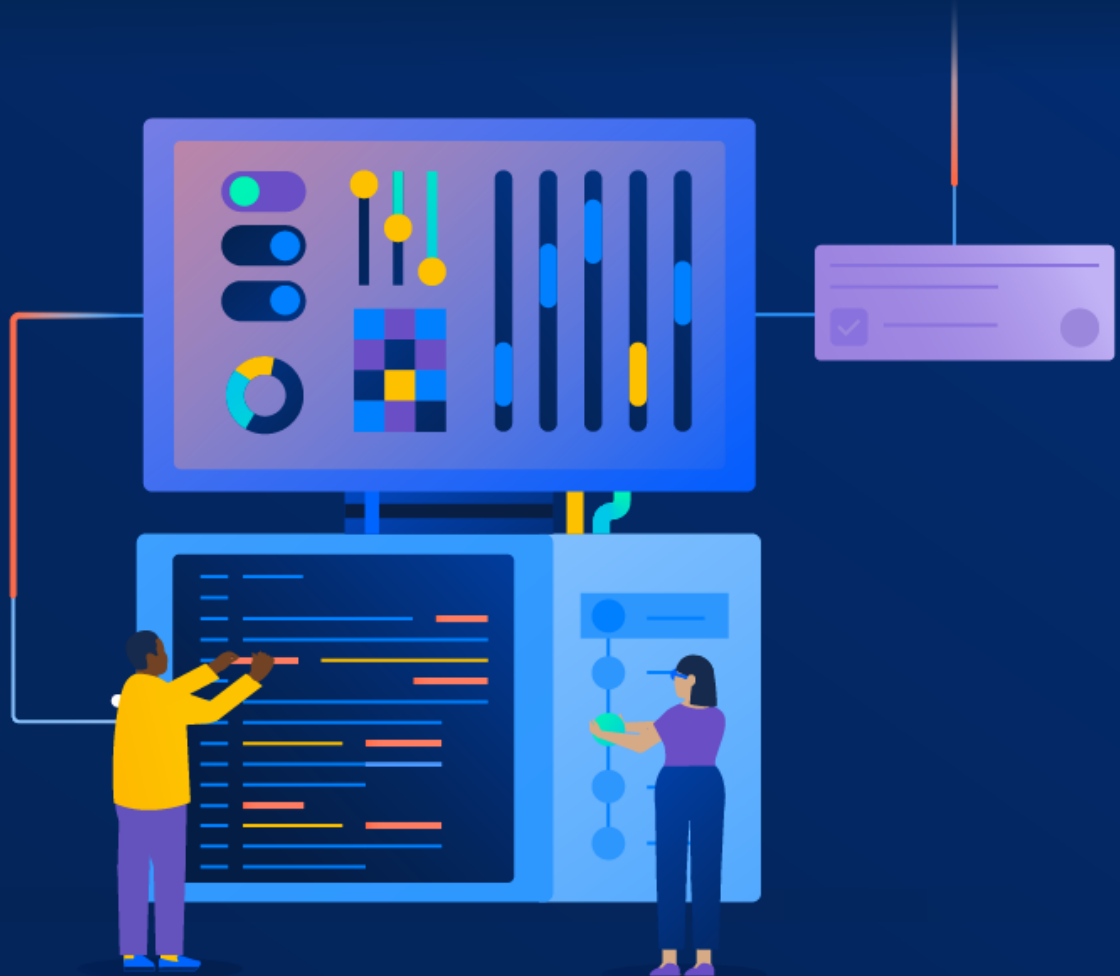


# 기존 ITSM 업그레이드의 통점 해결

제품 소유자는 기존 ITSM 시스템 업그레이드의  
대안에 관해 직접 알게 됩니다

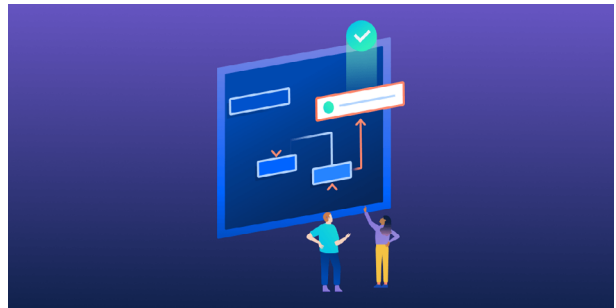


# 목차

요약	2
기존 ITSM 시스템의 업그레이드에 따른 문제점 해결 방안 모색 - ITSM 시스템에 대한 더 나은 접근법	4
ITSM 시스템 업그레이드에 대한 더 건전한 접근법	5
ITSM 시스템 업그레이드에 대한 더 건전한 접근법	7

