

アトラシアンによる

はじめてのITSM クイックスタートガイド

初心者にも分かりやすい図版付き



 **ATLASSIAN**

CONTENTS

第1章	アトラシアンによるITSM 概要	5
	基本ユースケース	
第2章	サービスリクエスト管理	15
	応用例	
第3章	インシデント管理	33
	応用例	
第4章	変更実現	51
	応用例	
第5章	アセット管理	67
	応用例	
第6章	高度なカスタマイズ	79

アトラシアンによる

はじめてのITSM クイックスタートガイド

初心者にも分かりやすい図版付き

はじめに

クラウドやSaaSが身近なものになってきた昨今、サービスという考え方がますます重要性を帯びてきました。モノを作りそれをお客様 = 消費者に提供して終わるというサイクルではなく、サービスは、その価値を享受するお客様 = ユーザーに「継続して使い続けてもらう」ことが肝要になります。

あらゆる業界の企業がソフトウェアを活用したサービスの提供を取り組み始めた今、サービスを途切れなく継続的に提供し、ユーザーの満足度を高めていくにはどうすればよいか、そのような疑問を抱えているプロダクトマネージャーやIT部門のリーダーは増えていることでしょう。

そのようなときにITサービス管理 (IT Service Management: ITSM) が、大きな助けになるかもしれません。

全6章からなる本書では、上述のような問題意識を中心に、アトラシアン製品を駆使してどのようにITSMを実践していくかを、お届けします。

本書の内容につきましては、2022年5月時点の内容に基づいています (一部でその後の機能名変更を反映済み)。ご紹介している製品の画面や機能が変化する可能性がある点は予めご了承ください。



第 **1** 章

アトラシアンによる ITSM概要



第1章 アトラシアンによる ITSM 概要

本書の第1章では、アトラシアンによるITSM概要として、ITSMやITILの概念と関係性、そしてアトラシアン製品と各ITSMプラクティスの全体像についてご案内します。

ITサービスの提供という難題に取り組むにあたって、ITSMおよびその実践的なライブラリ集であるITILの理解は大きな助けになります。ITSMのプラクティスは多岐に渡りますが、そのすべてを同時に実現しなければならない、というわけではありません。

今できるところからはじめるというスタンスで、まずは本章で紹介する5つの主要なプラクティスに取り組んでみてください。以降の章では、それぞれのプラクティスについての詳しい設定や実践の方法をご案内します。

ITSMとITIL

ITSM（ITサービス管理）は、ビジネスチームが必要とするITサービスの安定的な提供と、継続的な改善を管理するためのマネージメント方法です。

ITシステムやソフトウェアというと、とかくシステム目線になりがちです。ですが、ITSMは、ITサービスを提供する立場で包括的に、ビジネス目線でITを見つめ直します。

このITSMのベストプラクティスをまとめたフレームワークが、ITIL®（ITインフラストラクチャライブラリ）です。ITILは歴史が長く、1980年代後半のイギリス政府のIT高度利用のガイドラインとしてスタートしています。以降、いくつかのバージョンアップを経て、最新のものは2018年に公開されたITIL 4です。



ITILはITサービスに限らず、幅広くサービスを提供する組織が効率的・継続的な運用を実現するためのノウハウが蓄積されたものです。

ITILをビジネス・組織に取り入れることで次のような効果が期待できます。

- 具体的な指標で追跡されるビジネスの優先事項とITチームの足並みをそろえる
- 複数の部門・チームにわたるコラボレーションを可能にする
- 合理化されたプロジェクト管理アプローチにより、ITチームと開発チームを連携させる
- ITチームがナレッジを共有し、継続的に改善できるように支援する
- より効率的なサービスを実現するためにリクエスト調整を改善する
- セルフサービスと優れたプラクティスにより顧客中心のサービスリクエスト管理を推進する
- 重大なインシデントに迅速に対応し、今後のインシデントを防止する

ITILがベストプラクティスとしてまとめているプラクティスは34項目に及びます(以前はプロセスと呼ばれていましたが、「ルール」の意味合いが強く響くため、プラクティスに変更されました)。

一般的マネジメントプラクティス	サービスマネジメントプラクティス	技術的マネジメントプラクティス
<ul style="list-style-type: none"> ● アーキテクチャ管理 ● 継続的な改善 ● 情報セキュリティ管理 ● ナレッジマネジメント ● 測定とレポート ● 組織変更管理 ● ポートフォリオ管理 ● プロジェクト管理 ● 関係管理 ● リスク管理 ● サービス財務管理 ● 戦略管理 ● サプライヤー管理 ● 要員および人材管理 	<ul style="list-style-type: none"> ● 可用性管理 ● ビジネス分析 ● キャパシティおよびパフォーマンス管理 ● 変更実現 ● インシデント管理 ● IT アセット管理 ● 監視およびイベント管理 ● 問題管理 ● リリース管理 ● サービスカタログ管理 ● サービス構成管理 ● サービス継続性管理 ● サービス設計 ● サービスデスク ● サービスレベル管理 ● サービスリクエスト管理 ● データ検証およびテスト 	<ul style="list-style-type: none"> ● デプロイ管理 ● インフラストラクチャおよびプラットフォーム管理 ● ソフトウェア開発および管理

個々の日本語名は弊社の解釈による翻訳です。

本書では、上述のような問題意識を中心に、アトラシアン製品を駆使してどのようにITSMを実践していくかを、お届けします。

サービスリクエスト管理

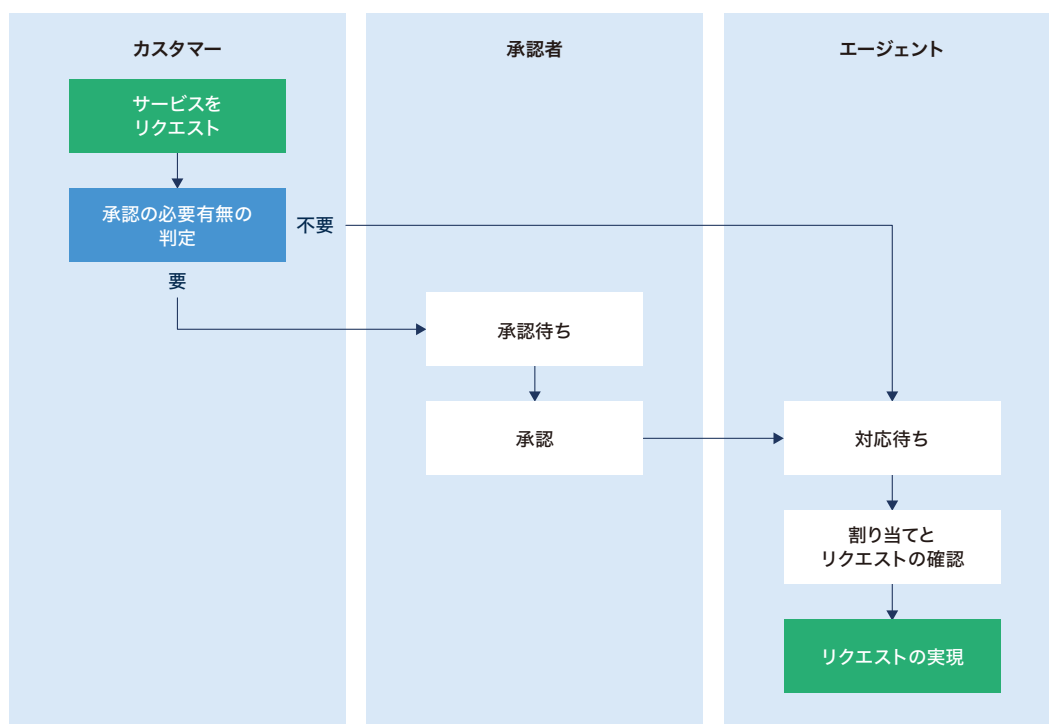
サービスリクエスト管理は、顧客やエンドユーザーからの広範囲に渡るサービスリクエストを受理し、処理し、要求を実現するための繰り返し可能な手順です。

次のようなものが考えられます。

- チームや部署に新しいメンバーが入ったので、パソコンを手配してほしい
- 新たなソフトウェアライセンスを発行してほしい
- パスワードを忘れてしまったので再発行してほしい

多くの場合、これらのリクエストは繰り返し発生するため、対応する側も繰り返し可能な手順で処理することが望まれます。

サービスリクエスト管理の一般的な流れ



サービスリクエスト管理を適切に設定することで、組織内外のリクエストの一元化、リクエスト処理の効率化を素早く実現できます。

インシデント管理

インシデント管理とは、突然のサービスの停止、中断、あるいはパフォーマンス低下などにより、通常のサービス提供が難しくなってしまった状態を、運用可能な状態に戻すための考え方です。

後述の問題管理とインシデントは、似て非なるものであると ITIL では定義します。ITIL は、問題を「1 つ以上のインシデントの原因、または潜在的な原因」と説明します。

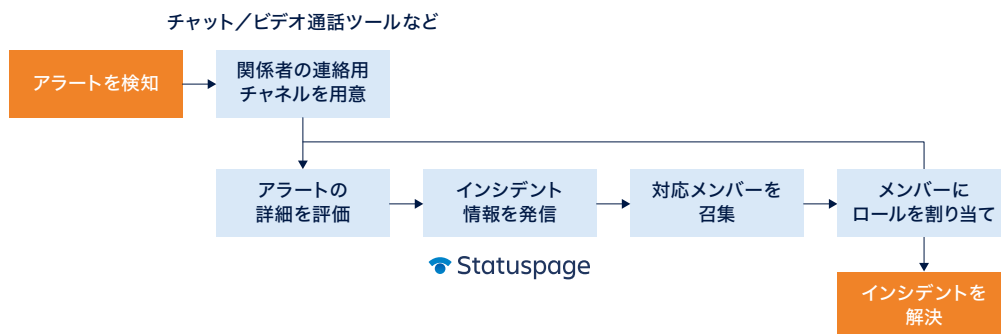
オンコールメンバー（障害対応担当者）が可能な限り迅速に対応し、サービスの中断状態を正常な状態にするのがインシデント管理の目的であるのに対して、そのような状態を引き起こした根本的な原因の管理を問題管理と捉えて、両者を峻別することが ITIL の意図するポイントになります。

ネットワークが遅くなり、ビジネス用のアプリケーションがダウンする「インシデント」では、ルーターの構成ミスがその背後にある根本的な「問題」である、などがその例になります。

あるいは、テクノロジーの外の世界で表現するならば、偏頭痛で医者に駆け込むことは「インシデント」（対応）になり、その根本原因として低気圧やストレスなどが「問題」となります。

ビジネスが IT サービスに依拠していくにつれて、インシデントは非常に大きな損失を引き起こしかねません。インシデント管理のプラクティスを定義し、それを多くのチームで共有し、学び続けていく体制と習慣を組織に根付かせていくことにより、インシデントがもたらす大きな損失を軽減できるでしょう。

アトラシアンでの、インシデント発生時におけるインシデントマネージャーの役割



インシデントをゼロにすることは不可能です。だからこそ、事前に強力なインシデント管理プラクティスを準備できるか、またその学びを組織に浸透していけるかが、インシデントの影響を軽減し、サービスの継続性を提供していけるかの、大きな差別化要因となるはずで

問題管理

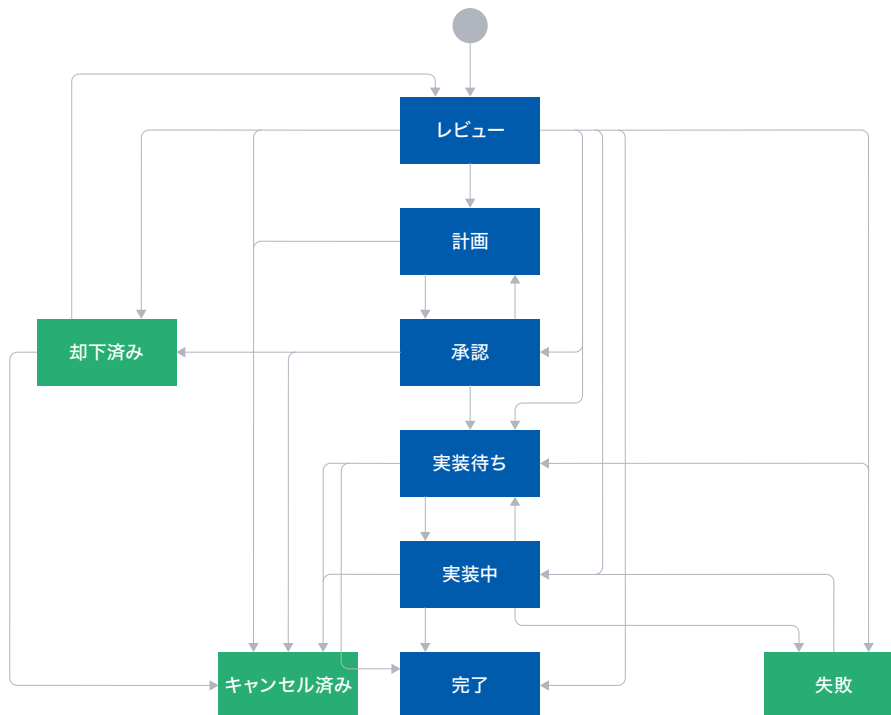
問題管理は、インシデントの根本原因を特定し、それを管理する考え方です。インシデント管理が通常のサービスを復旧させる最短の道を見つけることを目的としている一方で、問題管理はその根本原因の特定と、それを解決して再発を防止するための最良の方法を見つけることを目的としています。

問題管理は、そのため、インシデントが引き起こされた背景への総合的な理解が必要とされます。往々にして、組織が大きくなり専門の部署、チームが分離されていくにつれて、インシデントの根本原因の分析も、その部署ないしチームに閉じて行われてしまいます。うちの部署は悪くない、いや向こうのチームのミスである等、そうした組織のサイロ化は問題管理の真の効果を見えにくくしていくでしょう。

時間とともに組織が変化していくと、インシデントも、背後にある問題も、その本質が変わっていきます。組織がサービスを提供するためには、多くのチームが介在しています。システム管理、セキュリティ、開発チーム…。それらのチームを総合的に巻き込んで、問題の本質を検証していくことが求められるでしょう。

問題管理については、Jira Service Management での実装方法がリクエスト管理と同一であるため、本書では以降は取り上げません。

Jira Service Management での問題管理ワークフローの例



変更管理

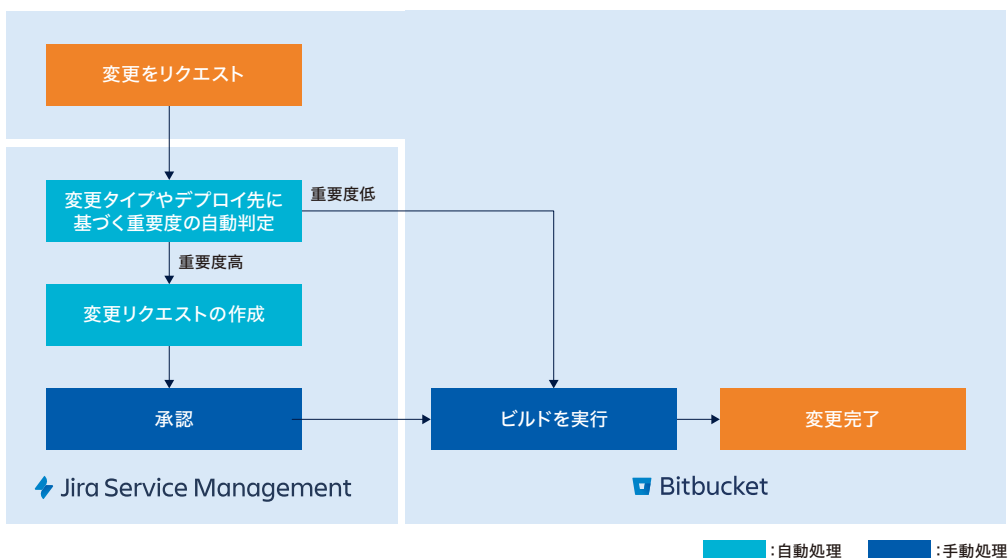
問題管理によって本質的な問題が特定され、インフラの強化やサービスに対する直接または間接的な変更が必要であると判断されたとします。このような状況で次に着目すべきは、それらの変更を安全かつ効率的に行うべき管理する、変更管理になります。

変更実現 (Change Enablement) とも呼ばれるプラクティスは、「コントロール」という官僚主義的な意味合いを避け、サービスを提供し、その変更を担う組織のすべてのチームに、サイロ化の解消、コンテキストと透明性の提供、ボトルネックの解消そしてリスクの最小化を実現します。

