの状況で使用できる。スパイクは、チームが技術的または機能的に不可欠な要素を学ぶ必要がある場合に役立つ。

5.2.8 イテレーションと増分が動くプロダクトの引渡しに役立つ方法

イテレーションは、チームが引渡しのカデンツと多くの種類のフィードバックを作成するのに役立つ。チームは、引渡しとフィードバックの価値の増分を生成する。この引渡しの第一の部分はデモである。チームは、デモを通してプロダクトがどのように見え、運用されているかのフィードバックを受け取る。チーム・メンバーは、成功するプロセスをどのように検査し、適応させるかを理解するためにレトロスペクティブを行う。

デモやレビューは、アジャイル・プロジェクトのフローに必要である。デモを、チームの引き渡しカデンツに合わせてスケジュールに入れる。

5.3 アジャイル・プロジェクトの課題のトラブルシューティング

アジャイル・アプローチは、変化の激しさ、不確実性、プロジェクトの複雑さに関連する課題を解決するニーズから生まれた。このような起源のため、アジャイル・アプローチには予測型アプローチの問題を提示する課題に対処するためのさまざまなツールや技法が含まれている。表5-1を参照。

ヒント

チームは、フィードバックと進捗を示すためにしばしばデモを行う必要がある。プロジェクト・ポートフォリオを決定する人々が実際の進捗状況を見ることができるよう、PMOおよび他の関係者にデモすることを奨励する。

表5-1.アジャイルのペインポイントとトラブルシューティングの可能性

ペインポイント	トラブルシューティングの可能性
チームの不明確な目的やミッション	ビジョン、ミッション、ミッション・テストなどの目的のためのアジャイル憲章の作成
チームの不明確な作業合意	価値、原則、作業同意などのアライメントのためのアジャイル憲章の作成
不明確なチーム・コンテキスト	範囲、コミットされた資産、将来の分析などのコンテキストのためのアジャイル憲章の作成
不明確な要求事項	スポンサーとステークホルダーがプロダクトのビジョンを立案する。例別の仕様、ユーザー・ストーリー・マッピング、インバクト・マッピングを使ってプロダクト・ロードマップの構築を検討する。チームとプロダクト・オーナーが共に期待と要求項目の価値を明確化する。ロードマップを経時的に小さい具体的な要求項目のバックログに細分化する。
貧弱なユーザー・エクスペリエンス	開発チームに含まれるユーザー・エクスペリエンス設計の実務慣行は、ユーザーが早い段階で頻繁にユーザーを関与させる。
不正確な見積り	ストーリーを分割してストーリーのサイズを縮小する。見積りには、チーム全体との相対見積りを使用する。ストーリーが何であるかを理解するために、アジャイル・モデリングまたはスパイクを検討する。
不明確な作業割当てや作業の進捗状況	チームが自分の作業を自分でマネジメントしていることを知るのに役立つ。作業の流れを見るのにカンバンを考慮する。ボードを行き来し、どの作業がどこにあるのかを見るために、毎日のスタンドアップを検討する。
チームは障害と取り組む	サーバント・リーダーはこれらの障害をクリアするのを支援できる。チームが持っている選択肢が分からない場合は、コーチングを検討する。場合によっては、チームやサーバント・リーダーが削除できなかったストーリーをチームがエスカレートする必要がある。
洗練が十分でないプロダクト・バックログ項目による作業遅延やオーバーラン	プロダクト・オーナーとチームがストーリーと一緒にワークショップする。ストーリーのための準備の定義を作成する。より小さなストーリーを使うためにストーリーを分割することを検討する。
欠陥	環境に適した技術的な実務慣行を検討する。いくつかの可能性として、ペアワーク、集成的プロダクト所有権、広範囲にわたるテスト（テスト駆動型技法と自動テスト技法）、そして頑強なDone(完了)の定義がある。
作業が完了していない	チームは、受入基準を含むストーリーのためのDone(完了)を定義する。また、プロジェクトのリリース基準を追加する。
技術的負債（コード品質の低下）	リファクタリング、アジャイル・モデリング、広範囲にわたるテスト、自動化されたコード品質分析、Done(完了)の定義

表5-1.アジャイルのペインポイントとトラブルシューティングの可能性（続き）

ペインポイント	トラブルシューティングの可能性
プロダクトの過剰な複雑さ	ソフトウェアおよび非ソフトウェアでは、チームに常に「最もシンプルに機能するものは何か」を考えるように奨励し、「シンプルさ（ムダなく作れる量を最大限にすること）」というアジャイルの原則を適用する。これらは複雑さを軽減するのに役立つ。
チームワーク・プロセスにおいて、遅いか、向上がない	各レトロスペクティブで改善すべき項目を3つ以上取り込まない。サーバント・リーダーに、チームがそれらの項目をどのように統合するかを学ぶのを助けるように依頼する。
前倒しすぎる作業は手直しにつながる	前倒しすぎる作業ではなく、学ぶべきチームのスパイクを検討する。さらに、プロジェクトの開始時にWIPを測定し、デザインではなく価値を提供するためにチームの選択肢が何かを確認する。イテレーションを短くして、頑強なDone(完了)の定義を作成する。
虚偽の始まり、無駄な努力	プロダクト・オーナーにチームの不可欠な役割を担うように依頼する。
非効率に並べられたプロダクト・バックログ項目	遅延コストを所要期間で割った値（CD3）やその他の価値モデルの値を含む価値でランク付けを行う。
「急ぐ」と「待つ」が不均一な作業フロー	チームの能力にあった、過剰ではない計画を立てる。マルチタスクをやめ、ひとつのチームに専念するよう要請する。全チームにわたる能力を均等化するために、チームにペア、スウォーム、またはモブとして作業するよう要請する。
ステークホルダーからの実現不可能な要求	サーバント・リーダーシップを発揮して、そのステークホルダー（おそらくプロダクト・オーナー）と協働する。
予期せぬ、または予測不能の遅延	チームやプロダクトへの要求の影響を理解するために、チームにもっと頻繁にチェックインし、カンバンを使って作業フローと仕掛中の作業数の制限を確認するよう要請する。また、障害とその除去を障害ボードで追跡する。
機能横断チームに代わるサイロ化されたチーム	プロジェクト・メンバーに機能横断チームとして自己組織化するように求める。サーバント・リーダーシップのスキルを使って、なぜアジャイルは機能横断チームを必要とするかをマネジャーが理解できるように支援する。

5.4 アジャイル・プロジェクトにおける測定

アジャイルへの移行は、異なる測定方法を使用することを意味する。アジャイルの使用は、チームと上層部にとって面倒となる新しいメトリックスを見ることを意味する。そのメトリックスは、顧客の価値に注力するために重要である。

状況報告における問題は、完了を予測したり、信号機を使用してプロジェクト状況を表すチームの能力である。たとえば、プロジェクト・リーダーはプロジェクトが「90%完了」と報告する。その時点で、チームは各ピースをプロダクトに統合しようとする。チームは、不足している要求事項を発見したり、驚いたりする。またはプロダクトが考えたようには統合されないことを知る。

プロジェクトは途中までしか完了しておらず、信号機を使った状況報告は実際の状態を反映していない。あまりに頻繁に、プロジェクト・チームは、プロジェクトの残りの部分を完了するために多くの時間が必要なことを認識する。ほとんどのプロジェクトでは、チームは問題を発見したために、せいぜい10%完了しただけだと考える。

予測型測定の問題は、多くの場合、現実を反映していないことである。リリース日の1か月前までは、プロジェクト状況を表す信号機が緑色に点灯していることがよくある。これはスイカ・プロジェクト（外側が緑、内側が赤）と呼ばれることもある。多くの場合、リリース日の1か月前までプロジェクトに関する経験値がないため、プロジェクトの状況表示が表面上突然赤色に変わったように見える。

アジャイル・プロジェクトは定期的に価値（完成した作業）を提供するので、アジャイル・プロジェクトのメトリックスには、過去の実績を提供する意味のある情報が含まれている。プロジェクト・チームは、このようなデータを改善された予測と意思決定のために使用することができる。

完成したフィーチャーなどの経験的測定値に比べ、完了率などの代用測定値はあまり役に立たない。価値マネジメントについての詳細は、第4.10節を参照のこと。アジャイルは、チームが問題と課題を把握して、それらを診断し対処できるようにするのに役立つ。

チームは定量的指標に加えて定性的指標の収集を検討することができる。これらの定性的指標の中には、例えば、提供されるフィーチャーに対するビジネス満足度、チームの士気、その他チームが定性的な尺度として追跡したいものなど、チームが選択した実務慣行に焦点を当て、それらの実務慣行をどれだけうまく利用しているかを評価するものもある。

5.4.1 アジャイル・チームが結果を測定する

アジャイルは、予測的測定ではなく、経験的で価値ベースの測定を好む。アジャイルは、チームが引き渡しすると予測しているものではなく、チームが引き渡したものを測定することである。

プロジェクトのベースラインやアード・バリューの見積り、ROIを持つことに慣れているチームは、プロジェクトに取り組むがベースラインをマネジメントしないことに困惑するかもしれない。アジャイルは、顧客に実証可能な価値のある作業プロダクトに基づいている。

ベースラインとはしばしば、予測の生成物である。アジャイルでは、チームは見積りを数週間先までに限定する。アジャイルでは、チームの作業にバラツキが少なく、チーム・メンバーがマルチタスクでない場合、チームの能力は安定する。これにより、数週間先までの予測がより良好になる。

チームがイテレーションまたはフローで作業を完了した後、チームは再計画することができる。アジャイルは、より多くの作業を行う能力は創出しない。しかし、作業の塊が小さければ小さいほど、それを引き渡す可能性が高いという証拠がある。

他の知的作業と同様に、ソフトウェア・プロダクトの開発とは、学習、すなわち価値を創出しながらの学習に関連する。ハードウェア開発とメカニカル開発はプロジェクトの設計部分が類似している。学習は、実証し、小さな増分の価値を引き渡し、それまでに達成したことに対するフィードバックを得ることによって行われる。他の多くのプロダクト開発プロジェクトにも学習が組み込まれている。

スポンサーは通常、プロジェクトがいつ完了するかを知りたがる。チームが信頼できるベロシティ（イテレーション当たりの平均ストーリーまたはストーリー・ポイント）または平均サイクル時間を設定すると、プロジェクトが今後どの程度の時間がかかるかを予測することができる。

例えば、チームの平均がイテレーションあたり50ストーリー・ポイントで、あと約500ポイントが残っていると見積もった場合、チームは約10回のイテレーションが残っていると見積もる。プロダクト・オーナーが残ストーリーを洗練し、チームが見積りを洗練するのに伴って、プロジェクトの見積りが上下することがあるものの、チームは見積りを提供することができる。

チームがストーリーあたり3日間のサイクル・タイムを平均とし、30ストーリーが残っている場合、チームは90営業日（約4～5か月）を残すことになる。

ハリケーン・スタイルの図表や、スポンサーが理解できるその他の変動性尺度を用いて、見積りの変動を反映させる。

学習はこのようにプロジェクトの大きな部分であるため、チームは不確実性を調整し、顧客に価値を提供する必要がある。チームは、プロジェクトの次の小さな部分を計画する。チームは経験的データを報告し、プロジェクトの不確実性をマネジメントするためにさらに小さな増分を再計画する。

一部のイテレーション・ベースのプロジェクトでは、プロジェクトが時間と共にどこに向かうかを確認するためにバーンダウン・チャートを使用する。図5-1は、チームが37ストーリー・ポイントの納入を計画したバーンダウン・チャートの例を示している。ストーリー・ポイントは、要求項目やストーリーの相対的な作業、リスク、複雑さを評価する。多くのアジャイル・チームは、取組みを見積もるためにストーリー・ポイントを使用する。点線のバーンダウン・ラインは計画を示す。図5-1では、チームは「3日目」によって、その引渡しにリスクがあることを確認できる。

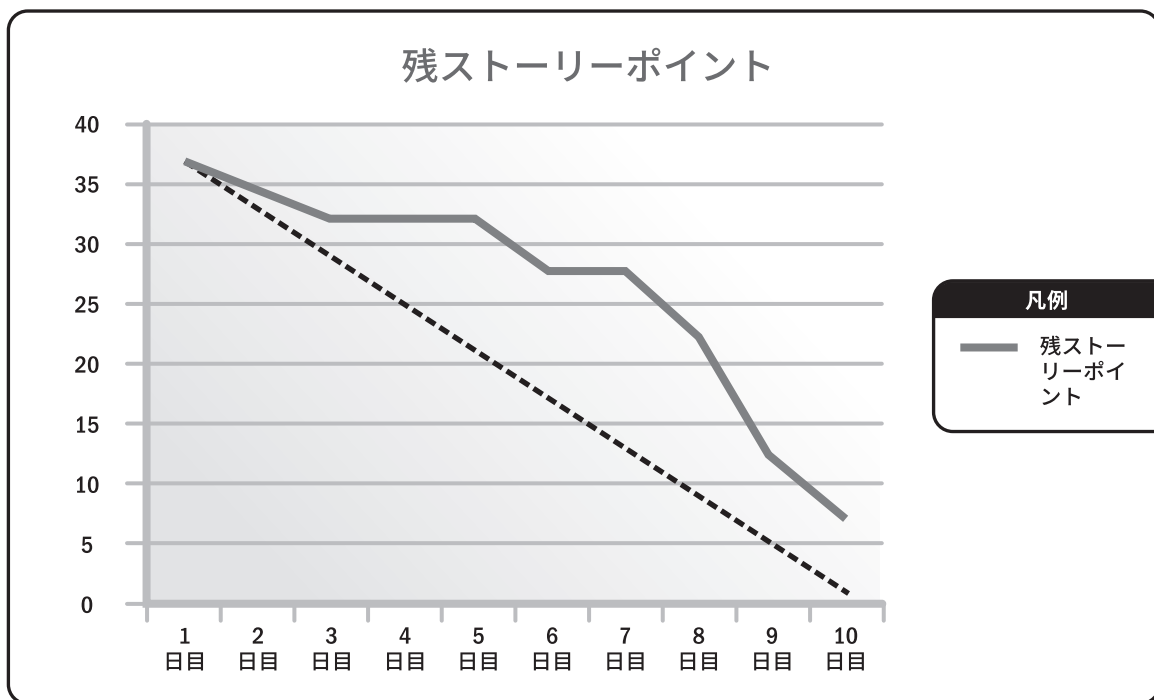


図5-1.残ストーリー・ポイントを示すバーンダウン・チャート

プロジェクト・チームの中には、バーンアップ・チャートを好むものもいる。図5-1で使用されたものと同じデータを図5-2のバーンアップ・チャートに示す。

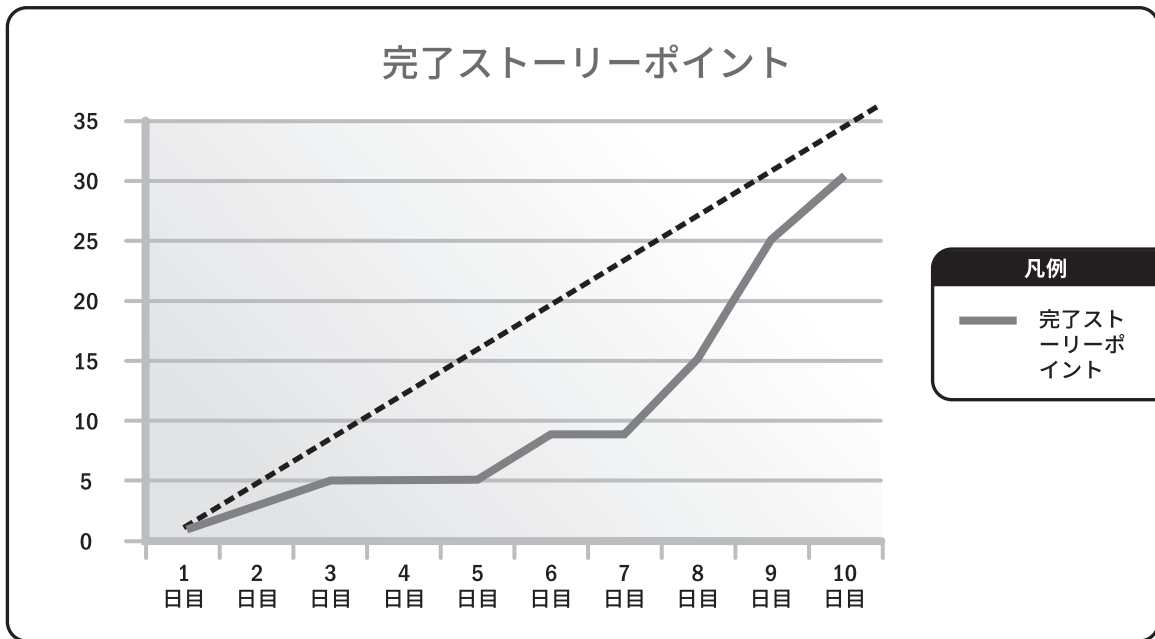


図5-2.完了ストーリー・ポイントを示すバーンアップ・チャート

バーンアップ・チャートは完了した作業を示す。図5-1と図5-2のふたつのグラフは同じデータに基づいているが、それぞれ異なる方法で表示されている。チームは、自分のデータを見る方法に好みがあるかもしれない。

チームが、イテレーションを介してまだ完了していないものを確認すると、チームは落胆し、おそらく受入基準を満たすことなく急いで作業を完了する可能性がある。しかし、チームが期待通りに作業を完了しなかったのには、いくつも妥当な理由がある可能性がある。バーンダウンは、チーム・メンバーのマルチタスクや、大きすぎるストーリー、またはチーム・メンバーの不在の影響を示す。

特に、アジャイルになじみがないチームでは、バーンアップはイテレーション中のスコープの変更を表示する。バーンアップは、何を達成したかをチームが見ることを可能にし、チームが次の作業ピースに進むのに役立つ。

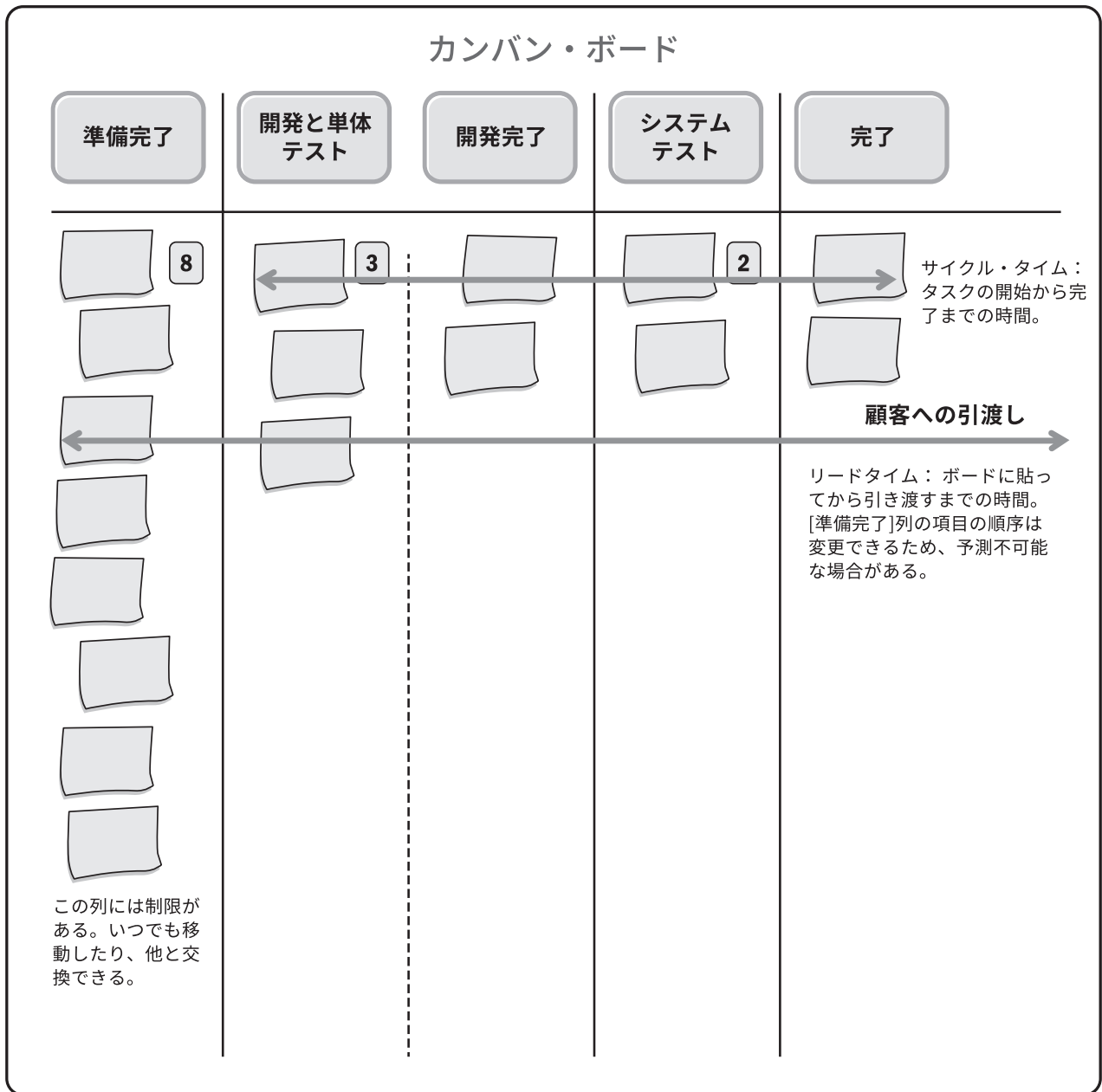
チームがバーンダウン・チャートかバーンアップ・チャートのいずれを使用しているかに関わらず、イテレーションが進行するのに従って完了したものを確認する。イテレーションの終了時点で、チームは、次回の達成能力の尺度（どのくらいのストーリーを完了できるか、すなわちストーリー・ポイント）に、このイテレーションで何を完了したかを基準として用いるかもしれない。これにより、プロダクト・オーナーはチームと共に、次のイテレーションでチームが引渡すことができるものを再計画する。

このイテレーションで実際に完了したフィーチャーのストーリー・ポイント・サイズの合計であるベロシティは、チームが過去のパフォーマンスの実績を利用することによって次の達成能力をより正確に計画できるようにする。

フロー・ベースのアジャイル・チームは、リード・タイム（ボードに追加された時点から完了した時点までの、アイテムの引渡しに要した合計時間）、サイクル・タイム（ひとつのアイテムを処理するために要した時間）、および応答時間（作業が開始されるまでアイテムが待機する時間）など、異なる測定値を使用する。チームは、サイクル・タイムを測定してボトルネックや遅延を確認するが、必ずしもチーム内ではない。

ヒント

チームは、安定したベロシティを達成するためには4~8回のイテレーションが必要なのが分かる。チームは、どのように作業し、どのように改善するかについて学習するために、各イテレーションからのフィードバックを必要とする。



サイクル・タイム：タスクの開始から完了までの時間。

リードタイム：ボードに貼ってから引き渡すまでの時間。
[準備完了]列の項目の順序は変更できるため、予測不可能な場合がある。

顧客への引渡し

図5-3.カンバンの例

リード・タイムは、特定フィーチャーを初めて見たときからそれを顧客にリリースするまでのサイクル時間を理解するのに有用である。このボックスに表示されている各列の上部にある仕掛り中の作業数（WIP）の制限により、チームは全体にわたって作業を移動させる方法を確認することができる。チームがWIP制限に至っている場合、チームは左から次の右隣の列に作業を移動することができない。代わりに、チームは一番右側の列から作業を行い「この作業を次の右隣の列に移動するにはチームとして何をするか」と考える。

各フィーチャーは一意であるため、サイクル・タイムも一意となる。しかし、プロダクト・オーナーは、フィーチャーが小さければそれだけサイクル・タイムも小さくなることに気づく。プロダクト・オーナーはスループットを確認したいため、より小さなフィーチャーを作成するか、またはチームと連携してその作業を行う。

バーンアップやバーンダウン（達成能力尺度）とリード・タイム、およびサイクル・タイム（予測可能性尺度）は、現時点での測定に有用である。それらは、あとどれだけ作業があるか、そして時間通りに終了できそうかをチームが理解するのに役立つ。

ストーリー・ポイントを測定することは、完成したストーリーまたはフィーチャーを測定することと同じではない。一部のチームは、実際のフィーチャーやストーリーを完成させることなくストーリー・ポイントを測定しようとする。チームがストーリー・ポイントのみを測定する場合は、完了した作業ではなく能力を測定することになり、それは「動くソフトウェアこそが進捗の最も重要な尺度です。」という原則に反する。

チームはそれぞれ固有の達成能力を持っている。チームがストーリー・ポイントを使用する場合は、所定時間内にチームが完了できるストーリー・ポイント数は、そのチームに特有であることに注意する。

ヒント

チームが外部の人やグループに依存している場合は、サイクル・タイムを測定して、チームが作業を完了するまでにかかる時間を確認する。リード・タイムを測定し、チームが作業を完了した後に外部依存関係を確認する。チームはまた、準備完了から最初の列（開発と単体テスト）までの時間、すなわち応答時間を測定し、新しい要求に対して応答するのに平均してどの程度かかるかを把握する。

チームが独自の尺度を提供する場合、チームは自分の作業を評価し、見積もり、引き渡すことがより可能になる。相対的見積りの欠点は、チームを比較したり、チーム間でベロシティを追加する方法がないことである。

チームは、フィーチャーのバーンアップやバーンダウン・チャート内で、およびプロダクト・バックログ・バーンアップ・チャート内で、完了した作業を測定することができる。これらのチャートは、図5-4に示すように、時間と共に完了の傾向を提供する。

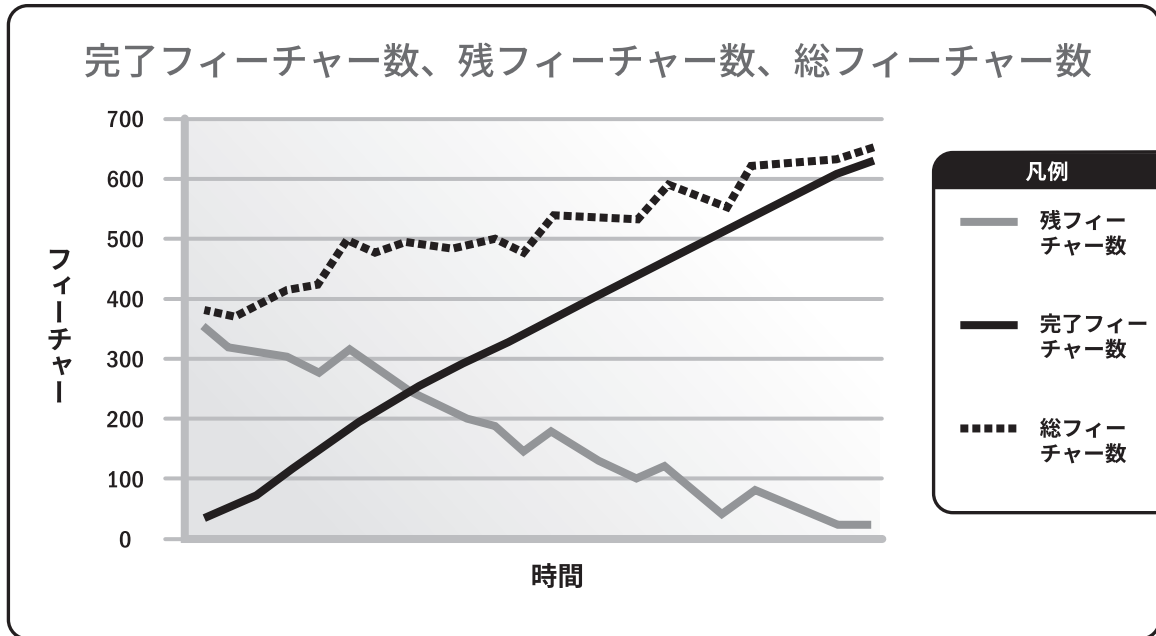


図5-4.フィーチャー図

フィーチャーのバーンアップ・チャートまたはバーンダウン・チャートは、要求事項がプロジェクトの間に追加されたことを示す可能性がある。完了フィーチャーの線は、チームが定期的なペースでフィーチャーを完了していることを示す。総フィーチャーの線は、時間と共にプロジェクトのフィーチャー数の合計がどう変化したかを示す。右肩下がりの残フィーチャーの線は、フィーチャーの完了率が変化していることを示す。フィーチャーがプロジェクトに追加されるたびに、右肩下がりバーンダウンの線が変化する。

アジャイルのアード・バリューは、図5-5に示すように、完了したフィーチャーに基いている。プロダクト・バックログのバーンアップ・チャートには、間隔マイルストーンまたはイテレーションで予想される作業合計と比較したときの完了した作業が表示される。

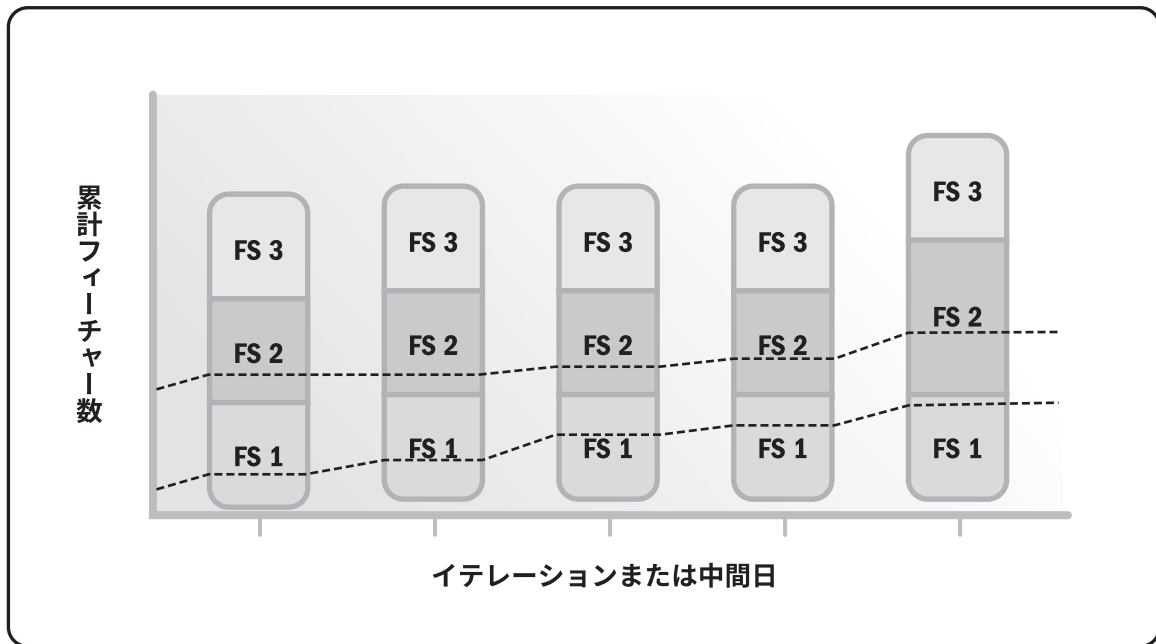


図5-5.プロダクト・バックログ・バーンアップ・チャート

チームは一度にひとつのストーリーしか終了できない。幾つかのストーリーを含んでいる大きなフィーチャーを完成するため、チームは完成すべき残りのストーリーを持っており、さらに何回が過ぎるまでは、その全部のフィーチャーを完成することができない。チームは、その完成した価値を、図5-5に示すように、プロダクト・バックログ・バーンアップ・チャートで表示できる。