## 요약

여기서는 제품 소유자가 기존의 ITSM(IT 서비스 관리) 시스템을 업그레이드하는 과정에서 직면하는 여러 가지 해결 과제와 ITSM 시스템 업그레이드를 위한 보다 유연한 솔루션으로서 Atlassian의 Jira Service Management를 탐색하는 과정에 대해 살펴봅니다.



 계속해서 읽으면서 다음 사항을 알아보세요.

- 1 제품 소유자가 기존 ITSM 시스템을 업그레이드하면서 어떻게 심각한 과제에 직면했는가. 업그레이드가 잦은 중단과 낮은 최종 사용자 인식 가치로 인해 복잡하고 시간 소모적인 과정이었던 이유를 알아보십시오.
- 2 많은 조직들이 ITSM 시스템 업그레이드 과정에서 이와 유사한 어려움을 겪으면서, 일부는 이러한 문제를 완화하기 위해 레거시 시스템을 ‘재부팅’하는 선택을 하는 이유를 들어보세요.
- 3 제품 소유자가 Atlassian의 Jira Service Management를 선택하게 된 이유를 알아보십시오. 처음에는 회의적이었으나 이 새로운 솔루션이 간편한 구성과 바로 사용할 수 있는 ITIL 프랙티스 등의 요소가 있어 ITSM 시스템 업그레이드에 있어 더 유연하고 효율적인 솔루션이라는 것을 알게됩니다.
- 4 Jira Service Management 업그레이드에 대한 Atlassian의 접근법에 따른 주요 이점 중 하나가 이 업그레이드는 애자일 방법론을 따르기 때문에 레거시 ITSM 벤더의 폭포 접근법에 비해 더 유연하고, 협업이 가능하며, 효율성이 높다는 점인지 알아보세요.
- 5 Jira Service Management가 어떻게 잦은 기능 릴리스, 효율적인 코드 전달/업그레이드를 위한 느슨하게 연결된 아키텍처, 손쉽게 확장되는 진정한 클라우드 제품을 통해 지속적인 향상을 지원하고 보다 강력한 보안과 컴플라이언스를 제공하는지 알아보세요. 이러한 접근법은 디지털 변혁 노력에서 필요한 보다 애자일한 무중단 페이스에 대한 요구에 부합됩니다.

## 제품 소유자는 기존 ITSM 시스템 업그레이드의 대안이 있다는 것에 관해 직접 알게 됩니다

저는 레거시 ITSM 시스템을 사용해 여러 서비스 관리 프랙티스용으로 기능을 개발한 스크럼 팀의 제품 소유자였습니다. 이번 역년도 마지막 스프린트가 마무리되는 시점에, 우리 팀은 새로운 기능을 배우고 실험할 수 있었던 혁신 스프린트를 즐길 수 있었습니다. 저는 이번 혁신

스프린트를 이용해 우리 ITSM 시스템의 새로운 릴리스를 살펴보았습니다. 첫눈에 봐도 새로운 기능이 정말 많이 추가된 것으로 보였습니다. 그런데 자세히 읽어보니 새로운 기능이 우리 이해관계자들에게 유용하기는 커녕 오히려 기술 부채를 늘리는 결과를 가져올 것이라는 생각이 들었습니다.

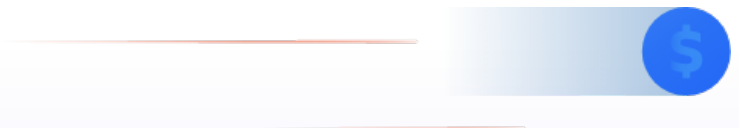
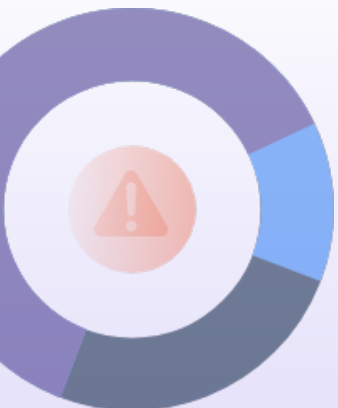
“

저는 Jira Service Management에서 ‘애셋’ 기능을 한동안 이리저리 살펴보았는데, 구성이나 데이터 로드하기가 쉽다는 생각을 했습니다.

