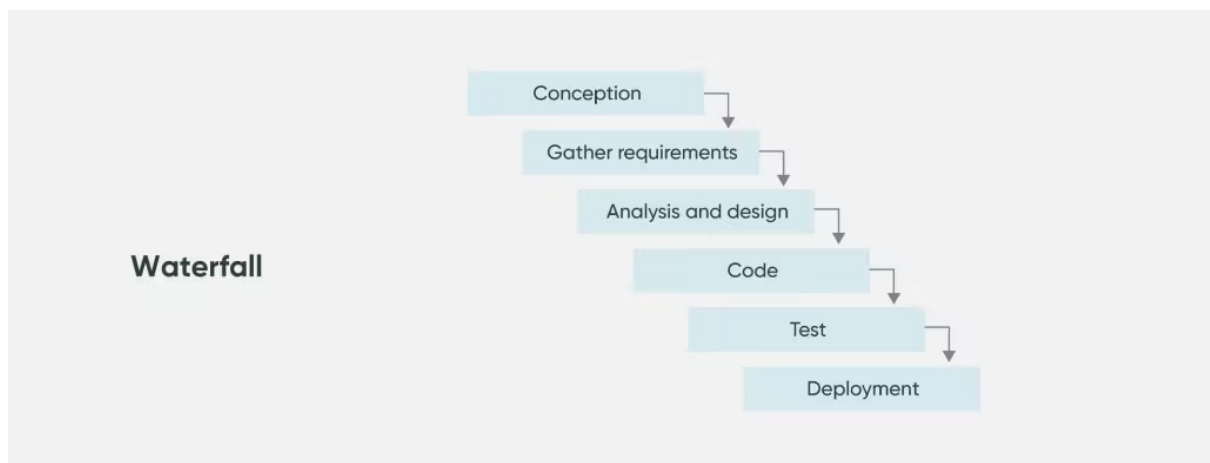




개발 프로세스 소프트웨어 개발 방법론

1. 개발 방법론 - waterfall / Agile
2. Agile을 돕는 협업툴 - Jira / Confluence

개발방법론 1. Waterfall



✓ Waterfall은 개발에 대한 기존 접근 방식인 **순차적 선형 방법론**이며 **초기에 등장한 개발 방법론** (1970)으로 소프트웨어 개발 수명주기를 개별 단계로 나눠 이전 단계가 완료된 경우에만 다음 단계로 진행할 수 있음.

✓ 특징 : 사용과 이해가 용이하며 작업, 연구, 문서화, 계획을 가장 중시

- '수행 전 다시 한번 더 확인'하는 접근 방식을 취해 프로젝트 초기에 모든 요구 사항을 정의하고 이를 충족하기 위한 세부 계획을 수립함.

✓ Waterfall의 단계

Waterfall은 프로젝트를 **7가지 개별 단계로** 구분되며, 각 단계들은 서로 **독립적**이고, **이전 단계를 완료한 후**에만 새로운 단계를 시작할 수 있다. 대부분의 단계는 '관문'으로 구분되어

일련의 요구 사항을 충족하고 관리 결정을 내려야만 프로젝트를 다음과 같은 단계로 진행할 수 있다.

- **구상개발** 팀이 먼저 이윤과 잠재적 비용을 비롯해 예정된 프로젝트를 평가.
- **문서화** 시스템 및 소프트웨어 요구 사항과 더불어 프로젝트에 필요한 기타 자원을 수집하고 문서화.
- **분석 및 디자인** 팀이 프로젝트를 분석하고 제품 또는 서비스의 기능을 결정합니다. 이 단계에서 필수 작업을 식별하고 이에 대한 계획을 수립.
- **코드 및 유닛 테스트** 소프트웨어 각 단위의 코딩이 시작되는 동시에 테스트도 진행됩니다. 유닛들은 초기 단계에서 정의된 소프트웨어 아키텍처에 통합.
- **전체 시스템의 테스트** 시스템 전반에 걸친 테스트가 진행됩니다. 여기에는 버그 테스트, UAT(사용자 수용 테스트)와 더불어 기타 필수 테스트가 포함.
- **문제 해결** 이전 단계에서 식별된 버그, 비효율성 및 문제를 해결하고 조치.
- **제공** 완성된 제품 또는 서비스를 최종 사용자에게 제공.

✓ Waterfall의 이점

Waterfall 방법론은 약 반세기 동안 다양한 개발 팀 간에 지속적으로 사용되어 왔으며, 9가지 이점을 가지고 있다.

1. 간단한 계획 및 설계

Waterfall 방법론은 단계마다 특정된 결과물과 분명한 검토 프로세스가 존재하기 때문에 관리 측면에서 가장 사용하기 쉽다.

2. 시스템 전체 접근 방식을 통한 개선된 설계

외부 시스템과의 통합을 허용하도록 다수의 구성요소를 설계해야 하는 프로젝트에서는 프로세스 초기에 설계를 완료하는 Waterfall 접근 방식이 분명한 장점으로 작용한다.

