
Microsoft Solutions Framework White Paper

Published: 2004

Microsoft Solutions Frameworkの詳細情報は
<http://www.microsoft.com/japan/msdn/vstudio/productinfo/enterprise/msf/> をご覧ください

MSF チーム モデル v. 3.1

目次

要約	4
はじめに	4
チーム モデルの基礎	6
基礎となる MSF の基本原則	6
主要コンセプト	11
実証済みのプラクティス	15
チーム モデルの概要	18
チーム モデルの役割群	21
プロダクトマネジメント役割群	21
プログラムマネジメント役割群	24
開発役割群	28
テスト役割群	31
ユーザエクスペリエンス役割群	33
リリースマネジメント役割群	37
チーム モデルのスケールリング	39
エスカレーションと説明責任	43
まとめ	45
付記	45

クレジット

Paul Haynes, Program Manager, Microsoft Solutions Framework

Allison Robin, Director, Microsoft Solutions Framework

Enzo Paschino, Program Manager, Microsoft Solutions Framework

Mark Short, Program Manager, Microsoft Solutions Framework

Pervez Kazmi, Program Manager, Microsoft Solutions Framework

Rob Oikawa, Program Manager, Microsoft Solutions Framework

Scott Getchell, Program Manager, Microsoft Solutions Framework

Laura Hargrave, Technical Editor, Microsoft Solutions Framework

Mike Lubrecht, Technical Writer, Microsoft Solutions Framework

Nany Huber, Technical Editor, Microsoft Solutions Framework

Suzana Vukcevic, Program Manager, Microsoft Belgium

Paulo Henrique Leocadio, Microsoft Consulting Services, Brazil

Paulo Rocha, Microsoft Consulting Services, New Zealand

Dolph Santello, Microsoft Consulting Services, US

Ralph Schimpl, Director, Microsoft Austria

本ホワイトペーパーの日本語翻訳にあたっては、Microsoft Consulting Services の小泉 浩氏、片山 龍児氏に多大なご協力をいただきました。この場を借りて御礼申し上げます。

本書に記載されている情報は、発行日の時点での検討問題に対する Microsoft の見解を表したものです。Microsoft は市場の動向に対応すべく努めておりますが、本書に含まれている情報は Microsoft の義務を表すものではありません。また、Microsoft は発行日以降に生じたいかなる情報の正確性についても一切保証いたしません。

本書は情報目的のみに提供されています。Microsoft は、**本書の内容について明示、黙示、法定を問わず一切の保証をいたしません。**

お客様ご自身の責任において、適用されるすべての著作権法に従ってご利用ください。著作権法に基づく権利を制限しない範囲において、この文書のいずれの部分も、いかなる形式によっても、またいかなる手段（電子的または機械的な手段、 fotocopy、録音、その他）によっても、目的の如何を問わず、マイクロソフト社の書面による明白な許可なしに複製、検索システムへ保管、転送することは禁じられています。

Microsoft は、このドキュメントの内容に関して、特許、特許申請、商標、著作権、またはその他の知的財産権を所有している場合があります。Microsoft の使用許諾契約書に特に明記されている場合を除き、このドキュメントは、これらの特許、商標、著作権、またはその他の知的財産の権利を許諾するものではありません。

© 2002 Microsoft Corporation. All rights reserved.

Microsoft および Visual Basic は、Microsoft Corporation の米国およびその他の国における登録商標です。

この文書に記載されている実際の会社名および商品名は、各社の商標である場合があります。

Part Number: 602-i400a

要約

Microsoft Solutions Framework (MSF) チーム モデルは、プロジェクトを成功に導くための、人とその活動の組織化に対する Microsoft のアプローチを示したものです。このモデルは、チームのメンバーがプロジェクト ライフサイクルにおけるそれぞれの目標を達成できるようにするために、役割群、職能領域、責務、およびガイドを定義します。

はじめに

Microsoft Solutions Framework と Microsoft Operations Framework (MOF) は、IT ライフサイクル全体を通じて IT プロジェクトや IT 運用を最大限に成功させるために、ソリューションの効果的な計画、構築、展開、および運用のためのガイドと実証済みのプラクティスを提供します。この情報は、ソフトウェア開発やサービス運用の大規模プロジェクトから Microsoft が得た経験、Microsoft のコンサルタントの経験、および世界中の IT 産業における共通のベスト プラクティスから集められたもので、ホワイト ペーパー、ガイド、ツール、テンプレート、ケース スタディ、およびコースウェアとして提供されます。この指針とプラクティスは、互いに補い合う、密接に統合された 2 つの知識体系にまとめられています。

Microsoft Solutions Framework

納期通りに予算内でビジネス ソリューションを作成するという複雑な課題も、実証済みのアプローチを採用することによって単純化できます。MSF は、成功する IT ソリューションの計画、構築、および展開 のための実証済みのプラクティスを提供します。MSF が提供するものは規定的な方法論ではなく、柔軟性と拡張性に富んだフレームワークであるため、あらゆる規模の組織やプロジェクト チームのニーズに対応できます。MSF のガイドは、ほとんどのプロジェクトが遭遇する人、プロセス、およびテクノロジーの各要素を管理するための原則、モデル、および規範によって構成されています。

MSF の詳細については、

<http://www.microsoft.com/japan/msdn/vstudio/productinfo/enterprise/msf/> を参照してください。

Microsoft Operations Framework

MOF は、Microsoft の製品や技術を使用して構築された IT ソリューションでミッションクリティカルなシステムの信頼性、可用性、可サポート性、および可管理性を達成するためのガイドを提供します。MOF の原則、モデル、および規範は、複雑な異種分散 IT 環境の運用に関連する人、プロセス、およびテクノロジーの問題に対応しています。

MOF の詳細については、<http://www.microsoft.com/japan/technet/itsolutions/techguide/mof/> 参照してください。

MOF は、業界のベスト プラクティスを集めた IT Infrastructure Library (ITIL) をベースとしています。ITIL の詳細については、<http://www.itil.co.uk/index.html> を参照してください。

チーム モデルの基礎

MSF チーム モデルは、従来のトップダウンの階層的なプロジェクト チームの構造によってもたらされる欠点のいくつかを補うために、数年の歳月をかけて開発されました。

MSF チーム モデルに基づいて構成されたチームは、小規模の分野横断的なチームになります。チームのメンバは目の前のプロジェクトに全力を傾けるために、実行責任を共有し、互いの能力のバランスを調整します。また、プロジェクトの共通のビジョン、プロジェクトの展開 へのフォーカス、品質とコミュニケーションについての高レベルの標準、および学習する意欲を共有します。このホワイト ペーパーでは、チーム内のさまざまな役割群について、それぞれの目標や職能領域と共に説明します。また、Microsoft のアプローチを使ってチームを編成する際に、小さいプロジェクトあるいは大規模で複雑なプロジェクトに合わせて規模調整するための方法についても説明します。

まず始めに、Microsoft のチーム モデルに適用されている MSF の基本原則、主要コンセプト、および実証済みプラクティスの概要を紹介します。このセクションで取り上げる主要な原理は、このドキュメント全体にわたって MSF チーム モデルの詳細について説明する中で参照されます。

基礎となる MSF の基本原則

MSF には、フレームワークのアプローチの土台となるいくつかの基本原則があります。このセクションでは、成功するチーム作業に関連する原則をいくつか紹介します。

説明責任(役割分担)を明確にし、実行責任を共有する

MSF では、作業の実施に対する共有された実行責任と、作業の完了を確実にすることに対する明確な説明責任が組み合わされています。

MSF チーム モデルでは、各役割にはそれぞれ対等な目標があり、プロジェクトに対する独自の観点を持っているということ、およびさまざまな品質目標のすべてを 1 人の個人が代表するのは不可能であるということを前提としています。このジレンマを解決するためには、ステークホルダーに対する明確な一連の説明責任と、全体の成功のための共有責任の両方を組み合わせたチームオブピアーズが必要です。

チーム内では、各役割が、それぞれの品質目標の達成についてチーム自体（およびそれぞれが所属する組織）に対して説明責任を負います。この意味で、各役割は最終的なソリューションの品質の担当分に対する説明責任を負うことになります。実行責任はチームオブピアーズにまたがって共有されます（チームの役割に基づいて割り当てられます）。実行責任は相互依存の関係にあります。これには 2 つの理由があります。各役割の作業を孤立させることはできないという必然的な理由と、各役割がプロジェクトの全体像を把握していた方がチームがより効果的になるという、より積極的な理由です。この相互の依存関係により、チームのすべてのメンバが、それぞれの直接の説明責任の領域を離れたところで意見を述べたり、貢献したりできるようになるため、チームの知識、能力、および経験を最大限に活用できます。すべてのチームメンバはプロジェクトの成功を自分のものと認知します。つまり、プロジェクトが成功した暁にはチームのすべてのメンバが荣誉と報酬を共有し、プロジェクトがあまり成功しなかった場合には各自がそれを教訓とすることに貢献し、その教訓から学ぶことによって、それぞれの専門技術を高めていくことが望まれます。

チームメンバーをエンパワーする

効果的なチームにおいては、各メンバは自分のコミットメントを果たすためにエンパワーされています。また、他のメンバのコミットメントに依存する場合でもそれが達成されるという信頼もあります。同様に、顧客にはチームがコミットメントを果たし、それに基づいて計画をたてると想定する権利があります。万が一なんらかの遅延や変更が生じた場合には、できるだけ早く顧客に報告する義務があります。

MSF チームでは、それぞれのコミットメントを果たすために必要な程度に各メンバがエンパワーされます。その代わりに、すべてのチーム メンバには、次のようなことに誠実さとやる気をもって取り組むことが求められます。

- 他者への献身に備えます。
- 自分のコミットメントを明らかにします。
- そのコミットメントを果たすために相応の努力をします。
- コミットメントを果たせない危険性がわかったらすぐにそのことを正直に伝えます。

活動に複数のメンバが必要になるとすぐ、各メンバの作業はチームの他のメンバの作業との依存関係に影響されるようになります。しかし、だからといって、それらの依存関係の全てを逐一監視するような時間はありません。効果的なチームは、同僚がエンパワーされておりチームの目的にコミットしているという確信を築きます。

これを陸上競技のリレー チームにたとえてみましょう。リレーの第 2 走者は、いったん走り始めたら、第 1 走者との距離を確かめるために速度をゆるめて後ろを振り返ったりはしません。できる限り早く加速することに専念し、バトンが渡されることを確信してただ手を伸ばすだけです。この確信は、練習、経験、および信頼に基づきます。

複雑なプロジェクトでも、チームのメンバはこれと同程度の信頼を築く必要があります。この信頼は、どんなに小さなものであっても、コミットメントが果たされるたびに築かれていきます。信頼関係を育むための簡単なガイドラインを以下に示します。