チームがアーンド・バリューを測定する必要がある場合、例として図5-6のこのバーンアップ・チャートを使って検討できる。左の「Y」軸はスコープとしてのストーリー・ポイントを、そして右の「Y」軸はプロジェクト・スPENDを表示することに注意する。

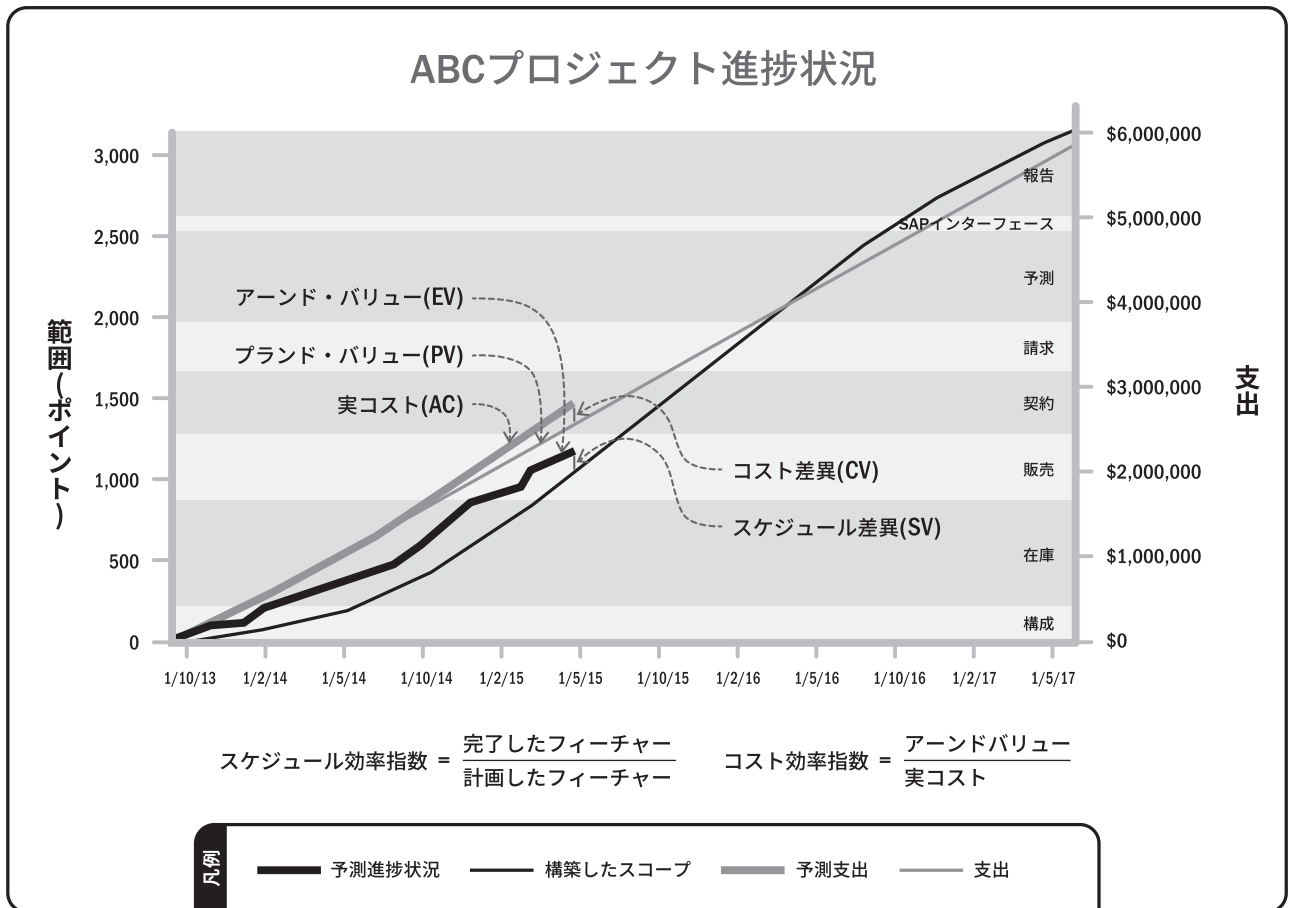


図5-6.アジャイル・コンテキストのアーンド・バリュー

スケジュール効率指数(SPI)やコスト効率指数(CPI)のような従来型のEVMメトリクスは、アジャイル用語に容易に翻訳することができる。例えば、チームがひとつのイテレーションで30ストーリー・ポイントを完成するよう計画したが、25しか完成しなかった場合、SPIは25/30すなわち0.83(チームは、計画した率の83%で作業している)となる。同様に、CPIは、今までのアーンド・バリュー(完成したフィーチャーの価値)をそれまでの実コストで割ったもので、図5-6に示すように $\$2.2\text{M} / \$2.8\text{M} = 0.79$ となる。これは、計画と比較した場合1ドルに対してたった79セントの結果を意味する(これは予測が正しいことを前提としている)。

図5-7に図示する累積フロー図は、ボード上の進行中の作業を示す。チームにテスト待ちのストーリーが多数ある場合、テストの列が膨張する。作業累積はひと目で分かる。

チームに完了作業ではなく仕掛り中の作業がある場合、チームは作業累積に問題がある。チームに仕掛り中の作業が多くあるときは、フィーチャーの引渡しが全般的に遅れる。チームが引渡しに時間がかかればかかるほど、同一期間中でのより多くのフィーチャーに対してチームはより多くのプレッシャーを感じる。

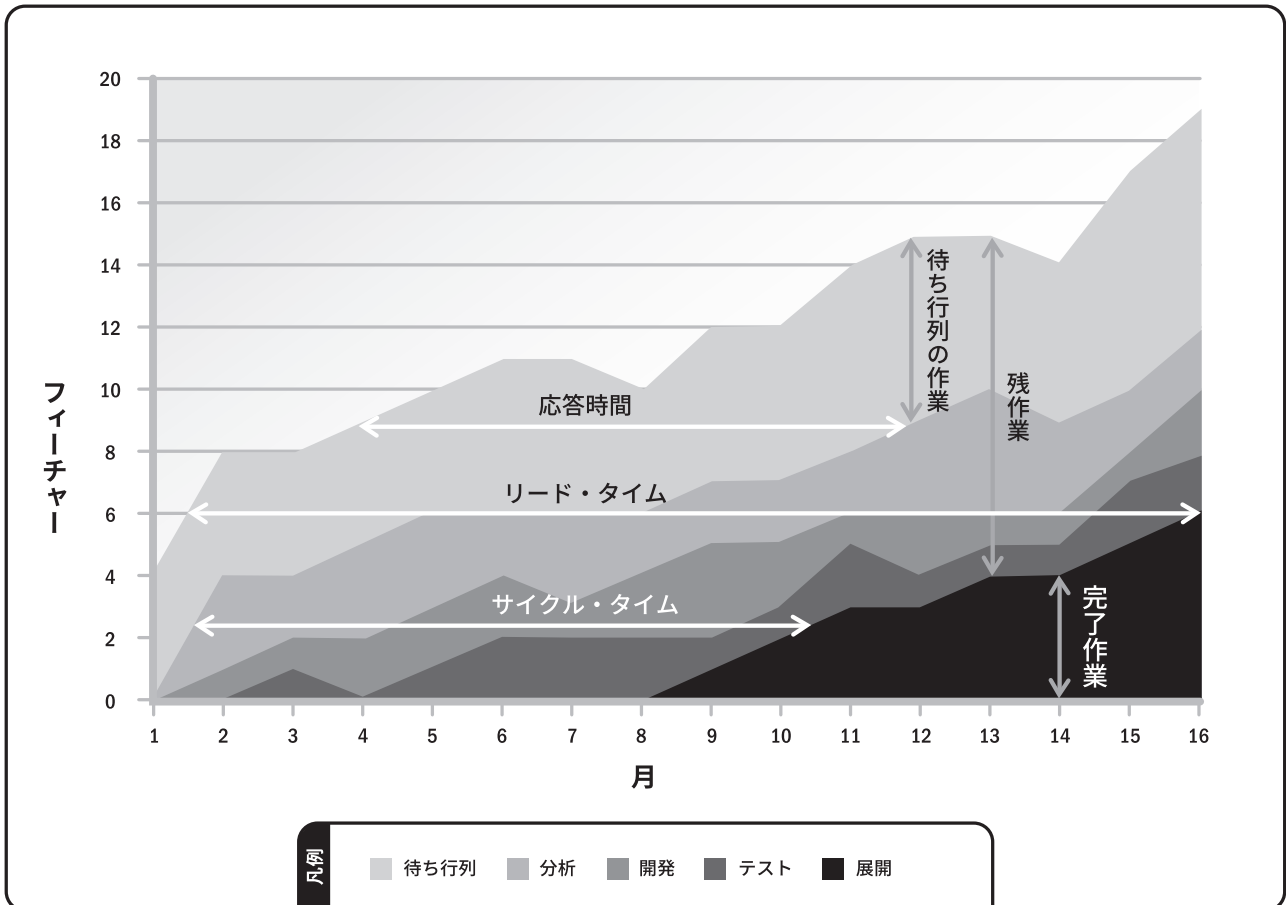


図5-7.完成したフィーチャーの累積フロー図

この累積フローをプロジェクトのタスク・ボードに適応する。

6

プロジェクトのアジリティに関する組織的な考慮事項

プロジェクトはすべて、組織環境の中に存在する。組織の文化、構造、方針は、プロジェクトの方向性と成果の両方に影響し得る。こうしたダイナミクスはプロジェクト・リーダーに課題をもたらすことがある。

プロジェクト・リーダーは、組織のダイナミクスを適切に変える能力を持たなくても、このようなダイナミクスをうまく操るよう期待される。

本章では、組織が、場合によってはプロジェクトの背景が、プロジェクトに影響を与える方法を探る。リーダーは、プロジェクトの成功を高めるために、変化の選択肢を探ることができる。

組織がサポートするように調整すると、プロジェクトのアジリティはより効果的になり、持続していく。

6.1 組織のチェンジマネジメント

組織のチェンジマネジメントには、アジリティをサポートする変更に影響を与えるためのスキルと技法が含まれる。

PMI刊行の『組織のチェンジマネジメント：実務ガイド』[2]は、有意義な変更をうまく導入するための包括的かつ全体的な手法を記述したものである。そこで提示されている推奨事項には次が含まれる。

- ◆ 変革のダイナミクスを表すモデル
- ◆ 変革を達成するためのフレームワーク
- ◆ プロジェクト、プログラム、ポートフォリオのレベルでのチェンジ・マネジメント実務慣行の適用。

第6.1.1項および第6.1.2項では、アジャイル環境に固有のチェンジマネジメントに関する考慮事項を探る。

図6-1に、これらふたつのトピックの関係を示す。

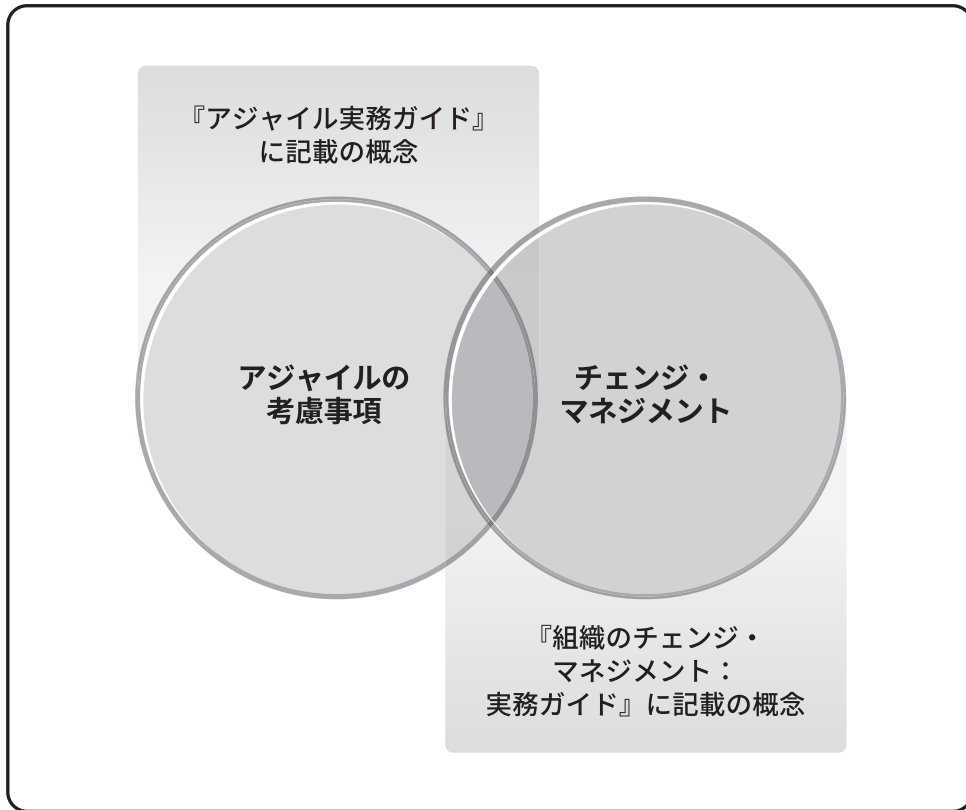


図6-1.チェンジマネジメントとアジャイル・アプローチの関係

6.1.1 チェンジマネジメントの駆動要因

すべてのプロジェクトは変革に関連する。しかし、アジャイル環境にはチェンジマネジメント実務慣行の使用をさらに動機づける、次のようなふたつの重要な要素がある。

- ◆ **引渡しを加速することに関連する変更** アジャイル・アプローチでは、プロジェクトの成果を早期にかつ頻繁に引き渡すことが重視される。しかし、受入れ側の組織では、加速されたペースでその成果を組み込む準備が十分にできていない可能性がある。引渡しの加速によって、その引渡しに応じる組織の能力が問われることになる。プロジェクトのフィーチャーをうまく発見し、提供するだけでは十分ではない。組織がプロジェクトのアウトプットにスムーズに応じなければ、目標とする投資対効果が遅れる。アジャイル環境では、顧客によるプロジェクト・アウトプットの受入れと、プロジェクト・アウトプットの整合性がますます重要になっている。
- ◆ **アジャイル・アプローチに関連する変更** アジャイル・アプローチを使用し始めたばかりの組織でも、大きな変革を経験する。より高いレベルのコラボレーションでは、チーム、部門、またはベンダー間のより頻繁なやりとりが必要になることがある。作業を複数の反復型プロトタイプに分割する場合、手戻りが伴うので否定的に見られがちになる。リーダーは、アジャイル・アプローチの使用への移行に伴う障害に対処する手段として、チェンジマネジメント技法を検討すべきである。

6.1.2 変革の準備

アジャイル・アプローチを使用し始めた組織は、そのアプローチと現在使用しているアプローチとの互換性を理解する必要がある。一部の組織は、部門間のコラボレーション、継続的学習、進化する内部プロセスといったアジャイルの原則を比較的容易にサポートするという特性を持っている。変革に対応する好ましいこれらの特性の例には次が含まれる。

- ◆ 経営上層部の変革への意欲
- ◆ 従業員を評価し、レビューし、査定する方法を変えていきたいという組織の姿勢
- ◆ プロジェクト、プログラム、ポートフォリオのマネジメント機能の集中化または分散化
- ◆ 短期的な予算編成とメトリックスを重視するか、長期的な目標を重視するかを選択
- ◆ タレント・マネジメントの成熟度と能力

逆に、組織のアジリティに関連する変革を達成する上で障害となる他の構造的な特徴もある。これらの例には次が含まれる。

- ◆ コンピテンシー・センターからの指導を得た機能横断チームではなく、縦割り組織に作業が分割されることにより、引渡しの加速化を妨げる依存関係がもたらされる。
- ◆ 調達戦略が長期的なコンピテンシーではなく、短期的な価格戦略に基づいている。
- ◆ リーダーの評価において、プロジェクト引渡しのエンド・ツー・エンドの流れや組織に関する全体的な最適化よりも、局所的な効率性が重視される。
- ◆ T字型スペシャリストを養成するのではなく、従業員はスキルの幅を広げるツールやインセンティブを制限されるため、特化したスキルしか持ちえない。
- ◆ 分散型ポートフォリオは、従業員を、ひとつのプロジェクトに集中させるのではなく、同時に多数のプロジェクトに参加させる。

組織にこれらの実務慣行を検討し修正する意欲がどの程度あるかによって、アジャイル・アプローチをいかに迅速かつ効果的に採用できるかが決まる。しかし、アジリティに伴うこれらの組織的な障害に応じて、プロジェクト・リーダーは、文化的互換性を加速させるために、次のようなさまざまなアプローチを試みることができる。

- ◆ 組織幹部による目に見える積極的な支援
- ◆ コミュニケーションやコーチングを含むチェンジマネジメント実務慣行
- ◆ 各プロジェクトへのアジャイル実務慣行の取入れ
- ◆ チームへのアジャイル実務慣行の漸進的な導入
- ◆ 適用可能なアジャイル技法と実務慣行を使用した、例示による指導

6.2 組織文化

組織文化はそのDNAであり、本質的なアイデンティティである。文化は常にアジャイル・アプローチの使用に影響を与える。組織文化は、予測可能性の高い計画からすべてが試行であるリーン・スタートアップに至る、ひとつの連続体に沿って進む。アジャイル・アプローチはリーン・スタートアップ文化にうまく適合するが、高度予測型の組織では、経験的測定や、小規模な実験や、学習を経てアジリティへと移行することができる。

6.2.1 失敗を恐れない環境づくり

組織文化を変えるのは難しいが、新しい方法や技法を積極的に試す組織では、最も重要な文化規範は失敗を恐れない環境を実現することである。

失敗を恐れず誠実で透明性の高い環境においてのみ、チーム・メンバーとリーダーは、プロジェクトの成功を真に反映してプロジェクトが進歩し続けることを確実にしたり、失敗したプロジェクトから学んだ教訓を適用して同じパターンに陥らないようにしたりすることができる。

6.2.2 文化の評価

すべてのプロジェクトには競合する願望の間で緊張関係がある。どうすれば品質を犠牲にすることなく速く進行することができるか。どうすれば期日を確実に守りながら柔軟性を維持することができるか。最も重要なことにどうすれば顧客の要求事項に応えつつ、満足してもらうことができるか。

プロジェクト・リーダーはすべてのステークホルダーのあらゆる期待に応えることが自分の仕事だと感じるかもしれない。しかし、どちらかを選択するように強いられた場合、組織のビジネス環境の文化や要求事項に応じて優先順位が決まることがよくある。例えば、モバイル通信プロジェクトは速度をより重視し、政府関連プログラムは汎用化と安定をより重視する。

「企業文化は戦略に勝る (Culture eats strategy for breakfast)」ピーター・ドラッカー

この言葉は、目的に対するコミットメントと情熱の重要性を強調している。あなたがチームでどのような戦略や計画を実行しようと、計画を実行する人々が成功の鍵となる。戦略を実施する人々が変革に情熱を持っていなければ、もっと言えば自分の仕事や組織に熱意がないのであれば、変革が実現する可能性はほとんどない。

これらのダイナミクスをうまくこなすため、プロジェクト・リーダーは組織内で最も頻繁に重要視される箇所を評価するように時間をかける必要がある。評価がどのようなものを図6-2に示す。この例では、プロジェクト・リーダーが、ステークホルダー、チーム・メンバー、およびシニア・マネジメントと組織の優先事項について会話を開始する。これらの優先順位は、両極間のスライド式スケール上の位置として記録される。結果は、それらの優先順位に最も適合するアジャイル・アプローチを見つけるために使用される。

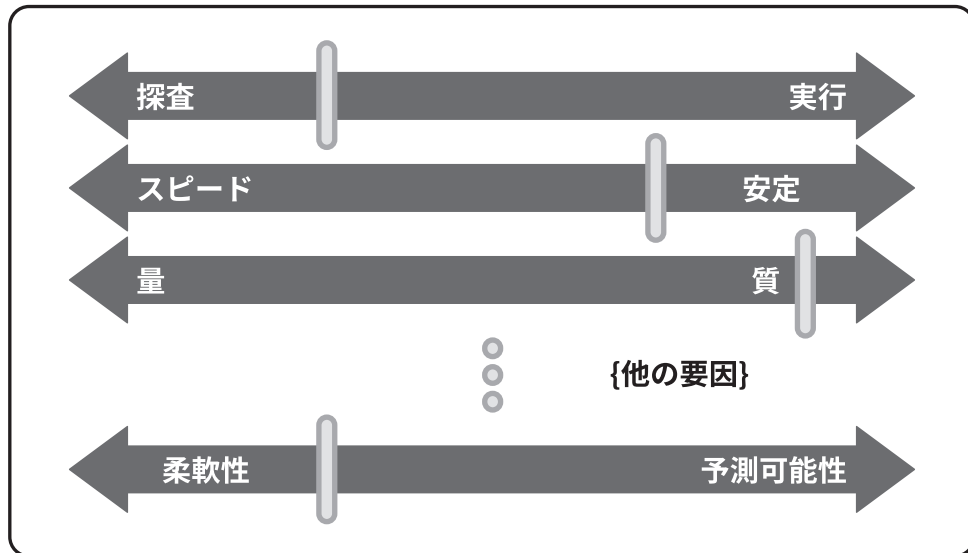


図6-2.組織文化の評価例

このようなダイナミクスを評価するためにいくつかのモデルが存在するが、使用されるモデルや方法はさほど重要ではない。それぞれの状況を生む根源は何であるかをプロジェクト・リーダーが把握する方がより重要である。組織が満たすべき、業界および組織の要求事項を理解することで、適切な会話、適切なトレードオフ、特に適切な技法を選択することが可能になる。

6.3 調達と契約

この実務ガイドで以前述べたように、アジャイル宣言は「契約交渉よりも顧客との協調」により価値を置いている。プロジェクトの失敗の多くは、顧客とサプライヤーの関係破綻に起因する。契約に携わる者が勝者対敗者という視点を持つと、プロジェクトはより多くのリスクを招く。協動的アプローチでは、すべての側が勝者となる、リスクと報酬の共有関係を追求する。このダイナミクスを正式化できるいくつかの契約技法には、次のものが含まれる。

- ◆ **契約構造の多層化** プロジェクトの当事者は、契約関係全体を単一の文書で正式化するのではなく、異なる文書でさまざまな側面を説明することで、より柔軟性を達成することができる。保証、仲裁などのほとんどの固定項目は、基本契約で規定することができる。その一方で、付則内に変更対象項目（例えば、サービス料金、プロダクト説明）を記載する。基本契約内でこれら付則の項目を参照できる。更に、スコープ、スケジュール、および予算などのより変更されやすい項目は、簡易な作業記述書で正式化することが可能である。契約の変化する要素を別の文書に分離すると修正が簡単になり、柔軟性も向上する。
- ◆ **提供価値の強調** 多くのベンダーとの関係は、事業価値を高める完全な成果物ではなく、中間成果物に焦点を当てた固定マイルストーンまたは「フェーズ・ゲート」により統制される。こうしたコントロールにより、プロダクトを改善するためのフィードバックの使用がしばしば制限される。そうではなく、価値駆動型の成果物に基づいてマイルストーンや支払条件を構成することにより、プロジェクトのアジリティを高めることができる。
- ◆ **漸進型の定額契約** プロジェクトの全スコープと予算をひとつの契約のみに制限する必要はない。ユーザー・ストーリーなどの定額の極小成果物にスコープを分解することができる。これにより顧客は、お金の使い方をより詳細にコントロールすることができる。サプライヤーは、オーバーコミットメントの財務リスクを単一のフィーチャーまたは成果物に限定することができる。

文化対構造

ヒント

組織文化を変える前に、組織構造を新しくするべきだという主張がある。一方、組織文化が意味のある方向に動くまでは、組織構造を新しくすることは表面的な調整に過ぎないという主張もある。実際には、一方のみを進めることはできない。アジリティを達成したいプロジェクト・リーダーは、組織内のこれらふたつの側面の現在および将来の状態を考慮する必要がある。

- ◆ **上限付きタイム・アンド・マテリアル** 従来型のタイム・アンド・マテリアル手法では、顧客は望ましくないリスクを負っている。その対策のひとつは予算全体を一定額に制限することである。これにより、顧客は当初予定されていないプロジェクトに新しいアイデアやイノベーションを取り入れることができる。顧客が新しいアイデアを取り入れる際には、既存の達成能力の中でマネジメントして、元の作業を新しい作業と置き換えなければならない。割り当てられた時間を超えないよう、作業を厳密に監視する必要がある。また、有用と考えられる場合には、追加の予備時間を予算内で計画することができる。
- ◆ **段階的タイム・アンド・マテリアル** もうひとつの選択肢として、財務リスク共有手法がある。アジャイルでは、品質基準とは「Done(完了)」が意味するものの一部である。したがって、サプライヤーは、引渡しが契約締切り日より早い場合には、より高い時間単価で報酬を受けることができる。逆に、引渡しが遅れると、サプライヤーの時間単価が低くなる。
- ◆ **早期キャンセルのオプション** アジャイル・サプライヤーが、スコープの半分しか完了していない状態で十分な価値を既に提供した場合、顧客が必要としないならば、残り半分を支払う義務はない。代わりに、顧客がキャンセル料を支払って、プロジェクトの残りをキャンセルする契約とすることができる。顧客は予算の浪費を抑え、サプライヤーは必要とされないサービスに対して収入を得る。
- ◆ **ダイナミック・スコープ・オプション** 予算が固定されている契約の場合、サプライヤーは顧客に対し、プロジェクトの指定された時点でスコープを変更するオプションを提供することができる。顧客は達成能力に合わせてフィーチャーを調整できる。そして、サプライヤーによるオーバー・コミットメントのリスクを限定しながら、イノベーションの機会を活用することができる。
- ◆ **チームの増強** 最も協調的な契約手法はおそらく、サプライヤーのサービスを顧客の組織に直接埋め込むことである。資金調達チームは、特定のスコープではなく実際にどの作業をすべきかに関する顧客の戦略的裁定権を持つ。

- ◆ **フルサービス・サプライヤーへの傾倒** リスクを分散させるために、顧客はマルチサプライヤー戦略を模索する。しかし、各サプライヤーがどうしてもひとつのことだけを行うように作業を請け負うことになり、利用可能なサービスやプロダクトが出現する前に依存関係が形成されてしまう。代わりに、完成した独立フィーチャー・セットの考え方のように、完全な価値を提供する関わり合いに重点を置くようにする。

アジャイル契約を作成することは可能である。アジャイルは、協調と信頼のシナジーに基づいて構築される。サプライヤーは、価値を早期に、かつ頻繁に引き渡すことによって支援することができる。顧客はタイムリーにフィードバックを提供することによって支援できる。

6.4 実務慣行

ニーズに応じて組織内で新しいコンピテンシーを創出する意欲と能力は、組織的アジリティの証しである。これらは、あっと驚くような変化である必要はなく、アジリティとその成果に重点を置いている組織では、さほど破壊的にならないこともある。透明性とオープンなコラボレーションが絶対的に重要になる。

機能横断チームが価値を実現するにあたり、チームと個人は組織内の様々なサポート機能に関連する問題に遭遇する可能性がある。

チームが定期的に価値を提供するとき、財務部門はプロダクトを異なる方法で資本化する機会を持つ。チームが他の組織と契約を結んでいる場合、調達部門はこれらの契約を変更して、他の組織が頻繁に価値を実現し、チームと同期するのを助ける必要がある。

チームが団結して協力的に働き始めると、内部マネジメント方針を変更する必要がある。人事部は個人奨励金があまり意味を持たないことに気付くほか、マネジャーは自己組織的な従業員の業績評価に苦勞するかもしれない。それぞれのケースで、これらは既存の実務慣行にてアジャイルな働き方をどの程度サポートすべきかを再検討する機会となる。

組織がアジリティを高めるにつれて、より多くの事業部門がコミュニケーションを取り合って責任を果たす方法を変えなければならなくなる。組織の他の領域に役立った変革が受け入れられれば、組織全体の有効性に役立つはずである。

6.5 複数チームの調整と依存関係（スケーリング）

多くのプロジェクトでは、既存のプログラム内でマネジメントされていない場合でも依存関係が発生する。したがって既存のプログラムマネジメントやポートフォリオマネジメントの環境でアジャイルがどう機能するかを理解する必要がある。

6.5.1 フレームワーク

スクラムやエクストリーム・プログラミングのような最も広く普及しているアジャイル・アプローチはコロケーションされた単一の小さな機能横断チームの活動に重点を置く。これは単一のチームを必要とする作業には非常に有用だが、プログラムやポートフォリオ内で複数のアジャイル・チームのコラボレーションが必要な施策については、ガイダンスが不十分な可能性がある。

そのような状況に対応するため、さまざまなフレームワーク（スケールド・アジャイル・フレームワーク（SAFe）、大規模スクラム、ディシプリンド・アジャイル）とアプローチ（例：スクラムのスクラム）が登場した。これらについての詳細は、付録A3に記載されている。

6.5.2 考慮事項

作業をスケールアップする方法は複数ある。チームは、いくつかのアジャイル・プロジェクトの作業を単一のアジャイル・プログラムにスケールアップしなければならない場合がある。一方、組織はポートフォリオ全体にわたってアジャイル・アプローチをサポートする構造を設計することもできる。

例えば、小さく始め、組織の環境でうまく機能するものをできるだけすばやく学ぶとよい。チームは、必ずしもすべてが完全にアジャイル・アプローチに変わってなくても、成果を上げることができる。

手法が何であれ、重要な成功要因はアジャイル・チームが健全であることである。ひとつのチームにアジャイル・アプローチを使用しても成功しない場合は、より広範に使用するようにスケールアップしようとするよりも、むしろチームがアジャイルな方法を作業することを妨げている組織上の障害に対処すべきである。

大規模なアジャイル・プロジェクトは、さまざまなチームの取組みを調整して顧客に価値をもたらすことを目標とする。それを行う方法はひとつだけではない。チームは、正式なフレームワークを使用したり、アジャイル思考を適用したりして、既存のプログラムマネジメントの実務慣行を調整することができる。

6.6 アジャイルとプロジェクトマネジメント・オフィス(PMO)

PMOは、組織全体の事業価値を守るために存在し、プロジェクトの目標達成をサポートする。PMOはチームを教育（またはトレーニングを手配）し、プロジェクトを支援することもある。PMOは、ときには既定のプロジェクトまたは一連のプロジェクトの相対的な事業価値についてマネジメントにアドバイスする。

アジャイルは時間と共に文化の変化を生み出すので、PMOを含めて組織も変革する必要がでてくるかもしれない。例えば、マネジャーはどのプロジェクトにいつ資金調達するかを決定し、チームはトレーニングやアドバイスに必要なものを決める。

6.6.1 アジャイルPMOは価値駆動

いかなるプロジェクトも、適切な時に、適切な価値を、適切な関係者に提供する必要がある。PMOの目的はこの目標を促進し、実現することである。アジャイル・ベースPMOのアプローチは、顧客とのコラボレーションというマインドセットに基づいており、すべてのPMOプログラムに存在する。多くの場合、PMOはコンサルティング事業であるかのように機能し、既定のプロジェクトで特定ニーズを満たすために取組みを調整することを意味する。ツールやテンプレートを必要とするプロジェクトもあるし、組織幹部の指導からベネフィットを受けるプロジェクトもある。PMOは必要とされるものを提供し、常に顧客のニーズを理解し適合できるように努めなければならない。この企業内起業家アプローチは、サポートするプロジェクトにとって最も価値があると認識されているPMOアクティビティに焦点を当てる。