그런 다음 프로젝트와 서비스 할당 그룹을 만들기 시작했습니다. 구성을 시작한 지 몇 시간 만에 (그리고 문서도 일체 보지 않고), IT 애셋과 CI, 주요 인시던트 기능을 포함하는, 실제 작동되는 서비스 카탈로그와 서비스 데스크를 개발할 수 있었습니다.

프로세스 소유자와 그 외 이해관계자들과 수많은 회의를 할 생각을 하니 벌써 머리가 지끈거리기 시작했습니다. 새로운 릴리스의 내용을 설명해야 하고, 개발 일정이 중단되고, 그리고 이렇게 추가적인 업무를 하고도 결과적으로 유용한 영향은 적을 것이라는

생각만으로도 벌써 심한 스트레스가 느껴졌습니다. 얼마나 많은 기능을 사용하게 되든, 업그레이드 시즌의 지루한 프로세스와 생산성 손실은 항상 똑같았습니다.



# 기존 ITSM 시스템 업그레이드에 따른 어려움

여기서는 제품 소유자가 기존의 ITSM(IT 서비스 관리) 시스템을 업그레이드하는 과정에서 직면하는 여러 가지 해결 과제와 ITSM 시스템 업그레이드를 위한 보다 유연한 솔루션으로서 Atlassian의 Jira Service Management를 탐색하는 과정에 대해 살펴봅니다.



## 요약:

- 1 소프트웨어 업그레이드는 수많은 복잡한 단계를 거쳐야 하고 그 결과 최종 사용자 입장에서는 사용 중단과 답답한 상황을 초래하는 힘든 과정이 될 수 있습니다.
- 2 많은 조직들이 사용을 중단해야 하기 때문에 생산성에 영향을 미치는 장시간에 걸친 업그레이드를 겪고 있기 때문에 소프트웨어 업그레이드에 따른 해결 과제는 단발적인 문제가 아닙니다.
- 3 일부 조직의 경우 심지어 레거시 시스템을 '재부팅'해 소프트웨어 업그레이드와 관련된 어려움을 완화하는 것도 고려하고 있습니다.

이 문제를 직접 겪어보지 않는 사람들은 제 설명이 과장된 얘기라고 생각할 수 있으니 아래에 나와 있는 우리 팀의 업그레이드 과정을 요약한 것을 한 번 살펴보시기 바랍니다. 많은 작업이 진행되는데, 저는 릴리스 전부터 릴리스 후까지 3개의 대단계, 11개의 넓은 소단계로 정리했습니다.

<p><b>릴리스 전</b></p>	<ul style="list-style-type: none"> <li>• 제품 소유자는 릴리스 노트를 검토하고 새로운 기능 및 픽스를 확인합니다.</li> <li>• 개발(Dev) 팀은 샌드박스에 릴리스를 설치해 기능을 사용하고 문서화합니다.</li> <li>• 제품 소유자는 프로세스 팀과 만나 유용한 기능에 대해 합의합니다.</li> </ul>
<p><b>릴리스 프로세스</b></p>	<ul style="list-style-type: none"> <li>• 옴스(Ops) 팀이 업그레이드에 대비해 모든 환경을 복제합니다(새로운 개발 작업에 대해 코드 동결 적용).</li> <li>• 개발 팀이 릴리스 코드를 패키징합니다. 개발 환경에 릴리스를 설치하고, 상층 코드를 비활성화하며, 예비 코드 테스트를 실시합니다. 이어서 릴리스를 QA로 마이그레이션합니다.</li> <li>• QA 팀은 테스트 프레임워크를 사용해 기능을 테스트합니다. 이 테스트는 구체적인 기능에 대한 테스트이지 엔드투엔트 프로세스 테스트가 아닙니다. QA 테스트가 성공적으로 완료되면 개발 팀에서 코드를 UAT로 마이그레이션합니다.</li> <li>• 제품 소유자와 프로세스 팀은 UAT 환경에서 엔드투엔트 프로세스 테스트를 실시합니다. UAT 테스트가 성공적으로 완료된 후, 개발 팀은 제품에 대한 릴리스 절차를 문서화합니다.</li> <li>• 개발 팀은 다른 스크럼 팀들과 조율해 프리프로덕션 환경에 코드를 설치하고 필요한 업데이트를 실시합니다. QA 팀은 프리프로덕션 환경 테스트를 실시합니다.</li> <li>• 옴스 팀은 프로덕션에 코드를 설치합니다. AQ 팀과 프로세스 팀은 프로덕션 테스트를 실시합니다.</li> </ul>
<p><b>릴리스 후</b></p>	<ul style="list-style-type: none"> <li>• 개발 팀은 새로 발견된 포스트프로덕션 버그가 있으면 해결합니다.</li> <li>• 개발 팀은 테스트를 설치하고 필요한 핫픽스를 배포합니다.</li> </ul>