- 与えられたコミットメントを果たすためにチームメンバをエンパワーします。そのためには、作業を遂行するために必要なリソースと、作業に影響を与える決定に対する責任をチームのメンバに与える必要があります。また、メンバが自分の権限の範囲を把握し、その範囲を超えた問題を処理するためのエスカレーションパスを理解している必要もあります。
- 他者への献身に備えます。これには、気持ちの問題（行動を起こす意欲を持って会議に臨む）、能力やスキルなどの準備、およびコミットメントの含意や現在の作業負荷とリソースに対する影響の理解が含まれます。したがって、そこに含まれる意味がはっきりするまで大きなコミットメントをすべきではありません。その代わりに、まずそこに含まれる意味について調べてから確固としたコミットメントを行うなど、自分が把握できる範囲での小さなコミットメントを申し出るようにします。こうした小さなコミットメント達成の積み重ねがチームの信頼を築きます。
- 請合うコミットメントを明確に定義します。これにより、チームのメンバ間の、信頼関係を損ないかねない誤解を防ぐことができます。
- そのコミットメントを果たすために相応の努力をします。チームが複数の異なる組織からのメンバにより構成されている場合は、どこまでが相応かということに対する期待が異なるかも知れません。たとえば、週末に働くのを相応と考えるメンバもいれば、それを特別なことと考えるメンバや、そもそも週末には会社のビルに入れないメンバもいるかもしれません。
- コミットメントを果たせない危険性がわかったらすぐにそのことを正直に伝えます。ときに事情が変わることは避けられません。業務の優先順位が変わることもあれば、予期せぬ事態が発生したり、単純に予想より多くの作業時間がかかったりすることもあります。しかし、そのことを早めに伝達すれば、それに依存する他のメンバが対応計画を立てることができます。また、問題の解決方法についての提案を得られる可能性もあります。

ほとんどの組織ではこうした振る舞いは文化に根付いており、当たり前のことと見なされてめったに議論されません。しかし、これらの意義が十分に理解も尊重もされていない組織と共に働かなければならないこともあります。そうした組織では、情報のオープンな流れを阻害する非難の文化がよく見られます。そのような場合は、この点に関して何を期待するのかをチームのリーダーが明確に述べ、新しいメンバがチームのやり方に順応できるようにする必要があります。

ビジネス上の価値に集中する

MSF チーム モデルでは、チームの意思決定は顧客のビジネスに対する正しい理解とプロジェクトへの顧客の積極的な参加に基づいて行われるべきだとしています。プロダクトマネジメントの役割はチームに対する顧客の代弁者として行動し、しばしば、顧客組織のメンバが担当します。プロダクトマネジメントはビジネス ケースを担当し、これにより、早期の戦略的作業からの連続性が提供されます。プロジェクトの重要な決定がビジネスに対する正しい理解に基づいて行われるようにすることは、プロダクトマネジメントの責務です。

リリースマネジメントの役割は、展開 と運用がスムーズに行われるようにするという明白な責務を持ちます。その作業を通じて、この役割は、ソリューションの開発、ソリューションの展開、および継続的な運用を繋ぐ仲介役として行動します。これにより、プロジェクトを遂行するグループは、自分たちの決定が稼働運用における価値の提供にどのような影響を及ぼすのかを常に意識するようになります。

ビジョンを共有する

MSF では、チームの目標が IT ソリューションの提供であろうと運用環境における IT サービスの提供であろうと、チームのアプローチを集中させることができるように、チームでビジョンを共有することを強く推奨しています。

プロジェクトまたはプロセスの目標や目的が何なのかを明確にすることが重要です。というのも、チームのメンバや顧客はソリューションが組織に何をもたらすのかについてそれぞれに仮定するからです。ビジョンを共有することでそれぞれが仮定している内容が明らかになり、すべての関係者が同じ目標の達成に向けて取り組めるようになります。ビジョンの共有は、MSF チーム モデルの土台の 1 つです。

すべての関係者が共有されたビジョンを理解し、それに向かって取り組むようになれば、そのビジョンによって表されるチームのより大きな目的に沿った決定を各人が下せるようになります。

共有されたビジョンがないと、チームのメンバ間で目標についての考えが対立する恐れがあるため、チームが一丸となってプロジェクトに取り組むことがはるかに困難になります。また、プロジェクトが終了しても、プロジェクトが成功したかどうかはどのビジョンを測定基準にするかによって変わるため、チームのメンバがそれぞれ異なるビジョンを持っているとその判断が難しくなります。

俊敏であり続け、変化を予期する

MSF は、物事は常に変化しており、そうした変化から IT ソリューションを提供するプロジェクトを隔離することはできないという前提に立っています。MSF チーム モデルは、中心的な役割の全てがプロジェクト全体を通じていつでも参加しており、こうした変化によって生じる決定に貢献できるようにします。新しい変化が起こると、MSF チーム モデルは、そうした問題に対処する俊敏性を促進します。チームのすべての役割が意思決定に貢献することにより、問題を全ての不可欠な観点から調べたりレビューしたりできることを確実にします。

オープンなコミュニケーションを促進する

これまで多くの組織やプロジェクトでは、必要になったときにのみ情報を知らせるという原則が採られてきましたが、これは頻繁に誤解を招いたり、チームのソリューション提供能力を損なったりする結果を招きます。

これに対して MSF では、チームのメンバ間およびチームと主要なステークホルダーとの間の、オープンかつ率直なコミュニケーションを提案しています。こうして情報が自由に流れるようにすることによって、誤解や無駄な労力が生じる機会が減るだけでなく、すべてのメンバがプロジェクトの不確実性の削減に貢献できるようになります。

チームオブピアーズのアプローチでは、重要な意思決定にすべての役割が関与します。ソリューション提供のプロセスを開始するにはチームによるビジョンの共有が欠かせないとされる理由の 1 つがここにあります。ビジョンの共有は、MSF のリスク管理のアプローチの基礎にもなっています。MSF のリスク管理は、リスクの識別と分析にチームのすべてのメンバが参加することと、これを奨励するために非難のない文化を推進することを強く推奨しています。うまく機能しているのはどこで改善の余地があるのはどこなのかを率直に話し合うことは、MSF が目指す学習環境の基礎となります。

個人情報や取引情報の機密性など、チームのオープンなコミュニケーションの制約となる重要な要因もいくつかあります。しかし、チームのメンバが情報提供を控えようとする場合は、必ず、そこに何よりも優先される理由があるかどうかを自問する必要があります。オープンなコミュニケーションによって信頼関係が築かれていれば、まれに情報提供を控える必要が生じたとしても、そこに重要な理由があること、それがプロジェクトのためになることを同僚に説明できるでしょう。

主要コンセプト

MSF チーム モデルの成功例には、いくつかの特徴が共通して見られます。ここでは、それらの特徴を洗い出して、主要コンセプトとして紹介します。

チームオブピアーズ

"チームオブピアーズ" の概念では、各役割に等しく価値を置きます。これにより、役割間のコミュニケーションの制約が取り除かれ、チームの説明能力が増し、6 つの品質目標のいずれも等しく重要であり、すべてが達成されなければならないという概念が強化されます。チームオブピアーズが成功するためには、すべての役割が製品の品質の責任を担い、顧客の代弁者として機能し、解決しようとしているビジネスの問題を理解していなければなりません。

各役割はチームにおいてそれぞれ同じ価値を持ちますが、チームオブピアーズは役割の "間" に存在するものであり、コンセンサス主義の意思決定と混同しないようにする必要があります。各役割内では、作業の分配やリソースの管理のために、なんらかの形の組織階層が必要です。各役割のリーダーはチームの管理、指導、調整の責務を負い、その他のメンバはそれぞれの目標を達成することに専念します。

顧客中心志向

優れたチームは皆、顧客満足を第 1 の優先事項としています。開発全体を通じて顧客志向を貫くには、顧客のビジネスの問題の理解および解決に対するチームのコミットメントが必要です。顧客中心志向が成功しているかどうかを判断するには、設計上の各製品機能から顧客またはユーザー要求までさかのぼることができるかどうか 1 つの基準になります。また、顧客満足を達成するためには、開発プロセス全体を通じて顧客に積極的に設計に関与してもらいフィードバックを提供してもらうことも重要です。これにより、チームと顧客の双方が、互いの期待とニーズをより良く整合することができます。

製品志向

製品志向は、Microsoft のように商用ソフトウェア製品を出荷するのか、それとも社内顧客のためにアプリケーションを開発するのか、についてのものではありません。これは、自らの仕事の結果を製品として扱うという考え方です。

製品志向を実現するための第 1 歩は、現在行っている作業をそれ自体 1 つのプロジェクトとして見るか、またはより大きなプロジェクトへの貢献として見ることです。実際 MSF では、チームのメンバが自身について個人としてよりプロジェクト チームの一員として考えるように、プロジェクトのアイデンティティを確立することを唱えています。そのためのテクニックの 1 つとして、Microsoft では、プロジェクトにコード ネームを付けています。これにより、プロジェクトやチームの識別が明確になり、説明責任の意識が高まります。またコード ネームは、チームの士気を高めるための手段としても役立ちます。チームのプロジェクトのコード ネームを T シャツやコーヒークップ、その他のギフト アイテムにプリントすることも、チームのアイデンティティや精神の創造と発揚に役立ちます。これは、組織内のさまざまなグループから集めたメンバで構成される "バーチャル チーム" のプロジェクトではとりわけ有効です。

プロジェクトに従事しているということが理解できれば、後は最終的な成果物が何であろうと製品と見なされるべきだということを理解するだけです。製品の作成に適用される (MSF が推奨しているような) 原則やテクニックは、プロジェクトを成功させるうえでも役に立ちます。

また、製品志向を持つということは達成に重点を置くことであり、プロジェクトの最終的な成果物をそこに至るまでのプロセスより重視するということです。プロセスが悪いとか重要でないとか言っているわけではありません。プロセスは最終的な目標を達成するために使用されるものであり、プロセスを使用すること自体を目的とするべきではないと言っているだけです。製品志向が受け入れられていれば、チームのすべてのメンバがその製品の提供に対する責任を感じるようになるはずです。

かつて Microsoft のプログラム マネージャだった Chris Peters は、1991 年のプレゼンテーションの中で、ソフトウェア開発における製品志向について次のように述べています。

「すべてのメンバの仕事はまったく同じで、皆まったく同じ職務記述書を持っています。そしてそれは、製品の出荷です。コードを書くことでもなければ、テストを行うことでもありません。仕様書の作成でもありません。製品の出荷がすべてのメンバの仕事であり、製品開発グループの仕事です。

「開発者やテスタとしての役割は二次的なものです。それが重要ではないと言っているわけではありません、もちろん重要です。しかしそれは、本当の仕事、すなわち製品の出荷の次に重要ということです。

「朝目を覚まして職場に向かうときに、「大事なことは何か？ 製品の出荷か、それともコードを書くことか？」と自分に問いかけてみてください。その答えは、製品の出荷です。コードを書くことが目標ではありません。コードを書かないようにすることが目標です。」

無欠陥志向

成功するチームでは、製品の品質に対する責任をすべてのメンバが感じています。品質に対する責任は、あるチームメンバから別のチームメンバまたは職能に委任することはできません。同様に、すべてのチームメンバが顧客の代弁者となり、開発サイクル全体を通じて常に製品の最終的な使い勝手を考慮しなければなりません。

無欠陥志向は、品質に対するコミットメントです。それは、たとえ明日納品しなければならなくなっても何かしら納品できるように、常に可能な限り高い品質で作業することをチームの目標にするということであり、ほとんど出荷できる状態の製品が毎日あるようにするという考え方です。完全無欠のコードにするということではありません。ピジョン化フェーズでプロジェクトのスポンサーによって設定され、チームで受け入れた品質基準を、製品が超えるか満たすようにするということです。