6.6.2 アジャイルPMOは招集指向

価値観に基づく憲章での進展を加速化するため、PMOは、例えば短期的に成果を得るために全員に同じ方法で行わせるなど、特定のソリューションやアプローチを強制したくなる場合がある。しかし、よく考えると、従業員が関与したいという思いを汲み取らなければならない。これは、PMOサービスに関与することに関心を持つ人のみを招集することで実現される。PMO実務慣行との関わり合いが強くなればなるほど、その慣行自体が持続する。PMOがクライアントに価値を提供している場合、クライアントはそうしたサービスを要求し、それに伴う実務慣行を採用する可能性が高くなる。

6.6.3 アジャイルPMOは学際的

プロジェクト固有のニーズをサポートする上で、PMOは異なるプロジェクトには異なる能力が必要になるため、プロジェクトマネジメントそのものを超えたいくつかのコンピテンシーに精通している必要がある。例えば、あるプロジェクトでは人材育成の課題に取り組むために組織設計が必要となったり、別のプロジェクトではステークホルダーの関与を得るために組織のチェンジマネジメント技術が必要だったり、顧客の目標をサポートする独自のビジネスモデルが求められたりする場合がある。

組織によっては、PMOを次のようなサービスを提供するアジャイルCoE（組織横断的専門集団）としているところもある。

- ◆ **標準の開発と実施** ユーザー・ストーリー、テスト・ケース、累積フロー図などのテンプレートを提供する。アジャイル・ツールを提供し、反復型開発コンセプトに関するサポート・グループを教育する。
- ◆ **トレーニングとメンタリングによる人材育成** 従業員がアジャイルのマインドセットに移行し、スキルを向上させるためのアジャイルのトレーニングコース、コーチ、およびメンターを手配する。従業員が現地のアジャイル・イベントに出席するよう奨励し、支援する。
- ◆ **マルチプロジェクトのマネジメント** プロジェクト間のコミュニケーションを取ることで、アジャイル・チームの間の調整を図る。進捗、課題、レトロスペクティブでの気づきや改善の試みなどの共有を検討する。適切なフレームワークを使用して、プログラム・レベルでの主要顧客のリリース、およびポートフォリオ・レベルでの投資テーマのマネジメントを支援する。
- ◆ **組織としての学習の促進** プロジェクトのベロシティ・プロフィールを収集し、レトロスペクティブでの気づきを収集し、保存し、検索できるようにする。
- ◆ **ステークホルダーのマネジメント** プロダクト・オーナー向けトレーニング、受入れテストについてのガイダンス、およびシステムの評価やフィードバックの方法を提供する。プロジェクトの当該分野専門家（SME）の重要性を支持する。
- ◆ **チーム・リーダーの採用、選定、評価** アジャイル実務者にインタビューするためのガイドラインを作成する。
- ◆ **プロジェクトのための特殊タスクの実行** レトロスペクティブ・ファシリテーターを養成して供給し、アジャイル・プロジェクト・トラブルシューターとの合意を形成し、メンターおよびコーチを供給する。

6.7 組織構造

組織の構造は、新しい情報や変化する市場ニーズにシフトする能力を大きく左右する。主要な特徴を次に示す。

- ◆ **地理** 地理的に分散したプロジェクト組織には、プロジェクトでの作業を妨げるいくつかの課題がある。プロジェクト・リーダーや地域マネジャーは、異なる目標、場合によっては競合する目標を持っている場合がある。さらに、文化的な違い、言葉の壁、低い可視性は生産性を鈍化させかねない。幸いなことに、アジャイル・アプローチの使用によって、より多くのコラボレーションと信頼が促進される。このような環境にあるプロジェクト・リーダーは、状況に合わせて技法をテラリングし、そのために必要な取り組みについての期待をマネジメントするために、チームおよびエグゼクティブ・レベルで対話を促す必要がある。
- ◆ **機能化された組織構造** 組織の中には、高度なプロジェクト型から、マトリクス型、さらには高度な機能型までのスペクトル上に構成されているものもある。高度な機能型構造を持つプロジェクトには、組織全体にわたるコラボレーションに対する抵抗が見られる場合がある。
- ◆ **プロジェクト成果物のサイズ** プロジェクト成果物のサイズを縮小することは、部門間のより頻繁なやりとりを促進する。これにより頻繁なコミュニケーションや、組織全体にわたる価値の迅速な流れを促す。
- ◆ **プロジェクトへの人材の配置** もう一つ別のアプローチでは、各部門から一個人に依頼して、一時的ではあるが最優先プロジェクトに専任として配置する。
- ◆ **調達重視の組織** 一部の組織は、主にベンダーを通してプロジェクトを実施している。プロジェクト目標は明らかかもしれないが、ベンダーは自ら財務上の実行可能性を考える責任がある。さらに、いったんベンダーが義務を果たして関与がなくなると、関連するプロジェクト知識も失われてしまうため、柔軟性とスピードを持続する上で必要な社内コンピテンシーが制限されてしまうレトロスペクティブや、ベンダーがまだ関与しているときの改善可能領域のフォローアップなどといったアジャイル技法は、プロダクト知識の損失を軽減するのに役立つ。

6.8 組織の発展

個々の困難な領域に対処したり、新しいハイブリッド・アプローチやアジャイル・アプローチを実施する際は、漸進的に作業を進めることが推奨される。目に見える価値やその他の考慮事項に基づいてチームが変更プロセスを導入し優先順位を付けることが可能な独自の変更バックログを持つアジャイル・プロジェクトとして扱うことが、一般的な実務慣行である。変更はそれぞれ実験として扱うことができ、そのままで適合するか、あるいはさらなる微調整や検討の必要があるかを判断するために短期間でテストされる。

「Done (完了)」では既に使用されている新しいアプローチを、「In Progress (実行中)」では現在試行されているものを、「To Do (実行前)」では取り込まれるのを待っているものを示すカンバンを使って進捗状況を追跡する。ランク付けされたバックログを持つ最初のカンバンについては、図6-3を参照のこと。作業の進行に伴ってカンバンに表記される例を図6-4に示す。

ランク付き バックログ	実行中		リスク： マネジメン トが軽減	事後対応の 決定	待機：手詰 り項目	完了
	アクション 項目：分析	アクション 項目：解決				
変更 1						
変更 2						
変更 3						
変更 4						
変更 5						
変更 6						
変更 7						
変更 8						
変更 9						
変更 10						

図6-3.初期にランク付けされた変更のバックログ

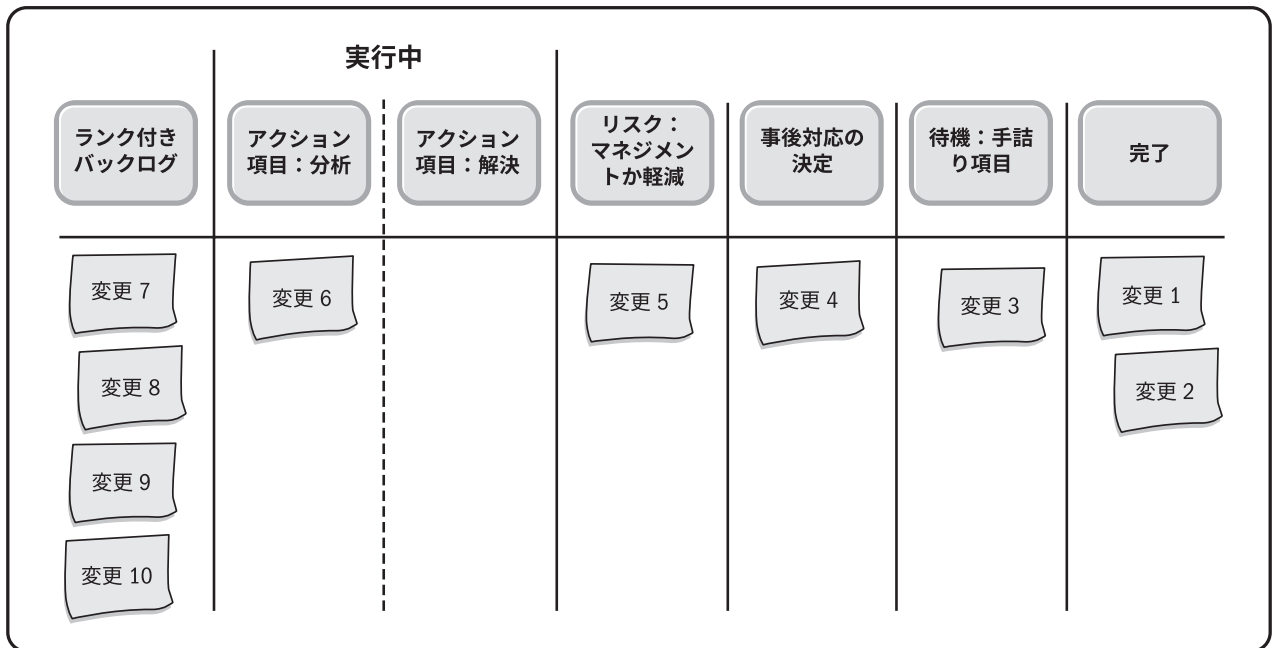


図6-4.バックログとカンバンを使用した変更作業の体系化と追跡

これらのツールを使用して体系化しマネジメントすることにより、変更実施の進捗状況が可視化され、実施手法のモデル化が可能になる。透明かつ好ましい方法で変更を展開することで、成功の可能性が高まる。

7

行動の喚起

アジャイル宣言が2001年に初めて刊行されて以来、プロジェクトのマネジメントにアジャイルを採用した件数とその手法は劇的に増加した。アジャイルのマインドセットを採用したり、アジャイルのマインドセットでの業務遂行を望んだりすることは、もはや特定規模の組織や情報技術に特化した組織に限定されない。このマインドセットは普遍的に適用され、その手法はさまざまな状況で成功している。

今日、「アジャイルである」ことへの要求はかつてなく高まっている。アジリティへの最善の道をめぐる議論は、対話とイノベーションを進化させ続けている。ただし、真実は変わらない。すなわち、検査、適応、そして透明性は、価値を円滑に提供する上で不可欠である。

この実務ガイドには期待したものが必ずしもすべてあるとは限らない。この実務ガイドのコア・チームは、私たちが情熱をこめて提示しようとした要素や手法に、読者が賛同しないかもしれないと認識している。対話を続け、この実務ガイドを改善を続けるためには、読者の熱意を必要とする。学習し、試行し、フィードバックを得て再び試行する。これは読者の道のりである。ガイダンスについてフィードバックを示し、この実務ガイドの今後の版に貢献するよう、レトロスペクティブへの協力を希望する。結局のところ、適応なしの調査は努力を無にする。

最後に、プロジェクトマネジメントとアジャイルの幅広いコミュニティに携わり、これらのトピックに関するさらなる対話を進めよう。カンファレンスやミーティングではPMIとAgile Alliance®それぞれの代表者を見つけて議論してほしい。ソーシャルメディアを使って、考えや意見をブログに投稿してほしい。

<https://www.projectmanagement.com/blogs/347350/Agile-in-Practice> の Agile in Practice というブログで、この実務ガイドの内容に関してフィードバックを投稿したり、会話したりすることができる。

付録A1

『PMBOK®ガイド』マッピング

表A1-1は、プロジェクトマネジメント・プロセス群の『PMBOK®ガイド 第6版』で定義された知識エリアへのマッピングを図示したものである。

この付録は、ハイブリッド・アプローチおよびアジャイル・アプローチが、『PMBOK®ガイド』知識エリア（表A1-2を参照）に記載されている属性にどのように対応しているかについて説明しており、成功の可能性を高めるために考慮すべきガイドラインとともに、何が同じで何が異なるかを取り上げる。

表A1-1.プロジェクトマネジメント・プロセス群と知識エリアのマッピング

知識エリア	プロジェクトマネジメント・プロセス群				
	立上げ プロセス群	計画 プロセス群	実行 プロセス群	監視・ コントロール・ プロセス群	終結 プロセス群
4. プロジェクト 統合マネジメント	4.1 プロジェクト憲章 の作成	4.2 プロジェクトマ ネジメント計画書 の作成	4.3 プロジェクト 作業の指揮・マネ ジメント 4.4 プロジェクト 知識のマネジメ ント	4.5 プロジェクト 作業の監視・コン トロール 4.6 統合変更管理	4.7 プロジェクト やフェーズの終結
5. プロジェクト・ スコープ・ マネジメント		5.1 スコープ・マネ ジメントの計画 5.2 要求事項の 収集 5.3 スコープの定義 5.4 WBSの作成		5.5 スコープの妥当 性確認 5.6 スコープのコン トロール	
6. プロジェクト・ スケジュール・ マネジメント		6.1 スケジュール・ マネジメントの計画 6.2 アクティビティ の定義 6.3 アクティビティ の順序設定 6.4 アクティビティ 所要期間の見積り 6.5 スケジュールの 作成		6.6 スケジュール のコントロール	
7. プロジェクト・ コスト・ マネジメント		7.1 コスト・マネジ メントの計画 7.2 コストの見積り 7.3 予算の設定		7.4 コストのコン トロール	
8. プロジェクト 品質マネジメント		8.1 品質マネジ メントの計画	8.2 品質のマネジ メント	8.3 品質のコン トロール	
9. プロジェクト 資源マネジメント		9.1 資源マネジメ ントの計画 9.2 アクティビティ 資源の見積り	9.3 資源の獲得 9.4 チームの育成 9.5 チームのマネ ジメント	9.6 資源のコン トロール	
10. プロジェクト・ コミュニケーション・ マネジメント		10.1 コミュニケー ション・マネジメ ントの計画	10.2 コミュニケー ションのマネジメ ント	10.3 コミュニケー ションの監視	
11. プロジェクト・ リスク・ マネジメント		11.1 リスク・マネ ジメントの計画 11.2 リスクの特定 11.3 リスクの定性 的分析 11.4 リスクの定量 的分析 11.5 リスク対応の 計画	11.6 リスク対応策 の実行	11.7 リスクの監視	
12. プロジェクト 調達マネジメント		12.1 調達マネジ メントの計画	12.2 調達の実行	12.3 調達のコン トロール	
13. プロジェクト・ ステークホルダー・ マネジメント	13.1 ステークホル ダーの特定	13.2 ステークホル ダー・エンゲー ジメントの計画	13.3 ステークホル ダー・エンゲー ジメントのマネジメ ント	13.4 ステークホル ダー・エンゲー ジメントの監視	

表A1-2. 『PMBOK®ガイド』 知識エリアに定義されたアジャイルの適用

『PMBOK®ガイド』 知識エリア	アジャイル作業プロセスでの適用
<p>第4章 プロジェクト統合マネジメント</p>	<p>反復的でアジャイル的な手法は、統合マネジメントにおける特定分野の専門家としてチーム・メンバーの関与を促進する。チームメンバーは計画と構成要素をどのように統合するか判断する。</p> <p>『PMBOK®ガイド』の統合マネジメントの主要概念に述べられているプロジェクト・マネジャーへの期待値は、適応型環境でも変わらないが、詳細なプロダクトの計画と作成のコントロールは、チームに委任される。プロジェクト・マネジャーは、協調的な意思決定環境を構築し、チームが確実に変更に対応できることに重点を置く。この協調的な手法は、チーム・メンバーが狭い専門化よりもむしろ幅広いスキル・ベースを有するときに、いっそう強化される。</p>
<p>第5章 プロジェクト・スコープ・マネジメ</p>	<p>進展する要求事項、ハイリスク、または著しい不確実性を伴うプロジェクトでは、スコープはしばしばプロジェクトの開始時点では理解されていなかったり、プロジェクトの途中で変化したりする。アジャイル型では、意図的にプロジェクトの初期段階ではスコープの定義や合意にあまり時間をかけないようにし、むしろ継続的な発見と調整のためのプロセスを確立するためにより多くの時間を費やす。新たな要求事項が発生してくる多くの環境では、実際のビジネス要求事項と、当初表明されたビジネス要求事項との間にしばしばギャップがある場合が多い。したがって、アジャイル型では、要求事項を調整するように意図的にプロトタイプを構築してレビューし、いくつかのバージョンをリリースする。その結果、スコープは定義され、プロジェクト全体を通して再定義される。アジャイル手法では、要求事項はバックログを構成する。</p>

表A1-2. 『PMBOK®ガイド』知識エリアに定義されたアジャイルの適用（続き）

『PMBOK®ガイド』知識エリア	アジャイル作業プロセスでの適用
<p>第6章 プロジェクト・スケジュール・マネジメント</p>	<p>適応型手法は、短期サイクルで作業を行い、結果をレビューし、必要に応じて適応させる。これらのサイクルは、適応型手法と成果物の適合性への迅速なフィードバックを提供し、『PMBOK®ガイド』の「プロジェクト・スケジュール・マネジメントにおける傾向と新たな実務慣行」の項で説明しているとおり、一般的に反復型スケジューリングおよびオンデマンド・スケジューリング、プルベース・スケジューリングという形で具現化される。</p> <p>大規模組織では、小規模プロジェクトと長期的なロードマップを必要とする大型施策が混在する可能性がある。これらのプログラムの開発ではスケールリング要因（例：チームのサイズ、地理的分布、法令順守、組織の複雑さ、技術の複雑さ）を使用してマネジメントする。大規模な組織全体にわたるシステムにおいて、そのライフサイクル全体に対応するためには、予測型手法、適応型手法、あるいは両方を用いた広範な技法を採用する必要があることがある。組織は、いくつかの中核的手法からの実務慣行を組み合わせたり、既に行った手法を採用したり、あるいはより伝統的技法の一部の原則や実務慣行を採用する必要があることがある。</p> <p>プロジェクト・マネジャーの役割は、予測型開発ライフサイクルを使ってプロジェクトをマネジメントしても、あるいは適応型環境でプロジェクトをマネジメントしても変わることはない。しかし、適応型手法の使用に成功するには、プロジェクト・マネジャーは、効果的に適用する方法を理解するためにツールと技法に習熟している必要がある。</p>
<p>第7章 プロジェクト・コスト・マネジメント</p>	<p>プロジェクトの不確実性が高かったり、スコープが完全に定義されていない場合、変更が頻繁にありうることから、詳細なコスト計算はあまり意味をなさないだろうその代わり、プロジェクト人件費の概算予測を迅速に作成するために簡易見積りの方法が用いられ、変更が生じた際に容易に調整できる詳細な見積りは、ジャストインタイム方式で行う短期計画のために用いられる。</p> <p>変動要因が多く予算が厳しいプロジェクトでは、スコープおよびスケジュールはコスト制約条件内に収まるように、より頻繁に調整される。</p>

表A1-2. 『PMBOK®ガイド』知識エリアに定義されたアジャイルの適用（続き）

『PMBOK®ガイド』知識エリア	アジャイル作業プロセスでの適用
<p>第8章 プロジェクト品質マネジメント</p>	<p>多くの変更に対処するために、アジャイル型の方法ではプロジェクトの終了に向けてというよりも、プロジェクト全体にわたって組み込んだ頻繁な品質活動とレビューのステップが必要になる。</p> <p>品質プロセスの有効性について振り返り（レトロスペクティブ）を繰り返して定期的に点検する。振り返りでは問題の根本原因を探し、次に品質改善に向けた新たな手法の試行を提案する。その後の振り返りではすべての試行プロセスを評価し、そのプロセスが機能しているかどうか、そして続行すべきか、調整を加えるべきか、あるいは使用を停止すべきかを判断する。</p> <p>頻繁な増分開発を促進するために、アジャイル型手法では、できるだけ多くのプロジェクト成果物の要素を組み込むようにしながら、作業の小さな集まりに焦点を当てる。小さな集まりの仕組み（スモール・バッチ・システム）は、変更による全体的なコストが低いプロジェクト・ライフサイクルの早期のうちに、不整合や品質問題を見つけることを目的とする。</p>
<p>第9章 プロジェクト資源マネジメント</p>	<p>変動性の高いプロジェクトは、広い知見を持つスペシャリストがいる自己組織化チームのような、精鋭と協業を最大化するチーム構造から恩恵を受ける。</p> <p>協業は、生産性を高め、革新的な問題解決を促進することを意図している。協業チームは、個々の作業アクティビティの統合をさらに加速促進し、コミュニケーションを改善し、知識の共有を強化し、そして他の利点に加えて仕事の割当ての柔軟性を提供する。</p> <p>協業の利点は他のプロジェクト環境にも適用されるが、上位からの指示や意思決定のための時間をもてないうえに、変動性が高く頻繁な変更があるプロジェクトを成功させるためには、協業チームが重要であることが多い。</p> <p>変動性の高いプロジェクトにおける物的および人的資源のための計画では予測性がかなり低い。こうした環境では、コストをコントロールしスケジュールを達成するために、迅速な供給や無駄を省く手法に関する合意が重要である。</p>

表A1-2. 『PMBOK®ガイド』知識エリアに定義されたアジャイルの適用（続き）

『PMBOK®ガイド』知識エリア	アジャイル作業プロセスでの適用
<p>第10章 プロジェクト・コミュニケーション・マネジメント</p>	<p>さまざまな曖昧さと変化に左右されるプロジェクト環境では、日々発生し、変化する細目をより頻繁かつ迅速に伝える特有のニーズがある。こうしたニーズは、チーム・メンバーの情報へのアクセスを合理化し、チームのチェックポイントを頻繁に設け、チーム・メンバーをできる限り同一場所に配置するように促す。</p> <p>さらに、プロジェクト生成物を可視化するために掲示し、定期的にステークホルダーによるレビューを行うことは、経営層およびステークホルダーとのコミュニケーションを促進することを意図している。</p>
<p>第11章 プロジェクト・リスク・マネジメント</p>	<p>高いばらつきをもつ環境は、当然ながら、より高い不確実性とリスクがある。これに対処するために、適応型手法を使用してマネジメントされたプロジェクトは、増分作業プロダクトの頻繁なレビューと機能横断型プロジェクト・チームを活用して知識共有を促進し、リスクが理解されマネジメントされていることを確認する。各反復の内容を選択する際にはリスクが考慮され、反復ごとにリスクが特定、分析、マネジメントされる。</p> <p>さらに、要求事項は、定期的に更新する生きている文書として保管され、現行のリスク・エクスポージャーの理解が深まるにつれ、作業がプロジェクトの進捗に応じて再度優先順位付けする。</p>

表A1-2. 『PMBOK®ガイド』知識エリアに定義されたアジャイルの適用（続き）

『PMBOK®ガイド』知識エリア	アジャイル作業プロセスでの適用
<p>第12章 プロジェクト調達マネジメント</p>	<p>アジャイル型環境では、チームを拡張するために特定の納入者が使用されることがある。この協業的な作業関係は、プロジェクトに関連するリスクと報酬を購入者と納入者の両方が共有する、共有リスク調達モデルにつながる可能性がある。</p> <p>大規模プロジェクトでは、成果物の一部には適応型アプローチを使用し、他の部分にはより安定したアプローチを使用することがある。このような場合、マスター・サービス契約（MSA）などの包括的な合意を契約全体に使用し、適応型の作業は付録や補遺に記載する。これにより、契約全体に影響を与えることなく、変更は適応型のスコープで対応できる。</p>
<p>第13章 プロジェクト・ステークホルダー・マネジメント</p>	<p>変化の度合いが大きいプロジェクトでは、プロジェクト・ステークホルダーの積極的な関与と参加が必要になる。タイムリーで生産的なディスカッションと意思決定を促進するために、適応型のチームは何層ものマネジメントを通すのではなく、直接ステークホルダーと関わり合う。多くの場合、クライアント、ユーザー、および開発者は、より多くのステークホルダーの関与とより高い満足につながるダイナミックな共創プロセスにおいて情報を交換している。プロジェクト全体を通じたステークホルダー・コミュニティとの定期的な交流は、リスクを軽減し、信頼関係を構築し、プロジェクト・サイクルの初期段階での早期の調整をサポートし、コストを削減し、プロジェクトの成功の可能性を高める。</p> <p>組織内および組織を超えた情報の共有を加速するため、アジャイル手法では積極的な透明性を促進する。プロジェクト会議やレビューにステークホルダーを招請したり、プロジェクト生成物を共有スペースに掲示することの意図は、プロジェクトの変更に伴う不整合や依存関係、その他の課題をできるだけ早く明るみに出すことにある。</p>

付録A2 アジャイル宣言とのマッピング

本付録はアジャイル宣言の要素が『アジャイル実務ガイド』にどのように取り上げられているかを記述したものである。

表A2-1. 『アジャイル実務ガイド』で取り上げられているアジャイル宣言の価値

価値	セクションおよびタイトル別アジャイル実務ガイドの範囲
プロセスやツールよりも個人と対話を	4.2 チームに力を与えるサーバント・リーダーシップ 4.3 チーム構成 5.1 プロジェクト憲章とチーム憲章 5.2.4 デイリー・スタンドアップ 6.2 組織文化
包括的なドキュメントよりも動くソフトウェアを	5.2.2 バックログの準備 5.2.3 バックログの洗練 5.2.5 デモンストレーションとレビュー 5.2.7 チームの価値創出を助ける実務慣行の実施
契約交渉よりも顧客との協調を	4.3 チーム構成 5.4 アジャイル・プロジェクトにおける測定 6.2 組織文化 6.3 調達と契約 6.7 組織構造
計画に従うことよりも変化への対応を	5.2.1 レトロスペクティブ 5.2.3 バックログの洗練 5.2.5 デモンストレーションとレビュー

表A2-2.アジャイル宣言の背後にある原則の実務ガイドとのマッピング

原則	アジャイル実務ガイドの範囲
顧客満足を最優先し、価値のあるソフトウェアを早い段階で継続的に提供する。	3.1 プロジェクト・ライフサイクルの特性 5.2.7 チームの価値創出を助ける実務慣行の実施
要求の変更はたとえ開発の後期であっても歓迎する。変化を味方につけることによって、お客様の競争力を引き上げる。	5.2.3 バックログの洗練
動くソフトウェアを、2-3週間から2-3ヶ月というできるだけ短い時間間隔でリリースする。	5.2 一般的なアジャイル実務慣行
ビジネス側の人と開発者は、プロジェクトを通して日々一緒に働かなければならない。	4.2 チームの奮起を促すサーバント・リーダーシップ 5.2.2 バックログの準備 5.2.3 バックログの洗練
意欲に満ちた人々を集めてプロジェクトを構成する。環境と支援を与え仕事が無事終わるまで彼らを信頼する。	4.3 チーム構成 5.1 プロジェクト憲章とチーム憲章 5.2.1 レトロスペクティブ
情報を伝えるもっとも効率的で効果的な方法はフェイス・トゥー・フェイスで話をするることである。	4.3.4 チームの構造 5.2.4 デイリー・スタンドアップ
動くソフトウェアこそが進捗の最も重要な尺度である。	5.2.7 チームの価値創出を助ける実務慣行の実施 5.2.8 イテレーションと増分が動作するプロダクトの引渡しに役立つ方法
アジャイル・プロセスは持続可能な開発を促進する。一定のペースを継続的に維持できるようにしなければならない。	5.1 プロジェクト憲章とチーム憲章
技術的卓越性と優れた設計に対する不断の注意が機敏さを高める。	5.2 一般的なアジャイル実務慣行
シンプルさ（ムダなく作れる量を最大限にすること）が本質である。	5.2.2 バックログの準備 5.2.3 バックログの洗練
最良のアーキテクチャ・要求・設計は、自己組織的なチームから生み出される。	4.3 チーム構成
チームがもっと効率を高めることができるかを定期的に振り返り、それに基づいて自分たちのやり方を最適に調整する。	5.2.1 レトロスペクティブ

付録A3

アジャイルおよびリーン・フレームワークの概要

本付録では、一般的に使用されるアジャイル・アプローチのいくつかについて記述する。これらのアプローチは、そのまま使用することも、与えられた環境や状況に最も適するように組み合わせることもできる。これらを使わなければならないわけではない。すなわち、アジャイル宣言のマインドセット、価値観、原則に従う限り、アジャイル・アプローチを最初から開発することができる。アジャイルの原則を守って持続可能なペースで価値を提供し、開発されたアプローチが顧客とのコラボレーションを促進する限り、特定のアプローチは要求されない。各アプローチの詳細へのリンクは、本ガイドの参考文献リストを参照のこと。

A3.1 『アジャイル実務ガイド』の選定基準

本実務ガイドには、多数のアジャイル・アプローチおよび技法が明示的に含まれている。図A3-1は、ガイダンスの深度とそのライフサイクルの幅に基づいてアジャイル・アプローチのひとつのサンプルを示したものである。議論のために選択された特定のアプローチは、次のような観点でよく知られている例である。

- ◆ **包括的な使用のために設計されている。** アジャイル・アプローチの中には、見積りや反映といった単一プロジェクトの活動に集中しているものもある。例の一覧には、より包括的なアジャイル・フレームワークのみが含まれている。他のアプローチに比べ、より充実した機能を備えているものもあるが、選択されたアプローチはすべて広範囲のプロジェクト活動の指針となるよう意図されている。
- ◆ **一般的な使用に形式化されている。** アジャイル・フレームワークの中には本質的に専有であり、単一の組織、またはひとつの状況における特定用途を対象としているものもある。第A3.2節から第A3.14節に記述されているフレームワークは、さまざまな状況における一般用途を目的としたフレームワークに焦点を当てている。
- ◆ **近年のプロジェクトでよく使用されている。** アジャイル・フレームワークの中には、包括的に設計され、十分に形式化されているものもあるが、ほとんどのプロジェクトや組織で一般的に使用されていないという現実がある。本付録に記述されているアジャイル・フレームワークは、かなりの数の業界で採用されていることが最近の業界調査で明らかになった。