変化のスピードが速い現代の状況で、それに伴うセキュリティ対応、新たなコスト・ビジネス要件のためのサービスの更新が求められます。

安定的にサービスを運用することと、変化に対応してサービスを更新し、変更を加えること。この2つの矛盾した要件を実現していくことが、変更管理あるいは変更実現に求められる命題になります。

アトラシアン製品で実現する変更管理ワークフロー



アセット管理

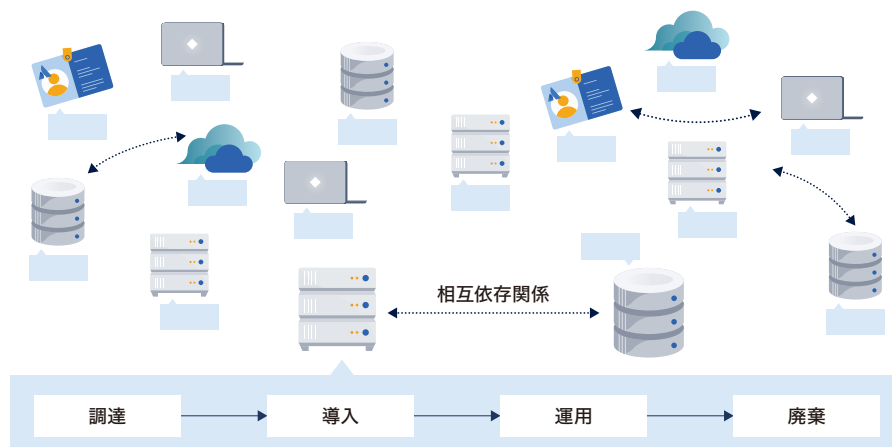
変更を与える対象となる、ハードウェア、ソフトウェアシステムなどをITアセットと呼びます。また、サービスを構成するためのもの以外に、組織内で使用するすべてのコンピュータやソフトウェアライセンスも同様にITアセットと捉えることができます。

アセット管理とは、ITアセットのライフサイクル、つまり調達、導入、運用、廃棄の全容を管理し、その使用を最適化することにあります。アセットの利用の情報を最新に保つことにより、無駄を省きまた不要な購入を回避して、コストの節約が可能になります。

アセット管理の目的と手法は、そのままIT以外にも適用できます。組織の重要な資産のライフサイクルを管理することで、組織全体に影響を与える意思決定を行う際の、信頼できる情報源を提供します。

さらに、ITアセットを総合的に把握しその関連性と依存関係も見えるようにすることで、すでに述べたインシデント管理、変更管理、問題管理の各プラクティスの強力なサポートとなります。

アセット管理



提供するサービスを構成するアイテムの相互の依存関係が見えることで、インシデントの原因やその影響範囲、あるいは変更を加える際にどのアイテムとどのアイテムが影響し合うのか、などの重要な判断の基準になります。アセット管理ではそれぞれのもののライフサイクルに着目する一方で、このような構成アイテムの関係性を考慮する見方を、構成管理と呼びます。

アセット管理、構成管理そのどちらも、組織全体でのビジネス上の意思決定をする際に、必要不可欠な情報源を提供するのに大いに役立つでしょう。

以上、ここまでITSMの主要なプラクティスを紹介しました。第2章以降では、紹介したそれぞれのプラクティスをアトラシアン製品で具体的に実現する方法を、ステップバイステップで説明します。

第 2 章

基本ユースケース

サービスリクエスト管理



第2章 サービスリクエスト管理

本章は「Jira製品 基本の使い方ガイド」の内容を前提としています。Jiraの操作方法についてご不明な点がある場合はこちらのドキュメントをご確認ください。

IT部門は現在、自社のデジタルトランスフォーメーションのイニシアチブを支援するための大きなプレッシャーにさらされています。このような状況で既存のツールが制約となり、新しいビジネス要件の柔軟な導入が妨げられてしまうというお話をよく聞かれます。こうしたツールは知識のサイロ化を進め、“エキスパート”が休暇を取ると業務が止まってしまうような状況につながります。

サイロ化されたツールは、開発、IT、およびビジネスチームの間にギャップを生みます。このようなチームは、ビジネスの変更に対応したり問題を素早く解決したりするために必要なコンテキスト情報を十分に得ることができません。

本書でご説明するJira Service Managementは、チームが高速なITSM（ITサービス管理）を実現するのを支援するサービス管理プラットフォームです。チームとプラクティスを1つの環境に集約し、自社の作業方法に合わせて最適化することができます。

サービスプロジェクトにおける主なロール

Jira Service Managementで作成されるプロジェクトは「サービスプロジェクト」と呼ばれます。一般的なサービスプロジェクトには、主に2つのロールがあります。

カスタマーは、ヘルプセンターにアクセスしてリクエストを起票するユーザーです。Jira Service Managementのデフォルト設定では、カスタマーはヘルプセンターにのみアクセスでき、サービスプロジェクトの内部にアクセスすることはできません。

エージェントは、プロジェクト内部にアクセスしてカスタマーからのリクエストに対応します。カスタマー対応の一環としてヘルプセンターにもアクセスできます。

ロール	目的	ヘルプセンターへのアクセス	サービスプロジェクト内部へのアクセス
カスタマー	リクエストの起票	○	×
エージェント	リクエストへの対応	○	○

2つのロールで目的が異なるため、利用できる画面や機能もそれぞれで異なります。以降の説明でも、カスタマー側とエージェント側に分けて紹介する機能があります。

サービスプロジェクトを作成する

それでは早速サービスプロジェクトを作ってみましょう。

Jiraでヘッダーメニューの[プロジェクト]をクリックして[プロジェクトを作成]を選択します。

- 1 [サービス管理] カテゴリをクリックすると、さまざまなプロジェクトテンプレートを確認できます。画面右側でそれぞれのテンプレートをクリックすると、プロジェクトに含まれる「リクエストタイプ」や「ワークフロー」を確認できます。リクエストタイプはカスタマーが起票できるリクエストの種類を表し、課題タイプに紐付けることができます。

プロジェクト テンプレート / サービス管理

プロジェクト テンプレート

一般的なサービス管理

チーム支援のために設計されたサービス デスクを使用して、優れたサービスを迅速に提供できます。リクエストを記録し、必要な回答を迅速に見つけ、リクエスト フォームにリンクされている記事を利用して自分で問題解決できる一元的な場所を従業員に提供します。

セルフサービスを推進する

Confluence を利用してナレッジ ベースを設定し、従業員がサポート エージェントに問い合わせなくても一般的な質問に対する回答を見つけられるようにします。従業員がリクエストを送信する前に関連記事を提示することで、エージェントの時間が空き、もっと複雑な課題に集中的に取り組むことができます。

新しい作業を簡単に処理

メール、チャット ツール、ヘルプ センター、その他のチャンネルからのリクエストがキューによって1か所にまとめられます。どのチームも、新しいリクエストを簡単に追跡、分類、割り当てできます。

製品

- Jira Service Management

推奨

サポート リクエストの効率的な管理を求めているあらゆるチーム

リクエスト タイプ

質問する

メールによるリクエスト

リクエストまたはインシデントを送信

ワークフロー

- やるべきこと
- 保留中
- 進行中
- 完了

テンプレートを 使用

- 2 [ITサービス管理] を選択して [テンプレートを 使用] をクリックし、先に進みましょう。

IT サービス管理

IT チーム向けに設計された IT サービス管理テンプレートは、高ペロシティのサービス管理を容易にします。強力な ITSM カテゴリと専用機能が含まれているため、短時間でインシデントを解決し、変更を承認し、問題を解決できます。

インシデントをより迅速に解決

ワンクリックで Opsgenie にエスカレーションします。インシデントで共同作業し、短時間で解決できます。非常に重大な問題の場合、オンコール スクジューリング、アラート、インシデント スウォーミングなどを使用し、対応に必要な情報を適切な担当者に提供します。インシデント 管理の詳細

変更のトラッキングと承認

開発チームとの障壁を取り除き、変更を自動化することでワークフローを加速します。自動化された変更リクエスト管理、高度な承認ワークフロー、Bitbucket Pipelines、Jenkins、CircleCI などの CI/CD ツールとの統合を利用して、負荷を軽減します。変更管理の詳細

製品

- Jira Service Management

推奨

IT チーム、DevOps

インシデントおよび変更管理機能を備えた、強力なサービス管理ソリューションを求めているあらゆるチーム

サービス リクエストのタイプ

IT ヘルプの依頼

アカウントの問題を解決

ゲスト用の Wi-Fi アカウントを用意する

オフィスの VPN をセットアップ

管理者アクセス権限をリクエスト

新規アカウントをリクエスト

新入社員のオンボーディング

テンプレートを 使用

サービスプロジェクトの機能を確認する(エージェント側)

1 プロジェクトが作成できました。



2 まずは、先ほど少し触れた「リクエストタイプ」の設定内容を確認してみましょう。[プロジェクト設定] > [リクエストタイプ] の順に選択します。リクエストタイプの一覧と、それぞれに対応する課題タイプを確認できます。リクエストタイプはカスタマー向けのリクエストカテゴリと考えることができます。起票後、そのリクエストは、リクエストタイプに紐付けられた課題タイプのワークフローに従って進められます。



リクエストタイプは自由に追加/編集できます

- ③ [プロジェクト設定に戻る] をクリックして、[SLA] を選択します。ここではリクエストの対応目安時間（SLA: Service Level Agreement/サービスレベル合意）を設定できます。

- ④ [プロジェクトに戻る] をクリックしてプロジェクトに戻りましょう。[キュー] 画面が表示されるので、早速 [リクエストの作成] をクリックしてリクエストを起票してみます。[タイプ] から「管理上のアクセス権限をリクエスト」リクエストタイプを選択して、要約を入力し、リクエストを作成します。

- ⑤ [キュー] 画面を更新すると、一覧にリクエストが追加されたことがわかります。要約部分をクリックして詳細を見てみましょう。



- ⑥ 画面の右側にSLAが表示されます。デフォルトの設定では、「最初のレスポンスまでの時間」と「解決までの時間」がそれぞれ、ステータスが「サポートからの連絡待ち」であるときに計測されます。



- 7 また、[アクティビティ] セクションの [コメント] には、[内部メモを追加] と [カスタマーに返信] の2つのオプションがあります。他のJira製品との大きな違いとして、Jira Service Managementでは、内部向けのコメントを追加できます。エージェント間でのやり取りを残したり、調査結果を記録しておきたかったりするときに便利です。内部メモを追加するとこのように灰色の「内部メモ」ラベルつきで表示されます。



続けて、カスタマー向けの機能や画面を確認してみましょう。

サービスプロジェクトの設定を確認する（カスタマー側）

プロジェクトに戻り、サイドバーから [チャンネル] をクリックして、[ヘルプセンター] のURLにカーソルを合わせて [開く] をクリックします。表示された画面の左上の [ヘルプセンター] をクリックします。ここが、カスタマーが最初にアクセスする場所です。

- 1 ここには、このJiraサイトに存在し、カスタマーにアクセス権が付与されているすべてのサービスプロジェクトが表示されます。カスタマーは自身の要件に合うプロジェクト、リクエストグループの順にクリックし、リクエストタイプを選択して、リクエストを起票できます（この画面はサイト内に複数のサービスプロジェクトがある場合にのみ表示されます）。



- 2 また、[ヘルプやサービスを検索] と書かれているテキストボックスにテキストを入力すると、関連するリクエストタイプや、プロジェクトに設定済みである場合はナレッジベースの関連記事を横断して検索できます。ナレッジベースについては後ほど改めてご説明します。



- ③ それではリクエストを起票してみましょう。先ほど作成したITサービス管理プロジェクトを選択し、「コンピューター」リクエストグループをクリックして「新たなモバイル機器」をクリックします。適当な要約を入力し、「マネージャーは誰ですか?」ではご自身のアカウントを指定して [送信] をクリックします。

ヘルプセンター / ITサービス管理
ITサービス管理
ようこそ。こちらのオプションを使ってITサービス管理のリクエストを作成できます。

お問い合わせ内容
コンピューター

どのようなリクエストですか?
新たなモバイル機器
携帯電話が必要ですか、あるいは交換の時期ですか?お知らせください。

次のユーザーの代理でリクエストを作成:
アトラス 花子 (amaraha0319@gmail.com)

要約
アトラス花子のモバイル端末のリブレース
e.g. New mobile phone for Alice Smith

おすすめの記事
VPNのトラブル (社外網)
ソース: ITサービス管理

マネージャーは誰ですか?
アトラス 花子

Phone details and justification

- ④ リクエストが起票されました。このタイミングでカスタマーのメールアドレス宛に、リクエストが起票された旨の通知メッセージが送信されています。カスタマー側のビューでは、要約、起票時に指定したマネージャーのほかに、画面右上でステータスや共有先を確認できます。

また、このリクエストは現在「承認を待っています」ステータスにあり、承認が必要です。このまま直接承認を行うこともできますが、ここでは実際の運用を考慮し、起票者と管理者が別であると仮定して機能を確認してみたいと思います。

ヘルプセンター / ITサービス管理 / IT-2
アトラス花子のモバイル端末のリブレース

アトラス 花子 さんによるリクエスト (今日 3:53 午後) 詳細を非表示
マネージャーは誰ですか?
アトラス 花子

アクティビティ
自動応答 今日 3:53 午後
リクエストには承認が必要です。1件の承認が必要です。
コメントを追加

ステータス
承認を待っています
このリクエストにはあなたの承認が必要です
承認
却下

通知を受け取る
リクエストをキャンセル

リクエストタイプ
新たなモバイル機器

共有先
アトラス 花子 (作成者)
共有

承認を待っています
1件の承認が必要です。
アトラス 花子

- 5 ご自身（承認者）のメールボックスを確認すると、承認を依頼するメールが届いています。

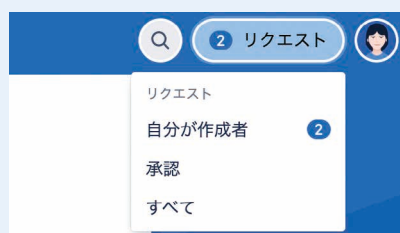


- 6 メールを開いて [承認] をクリックし、ポータルにログインすると（サインアップが必要になる可能性があります）、リクエストが承認された状態でこの画面に戻ることができます。



POINT

ヘルプセンターの右上にある [リクエスト] メニューをクリックすると、自身のアカウントで作成したリクエストや自身の承認待ちのリクエストの一覧にアクセスできます。



ナレッジベースについて

JiraサイトをConfluenceと連携させると、Confluenceにあるナレッジベース用のスペースをサービスプロジェクトにリンクすることができます。これにより、次のようなメリットがあります。なお、Confluenceの契約をお持ちでない場合でも、[ナレッジベース]メニューからConfluenceのFreeプランを有効化いただけます。

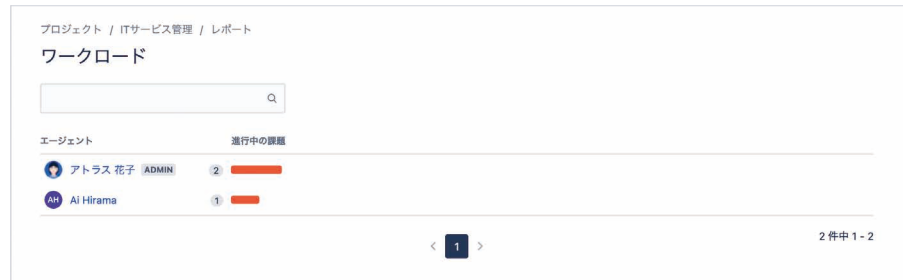
- カスタマーはヘルプセンターでナレッジベース記事を検索できる (p.23 手順2を参照)
- カスタマーによる起票時に、入力内容に基づいて関連ナレッジベース記事が提案される
- エージェントはナレッジベース記事を引用してリクエストを素早く解決できる



運用状況を把握する

SLAの達成状況や顧客満足度などの、サービスプロジェクト独自の指標をレポートとして確認できます。プロジェクトのサイドバーの[レポート]をクリックすると、デフォルトで10種類以上のレポートを確認できます。次のようなレポートがあります。