この概念は、自動車の組み立てラインにたとえるとわかりやすくなります。従来は、作業員が個々の部品を組み合わせて自動車を組み立てていました。この場合、それぞれの作業員は、自分の担当範囲の品質に対してのみ責任を負います。自動車が組み立てラインから送り出されてくると、販売のための品質基準が満たされているかどうかを検査員が確認します。しかし、製造プロセスの最後にすべての問題を検出するというこのやり方はコスト的に問題があります。というのも、その時点では、問題の修正に多大なコストがかかるからです。また、品質を予測できないため、プロセスの最後に製品が販売可能かどうかを判断するためにどのくらいの時間が必要かを予測することもできませんでした。

これに対して、近年の自動車製造においては、“品質第一” になっています。つまり、ドアの取り付けやラジオの設置などの作業がなされる都度、検査員が進行中の作業を検査して、その自動車に対して定められている品質基準が満たされているかどうかを確認します。組み立てプロセス全体を通じてそのレベルの品質が維持されれば、プロセスの最後に自動車の品質を確認するために必要となる時間やリソースを大幅に減らすことができます。また、検査員が最終的に行う検査では、個々の部品について検査する必要はなく、全体の調整を確認するだけで済むため、プロセスの予測可能性も向上します。

意欲的学習

学習意欲には、知識の収集および共有による継続的な自己改善に対するコミットメントが含まれます。これにより、チームのメンバは、失敗から学んだ教訓を糧としたり、他のメンバの実証済みのプラクティスを実施することによって成功を繰り返したりすることができます。マイルストーン レビューの実施と非難のないポストモーテムは、MSF プロセス モデルの構成要素であり、チームのコミュニケーションに対するコミットメントを促進します。学習、レビュー、およびポストモーテムの時間をスケジュールに組み込んだチームは、継続的な改善と成功に向けて環境を整えることができます。また Microsoft では、学習と知識の共有を個人の成績評価の指標に加えることによって、学習意欲に満ちた文化を作り出すことに成功しています。

チームのモチベーション

モチベーションの低いチームには 2 つの問題があります。第 1 に、チームの個々のメンバの仕事の能率が下がり、品質や生産能力の低下につながります。第 2 に、目先の目標にとらわれるようになり、自分たちの作業が同僚に与える影響を意識しなくなります。高度な知的やり取りに基づく IT プロジェクトにとっては、どちらも深刻な影響を与える問題となります。

MSF では、チームの士気やモチベーションを高める努力をするように主張しています。Microsoft で働いたことのある人々は、これを同社の特徴の 1 つとして認識しています。モチベーションを高めるためのテクニックを以下にいくつか紹介します。

- チームのビジョンを明確にします。
- プロジェクトのコード ネームやマスコット、T シャツ、カップなどのチーム グッズを使用してチームのアイデンティティを確立します。
- 社交行事やチームのイベントを企画してメンバの親睦を深めます。
- チーム確立セッションをスケジュールに組み込みます。このセッションでは、チームのメンバがさまざまな共同作業の方法を試してみることができます。通常は、実際の作業環境とは別の環境で行われます。
- 人的または技術的な能力を伸ばす機会の提供、仕事と生活のバランスへの影響の管理など、個々のメンバの個人的な目標が考慮されていることを確実にします。
- 個々のメンバが最大限にエンパワーメントされていると感じるようにし、各人の意見に耳を傾けます。
- 成功を称えます。

実証済みのプラクティス

MSF チームのメンバは、常に成功に焦点を定めて行動するために、以下の実証済みプラクティスを実践します。

少人数で、他分野にまたがるチームを使う

少人数で他分野にまたがるチームには、大規模なチームに比べて反応がすばやいなど、生来の利点があります。したがって、大規模なプロジェクト チームでは、チーム内チームを作成して、複数の小規模なグループが並行して作業するのが得策です。特定分野を専門とする、もしくは特定分野に的を絞っているメンバには、必要に応じて活動を管理する権限を与えます。

チームの中はもちろん役割群の中にさえ、特定のスキル集合を必要とする複数の専門分野があります。さまざまな経歴、訓練経験、専門分野を持つメンバによってチームや役割が構成されていれば、それぞれの独自の観点が役割や究極的にはソリューション全体に持ち込まれるため、全体としての製品品質の向上につながります。

同じ場所で一緒に仕事する

チーム モデルの目標の 1 つは、コミュニケーションのオーバーヘッドを削減することによって、チームの効果的なコミュニケーションを阻害する要素を減らすことです。チームの構造のほか、チームの地理的な分散や場所も、チームの内外との効果的なコミュニケーション能力に大きく影響します。

Microsoft について研究した著書『*Microsoft Secrets*』の中で、Michael A. Cusumano と Richard W. Selby は次のように述べています。

「... 開発が 1 か所で行われる場合は、プロジェクトのメンバが実際に定期的に集まって意見交換しながらアイデアを練ることが可能です。この頻繁かつ簡便なコミュニケーションによって、重要な問題の悪化を防ぐことができます。」

チームが 1 か所に集まって作業することは、チームのアイデンティティや団結力を高めるうえでも役立ちます。

ビルの同じ区画で作業したり、オフィスを共有したり、チームが集まるための場所を確保したりするなどのコロケーション(同じ場所に配置すること)は、かつてはオープンなコミュニケーションを促進するうえで最も効果的な方法とされていました。オープンコミュニケーションは MSF チームの成功に不可欠な要素です。

コロケーションは現在でも第 1 の選択肢ですが、ビジネスの性質の変化やコミュニケーションのためのテクノロジーの発展の結果、現在では "バーチャル チーム" でも十分成功を収めることが可能です。

バーチャル チームとは、主に電子的な手段を用いてコミュニケーションやコラボレーションを行う従業員のチームです。バーチャル チームでは、組織の境界や空間、時間などをまたぐコミュニケーションが行われます。インターネットを介したリアルタイムの共同作業は、人々の仕事や情報共有の仕方を根本から変えつつあります。インターネットは、チームのメンバー間のコミュニケーションの新たな標準となりつつあります。また、コラボレーション ソフトウェアによって、さらなる生産性向上への道も開かれています。

このバーチャル チームの概念は重要です。というのも、役割を調和した組織単位内に閉じこめる組織の境界がなくなると、そのバーチャルな側面によって、より強力なコミュニケーション、信頼、関係、明確な実行計画、自動化ツール（実施事項が失われないようにするためのプロジェクトやタスクの追跡を支援する）などが必要となるからです。

バーチャル チームにおいては、各役割がそれぞれの責務を果たすために互いに依存関係や信頼関係を築くことができなければなりません。これは、組織の文化、適切な管理、(可能であれば) 同じ場所で実際に一緒に作業する時間を持つことなどによって可能になります。

業界調査によると、バーチャル チームのメンバを選出する際に、コミュニケーション スキルやチームへの適性はほとんど注目されていません。アナリストは、この見落としが多くのバーチャル チームの失敗の重要な要因であると指摘しています。バーチャル チームを発足させる際には、次のような特性を持つメンバを探してください。

- 独立して仕事ができる。
- リーダーシップのスキルを発揮できる。
- ソリューションで必要となる特定のスキルを持っている。
- 組織と知識を共有できる。
- 効果的な仕事の仕方の開発の役に立てる。

全員が参加してソリューション設計を作り上げる

製品仕様の作成には各役割が参加します。これは、各役割はそれぞれ設計に対する独自の観点をもち、個々の目標やチームの目標との関係もそれぞれ異なるためです。各役割が設計に参加することによって、チームのさまざまな観点から最高のアイデアが表出されるような環境が育まれます (1)。

チーム モデルの概要

MSF の土台となっているのは、プロジェクトが成功と見なされるには 6 つの主要な品質目標が達成されていなければならないという信念です。これらの目標がチームの原動力となり、チーム モデルを定義しています。プロジェクトの成功は確かにチーム全体の責任ですが、その一方でチーム モデルでは、説明責任と焦点を明確にするために、6 つの品質目標をそれぞれ異なる役割群に関連付けています。

プロダクトマネジメント、プログラムマネジメント、開発、テスト、ユーザエクスペリエンス、およびリリースマネジメントの 6 つのチーム モデルの役割群は、一連の職能領域の組み合わせと、それに関連する責務を特定するための共通の方法を定義します。役割群は多くの場合、単純に "役割" と呼ばれます。どちらの呼び方でもその概念は同じです。MSF とチーム モデルはある特定のソリューションのニーズを満たすためのスケーラビリティを備えています。役割または役割群は、プロジェクトの規模や複雑さ、職能領域の責務を果たすために必要なスキルによって、1 人の人物である場合もあれば大勢の人々である場合もあるのです。

MSF チーム モデルでは、役割群をビジネス ニーズに合わせることの重要性を強調しています。それぞれに異なる専門分野や焦点を必要とするような、関連する職能領域と責務を役割群としてまとめます。これにより、プロジェクトの基本的な目標のすべてを代表するスキルと観点を有するバランスの取れたチームを動機付けることができます。明確に定義された目標を持つことによって責務の理解が深まり、プロジェクト チームの「自分の仕事である」という意識が促進されます。そしてそれは最終的に、製品の品質向上につながります。プロジェクトの成功にはそれぞれの目標の達成が欠かせないため、それらの目標を代表する役割は、意思決定にあたって対等な発言権を持つ対等な存在と見なされます。

これらの役割群は、いかなる組織図や肩書きも示唆したり推奨したりしません。組織図や肩書きは組織やチームによってさまざまに変化するからです。役割は IT 組織内のさまざまなグループにまたがって割り当てられることが多く、ビジネス ユーザー コミュニティや外部のコンサルタントやパートナーを含むこともあります。重要なのは、特定の役割群とそれに関連する職能、責務、および目標への貢献を果たす個人をチームにおいて明確に決めることです。

役割群	目標	職能 領域	責務
プロダクトマネジメント	満足した顧客	マーケティング ビジネス上の価値評価 顧客代弁 製品計画	顧客の代弁者として活動する 共有されたプロジェクトのビジョン/スコープを推進する 顧客要求の定義を管理する ビジネス ケースを開発および保守する 顧客の期待を管理する 製品機能、スケジュール、および資源の間のトレードオフの決定を推進する マーケティング、伝道および広報宣伝活動を管理する 広報計画を作成、保守、および実行する
プログラムマネジメント	プロジェクトの制約内でのソリューションの提供	プロジェクトマネジメント ソリューション アーキテクチャ プロセス保証 管理サービス	製品を納期通りに出荷するように開発プロセスを推進する 第一プロジェクト アーキテクトとして製品仕様を管理する チーム内のコミュニケーションや交渉を促進する プロジェクトのスケジュールを維持し、プロジェクトの状況を報告する 重要なトレードオフ決定の実施を推進する プロジェクトのマスタ計画とマスタ スケジュールを作成、保守、および実行する リスク査定とリスクマネジメントを推進および管理する
開発	仕様に沿った構築	技術コンサルティング 実装アーキテクチャと設計 アプリケーション開発 インフラストラクチャ開発	物理設計の造作を特定する 各製品機能を完成させるための時間と作業量を見積る 製品機能の構築を行うあるいは監督する 展開 に向けて製品を準備する 技術的な専門知識をチームに提供する
テスト	製品のすべての品質問題を識別および対処した後のリリース承認	テスト計画 テストエンジニアリング テスト報告	すべての問題を既知とすることを確実にする テストの戦略および計画を作成する テストを実施する
ユーザエクスペリエンス	向上されたユーザーの有効性	テクニカル コミュニケーション トレーニング ユーザビリティ グラフィック デザイン 国際化 アクセシビリティ	チーム内でユーザーの代弁者として活動する ユーザー要求定義を管理する パフォーマンス サポート システムを設計および開発する ユーザビリティとユーザー パフォーマンス向上のトレードオフの決定を推進する ヘルプ機能およびヘルプ ファイルの仕様を提供する ユーザー トレーニングを開発および提供する
リリースマネジメント	円滑な展開と継続的な運用	インフラストラクチャ サポート 運用 商用リリースマネジメント	運用、サポート、および流通経路の代弁者として活動する 調達を管理する 製品の展開 を管理する 可管理性とサポート性のトレードオフの決定を推進する 運用、サポート、および流通経路との関係を管理する プロジェクト チームのロジスティクス支援を提供する