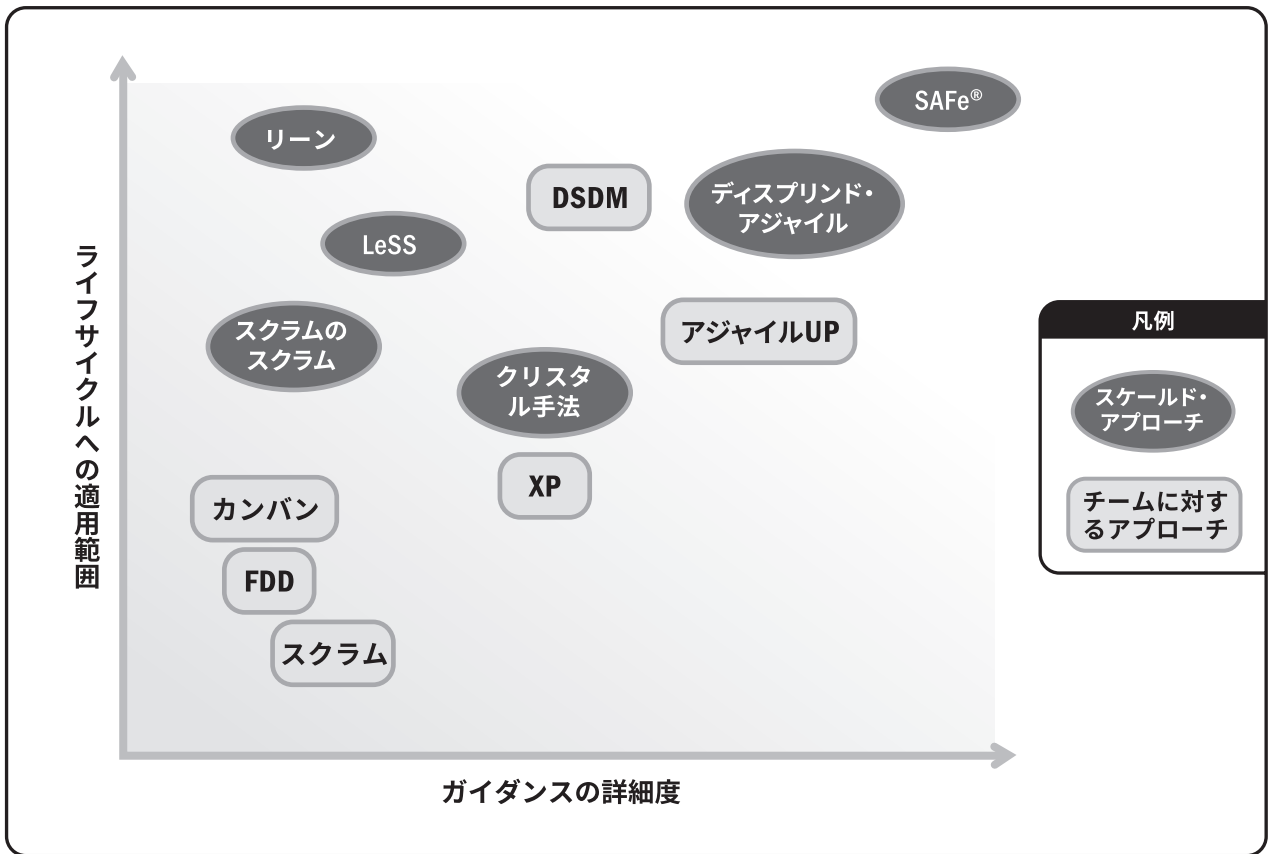


図 A3-1.適用範囲と詳細度でプロットしたアジャイル・アプローチ

A3.2 スクラム

スクラムは、プロダクト開発をマネジメントするために使用される単一チームのプロセス・フレームワークである。このフレームワークはスクラムにおける役割、イベント、生成物、およびルールから構成され、反復型アプローチを通して動くプロダクトを創出する。スクラムは、潜在的にリリース可能なプロダクト増分を生成する、スプリントと呼ばれる、1か月以内のタイムボックス化された固定の期間で実行される。表A3-1は、プロジェクトの実行に利用されるスクラムのイベントと生成物を示している。

スクラム・チームは、プロダクト・オーナー、開発チーム、およびスクラム・マスターから構成される。

- ◆ プロダクト・オーナーは、プロダクトの価値を最大に高める責任を負う。
- ◆ 開発チームは、作業対象プロダクトを創出するためにチーム外の人間に依存することなく、チーム内で必要なものをすべて持っているチーム・メンバーから構成された、機能横断型の自己組織化チームである。
- ◆ スクラム・マスターは、スクラム・プロセスが確実に維持され、スクラム・チームが実務慣行や規則を忠実に守ることを保証すると共に、障害の除去についてチームをコーチングする責任を負う。

表A3-1.スクラム・イベントと生成物

イベント	生成物
スプリント	プロダクト・バックログ
スプリント計画	スプリント・バックログ
デイリー・スクラム	増分
スプリント・レビュー	
スプリント・レトロスペクティブ	

A3.3 エクストリーム・プログラミング

エクストリーム・プログラミング (XP) は、頻繁なサイクルに基づくソフトウェア開発手法のひとつである。この名称は、既存のベストプラクティスを最も純粋でシンプルな形態にまで洗練し、その実務慣行をプロジェクト全体を通して継続的に適用するという理念に基づいている。

XPは、ソフトウェア・プロジェクトの結果を向上させることを目的とした包括的な実務慣行を普及させることで最もよく知られている。この方法は、最初は12の主要実務慣行のセットとして形式化されたが、徐々に進化していくつか他の実務慣行も採用するようになった。これらを表A3-2に示す。

表A3-2.エクストリーム・プログラミングの実務慣行

XP実務エリア	一次	二次
組織的	<ul style="list-style-type: none"> • 全員同席 • チーム全体 • 情報交換可能なワークスペース 	<ul style="list-style-type: none"> • 真の顧客関与 • チームの継続性 • 持続可能なペース
技術的	<ul style="list-style-type: none"> • ペア・プログラミング • テストファースト・プログラミング • インクリメンタルな設計 	<ul style="list-style-type: none"> • 共有コードや共同オーナーシップ • コードとテストからの文書化 • リファクタリング
計画	<ul style="list-style-type: none"> • ユーザー・ストーリー • 週次サイクル • 四半期サイクル • スラック 	<ul style="list-style-type: none"> • 根本原因分析 • 縮小チーム • 従量課金 • 交渉によるスコープ契約 • デイリー・スタンドアアップ
統合	<ul style="list-style-type: none"> • 10分ビルド • 継続的統合 • テストファースト 	<ul style="list-style-type: none"> • 単一のコードベース • 漸進型展開インクリメンタルなデプロイ • デイリー・デプロイ

この進化は、コア・バリュー（コミュニケーション、シンプルさ、フィードバック、勇気、敬意）のフィルターを通した技法を設計し採用した結果であり、主要な原則（人間性、経済性、相互利益、自己相似性、改善、多様性、振返り、流れ、機会、冗長性、失敗、品質、ベイビー・ステップ、責任の引受け）に基づく。

A3.4 カンバン方式

リーン生産におけるかんばん方式とは、在庫管理と補充のスケジューリングに用いるシステムである。この「ジャストインタイム」在庫補充プロセスは、もともとは、サプライヤーの在庫ではなく棚のギャップに基づいて再補充される食料品店で見られたものである。これらのジャストインタイム在庫システムに触発された大野耐一氏がかんばん方式を開発し、1953年にはトヨタの主要生産工場に適用した。

「かんばん」という言葉は、「視覚標識」とか「カード」という意味である。カードを使った物理的なカンバンは、誰もが見ることができるようにし、システムを介した作業の流れを可能にする。この情報ラジエーター（大型表示）は、作業を完了するために順番に行う必要がある作業の状況を表す列で構成されている。最もシンプルなカンバンには3つの列（すなわち、to do（実行前）、doing（実行中）、done（完了））があるが、利用するチームが必要とみなすどのような状態にも適応することができる。

カンバン方式は多くの設定で利用され、適用可能であり、顧客にも継続的に作業と価値の流れをもたらすことが可能である。カンバン方式は、他のアジャイル・アプローチより規範が少なく、「現状から始める」アプローチであるため、開始の妨げとなるものが少ない。組織は比較的容易にカンバン方式の適用を開始し、必要もしくは適切と判断された場合は、全面的な採用を進めていくことができる。

ほとんどのアジャイル・アプローチとは異なり、カンバン方式はタイムボックス化されたイテレーションの使用を規定していない。イテレーションはカンバン方式内で使用できるが、単一の項目をプロセスを通して継続的に進めていき、フローを最適化するために仕掛り中の作業数を制限するという原則は常に変わらない。カンバン方式は、チームや組織が次の条件を必要とする場合に最も適している。

- ◆ **柔軟性** チームは通常タイムボックスには拘束されず、作業のバックログにある最優先項目に取り組む。
- ◆ **継続的引渡しへの注力** チームはこの仕組みを通して作業を完了させることに注力し、仕掛り中の作業が完了するまでは新しい作業を開始しない。
- ◆ **生産性と品質の向上** 生産性と品質は進行中の作業数を制限することによって向上する。

- ◆ **効率の向上** 各タスクを点検して付加価値をもたらす活動かそうでない活動かを判断し、付加価値のない活動は排除する。
- ◆ **チーム・メンバーの注力** 仕掛り中の作業数が限られているため、チームは現在の作業に注力することができる。
- ◆ **ワークロードの変動** 作業が移動してくる途中で予測不能な事態が発生し、たとえ短期間であってもチームが予測可能なコミットメントをすることが不可能になった場合などのことである。
- ◆ **ムダの削減** 透明性によってムダが可視化され、除去することができる。

カンバン方式はリーン思考の原則に由来している。表A3-3は、カンバン方式を特徴付ける原則と主要な特性を示している。

カンバン方式は、組織の漸進的で進化するプロセスおよびシステム変更のための包括的なフレームワークである。この方式では、プロセスを通して作業を移動するために「プル・システム」が使用される。ひとつの項目が完了すると、チームは別の項目をひとつ、そのステップに取り出すことができる。

表A3-3.カンバン方式の原則と特性の定義

原則の定義	コア・プロパティ
現在の状態から開始	ワークフローの視覚化
漸進的、進化的な変化を追求することへの同意	仕掛り中作業の制限
現在のプロセス、役割、責任、およびタイトルの尊重	フローのマネジメント
あらゆるレベルでのリーダーシップの奨励	プロセス方針の明確化
	フィードバック・ループの実施
	協調して改善

図A3-2に示すカンバンは、初めはあまりに単純化された初歩的な技術のように見えるが、使ってみれば直ちにその威力を認識することになる。列への入出力に関するポリシーと同様、仕掛り中の作業数を制限することにより、カンバンはワークフロー、ボトルネック、ブロッカー、および全体的な状況を見抜く明確な知見を提供する。さらに、カンバンは情報を見る人すべてに情報ラジエーターとして機能し、チームの作業状況に関する最新の情報を提供する。

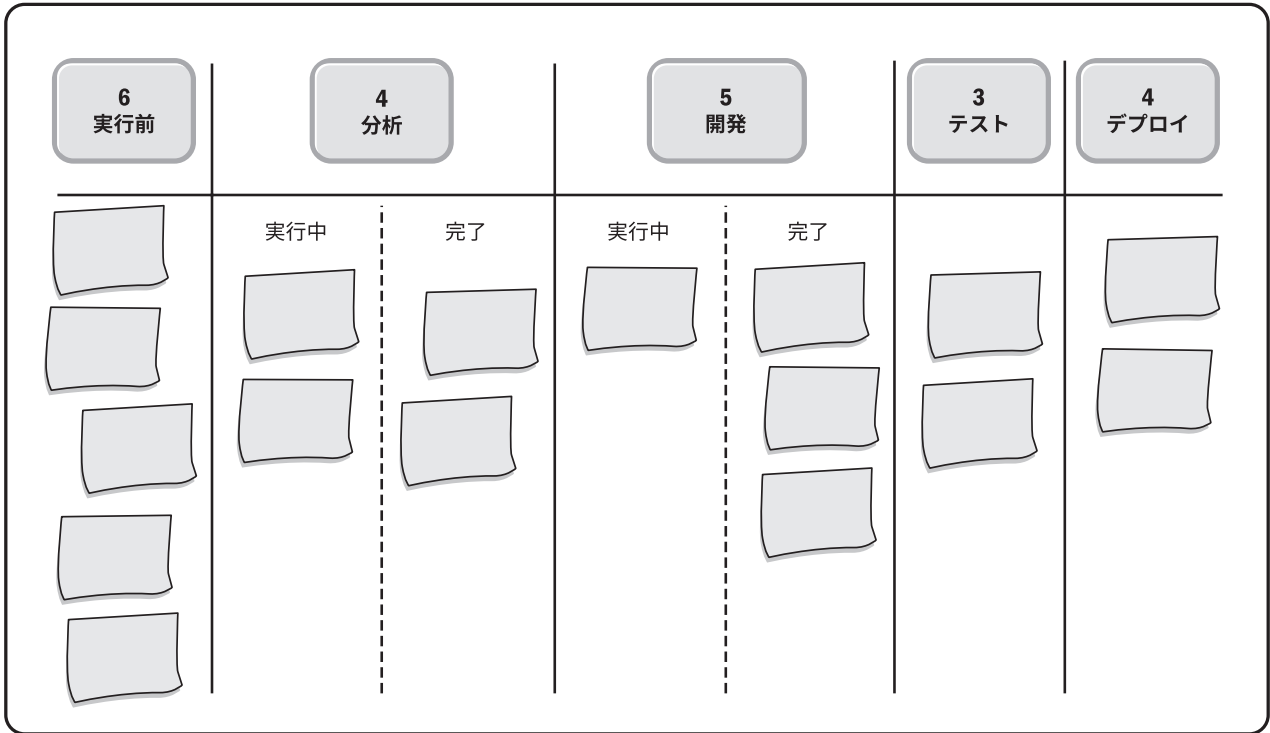


図 A3-2.仕掛り中の作業数制限を示すカンバンと、作業の流れを最適化するプル・システム

カンバン方式では、新しい作業を始めることよりも作業を完了することの方がより重要である。完了していない作業から得られる価値はないため、チームは協力して仕掛り中の作業（WIP）制限を実施および順守し、システムを通して各作業を「done（完了）」にする。

A3.5 クリスタル手法

クリスタルは手法の集合体である。クリスタル手法は、プロジェクトの規模（プロジェクトに
 関与する人数）とプロジェクトの重要性に基づいて、拡張性と厳密性を提供する。

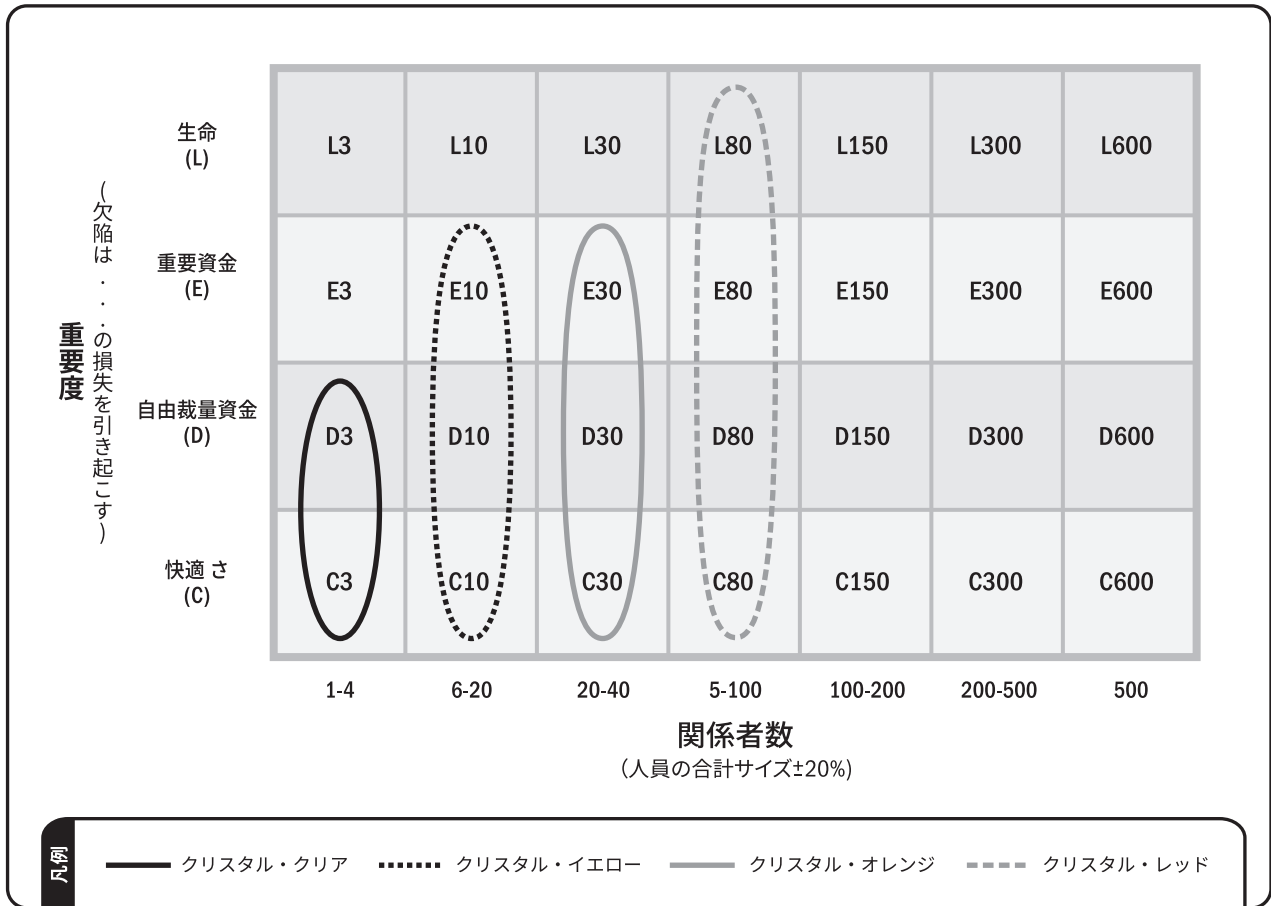


図 A3-3.クリスタル・ファミリー

クリスタル手法は、各プロジェクトにおいて、プロジェクト固有の特性を満たすために方針、実務慣行、およびプロセスに若干のテーラリングが必要になることを認識している。クリスタル・ファミリーは、使用する手法を決定するために「重み」に基づいて異なる色を使用する。「クリスタル」という言葉は宝石に由来しており、ここでは様々な「面」がそれぞれ基本的な原則や価値観を表している。「面」は、表A3-4に示す技法、ツール、標準、および役割を表している。

表A3-4.クリスタルのコア・バリューと共通プロパティ

コア・バリュー	共通プロパティ ^A
人 対話 コミュニティ スキル タレント コミュニケーション	頻繁な引渡し 内省的改善 緊密または浸透性コミュニケーション 個人の安全 注力点 エキスパート・ユーザーへの簡単アクセス 自動テスト、コンフィギュレーション・マネジメント、および頻繁な統合を備えた技術環境

^Aこれらのプロパティがプロジェクトにあればあるほど成功の確率が高くなる。

A3.6 スクラムバン

スクラムバンは、もともとはスクラムからカンバンに移行する方法として設計されたアジャイル・アプローチである。追加のアジャイル・フレームワークと方法論が出現したのに伴い、スクラムバンはそれ自体が進化するハイブリッド・フレームワークとなった。このフレームワークではチームはスクラムをフレームワークとして使用し、カンバンはプロセス改善のために使用する。

スクラムバンでは、作業は小さな「スプリント」に体系化され、カンバンを使用して作業を視覚化し、監視する。ストーリーはカンバンに掲示され、チームは仕掛り中の作業数制限を使って作業をマネジメントする。チーム間の協力関係を維持し、障害を取り除くためにデイリー・ミーティングが設けられる。計画立案が適切に設定され、通常、仕掛り中の作業数があらかじめ決めておいた制限より低いときに、チームは次の計画をいつ行うか判断できる。スクラムバンには事前に規定された役割はなく、チームは現在の役割を保持する。

A3.7 フィーチャー駆動開発

フィーチャー駆動開発（FDD）は、大規模なソフトウェア開発プロジェクトの特定のニーズを満たすために開発された。フィーチャーは、小規模なビジネス価値の達成能力に関連している。

フィーチャー駆動開発プロジェクトには6つの主要な役割があり、個人は次の役割のひとつ以上を担うことができる。

- ◆ プロジェクト・マネジャー
- ◆ チーフ・アーキテクト
- ◆ 開発マネジャー
- ◆ チーフ・プログラマー
- ◆ クラス・オーナー
- ◆ ドメイン・エキスパート

フィーチャー駆動開発プロジェクトは、5つのプロセスまたは活動を中心に体系化され、繰り返し実行される。

- ◆ 全体的なモデルを開発する。
- ◆ フィーチャー・リストを作成する。
- ◆ フィーチャー別に計画する。
- ◆ フィーチャー別に設計する。
- ◆ フィーチャー別に構築する。

これらの5つのプロセスのライフサイクル・フローと相互作用を図A3-4に示す。

フィーチャー駆動開発の活動は、ソフトウェア・エンジニアリングのベスト・プラクティスのコア・セットによってサポートされる。

- ◆ ドメイン・オブジェクトのモデリング
- ◆ フィーチャー別の開発
- ◆ 個々のクラスのオーナーシップ
- ◆ フィーチャー・チーム
- ◆ 検査
- ◆ コンフィギュレーション・マネジメント
- ◆ 定期的なビルド
- ◆ 進捗状況と結果の可視性

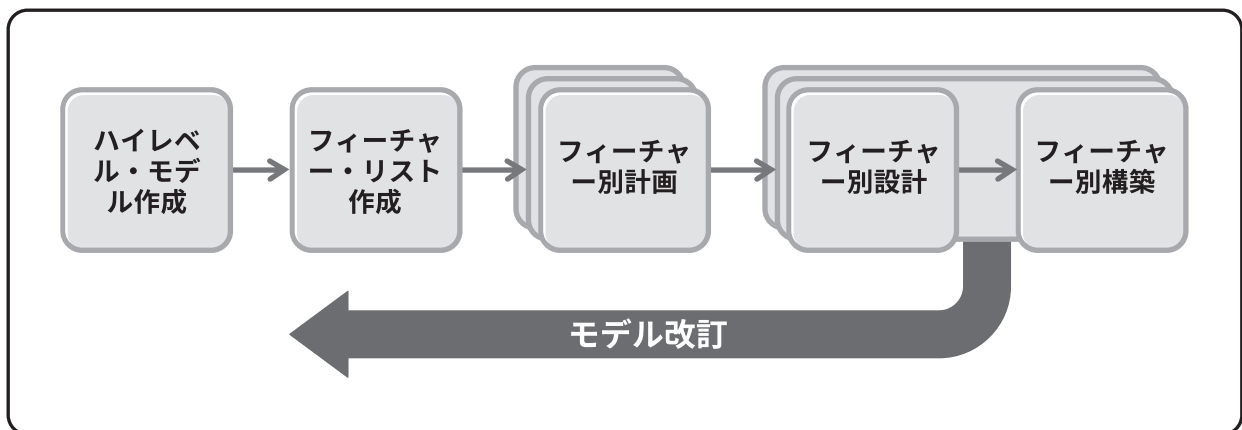


図 A3-4.フィーチャー駆動開発プロジェクトのライフサイクル

A3.8 動的システム開発アプローチ

動的システム開発アプローチ(DSDM)は、1990年代に普及した既存の反復アプローチに更に厳密性を加えるように設計されたアジャイル・プロジェクトのデリバリー・フレームワークであり、業界リーダーが非商用として共同開発したものである。

DSDMは制約駆動型のデリバリーを重要視することで最もよく知られている。このフレームワークでは、最初にコスト、品質、時間を設定し、図A3-5に示すように、これらの制約を満たすために形式化されたスコープの優先順位付けを使用する。

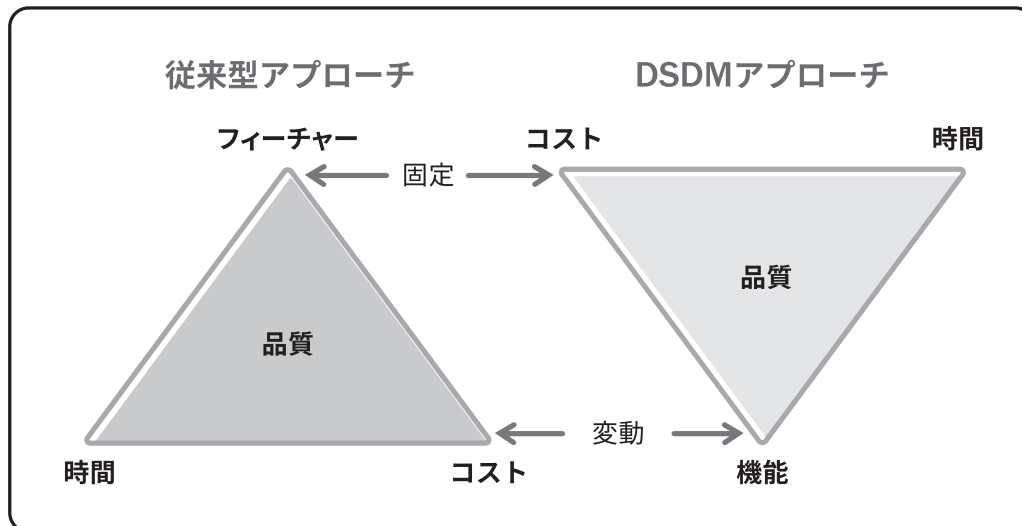


図 A3-5. 制約駆動型アジリティへのDSDMアプローチ

8つの原則がDSDMフレームワークの使用をガイドする。

- ◆ ビジネスニーズに注力する。
- ◆ 時間通りに引き渡す。
- ◆ 協業する。
- ◆ 品質で決して妥協しない。
- ◆ 確かな基盤から漸進的に構築する。
- ◆ 反復的に開発する。
- ◆ 継続的かつ明確にコミュニケーションする。
- ◆ 制御されていることを明確にする（適切な技法を使用する）。

A3.9 アジャイル統一プロセス

アジャイル統一プロセス (AgileUP) は、ソフトウェア・プロジェクト向けの統一プロセス (UP) の派生物であり、以前の統一プロセスよりも、より加速化されたサイクルとより軽いプロセスとして特徴づけられる。このプロセスの目的は、7つの主要分野すべてでより反復的なサイクルを実行し、正式なデリバリーの前に関連するフィードバックを組み込むことである。ガイドとなる原則に沿った分野を表A3-5に示す。

表A3-5.アジャイル統一プロセスの主要要素

リリース内の分野	分野をガイドする原則
モデル	チームは自分が何をしているか理解している
実施	シンプルさ
テスト	アジリティ
デプロイ	高価値活動への注力
コンフィギュレーション・マネジメント	ツールの独立性
プロジェクトマネジメント	フィットするようにテーラリング
環境	状況に特化

A3.10 スケーリングのフレームワーク

A3.10.1 スクラムのスクラム

スクラムのスクラム (SoS) とは「メタスクラム」とも呼ばれ、ひとつの大きなスクラム・チームの代わりに、それぞれ3人から9人のメンバーからなるふたつ以上のスクラム・チームが、作業を調整する必要があるときに使用する技法である。各チームの代表者は、毎日であってもよいが通常は週2~3回、他のチームの代表者との会議に出席する。毎日の会議はスクラムのデイリー・スタンドアップと同様に行われ、完了した作業、次の作業、現在ある障害、および他のチームを妨害する可能性のある今後の障害について代表者が報告する。その目標はチームが確実に作業を調整し、すべてのチームの効率を最適化するための障害を取り除くことである。

複数のチームを有する大規模なプロジェクトでは図 A3-6に示すように、各SoSの代表者がより大きなグループの代表者に報告するSoSと同じパターンに従う、スクラムのスクラムのスクラムを実施することも考えられる。

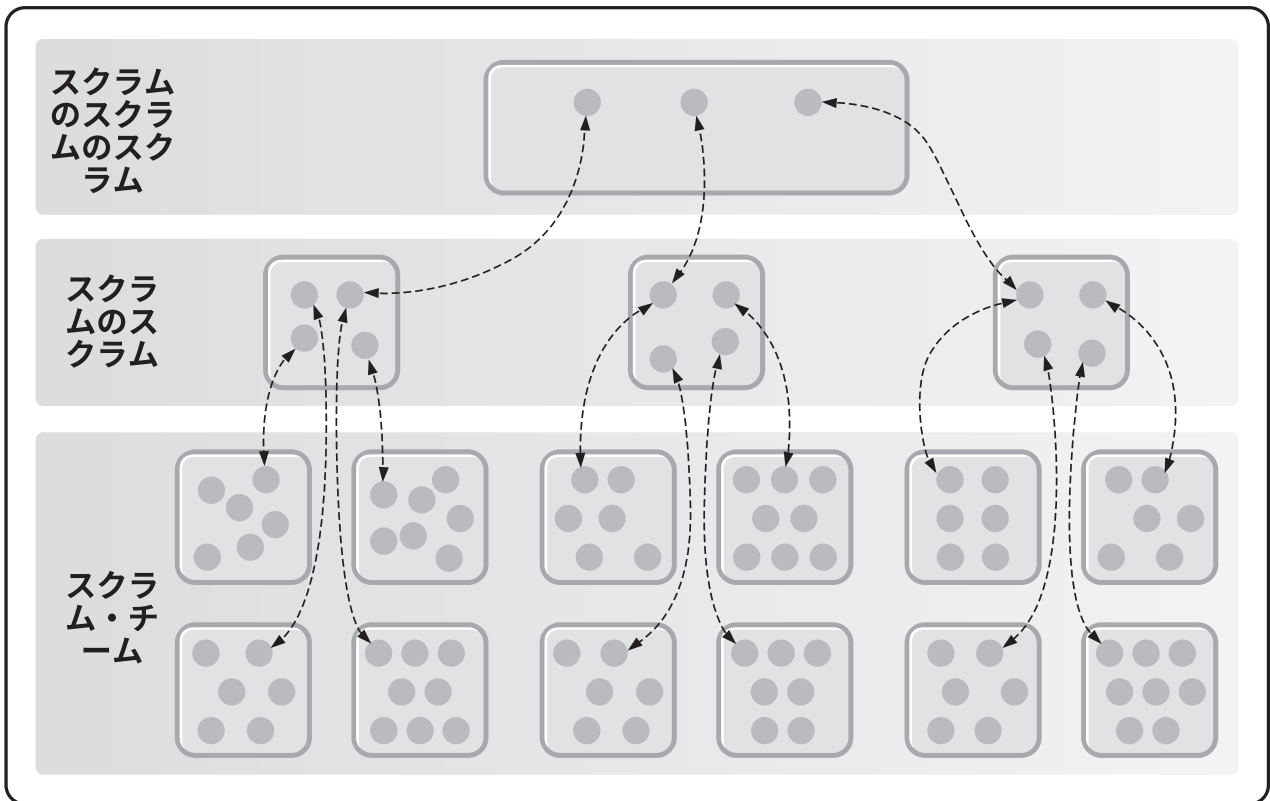


図 A3-6. SoSチームに参加するスクラム・チームの代表者

A3.11 スケールド・アジャイル・フレームワーク (SAFE)

スケールド・アジャイル・フレームワーク (SAFe®) は、企業のあらゆるレベルでの開発作業をスケールアップするためのパターンの知識ベースを提供することに重点を置く。

SAFe® では次の原則に焦点を当てる。

- ◆ 経済的な視点をとる。
- ◆ システム思考を適用する。
- ◆ 変化を想定し、複数の選択肢を残す。
- ◆ 素早い統合された学習サイクルで漸進的に構築する。
- ◆ 動くシステムの客観的な評価をもとに、マイルストーンを設定する。

- ◆ 仕掛り中の作業を制限し可視化する。バッチ・サイズを小さくし、キューの長さをマネジメントする。
- ◆ カデンツを適用し、ドメイン横断的な計画策定で同期する。
- ◆ ナリッジワーカーが自身のモチベーションを湧き立たせるようにする。
- ◆ 意思決定を分散させる。

SAFe® は、顧客への継続的な価値の提供に重点を置くバリュー・ストリームを中心として企業を組織化することを重視する。また、ポートフォリオ、プログラム、チーム・レベルでの実務慣行、役割、および活動の細密化を追求する。

A3.12 大規模スクラム (LESS)

大規模スクラム (LeSS) とは、図A3-6に示すスクラム方式を拡張するという共通目標に向けて複数の開発チームを組織化するためのフレームワークである。組織化の中核的原則は、従来型のシングル・チームによるスクラム・モデルの要素を可能な限り多く保持することである。これは、不要な混乱や複雑化を招きかねないモデルの拡張を最小限に抑えるのに役立つ。表A3-6にLeSSとスクラムの比較を示す。

