

## Scrum.PSM-III.v2026-06-18.q13

試験コード:	PSM-III
試験名称:	Professional Scrum Master level III (PSM III)
認定資格:	Scrum
無料問題数:	13
バージョン:	v2026-06-18
アクセス数:	112
ページビュー数:	130
<a href="https://www.jpnpdf.com/Scrum.PSM-III.v2026-06-18.q13-mondaishu.html">https://www.jpnpdf.com/Scrum.PSM-III.v2026-06-18.q13-mondaishu.html</a>	

### 最新問題: 1

プロダクト バックログを整理する際にプロダクト オーナーが考慮すべき変数は何ですか？

#### Answer:

プロダクトバックログの順序付けはプロダクトオーナーの重要な責任であり、経験主義を通じて価値を最大化するために不可欠です。順序付けは、単一の優先順位付けルールではなく、複数の変数の継続的な検査を反映しています。

#### 1. 価値と成果

主要な変数は価値です。プロダクトオーナーは次の点を考慮します。

- \* 顧客とユーザーの価値、
- \* ビジネスへの影響と成果
- \* 製品目標との整合性。

より高い価値やより緊急性の高い価値を提供するアイテムは、通常、上位に順位付けされます。

#### 2. リスクと不確実性

リスクや不確実性を軽減するアイテムは、多くの場合、早めに注文されます。これには以下が含まれます。

- \* 技術的リスク、
- \* 市場リスクまたはユーザビリティリスク、
- \* 統合または依存のリスク。

早期学習により、より適切な意思決定が可能になり、長期的なコストが削減されます。

#### 3. 依存関係

プロダクトオーナーは、バックログアイテムとチーム間の依存関係を考慮します。他の作業を妨げない、または依存関係を減らすアイテムは、フローを改善し、調整のオーバーヘッドを削減するために、優先順位を高く設定する場合があります。

#### 4. 労力、複雑さ、実現可能性

開発者が労力を見積もる一方で、プロダクトオーナーはこの情報を用いて、価値とコスト、複雑さ、実現可能性のバランスを取ります。短期的な制約内で実現可能な高価値項目が優先されることが多いです。

#### 5. フィードバックと学習

順序付けは、スプリントレビュー、ユーザーテスト、そして市場の反応からのフィードバックを反映します。以前のインクリメントから得られた知見に基づいて、項目は上下に移動する場合があります。

#### 6. 時間的敏感性と機会費用

いくつかの項目は、次の理由により時間的に重要です：

- \* 規制期限、
- \* 市場の窓、
- \* 競争圧力。

こうした品目を遅らせると、その価値が下がったり、なくなる可能性があります。

#### 最新問題: 2

機能チームとコンポーネントチームの違い、および Scrum ガイドの観点から見た場合にそれらがどのように機能するかについて説明します。

#### Answer:

スクラムでは、チーム構造が価値提供能力に大きな影響を与えます。よく議論される2つの構造は、コンポーネントチームとフィーチャーチームです。スクラムガイドではこれらの用語を明確に定義していませんが、スクラムチームの定義において、フィーチャーチームの特徴を強く支持しています。

コンポーネントチームは、データベース、フロントエンド、ミドルウェアなどの専門分野やシステムコンポーネントを中心に編成されます。これらのチームの業務は通常、製品機能への部分的な貢献であり、顧客価値を提供するためには複数のチーム間での調整と引き継ぎが必要になります。その結果、コンポーネントチームは依存関係を導入し、統合を遅延させ、スプリント内で独立して使用可能なインクリメントを作成することに苦労することがよくあります。

一方、フィーチャーチームは、完全な製品機能、つまりプロダクトバックログアイテムの提供を中心に編成されます。フィーチャーチームはクロスファンクショナルであり、設計、構築、テスト、そして「完了」した価値の増分の提供に必要なあらゆるスキルを備えています。フィーチャーチームは依存関係を最小限に抑え、各スプリントで顧客向けの機能を独立して提供できます。