満足した顧客

プロジェクトの成功には、顧客やユーザーのニーズを満たさなければなりません。予算や納期の目標が達成されても、顧客のニーズが満たされていないればプロジェクトが成功したことにはなりません。

プロジェクトの制約内でのソリューションの提供

プロジェクトの制約の中で遂行することは、すべてのチームの重要な目標です。どのようなプロジェクトにも付きものの基本的な制約として、予算とスケジュールがあります。たいいていのプロジェクトでは、予算とスケジュールが成功の目安になります。

仕様に沿った構築

製品仕様には、チームが顧客に提供する成果物の詳細が記されています。仕様は何を構築するかについてのチームと顧客の間の合意を表すものであるため、可能な限り仕様に忠実に提供することがチームにとって重要です。

製品のすべての品質問題を識別および対処した後のリリース承認

欠陥のないソフトウェアはありません。製品のリリース前にこうした欠陥を識別して対処することを実にすることは重要な目標の1つです。対処の仕方は、当該欠陥の修正から回避策を記したドキュメントの提供まで、どのようなものでもかまいません。対処済みの既知の欠陥を含む製品を回避策を添えて納品する方が、チームや顧客を後から驚かせるような未確認の欠陥を含む製品を納品するより望まれます。

向上されたユーザーの有効性

製品が成功するためには、ユーザーの仕事やパフォーマンスを高めるものでなければなりません。いくら機能や内容が豊富であっても、対象ユーザーが使えるものでなければ、その製品は失敗と見なされます。

円滑な展開と継続的な運用

時に円滑な展開の必要性が見過ごされることがあります。展開の印象は、それが正しいかどうかはともかく、製品自体の印象に影響します。たとえば、インストール プログラムに問題があると、実際には何の問題もなくとも、ユーザーはインストールされるアプリケーション自体にも同様に問題があると思うかもしれません。したがって、チームはただ単純に展開するだけでなく、円滑な展開に励み、製品のサポートや管理の準備を整えておく必要があります。たとえば、トレーニング、インフラストラクチャ、およびサポートを展開に先立って準備しておきます。

チーム モデルの役割群

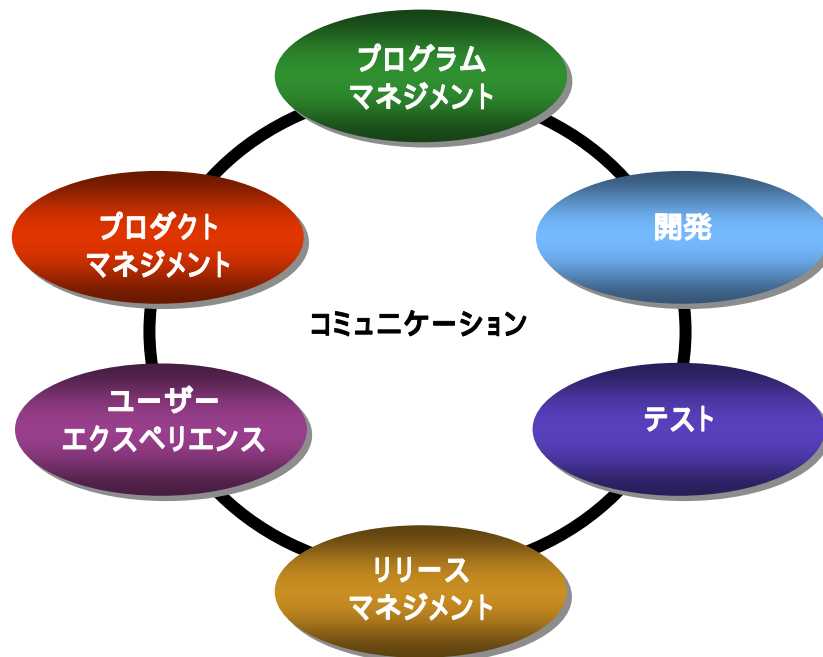


図 1: チーム モデルの役割群

プロダクトマネジメント役割群

プロダクトマネジメント役割群の主な目標は、顧客満足の達成です。プロジェクトの成功には、顧客のニーズを満たす必要があります。しかしながら、そのためにはまず顧客を明確に識別し、理解しなければなりません。ソリューションや一連の機能を要求している顧客と、費用を支払ったり活動を支えるスポンサーとが異なる場合もあります。したがって、両者の明確な区別と、それぞれの成功要因についての要求分析が必要になります。これを行ってはいじめて、期待を設定しそれを充足させる責務を適切な職能領域に割り当てることができます。予算や納期の目標が達成されても、顧客とビジネスのニーズが満たされていないならばプロジェクトが成功したことにはなりません。

MSF チーム モデルでは、一連の責務をより厳密に定義するために、各役割群を複数の職能領域に分けています。職能領域の一連の責務は、一緒に取り扱う場合には大抵共通のスキル セットを形成します。

プロダクトマネジメント役割群では、満足した顧客という目標を達成するために、製品計画、ビジネス上の価値評価、顧客代弁、およびマーケティングといった職能領域が必要になります。

職能領域

マーケティング

- マーケティングや対象顧客に向けた広報活動を推進します。
- 競合ソリューションとの差別化を図ります。
- 対象顧客が容易に取得できるようにソリューションを配布します。
- ソリューションの購入と使用によって顧客が肯定的な体験を得られるようにサポートを提供します。

ビジネス上の価値評価

- プロジェクトのビジネス上の正当性を定義し、維持します。
- ビジネス上の価値の実現とメトリックについて定義し測定します。

顧客代弁

- プロジェクトおよびソリューションの共有ビジョンを推進します。
- 顧客の期待および顧客とのコミュニケーションを管理します。

製品計画

- 顧客とビジネスの要求を収集、分析、および優先順位付けします。
- 市場調査を実施し、市場の需要および競合情報の収集と分析を行います。
- ビジネス メトリクスと成功基準を決定します。
- 複数のバージョンからなるリリース計画を識別します。

マーケティング

マーケティングとは、製品、ソリューション、またはサービスの販売促進、販売、および流通のプロセスまたはテクニックです。マーケティングには、発売マーケティング、維持マーケティング、広告宣伝など、さまざまな側面があります。ソリューションのライフサイクルの中で、マーケティングの焦点は変わっていきます。適切なレベルでマーケティング活動を行うためには、ソリューションが現在ライフサイクルのどの位置にあるのかを把握することが重要です。

ビジネス上の価値評価

ビジネス上の価値評価の職能領域においてプロダクトマネジメント役割群は、IT ソリューションへの投資に対する財務面および営業利益における収益についての予測値を、顧客のビジネス意思決定者（BDM; Business Decision Maker）の必要に応じて簡潔にまとめて提供します。

有用なソリューションを効果的に提供できるようにするためには、プロダクトマネジメントが顧客のビジネス、成功要因、および主要な業績指標を把握しなければなりません。こうした知識を獲得するプロセスは、ビジネス上の価値評価または重要成功要因の識別として定義できます。何が顧客を成功に導くのかを把握できれば、適切なソリューションの見極めや提案において役に立つのは明らかです。近年 IT 投資にはますます厳しい監視の目が注がれるようになっており、多くの IT 関連の契約では、プロジェクトが認められる前に財務上のレビューが必要とされるようになっています。客観的な費用便益分析を行うことによって、顧客満足の実現の可能性を高めることができます。財務的な成果の計算が終わると、IT 投資のビジネス ケースの作成が完了します。

顧客代弁

この職能領域の責務には、高次のコミュニケーションや顧客の期待の管理が含まれます。高次のコミュニケーションには、広報活動、企業の上層部や顧客への状況説明、ユーザーに対するマーケティング、デモンストレーション、および製品の発売が含まれます。いったんビジョンが設定されると、期待の管理がプロダクトマネジメントの主な役割となります。プロジェクトの成否は顧客の期待にも左右されるため、この役割はきわめて重要なものと見なされます。

期待を効果的に管理することがいかに重要かを、特定の期日までに製品の 10 の機能が顧客に納品されることになっていた場合を例に考えてみましょう。10 の機能すべてが納品されることを顧客が期待しているにもかかわらず、チームが 2 つの機能しか納品しなかった場合、そのプロジェクトは顧客からもチームからも失敗と思われるでしょう。

しかし、プロダクトマネジメントが、その機能の開発および運用の期間全体を通じて顧客との間に双方向のコミュニケーションを維持していれば、顧客の期待を変えることによってプロジェクトを成功に導くことも可能です。また、トレードオフの決定のプロセスに顧客を含めることによって、リスクの変化やその他の課題を伝えることもできます。この場合、前のシナリオとは違って、顧客が状況を評価できるようになります。その結果、10 の機能すべてを指定の期日までに納品することは非現実的であり、さしあたって 2 つだけ納品するのが妥当だということについて、チームと顧客の間に合意を成立させることができます。このシナリオでは、2 つの機能の納品が顧客の期待と合致するため、チームと顧客の双方がプロジェクトを成功と考えるでしょう。

製品計画

製品計画では、複数のバージョンのソリューションの要求と製品機能の集合を識別します。製品企画の目標は、プログラム マネージャや開発者ができるだけ短い時間でソリューションの要求を簡単に理解できるようにすることです。そのためには、第 1 に、ソリューションの現在の要求を徹底的に把握する必要があります。つまり、ビジネスのニーズは何か、顧客はソリューションをどのように使用するか、どのようなサポートの問題があるか、ほかにどのような代案が考えられるか、などを特定します。第 2 に、ソリューションを使用する顧客の価値を高める機能について調査します。たとえば、新たな事業分野への参入を可能にする能力、他のシステムとの統合、生産性の向上、他のソリューションからのアップグレード、サポート コストの削減などが考えられます。これらの情報に基づいて製品計画の担当者は、ソリューションの各リリースにどの製品機能を割り当てたらよいかを提案したり、機能リストの優先順位付けを行ったりできます。

製品計画の核となるのは調査と分析です。顧客やビジネスのニーズを理解するにも、市場の競合状況を把握するにも、結局、適切な調査と分析が欠かせません。これによって、不要な機能がソリューションに組み込まれるのを防ぐことができます。

プログラムマネジメント役割群

プログラムマネジメントの役割の焦点は、プロジェクトの制約の中でソリューションを提供するという目標を達成することです。この目標は、見方を変えれば、プロジェクトのスポンサーがプロジェクトの結果に満足するようにする、ということになります。この目標を達成するために、プログラム マネジメントは、スケジュール、製品機能集合、およびプロジェクトの予算を担当および推進し、適切なソリューションが適切なときに納品され、プロジェクト全体を通じてプロジェクトのスポンサーの期待が把握および管理されるようにします。プログラムマネジメントには、たとえば次のような機能領域が含まれます。