表A3-6. LeSSとスクラムの比較

LeSSとスクラムの類似性	スクラムに追加されたLeSS技法
ひとつのバックログ	スプリント計画は、正式には「何が」と「どのように」のふたつの部分に分かれている
全チーム向けのひとつのdoneの定義	有機的な機能横断チームの調整
各スプリント終了時に、潜在的に出荷可能なひとつのバックログ増分	全体的な機能横断チームの洗練
1人のバックログ・オーナー	機能横断チームの改善に注力した全体的なレトロスペクティブ
完全な機能横断チーム	
ひとつのスプリント	

本質を失うことなくスクラムを拡張するために、LeSSはシステム思考、バックログ全体への注力、透明性など、確固とした原則の使用を推進する。

A3.13 エンタープライズ・スクラム

エンタープライズ・スクラムは単一プロダクトの開発ではなく、より組織全体のレベルでスクラム・アプローチを適用するために設計されたフレームワークである。具体的には、フレームワークは組織のリーダーに次のように助言する。

- ◆ 組織のあらゆる側面にスクラムを使用するようにする。
- ◆ スクラム技法を一般化して、さまざまな側面で簡単に適用できるようにする。
- ◆ 必要に応じ補足的な技法でスクラム・アプローチをスケーリングする。

エンタープライズ・スクラムの目的は、既存の価値基準を打ち砕くようなイノベーションを可能にすることにより、プロジェクトの実行にとどまらずアジャイル・アプローチを使用することである。

A3.14 ディスプリンド・アジャイル(DA)

ディスプリンド・アジャイル(DA)とは、いくつかのアジャイルのベストプラクティスを包括的なモデルに統合するプロセス決定フレームワークのひとつである。一般的な技法は焦点が狭すぎる（スクラムなど）か、あるいは細密で規範的（AgileUPなど）かのいずれかであり、DAは、こうした技法の間のバランスを取れるようにしたものである。このバランスを達成するため、次の原則に従ってさまざまなアジャイル技法をブレンドする。

- ◆ **人間第一主義** さまざまなレベルの役割と組織要素を列挙する。
- ◆ **学習指向** 協力的な改善を促す。
- ◆ **完全デリバリー・ライフサイクル** 複数の目的に合ったライフサイクルを推進する。
- ◆ **目標駆動型** 特定の結果を達成するためのプロセスをテーラリングする。
- ◆ **企業の認識** 部門横断的なガバナンスに関するガイダンスを提供する。
- ◆ **スケーラブル** プログラムの複雑性を多面的にカバーする。

付属文書X1 コントリビューターとレビューア

X1.1 『アジャイル実務ガイド』 中核委員会

ガイドの起草を担当するプロジェクトの中核委員会メンバー (レビューアによる推奨事項の審査と裁定を含む)。

X1.1.1 PROJECT MANAGEMENT INSTITUTE代表:

Mike Griffiths, PMP, PMI-ACP, (委員長)
Jesse Fewell, CST, PMI-ACP
Horia Slușanschi, PhD, CSM
Stephen Matola, BA, PMP

X1.1.2 アジャイル・アライアンス代表:

Johanna Rothman, MS (副委員長)
Becky Hartman, PMI-ACP, CSP
Betsy Kauffman, ICP-ACC, PMI-ACP

X1.2 『アジャイル実務ガイド』 当該分野専門家レビューア

草案をレビューし、SMEレビューを通じて推奨事項を提供する、招待形式の当該分野専門家。

Joe Astolfi, PMP, PSM	Laura Paton, MBA, PMP
Maria Cristina Barbero, PMI-ACP, PMP	Yvan Petit, PhD, PMP
Michel Biedermann, PhD, PMI-ACP	Dwayne Phillips, PhD, PMP
Zach Bonaker	Piyush Prakash, PMP, Prince2
Robert Bulger, PfMP, CSM	Dave Prior, PMP, CST
Sue Burk	Daniel Rawsthorne, PhD, PMP
Shika Carter, PMP, PMI-ACP	Annette D. Reilly, PMP, PhD
Lauren Clark, PMP, CSM	Stephan Reindl, PMI-ACP, PMP
Linda M Cook, CSM, CSPO	Reed D. Shell, PMP, CSP
Pamela Corbin-Jones, PMI-ACP, CSM	Cindy Shelton, PMP, PMI-ACP
Jeff Covert	Teresa Short
Alberto Dominguez, MSc, PMP	Lisa K. Sieverts, PMP, PMI-ACP
Scott P. Duncan, CSM, ICP-ACC	Christopher M. Simonek, PMP, CSM
Sally Elatta, PMI-ACP, EBAC	Robert “Sellers” Smith, PMP, PMI-ACP
Frank R. Hendriks, PMP, PMI-ACP	Ram Srinivasan, PMP, CST
Derek Huether	Chris Stevens, PhD
Ron Jeffries	Karen Strichartz, PMP, PMI-ACP
Fred Koos	Rahul Sudame, PMI-ACP
Philippe B. Kruchten, PhD, PEng	Joanna L. Vahlsing, PMP
Steve Mayner, SPCT4, PMP	Erik L. van Daalen
Michael S. McCalla, PMI-ACP, CSP	Annette Vendelbo, PMP, PMI-ACP
Don B. McClure, PMP, PMI-ACP	Dave Violette, MPM, PMP
Anthony C. Mersino, PMI-ACP, CSP	Anton Vishnyak, PMI-ACP, CSM
Kenneth E. Nidiffer, PhD, PMP	Chuck Walrad, MA, MS
Michael C. Nollet, PMP, PMI-ACP	

X1.3 フォーマット・フォーカス・グループ

以下の個人は、『アジャイル実務ガイド』の新しい内容スタイルと書式設定要素の開発を支援した。

Goran Banjanin, PgMP, PMP	Rajaraman Kannan, PMP, MACS CP
Andrew Craig	Amit Khanna PMP, PMI-ACP
Cătălin-Teodor Dogaru, PhD, PMP	Ariel Kirshbom, PMI-ACP, CSP
Jorge Espinoza, PMP	Bernardo Marques, PMP
Jennifer M. Forrest, CSM, PMP	Noura Saad, PMI-ACP, CSPO
Helen Fotos, PMP, PMI-ACP	Kurt Schuler, PMP
Dave Hatter, PMP, PMI-ACP	Demetrius L. Williams, MBA, PMP
Christopher Healy, PMP	Liza Wood
Mike Hoffmann, MBA, PMP	Melody Yale, CSP, SPC4
Chadi Kahwaji, PMP	

X1.4 PMI STANDARDSメンバー諮問グループ (MAG)

『アジャイル実務ガイド』のためにPMIに代わって指示を出し、最終承認したPMI Standardsメンバー諮問グループのメンバー。

Maria Cristina Barbero, PMI-ACP, PMP
Brian Grafsgaard, PMP, PgMP
Hagit Landman, PMP, PMI-SP
Yvan Petit PhD, PMP
Chris Stevens, PhD
Dave Violette, MPM, PMP
John Zlockie, MBA, PMP, PMI Standards マネジャー

X1.5 AGILE ALLIANCE® 理事会

次の個人は、Agile Alliance® 理事会メンバーで、『アジャイル実務ガイド』のためにAgile Allianceに代わって指示および最終的な承認を出した。

Juan Banda	Paul Hammond (会長)
Phil Brock (取締役社長)	Victor Hugo Germano
Linda Cook	Rebecca Parsons (秘書)
Stephanie Davis	Craig Smith
Ellen Grove	Declan Whelan

X1.6 PMIサポート・スタッフと学術研究サポート

次の個人は、この草案の開発と承認において、フォーマット・フォーカス・グループを支援するために、およびPMIのマーケティング活動において、中核委員会を支援するために尽力した。

Melissa M. Abel, マーケティング・コミュニケーション・スペシャリスト
Karl F. Best, PMP, CStd, Standards スペシャリスト
Alicia C. Burke, MBA, CSM, Product Manager, Credentials
Edivandro C. Conforto, PhD, アジャイル・リサーチのPMIコンサルタント
Dave Garrett, CSPO, 変革担当副社長
Erica Grenfell, 組織関係担当副社長補佐
M. Elaine Lazar, MA, MA, AStd, プロジェクト・スペシャリスト
Andrew Levin, PMP, プロジェクト・マネジャー
Tim E. Ogline, ユーザー・エクスペリエンス・デザイナー
Stephen A. Townsend, ネットワーク・プログラム・ディレクター
Michael Zarro, PhD, UX研究者

X1.7 PMIプロダクション・スタッフ

Donn Greenberg, 出版マネジャー
Kim Shinnars, 出版プロダクション・アソシエイト
Roberta Storer, プロダクト・エディター
Barbara Walsh, プロダクション監督

X1.8 日本語翻訳監修ボランティアグループメンバー

Tsunefumi Watanabe, PMP 渡邊恒文
Yasuji Suzuki, PMP 鈴木安而
Koji Ogawa, PMP, PMI-ACP, CSM 小川浩司
Hiroki Itakura, PMP, PMI-ACP, CSM, SA 板倉裕樹
Tadatoshi Sekiguchi, PMP, PMI-ACP, CSPO 関口匡稔

X1.9 翻訳監修委員会メンバー

Barbara Walsh 出版プロダクション監督
Margaret Lyons 試験デベロッパー
Stephen Townsend ディレクターNetwork Programs
Vivian Isaak プレジデント Magnum Group, Inc. 翻訳会社
Brian Middleton 戦略ソリューションマネジャー Magnum Group, Inc. 翻訳会社

付属文書X2

テーラリングに影響を与える属性

X2.1 はじめに

本付属文書では、アジャイル・アプローチをいつ、どのようにテーラリングするかについて、高度なガイダンスを提供する。これは、新しい技術の変更や導入の正当化が可能な状況を判断するために使用され、その後は考慮すべき推奨事項を提供する。

X2.2 最初の注意事項

テーラリングは高度なトピックであり、実施に際しては、さまざまな環境で本来定義されているアジャイル手法を使用して成功してきた経験豊富な実務者が取り組むべきである。言い換えれば、ある手法をテーラリングしようと試みるなら、その前に、その手法で経験を積み、成功している必要がある。

アジャイル実務慣行の採用に悪戦苦闘したとき、その実施を再検討することはよくある反応である。「レトロスペクティブは不評だったのでやめることにした」といった表明はこの問題を示す良い例であり、その方法のテーラリングでは解決できない、チームのより根本的な問題を示唆している。プロセスの改善を目的としたレトロスペクティブ活動を省くと状況はさらに悪化する。

スキル獲得の「守破離」モデルは、規則に従う（「守」とは従い守ることを意味する）ことから、意識的に規則から離れて移動（「破」とは変更または脱線することを意味する）し、そして最終的には、揺るぎない実務慣行と改善を通して個々の道筋を見つけだす（「離」は分離または離別を意味する）までの一連の流れを説明している。プロセスをテーラリングする「破」レベルや、新しいカスタム・プロセスを作り出す「離」レベルに移行しようとする前に、まず「守」レベルから開始し、練習する必要がある。

最後に、テーラリングはチーム・メートや、変更によって影響を受ける可能性のある人と協力して行う必要がある。人々を変更に対してコミットさせ賛同させて移行に成功するには、プロセスを変えることについての思考および意思決定プロセスに関与させる必要がある。プロセスのテーラリングから人々を除外すると、たとえそれが技術的には理にかなっていても、その変更に対する抵抗や反感をもたらすことがある。多くの場合、経験豊富な指導者やリーダーは効果的に人々に関与させることができる。

X2.3 本付属文書の使用方法

本付属文書に記載されているガイダンスを役立てるには、まず設計されたとおりにアジャイル・アプローチを首尾よく使用することが推奨される。その後、状況に応じた表X2-1のテーラリング・ガイドラインを精査し、関連する推奨事項を読む。次に、影響を受ける人々と変更について話し合い、行動指針で合意を得る。

第5章で説明しているように、変更を評価する良い方法は、恒久的に採用する前に、イテレーションをひとつかふたつ試行してみることである。あるいは、いくつかのフィーチャーを提供するフロー・ベースのアプローチを考慮する。そして、レトロスペクティブ（振り返り）を行い、再評価する。

試行し、そのフィードバックを提供できるとわかっているときに、人は何か新しいことを試す可能性が高くなる。タイムボックス期間にわたって試行した後、チームはレトロスペクティブ（振り返り）でその有効性を検討し、そのまま維持するか、修正して改善するか、使用しないかを判断する必要がある。

最後に、採用されテーラリングされたアプローチは、これらの特性を共有するプロジェクトに使用される標準的なプロセスとして制度化することができる。また、新しいアプローチの採用（またはテーラリング）について説明する第5章のガイドラインに従うことが推奨される。

X2.4 テーラリングの推奨事項

アプローチをテーラリングする前に考慮すべきいくつかの優れた実務慣行を次に挙げる。

X2.4.1 排除に潜む落とし穴

アジャイル実務慣行の多くは、補完しあう対として機能する。例えば理解のギャップをすばやく埋めることができるため、コロケーションと頻繁なビジネス対話により要求事項の軽量化が可能になる。同様に、XPの厳密なテストは、思い切ったリファクタリングを可能にする。これは実務慣行が他方の実務慣行をサポートするという例である。対になった実務慣行の理解や対処もなく何かを除去することは、問題を解決するどころかさらに多くの問題を生み出す可能性がある。

X2.4.2 テーラリング・ガイドラインの表の使用

表X2-1を使用して、特定の状況に応じたテーラリングのための推奨事項を検討する。変更から影響を受ける人と変更について話し合い、まずは短期の試行を計画するとともに、変更をコミットする前に誠実なフォローアップ・レビューを行う。

表X2-1.テーラリング・ガイドライン

状況	テーラリングの推奨事項
特大規模のプロジェクト・チーム。	<p>大型のプロジェクトを複数の小さなプロジェクトとして再構築する。最初に技術トライアル・プロジェクトを行い、実施プロジェクトを行う。</p> <p>より少数のフィーチャーをより頻繁にリリースすることを検討する。これにより、より小規模のプロジェクト・チームの構成が可能になる。</p> <p>チームを重要な中核メンバーまで縮小することを検討する。過剰な人数は、プロセスを妨げることはあっても、助けることはない。チームのサイズを小さくすると、動乱もコストも削減することができる。</p> <p>大規模チームを複数の小規模チームに分割し、プログラムマネジメントを使用して同期と調整を行う。</p> <p>アジャイルおよびリーンなプログラムマネジメントを使用して、より大きな労力として組織化する。</p> <p>DA、SAFe®、LeSSなどのスケールド・アジャイル・フレームワークまたはリーン・フレームワークを検討する。これらのフレームワークには、それぞれ幾つかの有用なアイデアがある一方、実施上のリスクのほか、負担とコストがある。</p>

表X2-1.テーラリング・ガイドライン(続き)

状況	テーラリングの推奨事項
<p>分散されたチーム</p>	<p>多くのプロジェクトには(何人かの)分散されたチーム・メンバーが携わる。インスタント・メッセージング、ビデオ会議、電子チームボードなどのツールは、コミュニケーションのギャップの多くを橋渡しするのに役立つ。</p> <p>チームが安定している可能性が高い場合、将来発生する遠隔会話をより効果的にするために、できるだけ早く対面式のミーティングを設定する。実際に対面した人々は、より信頼が高まるため、フィルターなしの議論に入る可能性が高い。</p> <p>顔の表情やボディランゲージの手がかりがない遠隔参加者との会議を行う場合、参加を確実にし、決定の合意を確認するために、ラウンドロビン方式でのチェックを検討する。</p> <p>また、イテレーション・ベースのアジャイル・アプローチの使用を検討する。チーム・メンバーが多くのタイムゾーンに分かれている場合は、プロジェクト全体のインタラクションをあまり行わずに、より小規模なミーティング(一度に2~3人)をより頻繁に開催することを奨励する。</p>
<p>安全性に重大な影響を及ぼすプロダクトでは、アジャイル・プロセスですぐに使えるものを超えた追加の文書化や適合性チェックが必要となる場合がある</p>	<p>これらの環境では依然としてアジャイル・アプローチを使用できるが、ドメインに必要な適合性のレビュー、文書化、および認証といった適切な追加の作業を行う必要がある。その場合、作成した文書は完成したフィーチャーと共にチームが提供するものの一部になる。フィーチャーは文書化が終了するまでは完了とならない。</p> <p>プロダクト環境で必要な、さらなる厳格さを備えたアジャイルがもつ、改善されたコラボレーションとコミュニケーションのベネフィットを得られるように、ハイブリッド・アプローチ(複数のアジャイル・アプローチ)の使用を検討する。航空機の飛行システム開発業者および製薬会社は、ベネフィットを活用して適切なコントロールを維持するために、独自のプロセスを組み合わせたアジャイル・アプローチを使用する。</p>
<p>安定した要求事項と実行プロセス</p>	<p>アジャイルは本当に必要か。要求事項の不確実性が低く、変更が少なく、実行リスクが最小の場合、完全に揃ったアジャイル・アプローチは不要なこともある。どんなプロジェクトもコラボレーションと透明性の向上からベネフィットを得られるが、一部の反復型ビルドやレビュー・サイクルは過剰となることもある。</p> <p>ビルドやフィードバックのサイクルが日常的に要求事項を明らかにしたり精緻化したりしない場合は、レビュー時間のコストへの影響を最小化するためサイクル期間を大きくとることを検討する。</p> <p>設計および開発段階でのプロジェクトの変更の頻度が高く、それを顧客に公表するようなプロセスとなっている場合、各プロジェクト・フェーズに適切なライフサイクル・モデルを使用するハイブリッド・アプローチの方が理にかなっているかもしれない。</p>
<p>チームが機能組織内の機能サイロ内にある</p>	<p>アジャイルは、機能横断チームという考えに基づいて構築されている。経営者の関与なしに、機能横断チームを構成しようとするか見てみるよう人々に要請することを検討する。</p> <p>報酬システムが機能領域をもとに報酬を与えるように体系化されている場合は、まずそれを変更することを検討してみる。人は、何らかの形で自分の報酬に影響するまでは、プロダクトやチームの利益のためには行動しない可能性がある。</p>

表X2-1.テラリング・ガイドライン (続き)

状況	テラリングの推奨事項
透明性が不安を引き起こす	<p>アジャイルは透明性のある文化を創造する。すなわち、人は開発の至る所で自分の仕事を見せ、共有する。このような中間成果物の共有や、成功、失敗、および現在の状態についてオープンで誠実であることが透明性である。透明性には勇気がある。</p> <p>例を示して指導し、ステータス・ボードまたはホワイトボードを使用して意思決定プロセスにおける透明性を示す。</p>
チーム・メンバーの多くは技術領域の知識を持たない	<p>アジャイル・アプローチは、作業の順序設定や問題解決の際に使用するアプローチなど、作業項目に関して現場での意思決定を行うための自発的なチームを奨励し、活用する。チーム・メンバーの大多数が未経験者の場合、合意にもとづくアプローチは問題の発生ややり直しにつながる可能性がある。したがって、これらのチームでは、チームが必要なスキルを取得するまでは「任命」と「指揮」という追加の支援が必要になる。言い換えれば、アジャイルを使用すると宣言するだけでなく、経験の浅いチームでもすべてを把握しようと自ら努力させることが重要である。彼らは自発的で自ら方向を決める。ガイダンスを提供し、領域知識を集積するのに役立つコンピテンシー・センターの構築を検討する。</p>
経営幹部からの賛同不足	<p>経営幹部の賛同が欠けていると、チームはアジャイルのマインドセットとアプローチと、より予測型のそれとの間の衝突に遭遇する。</p> <p>共通の基盤や、組織のニーズに基づいて改善すべきところを探り、試行やレトロスペクティブ（振り返り）を使用して進める。</p> <p>経営幹部への教育やトレーニングを検討する。短期サイクル、小規模バッチ・サイズ、頻繁なレビュー、および小さな改善を伴うレトロスペクティブ（振り返り）など、リーン思考の観点からアジャイルを説明することを検討してみる。</p>
アジャイルの用語や言語が組織文化に合わない	<p>アジャイル言語ではないにしても、活動を理解し同意できるように用語を修正する。それぞれの用語が何を意味するかについて具体的に説明する。</p> <p>例えば、組織が「ゲーム」という言葉をプロフェッショナルではないと判断した場合は「計画ゲーム」といった用語は使用しない。代わりに「計画ワークショップ」という用語の使用を検討する。</p>

付属文書X3 アジャイル適合性フィルター・ツール

X3.1 はじめに

アジャイル文献には、どのような状況でアジャイル・アプローチを使用するのが適切かを評価するのに役立つ多くのアジャイル適合性フィルター・ツールが含まれている。1994年に動的システム開発手法（Dynamic Systems Development Method：DSDM）は、適合性と潜在的な問題領域の把握に役立つため、「アジャイル・プロジェクト適合性アンケート」と「組織適合性アンケート」を開発した。

クリスタル系の手法でも適合性基準を採用し、チームのサイズ、開発されているプロダクトやサービスの重要性に基づいてプロジェクトをランク付けした。クリスタルでは、小さくて重要度の低いプロジェクトはより軽量なコントロールとよりシンプルなアプローチで実行するように推奨している。大規模な、ミッションクリティカルまたは生命に影響を与えるようなプロジェクトでは、より厳密な検証が推奨されたことから、

これらのアプローチの開発以来、アジャイル・アプローチをいつどこで採用するかを判断するのに役立つため、さらに多くのモデルが作成された。BoehmとTurnerは、DSDMおよびクリスタルの要素を一部採用して、プロジェクトをアジャイル・アプローチで実施すべきか、それともより従来型のアプローチで実施すべきかを判断するのに役立つ評価モデルを開発し、これが好評を得た。

こうした以前のモデルにもとづき、そしてハイブリッドアプローチとの妥協案を考えるために拡大された、次のモデルが提案されている。このモデルは、組織がプロジェクトを予測型、ハイブリッド、またはアジャイル・アプローチのどれを使用して実施すべきかを評価し議論するのに役立つ、いくつかの適合性フィルター属性を合わせたものである。

X3.2 モデルの概要

組織属性とプロジェクト属性は、次の3つのカテゴリーの下で評価される。

- ◆ **文化** アプローチへの賛同とチームへの信頼を基盤とした支援を得られる環境があるか。
- ◆ **チーム** チームはアジャイルの採用に成功するのに適したサイズか。メンバーは成功するために必要な経験と、ビジネス担当者へのアクセスを持っているか。
- ◆ **プロジェクト** 変更頻度が高いか。漸進的デリバリーは可能か。プロジェクトはどの程度重要か。

これらのカテゴリーのそれぞれの質問に回答し、その結果をレーダー・チャートに描く。グラフの中央付近に値が集まっている場合は、アジャイル・アプローチに適していることを示す。結果が外側にある場合は、予測型アプローチがより適切であることを示す。値が中間部分（アジャイルと予測型の間）の場合は、ハイブリッド・アプローチが適していることを示している。一例を図X3-1に示す。

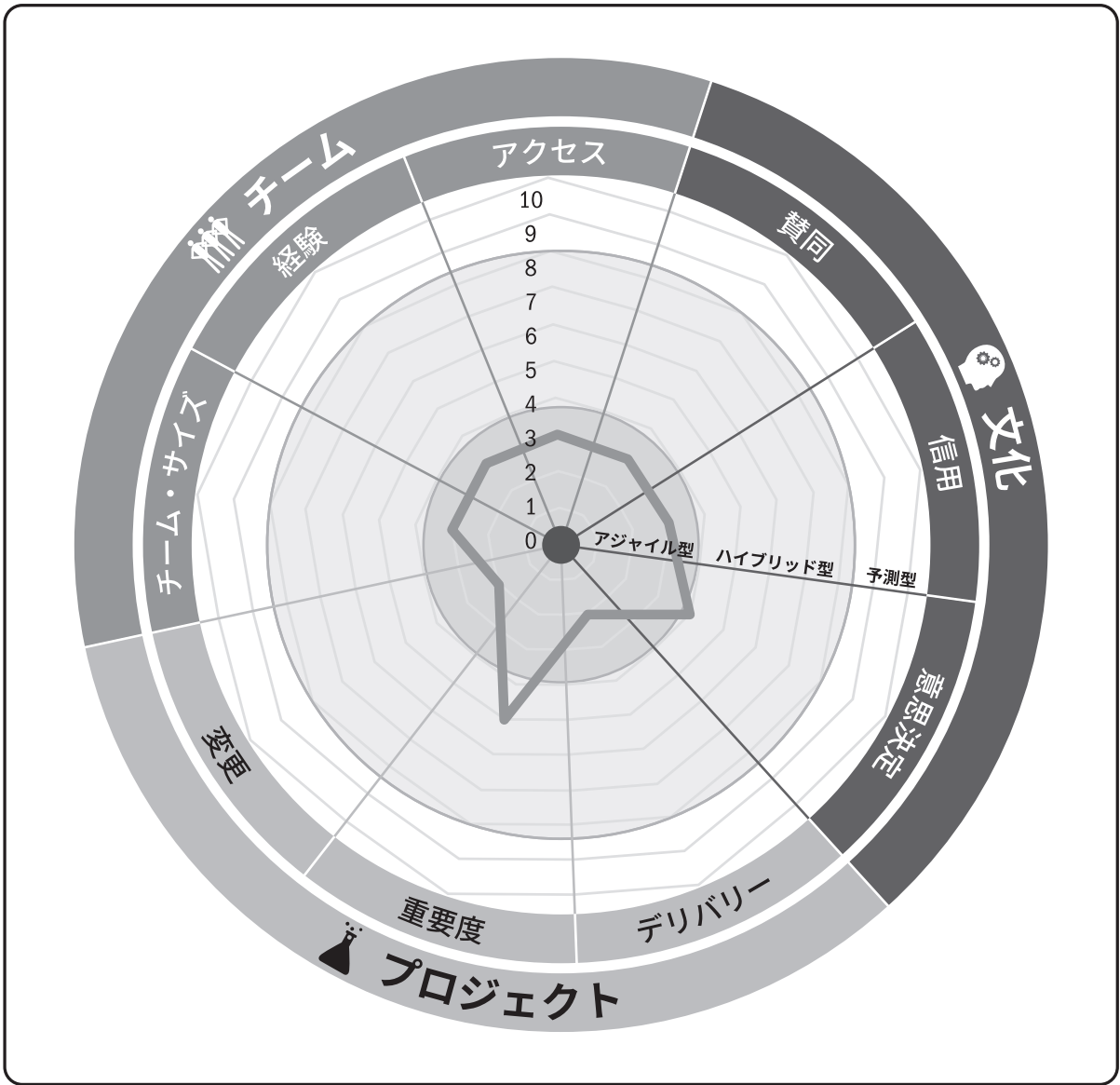


図 X3-1.アジャイル・アプローチの適合性モデル

X3.3 使用方法

X3.3.1 グループでアンケートを記入する

小規模プロジェクトの場合、このグループは単にスポンサー、テクニカル・リード、および顧客であってもよい。大規模なプロジェクトの場合、このグループには、スポンサー・グループ、プロジェクト実行チーム、影響を受けるビジネス・グループ、プロジェクト・ガバナンス・グループ、および顧客コミュニティからの代表者が含まれる。視点が限られ個人的な偏見を持っているため、単一のステークホルダーがプロジェクトを見積もったり計画したりしてはならない。また同様の理由により、ひとりの人がアプローチの適合性を評価すべきではない。

むしろこのツールの価値は、プロジェクト・スポンサーとの対話が奨励されていることにある。たとえ結果はハイブリッドアプローチを指している場合でも、ステークホルダーが主としてアジャイル・アプローチまたは予測型アプローチを進めたいと希望している場合は、ステークホルダーのコンセンサスに従う。このツールはハイレベルの診断に限定され、最終的な決定は関係者に任せられ、支持を受ける必要がある。

X3.3.2 1から10で質問を採点する

グループで質問の主観的評価を最も正確に反映するスコアについて話し合い、同意（または妥協）を得る。選択肢は、1、5、および10のスコアを表す回答スケールの開始点、中間点、および終了点として提供される。「ほぼ1であるが多少異なる」ものとして2を、または「5と10の間どこか」として7などのスコアを使ってもよいし、むしろその方が望ましい。ここでも、評価はひとつのディスカッション・ツールであり、見解は主観的で、白と黒が混ざったグレーの部分が予想される。

グループがスコアに合意できない場合は、問題をオープンかつ誠実に話し合う。妥協案を提示する（例えば、平均スコアを使用するか、またはPMOのスコアを青色の「X」で、開発チームのスコアを緑色の「0」でマーキングするなど）前に参加者が簡単な評価を完了することに同意できない場合は、プロジェクトがどの程度成功しそうかを再考する。問題の議論の際に、意見の相違を特定できているのであれば、合意に至るだろう。評価は予測型アプローチを示しているが誰もがアジャイル・アプローチを試みたい（またはその逆の）場合も同様である。問題点を理解し、アプローチの影響をどのように処理するかを議論すればよい。

X3.3.3 結果を解釈する

空白の適合性評価図に質問の回答を点として記入し、それらを結ぶ。アジャイル型ゾーンの中央付近に結果が集まっていれば、それは完全にアジャイル・アプローチに適していることを示している。

結果が主としてハイブリッド型ゾーンにあれば、アジャイル・アプローチと予測型アプローチのなんらかの組合せが最もよく機能することを示している。しかし、重要度の高いプロジェクトでは、追加の教育やトレーニング、追加の検証や文書化の厳格化など、いくつかのリスク軽減手順を伴うアジャイル・アプローチで十分な可能性もある。あるいは、概念実証 (PoC) や追加のプロセスを用いた予測型アプローチも有効である。

結果が主に予測型ゾーンにあれば、完全に予測型アプローチに適していることを示す。第X3.3.2項「質問を採点する」で述べたように、この診断ツールの目的は、影響を受ける当事者と、最も使用に適したアプローチについて有意義な対話を開始することにある。ツールが示唆するアプローチが受け入れられない場合は、別のアプローチを使用することができる。ツールはマネジメントすべき不一致を示すことから、結果をリスク・マネジメント・プロセスへのインプットとして使用する。

X3.4 適合性フィルターの質問項目

X3.4.1 カテゴリー：文化

X3.4.1.1 アプローチへの賛同

このプロジェクトにアジャイル・アプローチを使用することに、上級スポンサーから理解とサポートが得られているか。図X3-2を参照。

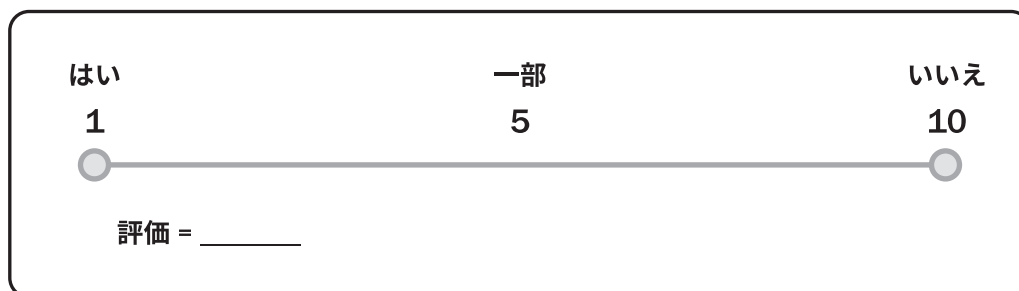


図 X3-2.アプローチへの賛同の評価

X3.4.1.2 チームへの信頼

チームと協働するスポンサーとビジネス担当者について考える。これらのステークホルダーは継続的なサポートと双方向のフィードバックを行って、チームがビジョンとニーズを、成功したプロダクトやサービスに変換することができると思いますか。図X3-3を参照。