3. 명확히 정의된 작업 범위

제품 요구 사항을 개발 시작 전에 문서화하고 합의하므로 예측 가능하고 구체적인 기능을 수립할 수 있다.

4. 보다 정확한 비용 추정

상세한 계획 수립과 사전 문서화로 잠재적 비용을 분명하게 파악하여 정확하게 예산을 편성할 수 있다.

5. 명확한 진행 상황 측정

작업 범위를 사전에 정립하므로 진행 상황을 간단하고 정확하게 측정할 수 있으며 진행 상황은 일반적으로 현황 보고서를 통해 일정, 예산 및 자원에 따라 녹색, 황색, 적색으로 표시된 작업 항목으로 파악할 수 있다.

6. 정의된 팀 역할

변화하는 요구 사항에 따라 역할을 유동적으로 변경하지 않고 개발 작업에 착수하기 전에 목표를 파악하고 수립한다.

7. 업무 분담

팀원은 정해진 단계에만 작업을 수행하므로 다른 프로젝트에도 역량을 할애할 수 있다.

8. 고객으로부터 프로젝트 독립성 확보

Waterfall 방법론은 보다 수월하고 *접촉이 필요 없는* 고객 경험을 제공합니다. 최종 사용자는 *요구 사항* 및 *검토* 단계 외에는 개입할 필요가 없다.

9. 포괄적 문서화

계획 수립과 문서화에 중점을 두기 때문에, 프로젝트를 정해진 경로를 따라 진행할 수 있으며, 검토가 쉽고, 결과도 분명하게 식별할 수 있다.

✓ Waterfall의 단점

Agile이라는 개발 프로세서의 등장으로 기존 Waterfall 방법론의 단점이 나타남.

1. 경직된 구조 사용

Waterfall은 초기 단계에 수립한 세부 계획에 의존하기 때문에 프로젝트가 예기치 못한 문제나 장애물을 만나거나 요구 사항이 변경되는 경우 대응할 수 없다. 또한 한 방향으로만 진행되기 때문에 이전 단계로 돌아가 수정하는 것이 매우 어렵다.

2. 불확실성을 허용하지 않음

확고하게 정의된 요구 사항으로 인해 새로운 아이디어, 혁신, 창의성을 발휘하기가 어렵고, 개발자는 개발 중 발견한 뜻밖의 기회를 활용하지 못할 수도 있다.

3. 고객 불만 발생 가능성

고객이 개발 프로세스에 적게 개입하기 때문에 프로세스와 동떨어진 느낌을 받을 수 있으며 프로젝트가 완료되기 전까지 결과물이 어떤지 알 수 없다.

한편 개발자는 고객이 어떤 결과를 기대하는지 파악하지 못하기 때문에 고객의 기대치와 개발 팀 작업의 간극이 더 커지게 되며, 프로젝트가 끝나고 나서야 테스트가 진행되므로 버그나 UX 문제를 놓칠 가능성이 높다.

4. 끝날 때까지 연기되는 테스트

특정 단계의 작업 기한이 불분명하면 프로젝트 전체의 일정이 지연될 가능성이 있다. 이러한 경우 팀은 때로는 마지막 단계까지 서두르게 되며, 이 과정에서 테스트 또한 소홀하여 제품의 품질이 떨어질 수 있다.

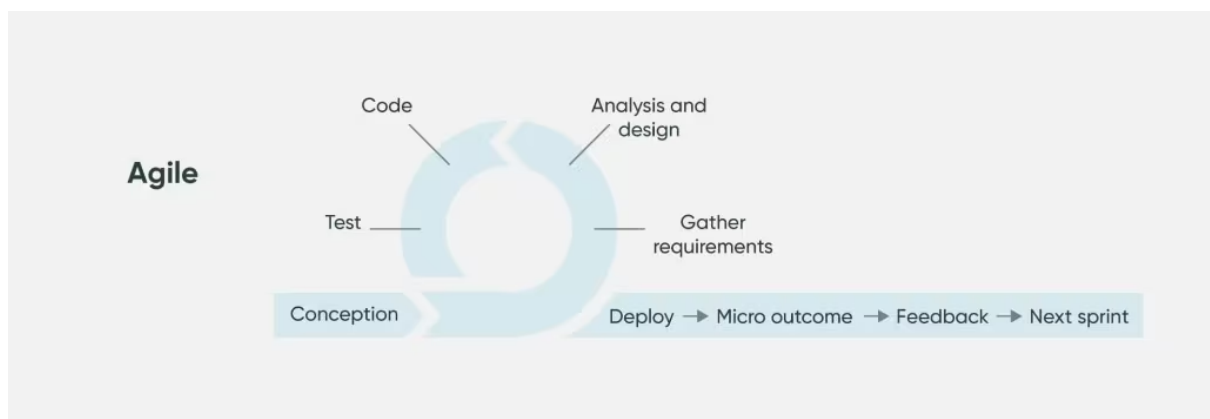
5. 초기 단계에 명확한 요구 사항 수립 필요

작업에 착수하기 전에 요구 사항을 분명하게 식별하고 승인해야 한다. 이것이 선행되지 않는 경우, 팀원이 각자 해석을 달리하여 구성원 간 단절이 발생할 수 있다.