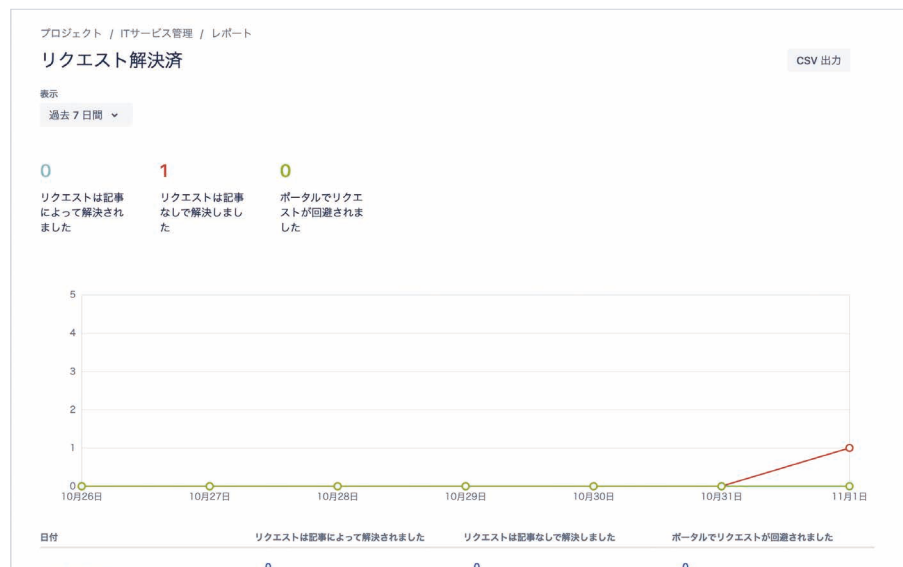
- ワークロード



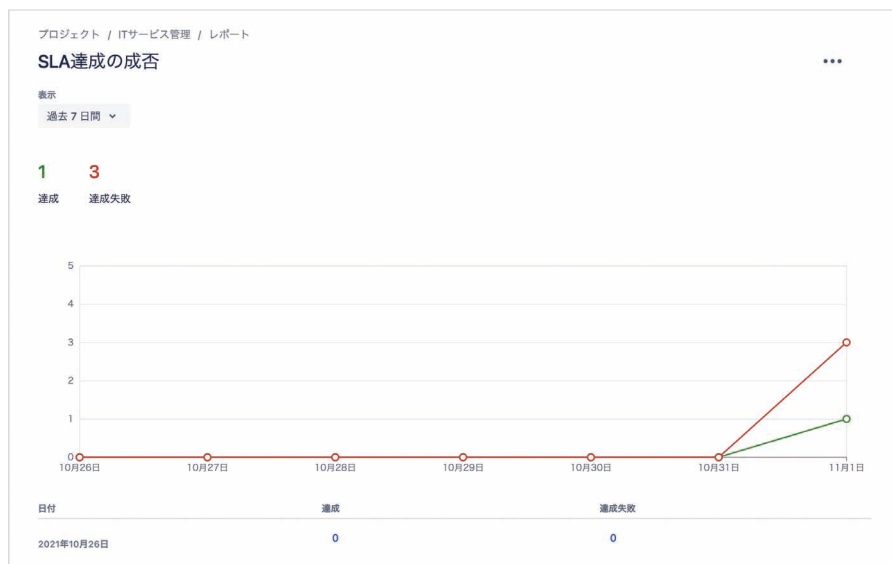
- 満足度 (顧客の満足度)



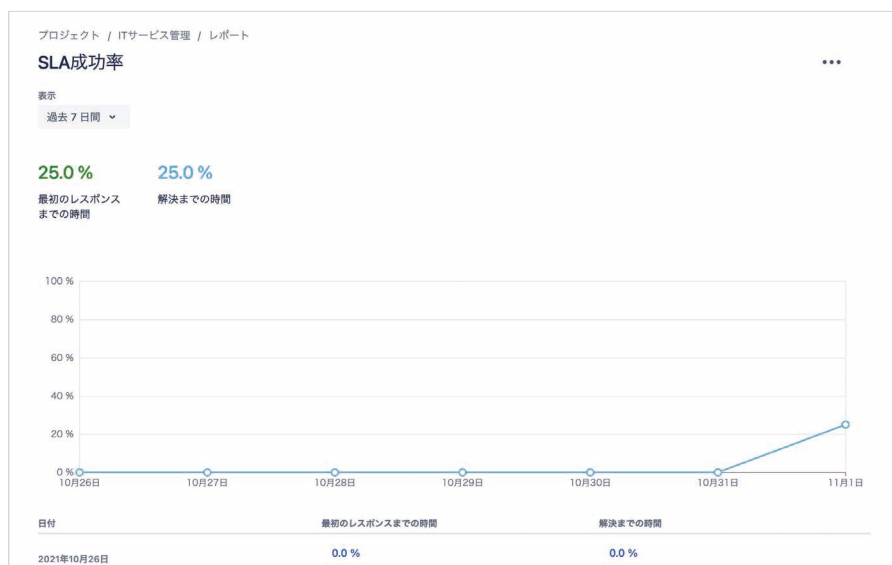
- 解決済みリクエスト



● SLA達成可否

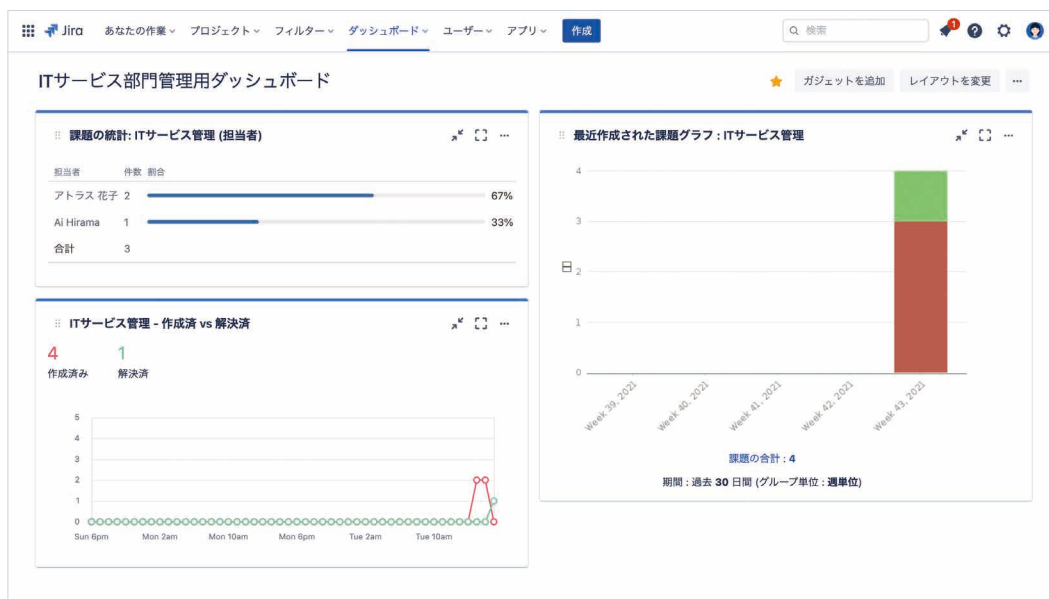


● SLA成功率



これらはすべて、対応するJira課題に基づいて動的に生成されるため、事前設定や手動でのデータ入力は不要です。

また、特定の指標を常に把握しておきたい場合は、対応するウィジェットをJiraのダッシュボードに追加できます。



さまざまなカスタマイズ

最後に、自社のブランディングやカスタマーに合わせてサービスプロジェクトをカスタマイズする方法をご紹介します。

- 1 最初にヘルプセンターをカスタマイズしてみましょう。サービスプロジェクトに移動し、[プロジェクト設定] > [ポータル設定] の順にクリックします。この画面では、ヘルプセンターに表示される各プロジェクトの名前や説明を変更できます。



- 2 下にスクロールして [ヘルプセンターのカスタマイズオプションを表示] をクリックすると、ヘルプセンター自体のカスタマイズを行えます。

バナー画像を変更した例



- ③ また、カスタマーの使用言語が複数にわたる場合、プロジェクトで多言語設定を構成できます。[プロジェクト設定] > [言語サポート] の順に選択して [言語を追加] すると、プロジェクトに言語を追加できます。追加後に言語部分のリンクをクリックすると、ポータルコンテンツやリクエストタイプなどの、カスタマーに表示されるさまざまな要素に翻訳を追加できます。



- ④ 最初にご紹介したSLAも、もちろんカスタマイズできます。[プロジェクト設定] > [SLA] の順にクリックします。ここではデフォルトのSLAを変更するほか、新しいSLAを追加することもできます。



- 5 右上のカレンダーアイコンをクリックすると、営業日や営業時間を設定できます。休日や祝日の情報を登録しておくことで、非稼働日にSLAが計算されるのを防ぐことができます。

カレンダー

カレンダーを追加 ✕

 サンプル勤務時間カレンダー ⋮

タイムゾーン
JIRA 既定 (GMT+09:00) 東京

勤務日

Monday	09:00 ~ 17:00
Tuesday	09:00 ~ 17:00
Wednesday	09:00 ~ 17:00
Thursday	09:00 ~ 17:00
Friday	09:00 ~ 17:00

休日

文化の日	定期的 03 11月
------	------------

Jira Service Managementを基本的なサービス管理のためにセットアップする方法についてご紹介いたしました。Jira Service Managementでは、本章でご説明したほかにも多数の機能が用意されています。個々のユースケースに合わせて製品をカスタマイズし、ぜひご活用ください。

第 **3** 章

応用例

インシデント管理



第3章 インシデント管理

本章は「Jira製品 基本の使い方ガイド」および本書「第2章 基本ユースケース：サービスリクエスト管理」の内容を前提としています。JiraやJira Service Managementの操作方法についてご不明な点がある場合はこちらのドキュメントをご確認ください。

インシデント管理とは、突然のサービスの停止、中断、あるいはパフォーマンス低下などにより、通常のサービス提供が難しくなってしまった状態を、運用可能な状態に戻すための考え方です。

残念ながら、インシデントを完全に無くすことは不可能です。しかしながら、組織の構造に合う適切なインシデント管理プラクティスを用意しておくことで、いざ問題が発生したときに迅速かつ効率的な対応を行うことができます。

なお、インシデント対応においては、修正のための内部でのやり取りや調整はもちろんのこと、外部向けのコミュニケーションも重要になってきます。適切な情報発信が行われないと、ユーザーのフラストレーションが溜まったり、ユーザーからの大量の問い合わせでチームが忙殺されてしまったりします。

総合すると、インシデント管理では次のような要素が重要となってきます。

- アラートを効率的に受け取ること
- 適切な担当者を割り当てること
- 問題の状況を素早く把握すること
- 問題の原因を素早く把握すること
- 適切な情報発信を行うこと
- インシデントを素早く解決すること

さらに、インシデントの解決後に、将来の再発を防止するためのふりかえりを行えると理想的です。

本章では、アトラシアン製のインシデント管理ソリューションを活用してこのような要素を実現する方法についてご説明します。

アトラシアン の インシデント 管理 ソリューション

アトラシアン の インシデント 管理 ソリューション では、サービス 管理 製品 である Jira Service Management と、それに 同梱 される インシデント 対応 製品 である Opsgenie を 主に 使用 します。

インシデント の 検知 方法 として は 一般 に、ユーザー による 報告 または モニタリング ツール による アラート 発信 が 考え られます。Jira Service Management では それ ぞれ 次 の よう に 対応 できます。

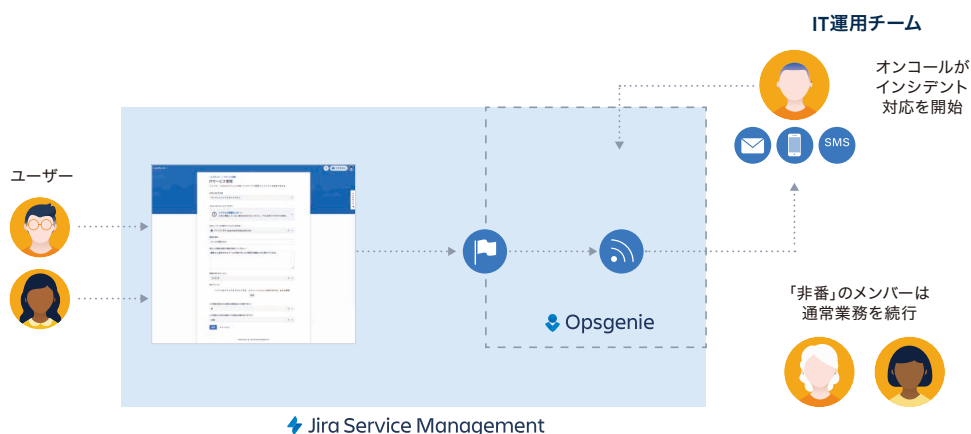
POINT

いずれ の ケース においても、インシデント 対応 者を 招集 する ため の アラート 通知 は その 時間帯 の 一次 担当者 に のみ 送信 されます。この ため、チーム の メンバー が 自身 に 不要 な 通知 で 忙殺 されて しまう、いわゆる 「アラート 疲れ」 に 陥る こと は あり ません。

また アラート 通知 の 送信 は、一次 担当者 が 対応 し ない 場合 は 二次 担当者 に、二次 担当者 が 対応 し ない 場合 は チーム 全員 に、の よう な 動的 な 条件 に 基づく 設定 である ため、一次 担当者 が 対応 でき なかった ため に チーム 全員 が インシデント を 見落 として しまう よう な こと も 防げ ます。

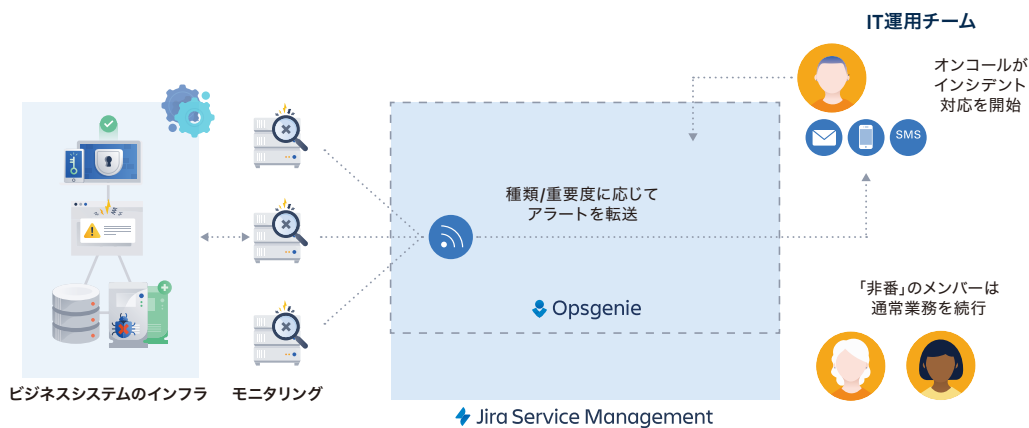
1 ユーザー による 問題 報告

ユーザー は 問題 に 気づいたら、サービス プロジェクト の ヘルプ センター から 報告 できます。これ によって 適切な 担当者 への アラート 通知 が 自動 的 に 行われ、時間 の ロス なく インシデント 対応 を 開始 することが できます。



2 インフラストラクチャのモニタリングツールが発信したアラート

ビジネスクリティカルなインフラストラクチャのモニタリングを行うツールを、Opsgenieと連携させることができます。アラートをOpsgenieに集約し、種類に応じたアラート通知条件をセットアップすることにより、重大なアラートを受信したときにのみ適切な担当者へのアラート通知を行うことができます。



POINT

モニタリングツールから発信されるアラートの連携機能について

Jira Service Managementに同梱されるOpsgenieではさまざまなモニタリングツールとの連携をセットアップし、受け取ったアラートに応じたアクションを指定することができます。

Opsgenieで提供される連携の種類についてはこちらをご確認ください。

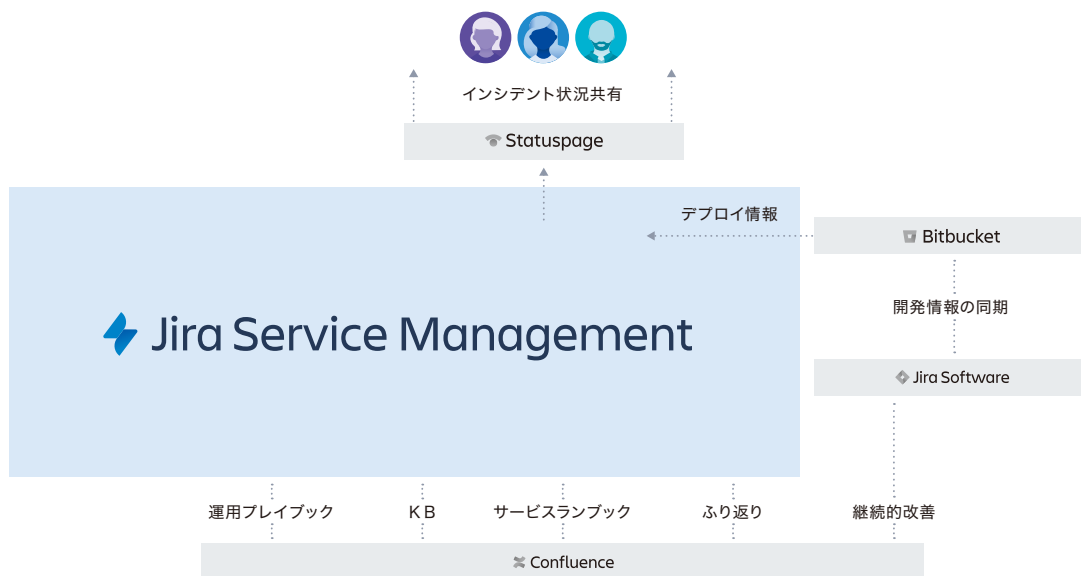
<https://support.atlassian.com/ja/opsgenie/docs/what-are-the-integration-types-and-actions/>