プロジェクトマネジメント

- 予算の追跡と管理を行います。
- マスタープロジェクトスケジュールを管理します。
- リスクマネジメントプロセスを推進します。
- チーム内のコミュニケーションや交渉を促進します。
- 進捗状況を追跡し、プロジェクトの状況報告を行います。
- リソースの配置を管理します。

ソリューション アーキテクチャ

- ソリューション全体の設計を推進します。
- 機能仕様書を管理します。
- ソリューションの範囲と重要なトレードオフの決定を管理します。

プロセス保証

- プロセスの品質保証を推進します。
- 問題点を明らかにし、改善を勧めます。

管理サービス

- プロジェクトマネジメントプロセスを実施し、それを使用するチーム リーダーをサポートします。
- チームが効率的に作業できるように一連の管理サービスを提供します。

プロジェクトマネジメント

プロジェクトマネジメントは、スケジュールの所有者として、チームのすべてのスケジュールを収集および確認し、マスタ スケジュールに統合します。そしてそのマスタ スケジュールを追跡し、チームやプロジェクト スポンサーへの報告を行います。

また、予算の所有者として、チームのすべての役割から資源の要求を集めてプロジェクトの予算の作成を促進します。すべての資源(ハードウェア、ソフトウェア、および人)に関する決定は、プロジェクトマネジメントが把握および合意している必要があります。また、計画に対して実際にどれだけ消費されたのかを追跡する必要があります。チームとプロジェクト スポンサーに状況を報告します。

更にプロジェクトマネジメントは、資源の調整、チームのコミュニケーションの促進、重要な決定の推進支援も行います。

プロジェクトマネジメントの機能領域と、プロジェクトマネジメントの責務に対する MSF のアプローチの詳細については、<http://www.microsoft.com/japan/msdn/vstudio/productinfo/enterprise/msf/>を参照してください。

ソリューション アーキテクチャ

ソリューション アーキテクチャはプログラムマネジメント役割群の機能領域であり、ソリューションの論理設計と機能仕様書に対する責務を負います。ソリューション アーキテクチャの焦点は、対象ユーザーがソリューションを使って、効果、効率、および満足についての明示された目標を達成できるようにすることです。

ソリューション アーキテクトの責務には次のようなものがあります。

- ソリューション全体の設計を推進します。
- 機能仕様書を管理します。
- ソリューションのスコープおよび重要なトレードオフの決定を管理します。

ソリューション アーキテクチャは、論理設計の所有者として、ソリューションのビジネス面（概念設計においてプロダクトマネジメントが代表する）と技術面（物理設計において開発が代表する）の間のきわめて重要な連結を提供します。ソリューション アーキテクチャは機能仕様書の管理人として活動します。ソリューションの内容と設計について、他の役割のさまざまな要求の中で合意に達するようにチームを推進し、合意に達したアプローチの正当性をプロジェクトのステークホルダーに説明します。また、製品機能を要求（および究極的にはビジネス上の価値の創出）までさかのぼって追跡できるようにすることもソリューション アーキテクチャの責務です。これにより、すべての製品機能が定められた要求を支援しているかどうかを確認したり、機能を変更した場合のソリューションの価値に対する影響をチームが評価したりできるようになります。

ソリューション アーキテクチャの活動には以下が含まれます。

- ソリューション コンセプトを作成し、それを顧客のエンタープライズ アーキテクチャに合わせて調整します。複数のバージョンからなるリリース戦略を考案します。要求獲得の計画のレビューを行います。
- アーキテクチャ/標準化グループの要求や、相互運用性に関する要求を獲得します。論理設計プロセスを推進します。製品機能を要求や便益までさかのぼって追跡できるようにするためのトレーサビリティ マップを提供します。機能仕様書を作成します。内部リリースを定義します。
- 機能仕様書に対する変更を管理します。トレーサビリティ マップを管理します。仕様をチームの他の役割や外部のステークホルダーに明確に伝えます。相互運用性の問題について他のプロジェクト チームと連絡をとります。
- トリアージ プロセスに参加します。ソリューションの内容に関するプロジェクト ステークホルダーの期待を管理します。
- エンタープライズ アーキテクチャ チームに更新を提供します。将来のバージョンのリリースのために要求を更新します。

ソリューション アーキテクチャの担当者は、広範な知識と経験に加え、技術的な問題をその裏側にあるビジネスのニーズに関連付けられる能力を持った、技術に精通した人物である必要があります。ソリューション アーキテクトは、ソリューションで使用されている個々のテクノロジーの専門知識については開発チームを頼ることもできますが、そうした技術的な詳細の含意を非常にすばやく把握し、それぞれの技術の相互関係やソリューションが展開される環境への影響を理解できなければなりません。また、そうした影響について顧客のアーキテクトと話し合い、提案するソリューションとエンタープライズ アーキテクチャとの対立をすばやく解決できる必要もあります。

プロセス保証

プログラムマネジメントの職能領域であるプロセス保証は、欠陥の発生源を排除することに重点を置きつつ、プロジェクト全体の品質目標の達成に焦点を当てたプロセスをプロジェクト チームが採用することを確実にします。プロセス保証には、次の 2 つの主な領域に対する責務があります。

- チームが使用する主要なプロジェクト プロセスの定義、およびチームがそのプロセスを実施する際の助言と指針の提供
- プロセスの関連性と有効性についての妥当性確認するためのレビューの実施、改善の提案、および遵守状況の監視

プロセス保証では次の活動に焦点を置きます。

- プロジェクトの設計に沿ってプロジェクトの規約やプロセスを定義します。
- プロジェクト プロセスの効果的な実施についての助言や指針を提供します。プロセスの遵守について確認します。マイルストーン レビューを実施します。プロセスの改善を提案します。

プロセス保証では、外部からの視点を持てるように、プロジェクト チームからある程度の独立が保たれていると有効です。このためプロセス保証は、フルタイムの役割にするにはプロジェクトの規模が小さすぎる場合でも、しばしばプロジェクト チームの外部から管理されます。

管理サービス

プログラムマネジメント役割群の機能領域である管理サービスは、プロジェクトマネジメントプロセスの実施とプロジェクト チームの管理サポートに対する責務を負います。

管理サービスは、プロジェクト チームが、プロジェクトマネジメントによって定義されたプロジェクト設計の仕様を満たすプロセスを実施することを確実にします。煩雑な手続きをできる限り減らして、プロジェクト チームが効果的に活動できるようにする必要があります。

管理サービスには以下の責務があります。

- プロジェクトマネジメントプロセスの実施と、それらを使用するチーム リーダーの支援。これには、マスタ計画やマスタ スケジュールの保守のためのチームの入力の集約と、チーム リーダーによる進捗報告の支援を含む。
- 効率的なチーム作業を支援する一連の管理サービス（例えば会議スケジュールの設定、一般的な調達、契約の管理など）の提供。

管理サービスでは次のような活動が焦点になります。

- 協力会社からのチーム メンバの効率的な補充など、プロジェクトの立ち上げのプロセスを支援します。契約を管理します。チームの設備（作業空間、電話、セキュリティ アクセスなど）を整備します。
- 一貫した計画の枠組みを確立します。チーム リーダーによる計画とスケジュールの作成を支援します。チームの入力を集約してマスタ計画やマスタ スケジュールを作成します。財務報告と進捗報告のプロセスを確立します。
- チーム リーダーによる進捗報告を支援します。全体的な進捗報告書および財務報告書を作成します。
- プロジェクト完了時に、すべての管理システムを閉じることを確実にします。
- 会議スケジュールの設定、リスクマネジメントプロセスや問題管理プロセスの実施などの一般的な管理サポート活動を遂行します。マスターリスクリストや活動リストなどを保守します。財務報告書と進捗報告書を作成します。チームの士気を高めるためにチームの場所を管理します。

管理サービスの役割には、高度な管理能力と細部に対する注意力に加え、プロジェクトの計画およびスケジューリングのテクニックの確かな経験が求められます。このほか、供給者組織で実施されているポリシーやガイドラインをよく理解している必要もあります。比較的大規模なプロジェクトでは、プロジェクトの監督者のそばで働くすばらしい機会となり、将来プロジェクトを監督するために必要な経験が得られます。

開発役割群

MSF プロジェクトにおける開発役割群の焦点は、"仕様に沿った構築" です。この品質目標を達成するためには、開発役割群は顧客の期待と機能仕様書に明記されている仕様を満たすソリューションを構築しなければなりません。また、開発役割群は、機能仕様書と共にソリューションの全体的な仕様を形成する、ソリューションのアーキテクチャと設計を堅守します。

開発役割群は、ソリューションの構築者としてだけでなく、チームの技術コンサルタントとしても務めます。技術コンサルタントとしては、設計やテクノロジーの選択について入力を提供したり、意思決定の妥当性確認や開発リスクの軽減に役立つ機能プロトタイプを作成したりします。

構築者としては、ソリューションと製品機能の物理レベルの設計を提供し、その設計に基づく提供に必要な作業量を見積もった後に、ソリューションを構築します。開発が自身の作業量とスケジュールを見積もるのは、開発にまつわるあらゆる不確定要素を日々扱っているのは開発役割群だからです。MSF ではこの概念をボトムアップの見積もりと呼んでおり、MSF の思想の基本要素の 1 つとなっています。そこでの目標は、より高品質なスケジュールの実現と、見積もりの提供者およびその業務の遂行についての説明責任の向上です。

技術コンサルティングの職能領域

- チームの技術コンサルタントとして務めます。
- テクノロジーの評価と妥当性確認を行います。
- 機能仕様書の作成とレビューに積極的に参加します。
- 組織の開発標準の定義に貢献します。

実装アーキテクチャと設計の職能領域

- エンタープライズ アーキテクチャ (EA) のアプリケーション、データ、およびテクノロジーの観点にソリューション固有の詳細を提供することによって、エンタープライズ アーキテクチャ (EA) をソリューションの実装アーキテクチャに関連付けます。
- ソリューションの論理設計と物理設計を所有および実装します。

アプリケーション開発の職能領域

- 設計仕様を満たすように製品機能のコーディングを行います。
- 開発の過程でコードレビューを行い、知識や経験を共有します。
- テスト役割の協力を得て、テスト計画で定義されている単体テストを実施します。

インフラストラクチャ開発の機能領域

- 設計仕様を満たすように製品機能を開発します。
- 開発の過程でコードレビューを行い、知識や経験を共有します。
- テスト役割の協力を得て、テスト計画で定義されている単体テストを実施します。
- 展開の自動化のためのスクリプトを開発します。
- 展開のドキュメントを作成します。

技術コンサルティングの機能領域

技術コンサルティングの機能領域は、プロジェクト ライフサイクル全体を通じて技術リソースとして務めます。開発役割群が技術コンサルタントとして行わなければならない作業には、概要の設計に対する入力提供、テクノロジーの評価と妥当性確認、開発プロセスの早い段階で開発リスクを軽減するための調査の実施などがあります。