6. 구축 대신 문서화에 더 많은 자원 할당

계획 및 문서화에 많은 자원이 투입되어 실제 제품 개발에 투입할 자원이 부족해질 수 있다.

개발방법론2. Agile 방법론



Agile은 (2000년대 초 등장) 팀 간의 공동 작업을 중심으로 하는 반복 가능한 소프트웨어 개발 방식으로 지속적 제공과 지속적 개선에 있어 중요한 역할을 한다.

- 지속적 제공 : 지속적 제공은 전체 소프트웨어 부분이 즉시 배포할 수 있는 형식으로 짧은 주기마다 제공되는 소프트웨어 개발 방식.
- 지속적 개선: 지속적 개선은 제품, 서비스, 프로세스를 지속적으로 재검토하고 업그레이드하여 비즈니스 결과를 개선하는 방법.

✓ Agile의 핵심 원칙

- **적응성Agile**
 - 프로젝트는 아키텍처, 디자인, 결과물, 요구 사항 및 기타 요소를 프로젝트 중간에 변경할 수 있는 유연성이 있어야 한다.
- **린(Lean) 개발Agile**
 - 개발에 대한 가장 단순한 접근 방식을 취하므로 불필요하거나 중복되는 단계를 제거한다.
- **팀워크Agile**
 - 효과적인 팀워크와 커뮤니케이션에 의존하므로 한 번에 여러 작업이 완료될 수 있다.
- **고객 참여Agile**
 - 반복 작업은 점진적으로 가치를 제공하므로 새로운 아이디어를 도입하고 제품을 수정할 때 고객과 협업할 수 있다.
- **지속 가능성Agile**
 - 팀이 성과에 주력하도록 압박하는 대신, 지속 가능하고 일관된 개발 속도를 자아내 결과를 토대로 고객 가치를 제공한다.
- **시간Agile**
 - 프로젝트에 소요되는 시간은 특정 작업을 완료한 후 검토하는 데 사용 되는 작은 시간 단위인 스프린트로 나누어지며, 테스트 전체 프로젝트가 완료되기까지 기다리는 대신 Agile 프로젝트의 모든 단계에서 테스트가 진행됩니다.

✓ Agile의 이점

1. 예측 가능한 일정 수립

사전 정의된 스프린트를 통해 새로운 기능을 신속하고 예측 가능한 방식으로 제공할 수 있으며, 베타 테스트 또한 기존보다 더 일찍 진행할 수 있다.

2. 팀 자율성

Agile은 단순성과 공동 작업에 집중하므로 팀은 자유롭게 자체적으로 조직화하고 중요한 의사결정을 내릴 수 있다.

3. 유연성

Agile의 팀 자율성을 통해 팀은 원하는 결과를 달성하는 데 가장 적합한 방법과 기술을 유연하게 선택할 수 있다. 동시에 프로젝트 자체의 적응력도 향상되므로 새로 생성되거나 변경된 백로그 항목을 개발 도중에 도입할 수 있으며 초기 베타 테스트를 통해 중요한 피드백이 제공되어 개발자는 주요 변경 사항을 프로젝트에 적용할 수 있다.

4. 보다 효율적인 커뮤니케이션

Agile은 효과적으로 내외부적인 커뮤니케이션을 수행하고자 하는 목표가 있기 때문에, 소통을 위한 팀이 필요하며, 직관성과 명료성을 중요하게 생각한다. 따라서 내외부적 커뮤니케이션의 이점을 가지고 있다.

5. 비즈니스 가치에 대한 주안점 개선

Agile 방법론에서는 고객 또는 최종 사용자가 기능의 우선 순위를 결정한다. 따라서 개발 팀은 어떤 기능이 비즈니스에 가장 높은 가치를 제공할지에 대한 통찰력을 얻을 수 있다.

6. 고객 주안점 개선

촉박한 기한과 어려운 장기 목표에 직면한 개발자는 고객의 중요성을 간과하게 되기도 한다. Agile은 고객의 요구 사항과 사용자 피드백을 통해 제품 개선의 기초를 제공하여 이러한 간과된 지향점을 재정립하게 하며, 이를 통해 고객 만족도뿐 아니라 수익도 향상할 수 있도록 돕는다.

✓ Agile의 단점

Agile은 가장 우수한 방법론으로 꼽히기도 하지만, 다음과 같이 채택하기 전에 알아야 할 몇 가지 단점을 가진다.

1. 고도의 고객 참여 요구

고객이 개발 팀과 긴밀하게 협력할 시간이나 의사가 없으면 프로젝트를 진행하는 데 필요한 피드백 또는 통찰력을 제공하지 않는다.

2. 팀 전체의 노력 수반

팀원들이 프로젝트를 효과적이고 효율적으로 완료하고자 최선의 노력을 기울이지 않으면 Agile의 자체 관리 측면이 무의미해진다.

