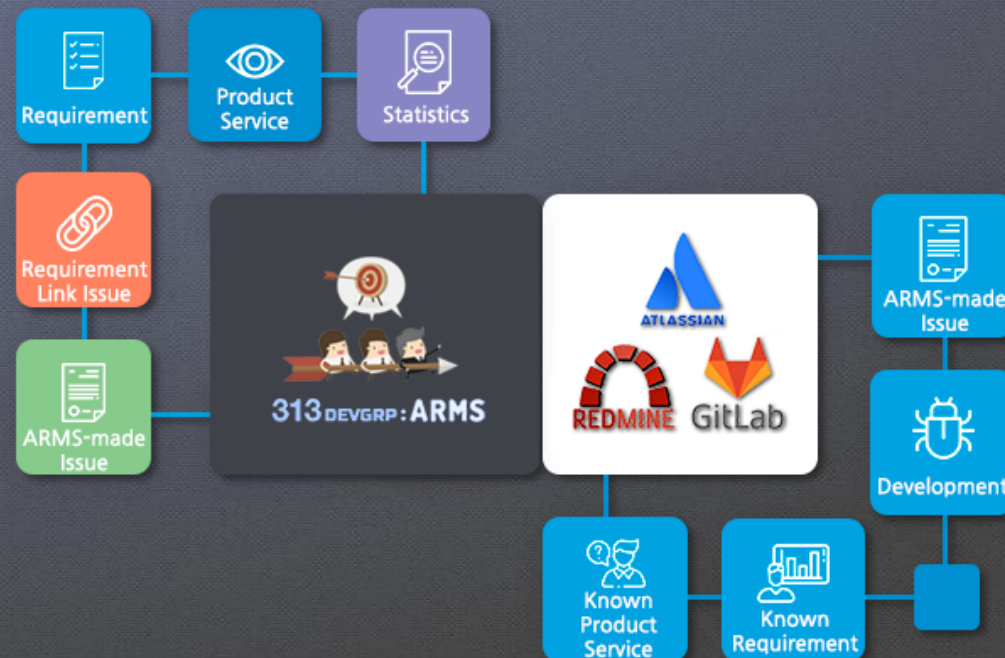



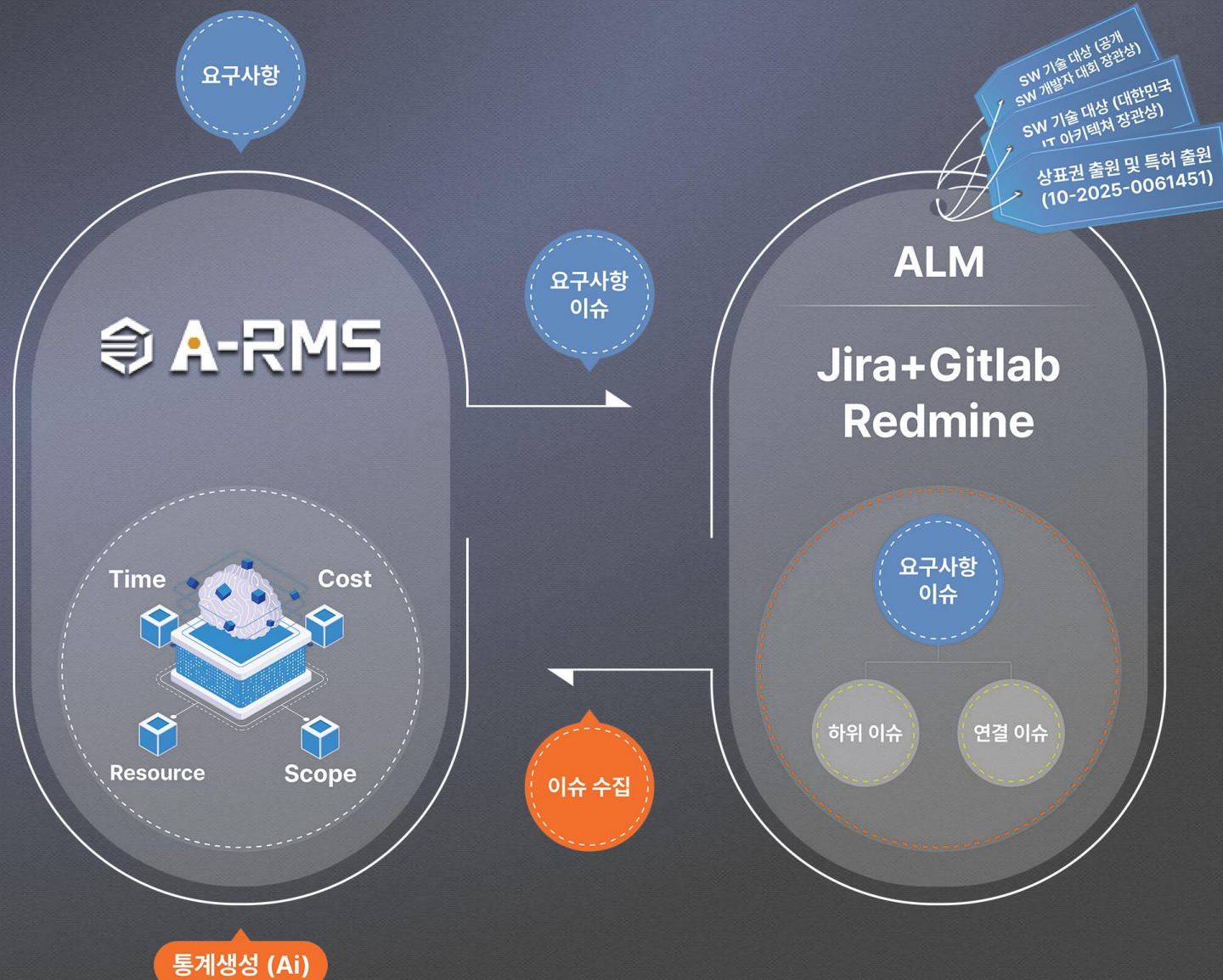
# **Ai** + **ALM** + **RMS** + **PMS** = **A-RMS**

313DEVGRP





A-RMS 는 = 인공지능을 활용 \* ALM (    ) 을 연결한 + 요구사항 기반 ( RMS ) → 프로젝트 관리 시스템 ( PMS ) 입니다.





# Our History

I . Background

**2024 SW기술 대상**  
과학기술정보통신부 장관상  
2024 대한민국 SW기술대상 및 IT 아키텍처 대상(장관상) 수상자 발표

SW기술 부문

**SAMSUNG**  
삼성전자

"비전검사 자동화 플랫폼 Visikit & DeepDive"

삼성전자

SW기술 부문

**이온커뮤니케이션즈**

"ICE (I-ON Contents Ecosystem)"

이온커뮤니케이션즈

IT아키텍처 부문

**A-RMS**

ALM+RMS+PMS-A-RMS

313DEVGRP

수상하신 모든 분을 축하드립니다.  
더불어 대외기간 동안 포기하지 않고 완주하신 모든 참가자 분들과 감사의 말씀과 격려의 박수를 드립니다.  
모두 수고하셨습니다!

**2024년 공개SW 개발자대회 수상작 발표**

오른소스 세상을 삼키다

No.	구분(종목)	상금	접수번호
1	대상 (2)	각 1,000만원	A0027
2	과학교육정보통신부 장관상		A0097
3	공상 (2)	각 500만원	A0106
4	정보통신산업진흥원 원장상		A0644
5	은상 (2)	각 250만원	A0121
6			A0447
7			A0175
8			A0217
9			A0391
10			A0443
11	동상 (10)	각 200만원	A0483
12			A0607
13			A0670

후원기업상 +

**금상(협회장상) 수상자 발표**

SW기술 부문

**SOFTZION**

"위시드 CRM"

소프트지온

IT아키텍처 부문

**여기어때.**

"AI-Shield (AI데이터 세트 활용 보안관계 분석대응 시스템)"

여기어때컴퍼니

수상을 축하드립니다!