Opsgenieと連携可能な製品についてはこちらをご確認ください。

<https://www.atlassian.com/ja/software/opsgenie/integrations>

3 アトラシアン製品の組み合わせで実現する網羅的なインシデント対応ソリューション

インシデント状況の共有ツールであるStatuspageや、CI/CD機能を含むコード管理ツールであるBitbucketとの連携をセットアップすると、顧客向けのインシデント情報の自動的な発信/更新機能や、インシデントに紐づくデプロイの確認機能を活用できます。インシデントのふりかえり会議の議事録や対応マニュアルを、コラボレーションツールであるConfluenceに保存することもできます。



以降では、ユーザーによる報告をトリガーとしたインシデント対応の流れを、StatuspageやBitbucketとの連携を活用しながらご紹介させていただきます。

ユーザーが問題を報告した場合のインシデント対応の流れ

- 1 まず、問題に気づいたユーザーは、ヘルプセンターにアクセスし、関連するリクエストタイプを選んで問題を報告します。

The screenshot shows a web form titled "ヘルプセンター / ITサービス管理" (Help Center / IT Service Management). The form is for reporting an issue and includes the following fields and options:

- お問い合わせ内容** (Inquiry content): "サーバーとインフラストラクチャ" (Server and Infrastructure).
- どのようなリクエストですか?** (What kind of request is it?): "システムの障害をレポート" (Report system outage). A note below says: "正常に機能していない場合はお知らせください。できる限りすみやかな返信。" (If it is not working normally, please let us know. We will reply as soon as possible.)
- あなたのユーザーのID (リクエストを生成)** (Your user ID (Generate request)): "アトラス 花子 [anaruha0319@gmail.com]"
- 連絡先** (Contact): "メールが届かない" (Email not received). A note says: "発生した問題の詳細や経緯を説明してください。" (Please explain the details and history of the problem that occurred.) Below this is a text area with a "送信" (Send) button.
- 影響を受けるサービス** (Services affected): "Web" (Web).
- 附件ファイル** (Attachment files): "ファイルをドラッグ & ドロップする。スクリーンショットも貼り付け。または参照" (Drag & drop files. You can also attach screenshots. Or refer to them.) Below this is a "参照" (Refer) button.
- この問題の修正または解決の難易度はどの程度ですか?** (How difficult is it to fix or solve this problem?): "高" (High).
- この問題は自身や組織にどの程度の影響がありますか?** (How much impact does this problem have on you or your organization?): "広範" (Wide).
- Buttons: "送信" (Send) and "キャンセル" (Cancel).

- 2 エージェントのキューにサービスチケットが起票されます。

エージェントは問い合わせ内容を確認してトリージを行います。Affected services フィールドでは同じサービスに対して報告されているインシデントを確認し、ほかに影響を受けているユーザーや問題の状況を判断するために役立てることができます。

The screenshot shows the details of an incident titled "メールが届かない" (Email not received). The breadcrumb is "プロジェクト / ITサービス管理 / IT-6".

メールが届かない

サブタスクを作成 | 調査 | 課題をリンク | ...

アトラス 花子 このリクエストを登録しました 経由 ポータル
リクエストをポータルで表示 [詳細を非表示](#)

説明
 顧客から送信されたメールが届かないとの報告を複数人から受けています。

Affected services

メール

このサービスに 2 件の未解決インシデントがあります

- IT-7 メール
- IT-6 メールが届かない

説明
なし

階層
階層 1

緊急性
高

影響
広範

なおこの時点で、サービスオーナーとして登録されているチームのオンコールメンバーにアラート通知が送信されます。

③ インシデント対応者は次のような機能を利用して効果的なコミュニケーションを行います。

詳細 ^

担当者	 Ai Hirama
報告者	 アトラス 花子
Request Type	 システムの問題をレポート
Request language	日本語
優先度	 Medium
Severity	なし
ラベル	なし
A) ナレッジベース	 記事を検索または作成
Request participants	なし
Organizations	なし
リンクされたアラート	<input type="button" value="表示"/>
B) チャットチャンネル	チャンネルを作成
C) 電話会議	電話会議を開始 
D) 関係者	関係者 0 人
応答者	2 人の応答者
重大なインシデント 	<input type="checkbox"/> 

A) ナレッジベース: ナレッジベース記事の検索

サービスのインシデント対応ハンドブックなどをナレッジベースとして用意しておき、インシデントのチケットに表示させることで、インシデント対応全般を大幅に効率化したり意識合わせを簡単に行ったりすることができます。

B) チャットチャンネル: コミュニケーション用のSlackチャンネルを作成

インシデントについて議論するための専用Slackチャンネルを1クリックで作成できます。

作成したSlackチャンネルにアクセスすると、インシデントの概要がシステムによって投稿およびピンされていることがわかります。Slackチャンネルから議論に参加するメンバーがいる場合であってもツールを行き来することなく効率的に情報を把握できます。



C) 電話会議: ビデオ会議を即座に開始 (Premium、Enterpriseプランのみ)

上のメニューはSlackの利用が前提となっていますが、Slackをご利用ではない場合であっても製品内の機能を利用して即座に電話会議を開始できます。ボタンを押すとすぐに会議が開始されます。通話用のリンクを手動でセットアップしたり、関係者に周知したりする必要はありません。

誰かが会議を開始するとメニューの横に赤い丸印が表示されるため、インシデントチケットにアクセスしてきたユーザーはすでに議論が始まっていることを把握し、すぐに参加できます。



D) 関係者: ステークホルダーへの定時連絡

直接対応は行わないが情報を把握しておく必要はある内部の関係者(ステークホルダー)をインシデントに追加できます。ここで追加した関係者には、画面内のメニューを利用して一括でメッセージを送信できます。

関係者に情報を通知

影響を受けるサービスのインシデント応答者と関係者リストに手動で追加されたその他のユーザーに、メールを送信します。

要約*

メール受信不可のインシデントについて (14:00時点)

メッセージ*

現在有識者を収集し、Slackチャンネルを作成して状況の調査を開始しています。|

キャンセル 送信

4 担当者が集まり、インシデントの調査が開始されました。

コミュニケーション担当は上述の関係者メニューやStatuspageを利用し、ステークホルダーやユーザーに適宜情報を共有します。Statuspageではインシデントごとに専用のページが用意されるため、ほかのユーザーから同様の問い合わせがあったときにページのURLを共有することで対応負荷を減らせます。また、公式情報を1か所にわかりやすくまとめられるため、担当者の一存などによる意図せぬ情報開示を防ぐのにも役立ちます。

5 インシデントの原因が、直近にデプロイされたコード変更にある可能性があることがわかりました。

Bitbucket Cloudを連携させていると、インシデントチケット内の「調査」ボタンをクリックして、サービスに関連付けられたパイプラインの直近の成功状況を確認できます。



疑わしいデプロイが見つかったら、それを潜在的な原因としてマークできます。

6 無事にインシデントを解消できました。

担当者はインシデントチケットを更新し、障害の解消をユーザーに報告します。

- 7 Confluenceでインシデント事後レポートを作成して共有します。
 これは情報のオープン化につながるほか、ほかのチームで同様の問題を起こさないよう周知する効果もあります。

DWEN-1: インシデント事後レビュー

作成者: アトラス 花子
 最終更新日: 数秒前 · Analytics

📄 インシデントの概要 | 👤 エグゼクティブ サマリー | 📄 事後分析レポート | 🕒 インシデント タイムライン | ✅ フォローアップ タスク

📄 インシデントの概要

事後分析の所有者	@Takahiro Minagawa
インシデント	🔴 DWEN-1: メール配信サービスの停止 完了
関連インシデント	なし
優先事項	P1
影響を受けるサービス	メールサービス
インシデントの日付	2022年4月1日
インシデントの期間	3時間
インシデント対応チーム	メールチーム
インシデント応答者	@アトラス 花子

👤 エグゼクティブ サマリー

2022年4月1日の9:05-12:20の間、メールサービスが停止し、社内の全メールの送受信が行えなくなった。前日の2022年3月31日にサーバー内のコンポーネントのアップグレード作業を行っており、そこでの設定ミスに起因することが判明。当該設定を更新して問題を解消した。

📄 事後分析レポート

説明	レポート
⚠️ 下準備	2022年3月31日 定期点検の一貫として、サーバー内のコンポーネントのアップグレードを実施。対象：

このレポートをJira Service Managementのインシデントチケットにリンクして情報の網羅性を確保します。これでインシデント対応は完了です！

セットアップ情報

先ほどご紹介したフローを実現するセットアップをご紹介します。

■サービスのセットアップ

Jira Service Managementでは社内のサービス(上記の例では「メール」)を登録し、それぞれにサービスオーナーを指定できます。サービスオーナーにはインシデントの起票時にアラートが送信されます。

それではこちらをセットアップしてみましょう。なお、サービスオーナーには、個人ではなく同梱のOpsgenie内で管理される「チーム」が指定されます。

- 1 Jira Service Managementプロジェクトでサイドバーの [オンコール] メニューをクリックし、[新しいチームを追加] リンクをクリックします。

- 2 画面右上の [Add team] ボタンを使ってチームを追加します。

POINT

この画面は本ガイドの執筆時点では英語でのみ提供されています。チーム名も英数字のみである必要があります。

- 3 Jira Service Managementの画面に戻り、サイドバーの [サービス] メニューをクリックします。

- 4 [サービスを作成] をクリックし、サービスの作成画面を開きます。Nameを入力してTierを選び、Service Ownersに2で作成したチームを指定します。

- 5 なお、前述のインシデント対応の流れではユーザーが問題の報告時にヘルプセンターでサービスをドロップダウンから選択しています。このドロップダウンはデフォルトで提供されていますが、表示されていない場合は次のページの手順に従って手動で追加していただけます。

<https://support.atlassian.com/ja/jira-service-management-cloud/docs/customize-the-fields-of-a-request-type/>

■ナレッジベースのセットアップ

ナレッジベースの詳細については「第2章 サービスリクエスト管理」をご確認ください。

■アラート通知のセットアップ

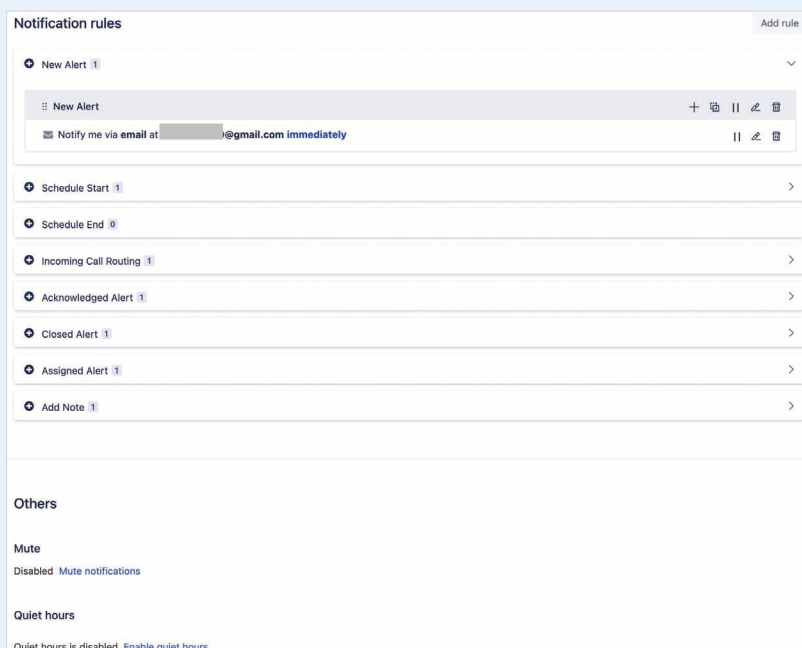
インシデント作成時のアラート通知は、デフォルトではユーザーのアカウントに紐付けられたメールアドレスに送信されます。受信者はこの設定を変更し、次のような方法でアラートを受信できます。

- メール（アカウントに紐付けられたメールアドレスとは別の宛先も指定可）
- SMS
- ボイスメール
- モバイルアプリ

また、通知の種類に応じてアラートの送信方法を変更したり、時間帯に応じて通知を無効化したりすることもできます。これらのセットアップにアクセスするには、Opsgenieにアクセスした状態で [Settings] → [Notifications] に移動します。

POINT

OpsgenieにアクセスするにはJira Service Managementの [アラート] メニューをクリックするか <https://<サイト名>.app.opsgenie.com/settings/integration/> にアクセスします。



■オンコールスケジュールのセットアップ

今日のシステムにおいて24時間365日の対応が要求されることは珍しくありません。Opsgenie内のオンコールスケジュール機能を利用し、チームの複数のメンバーによる輪番を可視化できます。ここでセットアップした内容はJira Service Managementのオンコールメニューに反映されるため、適切な対応者を特定する際にも役立ちます。

ここでは、Mailチームのセットアップ時のデフォルトの設定内容のご紹介を通じて機能を確認します。

- 1 Opsgenieにアクセスし、[Teams]メニューをクリックして対象のチームをクリックします。

POINT

OpsgenieにアクセスするにはJira Service Managementの[アラート]メニューをクリックするか <https://<サイト名>.app.opsgenie.com/settings/integration/> にアクセスします。

- 2 チーム作成時に用意されたデフォルトのオンコール条件が表示されます。この画面の内容は次のようになっています。
 - A) Routing rules: アラートごとのポリシー B) 適用条件
 - B) Escalation policies: チームへのアラート内容。オンコールスケジュール C) に従います
 - C) On-call schedules: チームのオンコールスケジュール

③ それぞれの内容を確認してみましょう。デフォルトでは次のようになっています。

Teams / Mail

On-call

Save view

Routing rules

Add routing rule ...

Route alerts to: Mail_escalation

Escalation policies

Add escalation ...

Mail_escalation

- 0 m: On call users in Mail_schedule, if not acknowledged
- 5 m: Next user in Mail_schedule, if not acknowledged
- 10 m: All members of Mail, if not acknowledged

On-call schedules

Add schedule ...