이걸 보는 것만으로도 벌써 지칩니다. 게다가 전체 작업을 모두 담은 것이 아닙니다. 업그레이드를 관리하는 팀들은 조직 변경 관리 문제를 해결하기 위해 막대한 시간과 노력을 들입니다. 최종 사용자들은 자신들이 보기에 그다지 가치도 없는 것을 위해 서비스 배포가 연 2회 중단되어야 한다는 사실에 답답하고 화가 납니다.

저는 이런 업그레이드 문제를 우리 팀만 겪는 문제라고 생각했었는데, 사용자 그룹 회의에 참석하고 여러 블로그를 읽어보고 난 후 이런 힘든 업그레이드 프로세스가 아주 흔한 문제라는 것을 알게 되었습니다.

일부 고객들은 업그레이드하는 데 8주 넘게 걸린다고 불만을 토로하고, 조직의 생산성이 타격을 받는다고 불평합니다.

경우에 따라 업그레이드 문제를 완화하기 위해 레거시 ITSM 시스템을 ‘재부팅’하려는 곳도 있을 정도입니다. 예를 들어, 고객들은 아직도 2022년 3월 샌디에이고 릴리스에 포함되었던 ServiceNow의 Next Experience UI를 사용하는 방법에 대해 고심 중입니다. 일부 기업 고객들의 경우 사용자에 대한 영향이 판단될 때까지 기능을 꺼 두는 경우도 있습니다.

## 해결 방안 모색 - ITSM 시스템에 대한 더 나은 접근법

### 요약:

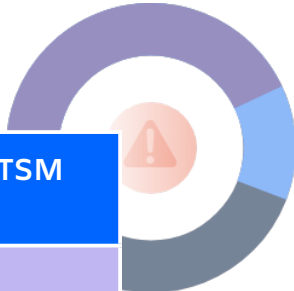
- 1 ITSM 문제를 계속 견디는 것과 보다 활용성이 뛰어난 솔루션을 모색하는 것 중 하나를 고르는 선택에 직면했을 때는 Atlassian의 Jira Service Management를 고려해볼 만한 가치가 있습니다.
- 2 Jira Service Management의 놀라운 점은 구성하기 쉽고 기능 서비스 카탈로그와 서비스 데스크를 신속하게 생성할 수 있다는 것입니다.
- 3 Jira Service Management에 대한 Atlassian의 접근법은 ‘애자일’로 유연성과 지속적인 기능 릴리스를 제공하는 반면, 레거시 ITSM 벤더들의 경우 매년 업데이트를 해야 하고 뻘뻘하게 연결된 아키텍처에 기반한 폭포식 접근법을 사용하는 경우가 많습니다.

제가 보기에는 두 가지 옵션이 있었습니다. 기존 ITSM 시스템에 내재적인 문제점을 받아들이고 책상 서랍에 제산제와 이부프로펜 진통제를 넉넉하게 준비해 두거나, 아니면 보다 활용도 높고 유연한 솔루션을 찾아보는 것이었습니다. 답답한 마음에 저는 Google에 가서 애널리스트 보고서도 읽어보고, ITSM 스페이스에 있는 친구들과 상의도 해봤는데, 결론은 Atlassian의 ITSM 솔루션인 Jira Service Management였습니다.

솔직히 이 솔루션이 우리 문제를 해결해줄 수 있을 거라는 데 회의적이었습니다. 저는 엔터프라이즈 ITSM 시스템 분야에서 일한 지 꽤 됐고 Jira Software로 일한 지도 몇 년 되었습니다. 그래서 Jira Service Management가 우리 문제를 해결해 주지 못할 거라고 확신했습니다.