Scrum Guide の観点から見ると、機能チームは Scrum の原則にさらに近づきます。

\* スクラム ガイドでは、スクラム チームは機能横断的であり、機能チームを直接サポートし、コンポーネントチーム構造に挑戦すると述べられています。

\* スクラムでは、各スプリントで使用可能なインクリメントを作成することが求められます。フィーチャーチームはこの期待に応えることができますが、コンポーネントチームは通常、他のチームに依存しなければ実現できません。

\* スクラムは経験主義（透明性検査、適応）に基づいています。フィーチャーチームにおける依存関係を減らすことで透明性が向上し、検査と適応が迅速化されます。

\* スクラムは価値の提供と説明責任を重視します。機能チームは成果に対する明確な所有権を維持しますが、コンポーネントチームは技術サイロ間で説明責任を分散させます。コンポーネントチームは、従来の組織構造や技術的な制約により存在する場合がありますが、理想的なスクラム実装というよりは、組織的な障害となるものです。プロフェッショナル・スクラムマスターIIIの観点から見ると、フィーチャーチームへの移行は、アジリティの向上、価値提供の向上、そしてスクラムガイドで定義されているスクラムの実現性向上につながります。

### 最新問題: 3

スクラムマスター仲間があなたの意見を求めてきました。彼のチームメンバーはスプリントゴールの設定に何の価値も感じておらず、スクラムマスター自身もその意義をうまく説明できていません。あなたならこのスクラムマスターに何を伝えますか？

#### Answer:

チームメンバーがスプリントゴールの設定に価値を感じていない場合、それはスクラムに対する根本的な誤解を示しています。スクラムマスターとして、私は同僚のスクラムマスターに、スプリントゴールはスクラムの中核要素であり、アラインメント、コミットメント、そして経験主義に不可欠であることを説明します。

まず、スプリントゴールは、スクラムチームがスプリントで作業を行う理由を説明します。スクラムガイドによると、スプリントゴールはスプリントの唯一の目標であり、スプリントバックログとの一貫性をもたらします。明確な「なぜ」がなければ、スプリントの作業は価値を提供するための目的のある取り組みではなく、無関係なタスクの集まりとなってしまいます。スプリントゴールは、チームが選択されたプロダクトバックログアイテムの背後にある意図を理解し、日々の意思決定をその意図に沿って行うのに役立ちます。

第二に、スプリントゴールはスクラムチームのコミットメントを表します。チームは、具体的なスコープが変更される可能性があったとしても、スプリントゴールの達成に向けて全力を尽くすことを約束します。このコミットメントは、集中力と責任の共有を促進します。チームは個々のタスクを最適化するのではなく、スプリントゴール全体の達成に向けて最適化します。

第三に、スプリントゴールは柔軟性を制限するのではなく、むしろ柔軟性を生み出すものです。スプリント中に新たな発見、リスク、あるいは機会が発生した場合、チームはスプリントゴールを損なわない限り、スプリントバックログを調整することができます。これにより、チームは目的の安定性を維持しながら変化に対応できます。

スプリント目標がないと、変更は恣意的になり、焦点を失うリスクが高まります。

4つ目に、スプリントゴールは効果的な検査と適応を可能にします。デイリースクラムでは、チームはスプリントゴールに向けた進捗状況を検査し、それに応じて計画を調整します。同様に、スプリントレビューでは、関係者がスプリントゴールが達成されたかどうかを

検査できます。スプリントゴールがなければ、検査のための意味のあるベンチマークは存在しません。

最後に、スプリント目標がなければ、スクラムは意図したとおりに実践されていないということを明確にすることが重要です。

スプリントゴールはスクラムの必須要素であり、これを削除すると透明性が損なわれ、フレームワークの実証的基盤が弱まります。

#### 最新問題: 4

あなたはスクラムチームでスクラムマスターとして働いています。開発チームは常に要件が明確でないと不満を漏らしています。プロダクトオーナーは、要件を明確に説明する時間があまりないと主張しています。どうすればよいのでしょうか？