3. 모든 결과물을 수용하기 위한 시간 불충분

일부 작업 또는 특정 하위 작업의 경우 단일 스프린트 동안 완료하기가 어려울 수 있습니다. 이러한 문제를 해결하기 위해 팀은 우선 순위를 변경하거나, 비용을 투입하여 추가 스프린트를 확보해야 한다.

4. 포괄적 거버넌스가 허용되지 않음

Agile의 반복적이고 점진적인 특성은 프로젝트 거버넌스 또는 감독과는 부합하지 않아 자율성이 부족한 팀의 경우 효과적으로 관리하기가 어려울 수 있다.

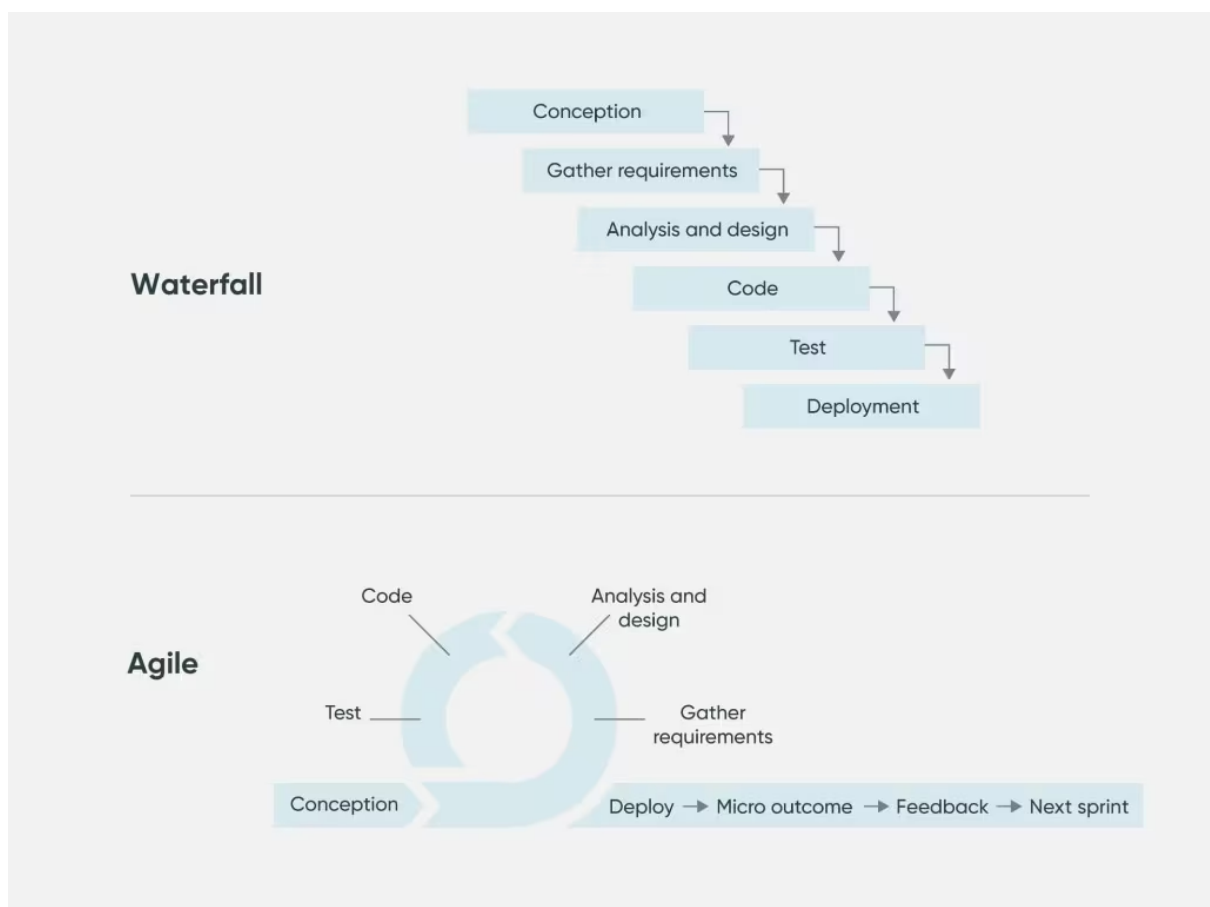
5. 빈약한 문서화

Agile은 문서보다는 소프트웨어 작업을 우선으로 하기 때문에 필수적인 항목이 누락될 수 있다. 종합적인 문서화를 통해 구현을 보다 공유 가능하도록 지원하고, 특정 의사결정을 뒷받침하는 아이디어를 식별하고, 팀이 이전 단계를 보다 손쉽게 검토할 수 있으므로 이러한 누락은 문제가 될 수 있다.

6. 문화 도입 필요

기업의 굳어진 프로세스, 도구, 정책, 조직 구조 및 통제는 Agile이 적합하지 않기도 하다. 따라서 효과적으로 Agile을 구현하기 위해서는 전사적으로 문화를 바꿔야만 하며, 이로 인해 기존 관습에 익숙한 개별 구성원 또는 부서는 괴리감을 느낄 수 있다.