저는 Jira Service Management에서 ‘애셋’ 기능을 한동안 이리저리 살펴보았는데, 구성이나 데이터 로드하기가 쉽다는 생각을 했습니다. 그런 다음 프로젝트와 서비스 할당 그룹을 만들기 시작했습니다. 구성을 시작한 지 몇 시간 만에 (그리고 문서도 일체 보지 않고), IT 애셋과 CI, 주요 인시던트 기능을 포함하는, 실제 작동되는 서비스 카탈로그와 서비스 데스크를 개발할 수 있었습니다.

Jira Service Management는 “지금 있는 곳에서 시작” 같은 애자일 방법론의 요소를 채택하고 있으며, 바로 사용할 수 있는 ITIL 프랙티스를 제공합니다. Atlassian은 도구 중심적 솔루션 대신 팀 중심적 접근법을 선택한 것이 확실했습니다.



## Jira Service Management 업그레이드에 대한 Atlassian의 접근법

## 시스템 업그레이드에 대한 레거시 ITSM 벤더의 접근법

### 애자일 vs. 폭포 방식

Atlassian은 Jira Service Management 업그레이드에 대해 유연성과 협업, 효율성을 최적화하는 애자일 방법론을 따릅니다.

레거시 ITSM 벤더는 업그레이드에 대해 구조화된 연속적 릴리스에 중점을 두는 폭포식 접근법을 사용합니다.

### 일정 변경 vs. 필수 연간 업데이트

Jira Service Management 고객은 환경에 적절한 릴리스 트랙을 설정함으로써 변경 속도를 조절할 수 있습니다. 연속적-제품이 변경 사항과 기능이 출시되는 대로 바로 받게 됩니다. 번들식-제품이 매달 두 번째 화요일에 그룹 단위로 변경 사항을 받습니다.

레거시 ITSM 벤더들은 보통 연 2회 새로운 기능이 담긴 소프트웨어 릴리스를 제공합니다.

### 지속적인 기능 릴리스 vs. 그때그때 버그 픽스

Atlassian의 접근법은 더 짧고 예측 가능한 릴리스 주기, 더 작고 높은 품질의 코드 세트, 보다 투명하고 응답성이 뛰어난 기능 전달을 통해 고객에게 향상된 비즈니스 가치를 제공합니다.

레거시 ITSM 벤더는 업그레이드에 대해 구조화된 연속적 릴리스에 중점을 두는 폭포식 접근법을 사용합니다.

### 느슨하게 연결된 아키텍처 vs. 뻘뻘하게 연결된 코드 세트

Jira Service Management는 개별적 구성 요소가 상호 독립적으로 구축되어 있는 느슨하게 연결된 아키텍처를 기반으로 합니다. 팀은 느슨하게 연결된 애플리케이션을 이용해 독립적으로 배포 및 확장되는 기능을 개발할 수 있으며, 이는 보다 효율적인 코드 전달/업그레이드를 제공합니다.

레거시 ITSM 시스템 아키텍처는 뻘뻘하게 연결되어 있어, 기능이 경직되어 있고 상호 의존적입니다. 구성 요소 하나가 바뀌면 전체 애플리케이션에 '도미노' 효과를 일으켜 더 많은 변경이 필요하기 때문에 전체적인 코드 변동성과 리스크가 커집니다.

### 진정한 클라우드 제품 vs. 사일로형 SaaS 제품

Jira Service Management는 Atlassian 클라우드 플랫폼에 맞게 설계 및 구축되었기 때문에 고객들은 임베디드 보안, 빌트인 컴플라이언스, 무제한 인스턴스, 간소화된 사용자별 라이선스 부여 등을 이용해 자신있게 확장할 수 있습니다.

레거시 ITSM 시스템은 온-프레미스로 구동되도록 설계하는 경우가 많으며, 벤더는 단순히 원격으로 애플리케이션을 호스트합니다. 이는 진정한 클라우드 솔루션이 아니며, 진정한 클라우드 시스템과 동일한 이점(확장, 업그레이드, 가격 및 보안)을 제공하지 못합니다.