ビジョン化フェーズにおいては、ユーザーや顧客の要求を実装する立場の観点から分析することがこの機能領域の焦点となります。初期条件における実装の実現可能性をプロジェクトの技術的な面から評価することによって、プロジェクトのビジョンスコープ文書の定義に貢献します。また、実装のためのさまざまなアプローチの長所と短所についての指針を提供し、初期のテクノロジーの選択の妥当性を確認します。その過程で、調査を実施したり、社内または社外の相手の意見を聞いたり、テクノロジーの提供者と議論したりすることもあります。さらなる確認のために、限定された機能のプロトタイプを概念検証 (proof of concept) として開発することもあります。新しいテクノロジーの使用が必要とされるプロジェクトや、プロジェクト チームの経験が乏しい領域のプロジェクトでは、特にこうした作業が役に立ちます。

実装アーキテクチャと設計の機能領域

この機能領域は、MSF プロジェクトにおけるソリューションの実装アーキテクチャの定義とソリューション設計の開発に関連する一連の責務を表します。

設計の観点から見ると、ソリューションの全体的なアーキテクチャや、そのエンタープライズ アーキテクチャにおける位置付けは、プログラムマネジメントの責務になります。開発の責務は、エンタープライズ アーキテクチャのアプリケーション、データ、およびテクノロジーの観点にソリューション固有の詳細を提供することによって、エンタープライズ アーキテクチャをソリューションの実装アーキテクチャに対応付けることです。

MSF では、概念設計、論理設計、および物理設計からなる 3 層の設計プロセスを提案しています。プログラムマネジメントとプロダクトマネジメントは、概念設計を共同で所有します。概念設計には、ユーザー シナリオ、およびそのユーザビリティ分析、概念データのモデリング、および初期のテクノロジーの選択が含まれます。開発は、ソリューション設計の論理的な側面と物理的な側面を所有します。論理設計と物理設計では、関連技術の知識や、技術の選択がソリューションの設計に与える影響の把握が必要となります。

アプリケーション開発の職能領域

アプリケーション開発の職能領域は、MSF プロジェクトにおけるソフトウェア アプリケーションの開発に関連する一連の責務を表します。この職能領域における開発の役割の最も重要な責務は、要求されたソリューションの製品機能を仕様と設計に沿って構築し、単体テストを実施し、テストの過程で見つかった品質の問題に対処し、ソリューションのコンポーネントを統合して最終的な成果物を仕上げることです。

開発役割は標準を規定する作業に貢献し、ソリューションの開発の間はその標準に準拠します。また、コードのレビューを行って、アプリケーションの機能の単体レベルでの品質を評価します。レビューは、知識と経験をチームのメンバが共有できるようにするので、MSF のプロジェクト チームの目標の 1 つである "意欲的学習" を支援します。開発の役割は、実装された機能の単体テストを十分に行い、その結果を文書化する必要があります。テストの役割は、積極的に開発役割に協力して、ソリューションの機能の品質評価を計画および実施します。その際には、機能を単独で評価するだけでなく、完全なソリューションの一部としても評価します。

インフラストラクチャ開発の職能領域

インフラストラクチャ開発の職能領域は、MSF プロジェクトにおけるソリューションのシステムおよびソフトウェア インフラストラクチャの開発に関連する一連の責務を表します。システム インフラストラクチャには、分散コンピューティング環境のネットワーク インフラストラクチャ、クライアントとサーバーのシステム、およびあらゆるサポート コンポーネントが含まれます。ソフトウェア インフラストラクチャには、クライアントとサーバーのオペレーティング システムのほか、必要なプラットフォーム ソフトウェア サービス、例えばディレクトリ、メッセージング、データベース、エンタープライズアプリケーション統合(EAI)、システム管理、ネットワーク管理など、を提供するソフトウェア製品が含まれます。

インフラストラクチャ開発において開発の役割は、設計で指定されているインフラストラクチャを "開発" します。これには、ソリューションの基礎的なテクノロジインフラストラクチャ（たとえばネットワーク サポートなど）を構成することや、設計で定義されているとおりのクライアントやサーバーのシステムを構成することが含まれます。インフラストラクチャのさまざまな側面は、そこでサポートされるアプリケーションの要求に影響され、また逆にアプリケーションはインフラストラクチャ要求の影響を受けます。たとえば、ミッションクリティカルな高性能ソリューションでは、バックエンド サーバーでクラスタリングや負荷分散が必要になることがあります。その場合、ソリューションのオペレーティング システムやプラットフォーム製品もそれに合わせて "開発" されなければなりません。ソリューションのニーズを満たすために、さまざまなソフトウェア プラットフォーム製品のインストール、構成、および最適化を行う必要があります。適切なテストおよび安定化の後、リリースマネジメントの役割の責任の下でインフラストラクチャ ソリューションを広く展開します。リリースマネジメントの役割は、事前にソリューションのインフラストラクチャ要求の獲得を管理します。

テスト役割群

テスト役割群の目標は、リリースの承認の前に製品の品質問題がすべて識別および対処されるようにすることです。欠陥のないソフトウェアはありません。重要な目標は、製品のリリース前に確実にこうした欠陥を見つけて対処することです。対処の仕方には、欠陥の修正から回避策を記したドキュメントの提供まで、あらゆる方法を取り得ます。回避策を添えることで対処した既知の欠陥を含む製品を納品する方が、未確認の欠陥を含む製品を納品してチームや顧客を後から驚かせるよりはましです。

プロジェクトの成功のためには、テスト役割群が主要な責務に集中する必要があります。これらの責務は、以下の 3 つの主要な職能領域に分類できます。

テスト計画

- テストアプローチと計画を作成します。
- 品質基準の設定に参加します。
- テスト仕様を作成します。

テストエンジニアリング

- 自動化されたテスト ケース、ツール、およびスクリプトの開発と保守を行います。
- テストを実施して製品の開発状況を正確に判定します。
- ビルド プロセスを管理します。

テスト報告

- 製品の品質に関連するデータをチームに提供します。
- すべてのバグを追跡し、製品のリリース前に問題が確実に解決されるようにするために問題を伝達します。

テスト計画

テスト役割群のテスト計画の機能領域は、製品の品質の問題がすべて識別され、対処されるようにするためには、チームがどのようにすればよいのかという問題に焦点を絞ります。

テストの役割はテストアプローチと計画を作成し、それによって、チームがソリューションのテストに使用する戦略の概要を示します。これらの計画には、テストの種類、テストの対象となる領域、テストの成功基準、およびテストに必要なリソース（ハードウェアと人）の情報が含まれます。

テスト計画の機能領域の重要な責務の 1 つは、品質管理の測度とソリューションの成功基準をプロジェクト チームに提供することによって、品質基準の設定に参加することです。

テスト計画の機能領域の最後の活動は、テスト仕様の作成です。テスト仕様とは、テスト計画で定義されているニーズを満たすために必要となるツールやコードの詳細な記述です。

テストエンジニアリング

テスト役割群のテストエンジニアリングの機能領域は、製品の品質の問題がすべて識別され対処されることを確実にするために、テスト計画で定義された活動を実行することに焦点を置きます。この機能領域で定義されている責務には、テスト ケースの開発と保守のための具体的な任務、テスト機能を実行するためのツール、スクリプト、およびドキュメントの作成、テスト手順の実行と単一の評価基準での報告を確実にするためのデイルービルドの管理、製品の開発状況を正確に判定するためのテストの実施（テスト ケース、ツール、およびスクリプトを実行して現在のビルドの問題を特定する）などがあります。

追跡とレポート

テスト役割群の追跡とレポートの機能領域は、現在、ソリューションの何処がうまくいってなくて何処がうまくいっているかをプロジェクト チームに対して明確に示すことによって、開発状況を正確に描くことに焦点を置きます。

見つかったすべての問題が製品のリリース前に確実に解決されるようにするために、問題の追跡が行われます。現在の製品の品質の状態に関連するデータと詳細な傾向分析をチームに提供するために、割り当て、優先順位、解決策、回避策などを含む、問題の状態についてのドキュメントを頻繁に作成します。

ユーザエクスペリエンス役割群

ユーザエクスペリエンス役割群の目標は、ユーザーの有効性の向上です。ユーザー エクスペリエンスは、アクセシビリティ、国際化、テクニカル コミュニケーション、トレーニング、ユーザビリティ、およびグラフィック デザインの 6 つの職能領域で構成されています。ユーザー エクスペリエンス役割群の各職能領域には、それぞれソリューションを成功させるために果たさなければならない責務がいくつかあります。以下は、各職能領域と、それに関連する責務の一覧です。

アクセシビリティ

アクセシビリティの概念と要求を設計に反映させます。

国際化

国際市場におけるソリューションの品質とユーザビリティを改善します。

テクニカル コミュニケーション

- サポート システムのドキュメント (ヘルプデスクのマニュアル、サポート技術情報の記事など) を設計および開発します。
- ヘルプやユーザー補助のドキュメントを作成します。

トレーニング

- 学習戦略を開発および実施します (構築や購入および提供)。

ユーザビリティ

- ユーザーの要求の収集、分析、および優先順位付けを行います。
- ソリューションの設計へのフィードバックや入力を提供します。
- 使用シナリオとユース ケースを開発します。
- プロジェクト チームに対してユーザーの代弁者として振舞います。

グラフィック デザイン

- ユーザー インターフェイス設計を推進します。

アクセシビリティ

アクセシビリティの職能領域において焦点となるのは、アクセシビリティの概念と要求を設計に反映させることによって、ソリューションが障害を持つユーザーなどにも利用できることを確実にすることです。アクセシビリティはさまざまな理由で重要です。まず、製品やソリューションは障害の有無に関係なく、すべての人々が利用できるものでなければなりません。アクセシビリティを考慮していない製品やソリューションは、完全には受け入れられません。また、政府の規制によってアクセシビリティの遵守が必要になることもよくあります。

アクセシビリティの概念と要求はソリューションの開発サイクル全体を通じて示されなければならない、以下を含む必要があります。

- 各機能仕様へのアクセシビリティのセクションの組み入れ
- ソリューションのヘルプ セクションへのアクセシビリティ情報の統合
- アクセシビリティのドキュメントの完成を確実にすること
- アクセシビリティのドキュメントがあらゆるユーザーに利用可能な形で提供されるのを確実にすること

国際化

国際化の職能領域の責務は、国際市場におけるソリューションの品質とユーザビリティを改善することです。この職能領域は、グローバリゼーションとローカリゼーションの両方のプロセスで構成されます。

グローバリゼーション

グローバリゼーションとは、ローカライズ担当者が修正や無用な対策を行わずに済むよう、ソリューションおよびその内容のローカライズの必要性を考慮してソリューションを定義および開発するプロセスです。言い換えれば、適切なグローバリゼーションが行われたソリューションでは、ローカリゼーションの手間は最小限で済むということです。

ローカリゼーション

ソリューションのローカリゼーションには、ソリューションのユーザー インターフェイス、ヘルプ ファイル、印刷物あるいはオンライン ドキュメント、マーケティング資料、Web サイトの修正などが含まれます。グラフィック要素を特定の言語に合わせて変更したり、内容そのものを修正したりする必要が生じることもあります。

テクニカル コミュニケーション

テクニカル コミュニケーションの職能領域では、ソリューションのドキュメント サポート システムの開発が焦点になります。

この職能領域の主な責務としては、まずヘルプ ツールなどのツール コンポーネントの作成があります。ヘルプ ツールでは、基本的な質問への回答、キーワードの説明、エラーの意味、よく寄せられる質問などをユーザーに提供できます。ヘルプのようなツールはユーザーと組織の双方にメリットがあります。ユーザーにとっては、問題に対する回答を適時的かつ効果的に得られるというメリットがあります。組織にとってのメリットは、サポート コストの削減です。