#### Answer:

この状況は、プロダクトバックログの透明性とコラボレーションの崩壊を示しており、経験主義と価値提供を直接的に脅かしています。スクラムマスターとしての私の責任は、問題を自ら解決することではなく、スクラムチームと組織が問題を解決できるように支援することです。

#### 1. 問題を再定義する: 要件と製品バックログ

まず、両者が問題を再考できるよう支援します。スクラムでは、従来の固定的な意味での「要件」を扱うのではなく、創発的で、順序付けられ、継続的に改良されるプロダクトバックログを扱います。プロダクトバックログアイテムの明確性が欠如していると、バックログが使用可能な状態になく、開発者にとって障害となります。

#### 2. 影響を透明化する

次に、不明確なバックログ項目の影響を透明化するための会話を促進します。

- \* 開発者は作業を確実に予測できないため、
- \* スプリント目標が危険にさらされる、
- \* やり直しや無駄の増加、
- \* 価値の提供が遅くなります。

この会話にはプロダクトオーナーが参加し、非難ではなく、根拠に基づいた証拠を提示する必要があります。目標は、責任の所在を明確にすることではなく、結果についての共通理解を得ることです。

#### 3. プロダクトオーナーの責任を強化する

スクラムガイドでは、プロダクトオーナーは価値の最大化とプロダクトバックログ管理の責任を負うことが明確に示されています。これには、プロダクトバックログアイテムが明確で、理解され、整理されていることを確認することも含まれます。「忙しすぎる」からといって、この責任が免除されるわけではありません。スクラムマスターとして、私はプロダクトオーナーに対し、対応能力の不足自体が組織にとっての障害となることを認識するよう指導します。

#### 4. ハンドオフではなくコラボレーションを可能にする

同時に、開発者には、明確さは単に提供されるものではなく、多くの場合、共同で作り上げていくものであることを指導します。スクラムは、開発者とプロダクトオーナーの緊密な連携を推奨します。例えば、以下のような手法が挙げられます。

- \* 定期的な製品バックログの改良、
- \* スプリント計画中の共同討論、
- \* スプリントの目標に関して焦点を絞った質問をすることで、詳細な事前仕様に頼ることなく、共通理解を大幅に向上させることができます。

#### 5. 組織の制約に対処する

プロダクトオーナーの対応能力の欠如が組織の過負荷や競合する責任による場合、これはシステム的な障害となります。その場合、スクラムマスターはこの問題を組織に提起し、対応能力が十分でないプロダクトオーナーが製品の成果を危険にさらしていることを経営陣に理解させる必要があります。

#### 最新問題: 5

スクラムチームの開発者から障害が報告されました。次のスプリントで予定されている作業には、チームメンバーが持ち合わせていない特定の知識と専門知識が含まれています。この障害にはどのように対処しますか？

#### Answer:

開発者が特定の知識や専門知識の不足を障害として挙げた場合、スクラム マスターは、スクラムの原則、特にクロスファンクショナル、経験主義、自己管理を強化しながら、価値の提供もサポートする方法で状況に対処する必要があります。

まず、これが本当に障害であるかどうかを確認することが重要です。スクラムでは、障害とはチームが自力で解決できないもののことです。スクラムマスターとして、私は開発者、そして必要に応じてプロダクトオーナーとの議論を促し、望ましい結果を達成するために専門知識が本当に必要かどうかを検証します。場合によっては、スコープやアプローチを調整したり、プロダクトバックログアイテムを改良して代替ソリューションを実行可能にしたりすることも可能です。この会話を通して、専門知識の必要性が当初想定されていたほど重要ではないことが明らかになるかもしれません。

第二に、専門知識が本当に必要な場合、スクラムマスターはチームが機能横断的なスクラムチームとして問題に取り組むよう促すべきです。スクラムでは、チームが価値を提供するために必要なすべてのスキルを保有、あるいは習得することが求められます。したがって、開発者には、必要な知識を自らどのように習得または獲得できるかを尋ねます。学習、調査、トレーニング、実験、プロトタイプ構築のための時間を確保することが選択肢として挙げられます。これらの活動はスプリントバックログの一部として計画することができ、長期的なチーム能力の向上に役立ちます。