はい
1
おそらくそう
5
おそらく無理
10
評価 = _____

図 X3-3.チームへの信頼の評価

X3.4.1.3 チームの意思決定力

作業を引き受ける方法について、チームには現場で独自の決定を下す自立性が与えられるか。図X3-4を参照。

はい
1
おそらくそう
5
おそらく違う
10
評価 = _____

図 X3-4.チームの意思決定力の評価

X3.4.2 カテゴリー：チーム

X3.4.2.1 チーム・サイズ

コア・チームのサイズはどの程度か。次のスケールを使用：1-9=1、10-20=2、21-30=3、31-45=4、46-60=5、61-80=6、81-110=7、111-150=8、151-200=9、201+=10。図X3-5を参照。

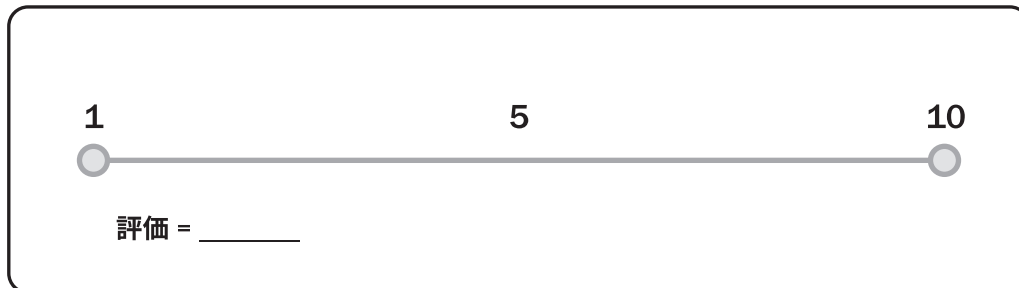


図 X3-5.チームのサイズの評価

X3.4.2.2 経験レベル

コア・チームの役割に求められる経験およびスキル・レベルについて考える。役割には経験が豊富な人と浅い人が混在しているのが普通であるが、アジャイル・プロジェクトがスムーズに進むためには、各役割に少なくともひとりの経験豊富なメンバーがいると簡単に進みやすい。各役割に経験豊富なメンバーがいるか。図X3-6を参照。

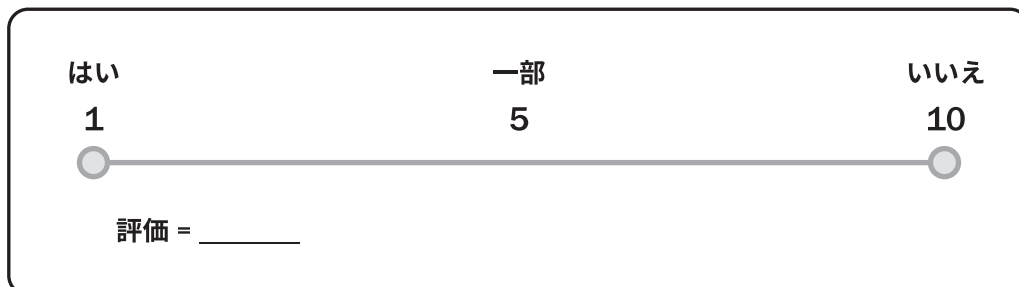


図 X3-6.経験レベルの評価

X3.4.2.3 顧客やビジネス担当者へのアクセス

チームは、少なくともひとりのビジネス担当者または顧客担当者に毎日アクセスして、質問し、フィードバックを得ることができるか。図X3-7を参照。



図 X3-7.顧客やビジネス担当者へのアクセスの評価

X3.4.3 カテゴリー：プロジェクト

X3.4.3.1 変更の可能性

変更されたり追加されたりする可能性のある要求事項は月単位で何パーセントか。図X3-8を参照。

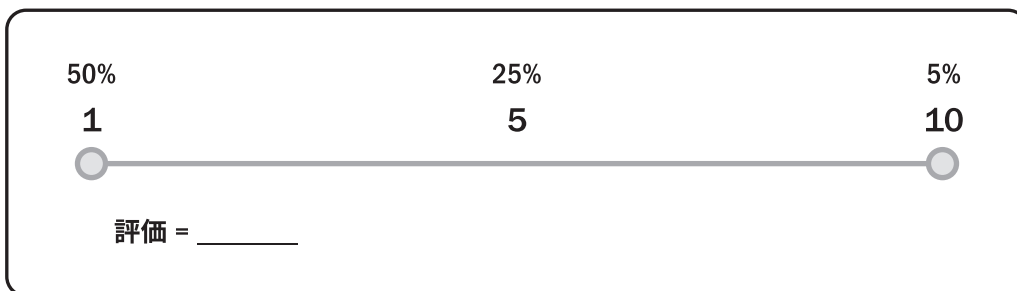


図 X3-8.変更の可能性の評価

X3.4.3.2 プロダクトまたはサービスの重要性

構築されているプロダクトまたはサービスの重要性を評価して追加の検証および文書の厳格さがどの程度必要とされそうかを判断する。欠陥が起こったときの影響による損失を考慮する評価法を使用して、失敗がどういった結果をもたらすかを判断する。図X3-9を参照。



図 X3-9.プロダクトまたはサービスの重要性の評価

X3.4.3.3 漸進型デリバリー

プロダクトまたはサービスは部分的に構築および評価できるか。また、ビジネス担当者や顧客担当者は、デリバリーされた増分に対してタイムリーなフィードバックを提供できるか。図X3-10を参照。



図 X3-10.漸進型デリバリーの評価

X3.5 適合性評価図

図X3-11は適合性評価に使用されるレーダー図である。

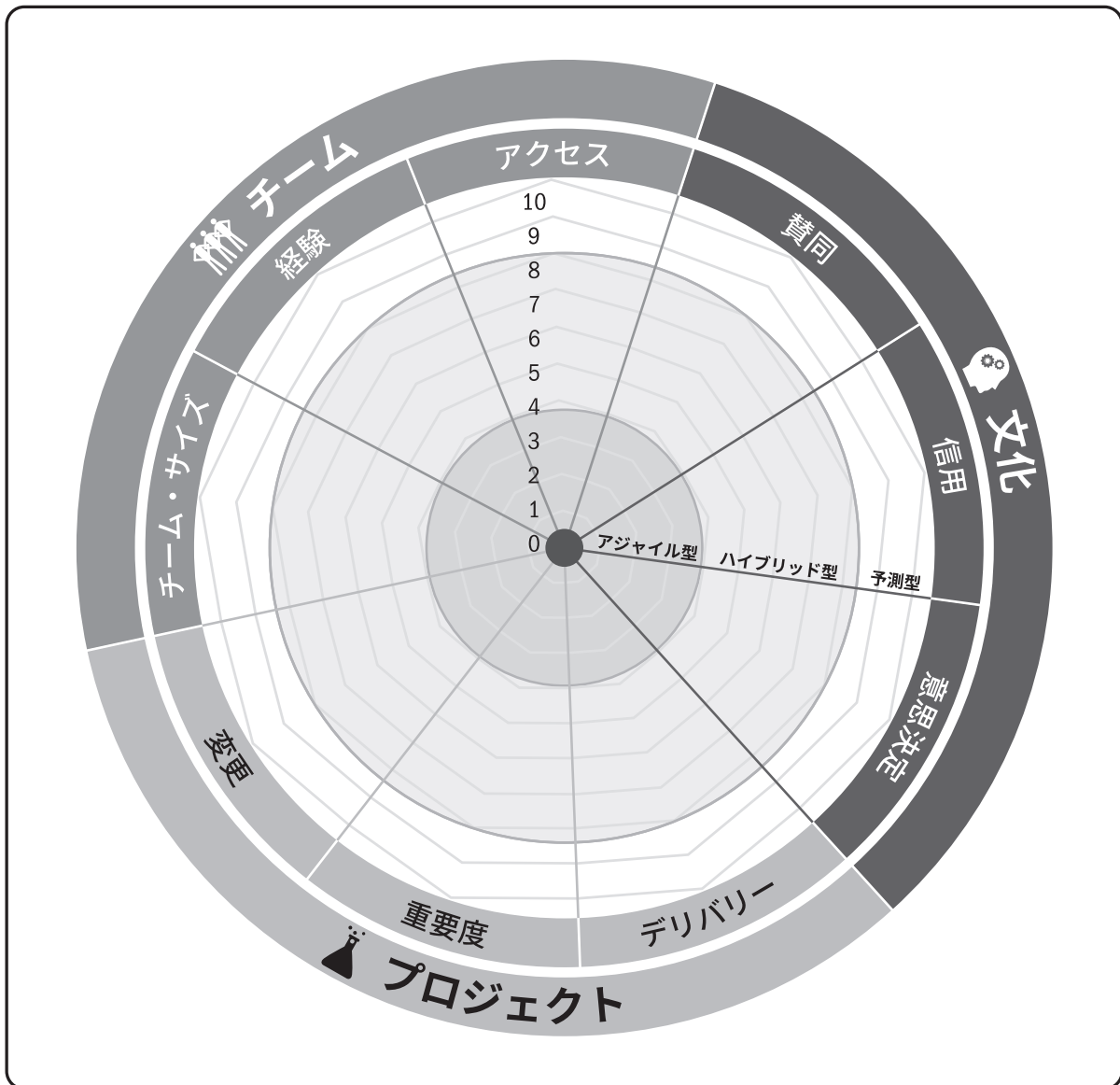


図 X3-11.適合性評価レーダー図

X3.5.1 ケーススタディ

レーダー図の仕組みを説明するために、非常に異なるタイプのプロジェクトを採点するために使用したモデルの例をふたつ示す。最初はオンライン薬局プロジェクトの例（図X3-12を参照）で、2番目は軍用メッセージング・システムの例（図X3-13を参照）である。このふたつのケーススタディは、プロジェクトごとに見られる差異を示している。中央のクラスタリングはアジャイル・アプローチが適していることを、周辺部のスコアは予測型アプローチの方が適している可能性を示している。プロジェクトによっては、中心に集まってはいるが、いくつかの軸で突き出しているものもある。これらのプロジェクトはハイブリッド・アプローチで最もよく解決される。

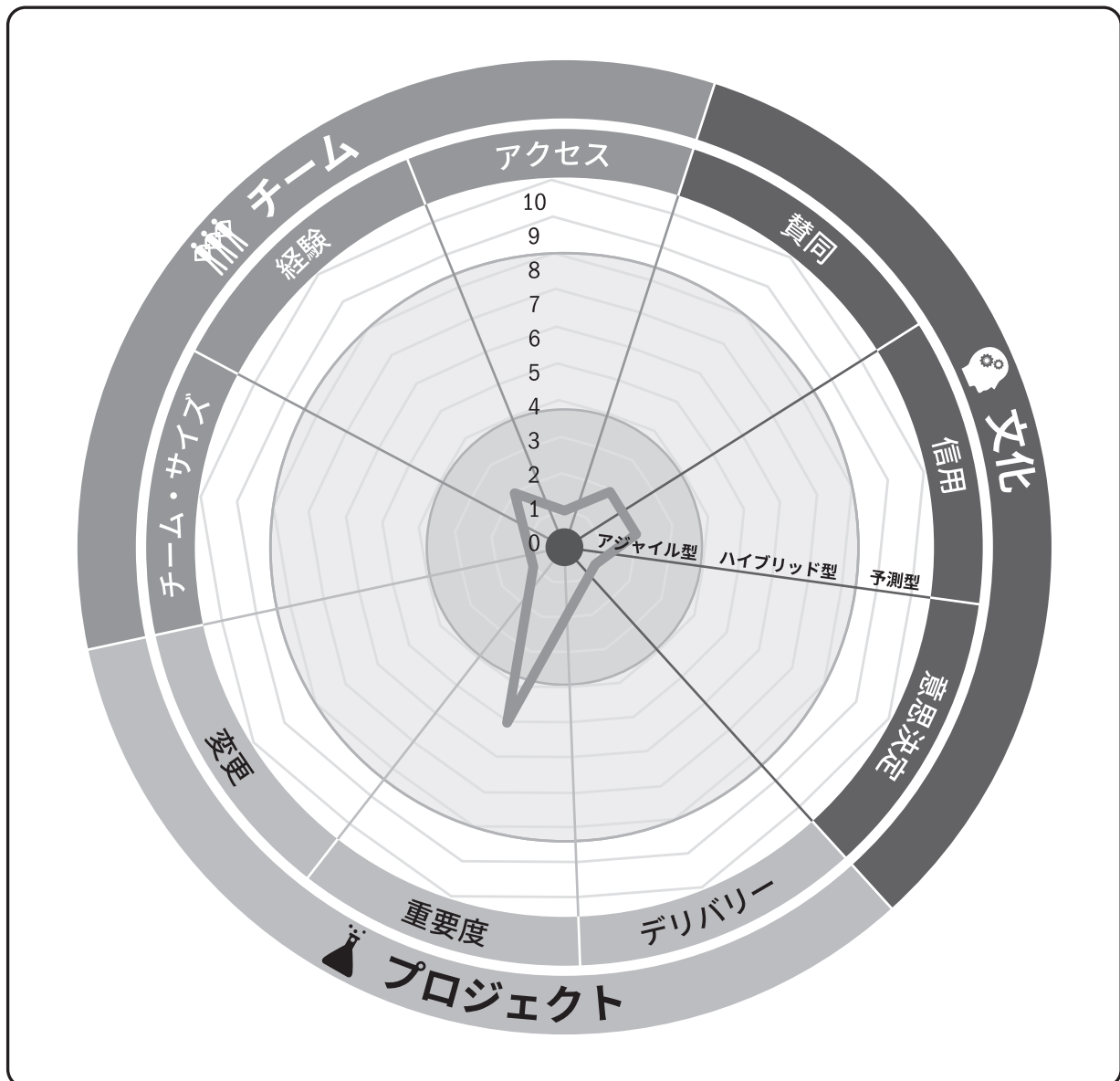


図 X3-12.薬局プロジェクト

X3.5.1.1 薬局プロジェクトの例

このプロジェクトは、カナダ産の安価な処方薬を主に米国の顧客に販売するオンライン薬局を開発するものだった。これらの医薬品の販売はカナダでも米国でも論争的となっており、その結果、業界は法規制の頻繁な変更と激しい競争にさらされている。このプロジェクトは、週ごとに大きな変更があるという非常に不安定な要求事項に直面した。そこで非常に短い（2日）イテレーションと毎週のリリースを使用して頻繁な変化に対応した。

図X3-12に示すように、モチベーションを引き出すやり方で取り組んでいる人たちの間には、高いレベルの賛同と信頼が明らかに見られる。ウェブサイトは視覚的であるため機能の新しい増分を示すことは容易だが、欠陥は薬局の必要資金を危機にさらすためシステムの重要性は相当に高かった。変更の頻度は非常に高かったが、経験を豊富に持ち合わせた小規模なチームは変更をうまく処理できた。また、知識豊富なビジネス担当者に簡単にアクセスできた。このアプローチはとりわけ成功し、非常にアジャイルであった。

X3.5.1.2 軍用メッセージング・システムの例

評価が行われた時点ですでに5年経過している軍用メッセージング・システムを開発するという大規模プロジェクトを最初の例と比較する。図X3-13を参照。

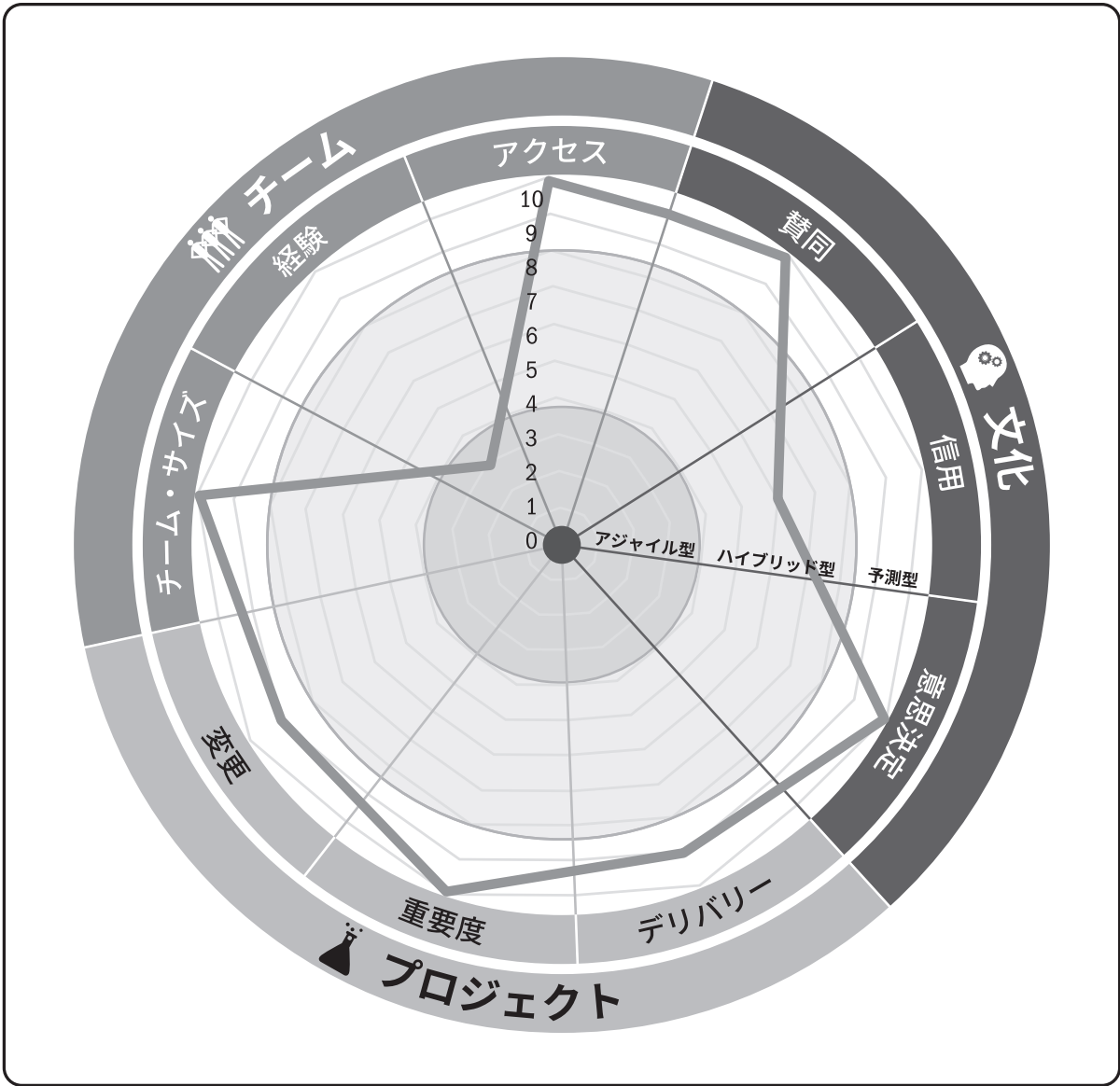


図 X3-13.軍用メッセージング・システムの例

アジャイル・アプローチは考慮されていなかったため、アジャイル・アプローチへの賛同は欠如していた。ベンダーへの信頼は入り混じっていたが、総じて敬意を得ていた。意思決定は現場ではなく、アーキテクチャと要求事項委員会によって行われた。設計の要素は実験室で漸進的にテストすることはできたが、エンド・ツー・エンドの機能のデモンストレーションのために統合することはできなかった。多くの生命が潜在的に危険にさらされており、重要性は非常に高かった。変更は非常に多くの下請け組織に影響を与えたため、要求事項は凍結された。

このプロジェクトは、ひとつのベンダーだけでも300人以上の大規模なものだったが、それぞれの役割には多くの経験豊富な実務者がいた。最後に、ビジネスや顧客へのアクセスは不可能だったが、契約アナリストは仕様上の質問をすることができ、10日以内に質問に回答するか、質問の明確化を求められた。プロジェクトの一部を切り出してアジャイル・プロジェクトとして実行することができたが、このイニシアチブの中心は単一の大きなプロジェクトだった。

X3.6 要約

アジャイル適合性フィルターは、アジャイル・アプローチへの潜在的な適合とギャップを特定するための有用なツールである。それらは採用するかしないかを決定するために使用するのではなく、すべての利害関係者との客観的な話合いのためのトピックスとして使用する必要がある。

参照文献

- [1] アジャイル型ソフトウェア開発の宣言(2001). <http://agilemanifesto.org/> より出典
- [2] Project Management Institute. 2013. 『組織のチェンジマネジメント:実務ガイド Newtown Square, PA: 著者。』
- [3] Project Management Institute. 2017. プロジェクトマネジメント知識体系ガイド(PMBOK®ガイド) – 第6版 Newtown Square, PA: 著者。
- [4] Project Management Institute. 2013. 『PMBOK®ガイド 第5版 ソフトウェア拡張版』 Newtown Square, PA: 著者。

参考文献一覧

以下は章やトピック別の追加資料である。

第2章—アジャイル入門

Briggs, Sara. “Agile Based Learning: What Is It and How Can It Change Education?” *Opencolleges.edu.au* 2014年2月22日に <http://www.opencolleges.edu.au/informed/features/agile-based-learning-what-is-it-and-how-can-it-change-education/> より出典

Manifesto for Agile Software Development, 2001, <http://agilemanifesto.org/>.

Peha, Steve. “Agile Schools: How Technology Saves Education (Just Not the Way We Thought it Would).” *InfoQ*. 2011年6月28日に <https://www.infoq.com/articles/agile-schools-education> より出典

Principles behind the Agile Manifesto, 2001, <http://agilemanifesto.org/principles.html>.

Rothman, Johanna. 2007. *Manage It! Your Guide to Modern, Pragmatic Project Management*. Raleigh: Pragmatic Bookshelf.

Sidky, Ahmed (Keynote). 2015. <https://www.slideshare.net/AgileNZ/ahmed-sidky-keynote-agilenz>.

Stacey Complexity Model. 2016. <http://www.scrum-tips.com/2016/02/17/stacey-complexity-model/>.

第3章—ライフサイクルの選択

“Agile Modeling (AM) Home Page: Effective Practices for Modeling and Documentation,” *Agile Modeling*, (日付なし), <http://www.agilemodeling.com/>

Anderson, David, and Andy Carmichael. 2016. *Essential Kanban Condensed*. Seattle: Blue Hole Press.

Anderson, David. 2010. *Kanban: Successful Evolutionary Change for Your Technology Business*. Seattle: Blue Hole Press.

- Benson, Jim, and Tonianne DeMaria Barry. 2011. *Personal Kanban: Mapping Work | Navigating Life*. Seattle: Modus Cooperandi Press.
- Burrows, Mike. 2014. *Kanban from the Inside: Understand the Kanban Method, connect it to what you already know, introduce it with impact*. Seattle: Blue Hole Press.
- Domain Driven Design Community. 2016. <http://dddcommunity.org/>.
- Gothelf, Jeff, and Josh Seiden. 2016. *Lean UX: Designing Great Products with Agile Teams*. Sebastopol: O'Reilly Media.
- Hammarberg, Marcus, and Joakim Sunden. 2014. *Kanban in Action*. Shelter Island: Manning Publications.
- “Kanban,” *Wikipedia*、2016年11月22日に <https://en.wikipedia.org/wiki/Kanban> より出典、2017年5月4日に改訂
- “Kanban (development),” *Wikipedia*、2016年11月29日に [https://en.wikipedia.org/wiki/Kanban_\(development\)](https://en.wikipedia.org/wiki/Kanban_(development)) より出典、2017年5月4日に改訂
- Larsen, Diana, and Ainsley Nies. 2016. *Liftoff: Start and Sustain Successful Agile Teams*. Raleigh: Pragmatic Bookshelf.
- “Learning Kanban,” *Leankit*、(日付なし)、<https://leankit.com/learn/learning-kanban/>.
- Leopold, Klaus, and Siegfried Kaltenecker. 2015. *Kanban Change Leadership: Creating a Culture of Continuous Improvement*. Hoboken: Wiley.
- “Make a big impact with software products and projects!” *Impact Mapping*、(日付なし)、<https://www.impactmapping.org/>.
- Patton, Jeff, and Peter Economy. 2014. *User Story Mapping: Discover the Whole Story, Build the Right Product*. Sebastopol: O'Reilly Media.
- Reinertsen, Donald. 2009. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Redondo Beach: Celeritas Publishing.
- Rothman, Johanna. “Dispersed vs. Distributed Teams,” *Rothman Consulting Group, Inc.*、2010年10月25日、<http://www.jrothman.com/mpd/2010/10/dispersed-vs-distributed-teams/>.
- Schwaber, Ken, and Jeff Sutherland. “The Scrum Guide™,” *Scrum.org*、2016年7月、<http://www.scrumguides.org/scrum-guide.html> および <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>.
- Skarin, Mattias. 2015. *Real-World Kanban: Do Less, Accomplish More with Lean Thinking*. Raleigh: Pragmatic Bookshelf.
- “The High Cost of Multitasking: 40% of Productivity Lost by Task Switching,” *Wrike.com*、2015年9月24日、<https://www.wrike.com/blog/high-cost-of-multitasking-for-productivity/>.
- Wells, Don. “Extreme Programming: A Gentle Introduction,” *Extreme Programming*、2013年10月8日、<http://www.extremeprogramming.org/>.

第4章—アジャイルの実装:

Amabile, Teresa, and Steven Kramer. 2011. *The Progress Principle: Using Small Wins to Ignite Joy, Engagement, and Creativity at Work*. Boston: Harvard Business Review Press.

“Early Warning Signs of Project Trouble—Cheat Sheet, 2017年、<https://agilevideos.com/wp-content/uploads/2017/02/WarningSignsOfProjectTrouble-CheatSheet.pdf>.

Dweck, Carol. 2006. *Mindset: The New Psychology of Success*. New York: Penguin Random House.

Kaner, Sam. *Facilitator's Guide to Participatory Decision-Making*. 3rd ed. 2014. San Francisco: Jossey-Bass.

Keith, Kent. *The Case for Servant Leadership*. 2008. Westfield: Greenleaf Center for Servant Leadership.

Rothman, Johanna. 2016. *Agile and Lean Program Management: Scaling Collaboration Across the Organization*. Victoria, British Columbia: Practical Ink.

Rothman, Johanna. “Dispersed vs. Distributed Teams,” *Rothman Consulting Group, Inc.*, 2010年10月25日、<http://www.jrothman.com/mpd/2010/10/dispersed-vs-distributed-teams/>.

Rothman, Johanna. 2007. *Manage It! Your Guide to Modern, Pragmatic Project Management*. Raleigh: Pragmatic Bookshelf.

Rothman, Johanna. 2016. *Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects*. Raleigh: Pragmatic Bookshelf.

Schwaber, Ken, and Jeff Sutherland. “The Scrum Guide™,” *Scrum.org*, 2016年7月、<http://www.scrumguides.org/scrum-guide.html> および <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>.

Sinek, Simon. 2011. *Start with Why: How Great Leaders Inspire Everyone to Take Action*. New York: Portfolio, Penguin Random House.

“The High Cost of Multitasking: 40% of Productivity Lost by Task Switching,” *Wrike.com*, 2015年9月24日、<https://www.wrike.com/blog/high-cost-of-multitasking-for-productivity/>.

体験レポート:

“Experience Reports,” *Agile Alliance*, (日付なし)、<https://www.agilealliance.org/resources/experience-reports/>.

プロジェクトとチームの健全性:

“Early Warning Signs of Project Trouble—Cheat Sheet.” 2017. <https://agilevideos.com/wp-content/uploads/2017/02/WarningSignsOfProjectTrouble-CheatSheet.pdf>

“TeamHealth Radar – Summary View,” *Agilehealth*. 2014. <http://agilityhealthradar.com/wp-content/uploads/2014/11/bigradar.gif>.

資源の効率：

Modig, Niklas, and Pär Åhlström. 2015. *This is Lean: Resolving the Efficiency Paradox*. London: Rheologica Publishing.

Rothman, Johanna. "Resource Efficiency vs. Flow Efficiency, Part 5: How Flow Changes Everything," *Rothman Consulting Group, Inc.*, 2015年9月20日、<http://www.jrothman.com/mpd/agile/2015/09/resource-efficiency-vs-flow-efficiency-part-5-how-flow-changes-everything/>.

スケーリング：

Disciplined Agile 2.X—A Process Decision Framework. 2016. <http://www.disciplinedagiledelivery.com/>.

Kniberg, Henrik. "Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds," *Crisp*, 2012年11月14日、<http://blog.crisp.se/2012/11/14/henrikkniberg/scaling-agile-at-spotify>.

"Overview—Large Scale Scrum," *LeSS*. 2016年。 <http://less.works/>.

"SAFe® for Lean Software and System Engineering," *SAFe®*. 2016. <http://www.scaledagileframework.com/>.

スキル：

Beck, Kent. *Paint Drip People*. 2016年8月4日、<https://www.facebook.com/notes/kent-beck/paint-drip-people/1226700000696195/>.

"Generalizing Specialists: Improving Your IT Career Skills," *Agile Modeling*, (日付なし)、<http://www.agilemodeling.com/essays/generalizingSpecialists.htm>.

Hunter, Brittany. "Of Software Designers & Broken Combs," *Atomic Object*, 2013年6月27日、<https://spin.atomicobject.com/2013/06/27/broken-comb-people/>.

第5章—アジャイルの実装:アジャイル環境での引渡し

Larsen, Diana, and Ainsley Nies. 2016. *Liftoff: Start and Sustain Successful Agile Teams*. Raleigh: Pragmatic Bookshelf.

レトロスペクティブ：

Derby, Esther, and Diana Larsen. 2006. *Agile Retrospectives: Making Good Teams Great*. Raleigh: Pragmatic Bookshelf.

Gonçalves, Luis, and Ben Linders. 2015. *Getting Value out of Agile Retrospectives: A Toolbox of Retrospective Exercises*. Victoria, British Columbia: Leanpub.

バックログ：

Adzic, Gojko, Marjory Bissett, and Tom Poppendieck. 2012. *Impact Mapping: Making a Big Impact with Software Products and Projects*. Woking, Surrey: Provoking Thoughts.

Patton, Jeff, and Peter Economy. 2014. *User Story Mapping: Discover the Whole Story, Build the Right Product*. Sebastopol: O'Reilly Media.

Rothman, Johanna. "We Need Planning; Do We Need Estimation?" *Rothman Consulting Group, Inc.*、2015年1月21日、<http://www.jrothman.com/mpd/project-management/2015/01/we-need-planning-do-we-need-estimation/>.

スタンドアアップ：

Brodzinski, Pawel. "Effective Standups around Kanban Board," *Brodzinski.com*、2011年12月30日、<http://brodzinski.com/2011/12/effective-standups.html>.

Fowler, Martin. "It's Not Just Standing Up: Patterns for Daily Standup Meetings," *Martinfowler.com*、2016年2月21日、<http://martinfowler.com/articles/itsNotJustStandingUp.html>.

Hefley, Chris. "How to Run Effective Standups and Retrospectives," *Leankit*、2014年9月15日、<https://leankit.com/blog/2014/09/run-effective-standups-retrospectives/>.

アーンドバリュー：

Griffiths, Mike. "A Better S Curve and Simplified EVM," *Leading Answers*、2008年6月6日、http://leadinganswers.typepad.com/leading_answers/2008/06/a-better-s-curve-and-simplified-evm.html.

第6章—アジャイル・プロジェクトに関する組織的な考慮事項

Bankston, Arlen, and Sanjiv Augustine. *Agile Team Performance Management: Realizing the Human Potential of Teams*、2010年6月14日、www.lithespeed.com/transfer/Agile-Performance-Management.pptx.