Mail_schedule (+09:00) JST Tokyo

Today < > Apr 11 - Apr 24 1 Day 1 Week 2 Weeks 1 Month Calendar Timeline

Rotations + Add rotation

	4/11 Mo	4/12 Tu	4/13 We	4/14 Th	4/15 Fr	4/16 Sa	4/17 Su	4/18 Mo	4/19 Tu	4/20 We	4/21 Th	4/22 Fr	4/23 Sa	4/24 Su
# Rot1	Ai Hirama													

Overrides + Add override Take on-call for an hour

	4/11 Mo	4/12 Tu	4/13 We	4/14 Th	4/15 Fr	4/16 Sa	4/17 Su	4/18 Mo	4/19 Tu	4/20 We	4/21 Th	4/22 Fr	4/23 Sa	4/24 Su

Final schedule

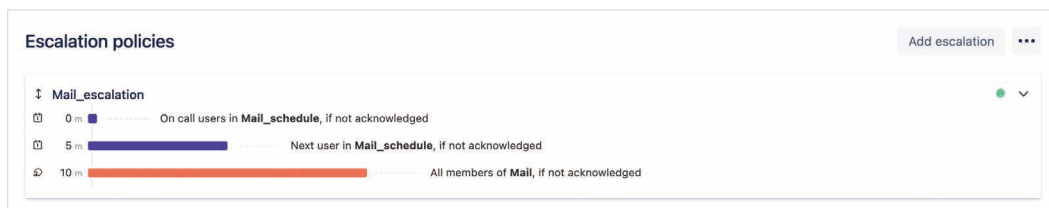
	4/11 Mo	4/12 Tu	4/13 We	4/14 Th	4/15 Fr	4/16 Sa	4/17 Su	4/18 Mo	4/19 Tu	4/20 We	4/21 Th	4/22 Fr	4/23 Sa	4/24 Su
Rot1's final		Ai	Hirama											

- 4 まず、一番下のOn-call schedulesを見てみましょう。デフォルトで作成された「Mai_schedule」カレンダーが表示され、上からRotations（通常輪番）、Overrides（一時対応）、Final schedule（最終的な輪番）を確認できます。

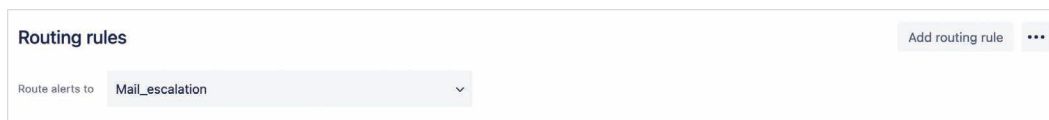
- 5 たとえば24時間体制で12時間ごとの交代制をとりたい場合、ここで [Rot1] の横にある鉛筆アイコンをクリックして [Edit rotations] をクリックしてセットアップを行えます。

6 1つ上のメニューに移動しましょう。Escalation policiesでは、状況に応じたアラート通知ルールをセットアップできます。デフォルトで作成されている「Mail_Escalation」ポリシーの内容は次のとおりです。

- 0分経過（アラートのトリガー時）: オンコールスケジュール「Mail_schedules」に則って一次担当者にアラート通知を送信
- 5分経過: アラートの確認が行われなかった場合、オンコールスケジュール「Mail_schedules」に則って二次担当者にアラート通知を送信
- 10分経過: アラートの確認が行われなかった場合、Mailチームの全員にアラート通知を送信



7 一番上のRouting rulesメニューでは、受け取ったアラートの種類に応じたポリシー転送ルールを指定できます。デフォルトでは、作成されたすべてのアラートを「Mail_Escalation」ポリシーに転送するように設定されています。



実は、ここまでにご説明した、スケジュール、ポリシー、ルールはいずれも複数作成できます。これらの組み合わせを通じ、さまざまな条件に応じてアラート通知の挙動や通知先の担当者を変えるような、柔軟なセットアップを行うことができます。

インシデント管理ソリューションに便利な他のアトラシアン製品

インシデント管理フローでご紹介した他のアトラシアン製品を簡単にご紹介します。

■ Statuspage について

Statuspage は、システムの状態をユーザーに共有するためのページを作成および公開できる製品です。次のような機能を利用できます。

- 複数のサービスやシステムの状態をまとめたステータスページを作成および公開
- ユーザーは任意のサービスやシステムにメール、RSS、SMS 等で購読し、インシデント発生時に通知を受け取ることが可能
- インシデントが登録されると対象のインシデント専用のページが自動的に作成されるため、顧客対応の担当者はその URL を共有することで対応負荷を軽減可能
- Jira Service Management、Opsgenie、サードパーティ製品と連携可能

詳細については次のページをご確認ください。

<https://www.atlassian.com/ja/software/statuspage>

■ Bitbucket Cloud について

上記フローで、インシデント Bitbucket Cloud 内のパイプラインを潜在的な原因として指定する機能をご紹介しました。Bitbucket Cloud はアトラシアンが提供している、CI/CD 機能を備えた Git 管理ツールです。Bitbucket Cloud の詳細については次のページをご確認ください。

<https://www.atlassian.com/ja/software/bitbucket>

サービスのセットアップ時にリポジトリの選択画面で Bitbucket Cloud のリポジトリを選択することで、当該サービスでインシデントが発生した際に対象のリポジトリのパイプラインの調査を直感的に開始することができます。

Jira Service Management でユーザーがアラートを報告する場合の一般的な流れ、利用可能なさまざまなセットアップ、他製品との連携機能をご紹介しました。

Jira Service Management と Opsgenie ではここでご紹介したもののほかにもさまざまな機能が提供されているため、さまざまなユースケースに合わせたカスタマイズを行っていただけます。ぜひ製品をお試しく下さい。

第 4 章

4

応用例

変更実現



第4章 変更実現

本章は「Jira製品 基本の使い方ガイド」および本書「第2章 基本ユースケース：サービスリクエスト管理」の内容を前提としています。JiraやJira Service Managementの操作方法についてご不明な点がある場合はこちらのドキュメントをご確認ください。

変更実現とは、ITサービスのリスクや中断を最小限に抑えながら重要なシステムやサービスの変更を実現することです。変更とは、直接または間接的にサービスに影響を与える可能性のある対象の、追加、変更、または削除です。

変更実現の仕組みがうまく整っていないと、承認者は承認依頼への対応のために複数のツールを切り替えなければならなくなったり、重要度やリスクに応じた切り分けを行えず、すべての承認を扱うことになって承認者自身がボトルネックになってしまったりします。このような状況では対応が場当たり的になってしまうほか、いつ何がどのように発生しているのかの把握も困難です。

アトラシアンの変更実現ソリューションは次のようなメリットを提供します。

- すべての関連情報をまとめ、1か所ですべてのコンテキストを把握
- 変更のリスクや影響範囲に応じて自動承認を行い、重要な変更の確認に集中
- いつどこでどのような変更が行われるかを直感的なカレンダー形式で把握

それでは早速見ていきましょう。

アトラシアンの変更実現ソリューションの流れ

アトラシアンの変更実現ソリューションでは、ユーザーによる変更の手動登録と、連携済みのCI/CDツールによる変更の自動登録の2種類の業務フローがあります。これらの両方でデフォルトの自動化機能を使い、重要な変更で人間による承認を要求しつつ、それ以外では自動承認を行えます。

1 変更実現: ユーザーによる手動登録

この業務フローでは次のように、デフォルトの自動化機能を2か所で利用して効率的な変更承認を実現しています。

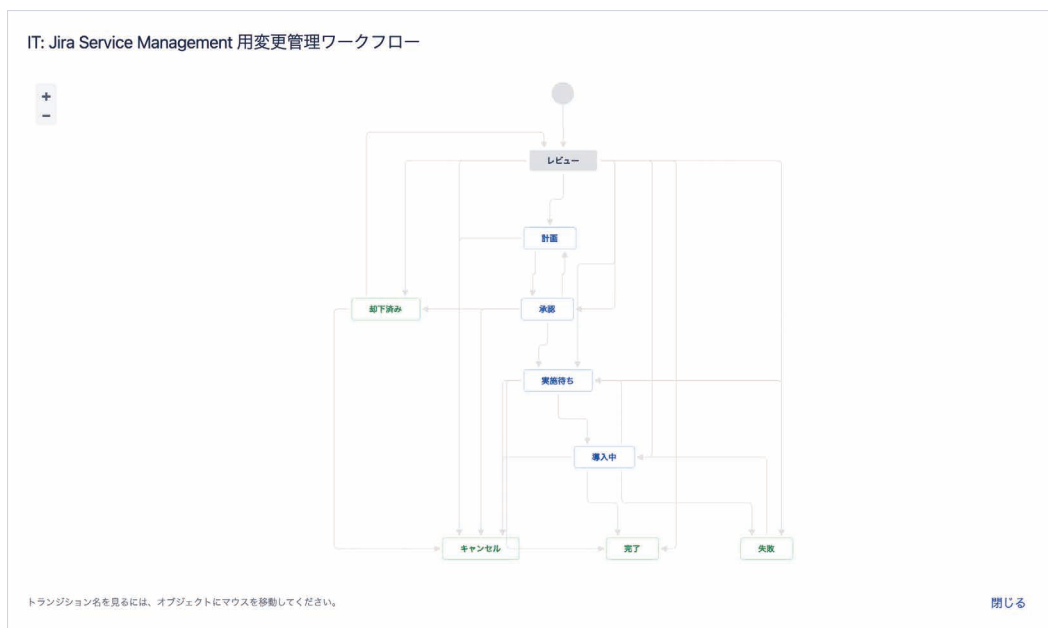
1. ユーザーがヘルプセンターから変更チケットを手動で起票
2. 登録内容に応じて変更リスクを自動的に設定 (第1の自動化ルール)
3. 一次対応者は変更の内容を確認し、ステータスを「承認」に設定
4. 変更のリスクに応じて次のようになる
 - 変更のリスクが「低」の場合は自動承認が実行される (第2の自動化ルール)
 - 変更のリスクが「低」以外の場合は自動承認は行われず、引き続き人間による承認待ちとなる



第1の自動化ルールでは、ユーザーがヘルプセンターで指定する「変更の影響を受けるサービス」に応じて「変更のリスク」フィールドが次のように設定されます。

変更の影響を受けるサービスのサービス階層	変更のリスク
1	クリティカル
2～4	低

第2の自動化ルールは、ワークフローの「承認」ステータスで実行されます。



エージェントがチケットをこのステータスに動かしたときに自動化機能が作動し、変更リスクに応じて次のように動作します。

トリガー	変更リスク	アクション
チケットが「承認」ステータスに到達	低	自動承認を行い、チケットを「実施待ち」にトランジション
チケットが「承認」ステータスに到達	高	何も行わず、人間による承認を待つ

これにより、変更の承認者は本当に大事な変更にのみ集中することができます。

POINT

本章ではデフォルトで用意されている自動化ルールを活用した例をお見せしていますが、自動化ルールは自由にカスタマイズ可能なため、自社の要件に合った方法で業務を効率化していただけます。自動化ルールの作成や活用方法は第6章でご説明します。

それでは、業務の流れをイメージしながら実際の画面をお見せしていきます。

■ 重要度の低いサービスの自動承認

変更に取り掛かってるユーザーがヘルプセンターにアクセスし、変更申請用のリクエストフォームを開きます。ここで、変更の要約、説明、影響を受けるサービス等を入力できます。まずは、重要度の低い「社内SNS」サービスへの変更を申請してみましょう。

POINT

サービスのセットアップ方法については「第3章 応用例：インシデント管理」をご確認ください。



POINT

変更申請用のリクエストフォームではデフォルトで次のフィールドが用意されています。フィールドは社内の変更管理要件に合わせて追加または削除できます。

- | | | | |
|--------------|----------|------------|---------|
| ・ 要約 | ・ 影響 | ・ 変更理由 | ・ 開始予定 |
| ・ 説明 | ・ 緊急性 | ・ 導入計画 | ・ 終了予定 |
| ・ 影響を受けるサービス | ・ 変更のリスク | ・ バックアウト計画 | ・ 実際の開始 |
| ・ 変更のタイプ | ・ 添付ファイル | ・ テスト計画 | ・ 実際の終了 |

変更チケットが作成されたら、エージェントビューを確認してみます。ユーザーの入力内容に加えて、変更リスクが「低」と自動的に設定されています。



一次対応者が内容を確認してステータスを「承認」に変更すると、その後ステータスが自動的に「実施待ち」に変更されます。変更のリスクが低いいため、自動承認が行われました。



この後、変更はJiraで設定されたワークフローに従って管理されます。

■ 重要度の高いサービスの手動承認

同じ条件で、対象のサービスを重要度の高い「メール」サービスにして起票してみましょう。起票が完了すると次のように「変更のリスク」が「クリティカル」になっています。



同様にチケットを「承認」に移動させると、先ほどとは異なり、承認待ちの状態ではワークフローが停止しました。承認者が「承認」または「却下」をクリックすると初めてワークフローが進みます。



このように、デフォルト設定をそのまま利用するだけで変更のリスクに応じた対応の切り分けを行い、承認者の業務の負担を軽減できます。また、このような処理はすべて自動的に行われるため、一次対応者がチケットの内容に応じた手動対応を行う必要がないのもポイントです。

2 変更実現: パイプライン連携による自動登録

ここでは、アトラシアンが提供するCI/CD機能を備えたコード管理ツールであるBitbucket Cloudを例に取り上げます。Bitbucket Cloudのデプロイパイプラインとサービスプロジェクトを連携させて、パイプラインの実行をトリガーに変更チケットを起票できます。

変更内容をサービスプロジェクト内で一元化するために役立つだけでなく、先ほどの例と同じように自動化を活用して承認有無を調整できます。

POINT

Jenkins、CircleCI、Octopus Deployでも同様の連携を行っていただけます。



それでは機能の流れをご紹介します。

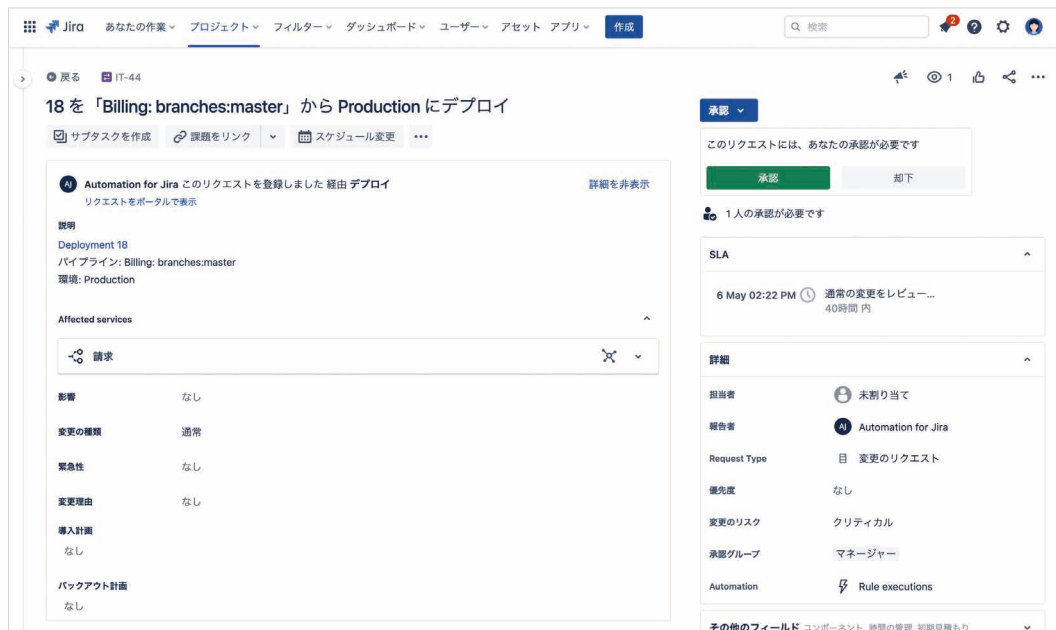
まず、開発者がコード変更をBitbucket Cloudにコミットします。すると、Bitbucket Cloudで自動的にパイプラインが実行されます。パイプラインの構成ファイルでは、テスト、ステージング、本番環境へのデプロイが定義されているため、それによって各環境へのデプロイが実施されます。

サービスプロジェクトには各デプロイについての変更チケットが起票されます。

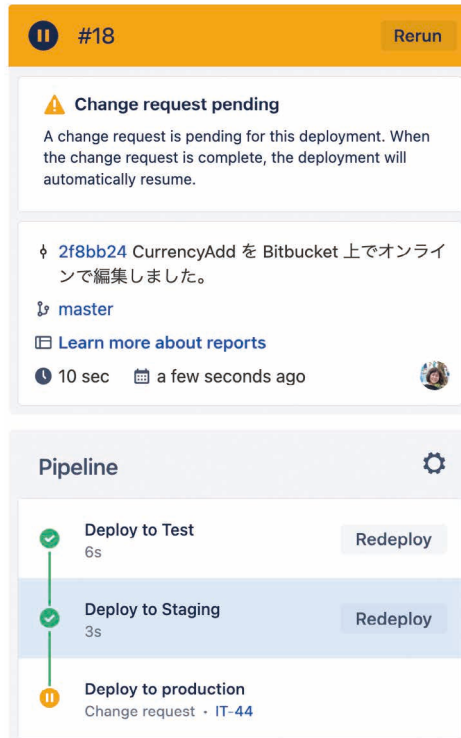
プロジェクト / ITサービス管理 / 変更							
全ての変更							
<input type="checkbox"/>	キー	要約	報告者	担当者	ステータス	変更の種類	作成日
<input type="checkbox"/>	IT-44	18を「Billing: branches:master」から Production にデプロイ	Automation for Jira	未割り当て	承認	通常	29/4/22
<input type="checkbox"/>	IT-43	18を「Billing: branches:master」から Staging にデプロイ	Automation for Jira	未割り当て	導入中	標準	29/4/22
<input type="checkbox"/>	IT-42	18を「Billing: branches:master」から Test にデプロイ	Automation for Jira	未割り当て	導入中	標準	29/4/22

テスト環境 (Test) とステージング (Staging) 環境へのデプロイは、ステータスが「導入中」、変更の種類が「標準」になっています。本番環境 (Production) へのデプロイはステータスが「承認」、変更の種類は「通常」です。これは、テスト環境とステージング環境への変更は自動承認を経て実装が進んでいるのに対し、本番環境への変更はリスクが大きいため、人間による承認を待っている状態です。