# ITSM 시스템 업그레이드에 대한 보다 건전한 접근법

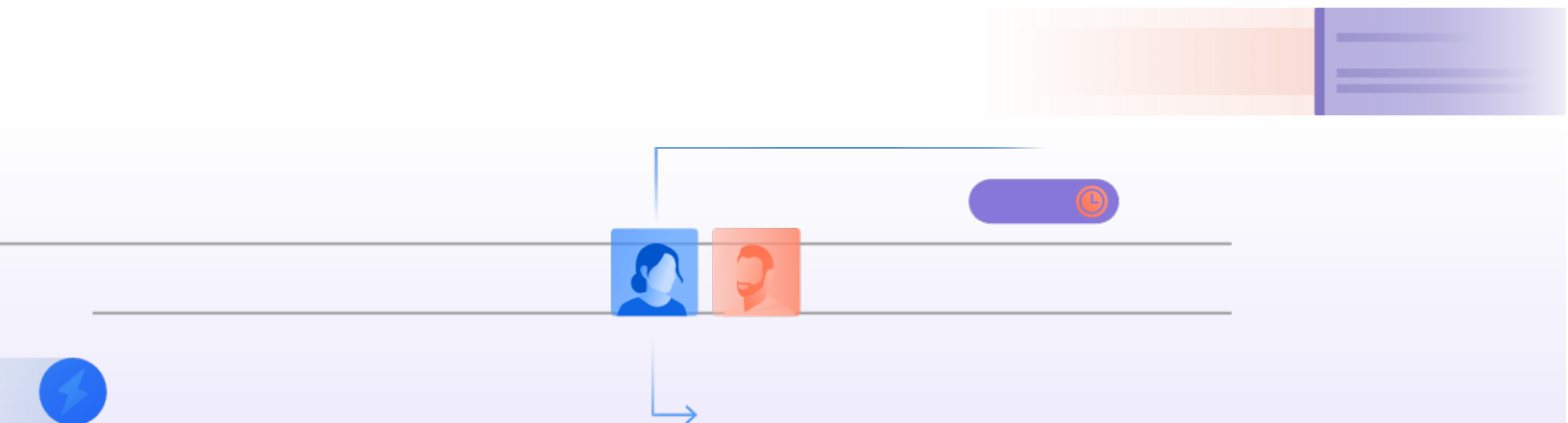
업그레이드의 문제를 접근하면서 Atlassian은 기술 진화를 계획함으로써 새로운 기능이 나왔을 때 플랫폼과 제품이 그 기능을 매끄럽게 도입할 수 있도록 합니다. Atlassian의 클라우드 로드맵은 공개적이기 때문에 고객들은 항상 다음에 무엇이 나올지 알 수 있습니다. Jira Service Management의 클라우드 기반 시스템은 고객들이 연속적으로, 또는 정해진 일정에 따라 사용할 수 있도록 기능을 출시합니다.



## 고객은:

- 1 주간 릴리스 노트를 구독해 예정 변경 사항에 대한 최신 노트를 받아볼 수 있습니다
- 2 독립된 환경(즉 샌드박스)을 만들어 프로덕션으로 도입하기 전에 기능을 실험해볼 수 있습니다
- 3 프로덕션 환경의 릴리스 트랙을 (연속적과 번들식) 설정해 변경 속도를 조절할 수 있습니다
- 4 연속적인 향상을 제공하는 소규모의 점증적 업데이트의 이점을 활용할 수 있습니다

Jira Service Management는 느슨하게 연결된 코드 세트를 기반으로 하기 때문에 보다 우수한 코드 유연성/재사용성과 보다 효율적인 애플리케이션 업그레이드를 제공합니다.



# Atlassian 플랫폼 기반

Jira Service Management는 기업 전체의 팀, 정보 및 워크플로 사이의 개방적이고 효율적인 협업을 촉진하는 Atlassian의 통합된 클라우드 기술 플랫폼을 기반으로 구축되었습니다.



## 애널리틱스

의사결정 속도를 높이고 IT, 개발 및 외부 데이터 소스 전반에 걸친 인사이트를 확보합니다. 로우 코드/무코드 도구와 템플릿 대시보드를 사용해 데이터를 시각화합니다.



## 자동화

수동 작업할 필요 없이 강력한 제품 교차적 자동화를 구현합니다. 규칙 기반 워크플로 자동화를 통해 지원과 인시던트, 운영 프로세스를 손쉽게 조율합니다.