Browder, Justin, and Brian Schoeff. *Perfect Strangers: How Project Managers and Developers Relate and Succeed*. CreateSpace Independent Publishing Platform, 2016, <https://www.createspace.com/>.

Griffiths, Mike. "Agile Talent Management," *Leading Answers*、2015年10月14日、http://leadinganswers.typepad.com/leading_answers/2015/10/agile-talent-management.html.

Kohn, Alfie. 1999. *Punished by Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes*. New York: Mariner Books.

Mar, Kane. "How to do Agile Performance Reviews," *Scrumology*、(日付なし)、<https://scrumology.com/how-to-do-agile-performance-reviews/>.

McChrystal, Stanley, Tatum Collins, David Silverman, and Chris Fussell. 2015. *Team of Teams: New Rules of Engagement for a Complex World*. New York: Portfolio, Penguin Random House.

Pink, Daniel. 2011. *Drive: The Surprising Truth About What Motivates Us*. New York: Riverhead Books.

第7章—行動の喚起

Dennis, Pascal. 2006. *Getting the Right Things Done: A Leader's Guide to Planning and Execution*. Cambridge: Lean Enterprise Institute.

Griffiths, Mike. "Introducing Agile Methods: Mistakes to Avoid—Part 3," *Leading Answers*, 2007年3月15日、http://leadinganswers.typepad.com/leading_answers/2007/03/introducing_agi_2.html.

Little, Jason. *Lean Change Management: Innovative Practices for Managing Organizational Change*. Happy Melly Express, 2014年、<http://www.happymelly.com/category/hm-express/>.

Rising, Linda, and Mary Lynne Manns. 2004. *Fearless Change: Patterns for Introducing New Ideas*. Upper Saddle River: Addison-Wesley Professional.

"The IDEAL Model," *Software Engineering Institute, Carnegie Mellon*, 2006, <http://www.sei.cmu.edu/library/assets/idealmodel.pdf>.

付録A1—PMBOK® ガイド マッピング

Larsen, Diana and Ainsley Nies. 2016. *Liftoff: Start and Sustain Successful Agile Teams*. Raleigh: Pragmatic Bookshelf.

付録A2—アジャイル宣言とのマッピング

Manifesto for Agile Software Development, 2001年、<http://agilemanifesto.org/>.

Principles behind the Agile Manifesto, 2001年、<http://agilemanifesto.org/principles.html>.

付録A3—アジャイルおよびリーン・フレームワークの概要

Agile Business Consortium, 2014年、<https://www.agilebusiness.org/what-is-dsdm>.

Ambler, Scott. "The Agile Unified Process," *Ambyssoft*, 2006年5月13日、<http://www.ambyssoft.com/unifiedprocess/agileUP.html>.

Anderson, David. 2010. *Kanban: Successful Evolutionary Change for Your Technology Business*. Seattle: Blue Hole Press.

Beedle, Mike. *Enterprise Scrum: Executive Summary: Business Agility for the 21st Century*, 2017年1月7日、<http://www.enterprisescrum.com/enterprise-scrum/>.

Cockburn, Alistair. 2004. *Crystal Clear: A Human-Powered Methodology for Small Teams*. Upper Saddle River: Pearson Education.

Cockburn, Alistair. "Crystal Methodologies," *alistair.cockburn.us*. 2014年3月28日、<http://alistair.cockburn.us/Crystal+methodologies>.

Disciplined Agile 2.X—A Process Decision Framework, 2016年、<http://www.disciplinedagiledelivery.com/>.

Joint MIT-PMI-INCOSE Community of Practice on Lean in Program Management. 2012. *The Guide to Lean Enablers for Managing Engineering Programs*. Newtown Square, PA: Author.

"Kanban," *Wikipedia*, 2016年11月22日に <https://en.wikipedia.org/wiki/Kanban> より出典、2017年5月4日に改訂

"Kanban (*development*)," *Wikipedia*, 2016年11月29日に [https://en.wikipedia.org/wiki/Kanban_\(development\)](https://en.wikipedia.org/wiki/Kanban_(development)) より出典、2017年5月4日に改訂

Reddy, Ajay, and Jack Speranza. 2015. *The Scrumban [R]Evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban*. Boston: Addison-Wesley Professional.

"Overview—Large Scale Scrum," *LeSS*, 2016年、<http://less.works/>.

"SAFe® for Lean Software and System Engineering," *SAFe®*, 2016年、<http://www.scaledagileframework.com/>.

Schwaber, Ken, and Jeff Sutherland. "The Scrum Guide™," *Scrum.org*, 2016年7月、<http://www.scrumguides.org/scrum-guide.html> および <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>.

"Scrum of Scrums," *Agile Alliance*, (日付なし)、<https://www.agilealliance.org/glossary/scrum-of-scrums/>.

"Scrumban," *Wikipedia*, 2017年3月2日、<https://en.wikipedia.org/wiki/Scrumban>.

"State of Agile Report: Agile Trends," *VersionOne*, 2017, <http://stateofagile.versionone.com/>.

Sutherland Jeff. "Agile Can Scale: Inventing and Reinventing SCRUM in Five Companies." *Cutter IT Journal* 14, no. 12 (2001): 5–11. http://www.controlchaos.com/storage/scrum-articles/Sutherland_200111_proof.pdf.

"The 2015 State of Agile Development," *Scrum Alliance®*, 2015, <https://www.forrester.com/report/The+2015+State+Of+Agile+Development/-/E-RES122910>

Wells, Don. "Extreme Programming: A Gentle Introduction," *Extreme Programming*, 2013年10月8日、<http://www.extremeprogramming.org/>.

Why Scrum? State of Scrum Report, 2016年、<https://www.scrumalliance.org/why-scrum/state-of-scrum-report/2016-state-of-scrum>.

付属文書X2—テラリングに影響を与える属性

Griffiths, Mike. “Agile Suitability Filters,” *Leading Answers*, 2007年、http://leadinganswers.typepad.com/leading_answers/files/agile_suitability_filters.pdf.

Jeffries, Ron. “We Tried Baseball and It Didn’t Work,” *ronjeffries.com*, 2006年5月2日、<http://ronjeffries.com/xprog/articles/jatbaseball/>.

Rothman, Johanna. “One Experimental Possibility: Self-Organization from Component Teams to Feature Teams,” *Rothman Consulting Group, Inc.*, 2014年9月23日、<http://www.jrothman.com/mpd/agile/2014/09/one-experimental-possibility-self-organization-from-component-teams-to-feature-teams/>.

用語集

1. 略語

ATDD	受入れテスト駆動開発
BDD	振舞い駆動型開発
BRD	ビジネス要件文書
DA	統制されたアジャイル
DoD	Done(完了)の定義
DoR	Ready(準備完了)の定義
DSDM	ダイナミック・システム開発アプローチ
EVO	価値の漸進的創出
LeSS	大規模スクラム
LSD	リーン・ソフトウェア開発
PDCA	プラン-ドゥ-チェック-アクト
PMO	プロジェクトマネジメント・オフィス
ROI	投資収益率
RUP	ラショナル統一プロセス
SAFe®	Scaled Agile Framework®
SBE	実例による仕様
XP	エクストリーム・プログラミング

2. 定義

A3. 関連する情報を1枚のA3サイズの手紙上にまとめて考えるアプローチや体系的な問題解決プロセス。

A way of thinking and a systematic problem-solving process that collects the pertinent information on a single A3-size sheet of paper.

DevOps. 開発と運用スタッフ間の協業状況を改善することによって、引渡しの円滑なフローを作るための実務慣行の集合。

A collection of practices for creating a smooth flow of delivery by improving collaboration between development and operations staff.

Doneの定義 (DoD: Definition of Done). 成果物が顧客の使用のために準備が整っていると考えられ、満たされるべく要求されたすべての基準を備えたチーム用チェックリスト。

A team's checklist of all the criteria required to be met so that a deliverable can be considered ready for customer use.

IDEAL. それが記述する5つのフェーズ（すなわちinitiating（開始）、diagnosing（診断）、establishing（確立）、acting（行動）およびlearning（学習））にちなんで名付けられた組織の改善モデル。

An organizational improvement model that is named for the five phases it describes: initiating, diagnosing, establishing, acting, and learning.

I字型 (I-shaped). チームが必要とするスキル領域における単一分野には深い専門性を有するが、他の部分には関心もスキルもない人物を指す。「T字型」と「折れたクシ」も参照。

Refers to a person with a single deep area of specialization and no interest or skill in the rest of the skills required by the team. See also *T-Shaped* and *Broken Comb*.

Readyの定義 (DoR: Definition of Ready). チームが作業を開始可能とするために必要なすべての情報を備えた、ユーザーセントリックな要求のためのチーム用チェックリスト。

A team's checklist for a user-centric requirement that has all the information the team needs to be able to begin working on it.

T字型 (T-shaped). チームが必要とするスキル領域において、単一分野に深い専門性を有しながら、他の分野にも広範な能力を有する人物を指す。「I字型」と「折れたクシ」も参照。

Refers to a person with one deep area of specialization and broad ability in the rest of the skills required by the team. See also *I-Shaped* and *Broken Comb*.

UXデザイン (UX Design). ユーザーとプロダクト間の相互作用に見られるユーザビリティとアクセサビリティの改善に注力することによって、ユーザー・エクスペリエンスを向上させるプロセス。

The process of enhancing the user experience by focusing on improving the usability and accessibility to be found in the interaction between the user and the product.

あ

アジャイリスト (Agilist). 「アジャイル実務者」を参照。

See *Agile Practitioner*.

アジャイル (Agile). アジャイル宣言に記載されている価値と原則の考え方を記述するのに使用される用語。

A term used to describe a mindset of values and principles as set forth in the Agile Manifesto.

アジャイル型ライフサイクル (Agile Life Cycle). 作業項目を絞り込んで頻繁に創出する反復型でもあり漸進型でもあるアプローチ。

An approach that is both iterative and incremental to refine work items and deliver frequently.

アジャイル・コーチ (Agile Coach). 変革を通して組織やチームをトレーニングし、メンタリングし、そしてガイドすることができるアジャイルの知識をもち経験のある個人。

An individual with knowledge and experience in agile who can train, mentor, and guide organizations and teams through their transformation.

アジャイル実務者 (Agile Practitioner). 機能横断的なチームにおいて、同じ考え方をもっている仲間と協業し、アジャイルのマインドセットを積極的に取り入れている人物。アジャイリストとも呼ばれる。

A person embracing the agile mindset who collaborates with like-minded colleagues in cross-functional teams. Also referred to as agilist.

アジャイル宣言 (Agile Manifesto). アジャイルの価値と原則についての最初の公式な定義。

The original and official definition of agile values and principles.

アジャイル統一プロセス (Agile Unified Process). アジャイルの技法や概念を使用して、ビジネス・アプリケーション・ソフトウェアを開発するための単純で理解しやすいアプローチ。それはラショナル統一プロセス(RUP)の簡易版である。

A simplistic and understandable approach to developing business application software using agile techniques and concepts. It is a simplified version of the Rational Unified Process (RUP).

アジャイルの原則 (Agile Principles). アジャイル宣言に具現化されているアジャイル型プロジェクト開発の12の原則。

The twelve principles of agile project delivery as embodied in the Agile Manifesto.

アジャイルのマインドセット (Agile Mindset). アジャイル宣言における4つの価値と12の原則によって支えられた考え方と振舞い。

A way of thinking and behaving underpinned by the four values and twelve principles of the Agile Manifesto.

アジャイルの融合 (Blended Agile). XPとカンバン方式を組み合わせるスクラムのように、ふたつ以上のアジャイルのフレームワーク、方法、要素、または実務慣行を一緒に使用する。

Two or more agile frameworks, methods, elements, or practices used together, such as Scrum practiced in combination with XP and Kanban Method.

アンチパターン (Anti-Pattern). 推奨できないような、周知はされているが不備のある作業パターン。

A known, flawed pattern of work that is not advisable.

い

イテレーション (Iteration). 価値を創出するのに必要な作業のすべてが実行される、プロダクトまたは成果物の開発のためにタイムボックス化されたサイクル。

A timeboxed cycle of development on a product or deliverable in which all of the work that is needed to deliver value is performed.

インパクト・マッピング (Impact Mapping). 新プロダクトの開発中に組織へのロードマップとして機能する戦略的計画技法。

A strategic planning technique that acts as a roadmap to the organization while building new products.

う

受入れテスト駆動開発 (ATDD: Acceptance Test-Driven Development). 引渡しの開始前に、受入れテストを作成するために使用される受入テスト基準を協業して作成する方法。

A method of collaboratively creating acceptance test criteria that are used to create acceptance tests before delivery begins.

え

エクストリーム・プログラミング (eXtreme Programming). より高品質のソフトウェア、変化する顧客の要求事項へのより優れた応答性、そしてより短いサイクルでのより頻繁なリリースへと導くアジャイル型ソフトウェア開発アプローチ。

An agile software development method that leads to higher-quality software, greater responsiveness to changing customer requirements, and more frequent releases in shorter cycles.

お

折れたクシ (Broken Comb). チームが必要とする複数のスキルにおいて、さまざまな深さの専門性を有する人を指す。ペイント・ドリップとしても知られている。「T字型」と「I字型」も参照。

Refers to a person with various depths of specialization in multiple skills required by the team. Also known as paint drip. See also *T-shaped* and *I-shaped*.

か

カイゼンのイベント (Kaizen Events). システムの改善を目指すイベント。

Events aimed at improvement of the system.

価値の漸進的創出 (Evo: Evolutionary Value Delivery). 複数の測定可能な価値要求事項をステークホルダーに提供することに注力という、他の方法にはない特定のコンポーネントを含んでいる最初のアジャイルアプローチとして広く認められている。

Openly credited as the first agile method that contains a specific component no other methods have: the focus on delivering multiple measurable value requirements to stakeholders.

カデンツ (Cadence). 実行のリズム。「タイムボックス」も参照。

A rhythm of execution. See also *Timebox*.

カンバン (Kanban Board). ボトルネックと作業量を見える化作業フローを改善できる可視化ツール。

A visualization tool that enables improvements to the flow of work by making bottlenecks and work quantities visible.

カンバン方式 (Kanban Method). オリジナルのかんばん在庫管理システムに触発された、特に知的作業に使用されるアジャイル方式。

An agile method inspired by the original Kanban inventory control system and used specifically for knowledge work.

き

技術的負債 (Technical Debt). プロダクト・ライフサイクルのそれまでの時点で実行されなかった作業の繰り延べ費用。

The deferred cost of work not done at an earlier point in the product life cycle.

機能横断チーム (Cross-Functional Team). 価値あるプロダクト増分を創出するのに必要なすべてのスキルを備えた実務者を含むチーム。

A team that includes practitioners with all the skills necessary to deliver valuable product increments.

機能仕様 (Functional Specification). システムやアプリケーションが果たすよう求められる特定の機能。通常は機能仕様書に描かれる。

A specific function that a system or application is required to perform. Typically represented in a functional specifications document.

機能要求 (Functional Requirement). プロダクトまたはサービスが果たすべき固有の振舞い。

A specific behavior that a product or service should perform.

く

クリスタル・ファミリー方法論 (Crystal Family of Methods). 特別な事情への適応性に焦点を当てた軽量のアジャイル型ソフトウェア開発アプローチの集合体。

A collection of lightweight agile software development methods focused on adaptability to a particular circumstance.

け

計画駆動型アプローチ (Plan-Driven Approach). 「予測型アプローチ」 参照。

See *Predictive Approach*.

継続的統合 (Continuous Integration). すべてのチーム・メンバーの作業プロダクトを頻繁に統合し、他のメンバーと妥当性を確認しあう実務慣行。

A practice in which all team members' work products are frequently integrated and validated with one another.

継続的引渡し (Continuous Delivery). しばしば、作業を小さなバッチとし、自動化技術を使用することによって、フィーチャーの増分をすぐさま顧客に引き渡す実務慣行。

The practice of delivering feature increments immediately to customers, often through the use of small batches of work and automation technology.

さ

サイロ化した組織 (Siloed Organization). 顧客に価値を提供するために必要とする中で、ある側面のサブセットについてだけマネジメントするように構成された組織。対照のため、「バリュー・ストリーム」を参照。

An organization structured in such a way that it only manages to contribute a subset of the aspects required for delivering value to customers. For contrast, see *Value Stream*.

サーバント・リーダーシップ (Servant Leadership). チームのパフォーマンスを可能な限り最大化するために、チーム・メンバーのニーズと育成について理解し取り組むことに焦点を当てる、チームへのサービスを通してリードする実務慣行。

The practice of leading through service to the team, by focusing on understanding and addressing the needs and development of team members in order to enable the highest possible team performance.

サービス・リクエスト・マネジャー (Service Request Manager). 継続的フローやカンバンの環境において価値を最大化するために、サービス・リクエストを注文する人物。プロダクト・オーナーと同等。

The person responsible for ordering service requests to maximize value in a continuous flow or Kanban environment. Equivalent to product owner.

し

自己組織化チーム (Self-Organizing Team). 機能横断的なチームで、チームの目標を達成するためにメンバーが柔軟に必要に応じてリーダーシップを発揮することを当然と認識している。

A cross-functional team in which people fluidly assume leadership as needed to achieve the team's objectives.

実例による仕様 (SBE: Specification by Example). 抽象的な記述ではなく現実的な例を使用した要求事項の収集と図示に基づいて、ソフトウェア・プロダクトの要求事項とビジネス指向の機能テストを定義するための協業的アプローチ。

A collaborative approach to defining requirements and business-oriented functional tests for software products based on capturing and illustrating requirements using realistic examples instead of abstract statements.

自動化されたコード品質分析 (Automated Code Quality Analysis). バグや脆弱性のためのコード・ベースのスクリプト・テスト。

The scripted testing of code base for bugs and vulnerabilities.

集団的コード所有権 (Collective Code Ownership). プロジェクトの加速と協業のための技法。いかなるチーム・メンバーもプロジェクト作業プロダクトや成果物を改訂する権限をもつことによって、チーム全体のオーナーシップと説明責任を強調している。

A project acceleration and collaboration technique whereby any team member is authorized to modify any project work product or deliverable, thus emphasizing team-wide ownership and accountability.

使用適合 (Fit for Use). 本来の目的を達成するために、現在の形で使用可能なプロダクトを記述する。

Describes a product that is usable in its current form to achieve its intended purpose.

障害 (Impediment). チームの目標達成を阻止する障害。ブロッカーとも呼ばれる。

An obstacle that prevents the team from achieving its objectives. Also known as a blocker.

情報ラジエーター (Information Radiator). 組織内のすべての人に情報を提供するための可視的な物理的ディスプレイで、チームに余計な負荷をかけることなく最新知識の共有を可能にする。

A visible, physical display that provides information to the rest of the organization, enabling up-to-the-minute knowledge sharing without having to disturb the team.

シングルループ学習 (Single-Loop Learning). 経験に照らしてさまざまな方法にチャレンジするのではなく、事前に定義された特定の方法だけを使用することによって問題を解決しようとする実務慣行。

The practice of attempting to solve problems by just using specific predefined methods, without challenging the methods in light of experience.

す

スウォーミング (Swarming). 複数のチーム・メンバーが、特定の障害を解決するために集合的に注力する技法。

A technique in which multiple team members focus collectively on resolving a specific impediment.

スクラム (Scrum). 特定の役割、イベント、および生成物で複雑なプロダクトを開発し維持するための、アジャイルのフレームワーク。

An agile framework for developing and sustaining complex products, with specific roles, events, and artifacts.

スクラム・チーム (Scrum Team). スクラムで使用される開発チーム、スクラム・マスター、およびプロダクト・オーナーの組合せを意味する。

Describes the combination of development team, Scrum Master, and product owner used in Scrum.

スクラムのスクラム (Scrum of Scrums). 特に重複する分野において、複数のチームがひとつのプロダクトに取り組み、その相互依存性に関わる進捗についての議論を調整し、ソフトウェアの引渡しを統合する方法に焦点を当てる、大規模なスクラムを運用する技法。

A technique to operate Scrum at scale for multiple teams working on the same product, coordinating discussions of progress on their interdependencies, and focusing on how to integrate the delivery of software, especially in areas of overlap.

スクラムバン (Scrumban). チームが作業アプローチとしてスクラムを採用し、それらがどう機能しているかを見て、理解し、継続的に改善するためのレンズとしてカンバン方式を使用するときに明らかになるマネジメント・フレームワーク。

A management framework that emerges when teams employ Scrum as the chosen way of working and use the Kanban Method as a lens through which to view, understand, and continuously improve how they work.

スクラム・ボード (Scrum Board). プロダクトとスプリント・バックログをマネジメントし、作業フローとそのボトルネックを表示するために利用される情報ラジエーター。

An information radiator that is utilized to manage the product and sprint backlogs and show the flow of work and its bottlenecks.

スクラム・マスター (Scrum Master). スクラム・フレームワークにおける、開発チームとプロセス・オーナーのコーチ。障害を取り除き、生産的なイベントを促進し、チームを混乱から守る。「フロー・マスター」も参照。

The coach of the development team and process owner in the Scrum framework. Removes obstacles, facilitates productive events and defends the team from disruptions. See also *Flow Master*.

スケールド・アジャイル・フレームワーク (SAFe®: Scaled Agile Framework). 組織体規模でリーン・アジャイル開発のために統合されたパターンの知識ベース。

A knowledge base of integrated patterns for enterprise-scale lean-agile development.

ストーリー・ポイント (Story Point). 相対的ユーザー・ストーリー見積り技法で使用される、単位の無い尺度。

A unit-less measure used in relative user story estimation techniques.

スパイク (Spike). プロジェクト内の、通常は長さが固定されている短い時間間隔で、その間にチームはリサーチを実施したり、その実行可能性を証明するためにソリューションの一側面のプロトタイプを作成したりする。

A short time interval within a project, usually of fixed length, during which a team conducts research or prototypes an aspect of a solution to prove its viability.

スプリント (Sprint). スクラムにおけるタイムボックス化されたイテレーションを記述する。

Describes a timeboxed iteration in Scrum.

スプリント計画 (Sprint Planning). スクラム・チームが現在のスプリントのために作業を計画している、スクラム内の協業的イベント。

A collaborative event in Scrum in which the Scrum team plans the work for the current sprint.

スプリント・バックログ (Sprint Backlog). スクラムのスプリント中に完了すべき、スクラム・チームによって特定された作業項目のリスト。

A list of work items identified by the Scrum team to be completed during the Scrum sprint.

スモーク・テスト (Smoke Testing). 開発作業中のシステムにおける最も重要な機能が、意図した通りに機能していることを確認するために行う一連の軽量のテストの実務慣行。

The practice of using a lightweight set of tests to ensure that the most important functions of the system under development work as intended.

せ

漸進型ライフサイクル (Incremental Life Cycle). 顧客がすぐに使用できる完成した成果物を提供するアプローチ。

An approach that provides finished deliverables that the customer may be able to use immediately.

そ

増分 (Increment). プロジェクト全体の成果物のサブセットとして、機能的であり、テスト済み、そして受入れ済みの成果物。

A functional, tested, and accepted deliverable that is a subset of the overall project outcome.

組織のバイアス (Organizational Bias). 探査対実行、スピード対安定、量対質、および柔軟性対予測可能性などの、コア・バリューによって特徴づけられた一連の尺度に基づく組織の選好。

The preferences of an organization on a set of scales characterized by the following core values: exploration versus execution, speed versus stability, quantity versus quality, and flexibility versus predictability.

組織変革のマネジメント (Organizational Change Management). 現在の状態から意図したビジネスのベネフィットを伴う将来の状態へと、個人、グループ、および組織を移行するための、包括的かつ周期的で構造化されたアプローチ。

A comprehensive, cyclic, and structured approach for transitioning individuals, groups, and organizations from the current state to a future state with intended business benefits.

た

大規模スクラム (LeSS: Large-Scale Scrum). 大規模スクラムとは、スクラムの本来の目的を維持しながら、スケーリングのガイドラインに沿ってスクラムの規模を拡張するプロダクト開発のフレームワーク。

Large-Scale Scrum is a product development framework that extends Scrum with scaling guidelines while preserving the original purposes of Scrum.

ダイナミック・システム開発アプローチ (DSDM: Dynamics Systems Development Model). アジャイル・プロジェクト創出のフレームワーク。

An agile project delivery framework.

タイムボックス (Timebox). 例えば1週間、2週間、3週間または1か月などに固定された期間。「イテレーション」も参照。

A fixed period of time, for example, 1 week, 1 fortnight, 3 weeks or 1 month. See also *Iteration*.

ダブルループ学習 (Double-Loop Learning). 現象にのみ着目するのではなく、根本原因をより詳細に述べ、改善された対応策を考案するために、潜在する価値や前提条件にチャレンジするプロセス。

A process that challenges underlying values and assumptions in order to better elaborate root causes and devise improved countermeasures rather than focusing only on symptoms.

段階的詳細化 (Progressive Elaboration). 得られる情報が増え、より正確な見積りが可能になるにつれ、プロジェクトマネジメント計画書がより詳細化していく反復プロセス。

The iterative process of increasing the level of detail in a project management plan as greater amounts of information and more accurate estimates become available.

て

ディ스럽リンド・アジャイル (DA: Disciplined Agile). 漸進型や反復型のソリューションの引渡しを行うにあたって、単純なプロセスとすることができるプロセス決定のフレームワーク。

A process decision framework that enables simplified process decisions around incremental and iterative solution delivery.

デイリー・スクラム (Daily Scrum). 毎日行う協業的で簡単な会議であり、チームは前日からの進捗状況をレビューし、当日の予定を明らかにし、発生が予想される障害についてハイライトする。デイリー・スタンドアップとしても知られる。

A brief, daily collaboration meeting in which the team reviews progress from the previous day, declares intentions for the current day, and highlights any obstacles encountered or anticipated. Also known as daily standup.

テスト駆動開発 (Test-Driven Development). 仕掛中の作業が継続的に妥当性確認され、欠陥ゼロの考え方で作業をできるようにするために、作業の開始前にテストが定義される技法。

A technique where tests are defined before work is begun, so that work in progress is validated continuously, enabling work with a zero defect mindset.

は

ハイブリッドアプローチ (Hybrid Approach). 複数のアジャイル要素と、非アジャイル的な最終結果をもたらす非アジャイル要素との組み合わせ。

A combination of two or more agile and non-agile elements, having a non-agile end result.

バックログ (Backlog). 「プロダクト・バックログ」を参照。

See *Product Backlog*.

バックログの洗練 (Backlog Refinement). 顧客要求のニーズを満たすために、要求事項をレビューし、更新し、また書いたりすることをチーム協業で行うプロジェクト要求事項や継続的活動の段階的詳細化。

The progressive elaboration of project requirements and/or the ongoing activity in which the team collaboratively reviews, updates, and writes requirements to satisfy the need of the customer request.

バリュー・ストリーム (Value Stream). 特定プロダクトやサービスの引渡しを通して顧客への価値のフローに焦点を当てる組織の構造。

An organizational construct that focuses on the flow of value to customers through the delivery of specific products or services.

バリュー・ストリーム・マッピング (Value Stream Mapping). 顧客向けにプロダクトやサービスを生産するために必要な情報や資材のフローを文書化し、分析し、そして改善するのに使用されるリーン・エンタープライズ技法。

A lean enterprise technique used to document, analyze, and improve the flow of information or materials required to produce a product or service for a customer.

バーンアップ・チャート (Burnup Chart). プロダクトのリリースに向けて完成した作業のグラフ表示

A graphical representation of the work completed towards the release of a product.

バーンダウン・チャート (Burndown Chart). ひとつのタイムボックス内における残作業対残余時間のグラフ表示。

A graphical representation of the work remaining versus the time left in a timebox.

反復型ライフサイクル (Iterative Life Cycle). 未完の作業に対するフィードバックを可能にし、その作業を改善し修正するアプローチ。

An approach that allows feedback for unfinished work to improve and modify that work.

ひ

ビジネス要求文書 (BRD: Business Requirement Documents). 特定のプロジェクトのためのあらゆる要求事項の一覧。

Listing of all requirements for a specific project.

ピボット (Pivot). プロダクトや戦略について新しい仮説をテストするために設計された計画的な軌道修正。

A planned course correction designed to test a new hypothesis about the product or strategy.

ふ

フィーチャー駆動開発 (Feature-Driven Development). クライアントによって価値づけられたフィーチャーの観点で駆動される、軽量なアジャイル・ソフトウェア開発アプローチ。

A lightweight agile software development method driven from a client-valued feature perspective.

プラン-ドゥ-チェック-アクト (PDCA: Plan-Do-Check-Act). 組織内でプロセスやプロダクトのコントロールと継続的改善を促進するために使用される反復的なマネジメント方法。

An iterative management method used in organizations to facilitate the control and continual improvement of processes and products.

振舞い駆動開発 (BDD: Behavior-Driven Development). テストファーストの原則と英語のようなスクリプトを使用するシステム・デザインと妥当性確認の実務慣行。

A system design and validation practice that uses test-first principles and English-like scripts.

フレームワーク (Framework). あるアプローチを支援するアイデアや事実の基本的体系または構造。

A basic system or structure of ideas or facts that support an approach.

フロー・マスター (Flowmaster). 継続的フローやカンバン方式の環境の下で働く、チームのコーチおよびサービス・リクエスト・マネジャー。「スクラム・マスター」と同等。

The coach for a team and service request manager working in a continuous flow or Kanban context. Equivalent to Scrum Master.

ブロッカー (Blocker). 「障害」を参照。

See *Impediment*.

プロジェクトマネジメント・オフィス (PMO: Project Management Office). プロジェクト関連のガバナンス・プロセスを標準化し、資源、方法論、ツールおよび技法についての共有を促進するマネジメント構造。

A management structure that standardizes the project-related governance processes and facilitates the sharing of resources, methodologies, tools, and techniques.

プロダクト・オーナー (Product Owner). プロダクトの価値を最大化し、構築される最終プロダクトに最終的な行動責任と責任説明を有する人物。「サービス・リクエスト・マネジャー」も参照。

A person responsible for maximizing the value of the product and who is ultimately responsible and accountable for the end product that is built. See also *Service Request Manager*.

プロダクト・バックログ (Product Backlog). プロダクトのためにチームが保持する、ユーザーセントリックである要求事項のオーダー・リスト。

An ordered list of user-centric requirements that a team maintains for a product.

へ

ペア・プログラミング (Pair Programming). プログラミングに焦点を当てたペア・ワーク。

Pair work that is focused on programming.

ペア・ワーク (Pair Work). 同じ作業項目に同時に取り組むように、ふたりのチーム・メンバーにペアを組ませる技法。

A technique of pairing two team members to work simultaneously on the same work item.

ペアリング (Pairing). 「ペア・ワーク」参照。

See *Pair Work*.

ペイント・ドリップ (Paint Drip). 「折れたクシ」参照。

See *Broken Comb*.

ペルソナ (Personas). 目標、動機、および代表的な個人的特性で説明される一連の類似したエンド・ユーザーを表す原型ユーザー。

An archetype user representing a set of similar end users described with their goals, motivations, and representative personal characteristics.

ほ

方針管理 (Hoshin Kanri). 戦略や方針の展開方法。

A strategy or policy deployment method.

も

目的適合 (Fit for Purpose). 本来の目的にふさわしいプロダクトを記述する。

Describes a product that is suitable for its intended purpose.

モビング (Mobbing). 複数のチーム・メンバーが、特定の作業項目へのそれぞれの役割分担に同時に注力しコーディネーションする技法。

A technique in which multiple team members focus simultaneously and coordinate their contributions on a particular work item.