このほか、ソリューションのドキュメントの設計と開発も、テクニカル コミュニケーションの職能領域の責務です。これらのドキュメントには、インストール、アップグレード、運用、トラブルシューティングのガイドなどが含まれるでしょう。

トレーニング

トレーニングの職能領域では、ソリューションを効果的に使用するために必要な技能的知識を提供することによってユーザーのパフォーマンスを向上させることが焦点になります。この技能的知識の移転は、学習戦略の実施によって達成されます。学習戦略の開発は、ユーザエクスペリエンス役割群の責務です。

学習戦略の開発は組織内で行われることもあれば、トレーニングと開発を専門とする別の組織に外注されることもあります。いずれの場合も、一般に次のような作業が必要になります。

- ユーザーの分析と組織の目標や目的の分析
- 求められるスキル水準集合の設定
- トレーニング計画の開発と実施
- 実施したトレーニング計画の有効性の評価、および必要に応じた修正

学習戦略には、講師によるトレーニング、テクノロジーを活用したトレーニング、自己学習、ジョブ エイドの使用など、1 つ以上のトレーニング提供方法が含まれます。多くの組織では、個人の学習スタイルに合わせてこれらの方法を組み合わせて利用しています。

ユーザビリティ

ユーザビリティの機能領域の焦点は、高いレベルの有効性、効率性、および満足と特定された目標を達成するために、対象ユーザーがソリューションを使用できることを確実にすることです。

この機能領域で定義されている主な責務の 1 つは、ユーザビリティの調査です。これには、ユーザー要求の収集、分析、および優先順位付けが含まれます。ソリューション開発の早い段階および開発活動全体を通して、ユーザーを理解するために時間を投資することにより、プロジェクトでユーザーのニーズを効果的に満たすことができる可能性が大幅に高まります。

ユーザビリティの機能領域には、このほかにも、使用シナリオとユース ケースの開発という重要な責務があります。ここで重要なのは、1 歩下がってソリューションが全体としてどのように使用されるのかに注目することです。この作業は、ユーザーがソリューションに どのように接するのかを開発チームが概念的および実証的な観点から理解するのに役立ちます。そしてそれは多くの場合、設計の改善、ひいては効率の向上につながります。

ユーザビリティの機能領域で定義されている最後の主要な責務は、ソリューションに対するフィードバックと入力の実装です。開発者にユーザーのフィードバックを提供するための時間を開発サイクル全体を通じて割くことによって、ソリューションのユーザー満足度を向上させることができます。

グラフィック デザイン

グラフィック デザインの機能領域では、ソリューション内のグラフィック要素が適切に設計されるのを確実にすることが焦点になります。この機能領域の主要な責務は、ユーザー インターフェイスの設計を推進することです。これには、ユーザーがやり取りすることになるオブジェクト（およびそれらのオブジェクトに適用されるアクション）の設計や、インターフェイスの主要な画面の設計が含まれます。

リリースマネジメント役割群

リリースマネジメント役割群の目標は、スムーズな展開 と継続的な運用です。リリースマネジメントは、MSF チームにおいて運用を直接扱う役割です。この役割には、次のような責務の機能領域が含まれます。

- プロジェクト開発グループと運用グループの間の主要な代弁者として活動します。
- リリース活動のためのツールの選択を管理し、最適な自動化を推進します。
- 運用環境へのリリースに対する運用基準を設定します。
- 管理、サポート、および展開の観点から設計に関与します。
- 運用のトレーニングを推進します。
- パイロット展開を推進およびサポートします。
- 運用環境へのソリューションの展開を計画および管理します。
- 安定化の測度が受け入れ基準を満たすことを確実にします。

インフラストラクチャ

- エンタープライズ インフラストラクチャの計画を立てます。
- 複数の場所（データ センター、ラボ、出張所など）にわたる物理環境の使用および計画を調整します。
- 一貫したインフラストラクチャ管理と標準のための方針と手順をチームに提供します。
- MSF チームにインフラストラクチャ サービスを提供します（サーバーの構築、標準イメージ、ソフトウェアのインストールなど）。
- チームのハードウェアおよびソフトウェアの調達を管理します。
- 運用環境を正確に反映したテスト環境とステージング環境を構築します。

サポート

- IT ユーザーに一次窓口と顧客サービスを提供します。
- 顧客との SLA（サービス レベル アグリーメント）の管理およびコミットメントの達成を確実にすることによってビジネスを支援します。
- インシデントおよび問題の解決を提供します。ユーザーの要求や記録されたインシデントに迅速に対応します。
- 開発および設計チームにフィードバックを提供します。
- フェールオーバーや回復の手順を開発します。

運用

- アカウントとシステム設定の制御。ユーザーアカウントとアクセス許可を管理します。
- メッセージング、データベース、電気通信の運用。ネットワークの運用。
- システム管理、バッチ処理。
- ファイアウォール管理。セキュリティ管理。
- アプリケーション サービス。
- ホスト統合サービス。
- ディレクトリ サービスの運用。

商用リリースマネジメント

- 製品登録コード。登録確認プロセス。
- ライセンス管理。
- パッケージ化。
- 流通チャネルの管理。
- 印刷および電子的な刊行物。

インフラストラクチャ

インフラストラクチャの職能領域は、MSF プロジェクトで果たさなければならない運用インフラストラクチャ関連の一連の責務を表します。この職能領域は、MSF リリースマネジメント役割群の一部です。MOF を使用するプロジェクトでは、MOF のインフラストラクチャ役割群の責務に対応します。

サポート

この職能領域では、構築および展開されたソリューションが "サポート可能" であることを確実にすることが焦点になります。MOF を使用するプロジェクトでは、MOF のサポート役割群の責務に対応します。

運用

この職能領域は、MSF プロジェクトで果たさなければならない一連の運用の責務を表します。この職能領域では、構築および展開されたソリューションが運用可能であり、既に運用されている他のサービスと両立することを確実にすることが焦点になります。MOF を使用するプロジェクトでは、MOF のサポート役割群の責務に対応します。

商用リリースマネジメント

この職能領域は、商用ソフトウェア製品のリリースに関連する一連の責務を表します。商用リリースマネジメントでは、製品を流通経路に乗せることが焦点になります。

チーム モデルの規模調整

かつて Microsoft のソフトウェア開発者だった Steve McConnell は、著書『Rapid Development』の中で次のように述べています。

「大規模なプロジェクトでは、コミュニケーションを形式化および合理化するための組織的な慣行が必要になります。... コミュニケーションを合理化するための方法はすべてなんらかの階層の作成を必要とします。つまり、チームとして機能する小さなグループを作り、グループの代表者を指名して、互いの連絡および管理者とのやり取りをその代表者らに任せるのです。」

MSF チーム モデルでは、大規模なチーム（10 人より多くのメンバを含むチーム）を小規模の分野横断的な製品機能チームに分割することを推奨しています。これらの小規模なチームは、頻繁に互いの作業を同調させながら、並行して活動します。

さらに、特定の役割のニーズを満たすために複数のリソースが必要な場合には、職能チームを使用できます。職能チームは、必要なリソースに応じて役割の中でグループ分けされます。

製品機能チーム

チーム モデルの各役割群は、(できるだけフラットな)階層構造に組織された 1 人以上の要員で構成されます。たとえばテストは、テスト マネージャあるいはリーダーに報告を行います。

製品機能チームはこの構造の上に重ねられます。製品機能チームは小規模のサブチームであり、各役割の 1 人以上のメンバからなるマトリクス組織を構成します。これらのチームは特定の製品機能集合を割り当てられ、そのすべての側面（設計やスケジュールなど）に対する責務を負います。たとえば、印刷サービスの設計と開発に割り当てられる製品機能チームなどが考えられます。

Steve McConnell は、『Rapid Development』の中で次のように述べています。

「製品機能チームは、エンパワーメント、説明責任、およびバランスの面で有利です。製品機能チームには関係する各グループの代表者が含まれているため、的確にエンパワーされます。チームの決定においては必要な観点がすべて考慮されるため、決定がくつがえされることはほとんどありません。」

「同様の理由から、製品機能チームでは説明責任も明確になります。製品機能チームには、的確な判断を下すために必要なすべての人物に接触できます。このため、判断が良くなかった場合、その責任はすべてチーム自体にあります。製品機能チームはバランスの面でも優れています。製品の仕様について、開発、マーケティング、または品質保証のいずれかが単独で決定権を握るのは好ましくありません。これらの各カテゴリの代表者を含むグループなら、バランスの取れた決定を下すことができます。」

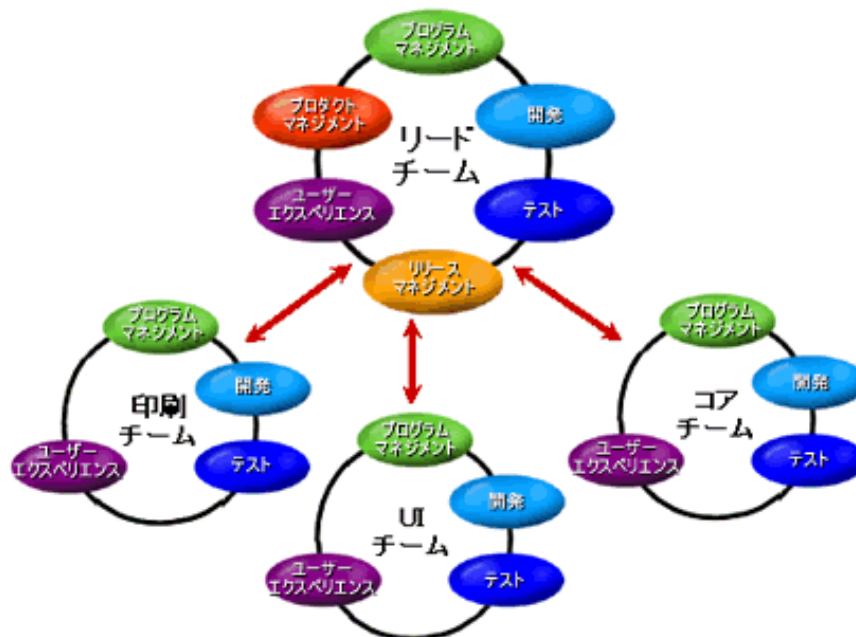


図 2: 製品機能チーム

メモ: この図は、製品機能チームの編成の要件を表しているわけではありません。たとえば、ユーザエクスペリエンスの役割はすべての製品機能チームに必要とは限りません。個々のソリューションの目標に応じたチームを編成する必要があります。

職能チーム

職能チームは、1つの役割の中に存在するチームです。チームやプロジェクトの規模が大きすぎるために、ある役割内のメンバをそれぞれの職能に基づいてグループ分けする必要性が生じた場合に使用されます。たとえば Microsoft では、製品開発チームに製品計画と製品マーケティングの担当が含まれているのが一般的です。これらの仕事は、どちらもプロダクトマネジメントの一側面です。一方では顧客が実際に望んでいる製品機能を実現することが、もう一方では製品のメリットを潜在的なユーザーに伝えることが焦点になります。

開発についても同様で、ユーザー、ビジネス、データなど、作業対象とするサービス層によって開発者をグループ分けできます。このほか、ソリューション開発者がコンポーネント開発者かによって開発者をグループ分けすることもよくあります。コンポーネント開発者はCなどの低水準言語の開発者であるのが一般的で、企業で活用できる再利用可能なコンポーネントを作成します。ソリューション開発者は、それらのコンポーネントを組み合わせるエンタープライズ アプリケーションを作成します。一般にソリューションの作成者は、Microsoft® Visual Basic® などの高水準言語を使用します。