## 협업

개인이 아닌 팀 전체에 걸쳐 개방적인 협업과 표면 작업을 지원합니다. 스마트 링크와 예측형 제품 교차적 검색, 멘션 및 반응을 통한 가시성을 제공합니다.



## 관리

관리자는 Atlassian 제품 포트폴리오 전반에 걸쳐 중앙 집중화된 미션 컨트롤에 액세스함으로써 가시성, 제어 능력 및 편의성을 위한 고급 기능을 사용할 수 있습니다.



## 확장성

마켓플레이스 앱, 당사가 호스트하는 앱 개발 플랫폼 및 통합을 이용해 새로운 기능을 추가, 결합, 사용자 지정 및 구축함으로써 기업의 진화하는 요구 사항을 충족할 수 있습니다.



## 데이터 관리

다양한 데이터 거버넌스 기능을 이용해 Atlassian 제품군 전체에 걸쳐 데이터에 대한 통제력을 유지함으로써 아무리 엄격한 보안 및 컴플라이언스 요건도 충족할 수 있습니다.



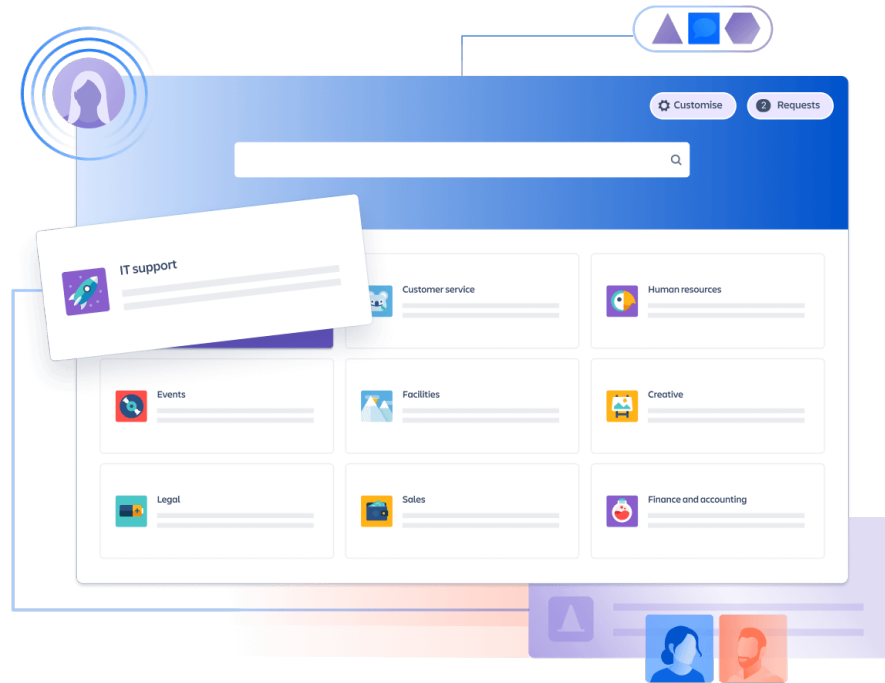
## 기본 시설

규모, 성능 및 안정성에 기반한 세계적 수준의 Atlassian의 클라우드 기반 시설 위에서 서비스 관리를 자신 있게 확장하세요.



우리는 모두 조직이 “더 열심히가 아니라 더 스마트하게 일”할 수 있도록 디지털 변혁을 가속화하고자 합니다. 저는 Jira Service Management의 애자일한 점증적 변경과 개선 업그레이드 접근법을 통해 우리가 계속해서 릴리스를 검증하고 최종 사용자 요구에 계속 부응할 수 있다는 것을 알게 되었습니다.

디지털 변혁 여정을 시작하려 하신다면 총 소유 비용(TCO)뿐 아니라 각 애플리케이션의 핵심 아키텍처와 구현 접근법까지 검토해 보는 것을 추천합니다. 달성하고자 하는 비즈니스 이니셔티브를 고려해 본 다음 요구 사항을 충족하고 모든 고객 상호작용에 가치를 더해주는 시스템을 선택하시기 바랍니다.



**Atlassian 솔루션 파트너에게 연락해 어떻게 Atlassian 솔루션 파트너와의 협업을 통해 서비스 관리에 대한 접근법을 새롭게 바꿀 수 있는지 알아보세요.**