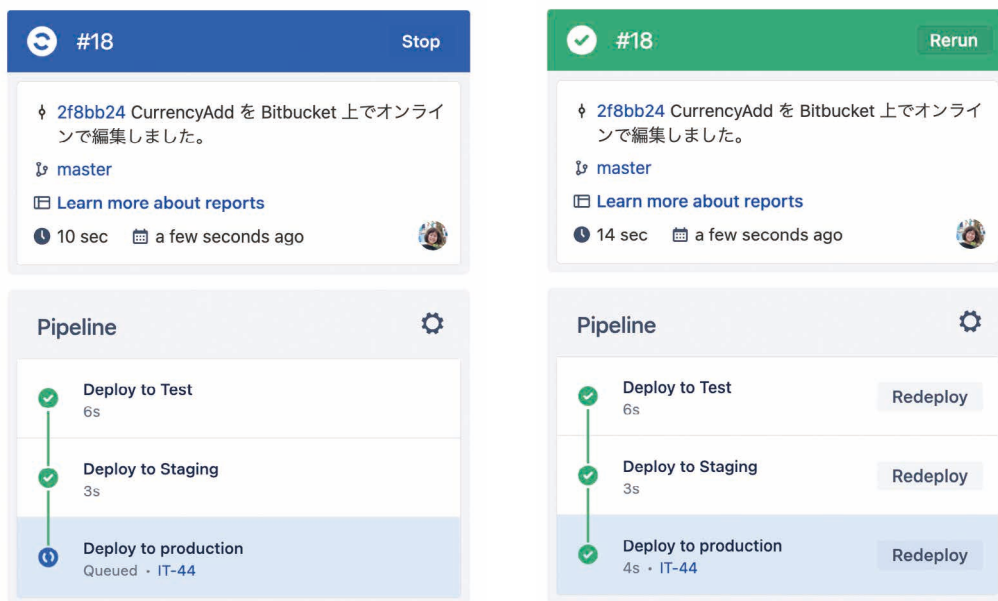
本番環境へのデプロイのチケットを開くと、次のように承認待ちの状態になっています。



このときに、トリガーとなっているBitbucket Cloud側でパイプラインの実行画面を確認すると、本番環境へのデプロイが次のように「Change request pending」の状態で停止していることがわかります。下部の停止アイコンのセクションに変更チケットIT-44へのリンクが表示されており、ここから同じチケットにアクセスすることもできます。



承認者が変更チケットIT-44を承認しました。Bitbucket Cloudでは承認通知を受け取り、パイプラインが自動的に再開されます。



このフローでは、変更管理者と開発者の双方にメリットがあります。

承認者は、リスクが高いと見なされる変更のみの承認に関与できるため、大量の変更申請の確認に時間を割かれるようなことがなくなります。開発者は、コードのコミットを行うだけで、あとはシステムが以降のプロセスを行ってくれます。パイプライン再開のために承認者による承認を画面の前で待ち続けるようなことがなくなります。

■ セットアップ情報

ご紹介したフローを利用するには、Bitbucket CloudのリポジトリとJira Service Managementのサービスの間
にセットアップが必要です。手順を簡単にご紹介します。



ご紹介した連携を Bitbucket Cloud で利用するには Bitbucket Cloud の Standard
または Premium プランが必要です。

【Bitbucket Cloudの準備】

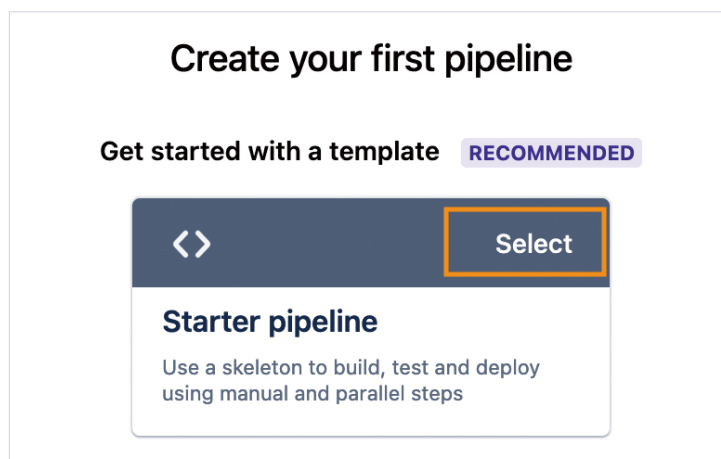
- 1 Bitbucket Cloud (<https://bitbucket.org>) でリポジトリを作成し、[Pipelines] メニューに移動します。

POINT

リポジトリの作成方法についてはこちらのページをご確認ください。

<https://support.atlassian.com/ja/bitbucket-cloud/docs/create-a-git-repository/>

- 2 少しスクロールし、[Starter pipeline] の右上の [Select] を選択します。



- 3 [Commit file] をクリックしてパイプラインを有効化しておきます。

パイプラインの詳細については次のページをご確認ください。

<https://support.atlassian.com/ja/bitbucket-cloud/docs/get-started-with-bitbucket-pipelines/>

【Jira Service Management側のセットアップ】

- ① サービスプロジェクトで [プロジェクト設定] > [変更管理] の順に移動し、 [パイプラインを接続] をクリックします。画面の内容に従って次のように設定します。

- A 接続するツール: Bitbucket
- B サービス: [既存のサービスを使う] を選択し、任意のサービスを選択
- C 環境: ここで選択した環境へのデプロイで変更チケットが作成されます
- D リクエストタイプ: 変更リクエスト

- ② 次のようにパイプライン連携がセットアップされます。

デプロイ パイプライン

デプロイ パイプラインを Jira Service Management に接続し、デプロイをトラッキングして IT サービスに加えられた変更を監査します。サービス プロジェクトのデプロイのトラッキングを設定する方法をご覧ください。

デプロイ パイプラインを有効化

デプロイのトラッキング

デプロイを変更リクエストとして自動で記録

環境のタイプ

開発 x 本番 x ステージング x テスティング x
⊗ ▼

デプロイは選択された環境について記録されます。

リクエストタイプ

変更のリクエスト
▼

各デプロイを記録するため、変更リクエストが自動的に作成されます。変更リクエストのタイプを管理します。

- ③ [デプロイ ゲートの構成] で [変更リクエストのワークフローのステータスを使用するデプロイを許可・却下する] を選択します。 [デプロイを許可] と [デプロイを実行しない] の2つのドロップダウンが表示されるので、デプロイの許可および却下を示すステータスを選択します。

デプロイ ゲートの構成

変更管理手順に基づいてデプロイを許可または却下します。CI/CD ツールの互換性を確認するには、デプロイ ゲートを設定する方法をご覧ください。

デプロイゲートを使用しない

変更リクエストのワークフローのステータスを使用するデプロイを許可・却下する

デプロイを許可

実施待ち
▼

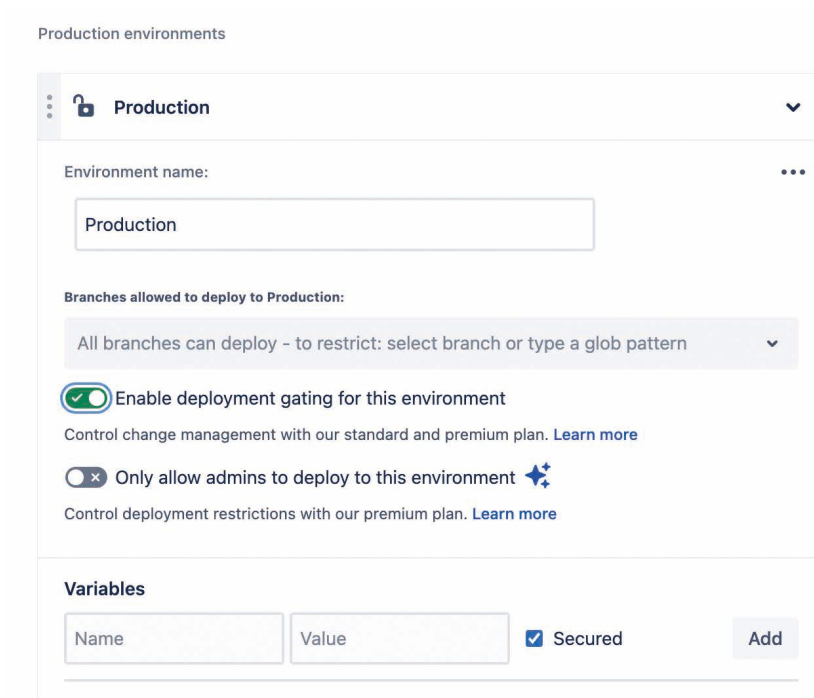
デプロイを実行しない

却下済み
▼

変更を保存
放棄

【Bitbucket Cloud側のセットアップ】

- 1 Bitbucket Cloudで最初にセットアップしたリポジトリに戻り、 [Repository settings] > [Deployments] に移動します。
- 2 ゲーティング（変更のデプロイの一次停止）を設定したい環境で [Enable deployment gating for this environment] のトグルを有効化します。



これで完了です！

【設定のテスト】

早速パイプラインをトリガーしてみましょう。

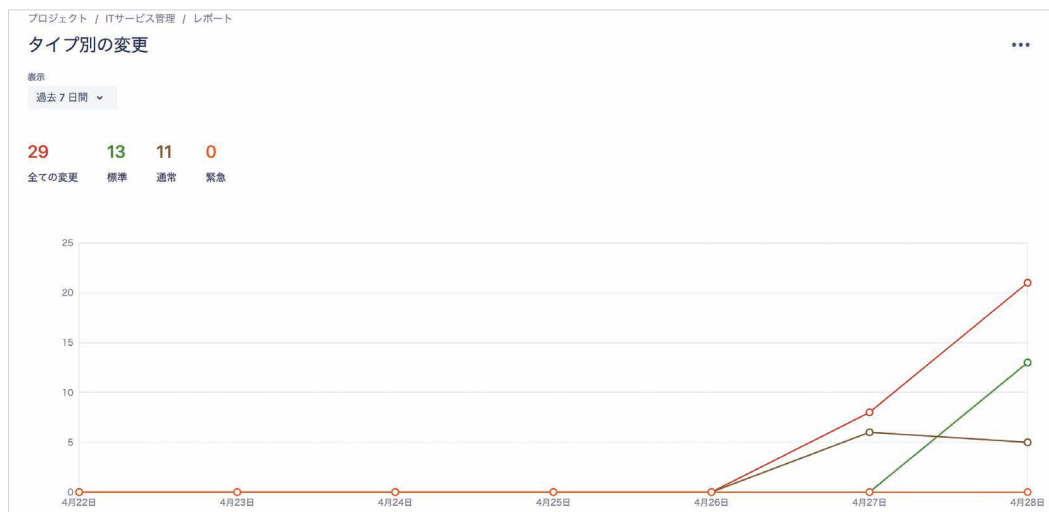
- 1 リポジトリに戻り、 [Pipelines] メニューに移動します。
- 2 [Run pipeline] をクリックし、任意のブランチとパイプラインを選択して [Run] をクリックします。
- 3 サービスプロジェクトに変更チケットが作成され、パイプラインがProductionで停止したら成功です！

変更実現を支えるさまざまな機能

最後に、変更実現のプラクティスを支えるさまざまな機能をご紹介します。

1 レポート機能

デフォルトで用意されている「タイプ別の変更」レポートでは、起票されている変更の種類別の推移を確認できます。



たとえば標準変更が増えているような場合、チームが良い傾向にあると考えることができるかもしれません。緊急変更が急に増えているようでしたら開発やテストの流れを見直す必要があるでしょう。

レポート機能では、解決までの時間などのほかのさまざまな種類の追跡も行えるほか、独自の指標でカスタムレポートを作成することもできます。

2 変更カレンダー機能

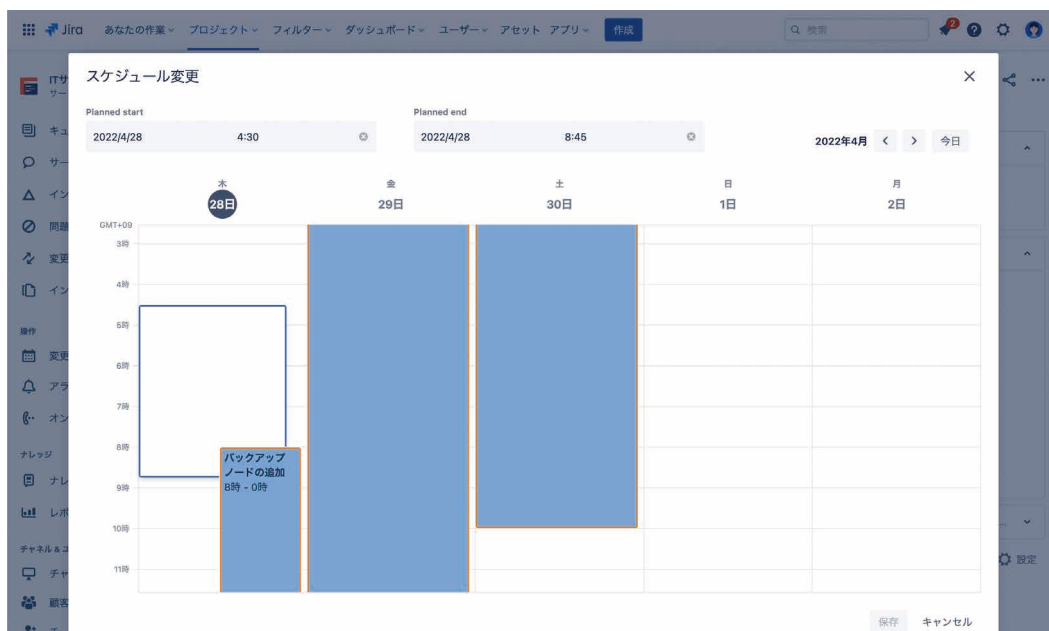
サービスプロジェクトに含まれる変更カレンダーでは、複数の変更の開始および終了予定日を網羅的に確認できます。



カレンダーはプロジェクト内メニューにあります。このプロジェクトに留まらず、複数のプロジェクトを横断して特定のサービスへの変更予定一覧を網羅的に確認することもできます。サービスの変更予定をひと目で把握できるため、意図せぬ競合の発生を防止できます。

また、メンテナンス予定を同じように変更チケットとして起票しておくことで、変更をいざ実施しようと思ったらメンテナンス中、のような事態を回避できます。

カレンダーは変更チケットの画面でも呼び出せます。チケットを開いて [スケジュール変更] ボタンをクリックすると、チケットの変更内容にフォーカスしたカレンダーを確認できます。



青い線で囲まれた白色の部分が、この変更チケットの内容です。ほかの変更と時間が重複していることがひと目でわかります。ここで変更をドラッグし、時間が重ならないように調整できます。

Jira Service Managementで実現する、変更実現ソリューションの流れをご紹介します。

変更実現ソリューションは、変更管理者と開発者両方の負荷を軽減しつつ必要な承認は確実にできるような設計されているため、現場の効率を大幅に高めるのに役立ちます。

第 **5** 章

応用例

アセット管理



第5章 アセット管理

本章は「Jira製品 基本の使い方ガイド」および本書「第2章 基本ユースケース：サービスリクエスト管理」の内容を前提としています。JiraやJira Service Managementの操作方法についてご不明な点がある場合はこちらのドキュメントをご確認ください。

アセット管理とは、組織内のアセット（資産）の計上、導入、維持、アップグレード、処分を適切なタイミングで行えるようにする考え方です。ここでのアセットとは、有形と無形のもの両方を指すことができます。

適切なアセット管理を行うことで次のようなメリットを実現できます。

- 「遊んでいる」アセットを発見し、ビジネスの意思決定の改善や無駄の軽減につなげる
- アセット間の依存関係を可視化し、アセットで問題が発生したときに影響範囲を素早く判断
- アセットの操作やライフサイクルを記録し、監査要件に対応

また、アセット管理ツールをチケットと連携させることもできます。たとえば、社内の備品情報と社内のサポートチームのチケットとを関連付けて、備品についての問い合わせに効率的に対応できます。あるいは、従業員をアセットとして登録し、新入社員用の準備を行うチームのチケットと連携させることもできます。アイディア次第であらゆる連携や効率化を行えるのがこのソリューションの特徴と言えるでしょう。