ゆ

ユーザー・ストーリー (User Story). 特定ユーザーのための成果物価値の簡単な記述。詳細を明確にするための会話のきっかけとなる。

A brief description of deliverable value for a specific user. It is a promise for a conversation to clarify details.

ユーザー・ストーリー・マッピング (User Story Mapping). 時間の経過と共に作成される一連の高付加価値フィーチャーを理解するのに役立ち、バックログでの欠落を特定し、ユーザーに価値を提供するリリースを効果的に計画する有用なモデルに作業を体系化するための視覚的な実務慣行。

A visual practice for organizing work into a useful model to help understand the sets of high-value features to be created over time, identify omissions in the backlog, and effectively plan releases that deliver value to users.

よ

予測型アプローチ (Predictive Approach). 作業計画とその作業計画のマネジメントをプロジェクトのライフサイクル全体を通して利用する、作業マネジメントのアプローチ。

An approach to work management that utilizes a work plan and management of that work plan throughout the life cycle of a project.

ら

ライフサイクル (Life Cycle). プロダクトが概念化され、作成され、使用されるという一連のプロセス。

The process through which a product is imagined, created, and put into use.

り

リファクタリング (Refactoring). 期待される振舞いを変更することなく、保全性やその他望ましい属性を強化することにより、プロダクトのデザインを改善するプロダクト品質技法のひとつ。

A product quality technique whereby the design of a product is improved by enhancing its maintainability and other desired attributes without altering its expected behavior.

リーン・ソフトウェア開発 (LSD: Lean Software Development). リーン・ソフトウェア開発とは、無駄を省く製造分野における原則と実務慣行をソフトウェア開発分野へ適応することであり、品質、スピード、顧客アライメントを実現するための原則と実務慣行のセットに基づいている。

Lean software development is an adaptation of lean manufacturing principles and practices to the software development domain and is based on a set of principles and practices for achieving quality, speed, and customer alignment.

れ

レトロスペクティブ（振り返り） (Retrospective). プロセスとプロダクトの両方を改善するために、参加者が作業と結果を探求する、定期的に行われるワークショップ。

A regularly occurring workshop in which participants explore their work and results in order to improve both process and product.

ろ

ローリング・ウェーブ計画法 (Rolling Wave Planning). 反復計画技法のひとつ。早期に完了しなければならぬ作業は詳細に、将来の作業はよりハイレベルで計画する。

An iterative planning technique in which the work to be accomplished in the near term is planned in detail, while the work in the future is planned at a higher level.

索引

A

A3 150
ATDD 受入れテスト駆動開発を参照

B

BDD 振る舞い駆動開発を参照
BRD ビジネス要件文書を参照

C

CPI コスト効率指数を参照

D

DA ディスプリンド・アジャイルを参照
DSDM ダイナミック・システム開発アプローチを参照
DevOps 150
DoD Done (完了) の定義を参照
DoR Ready (準備完了) の定義を参照
Done (完了) の定義 (DoD) 150

E

EV アーンド・バリューを参照
EVO 価値の漸進的創出を参照

F

FDA 承認プロセス 26

I

IDEAL 150
I字型 42, 150

L

LSD リーン・ソフトウェア開発を参照
LeSS 大規模スクラムを参照

M

MVP 最小限の実行可能なプロダクトを参照

O

OCM 組織の変更マネジメントを参照

P

PDCA プラン-ドゥ-チェック-アクトを参照
『PMBOK®ガイド』17, 38
『PMBOK®ガイド 第5版 ソフトウェア拡張版』17
PMOと企業内起業家アプローチ 81
PMOプロジェクトマネジメント・オフィスを参照
Project Management Institute (PMI®) 1, 43

R

ROI 投資収益率を参照
RUP ラショナル統一プロセスを参照
Ready (準備完了) の定義 (DoR) 150

S

SAFe® スケールド・アジャイル・フレームワーク®を参照
SBE 実例による仕様を参照
SMEs 当該分野専門家を参照
SPI スケジュール効率指数を参照

T

TDD 振る舞い駆動開発を参照
T字型 42, 150

U

U.S. FDA 承認プロセス 26
UXデザイン 150

W

WIP 進行中の作業を参照

X

XP エクストリーム・プログラミングを参照
XPに基づくエンジニアリングの実務慣行 31

あ

アクション項目 51
アジャイリスト アジャイル実務者を参照
アジャイル
アジャイルの採用 87
アジャイルの実装 33-47
アジャイルという用語の普及 10
さまざまなアプローチとアジャイル 11
定義 150
アジャイル・アプローチ
アジャイル・アプローチと結合した予測型ア
プローチ 27
アジャイル・アプローチの混合 31
アジャイルの構成要素 10
移行 73
予測型構成要素とアジャイル・アプローチ 28
アジャイル・アライアンス 1, 43
アジャイル型開発に後続する予測型展開 26-27
アジャイル型学習 2
アジャイル型ソフトウェア開発の宣言 8
アジャイル型ライフサイクル
アジャイル宣言とアジャイル型ライフサイクル 25
定義 150
特性 24-25
反復ベース 24
フロー・ベース 24
ライフサイクルの連続性とアジャイル型ライ
フサイクル 19

アジャイル環境 アジャイル環境の作成 33-47
アジャイル・コーチ 151
アジャイル実務慣行 50-57
アジャイル実務者
定義 151
プロジェクト・マネジャーの役割とアジャ
イルの実務者 37
アジャイル宣言
アジャイル型ライフサイクルとアジャイル
宣言 25
価値 2, 8, 10, 35, 77
刊行 87
原則 9, 10, 50
根底にある教義 38
実務慣行とアジャイル宣言 10
定義 151
マインドセットとアジャイル宣言 8-12
アジャイル・チーム
成功の属性 39-40
役割 40-41
アジャイル・チームの役割 40-41
アジャイル適合フィルター 25
アジャイル統一プロセス 151
アジャイルの融合 151
アジャイルのマインドセット
アジャイル宣言とアジャイルのマインドセ
ット 8-12, 10
アジャイルのマインドセットで開始 33
顧客との協調 81
サイロ化した組織とアジャイルのマインドセ
ット 47
定義 151
普遍的な適用 87
変化のスピードとアジャイルのマインドセ
ット 3
アジャイルの原則
アジャイル型学習とアジャイルの原則 2
機能横断チームとアジャイルの原則 43
定義 151
変革の準備とアジャイルの原則 73
アジャイルの役割 40-41
アジャイル PMO プロジェクトマネジメント・オ
フィスを参照
アジャイル方式
カンバン方式とアジャイル方式 12
フレームワークとアジャイル方式 11, 80

アプローチ

- アプローチの融合 31
- ガイド内で使用される用語 11

安全環境 75

アンチパターン (落とし穴)

- スタンドアップとアンチパターン 55
- 定義 151

アード・バリュー(EV) 61

- アード・バリューの測定 68-69
- 完了したフィーチャーとアード・バリュー 67-68

い

移行戦略としてのハイブリッド型ライフサイクル 30

依存関係、複数チームの調整、デモ 80

イテレーション

- 動くプロダクトの引渡しとイテレーション 57
- ストーリーポイントとイテレーション 61, 64
- 定義 151

ベロシティとイテレーション 64

「今いるところから出発する」手法 13, 16

インパクト・マッピング

- 定義 151
- プロダクト・オーナーとインパクト・マッピング 52

う

ウォーターフォール・ライフサイクル 17 予測型ライフサイクルも参照

受入基準

- イテレーションと受入基準 63
- 実務慣行と受入基準 56

受入れテスト 82

受入れテスト駆動開発(ATDD)

- 価値創出と受入れテスト駆動開発 56
- 定義 151

え

エクストリーム・プログラミング(XP)

- 価値創出と期待値 56
- コラボレーションと期待値 80
- 定義 152
- 融合アプローチと期待値 31

お

- 応答時間 64,66
- 折れたクシ 152

か

会議 デイリー・スタンドアップを参照

ガイドの基本 基本を参照

カイゼンのイベント 152

概念実証 22

学習

価値と学習 61-62

継続的 73

体系的 82

価値創出と検査 87

価値の漸進的創出 (EV0) 152

価値ビジネス 価値の創出も参照

学習と価値 61-62

加速 30

価値のフローを最適化 38-39

契約の技法と価値 77

創出 16, 23, 56

中間 29

メトリクスと価値 60

カデンツ

動くプロダクトの引渡しとカデンツ 57

定義 152

完全性

作業の合意と完全性 50

主観的な性質 23

感情的知性 36

カンバン

作業の進捗とカンバン 86

順位付けされた変更のバグログ 85

定義 152

融合アプローチとカンバン 31

例 65

カンバン方式 16, 152

カンバン方式の出現 12

融合アプローチとカンバン方式 31

リーン・アプローチとカンバン方式 11

リーンとカンバン方式 12-13

カンバンを「行き来する」 53

完了した作業 価値を参照

き

技術的スキル 36
技術的負債 152 規制環境 36
期待値の設定 45
機能横断チーム
アジャイル・チーム・メンバーの役割 41
アジャイルの原則と機能横断チーム 43
機能的なプロダクト増分と機能横断チーム 39
サーバント・リーダーシップと機能横断チーム 33
スクラム・フレーム・ワークと機能横断チーム 31
定義 152
ビジネス慣行と機能横断チーム 79
プロジェクト・リーダーシップと機能横断チーム 47
プロダクト開発と機能横断チーム 43
変更の多いプロジェクトと機能横断チーム 38
機能化された構造 83
機能仕様 152
機能要求 152
基本 1-5
アジャイル型学習と基本 2
ガイドの作成 1
ガイドの体系 5
ガイドの理由 2
スコープ内の項目とスコープ外の項目 4
破壊的技術と基本 3

く

クラウド・コンピューティング 3
クリスタル・ファミリー方法論 153
グループの規範 50

け

計画
再計画と計画 61
反復ベースのアジャイルと計画 55
フィードバックと計画 29
ライフサイクルと計画 20
計画駆動型アプローチ 153
継続的学習 73
継続的統合
価値創出と継続的統合 56
定義 153
融合アプローチと継続的統合 31

継続的引き渡し 153
契約と早期キャンセルのオプション 78
契約と中止の選択 78
契約の技法 77-79
価値創出と契約の技法 77
構造の多層化 77
上限付きタイム・アンド・マテリアル 78
漸進型の定額契約 77
早期キャンセルのオプション 78
ダイナミック・スコープ・オプション 78
段階的タイム・アンド・マテリアル 78
チームの増強 76
フルサービス・サプライヤー 79

こ

行動規範 50
行動の喚起 87
顧客価値 価値を参照
顧客とサプライヤーの関係破綻 77
顧客との協調マインドセット 81
顧客のフィードバック・ループ 2
顧客満足 2, 25
顧客要求事項 要求を参照
コスト効率指数 (CPI) 69
コーチング 38, 55
コミュニケーション
ファシリテーターとコミュニケーション 35
分裂したチームとコミュニケーション 46
コラボレーション (協業、協働、協調)
憲章の作成プロセスと協働 49
顧客との協調マインドセット 81
迅速な作業と協働 39
善意とコラボレーション 37
促進 35, 38
透明性とコラボレーション 79
部門間 73
リスクと報酬の共有関係 77
集団的コード所有権 154
コロケーション・チーム 39, 43, 44, 45
コンピテンシー
社内 83
障害とコンピテンシー 74
PMO とサービス 82

さ

最小限の実行可能なプロダクト (MVP) 23
サイクル・タイム
外部依存関係とサイクル・タイム 66
フロー・ベースのアジャイル・チームとサイクル・タイム 64
リード・タイムとサイクル・タイム 66
サイロ化した組織
機能横断チームとサイロ化した組織 47
定義 153
作業
ムダな 14
累積 70
作業の切り替え 44, 45
作業の合意 50
作業の割当て 27
作業場所、チーム 46
作業累積 70
サーバント・リーダー
憲章の作成プロセスとサーバント・リーダー 49, 50
サーバント・リーダーを使用するプロジェクト・マネジャー 38
責任 34, 36-37
組織的な障害とサーバント・リーダー 35
特性 34
ファシリテーションとサーバント・リーダー 35, 52
役割 33
サーバント・リーダーシップ
アジャイル・チームとサーバント・リーダーシップ 39
チームの権限委譲とサーバント・リーダーシップ 33-38
定義 153
プロジェクト・マネジャーとサーバント・リーダーシップ 38
サービス
引渡し 35
PMO とサービス 82
サービス・リクエスト・マネジャー 153
暫定スペシャリスト 45, 83

し

視覚的なツール カンバンも参照
思考プロセス アジャイルのマインドセットを参照
自己管理 36
自己管理チーム 39
自己組織化チーム
金融機関の例 44
ケースの例 43
スタンドアップと自己組織化チーム 54
定義 153
プロジェクト・マネジャーと自己組織化チーム 37
システム・レベル・テスト 56
実務慣行 56
実例による仕様 (SBE) 153
自動化 7
自動化されたコード品質分析 154
自動テスト 31, 56
社会契約 プロジェクト憲章を参照
従来型の EVM メトリクス 69
使用期間または使用量に対して支払う方式 3
使用適合 154
障害
定義 154
サーバント・リーダーとインパクト・マッピング 35
障害と組織のアジリティ 74
状況確認会議 54
上限付きタイム・アンド・マテリアル・アプローチ 78
情報ラジエーター 154
シリアル・ライフサイクル 17 予測型ライフサイクルも参照
進化する内部プロセス 73
シングルループ学習 154
信号機を使った状況報告 60
進行中の作業(WIP) 31, 39
カンバンと変更 66
タスク・ボードと進行中の作業 25
累積フロー図と進行中の作業 70
人材の開発 82
進捗状況の追跡 27 カンバンも参照
人的資源 79, 82

す

- スウォーミング 39, 154
- スクラム
 - コラボレーションとスクラム 80
 - 定義 154
 - フレームワーク 31
- スクラムのスクラム 154
- スクラム・チーム 154
- スクラムバン 154
- スクラム・ボード 155
- スクラム・マスター
 - スクラム・フレーム・ワークとスクラム・マスター 31
 - 定義 155
- スケーリング 80
- スケーリングと複数チームの協働
- スケールド・アジャイル・フレームワーク (SAFe®) 155
- スケジュール効率指数 (SPI) 69
- スコープ外の項目 4
- スコープ・クリープ 28
- スコープ内の項目 4
- スコープ内の項目とスコープ外の項目 4
- スタンドアップデイリー・スタンドアップを参照
- ステークホルダー
 - 教育 37
 - プロジェクトリーダーとステークホルダー 75
 - マネジメント 82
- ステーシー複雑性モデル 14, 15
- ストーリーユーザー・ストーリーを参照
 - 一度にひとつ終了 68
 - 信頼できるベロシティとストーリー 61
 - バックログの洗練とストーリー 52, 53
- ストーリー・カード 31
- ストーリー・ポイント
 - イテレーションとストーリー・ポイント 61, 64
 - 完了済み 63
 - 測定する 66
 - 測定とストーリー・ポイント 66
 - 定義 155
 - バーンアップ・チャートとストーリー・ポイント 63
 - バーンダウン・チャートとストーリー・ポイント 62
 - ベロシティとストーリー・ポイント 64

スパイク

- 価値創出とスパイク 56
- 定義 155
- バックログの洗練とスパイク 52

スプリント

- 155

スプリント計画

- スクラム・フレーム・ワークとスプリント計画 31
- 定義 155

スプリント・バックログ

- 155

スモーク・テスト

- 価値創出とスモーク・テスト 56
- 定義 155

スループット

- 42
- スタンドアップとスループット 54
- プロダクト・オーナーとスループット 66
- マルチタスクとスループット 44

せ

成果物 サービスも参照

- 価値駆動 77
- 極小成果物 77
- 中間 15
- プロジェクト規模の縮小 83
- 要求と成果物

生産性

- 促進 39-40
- タスクの切替えと生産性 44-45

生産性とタスクの切り替え

- 44-45

制約条件

- 20, 31, 42

宣言 アジャイル宣言を参照

先行見積り

- 27

漸進型施策

- 20

漸進型予算編成

- 36

漸進型ライフサイクル

- 定義 155
- 特性 22-23
- さまざまな大きさの増分と漸進型ライフサイクル 22
- ライフサイクルの連続性と漸進型ライフサイクル 19

専任チーム・メンバー

- 44-45

戦略

- 文化と戦略 75
- 目的に対する情熱と戦略 75

そ

- 相対的見積り 67
- 増分と動くプロダクトの引渡し 57
- 測定 (尺度)
 - アジャイル・プロジェクトと測定 60-70
 - アード・バリューと測定 68-69
 - EVM. 69
 - 所産と測定 61-70
 - ストーリーポイントと測定 66
 - 定性的 60
 - 能力 66
 - フロー・ベースのアジャイル・チームと測定 64
 - ベースラインと測定 61
 - 変動性尺度 61
 - 予測可能性 66
- ソーシャル・メディア 2
- ソフトウェア開発
 - アジャイル実務慣行とソフトウェア開発 2
 - アジャイル宣言とソフトウェア開発 8
 - 学習とソフトウェア開発 61
 - ソフトウェア開発における思考リーダー 8
- 組織
 - サイロ化 47, 153
 - 組織の発展 84-86
 - 調達重視 83
- 組織構造 83
- 組織としての学習 82
- 組織のアジリティの障害 74
- 『組織のチェンジマネジメント:実務ガイド』 3, 71
- 組織のバイアス 156
- 組織の文化 75-77
 - 安全環境と組織の文化 75
 - 組織構造対 77
 - 組織の文化評価 74-75
 - 評価の例 76
 - PMO と組織の文化 81
- 組織の変更マネジメント(OCM) 3, 71-74, 156
 - アジャイル・アプローチと組織の変更マネジメント 71-72
 - 駆動要因 73
 - 変革の準備と組織の変更マネジメント 73-74
- 組織的なサイロ サイロ化した組織を参照
- 組織的な障害 35

た

- 大規模スクラム (LeSS) 156
- ダイナミック・システム開発アプローチ(DSDM) 156
- ダイナミック・スコープ契約アプローチ 78
- タイム・アンド・マテリアル・アプローチ
 - 上限付き 78
 - 段階的 78
- タイムボックス スパイクも参照
 - 使用 12
 - スタンドアップとタイムボックス 53
 - 定義 156
- 対面式ペアリング 46
- タスク・ボード カンバン、プロジェクト・タスク・ボード
 - 「行き来する」 53
- ダブルループ学習 156
- 段階的詳細化 156 バックログの洗練も参照
- 単体テスト 56

ち

- 遅延 64
- チーム アジャイル・チーム、機能横断チーム、自己組織化チームも参照
 - 憲章の作成プロセスとチーム 49-50
 - コア・メンバー 45
 - 構成 38-47
 - コロケーション 39, 43, 44, 45
 - 作業累積 70
 - 自己管理 39
 - 中核執筆、ガイド、チーム 1
 - 調整、複数チーム 80
 - 引渡し 35
 - ビジネス慣行とチーム 79
 - ファシリテーターの役割 41
 - 分散した 43, 46
 - 分裂した 43, 44, 45
- チーム憲章 49-50
- チームの価値観 50
- チームの構造 43
- チームの作業場所 46
- チームの増強の契約アプローチ 76
- チーム・ファシリテーターの役割 41
- チーム・リーダー 82
- 中間成果物 15

調整

- サーバント・リーダーシップと調整 35
- 複数のチーム 80

調達

- 契約と調達 77-79
- ビジネス慣行と調達 79
- 調達重視の組織 83
- 地理的に分散したチーム 46
- 地理的に分散したプロジェクト組織 83

て

- 定額の極小成果物 77
- 定義可能な作業のプロジェクト 7
- 定性的指標 60
- ディスプリンド・アジャイル (DA) 156
- デイリー・スクラム 156
- デイリー・スタンドアップ 27, 44, 53-54
 - 落とし穴とデイリー・スタンドアップ 54
 - 反復ベースのアジャイル 53
 - フロー・ベースのアジャイル 54
- 適応
 - 価値創出と適応 87
 - プロセスと適応 15, 28
- 適合フィルター 25
- 適合性を改善するテラリング・オプション 32
- テスト
 - 受入れ 82
 - 自動 31, 56
 - 全レベル 56
 - 不確実性とテスト 16
- テスト駆動開発(TDD)
 - 価値創出とテスト駆動開発 56
 - 定義 157
 - 融合アプローチとテスト駆動開発 31
- 手直し
 - 潜在的な手直しを減少 23
 - 手直しのリスク 13, 14
- デモ
 - 引渡しとデモ 57
 - レビューとデモ 55
- テラリング
 - 影響のあるプロジェクト要因 32
 - 時期尚早または行き当たりばったり 12
 - ハイブリッド移行とテラリング 30
 - PMOとテラリング 81
- テレビ会議 46
 - バーチャルな作業空間 46

と

- 当該分野専門家(SMEs) 43, 82
- 統合 継続的統合を参照
- 投資収益率(ROI) 30, 61
- 透明性
 - 価値創出と透明性 87
 - コラボレーションと透明性 79
 - 成功と透明性 85
- トラブルシューティング 57-59, 82
- トレードオフ 76
- トレーニング 82

に

- 人間関係のスキル 36
- 人間関係のスキル対技術的スキル 36

の

- 能力の尺度
 - 現時点での測定と能力の尺度 66
 - ストーリーポイントと能力の尺度 66
 - 反復ベースのアジャイルと能力の尺度 55

は

- ハイブリッドアプローチ 27, 157
- ハイブリッド型ライフサイクル
 - 移行戦略として 30
 - 特性 26-27
 - 目的適合として 29
 - 例 26
- 破壊的技術 2, 3
- バックログ プロダクト・バックログを参照
- バックログの洗練 52-53
 - 会議の実施 53
 - 洗練作業期間とバックログ洗練 52
 - 定義 157
- バッチ・サイズ 42
- バリュー・ストリーム 157
- バリュー・ストリーム・マッピング 157
- バーンアップ・チャート
 - アード・バリューとバーンアップ・チャート 68-69
 - スコープ変更とバーンアップ・チャート 64
 - ストーリーポイントとバーンアップ・チャート 63
 - 定義 157
 - フィーチャー図とバーンアップ・チャート 67
 - プロダクト・バックログ 68

汎化スペシャリスト 42
バーンダウン・チャート
ストーリーポイントとバーンダウン・チャート 62
定義 157
フィーチャー図とバーンダウン・チャート 67
反復型ライフサイクル
単一プロダクトの引渡し 21
定義 157
特性 21-22
ライフサイクルの連続性と反復ベースのアジャイル 19
反復ベースのアジャイル
計画 55
スタンドアップと反復ベースのアジャイル 53
バーンダウン・チャートと反復ベースのアジャイル 62
フロー・ベースのアジャイルとの比較 24, 25

ひ

非アジャイル・アプローチ 17
引渡し ビジネス価値の創出も参照
イテレーション、増分、引渡し 57
加速 73
顧客ベース 29
主観的な性質 23
進行中の作業と引渡し 70
頻繁 55
引渡しチーム 35
引渡しを加速することに関連する変更 73
ビジネス価値の創出 16, 23, 29
ビジネス慣行 79
ビジネス・サービス サービスを参照
ビジネス満足度 60
ビジネス要求文書 (BRD) 157
ピボット 157

ふ

ファシリテーター 35, 51
フィーチャー駆動開発 158
フィーチャー図 67
フィーチャーのバーンアップ・チャートまたはバーンダウン・チャート 67
フィーチャーの引渡し 引渡しを参照
フィッシュボウル・ウィンドウ 46

フィードバック
アジャイル・チームとフィードバック 39, 42
イテレーションとフィードバック 57
組み込み 43
計画とフィードバック 29
デモとフィードバック 55
プロトタイプとフィードバック 22, 23
ムダ、手直し、フィードバック 15
フィードバック・ループ 2, 15
「フェーズ・ゲート」 77
「フォロー・ザ・サン」の開発の実務慣行 44
不確実性 変更も参照
不確実性と複雑性のモデル 14
不確実性の高いプロジェクト 7
複雑性ステーション 複雑性モデルも参照
トラブルシューティングと複雑性 57
ハイブリッド型ライフサイクルと複雑性 26
反復型ライフサイクルと複雑性 21
不確実性と複雑性 7, 13
変更の多いプロジェクトと複雑性 38
複雑性と不確実性 7, 13
技術的な不確実性 14
探索 16
中程度から低程度 30
要求と不確実性 13, 14, 16, 22, 24
ライフサイクル、リスク、不確実性 13-16
複数のプロジェクト・マネジメント 82
プラン・ドゥ・チェック・アクト (PDCA) 158
フルサービス・サプライヤー 79
振る舞い駆動開発 (BDD)
定義 158
価値創出と振る舞い駆動開発 56
フレームワーク
アジャイル方式 80
定義 158
フロー効率 42
フロー・ベースのアジャイル・ライフ・サイクル
スタンドアップとフロー・ベースのアジャイル・ライフ・サイクル 54
反復ベースのアジャイルとの比較 24, 25
フロー・マスター 158
プロジェクト
固有の特性とプロジェクト 18
大規模 15
プロジェクト完了とスポンサー 61
プロジェクト憲章 49-50

- プロジェクト作業 7
 - プロジェクト知識とベンダー 83
 - プロジェクトの失敗 77
 - プロジェクトのタスク・ボード
 - 「行き来する」 53
 - 進行中の作業とプロジェクトのタスク・ボード 25
 - 累積フロー図とプロジェクトのタスク・ボード 70
 - プロジェクトのビジョン 49
 - プロジェクトマネジメント・オフィス(PMO) 81-82
 - 学際的 82
 - 価値駆動 81
 - 招集指向 81
 - 定義 158
 - デモとプロジェクトマネジメント・オフィス 57
 - プロジェクトマネジメント知識体系『PMBOK®ガイド』を参照
 - プロジェクトマネジメントのゴール 29
 - プロジェクト・マネジャー
 - アジャイル環境とプロジェクト・マネジャー
 - サーバント・リーダーシップとプロジェクト・マネジャー 38
 - 定義 38
 - 役割 37
 - プロジェクト要因とテーラリング・オプション 32
 - プロジェクト・ライフサイクル ライフサイクルを参照
 - プロジェクトリスクとハイブリッド型ライフサイクル 29
 - プロジェクトリーダーとステークホルダー 75
 - プロダクト・オーナー
 - アジャイル・チーム・メンバーの役割 41
 - 機能横断チームとプロダクト・オーナー 38
 - スクラム・フレーム・ワークとプロダクト・オーナー 31
 - スループットとプロダクト・オーナー 66
 - 定義 158
 - プロダクト・ロードマップとプロダクト・オーナー 52
 - プロダクトの知識 83
 - プロダクトの引渡し 引渡しを参照
 - プロダクト・バックログ バックログの洗練も参照
 - 最初の順位付けされた変更のバックログ 85
 - 準備 52
 - スクラム・フレーム・ワーク 31
 - 定義 158
 - プロダクト・バックログ・バーンアップ・チャート 68
 - プロダクト・ロードマップ 52
 - ブロッカー 障害を参照
 - プロトタイプ 15, 22
 - 文化 組織の文化も参照
 - 分散したチーム 43
 - 分裂したチーム 43, 44, 45
- ## へ
- ペア・プログラミング 102, 158
 - ペアリング ペア・ワークを参照
 - ペア・ワーク 39
 - ペイント・ドリップ 折れたクシを参照
 - ペインポイントとトラブルシューティング 57-59
 - ベースライン 61
 - ベロシティ
 - 相対的見積りとベロシティ 67
 - 定義 64
 - ペルソナ 159
 - 変更(変化、変革) 不確実性も参照
 - アジャイル・アプローチと変更 73
 - アジャイルのマインドセットと変化のスピード 3
 - 安全性と変更 75
 - カンバンと変更 85
 - 引渡しの加速と変更 73
 - 変革への障害 74
 - 変更の準備 73-74
 - 要求と変更 24
 - 変更管理委員会 35
 - 変更の多いプロジェクト 38
 - 変更マネジメント 組織の変更マネジメントを参照
 - 変更要求プロセス 7, 8-12
 - ベンダーとプロジェクト知識 83
 - 変動性の尺度 61
- ## ほ
- 方針管理 159
 - 保険引き受け業務システム 29
 - ボストン・ビッグ・ディグ 15
 - ボトルネック 35, 42, 64

ま

マインドセット アジャイルのマインドセットを参照
マルチタスク
生産性とマルチタスク 44-45
バーンダウンとマルチタスク 63

み

見積り
先行 27
相対的 67
ミニ・ウォーターフォール 39

む

ムダな作業 14
ムダの減少 15

め

メトリクス 測定を参照
メンタリング 37, 82

も

目的適合
定義 159
ハイブリッド型ライフサイクルと目的適合 29
モビング 39, 159
問題
スタンドアップと問題 54
トラブルシューティング 57-59
問題解決の促進 39-40
問題とパーキング・ロット 54

や

役割
アジャイル・チームと役割 40-41
暫定スペシャリストと役割 45
プロジェクト・マネジャー 37

ゆ

融合アプローチ 31
ユーザー・ストーリー
極小成果物として 77
定義 159
デモとユーザー・ストーリー 55
ユーザー・ストーリー・マッピング 159

よ

要求事項
すべてに対処 39
反復的探索 15
フィーチャーのバーンアップ・チャートまたはバーンダウン・チャートと要求事項 67
不確実性と要求事項 13, 14, 16, 22, 24
不足 60
文化と要求事項 75
プロトタイプと要求事項 22
予測型ライフサイクルと要求事項 20
予測型アプローチ
アジャイル構成要素のある 28
測定と予測型アプローチ 60
予測型アプローチと結合したアジャイル・アプローチ 27
予測型構成要素のあるアジャイル・アプローチ 28
予測型ライフサイクル
特性 20-21
ライフサイクルの連続性と予測型ライフサイクル 19

ら

ライフサイクル アジャイル型ライフサイクル、ハイブリッド型ライフサイクル、漸進型ライフサイクル、反復型ライフサイクル、予測型ライフサイクルも参照
計画とライフサイクル 20
選択 17
タイプ 17
定義 159
特性 18
ライフサイクルの連続性 19
ラショナル統一プロセス (RUP) 149

り

- リーダーシップ サーバント・リーダーシップ
を参照
- リード・タイム
 - 外部依存関係とリード・タイム 66
 - サイクル・タイムとリード・タイム 66
 - フロー・ベースのアジャイル・チームとリ
ード・タイム 64
- リスク
 - 兼任とリスク 45
 - 顧客とサプライヤーの関係とリスク 77
 - 漸進型の定額契約とリスク 77
 - ハイブリッド型ライフサイクルとリスク 29
 - 不確実性、ライフサイクル、リスク 13-16
 - 変更の多いプロジェクトとリスク 7
- リスクと兼任 45
- リスクの定量的分析 37
- リファクタリング
 - 定義 159
 - 融合アプローチとリファクタリング 31
- リモート・ペアリング 46
- リーン
 - アジャイル・アプローチとリーン 11
 - カンバン方式とリーン 12-13
- リーン思考 11, 12
- リーン・ソフトウェア開発 (LSD) 160

れ

- レトロスペクティブ（振り返り） 27, 50-51
 - 重要な時点 51
 - 定義 160
 - プロダクト知識とレトロスペクティブ
（振り返り） 83
- レビューとデモ 55

る

- 累積フロー図 70, 82

ろ

- ロードマップ、プロダクト 52
- ローリング・ウェーブ計画法の定義 160