職能チームでは、グループ内に階層構造が含まれる場合もよくあります。たとえば、多くのプログラム マネージャはプログラム マネージャのリーダーを介して報告を行い、リーダーがグループ プログラム マネージャへの報告を行います。このような構造は、役割群のレベルではなく職能領域においてみられることもあります。ここで重要なのは、この階層によってプロジェクト レベルのチーム モデルが妨げられることはないということです。役割の目標もプロジェクト チームに対する全体的な説明責任も、それによって変わることはありません。

役割の兼任

チーム モデルは 6 つの役割で構成されていますが、チームに少なくとも 6 人のメンバがいなければならないというわけではありません。また、1 つの役割に 1 人の人物が必要ということもありません。肝心なのは、すべてのチームで 6 つの目標が代表されていなければならないということです。典型的には、各役割に少なくとも 1 人の人物が割り当てられていれば、それぞれの役割の関心が見過ごされることはなくなります。しかし、そのようなやり方で各役割にメンバを割り当てるのがすべてのプロジェクトで得策とも限りません。チームのメンバが役割を兼任しなければならない場合もよくあります。

小規模なチームでは、チームのメンバの間で 6 つの役割を分担しなければなりません。役割の兼任には 2 つの原則があります。第 1 に、開発チームのメンバは役割を兼任してはなりません。開発者はプロジェクトの構築者であり、その本来の作業から注意をそがれてはなりません。開発チームに他の役割を割り当てても、そうした他の責務のためにスケジュールの遅れを招く恐れが増すだけです。

第 2 に、もともと利害が対立する役割どうしを組み合わせないようにします。たとえば、プロダクトマネジメントとプログラムマネジメントは利害が対立するため、この 2 つの役割は組み合わせないようにする必要があります。プロダクトマネジメントが顧客を満足させたいのに対して、プログラムマネジメントは納期と予算の達成を求めます。このため、この 2 つの役割を組み合わせた場合、顧客から変更の要求があった時に、顧客満足に十分な配慮がないまま変更が考慮されないか、プロジェクトへの影響を考慮せずにその変更が受け入れられる危険性があります。これらの役割に別々のチーム メンバを割り当てていれば、判断が一方に偏るおそれなくなります。このほかに、テストと開発の組み合わせについても同じことが言えます。

次の表は、役割の危険な組み合わせ（N = Not Recommended (避けるべき) または U = Unlikely (普通はしない)) と共働的な組み合わせ（P = Possible (可能)) を表しています。ただし、実際のチーム編成については常に言えることですが、役割の分担が成功するかどうかは実際のメンバしだいであり、個々のメンバの経験やスキルにかかっています。

	プロダクト マネジメント	プログラム マネジメント	開発	テスト	ユーザ エクスペリエンス	リリース マネジメント
プロダクト マネジメント		N	N	P	P	U
プログラム マネジメント	N		N	U	U	P
開発	N	N		N	N	N
テスト	P	U	N		P	P
ユーザ エクスペリエンス	P	U	N	P		U
リリース マネジメント	U	P	N	P	U	

P 可 能
(Possible)

U 普通はしない
(Unlikely)

N 避けるべき
(Not Recommended)

図 3: 小規模プロジェクトでの役割の組み合わせ

行と列の交差部分に "N" と記されている場合、その役割の組み合わせは利害が対立しますので、どうしても避けられない場合を除いては、避けるべきです。また止むを得ずこの組み合わせを兼任する場合には、リスクの最小化計画およびコンティンジェンシ計画によって関連するリスクに対処する必要があります。もちろん、各役割の目標の間には、程度の差こそあれ多少の対立が存在します。こうした対立がチーム モデルの原動力となっていますが、役割を組み合わせようとする場合にはそれが問題の種にもなります。とはいえ、役割を組み合わせるのは珍しいことではありません。チームで適切な組み合わせを選択し、関連するリスクを積極的に管理すれば、問題の発生を最小限に抑えることができます。

エスカレーションと説明責任

MSF チーム モデルと組織図の違い

MSF チーム モデルを適用する際によく問題になるのは、"責任者は誰か" ということです。組織図には、責任者は誰で、誰が誰に報告するのかが示されています。これに対して、MSF チーム モデルでは、プロジェクト チームの主要な役割と責務は示されていますが、人事管理の観点から見たチームの管理構造は定義されていません。多くの場合、プロジェクト チームには複数の異なる組織からのメンバが含まれており、管理上、別のマネージャに報告を行うメンバがいる場合もあります。

しかし、チームのコンセンサスが得られない問題が出てくることもあります。十分な努力をしても合意に至らなかった場合は、プロジェクトを前に進めるために、プログラムマネジメントの役割が主導権を握らなければならないこともあります。プログラムマネジメントの役割の第 1 の目標はプロジェクトの制約の中で提供することであり、時間もその制約の 1 つです。したがって、プログラムマネジメントの役割の目標でありチームの目標でもあるこの目標を達成するために、プログラムマネジメントの役割がプロジェクトを再び軌道に乗せるべく一時的にトップダウンの意思決定を行わなければならないこともあります。通常は複数の役割にわたってリーダーシップが共有されているわけですが、このような場合には、変化の必要性が理解され、プロジェクトの目標に到達するためのこの権威ある決定に対してチームからより強いレベルの賛同が得られます。問題が解決されて、チームがコンセンサスを回復できれば、すぐにリーダーシップの責務は共有された状態に戻ります。チームオブピアーズでは、こうした問題に対処しながらも、プロジェクト チームに対する非階層的なアプローチを維持できるだけの柔軟性や適応性があることが証明されています。

外部との連携における説明責任

チームが成功を収めるためには、外部のグループとのやり取り、コミュニケーション、および連携が必要です。その対象は、顧客やユーザーから他の開発チームまで広がります。顧客はほとんどの場合、ソリューションに対する明確な説明責任がチームの単一の接点に属するように求めます。これに対してチームオブピアーズでは、ソリューションを成功させるためにはチーム内での説明責任の分担が必要になります。しかし、説明責任と報告の構造（コミュニケーション計画に示されている）とを明確に区別することが重要です。そうしないと、顧客にも開発チームにも、その情報を提供する責任を負っているのはチームの誰なのかがわからなくなります。

次の図は、ビジネス面またはテクノロジー面の連携における一般的な責任の所在を示しています。主な推進役は、プログラムマネジメント、プロダクトマネジメント、ユーザーエクスペリエンス、およびリリースマネジメントです。これらの役割では内部と外部の両方が焦点になりますが、開発とテストでは内部のみが焦点になり、外部とのコミュニケーションからは遮断されます。

これは、開発者とテストは外の世界から隔離される必要があるという意味ではありません。MSF チームが目指している顧客志向の発想の実現のためには、顧客組織や実際のユーザーとの接触は、とりわけプロジェクト初期の発達段階においては、きわめて重要で有り得ます。ただし、そうしたコミュニケーションは正式なものにするべきではありません。というのも、プロジェクトの後半において開発チームやテスト チームがソリューションの納品に焦点を絞る際には、それらはおろそかにされることになるからです。

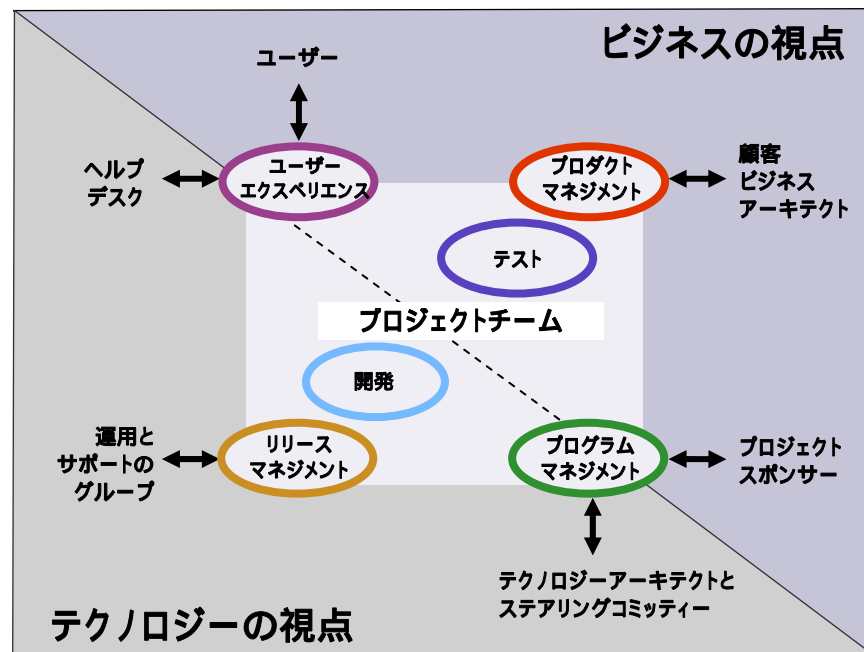


図 4: 説明責任

この図は高レベルの視点を表しています。通常チームは、品質保証、財務、法務など、さらに多くの外部グループと連携する必要があります。外部グループとの接点を明確にし、それが理解され、および開発とテストの役割が不必要に中断させられることなく効果的に作業できるように、それらの役割を外部から保護し続けることが重要です。

また、さまざまな役割を通じた外部との連携はチームに貴重な情報やアドバイスをもたらしますが、製品機能、スケジュール、リソースなどのプロジェクトのトレードオフの基本的な優先順位や個々の決定を変更する権限はチームの個々のメンバにも全体としてのチームにもないということを強調しておくことが重要です。それらの変更の権限はプロジェクトの顧客またはスポンサーにあり、プロジェクト チームはそれに基づいて変更を実施します。ここからも、チームオブピアーズすなわち対等なパートナーからなるチームが組織の権威、階層、構造とどのように異なり、またそれらに整合するものであることがわかります。

まとめ

MSF チーム モデルは、プロジェクトの成功を保証するものではありません。プロジェクトが成功するかどうかは、チームの構造だけでなく、さまざまな要因によって決まります。しかし、チームの構造が重要なことに変わりはありません。

『Rapid Development』の中で Steve McConnell は、この点を次のように説明しています。

「スキルと熱意と勤勉さを兼ね備えたメンバを集めても、チームの構造に問題があれば、メンバの努力を成功へと導くどころかかえって彼らの足を引っ張ることになります。その結果、開発期間は長くなり、品質は低下し、メンバのやる気は損なわれ、離職が増え、挙句の果てにプロジェクトが取りやめになることもあります。」

MSF チーム モデルは、まさにこの点に対処するためのものです。適切なチームの構造は成功のための基本です。MSF チーム モデルを実装し、その基となる原則を使用することによって、チームをより効果的に成功に導くことができます。

詳細については、以下のリンク先を参照してください。

Microsoft Solutions Framework:

<http://www.microsoft.com/japan/msdn/vstudio/productinfo/enterprise/msf/>

Microsoft Operations Framework:

<http://www.microsoft.com/japan/technet/itsolutions/techguide/mof/>

付記

¹ 設計へのチーム参加の最適化の詳細については、『Dynamics of Software Development』(Jim McCarthy 著、Microsoft Press 発行、1995 年) を参照してください。