第三に、スクラムマスターは、チームの自己管理を損なうことなく、外部の専門知識を効果的に活用できるよう支援します。スプリントの計画やリファインメントの過程で、チームは作業とスプリントバックログの所有権を維持しながら、洞察を得たり、アプローチを検証したり、不確実性を軽減したりするために、社内外の専門家に相談することができます。

最後に、これらの選択肢のいずれでも障害が解決しない場合、スクラムマスターは組織がスクラムチームを支援できるよう支援する責任を負います。これには、組織内の他の部署、あるいは必要に応じて組織外からの専門知識へのアクセスを促進することが含まれる場合があります。スクラムマスターは問題を自ら解決するのではなく、チームが前進できるよう組織的な障壁を取り除くよう尽力します。

#### 最新問題: 6

「技術的負債は開発チームの唯一の懸念事項です」。スクラムマスターとして、あなたはこの意見に同意しますか？あるいは、なぜ同意しないのでしょうか？

#### Answer:

スクラムマスターとして、技術的負債は開発チームだけの懸念事項であるという意見には同意できません。開発者は技術的負債を認識し理解する責任がありますが、その影響はチームをはるかに超えて、製品レベルおよび組織レベルのアジリティ、品質、そしてデリバリーにまで及びます。

まず、技術的負債はチームのアジャイル性維持能力に直接影響を及ぼします。技術的負債が蓄積するにつれて、製品の変更に必要なコストと労力が増加します。その結果、開発が遅延し、予測可能性が低下し、最終的には、妥当な期間内に動作するソフトウェアを提供することが困難、あるいは不可能になります。アジリティが低下すると、開発チームだけでなく、組織全体が損害を被ります。

第二に、技術的負債は製品の品質とデリバリーに重大な影響を及ぼします。技術的負債のレベルが高いと、多くの場合、欠陥、不安定性、そして統合の問題につながります。これは、スプリントごとに「完了」インクリメントを提供するというスクラムの原則を損ないます。製品が確実にデリバリーまたは検査されない場合、顧客と利害関係者は直接的な影響を受け、技術的負債は共通の懸念事項となります。

第三に、技術的負債の発生を特定するには開発者が最適な立場にありますが、その解決にはスクラムチーム全体の連携が不可欠です。プロダクトオーナーは、スプリントにおけるすべての作業が必ずしも新しい機能を生み出すわけではないことを理解する必要があります。技術的負債の削減への投資は、将来の価値、持続可能性、そしてデリバリー能力への投資です。ステークホルダーも、このトレードオフについて透明性を確保する必要があります。

4つ目に、スクラムでは技術的負債を無期限に先送りするのではなく、可視化し、継続的に対処することを推奨しています。これには、技術的負債関連の作業をプロダクトバックログに追加し、機能的な作業と並んで優先順位を付けるといったことが含まれます。技術的負債を「目に見えない」もの、あるいは純粋に技術的なものとして扱うことは、経験主義と長期的な価値創造を損ないます。

#### 最新問題: 7

Scrum と経験主義 / 経験的プロセス制御の関係は何ですか？

#### Answer:

スクラムは基本的に経験主義（経験的プロセス制御とも呼ばれる）に基づいています。これは、ソフトウェア開発のような複雑な作業は事前に完全に理解したり予測したりすることはできないとスクラムが認識していることを意味します。その代わりに、経験、観察、そして証拠に基づいて意思決定が行われ、継続的な閉じたフィードバックループが形成されます。

経験的プロセス制御は、透明性、検査、適応という3つの柱に基づいています。スクラムは、これらの柱それぞれを明確にサポートし、強化する役割、イベント、成果物からなる構造化されたフレームワークを提供します。

#### 透明性

透明性は、プロセスと製品のすべての重要な側面が、結果の責任者にとって可視化されることを保証します。スクラムでは、プロダクトバックログ、スプリントバックログ、プロダクトインクリメントといった明確に定義された成果物を通して透明性が確保され、それぞれが共通の完了定義によって管理されます。スクラムイベントは、進捗状況、課題、現在の状態を共有する機会を定期的に設けることで、透明性をさらに高めます。