한국소프트웨어기술진흥협회

2014 2015 2016 2017 ..... 2024

## 313DEVGRP

## A-RMS

2025

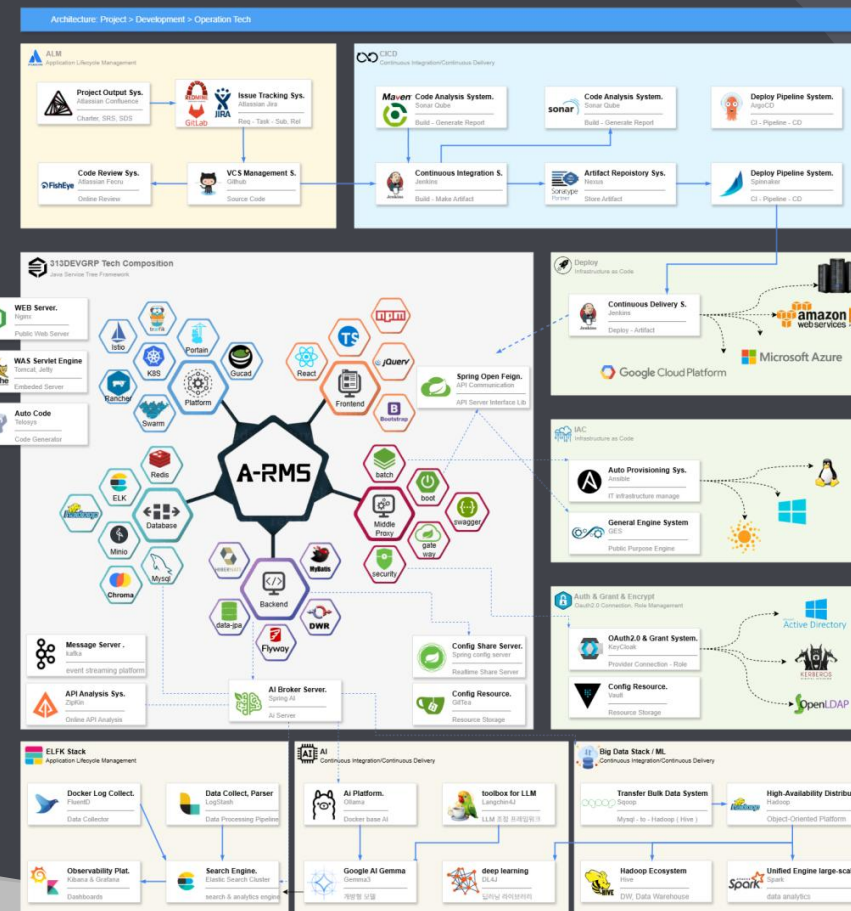
누구에게나 주어지는 10년,  
우리는 기술을 쌓았습니다.

기능에서 시스템으로  
코드에서 플랫폼으로

겹겹이 쌓인 기술 위에,  
겹겹이 쌓인 경험 위에,

지금의 A-RMS를  
만들어 냈습니다.

## Product Line Engineering





# 단순한 감축은 답이 아니다

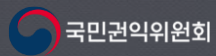
사람을 줄일 것인가, 일을 보이게 할 것인가

## I . Background

'줄이는 것'만으로 충분한가?

감축보다 중요한 건, **인력 효율화**로 경쟁력 확보

- ▶ “사업 접는 게 이득... 상반기 하루 5개 기업 파산” – 한국경제
- ▶ “IT업계 찬바람... 1분기 2.2만명 감원” – 지디넷코리아
- ▶ “ICT 대기업 칼바람... 구조조정의 딜레마” – 뉴데일리



공공

엑셀 수기 관리로  
협업·진척 추적이 어려움



SI

외주 커뮤니케이션 착오로  
요구사항 불일치



포털

요구사항 범위 불명확으로  
작업 혼선 반복



보안

인력 부족 속 병렬 작업으로  
업무 충돌 지속



게임

우선순위·일정 조율  
실패로 출시 지연



연구

예산 기준에 밀려  
핵심 기능 후순위

Pain Point

문서-시스템 단절로 요구사항 반영 지연

중복 업무와 비효율로 리소스 낭비

전달자-실행자 범위 오해로 재작업 반복

투자 판단 오류로 예산 배분 실패

Challenge

**비즈니스 적시성**  
출시 지연과 일정 실패

**비즈니스 효율성**  
인력·비용 대비 낮은 성과

**비즈니스 적합성**  
작업 범위 불명확

**비즈니스 수익성**  
투자 대비 기회비용 손실



# 危機

심각한 위기가 허비되는 것을 그냥 놔둬서는 안 된다.

이 말은 곧 위기란 그전에는 **할 수 없던 일을 할 수 있는 기회**라는 것이다.



## 비즈니스 적시성 실패 ( Time )

신작 출시 지연, 위메이드 1분기 영업손실 105억원 기록



제품\*(서비스) 출시가 지연 되고  
있습니까?

▲ 2014년 1분기 실적발표 후 위메이드 엔터테인먼트 (재무제표, 위메이드 엔터테인먼트)

위메이드 엔터테인먼트는 2014년 1분기 매출액이 33% 감소하며 398억원의 기록하고 영업손

지연을 막을 수 있는 대책은 무엇이라고  
생각하십니까?

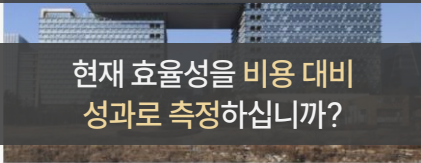
위메이드는 2분기부터 완성도 높은 온라인게임과 모바일게임 신작들로 국내외 게임 시장을  
동시에 공략해 향후 실적전망을 밝게 하고 있다. 온라인게임 부문에서는 지난달 16일 서비스를  
시작한 '이카루스'가 일주일 만에 PC방 점유율 5%, MMORPG 장르 점유율 1위를 기록하며

## 비즈니스 효율성 관리 실패 ( Resource )

[게임썰전] 위기의 엔씨소프트, 2012년 구조조정 재현되나

엔씨소프트 인적 구조조정설 업계 및 사내에도 확산 중  
실적 급감하고 주가 급락하는데 직원 수 4800명 육박 업계 최다...평균 연봉은 1억원 '올  
썰'  
신작 부재 속 구원투수 해야할 '쓰론엔리버티'는 불안요소 산재  
2012년에 있었던 구조조정 재현설도 나와...당시 400명 회사 떠나

투입 인력의 효율, 비용은 적절합니까?



현재 효율성을 비용 대비  
성적으로 측정하십니까?

[포인트데일리 김국현 기자] 엔씨소프트 인적 구조조정설이 업계에 확산되고 있다. 심각한 실적 하락  
과 주가 급락 속에서 너무 많이 늘어난 직원들에 대한 구조조정이 임박했다는 소문이다. 엔씨소프트

## 비즈니스 범위 관리 실패 ( Scope )

"절반이 적자" 카카오 형제들 비명...구조조정 확산되나

카카오엔터테인먼트, 고연자 직원 희망퇴직 진행  
영업익 못내는데 투자 말라...공동체 대부분 '불안'

카카오 주요 자회사 2022년 영업손실 (단위: 억)

카카오 카카오 카카오 카카오 카카오 카카오 카카오

지시사항 + 요구사항 전파가  
제대로 되고 있습니까?

kakao

작업자까지 정확하게 전달되어  
이해되고 있다고 보십니까?

카카오 주요 자회사의 절반 이상이 지난해 영업손실을 낸 만큼 '도미노 구조조정'이 현실화할 수 있  
다는 우려다. 투자유치와 IPO(기업공개) 기반의 카카오식 성장방식이 난관에 봉착했다는 평가도  
나온다.

## 비즈니스 투자 성과 실패 ( Cost )

카카오에 3조원 물린 투자자들 자금회수 '비상'...구조조정 불가피

카카오그룹에 악재 잇따르며 카카오에 투자한 재무적투자자 곤란해져  
본업 혁신보다 자본 축적에 치중한 카카오 경영진과 이에 편승한 투자자들  
고통스러운 정도의 엑시트 과정이 필연적으로 뒤따를 수밖에 없어

카카오그룹에 대형 악재가 잇따르면서 카카오엔터테인먼트-카카오모빌리티 등 카카오의 비상  
장 계열사에 투자한 재무적투자자(FI)들의 엑시트(투자금 회수) 계획에도 빨간불이 켜졌다. 주  
요 계열사의 실적부진과 주가 하락에 따라 투자자들은 투자금 회수를 위한 엑시트를 추진하고 있다.

프로젝트 투자 대비 수익이  
명확하게 측정되고 있습니까?

단면을 드러냈다는 비판의 목소리도 나온다.

카카오 주요 계열사 투자 유치 현황

투자 우선순위 결정을  
어떤 근거로 하고 계십니까?