本章では、アトラシアンのアセット管理ソリューションの活用例をお見せします。

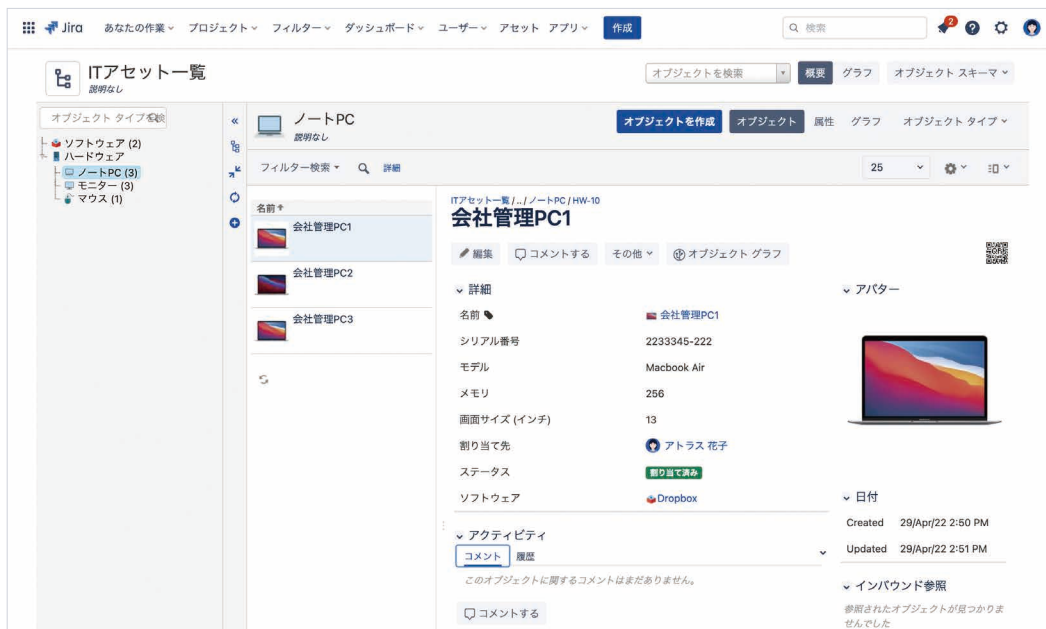
アセット管理

1 アトラシアンのアセット管理機能

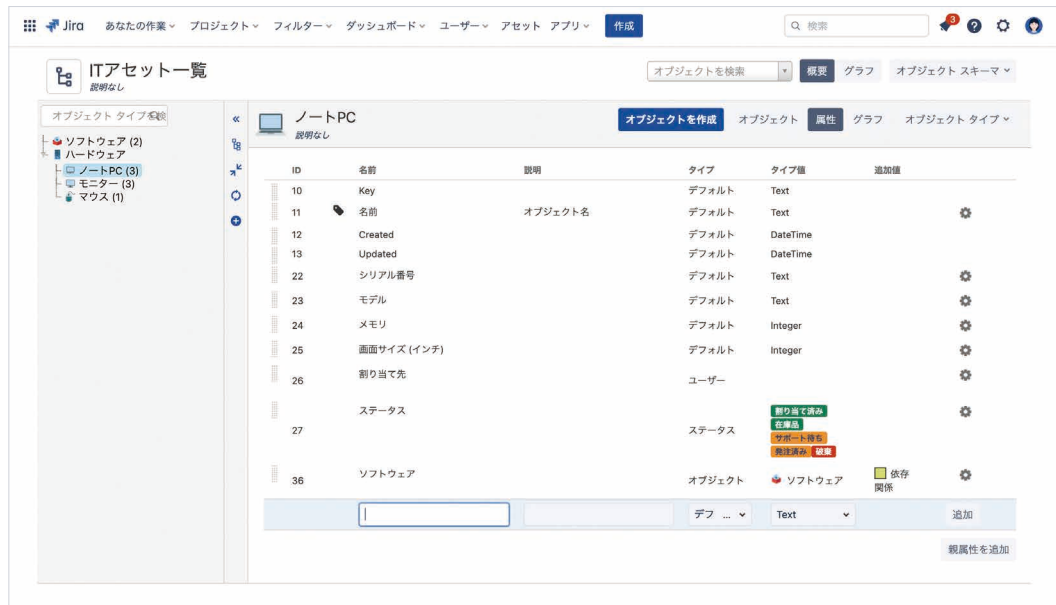
アセット機能は、Jira Service ManagementのPremiumおよびEnterpriseプランに無料で付属するアセット管理ツールです。アセット機能では、「オブジェクトスキーマ」内でさまざまな「オブジェクトタイプ」を定義し、それに「オブジェクト」として現実世界の実体を当てはめて管理していきます。たとえば、次のような構成が考えられます。

オブジェクトスキーマ (全体の名前)	オブジェクトタイプ (備品のカテゴリ)	オブジェクト (実際の管理対象備品)
ハードウェア備品一覧	ノートPC	会社管理PC1、会社管理PC2、会社管理PC3…
	モニター	26インチディスプレイ1、26インチディスプレイ2、30インチディスプレイ…
	マウス	マウス1、マウス2、マウス3…

実際の画面はこのように見えます。



[詳細] セクションの配下にあるさまざまな項目は属性と呼ばれ、オブジェクトタイプ単位で定義されます。ここで紹介している「ノートPC」オブジェクトタイプの属性設定は次のようになっています。



各属性の入力項目では、テキストや数値などのほかに、ユーザー情報(上記「割り当て先」属性)、ステータス、他オブジェクト(上記「ソフトウェア」属性)を利用できます。

また、グラフメニューを利用すると、オブジェクト同士の関連性を確認できます。この例ではノートPCオブジェクトとソフトウェアオブジェクトとの間に関連性があるため、次のように見えます。



アセット間の依存関係を簡単に可視化できるため、変更やインシデントの影響を確認するのにとても便利です。

最初にご案内したように、アセット機能では有形無形問わずあらゆるものを管理できます。たとえば次のような定義方法が考えられます。

オブジェクトスキーマ	オブジェクトタイプ
企業情報	オフィス、地域、業者、従業員…
施設	オフィス、フロア、部屋、管理者、消火設備…
従業員	氏名、役職、上長、勤務地…
本番環境	サーバー、データベース、ロードバランサ、ファイアウォール…

POINT

オブジェクトタイプの属性を定義するときには、可能な限り別のオブジェクトタイプを参照するようにします。たとえば上の例にある施設スキーマで、一次的なオブジェクトである「オフィス」と「フロア」を次のようにセットアップいたします。

オブジェクトタイプ	オブジェクト
オフィス	東京、大阪、福岡、札幌
フロア	1F、2F、3F

このときに、同じオブジェクトスキーマ内の「消火設備」オブジェクトタイプでは設置場所を手入力するのではなくこれら2つのフィールドを参照するようにします。



疎結合の状態にすることで、オフィスの移転などで情報が変わったときの反映作業を最小限に抑えることができます。

スキーマ同士の相互参照をセットアップすることもできます。たとえば上記の例の従業員スキーマで、個々の従業員の勤務地として施設オブジェクトスキーマのフロアを参照できます。

2 アセット機能とJira Service Management

アセット機能は単体でも利用できますが、Jira Service Managementと連携させるとさらに便利です。代表的な連携例をご紹介します。

■ ノートPCの修理の申請フロー

アセットとして社内の備品情報を格納し、それを問い合わせ者や社内サポートチームが参照できるようにしておきます。従業員はヘルプセンターにアクセスし、ノートPCの修理メニューを選びます。ポータルでは、問題の概要のほかに「対象のノートPC」という名前のドロップダウンが表示されています。ここを選択すると、問い合わせ者に割り当てられたPCの一覧が表示されます。このため、修理が必要な対象を簡単かつ確実に選べます。



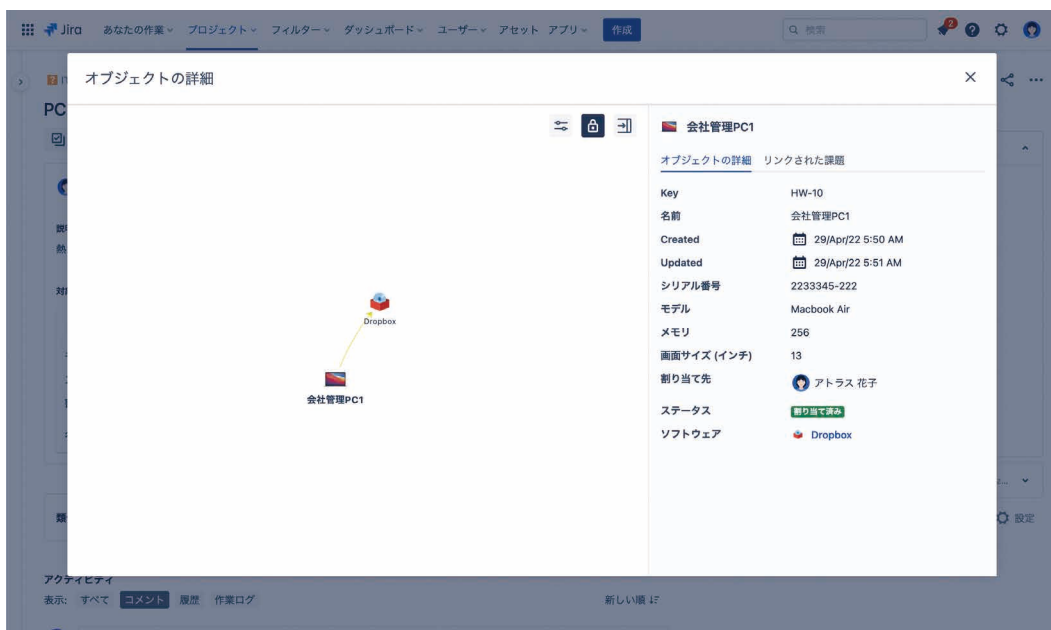
対象のノートPCを選択し、リクエストを起票しました。早速エージェントビューで内容を確認してみましょう。エージェントビューにも「対象のノートPC」というフィールドが表示されています。



このフィールドの右端にある下向き矢印をクリックすると、報告されたPCの詳細情報を確認できます。



また、下向き矢印の左側にあるアイコンをクリックすると、報告されたPCの依存関係をJira内で確認できます。



このように、すべての必要情報を1か所で把握できるため、エージェントは最小限の労力で報告されたアセットの位置づけを確認し、必要な対応を行うことができます。

■ セットアップ情報

サービスプロジェクトでアセットのデータを取り込むには、Jira側でアセット用のカスタムフィールドを作成する必要があります。ここでは、最初の例でご説明した「ITアセット一覧」オブジェクトスキーマの「ノートPC」オブジェクトタイプを取り込みます。

POINT

アセット内のデータはここまでのスクリーンショットのようにセットアップ済みであるとしています。

【カスタムフィールドの作成】

- 1 Jiraで、歯車アイコン > [課題] > [カスタムフィールド] に移動します。
- 2 [カスタムフィールドを作成] をクリックし、[詳細] メニューから [アセット オブジェクト] を選びます。
- 3 名前（ここでは「対象のノートPC」）を入力して作成し、対象のサービスプロジェクトの画面を関連付けます

【カスタムフィールドのセットアップ】

- 1 カスタムフィールド一覧に戻ったら、画面上部の検索ウィンドウを使って先ほど作成したフィールドを見つけ、3点アイコン > [コンテキストと既定値] を選択します。
- 2 [アセット オブジェクトのフィールド設定を編集] をクリックします。
- 3 この画面では、作成したカスタムフィールドにおいてアセット機能から取り込む情報を定義します。次のように情報を入力します。



項目	入力内容
オブジェクトスキーマ	ITアセット一覧
範囲を絞り込む (CMDB クエリ)	objecttype="ノートPC"
課題の範囲を絞り込む (CMDB クエリ)	割り当て先 == \${reporter}
指定した属性での絞り込み検索を許可する	名前、シリアル番号
課題ビューに表示されるオブジェクトの属性	モデル、ステータス、割り当て先、名前

【リクエストタイプにカスタムフィールドを設定】

- 1 対象のサービスプロジェクトに移動し、[プロジェクト設定] > [リクエストタイプ] で対象のリクエストタイプの編集画面を開きます。
- 2 [フィールドを追加] をクリックし、先ほど作成したフィールドを追加します。



- 3 これで完了です！早速ヘルプセンターに移動して挙動を確認してみましょう。

POINT

ここでは基本的なアセット連携をご紹介しましたが、アセット連携をJiraの自動化機能と組み合わせることでさらなる効果を実現できます。自動化機能は次の章で取り上げます。

3 インシデント管理ソリューションでのアセット機能の活用例

第3章でご説明したインシデント管理ソリューションにアセット機能を活用することもできます。

アセット機能ではデフォルトで Services という名前のオブジェクトスキーマが用意されており、Jira のサービスプロジェクト内でインシデントや変更管理用に定義された個々のサービスがオブジェクトとして同期されています。このため、各サービスの背後のシステム環境（データベース、アプリなど）についての情報をアセット機能で定義し、Services オブジェクトスキーマのサービスオブジェクトへの参照関係を設定しておくことで、次のようなフローでスムーズなインシデント対応を支援できます。

- 1 報告者はヘルプセンターでインシデントを報告する際、ドロップダウンから対象のサービスを選択
- 2 対応者は、対象サービスに紐付けられた他のサービスやアセットを Jira 内で把握し、インシデント要因として考えられる上流の要素や、このインシデントの影響を受ける下流の要素をひと目で確認可能（表示されるアセットをクリックすると詳細な構成情報を確認可能）

オブジェクトの詳細	
Key	SVC-18486
Name	カスタマーサポート
Created	08/Apr/22 7:16 AM
Updated	08/Apr/22 7:16 AM
Description	サポートを含む顧客むけのサービス
Tier	Tier 1
Service ID	ari:cloud:graph::service/e973fb61-e8d5-49a5-8fd1-1e040ad12517/ceda413c-b70-b-11ec-a563-0abe3f4a6601
Revision	707792578

オブジェクトの詳細	
Key	AP-9804
Name	Oracle CRM
Created	07/Apr/22 6:58 AM
Updated	11/Apr/22 4:31 AM
Status	PRODUCTION
Vendor	Oracle
Deployment Method	On-Premises
SaaS Provider	Oracle
Support Groups	Platform Dev
Approval Group	Platform Support
Classification	Operations
Business Capability	Customer Management
Services-TEXT	Customer Support
Access Levels	Admin User Power User
Services	< カスタマーサポート
Host	centos-srv-7

Jira Service Managementで実現する、アセット管理ソリューションの流れをご紹介します。

アセット管理ソリューションは、柔軟性のある情報整理と、問い合わせ対応時の簡単な参照の、いわゆる「いいとこ取り」を実現します。この柔軟性や俯瞰性をインシデント管理や変更実現のソリューションに拡張できるのも大きな強みです。

第 **6** 章

応用例

高度なカスタマイズ



第6章 高度なカスタマイズ

ここまでの章では、Jira Service ManagementをITSMのさまざまなプラクティスに活用する方法をご紹介してきました。

本章ではITSMプラクティスからは少し離れ、Jira Service Managementそのものを拡張するさまざまな便利なカスタマイズ機能をご紹介します。これらはいずれもJira Service Managementの全プランに無料で同梱されます(チャット機能のみ有料プランのみへの同梱)。

フォーム

サービスプロジェクトのリクエストタイプにデフォルトで設定されているものとは別に、より高度なフォームを作成できます。

ヘルプセンター / ITサービス管理

ITサービス管理

ようこそ。こちらのオプションを使ってITサービス管理のリクエストを作成できます。

お問い合わせ内容
ログインとアカウント

どのようなリクエストですか？
新しい従業員を研修
新入社員のアクセス権限をリクエスト

新規従業員情報

名前*	上司*
<input type="text"/>	選択...
役職*	開始日*
<input type="text"/>	例: 2022/4/30
従業員のタイプ	場所
選択...	<input type="text"/>

オフィスのセットアップ

勤務地 <input type="text"/>	新しいワークステーションは必要ですか？*
	<input type="radio"/> はい <input type="radio"/> いいえ
社員証は必要ですか？	就業時間外のアクセスは必要ですか？
<input type="radio"/> はい <input type="radio"/> いいえ	<input type="radio"/> はい <input type="radio"/> いいえ
名刺は必要ですか？	駐車場は必要ですか？
<input type="radio"/> はい <input type="radio"/> いいえ	<input type="radio"/> はい <input type="radio"/> いいえ

ITセットアップ - 必要なIT資産またはタスクを確認

電話 <input type="checkbox"/> 固定電話、内線 <input type="checkbox"/> 音声メッセージボックス <input type="checkbox"/> 共有音声メッセージメールボックスへのアクセス <input type="checkbox"/> 携帯電話	ハードウェア <input type="checkbox"/> 新しいデスクトップ <input type="checkbox"/> 新しいラップトップ <input type="checkbox"/> 既存のデスクトップをセットアップ <input type="checkbox"/> 既存のラップトップをセットアップ	ネットワーク <input type="checkbox"/> メールアカウント <input type="checkbox"/> 共有ドライブアクセス <input type="checkbox"/> 標準ソフトウェア設定 <input type="checkbox"/> 特定のソフトウェア設定
---	---	--

記入者

注記

名前	記入日
選択...	例: 2022/4/30

Powered by Jira Service Management

デフォルトのフォームとは異なり、入力項目を表として表示したり、正規表現を利用した検証を行ったりすることができます。また、サービスプロジェクトデフォルトのフォームでは入力項目がJiraフィールドに紐付けられている必要がありますが、フォーム機能のフォームではJiraフィールドに紐付かない入力項目を用意できるため、セットアップをより気軽に行えます。