透明性がなければ、検査は誤解を招き、効果がありません。

#### 検査

Scrumでは、製品とプロセスの両方を頻繁かつ定期的に検査することが規定されています。各Scrumイベントは検査ポイントとして機能します。

- \* デイリースクラムはスプリント目標の進捗状況を検査し、
- \* スプリントレビューは増分を検査し、製品バックログを適応させます。
- \* スプリントレトロスペクティブでは、チームの作業方法を検査します。

これらの検査は、タイムボックス化され軽量化されているため、過剰なオーバーヘッドを回避しながらもタイムリーなフィードバックが可能になります。

#### 適応

検査は、適応につながる場合にのみ意味を持ちます。スクラムは、学習内容に基づいて計画、プロセス、バックログの内容を変更できるようにすることで、適応を明確に実現します。スプリントバックログはスプリント中に適応され、プロダクトバックログはスプリントレビュー後に適応され、チームプラクティスはスプリントレトロスペクティブ後に適応されます。

#### クローズドフィードバックループ

透明性、検査、適応は、閉じたフィードバックループを形成します。スクラムの短いイテレーション（スプリント）は、学習を頻繁に行うことを保証し、スクラムチームと関係者が変化に迅速に対応し、リスクを軽減し、時間の経過とともに成果を向上させることを可能にします。

#### 最新問題: 8

開発チームは自己組織化されていると主張し、デイリースクラムはもはや必要ないと述べています。チームは1日を通して共同作業を行っており、デイリースクラムは不要な儀式になっていると感じています。

## Answer:

開発チームが自己組織化を理由にデイリースクラムを中止するのは、自己管理とスクラムイベントの目的の両方を誤解していることを示しています。スクラムマスターとして、私は強制ではなく、指導、コーチング、そして経験主義を通してこの問題に対処します。

スクラムではデイリースクラムが必須

まず、デイリースクラムはスクラムの必須イベントであることを明確にする必要があります。スクラムガイドでは、デイリースクラムを次のように定義しています。

スプリント期間中、開発者向けに毎日開催される15分間のイベント。このイベントを廃止することは、たとえ非公式な場でどれだけ良好なコラボレーションが行われているとしても、チームがスクラムを実践していないことを意味します。

自己組織化は経験主義を無視することを意味しない

自己組織化（自己管理チームは、作業の進め方を決定するものであり、検査や適応を行うかどうかは決定しません。スクラムイベントは、経験的プロセス管理を維持するために存在します。デイリースクラムは具体的に以下のことを可能にします。

- \* スプリント目標に向けた進捗状況の透明性
- \* スプリントバックログと現在の計画の検査、
- \* 今後 24 時間の仕事への適応。

一日を通しての非公式なコラボレーションは、デイリースクラムが提供する共有された意図的な検査の瞬間に取って代わるものではありません。

デイリースクラムは儀式やステータスミーティングではない

デイリースクラムが無駄な儀式のように感じられるなら、それは正しく活用されていないサインです。デイリースクラムは、スクラムマスターやプロダクトオーナーのためのステータス報告やミーティングではなく、開発者のための計画イベントであり、スプリントゴールをいかにして達成するかに焦点を当てるべきです。

スクラム マスターとして、私はチームを指導してデイリースクラムを改善します。たとえば、次のようになります。

- \* スプリント目標に向けた進捗状況を中心に議論し、
- \* 障害とリスクを明確にし、
- \* チームの状況に適したさまざまな形式を使用する。

デイリースクラムの廃止に伴うリスク

デイリースクラムを廃止すると透明性が低下し、検査と適応が遅れます。統合の問題、不整合、スプリント目標への脅威といった問題が、手遅れになって表面化してしまう可能性があります。リスクと無駄が増大します。