카카오모빌리티	구분	금액
	TPG벤처스 및 국민연금	565억원
	(주)LG	1400억원
	GS그룹	950억원

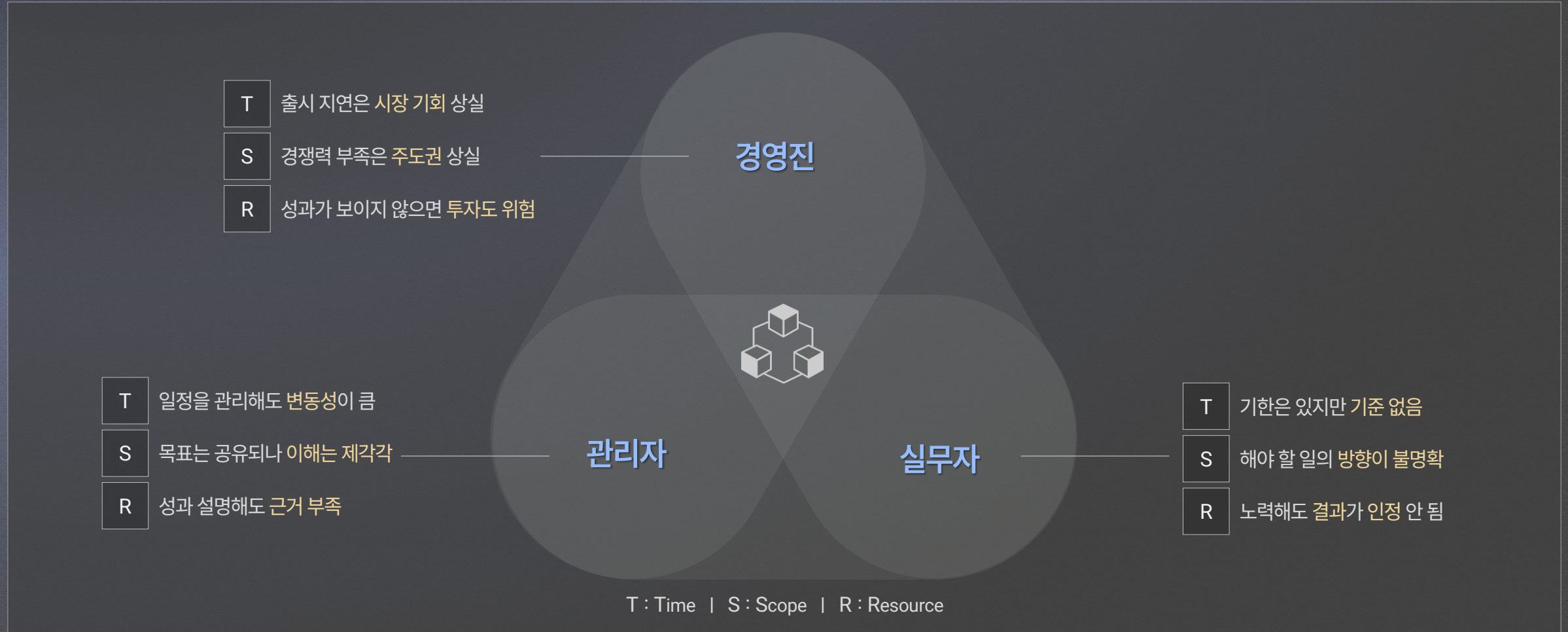


# 실무자는 답답하고, 리더는 애매하며, 경영진은 불확실합니다

일이 되게 하는 솔루션. A-RMS.

I . Background

보이지 않는 '요구'가 프로젝트를 무너뜨린다





# 모두가 알지만, 알고도 『그냥』 지나친 관행

질문 : 『 RFP, Project Charter, 기획서 등등 』 프로젝트 문서를 , 『 개발자 및 작업자 』가 얼마나 이해하고 있습니까?

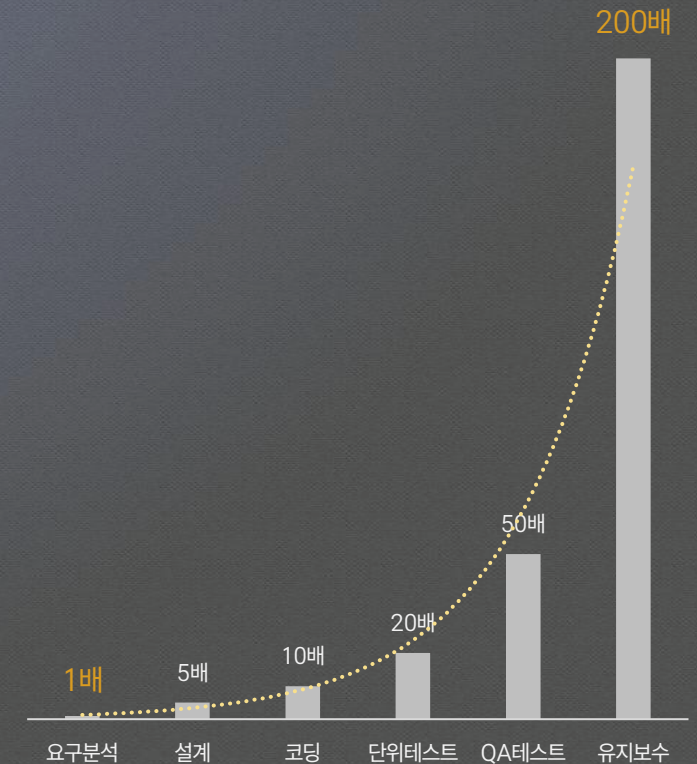
## I . Background



프로젝트 문서를 ‘전달한 것’과 ‘이해된 것’은 다를 수 있다

- ▶ 요구사항 변경의 갈등은 ‘서로 다른 요구 오해’에서 시작됩니다
- ▶ 프로젝트 팀은 ‘변경’이라 부르지만, 실제로는 ‘요구 오해를 바로잡는 과정’입니다
- ▶ 요구 오류가 늦게 발견될수록 결함 제거 비용은 10~13배씩 증가합니다 (출처: Karl Weigers)

요구사항 수정 비용





# “악순환”의 고리, 그 시작은 ...

질문 : 프로젝트 취소, 납기 지연, 비용 초과, 프로젝트 실패의 『근본적인 원인은』 무엇이라고 생각하십니까?

## I . Background



이해되지 않은 요구는 가장 비싼 재작업이 된다

- ▶ 문제의 핵심은 요구사항 오류는 초기에 잡아야 비용이 적게 듭니다
- ▶ 요구사항의 일관성과 완전성 결함은 사전 제거로 재작업을 줄여야 합니다
- ▶ 요구가 명확하지 않으면 작업자는 '이해' 보다 '지시'만 따르게 됩니다







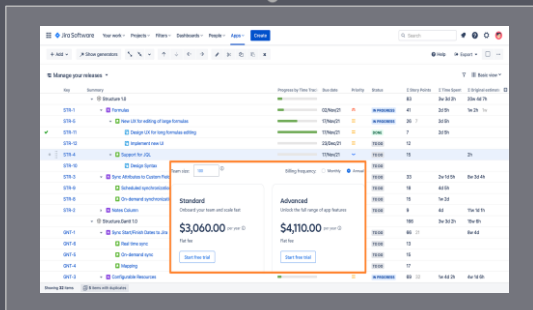
# RMS or PMS

기존에도 알고 있었다.

요구사항은 없고, **ISSUE** 만 존재한다.

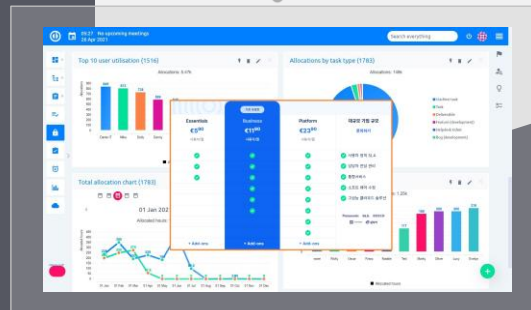
ISSUE 에 의한 **오염된** 통계를 본다.

연결 정보가 없을 수밖에 없다.