✓ Agile과 Waterfall 중 적합한 접근법 선택



Agile과 Waterfall 모두 각자의 장단점이 존재한다. 이를 유념하여 두 접근법의 특정 사용 사례들을 이해하면 기관에서 진행하는 프로젝트에 가장 적합한 방법론을 채택할 수 있다.

요구 사항

엄격한 프로젝트 요구 사항이 있는 경우에는 Waterfall이 적합하며, 요구 및 규제 사항이 적은 경우에는 Agile을 채택해 창의성과 자율성을 발휘할 수 있다.

기존 프로세스

엄격한 프로세스가 존재하는 경우 Agile을 도입하기 어려우므로 기존 Waterfall 접근법을 도입하는 편이 이롭다. Agile은 보다 유연한 프로세스에 보다 효과적이다.

사용자 참여

Waterfall은 고객, 최종 사용자, 제품 소유자가 개발 팀과 밀접하게 작업하는 것을 지양할 때 효과적이다. 더 많은 개입을 원하는 사용자가 있는 경우 Agile이 더 유용하다.

기존 프로젝트와 혁신 프로젝트 비교

기능이 이미 정의되고 통합이 이루어져 있는 기존의 레거시 프로젝트를 개선하는 데는 Waterfall 접근법이 유리하다. 반면에 새로운 분야를 개척하고 누구도 시도하지 않았던 프로젝트를 진행하는 경우, Agile의 반복적인 접근 방식을 통해 팀은 프로젝트를 진행하며 새로운 것을 배우고 적응해 나아갈 수 있다.

일정

Waterfall 방법론은 예측 가능한 결과를 설정하고 확실한 기한을 정한 장기 프로젝트에 적합하다. 기한이 짧고 유연한 프로젝트는 Agile이 적합하다.

예산

Waterfall은 모든 작업과 비용을 프로세스 초기에 문서화하기 때문에 예측 가능성이 높아 확정된 예산을 정해놓고 시작 가능하다. Agile은 예산 편성 시 경직성이 낮고 기능 구현과 개발 속도에 중점을 두고 있으며, 예산의 변동성이 있다.

프로젝트 규모 및 복잡성

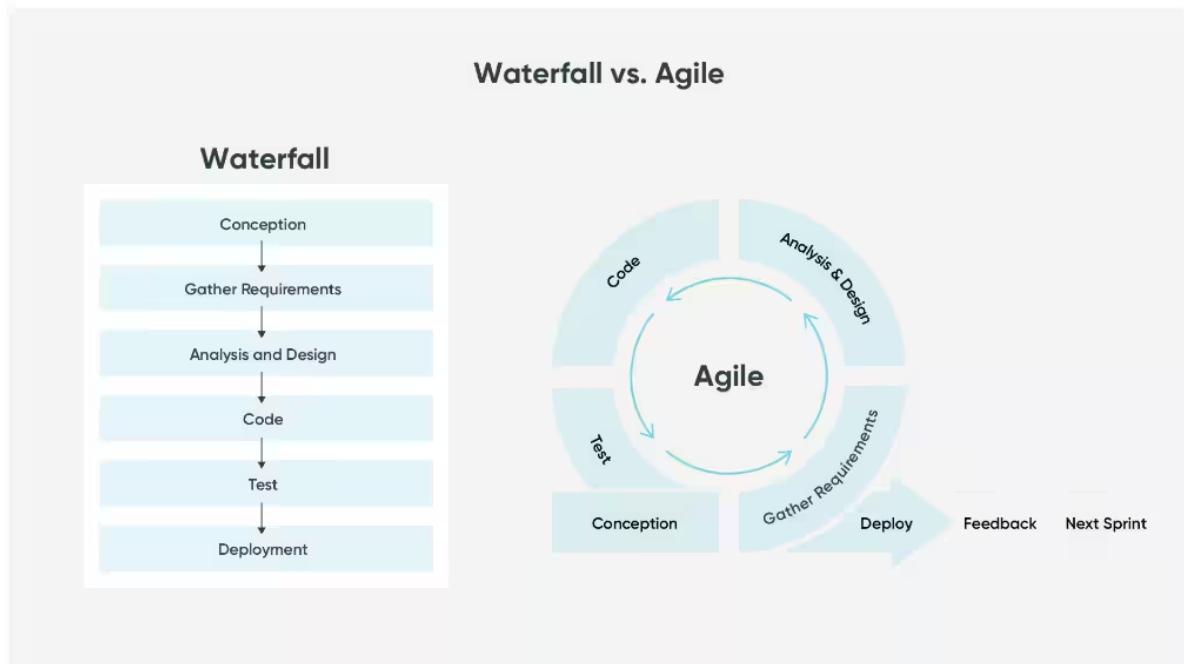
잘 정립된 소형 프로젝트는 보통 Waterfall이 적합하며, 복잡한 대형 프로젝트는 Agile 접근법이 유리하다.

조직적 요소

원격 작업자나 다른 기관과 협업하는 경우에는 대면 협업의 필요성이 낮은 Waterfall이 적절하다. 그러나 단일 기관에 소속된 동일한 위치의 팀원만이 프로젝트에 참여하는 경우에는 Agile이 더 효과적이다.