時間が経つにつれて、予測可能性と価値の提供が損なわれます。

## 最新問題: 9

開発チームのメンバーが非倫理的な行動をとった例にはどのようなものがありますか？

## Answer:

スクラムにおける開発チームメンバーの非倫理的な行動の例としては、ユーザー、ステークホルダー、あるいは組織に潜在的な悪影響を与えることを認識しながら、低品質または

安全でないソフトウェアを故意に納品することが挙げられます。このような行動は、スクラムに埋め込まれた倫理的期待に反し、スクラムの複数の価値観に違反します。

例えば、開発者は、作業を早く終わらせるため、あるいは生産性を高く見せるために、既知の欠陥、セキュリティ上の脆弱性、あるいは技術的負債を意図的に無視することがあります。安全でない、あるいは不安定であることが分かっているソフトウェアをリリースすることは、エンドユーザーを危険にさらし、製品の真の状態を誤って伝えることとなります。これは、困難な問題への対処や懸念の提起を避けるため、品質へのコミットメントと勇気を損ないます。

もう一つの非倫理的な例は、スクラムチームやステークホルダーから重要な情報を隠蔽することです。これには、リスクを隠蔽したり、障害を軽視したり、進捗状況や課題について透明性を欠いたりすることが含まれます。

このような行為はオープン性を侵害し、経験主義と効果的なコラボレーションに不可欠な信頼を損ないます。

非倫理的な行動は、チームメンバーのサポートを怠ることに表れることがあります。例えば、他者への支援を拒否したり、同僚の意見を無視したり軽視したり、チームの結束を損なうような働き方をすることは、スクラムの価値観である「尊重」に反します。スクラムでは、スプリントゴールの達成に向けて、チームメンバーが協力し、互いにサポートし合うことが求められます。

最後に、スクラムチームによる合意に反すること、例えば完了定義や合意された作業契約を無視することは非倫理的です。これは説明責任を損ない、作業の品質と完全性についてステークホルダーに誤解を与える可能性があります。

#### 最新問題: 10

プロダクトオーナーは依然として距離を置いています。スプリントに必要なプロダクトバックログを引き渡しているにもかかわらず、スプリント期間中は開発チームと連携していません。スクラムマスターにとって価値のある行動は何でしょうか？

#### Answer:

プロダクトオーナーが遠ざかることは、価値提供、透明性、そして経験主義にとってリスクとなります。プロダクトオーナーはスプリントのプロダクトバックログを提供していますが、スプリント中のコラボレーション不足は学習と情報に基づいた意思決定を阻害します。スクラムマスターとして、プロダクトオーナーの代わりに務めるのではなく、コーチング、コラボレーションの促進、そしてシステム上の障害への対処に重点を置くべきです。

#### 1. 影響を透明化する

スクラムマスターは、プロダクトオーナーの不在の影響を可視化できるようにする必要があります。

- \* 製品バックログ項目を明確にする能力の低下、
- \* 発見があった場合の意思決定が遅くなる、
- \* スプリントの目標と製品価値に対するリスクが増加します。

この透明性は、プロダクトオーナーとの丁寧な会話を通じて、また必要に応じてスプリントレトロスペクティブなどのスクラムイベントを通じて確立される必要があります。

## 2. プロダクトオーナーに説明責任について指導する

スクラムガイドでは、プロダクトオーナーは価値の最大化とプロダクトバックログ管理の責任を負い、開発者との継続的な連携が不可欠であるとされています。スクラムマスターは、スプリントプランニングでバックログを引き渡すだけでは不十分であり、スプリント中の対応が実証性にとって不可欠であることをプロダクトオーナーに理解させるよう指導する必要があります。

## 3. プロダクトオーナーを交代させることなく、より良いコラボレーションを実現する

スクラムマスターは、次のようなコラボレーションの機会の創出を支援する必要があります。

- \* スプリント中に定期的に明確化の機会を設ける。

- \* 製品バックログの改善により、未回答の質問が少なくなり、

- \* 限られたプロダクトオーナーの可用性を有効に活用するために、開発者が焦点を絞った質問を準備するのを支援します。

ただし、スクラムマスターがプロダクトオーナーの責任を引き継ぐことはできません。そうすると、説明責任が曖昧になってしまうからです。

## 4. 組織的な原因に対処する

プロダクトオーナーの不在が、作業負荷、役割の混乱、あるいは組織的なプレッシャーによるものである場合、それは組織にとっての障害となります。スクラムマスターは、この問題をリーダーシップに提起し、プロダクトオーナーが不在になることが製品の成果に及ぼすリスクを組織が理解できるように支援する必要があります。