Jira Plugin

통계 정보	X
분석 정보	X
지표 정보	X

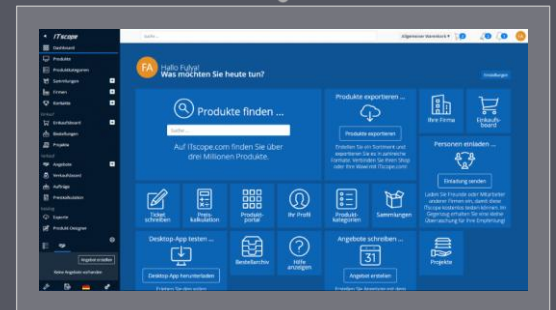


Redmine Plugin

통계 정보	X
분석 정보	X
위키 시스템	X

Jama

이슈 수집	X
지표 정보	X
가시성	X



ITSCOPE

ALM 교체	Y
통계 정보	X
변화 유무	Y

번호	위계	요구ID	구분	요청사항	설명	ISP	RFP	수행 개척/기	비커지 기	비고란
1	ISP 3장	R.JK_001	마일리지	지식행정 활동 통계	지식행정시스템을 통해 자동 수집되는 마일리지	III-36	X	X	0	
2		R.JK_002	마일리지	이벤트 마일리지	각종 이벤트를 통해 표시되는 마일리지	III-36	X	X	0	결정평가
3		R.JK_003	마일리지	마일리지 집계	위반행위에 대해 마일리지 산정	III-42	X	X	0	
4		R.JK_004	마일리지	복합의 지식행정 활동 기준	각 활동에 대한 기준화된 점의 및 배점별 점수 운영	III-42	X	X	0	
5		R.JK_005	마일리지	마일리지 평가 기준	지식유형별로 마일리지 점수의 부여 기준을 설정	III-43	X	X	0	
6	ISP 3장	R.JK_006	마일리지	CoP 마일리지	CoP내에서 활동한 결과에 대한 마일리지는 지식유형별로 등록, 피검증, 검증, 의견, 피의견, 피평가 등의 항목으로 운영할 수 있음	III-87	X	X	0	
7		R.JK_007	평가 방안	업무 BP 평가 방식	업무BP는 환경부 지식행정 고도화의 핵심적인 컨텍스트로 향후 지속적인 관리와 평가가 필요함	III-44	X	X	0	
8		R.JK_008	평가 방안	공통성과 지표	업무BP별 점수 부여 절차적 통계기반의 지식활동과 오프라인활동에 대한 점성적 평가로 종합하여 평가함	III-49	X	X	X	별도집계(신규개발)
9	ISP 3장	R.JK_009	평가 방안	CoP 활동 평가	CoP 활동 관련 마일리지와 결과 보고서의 점성적 평가로 세부 평가준 하도록 함	III-83	X	X	Δ	결정평가(기능개선)
10		R.JK_010	분류체계	지식행정분류체계 구성	지식의 유형별로 구분된 점 구성에 대하여 다양한 분류기준 중에서 선정된 BRM분류 구분, COP분류 구분, 지식유형 구분, 조직유형 구부의 적용	IV-110	X	X	0	
11		R.JK_011	지식저장과 속성정의	인물정보 속성정의	업무상 만나게 되는 인물에 대한 정보를 등록하고 공유하는 장소이며, 단순히 인물에 대한 형식적인 정보 뿐만 아니라 그 사람의 의미, 좋아하는 음식, 취향등의 입체적인 정보를 아울러 등록함으로써 인물관리를 더욱 세부적으로 관리	IV-124	X	X	0	
12	ISP 3장	R.JK_012	지식저장과 속성정의	표준지식저장과 속성정의	홈페이지에 있는 법령, 배규, 고시, 훈령을 환경부의 업무영역에서 가장 표준적으로 적용해야 함 지식으로 정의하고 공유하는 방안	IV-126	0	0	0	연계 취소되어 링크로만 자료 공개 [회의록_20071001_표준지식 홈페이지 연계 회의.hw] 참조
13		R.JK_013	지식저장과 속성정의	O&A형 지식저장과 속성정의	법령을 EXB에서 구성하고 홈페이지에 있는 고시/훈령/배규를 링크하여 보여줌	IV-128	X	0	0	
14		R.JK_014	지식관리 프로세스	지식관리활동 정의	O&A형의 지식저장과는 사용자가 업무상 발생하는 구체적인 점이나, 문제해결요청시 일반사용자나 지식전문가에게 업무의응답을 수행할 수 있는 장소	IV-132	X	X	0	
15	ISP 3장	R.JK_015	지식관리 프로세스	지식유형별 관리 프로세스	지식관리 프로세스 상의 지식관리 활동은 지식의 창출, 공유, 고도화를 위해 여러가지로 구분(등록, 수정, 검증, 조회, 평가, 인용, 고도화, 평가)	IV-136	X	X	0	
16		R.JK_016	업무 BP 프로세스	업무 Best Practice 등록	지식유형별로 지정된 분류체계에 의해 지식을 등록하고 해당 분류체계내에 맞는 지식관리 프로세스를 차별화시켜 관리	IV-140	X	0	0	
17		R.JK_017	업무 BP 프로세스	업무 Best Practice 인수인계	업무 BP를 잘 작성한 것을 후임자에게 인수인계 기능을 이용하여 권위를 인정하여 쉽게 인수인계함	IV-140	0	0	X	인수인계(신규개발)
18	ISP 3장	R.JK_018	업무 BP 프로세스	업무 Best Practice 평가	작성한 업무BP를 상위 부서장에게 평가를 요청하여 점수를 받음	IV-140	X	X	0	R.JK_007과 동일
19		R.JK_019	지식관리 프로세스	개인지식 관리 프로세스	공유대상 지식판단, 등록대상 지식분류, 등록양식 작성, 지식등록, 전자화 프로세스/Know-Where 정리 등의 액티비티 구성	IV-142	X	X	0	

## I . Background



# “선순환”의 고리, 그 시작은 『측정 + 변화』

질문 : 작업자의 워크플로우를 변화 없이, 작업자의 이해도를 증진 시킬 수 있다고 생각하십니까?

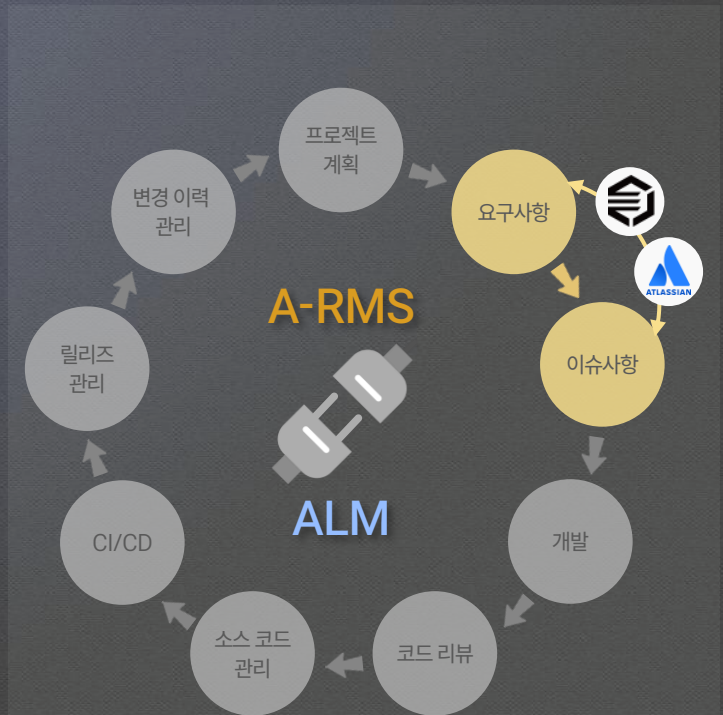
## I . Background



작업자의 시스템 교체가 아닌, 기존 ALM 그대로 사용합니다

- ▶ 핵심은 기획의도와 요구사항을 시스템에 연결하는 것입니다
- ▶ 요구사항·지시사항을 작업자에게 직접 전달하는 구조를 만듭니다
- ▶ 관리자는 흐름을 예측하고, 작업자는 의도를 이해해 혼선을 줄입니다

## ALM 통합 : 요구사항 관리 시스템





# 『측정』할 수 없다면 『관리』할 수 없고, 관리할 수 없으면 『개선』할 수 없다

I . Background

질문 : 현재 측정하고 있는 데이터가 ISSUE 데이터를 기반으로 하고 있습니까?



정확한 진척도 확인



연결된 ISSUE와 요구사항의 이해



기획의도에 맞게 요구사항 해결

측정할 수 없다면 관리할 수 없고, 관리할 수 없으면 개선할 수 없다

- ▶ 해결의 효과는 요구사항 오류 수정 비용이 가장 싸다는 것입니다
- ▶ 요구사항이 일관되지 않으면 재작업이 반복됩니다
- ▶ 변화가 없다면 작업자는 목적 없이 '시키는 대로'만 하게 됩니다

요구사항 + ISSUE

ALM (ISSUE)을  
A-RMS로 수집·분석·통계

02

Requirements



Requirements

01

ALM (ISSUE)로  
요구사항을 주입



# 예측 가능한 비즈니스, 그 시작은 『작은 변화 – 가시성』입니다

질문 : 프로젝트 취소, 납기 지연, 비용 초과, 프로젝트 실패의 『근본적인 원인』은 무엇이라고 생각하십니까?