✓ tip! Agile과 Waterfall 모두 뚜렷한 이점이 있기 때문에, 전 세계 회사들은 두 방법론의 단점을 제한하고 장점을 결합하는 방법을 모색하고 있다.

하이브리드 프로젝트 관리 : 하이브리드 프로젝트 관리는 다양한 관리 접근 방식을 단일 방법론으로 통합하는 프로젝트 관리의 한 형태!



✓ Agile을 돕는 협업툴 - Jira / Confluence

1. JIRA란?

1. JIRA 개념

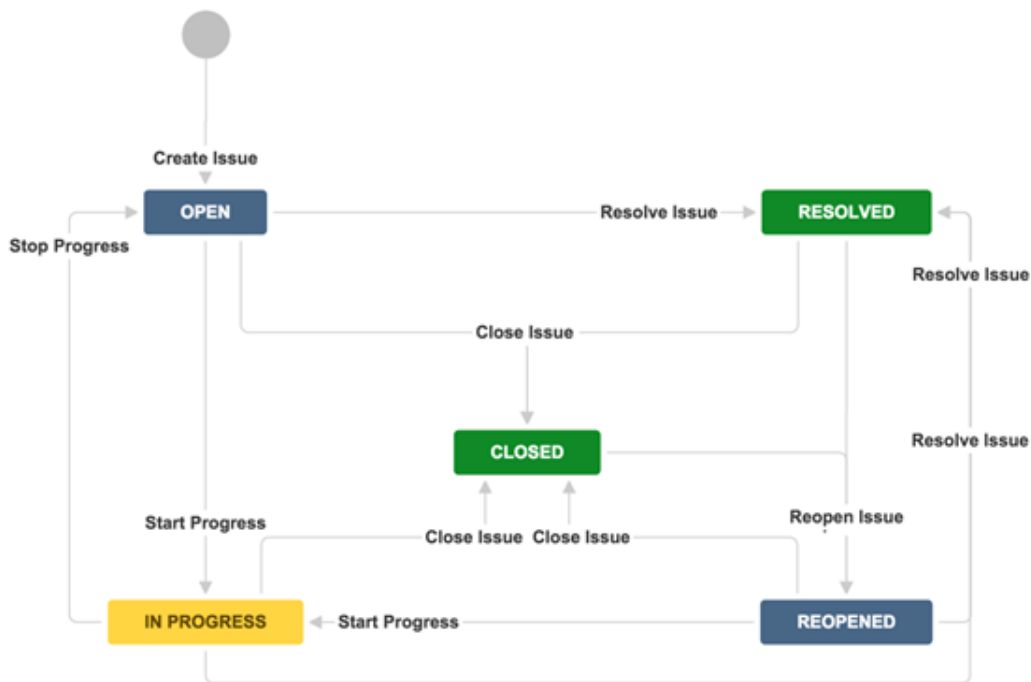
- 호주 회사 Atlassian에서 버그 및 문제를 추적하고 프로젝트를 관리할 수 있도록 만든 도구
- 대부분의 Agile 방식 프로젝트 관리를 위해 설계되어있음.
- 문제를 찾을 수 있는 대시보드를 제공함으로써 버그 추적 및 , 프로젝트 관리를 용이하게 하는 역할

2. JIRA 워크플로

1. Open Issue - Issue가 발생하면, 담당자에게 표시가 된다.
2. In Progress Issue - 특수문자 인식 및 작업을 시작했음을 "진행 중"으로 표시한다.
3. 해결된 문제 - 담당자가 문제를 획일적으로 해결한 후 표시한다. 동시에 문제 제기자나 다른 관련이 된 사람에게 문제를 표시할 수 있고, 이를 확인한 후이면 성공, 문제가 닫히

고 추가 변경사항이 필요하면 문제가 다시 열린다.

4. Reopened Issue - Issue에 대한 해결이 완전히 되지 않았거나, 또 다른 Bug들이 발견 되었을 때, 이전 Issue 책임자나 다른 책임자에게 표시할 수 있다.
5. 이슈 닫기 - 이슈가 피스로 풀린 것이 확인된다, 이슈가 툰다. 일반적으로 이슈를 보고한 사람이 이 작업을 수행한다.



2. Confluence란?

1. Confluence 개념

- Confluence는 협업도구로, 지식공유 및 협업을 할 수있게 돕는 도구이다.
- 사용자는 블로그 및 페이지를 만들 수 있고 댓글을 달고 글을 편집할 수 있다.
- 로드맵, 체크리스트가 있는 메모, 지식 관련 문서 등을 중앙에서 관리할 수 있다.
- JIRA와 통합이 가능하다.