### 最新問題: 11

人事部の担当者から連絡がありました。担当チームのプロダクトオーナーが本日をもって不在となり、このスプリントの残りの期間に対応できないとのこと。次のスプリントには復帰するかもしれませんが、現時点では未定です。スクラムチームはどう対応すべきでしょうか？

#### Answer:

プロダクトオーナーが予期せず利用できなくなった場合、スクラムチームは、スクラムの責任を尊重しながら、継続性、透明性、価値提供を維持する方法で対応する必要があります。

#### 短期的な対応

短期的には、スクラムチームは現在のスプリントと、場合によっては次のスプリントまで作業を継続できる必要があります。スクラムは短期的な中断に対して耐性を持つように設計されています。チームは以下のことに頼って作業を進めることができます。

- \* プロダクトオーナーが以前に伝えたプロダクトビジョン

- \* プロダクトバックログの現在の状態と順序。これは、プロダクトオーナーの価値決定をすでに反映しているはずで

この期間中、開発者はスプリント目標に向けて作業を継続し、スクラムマスターはスクラムイベントの実施と生産性の維持を確保します。説明責任を軽視するため、非公式にプロダクトオーナーの役割を担うべきではありません。

#### 長期的な影響

プロダクトオーナーの不在が短期間を超えると、スクラムチームの妨げになります。プロダクトオーナーは、製品価値を最大化し、プロダクトバックログを管理する責任を負います。

不在が長引くと、バックログの効率的な順序付け、関係者の協力、価値に基づく意思決定が妨げられます。

この場合、スクラムマスターは組織に対して障害を明確に示す必要があります。これには、価値提供への影響を説明し、明確なプロダクトオーナーの責任の必要性を経営陣に理解してもらうことが含まれます。その後、組織は意思決定と責任の継続性を確保するために、新しいプロダクトオーナーを任命する必要があります。

#### 最新問題: 12

技術システムは、大規模なものから小規模なものまで、複合要素に分解できます。基本的な構成要素は、アクティビティ、ワークフロー、機能、特徴、能力、その他類似の用語で表現されます。

このシステム分解は、大規模プロジェクトのスクラムチームにどのような影響を与えますか？

#### Answer:

技術システムは、複雑さを管理するために、アクティビティ、ワークフロー、機能、フィーチャ、コンポーネントなどのより小さな要素に分解されることがよくあります。分解は大規模なシステムを理解し構築するために不可欠ですが、スクラムチーム、特に小規模な環境においては大きな意味を持ちます。

##### 1. コンポーネント中心のチーム構造のリスク

システムの分解がチーム構造を左右する場合、組織は技術レイヤーや機能ごとにコンポーネントチームや専門チームを編成することがよくあります。スケールドスクラムでは、これにより以下のことが可能になります。

- \* チーム間の依存関係、
- \* 調整オーバーヘッド、
- \* 統合リスク。

このような構造では、チームが各スプリントでエンドツーエンドの統合された増分を提供することが難しくなり、経験主義が弱まり、フィードバックが遅れます。

##### 2. 価値提供と検査への影響

スクラムは、稼働中の製品インクリメントを頻繁に検査することに依存しています。作業が狭く定義された技術コンポーネントに分解されると、個々のチームは有用な価値ではなく、部分的な成果物しか提供できなくなる可能性があります。これにより透明性が低下し、特に複数のチームが関与している場合、製品レベルでの有意義な検査が困難になります。

### 3. 特徴指向分解の優先

スクラムでは、作業を水平方向の技術レイヤーではなく、垂直方向の価値指向のスライス（機能または能力）に分解することを推奨します。これにより、各スクラムチームは以下のことが可能になります。