## I . Background



예측 가능한  
추가패치(Time) · 신규기능(Scope) · 자원배분(Resource)

관리 가능한  
일정관리(Time) · 진척관리(Scope) · 성과관리(Resource)

가시적인  
개발일정(Time) · 요구확인(Scope) · 인원배치(Resource)

Scope

▶ 목표 설정 — A-RMS는 제품·서비스 단위로 요구사항을 관리합니다

Time

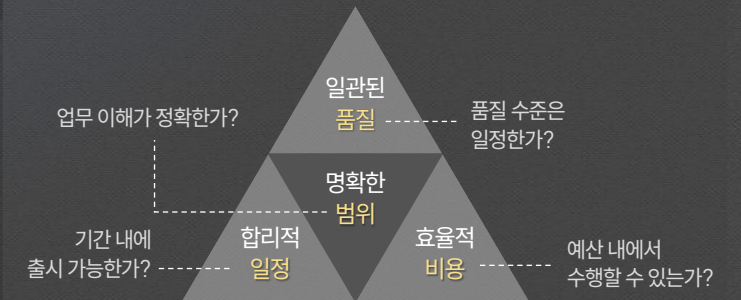
▶ 출시 시점 — A-RMS는 버전별로 일정을 구분하고 요구사항을 관리합니다

Resource

▶ 인력 배치 — A-RMS는 ALM 이슈 할당자를 수집해 정량적 성과를 측정합니다

Cost

▶ 투자 예측 — A-RMS는 요구사항 기반 데이터를 역산해 비용을 추적합니다





# A-RMS 관심 + 목표 = 효과

프로젝트의 사전 예측을 실현하다

## I . Background

### Time 분석

- 개발·테스트는 일정 내 가능한가?
- 시험 제품은 언제 준비되는가?
- 출시 일정은 가능한가?

### Scope 분석

- 제품은 기획대로 나왔는가?
- 요구사항이 의도대로 구성됐는가?
- 요구사항이 명확히 전달·관리되는가?

### Cost 분석

- 요구사항 단위별 소요 비용은?
- 총 요구사항 비용은?
- 전체 개발 및 운용 비용은?
- 성과 대비 비용 효율은?

### Resource 분석

- 투입된 인력은 어느 정도인가?
- 인력별 성과 차이는 어떤가?
- 팀 단위 퍼포먼스는 어떤 수준인가?

A-RMS

### 실질적인 채용 효과

#### ■ 채용분야 및 지원자격

채용 분야	인원	주요 업무	처우
PM 및 중간 관리자 전문 인력	0 명	<ul style="list-style-type: none"><li>- 사내 제품 및 서비스를 대상으로 전일동안 진행된 업무를 조사하여 1) 어떤 업무가 진행됐고 2) 누가 실행했으며 3) 일정 대비 열만큼 진행됐는지 4) 투자 비용 대비 성과 분석까지 통계 리포트를 익일 09시까지 임직원에게 결과를 송부 해야 합니다.</li><li>- 전일 진행된 요구사항 및 지시사항을 기준으로 작업자의 워크시스템 - TASK를 확인하여 요구사항 추적 매트릭스(Requirements Traceability Matrix)를 구성 후 통계 리포트의 Row Data로 제공할 수 있어야 합니다.</li><li>- 각 요구사항 및 지시사항을 ( 변경 포함 ) 해당 팀의 워크시스템 ( Jira, Redmine, Gitlab )에 즉각적으로 반영해 주어야 합니다.</li><li>- 문서 및 산출물을 관리하며 관리자 및 작업자가 소통할 수 있는 창구 역할을 수행하여야 합니다.</li></ul>	<ul style="list-style-type: none"><li>- 월 급여 : 50 만원 ( 직원수 변동에 차등 지급 )</li><li>- 근무 시간 : 365일 24시간 근무</li><li>- 근무 형태 : 계약직 ( 1년 성과 보고 후 재계약 조건 )</li></ul>