3. JIRA와 Confluence 차이점

비교 기준	Confluence	JIRA
설명	Confluence는 JIRA에서 수행 중인 프로젝트에 대한 모든 정보 자산을 저장하고 구성하는 협업 도구.	JIRA는 팀이 전체 수명 주기 동안 문제를 쉽게 관리할 수 있도록 하는 문제 관리 플랫폼 기본적으로 소프트웨어 개발에서 개발 노력을 관리하고 추적하는 방법으로 사용.
지원되는 플랫폼	웹, 아이폰, 안드로이드	웹, 아이폰, 안드로이드
일반 고객	소규모, 중간 규모 및 엔터프라이즈 고객	소규모, 중간 규모 및 엔터프라이즈 고객
가격	월 \$10부터 시작	월 \$10부터 시작
액세스 제어	지원	지원되지 않음
행동 관리	지원되지 않음	지원
활성 디렉토리 통합	통합 불가	통합 가능
활동 대시보드 및 추적	사용 가능	사용 가능
애플리케이션 통합	지원되지 않음	지원
할당 관리	지원되지 않음	지원
캘린더 관리	지원	지원되지 않음
콘텐츠 관리	지원	지원되지 않음
문서 관리	지원	지원되지 않음
동적 워크플로우	지원되지 않음	지원
프로젝트 기획	지원	지원되지 않음
프로젝트 관리	지원	지원되지 않음
프로젝트 시간 추적	지원되지 않음	지원
타사 통합	지원	지원
보고 및 통계	지원되지 않음	지원
프로젝트 워크플로	사용 불가	사용 가능

4. JIRA와 Confluence를 통합해야 하는 이유

- Confluence에서 직접 릴리스 노트, 요구 사항 사양 및 코드 검토를 유지하고 구성하고, 모든 프로젝트 문서를 중앙 위치에 보관하기 때문에 팀원이 필요한 것을 찾기 위해 이메일, 컴퓨터 폴더 또는 Google 드라이브를 검색할 필요가 없어진다.

- Jira와 Confluence를 결합하여 팀 간 커뮤니케이션을 활성화할 수 있다. 따라서 엔지니어링 부서와 나머지 비즈니스 팀 간의 장벽을 허물고, 비즈니스 이해 관계자는 릴리스 상태를 확인하고 팀 진행 상황을 완벽하게 파악할 수 있어진다.
- 팀이 Jira 소프트웨어와 다른 리소스 간에 전환할 때 낭비되는 시간을 줄이고 컨텍스트 간에 전환하여 더 빠르게 제공할 수 있다. 또한 소프트웨어 개발을 수행하는 팀원은 Jira 내에서 올바른 문서에 액세스할 수 있다. 가젯을 추가하고 개발자가 현재 작업 중인 작업과 관련된 문서를 쉽게 볼 수 있도록 하여 Confluence 페이지를 Jira로 가져올 수 있으며, 신뢰할 수 있는 연결을 설정하고 사용자가 제한된 Confluence 및 Jira 페이지에 액세스할 수도 있다.
- 또한 비즈니스 팀이 보고를 통해 이점을 얻을 수 있다. Jira는 프로젝트 및 문제에 대한 데이터를 계속 생성하지만 비즈니스 팀은 이러한 보고서를 자세히 확인할 시간이나 기술이 없을 수 있다. 따라서 Jira 정보를 Confluence 보고서에 연결하면 비즈니스 팀이 Jira에서 사용할 수 있는 풍부한 데이터를 활용할 수 있도록 간단한 사용자 인터페이스에서 상태 보고서를 이해할 수 있는 차트로 표시할 수 있다.

5. JIRA 와 Confluence는 언제 사용해야 하는가?

✓ 업무 흐름(work flow) 을 관리할 때는 JIRA 를, 문서 기반 협업을 할 때는 Confluence 를!

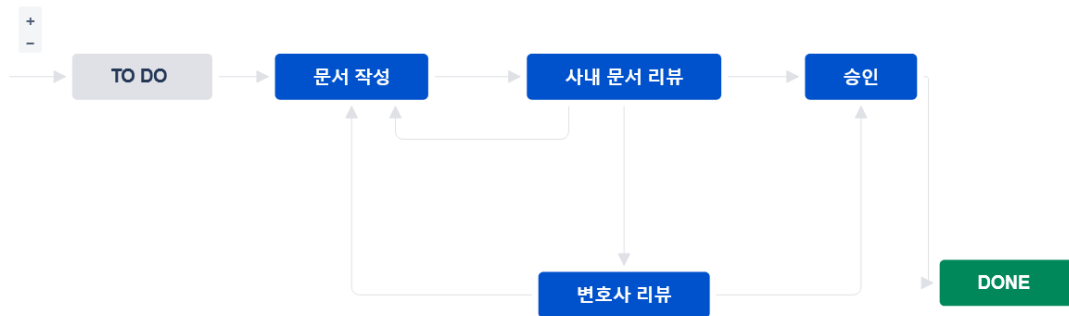
JIRA 의 용도



업무 흐름(Work flow) 을 추적해야 하는 경우 JIRA 를!