- \* 部門横断的、
- \* 使用可能な増分を独立して提供できる、
- \* 他のチームへの依存度が低くなります。

大規模なプロジェクトでは、機能指向の分解によって依存関係が削減され、フローが改善されます。

### 4. 統合と経験主義への影響

不適切な分解は統合コストを増大させ、多くの場合、統合の遅延や不定期化につながります。スクラムでは、統合されていない作業は「完了」ではないため、早期かつ頻繁な統合が求められます。スケールドスクラムでは、分解の選択が統合を継続するか延期するかに直接影響し、リスク管理に大きな影響を与えます。

### 5. 組織と学習への影響

システムの分解は学習と適応性にも影響を与えます。チームが個別のコンポーネントではなく完全な機能を所有することで、以下の点をより深く理解できるようになります。

- \* 顧客のニーズ、
- \* システムの動作、
- \* 製品全体にわたるトレードオフ。

この幅広い理解により意思決定が改善され、システム全体の継続的な改善がサポートされます。

### 最新問題: 13

スクラムチームのスプリントは1ヶ月です。開発チームは、この期間が長すぎるため、デイリースクラムは少々長すぎると主張しています。代わりに、週1回の進捗報告ミーティングを開催したいと考えています。これについて、どう思われますか？

#### Answer:

スクラム マスターの観点から見ると、デイリー スクラムを週次更新ミーティングに置き換えることはスクラムに準拠しておらず、スプリントの長さに関係なく、チームが効果的に検査して適応する能力が大幅に弱まります。

まず、スクラムはデイリースクラムを必須イベントとして明確に定義しています。スクラムガイドでは、デイリースクラムはスプリント期間中、開発者向けに毎日15分間のイベントとして実施されるとされています。スプリントの長さ（週間でも1ヶ月でも）によって、このイベントの目的や必要性が変わることはありません。したがって、デイリースクラムを実施しないことを選択した場合、チームはもはやスクラムを実践しているのではなく、スクラムに類似したプロセスを実行していることになります。

第二に、デイリースクラムはステータスミーティングではありません。その主な目的は、開発者がスプリントゴールに向けた進捗状況を検査し、作業を同期させ、必要に応じてスプ

リントバックログを調整することです。週次ミーティングでは、検査と調整の頻度が大幅に減少し、統合の問題、不整合、スプリントゴールへのリスクといった問題の発見が遅れてしまいます。

第三に、デイリースクラムの廃止は、スクラムの経験主義の3つの柱の一つである透明性に悪影響を及ぼします。毎日の同期がなければ、進捗、障害、発見に関する重要な情報が古くなったり、隠されたりしてしまいます。この透明性の低下は、作業が合意された標準から逸脱したり、適切に統合されなかったり、スプリント終了までにスプリント目標をサポートしなくなったりする可能性を高めます。

4つ目に、1ヶ月のスプリントであれば検査頻度を低く抑えられるという主張は、経験主義の誤解を反映しています。スプリントが長くなるとリスクが増大するため、頻繁な検査と適応の重要性は低下するどころか、むしろ高まります。デイリースクラムは、チームを再編成し、新たな問題に早期に対応するための定期的な機会を提供し、無駄や手戻りを削減します。

最後に、スクラムマスターとしての私の役割は、スクラムチームにスクラムイベントの目的と価値を教え、指導することです。デイリースクラムを廃止するのではなく、開発者がデイリースクラムをどのように活用するかを改善できるよう支援します。例えば、報告会ではなく、スプリントゴールに向けた進捗と今後24時間の実行可能な計画に焦点を当てるようにします。

**Valid PSM-III Dumps** shared by GoShiken.com for Helping Passing PSM-III Exam!

GoShiken.com now offer the **newest PSM-III exam dumps**, the GoShiken.com PSM-III exam questions have been updated and answers have been corrected get the **newest** GoShiken.com PSM-III dumps with Test Engine here:

<https://www.goshiken.com/Scrum/PSM-III-mondaishu.html> (36 Q&As Dumps, **30%OFF**

**Special Discount: Freepdfdumps**)