### A-RMS 일의 관점을 바꾸다

‘사후’가 아닌  
‘사전’ 대응

요구사항의  
‘가시성’ 제공

‘진척 관리’로  
프로젝트 품질 확보

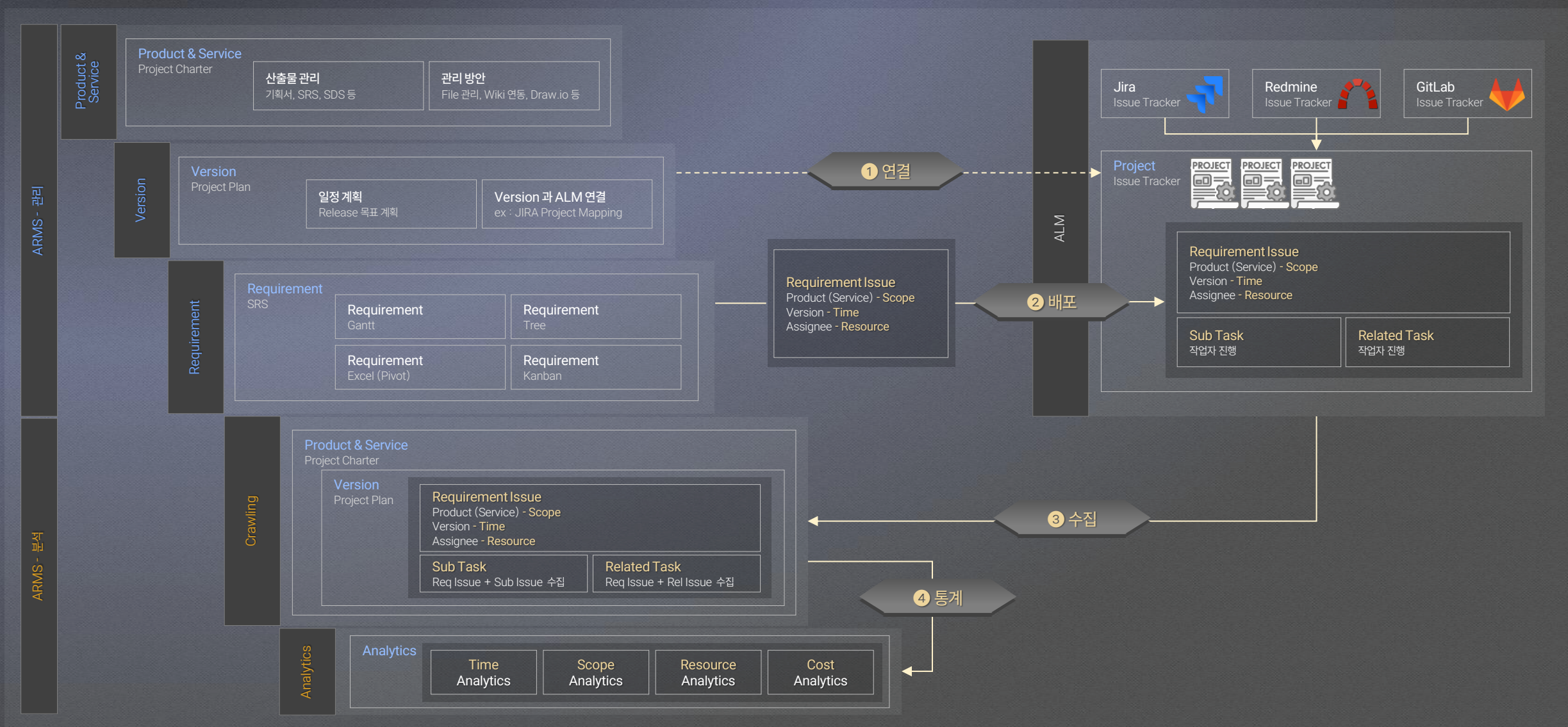
리스크의 ‘징조’를  
조기 식별



**Ai + ALM + RMS + PMS = A-RMS**

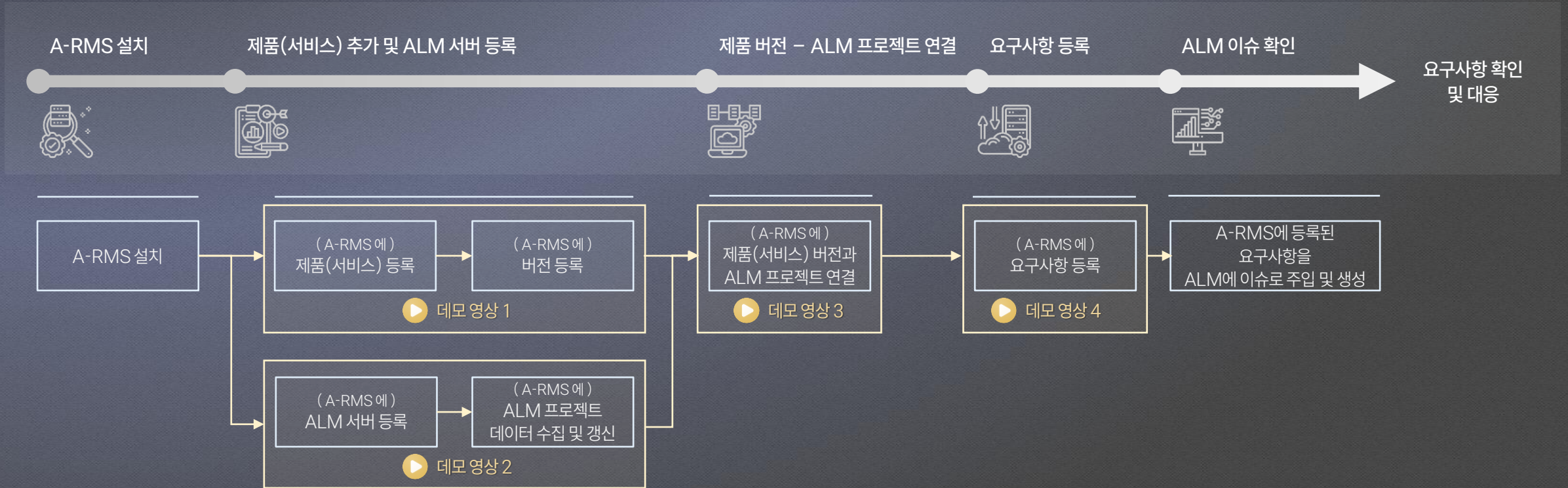
CONCEPT & OVERVIEW





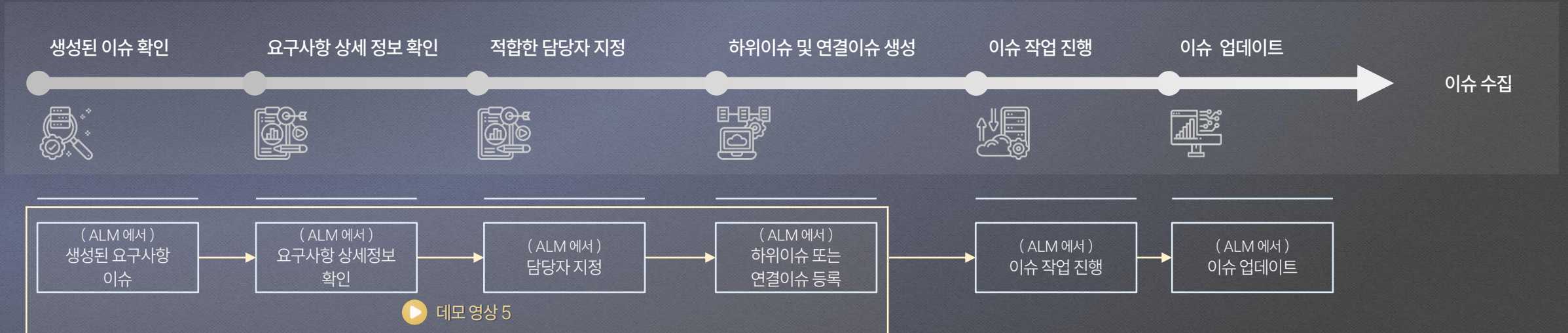


- 제품(서비스)를 등록하고, ALM을 연결합니다.
- 선택한 제품(서비스)의 요구사항을 등록하여, 이슈를 생성합니다.



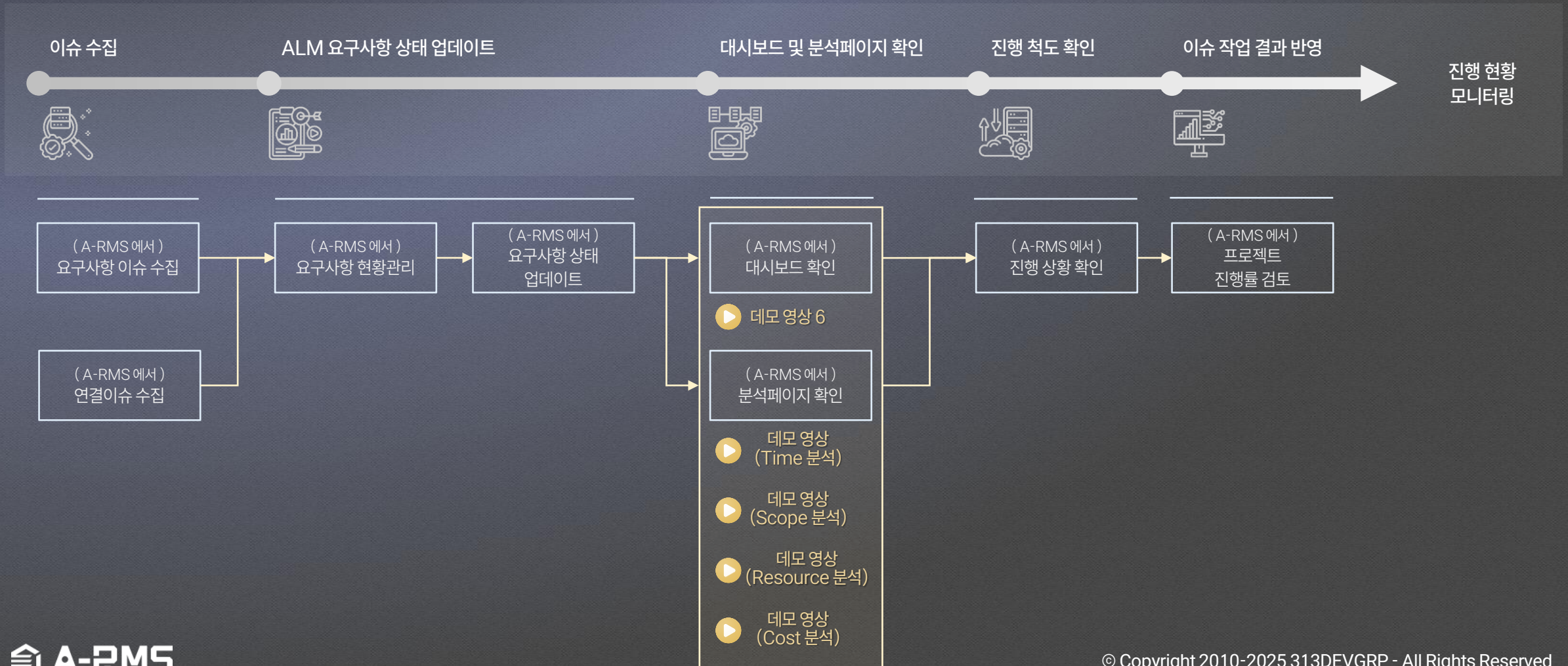


- 고객이 등록한 요구사항 이슈를 확인하고, 관련 담당자 지정합니다.
- 요구사항의 담당자는 이슈를 등록하고 관련 작업을 진행합니다.





- 수집된 이슈와 요구사항의 상태를 종합적으로 확인하여, 진행 현황에 대한 판단 및 피드백을 할 수 있습니다.







제품(서비스) 등록



버전 등록



ALM 등록



버전 - ALM 연결



요구사항 등록



요구사항 전파



요구사항 수집



분석 통계 리포트



{ 요구사항 등록 } + { 이슈사항 수집 } = 통계 분석 리포트





**Ai + ALM + RMS + PMS = A-RMS**

“ 단 하나의 규칙 : 작업자는 요구사항을 기준하여 일한다 ”

DEMO : <https://www.a-rms.net>  
ID : admin  
PW : qwe123





## 1. Dashboard

## 2 Product ( Service )

### 2.1 제품 관리

### 2.2 버전 관리

## 3 ALM ( Jira, Redmine, ... )

### 3,1 서버 관리

### 3.2 연결 관리

## 4. Requirement

### 4.1 요구 관리

### 4.2 현황 관리

### 4.3 간트 차트 ( Gantt )

### 4.4 칸반 보드 ( Kanban )

## 5. Analysis

### 5.1 일정 분석 Time

### 5.2 범위 분석 Scope

### 5.3 자원 분석 Resource

### 5.4 비용 분석 Cost

## 6. SWOT

## 7. ADOC



ARMS Dashboard 는 제품(서비스)를 기준으로 Time, Scope, Resource 에 대한 기본 분석을 제공합니다.





1. Dashboard

2 Product ( Service )

2.1 제품 관리

2.2 버전 관리

3 ALM ( Jira, Redmine, ... )

3,1 서버 관리

3.2 연결 관리

4. Requirement

4.1 요구 관리

4.2 현황 관리

4.3 간트 차트 ( Gantt )

4.4 칸반 보드 ( Kanban )

5. Analysis

5.1 일정 분석 Time

5.2 범위 분석 Scope

5.3 자원 분석 Resource

5.4 비용 분석 Cost

6. SWOT

7. ADOC



Product(Service) 는 회사의 제품 혹은 서비스를 등록하며, SRS, SDS, RFP 등의 산출물을 관리 할 수 있습니다.

Product.service Register

1. 제품(서비스) 선택

Export ▼

Help ?