さらに、フォーム機能では動的な制御が行えるため、入力内容に応じてメニューを追加で表示するようなことも可能です。たとえばこのフォームでは、「駐車場は必要ですか?」で「はい」が選択されたときにのみ、駐車場用の追加メニューを表示しています。

オフィスのセットアップ

<p>勤務地</p> <input style="width: 90%;" type="text"/>	<p>新しいワークステーションは必要ですか? *</p> <p><input type="radio"/> はい</p> <p><input type="radio"/> いいえ</p>
<p>社員証は必要ですか?</p> <p><input type="radio"/> はい</p> <p><input type="radio"/> いいえ</p>	<p>営業時間外のアクセスは必要ですか?</p> <p><input type="radio"/> はい</p> <p><input type="radio"/> いいえ</p>
<p>名刺は必要ですか?</p> <p><input type="radio"/> はい</p> <p><input type="radio"/> いいえ</p>	<p>駐車場は必要ですか?</p> <p><input checked="" type="radio"/> はい</p> <p><input type="radio"/> いいえ</p>

駐車場

タイプ

標準

身障者用

キャンパス

A

B

C

このような条件分岐のセットアップは次のように直感的に行えます。



まとめると、次のような違いがあります。

	サービスプロジェクトデフォルトのフォーム	フォーム機能で追加されるフォーム
入力項目のテキスト、複数選択、択一選択	○	○
入力項目を表として整理	×	○
入力内容を Jira フィールドに関連付け	必須	任意
入力項目を動的に表示 (条件分岐)	×	○

1 さまざまな場所でのフォームの見え方

■ 問い合わせ側

ヘルプセンターでは、次のように既存のリクエストタイプに埋め込まれて表示されます。

ヘルプセンター / ITサービス管理

ITサービス管理

ようこそ、こちらのオプションを使ってITサービス管理のリクエストを作成できます。

お問い合わせ内容
ログインとアカウント

どのようなリクエストですか？
新しい従業員を研修
新入社員のアクセス権限をリクエスト

新規従業員情報

名前*	上司*
<input type="text"/>	選択...
役職*	開始日*
<input type="text"/>	例: 2022/4/30 <input type="text"/>
従業員のタイプ	場所
選択...	<input type="text"/>

オフィスのセットアップ

勤務地	新しいワークステーションは必要ですか？*
<input type="text"/>	<input type="radio"/> はい <input type="radio"/> いいえ
社員証は必要ですか？	営業時間外のアクセスは必要ですか？
<input type="radio"/> はい <input type="radio"/> いいえ	<input type="radio"/> はい <input type="radio"/> いいえ
名前が必要ですか？	駐車場は必要ですか？
<input type="radio"/> はい <input type="radio"/> いいえ	<input type="radio"/> はい <input type="radio"/> いいえ

ITセットアップ - 必要なIT資産またはタスクを確認

電話	ハードウェア	ネットワーク
<input type="checkbox"/> 固定電話、内線 <input type="checkbox"/> 音声メッセージボックス <input type="checkbox"/> 共有音声メッセージメールボックスへのアクセス <input type="checkbox"/> 携帯電話	<input type="checkbox"/> 新しいデスクトップ <input type="checkbox"/> 新しいラップトップ <input type="checkbox"/> 既存のデスクトップをセットアップ <input type="checkbox"/> 既存のラップトップをセットアップ	<input type="checkbox"/> メールアカウント <input type="checkbox"/> 共有ドライブアクセス <input type="checkbox"/> 標準ソフトウェア設定 <input type="checkbox"/> 特定のソフトウェア設定

記入者

注記

名前	記入日
選択...	例: 2022/4/30 <input type="text"/>

Powered by Jira Service Management

起票後も引き続き表示されるため、問い合わせ者は自分が送信した内容をいつでも確認できます。この画面での編集は、許可することも禁止することもできます。



■ エージェント側

もちろん、エージェント用の画面にもフォームは表示されます。



2 さまざまなテンプレート

フォームでは300以上のテンプレート（うち、執筆時点で日本語に翻訳済みのものは18個）が用意されているため、要件に近いものを選び、そこから自社のニーズに合わせてカスタマイズしていくことができます。




チャット

サービスプロジェクトのチャット機能(Jira Service ManagementのFree以外のプランで提供)では、サービスプロジェクトをご利用のSlackワークスペースに接続して次のようなことが行なえます。

- Slack内のメッセージに絵文字リアクションを行うだけでサービスチケットを起票
- 起票後はSlackメッセージとサービスチケットが同期されるため、チケットへのコメントをSlackスレッドの更新として把握可能(逆方向も可)

■ Slackメッセージに絵文字リアクションを行い、チケットを作成して通知を受け取り




The screenshot shows a Slack thread titled "Thread #help-it". The thread starts with a message from user "ahirama" at 1:01 PM: "デスクフォンが壊れたので新しいものを用意していただけますか？" (My smartphone is broken, can you prepare a new one?). Below the message are reaction icons for "1" and a smiley face. The thread shows "3 replies". The first reply is from the "Assist" app, 3 minutes ago, with the text "[サポート対応待ち] Ticket IT-265 (edited)". A blue vertical bar highlights a section titled "Requested by @ahirama" with the same question as the original message. Below this, it says "Assigned to: アトラス花子" and a "Change status" button. The second reply is also from the "Assist" app, 3 minutes ago, with a green checkmark icon and the text "Nice work! Your ticket has been created. Visit @Assist to learn more."

■ エージェントはサービスデスクまたはSlackのトリアージ用チャンネルで対応を実施


アクティビティ


表示: [すべて](#) [コメント](#) [履歴](#) [Worklogs](#) 古い順 ↑

 アトラス 花子 7分前 ・ 🔒 内部メモ
 前回の変更申請が2022年3月です。どのように利用しているのかを一度確認したほうがいいかもしれません。
 編集 · 削除 · 🗨️

 アトラス 花子 6分前
 お問い合わせいただきありがとうございます。記録を確認すると、比較的最近リプレースしたばかりのようです。壊れてしまったとことで、背景を聞かせていただくことはできますか？
 編集 · 削除 · 🗨️


Thread 🔒 triage-it ×


 **Assist** APP Today at 1:01 PM
 [サポート対応待ち] Ticket IT-265
 Created via #help-it (edited)


 **Requested by @ahirama**
 デスクフォンが壊れたので新しいものを用意していただけますか？
 Assigned to: アトラス花子

[Edit](#) [Change status](#) [Expand](#)

3 replies

 **Assist** APP 7 minutes ago
 The conversation in this thread will be synced with the ticket
 Pro tip: Use the 🔒 emoji at the start of your message to add a private note to a ticket

 **アトラス花子** 6 minutes ago
 🔒 前回の変更申請が2022年3月です。どのように利用しているのかを一度確認したほうがいいかもしれません。

 **アトラス花子** 5 minutes ago
 お問い合わせいただきありがとうございます。記録を確認すると、比較的最近リプレースしたばかりのようです。壊れてしまったとことで、背景を聞かせていただくことはできますか？

POINT

もちろん、内部コメントもサービスチケット ⇄ Slackで同期できます。内部コメントは、Jira側では通常の手順、Slack側では 🔒 絵文字を利用して追加できます。

チャット機能のメリット

チャット機能を利用することで次のようなメリットを活用できます。

■ 問い合わせ側

- Slack内で問い合わせの起票～対応完了を完結できるため、ツールの切り替えやヘルプセンターを探す手間がなくなる

■ エージェント側

- 作業場所がSlackかサービスプロジェクトかにかかわらず、作業内容がサービスチケットとして自動的に記録されるため、確実な作業管理を実現
- 問い合わせがダイレクトメールで来るような場合であっても、それをサービスチケットとして起票することでチームに共有し、作業を分散可能

チャット機能は、サービスプロジェクトの [プロジェクト設定] > [チャット] から簡単に有効化していただけます。ぜひお試しください。

POINT

チャット機能は、本章の執筆時点ではSlack用にのみ提供されています。今後Microsoft Teams用のチャット機能も提供予定です。

自動化

自動化機能では「自動化ルール」を作ってプロジェクトに設定できます。

1 自動化ルールの構成要素

1つの自動化ルールは、主に次の要素で構成されます。

- トリガー：ルールの作動条件
- 条件：ルールの適用対象の絞り込み
- アクション：ルールで実現する内容

たとえば、優先度が「高」のチケットが起票されたらそれを特定の担当者に割り当てるような自動化を設定したいとします。この要件は次のようにブレイクダウンできます。

トリガー	条件	アクション
チケット（課題）の作成	優先度=高	特定の担当者に割り当て

ここから実際にルール化すると、次のようになります。

高優先度のチケットの自動割り当て

📄 ルール詳細

📄 監査ログ

+ いつ: 課題の作成時
ルールは、課題が作成されたときに実行されます。

⚙️ 優先度 と等しい
高

🔒 その場合: 次に課題を割り当て:
👤 アトラス 花子

○ コンポーネントを追加

それぞれの要素は次のように設定されます。非常に直感的にセットアップできるのがおわかりいただけるかと思います。



ほかにも次のような自動化ルールが考えられます。

- 毎週月曜日の朝9時に週次タスク用のチケットを作成
- チケット内のボタンをクリックすると、定型タスク用の子チケットを作成
- チケットのステータスが変更されたら期日を+5日

2 スマートバリューでチケットの内容を動的に取得

自動化機能では、チケットの内容を動的に取得する「スマートバリュー」も利用できます。たとえば上記ルールに、担当者の割り当てについて報告者にお知らせするようなコメントを追加する要素を追加してみます。

スマートバリュー ({{reporter.displayName}}、{{assignee.displayName}}) を使って報告者と担当者の名前を動的に取得しています。

このルールが作動すると次のようになります。

POINT

- スマートバリューの詳細については次のページをご確認ください。
<https://support.atlassian.com/ja/cloud-automation/docs/smart-values-in-jira-automation/>
 - 次のURLで人気の自動化ルールを公開しています。こちらでルールのセットアップをお試しいただくこともできます。
<https://www.atlassian.com/ja/software/jira/automation-template-library#/>
- また製品内では、サービスプロジェクトの [プロジェクト設定] > [自動化] で製品内の自動化ライブラリをご利用いただけます。こちらでは、登録されているルールをクリック1つでご利用の製品内で有効化していただけます。

少しずつイメージが湧いてきたでしょうか？次に、ここまでの章でご紹介したそれぞれのITSMソリューションで自動化を活用するユースケースをご紹介します。

3 ITSMソリューションと自動化

Jira Service Managementで自動化機能を利用することで、ソリューションを大きく強化できます。たとえばリクエスト管理プラクティスでは次のようなルールを利用できます。

- チケットへの回答がないまま5日間が経過したチケットを自動クローズ

各ソリューションでの自動化の活用例をご紹介します。

■ インシデント管理ソリューション

インシデント管理ソリューションでは次のような自動化が考えられます。

- インシデントの登録時に、影響を受けるサービスに応じてインシデント対応ランブックの情報を動的に追加
- インシデントの登録時に、影響を受けるサービスの過去のインシデントをリンク
- インシデントチケットのクローズ時に、インシデントチケットの内容がコピーされたインシデント事後レビュー(PIR)用チケットを作成

■ 変更実現ソリューション

変更実現の章でご紹介した、変更対象のサービスやデプロイ先に応じた承認有無の処理も、自動化の機能を利用して行われています。[プロジェクト設定]>[自動化]でデフォルトの自動化ルールをぜひご確認ください。



また、アセット機能をサービスや社内のプラクティス情報のデータベースとして利用することで、次のような高度な自動化も行えます。

アセット機能に格納する情報	自動化ルール	
	トリガー	アクション
プラクティス情報	標準変更チケットの作成	<ol style="list-style-type: none"> 対象のアプリケーションの事前承認者をアセット機能から取得 事前承認者の名前を変更チケットに追加し、アセット機能から取得した、導入/バックアウト/テスト計画を変更チケットに入力 標準変更において必要な作業項目をアセット機能から取得し、それぞれについて作業用のチケットを作成
プラクティス情報、サービス情報	営業時間面/ロールバック/テスト面のリスクに対応するフィールドを持つ変更申請の起票	<ol style="list-style-type: none"> それぞれのリスクについて、入力値を元に対応するスコアをアセット機能から取得 変更対象のサービスのスコアをアセット機能から取得 次の計算を実施 <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;"> $\text{営業時間面のリスクスコア} + \text{ロールバック面のリスクスコア} + \text{テスト面のリスクスコア} + \text{変更対象サービスのスコア} = \text{変更リスクの値}$ </div> 変更リスクが10未満の場合は承認をスキップ、10以上の場合は承認待ち <div style="border: 1px solid gray; padding: 10px; margin-top: 10px;"> <p style="text-align: center;">POINT</p> <p>自動化では簡単な変数の定義/利用や演算も行えます。</p> </div>
サービス情報	チケットの「影響を受けるアプリケーション」フィールドの更新	<ol style="list-style-type: none"> 対象のアプリケーションのサポート担当チームをアセット機能から取得 チケットの「サポート担当チーム」フィールドを1で更新
サービス情報	インシデントチケットの起票	<ol style="list-style-type: none"> 対象のサービスの階層をアセット機能から取得 階層に応じて影響と緊急度を指定 チケットのそれぞれのフィールドを1で更新

■ アセット管理ソリューション

アセット管理ソリューションでは、アセット機能との連携を通じて次のようなことを行えます。

アセット機能に格納する情報	自動化ルール	
	トリガー	アクション
施設情報	特定の部屋に割り当てられた備品について問い合わせ	<ol style="list-style-type: none"> 1. 部屋に紐付けられた備品一覧をアセット機能から作成 2. コメントとして追加 3. チケットをクローズ
施設情報	別のオフィスでの勤務申請	<ol style="list-style-type: none"> 1. 対象のオフィスの管理者をアセット機能から取得 2. 1をチケットの承認者に設定
ITアセット情報	対象のオブジェクトスキーマの更新	<ol style="list-style-type: none"> 1. アセット機能で在庫数を確認 2. 基準値よりも少なければ発注用のチケットを作成
ITアセット情報	毎月1日の9時	<ol style="list-style-type: none"> 1. アセット機能で保証期限を確認 2. その月に保証が切れるアセットの更新用チケットを作成
ITアセット情報	ノートPCの割り当て申請	<ol style="list-style-type: none"> 1. アセット機能で未割り当てのノートPC一覧を検索 2. エージェントが参照できるよう、1の一覧をチケットに追加
契約済みのソフトウェア情報	新入社員用のPCのセットアップ申請	<ol style="list-style-type: none"> 1. デフォルトで提供可能なソフトウェア一覧をアセット機能から取得 2. 2を提示 3. 選択内容に応じてソフトウェアアクセスのセットアップ用の子チケットを作成
従業員情報	新規ハードウェア申請	<ol style="list-style-type: none"> 1. 報告者の上司をアセット機能から取得 2. 1を承認者に設定
従業員情報	退職情報の共有	<ol style="list-style-type: none"> 1. 退職者のオブジェクトをアセット機能で更新 2. アセット機能でのオブジェクトのステータスを「退職済み」に更新
従業員情報	チケットの起票	<ol style="list-style-type: none"> 1. 報告者の職位をアセット機能から取得 2. 課長以上の役職であればチケットに「優先」ラベルを追加

あるいは、アセット機能を利用しない次のようなシンプルな自動化も考えられます。

- 新規ハードウェアの購入申請で、エージェントがチケット内のボタンを押し、チケットの内容がコピーされた、調達チーム用のチケットを作成
- 新入社員のセットアップ申請の起票時に、入室バッジの必要有無についての選択内容に応じ、施設チーム用にバッジ申請チケットを作成

おわりに

全6章にわたり、アトラシアン のITSMソリューションの概要、リクエスト/インシデント/変更/アセット管理プラクティスの実現方法、Jira Service Managementのカスタマイズ方法についてご説明しました。

Jira Service Managementの強みは、固定されたソリューションをそのまま提供するのではなく、ユーザーが自社の要件に合わせてカスタマイズできる柔軟性を持つ点にあります。ぜひ、お客様それぞれに最適な方法に合わせて製品をご活用ください。





アトラシアン株式会社

〒220-8114 神奈川県横浜市西区みなとみらい 2丁目2-1 ランドマークタワー1407

導入前のご相談・
お問い合わせ

045-330-9081

受付時間 9:00-18:00 月曜日～金曜日(祝日・弊社休業日を除く)

メールによるお問い合わせ

japan-sales@atlassian.com

お問い合わせフォーム

<https://www.atlassian.com/company/contact/japan>