- 만약 개인정보보호법을 준수하기 위해 제공중인 웹 서비스에 올릴 "개인정보처리방침" 문서를 작성하는 업무를 받았다고 가정했을 때, 사내에는 문서를 작성하고 검토하고 외부에 배포하는 절차가 있을테고 그 절차를 "문서작성 workflow" 라고 할 수 있다. 문서는 관련 팀에서 초안을 작성하고 내부 검토를 거쳐서 승인후 배포되며 "개인정보처리방침" 같이 법/제도와 관련된 문서는 변호사의 검토를 다시 거친후에 외부에 공개하는 work flow 를 갖고 있다고 가정할 수 있다. 그 때, JIRA 의 work flow 에디터 기능을 사용하면 다음과 같이 모델링할 수 있다.

DOCA: Document Approval Workflow

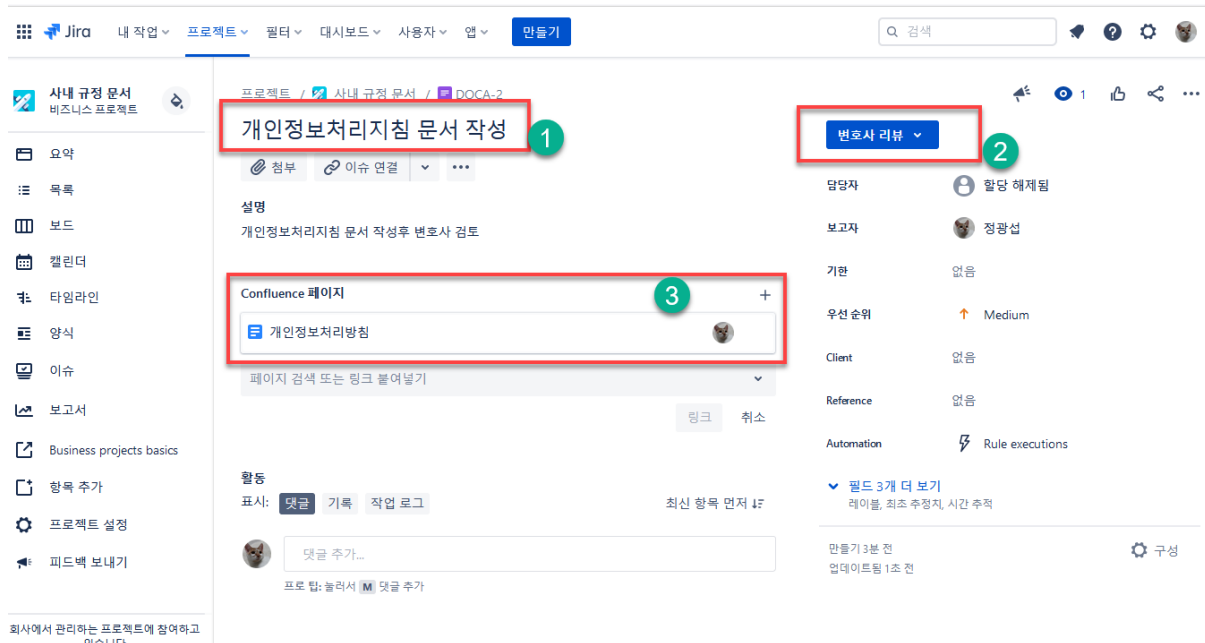


마우스를 객체 위로 가져가 전환 이름을 확인하십시오.

닫기

이제 문서를 작성할 때 각 문서가 어느 workflow 상에 있는지 추적하기 위해 "개인정보처리방침" issue 를 JIRA 에 등록하고 진행 단계마다 work flow 에 맞게 상태를 변경해 주면 된다!

다음 예시는 JIRA 에 task 를 등록하고(1) 상태를 관리하며(2) Confluence 와 연결한(3) 화면



Confluence의 용도

"개인정보처리방침" 문서는 관련 부서에서 초안을 작성하고 내용을 검토하고 오타나 어색한 문구등을 조율해야 한다. 이렇게 문서 기반으로 협업과 공유를 할 경우는 Confluence 를 사용하여 편집과 리뷰를 공동으로 하면 수정 → 취합 → 배포에 드는 노력과 시간을 최소화할 수 있다.

문서의 내용은 공동으로 작성하며 진행 상태는 Confluence Page 와 JIRA Issue 를 링크하고 각 단계가 끝나면 JIRA 의 상태를 변경해 주어서 work flow 를 수정해 주며, 내부 리뷰까지 끝났으면 변호사에게 공유하고 JIRA 의 status 를 "변호사 리뷰" 상태로 변경해 주면 이슈의 상태와 실제 수행하는 업무(문서 작성)를 각각 용도에 맞게 사용하면서 진행 상황을 추적하고 협업과 공유를 할 수 있다.



이렇게 JIRA 는 Work flow 관리, Confluence 는 문서 기반 협업의 용도로 사용하며 Confluence Page 와 JIRA Issue 를 연결하면 더 강력하게 협업을 진행할 수 있다!

참고

<https://www.servicenow.com/kr/products/business-management/what-is-agile-vs-waterfall.html>

<https://www.lesstif.com/ats/jira-confluence-113345338.html>

<https://doosil87.github.io/devops/2019/09/01/Jiraandconfluence.html>

<https://www.educba.com/jira-vs-confluence>

<https://www.servicenow.com/kr/products/business-management/what-is-hybrid-project-management.html>