제품(서비스) 편집

제품(서비스) 추가

Show 10 ▼ entries

Search: 🔍

제품(서비스) 이름

A-RMS

HyperMig

Showing 1 to 2 of 2 entries

2. 선택된 제품(서비스) 산출물 조회

Help ?

선택된 제품(서비스) : A-RMS

제품(서비스) 산출물 등록하기

ES Framework 컴포넌트 구조 초안

Es Framework 집계 쿼리 생성 구조

Es Framework 조회 쿼리 생성 구조

ES Framework 설계 GCP 스타일로 변환

ES Framework RecentTrue

3. 제품(서비스) 산출물을 확인하고 편집/삭제

Help ?

선택된 제품(서비스) 산출물 : ES Framework 컴포넌트 구조 초안

제품(서비스) 산출물 편집하기

제품(서비스) 산출물 저장

상세 보기

편집 하기

삭제 하기

제품(서비스) 이름 : A-RMS

제품(서비스) 산출물 이름 : ES Framework 컴포넌트 구조 초안

drawio : drawio 편집하기 가기

drawio 첨부파일 :

1 / 1

제품(서비스) 산출물 컨텐츠 :

제품(서비스)의 기획서 및 Project Charter의 내용을 기록합니다.







해답 : ARMS 는 요구사항을 ALM 이슈와 대응하기 위한 모델링 해답을 찾았습니다.







단 한번의  
설정 작업



1. Dashboard

2 Product ( Service )

2.1 제품 관리

2.2 버전 관리

3 ALM ( Jira, Redmine, ... )

3.1 서버 관리

3.2 연결 관리

4. Requirement

4.1 요구 관리

4.2 현황 관리

4.3 간트 차트 ( Gantt )

4.4 칸반 보드 ( Kanban )

5. Analysis

5.1 일정 분석 Time

5.2 범위 분석 Scope

5.3 자원 분석 Resource

5.4 비용 분석 Cost

6. SWOT

7. ADOC



연결 관리는 제품 (\*서비스)의 버전 - 에 ALM 프로젝트를 연결하여 요구사항을 관리 할 수 있는 설정 기능입니다.

단 한번의  
설정 작업



1. Dashboard

2 Product ( Service )

2.1 제품 관리

2.2 버전 관리

3 ALM ( Jira, Redmine, ... )

3,1 서버 관리

3.2 연결 관리

4. Requirement

4.1 요구 관리

4.2 현황 관리

4.3 간트 차트 ( Gantt )

4.4 칸반 보드 ( Kanban )

5. Analysis

5.1 일정 분석 Time

5.2 범위 분석 Scope

5.3 자원 분석 Resource

5.4 비용 분석 Cost

6. SWOT

7. ADOC



맵핑 관리는 ARMS의 4가지 설정 정보와 고객사의 임의적인 상태 그리고 ALM 의 이슈 상태를 Mapping 하는 기능을 제공합니다.

단 한번의  
설정 작업



해답 : ARMS 는 요구사항을 ALM 이슈와 대응하기 위한 모델링 해답을 찾았습니다.



단 한번의  
설정 작업



1. Dashboard

2 Product ( Service )

2.1 제품 관리

2.2 버전 관리

3 ALM ( Jira, Redmine, ... )

3.1 서버 관리

3.2 연결 관리

4. Requirement

4.1 요구 관리

4.2 현황 관리

4.3 간트 차트 ( Gantt )

4.4 칸반 보드 ( Kanban )

5. Analysis

5.1 일정 분석 Time

5.2 범위 분석 Scope

5.3 자원 분석 Resource

5.4 비용 분석 Cost

6. SWOT

7. ADOC



요구 관리는 제품(\*서비스)의 버전을 기준으로 생성됩니다. 각 요구사항은 ALM 의 요구사항 이슈로 전송되며, 요구사항 목록은 피벗형태로 가시성을 제공합니다.

**A-RMS Requirement Management**

요구사항 조회

1. 제품 서비스 선택 → 버전 선택 → 요구사항 선택

- ✓ 선택된 제품 서비스 : A-RMS
- ✓ 선택된 버전 : 선택되지 않음
- ✓ 선택된 요구사항 : [ARMS-REQ] ALM Issue 컬럼

Select Version

요구사항

- A-RMS 팀 구성 확인
- [ARMS-REQ] QA
- [ARMS-REQ] 기술 학습
- [ARMS-REQ] 경진 대회
- [ARMS-REQ] 백오피스
- [ARMS-REQ] ES Framework
- [ARMS-REQ] 서버 관리
- [ARMS-REQ] 마케팅
- [ARMS-REQ] 인증 및 자격
- [ARMS-REQ] 비즈니스 프로세스
- [ARMS-REQ] Tree Framework
- [ARMS-REQ] A-DOC
- [ARMS-REQ] 미팅 및 회의
- [ARMS-REQ] ES 전용 게시판 기능을 구현해 주세요
- [ARMS-REQ] ALM Issue 컬럼 추가 및 현황관리 보기 종류 추가(수정일기준)

요구사항 요약

요구사항 난이도 : 매우 어려움, 어려움, 보통, 쉬움, 매우 쉬움

요구사항 상태 : 열림, 진행중, 해결됨, 닫힘, 기타

요구사항 시작날짜 : 2025/04/08

요구사항 종료날짜 : 2025/04/11

요구사항 요청자 : [admin] - 9af24080-050d-4943-b40e-d789c0f976ee

요구사항 요청일 : 2025. 4. 8. 오후 3:34:53

요구사항 리뷰어 : 리뷰어 설정을 위한 계정명을 입력해 주세요



1. Dashboard

2 Product ( Service )

2.1 제품 관리

2.2 버전 관리

3 ALM ( Jira, Redmine, ... )

3.1 서버 관리

3.2 연결 관리

4. Requirement

4.1 요구 관리

4.2 현황 관리

4.3 간트 차트 ( Gantt )

4.4 칸반 보드 ( Kanban )

5. Analysis

5.1 일정 분석 Time

5.2 범위 분석 Scope

5.3 자원 분석 Resource

5.4 비용 분석 Cost

6. SWOT

7. ADOC



현황 관리는 제품(\*서비스)의 버전별로 요구사항 하위 및 연결 이슈의 상태 정보를 확인 할 수 있으며, 각 요구사항별로 ALM 접근 정보를 제공하는 기능입니다.

The screenshot displays the A-RMS interface, which is used for managing requirements and issues. The interface is divided into several sections:

- Service - Issue List:** A table showing requirements with columns for '요구사항 구분' (Requirement Category), 'ALM Issue Key', and 'Version'. The table lists various requirements, including '요구사항', 'ARMS-154의 하위이슈', and 'ARMS-18의 하위이슈'.
- Resource:** A section showing resource usage with a table of 'moon', '최소한', '홍성훈', '양형석', and 'dongmin lee'.
- Progress:** A section showing progress with a table of '완료', '진행 중', and '해미 할 일'.
- Statistics:** A section showing statistics with a table of '총 개종(서비스) 버전 개수', '총 요구사항 이슈 개수', '요구사항 이슈에 연결된 통계', '연결된 ALM 서버 개수', '연결된 ALM 프로젝트 개수', and '연결된 ALM 이슈 개수'.
- Issue List:** A table showing issues with columns for '요구사항', 'ALM Issue Key', and 'Version'. The table lists various issues, including '요구사항', 'ARMS-154의 하위이슈', and 'ARMS-18의 하위이슈'.
- Issue Detail:** A modal window showing details for a specific issue, including 'ALM Issue Title', 'ALM Issue Key', 'ALM Project', 'Issue Status', 'Issue Priority', 'Issue Assignee', 'Issue Reporter', 'ALM Created', and 'ALM Updated'.
- Issue List (Bottom):** A table showing issues with columns for 'ALM Issue Type', 'ALM Priority', 'ALM Status', and 'ALM Overall Up'.



1. Dashboard

2 Product ( Service )

2.1 제품 관리

2.2 버전 관리

3 ALM ( Jira, Redmine, ... )

3.1 서버 관리

3.2 연결 관리

4. Requirement

4.1 요구 관리

4.2 현황 관리

4.3 간트 차트 ( Gantt )

4.4 칸반 보드 ( Kanban )

5. Analysis

5.1 일정 분석 Time

5.2 범위 분석 Scope

5.3 자원 분석 Resource

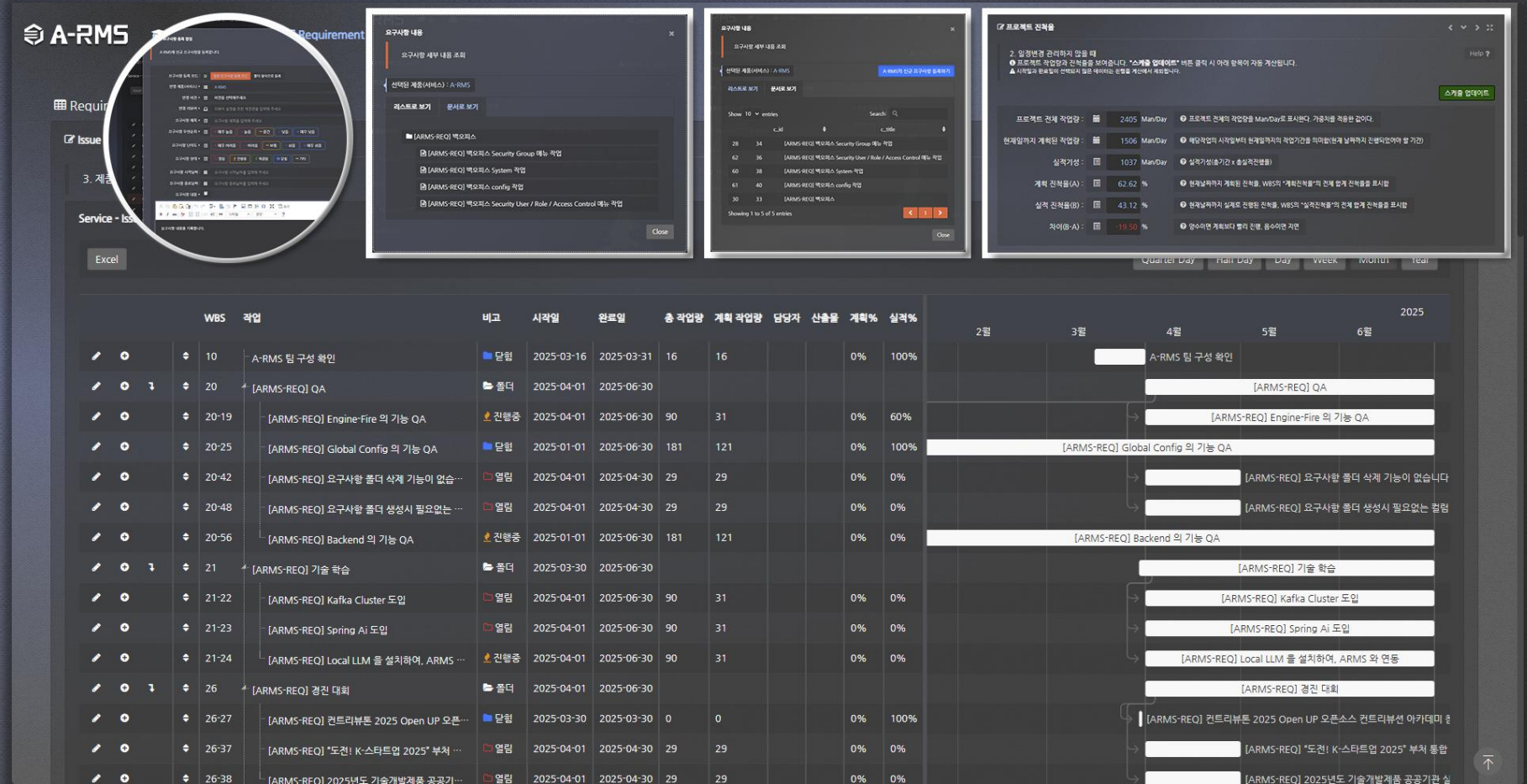
5.4 비용 분석 Cost

6. SWOT

7. ADOC



간트 차트 ( Gantt ) 는 요구사항을 관리하고 일정을 수립하며, 포함 관계를 설정하는 등의 기능을 제공합니다.







칸반 보드는 요구사항의 상태를 쉽게 변경하기 위한 용도로 제공하는 기능입니다.

**A-RMS**

[Ai](#)
[Product](#)
[Requirement](#)
[Management](#)
[System](#)

---

### Requirement Kanban

Service - Version 선택

1. 제품(서비스) 선택

\* 제품(서비스) - 버전 선택

1. 제품 서비스 선택 → 버전 선택

- ✓ 선택된 제품 서비스 : A-RMS
- ✓ 선택된 버전 : 2025년 4분기 ( GitLab 활용 ), 2025년 3분기 ( Redmine 활용 ), 2025년 2분기, 2025년 1분기, BaseVersion

A-RMS

All selected

### 칸반 보드

2. 칸반 보드를 통해 요구사항 상태 관리

🔥 : 36.7%
💡 : 41.7%
🟢 : 1.7%
📁 : 20%
📊 : 0%

열림	진행중	해결됨	닫힘	기타
[ARMS-REQ] Kafka Cluster 도입	[ARMS-REQ] A-RMS 와 A-DOC 연동 기능	[ADOC-REQ] 삼성에서 만든 프로젝트 뷰 기능 목록 작성 요청	A-RMS 팀 구성 확인	
[ARMS-REQ] 이석주 교수님 미팅 4월 1주 중	[ARMS-REQ] ( Resource ) Expand 기능을 만들어 주세요	[ADOC-REQ] 삼성에서 만든 프로젝트 뷰 기능 목록 작성 요청	[ARMS-REQ] 컨트리뷰톤 2025 Open UP 오픈소스 컨트리뷰션 아카데미 ...	
[ARMS-REQ] Spring Ai 도입	[ARMS-REQ] 서버 관리 ( Server 모듈 및 IaC 등 )		[ARMS-REQ] 데이터는 최신 업데이트 일자로 정렬	
[ARMS-REQ] "도전! K-스타트업 2025" 부처 통합 창업경진대회	[ARMS-REQ] Engine-Fire 의 기능 QA		[ARMS-REQ] 상표권 등록	
[ARMS-REQ] 2025년도 기술개발제품 공공기관 실용지원 사업	[ARMS-REQ] 백오피스 Security Group 메뉴 작업		[ARMS-REQ] Global Config 의 기능 QA	
[ARMS-REQ] 요구사항 폴더 삭제 기능이 없습니다.	[ARMS-REQ] 랜덤 페이지를 완성 해주세요		[ARMS-REQ] dmove 1차 컨퍼런스 후속 조치	
	[ARMS-REQ] ( Time )		[ARMS-REQ] 세근님이랑 프로젝트뷰 ( 삼성 ) 리뷰	

### Statistics

2. 제품(서비스) 요구사항 통계

- ✓ 총 요구사항 개수 : 60 개
- ✓ 열린 요구사항 개수 : 22 개
- ✓ 진행 중인 요구사항 개수 : 25 개
- ✓ 해결된 요구사항 개수 : 1 개
- ✓ 닫힌 요구사항 개수 : 12 개
- ✓ 기타 요구사항 개수 : 0 개



해결 : ARMS 는 ALM 의 구조와 일치하면서도, 요구사항 관리에 기능이 집중되어 있습니다.





1. Dashboard

2 Product ( Service )

2.1 제품 관리

2.2 버전 관리

3 ALM ( Jira, Redmine, ... )

3.1 서버 관리

3.2 연결 관리

4. Requirement

4.1 요구 관리

4.2 현황 관리

4.3 간트 차트 ( Gantt )

4.4 칸반 보드 ( Kanban )

5. Analysis

5.1 일정 분석 Time

5.2 범위 분석 Scope

5.3 자원 분석 Resource

5.4 비용 분석 Cost

6. SWOT

7. ADOC



일정 분석 ( Time ) 은 발행된 요구사항 이슈의 하위 및 연결 이슈를 수집하여 일정별로 분석한 결과를 제공하는 기능입니다.





1. Dashboard

2 Product ( Service )

2.1 제품 관리

2.2 버전 관리

3 ALM ( Jira, Redmine, ... )

3,1 서버 관리

3.2 연결 관리

4. Requirement

4.1 요구 관리

4.2 현황 관리

4.3 간트 차트 ( Gantt )

4.4 칸반 보드 ( Kanban )

5. Analysis

5.1 일정 분석 Time

5.2 범위 분석 Scope

5.3 자원 분석 Resource

5.4 비용 분석 Cost

6. SWOT

7. ADOC



범위 분석 ( Scope ) 은 발행된 요구사항 이슈를 기반으로 버전 범위 및 제품의 범위를 분석한 결과를 리포트하는 기능 입니다.





1. Dashboard

2 Product ( Service )

2.1 제품 관리

2.2 버전 관리

3 ALM ( Jira, Redmine, ... )

3.1 서버 관리

3.2 연결 관리

4. Requirement

4.1 요구 관리

4.2 현황 관리

4.3 간트 차트 ( Gantt )

4.4 칸반 보드 ( Kanban )

5. Analysis

5.1 일정 분석 Time

5.2 범위 분석 Scope

5.3 자원 분석 Resource

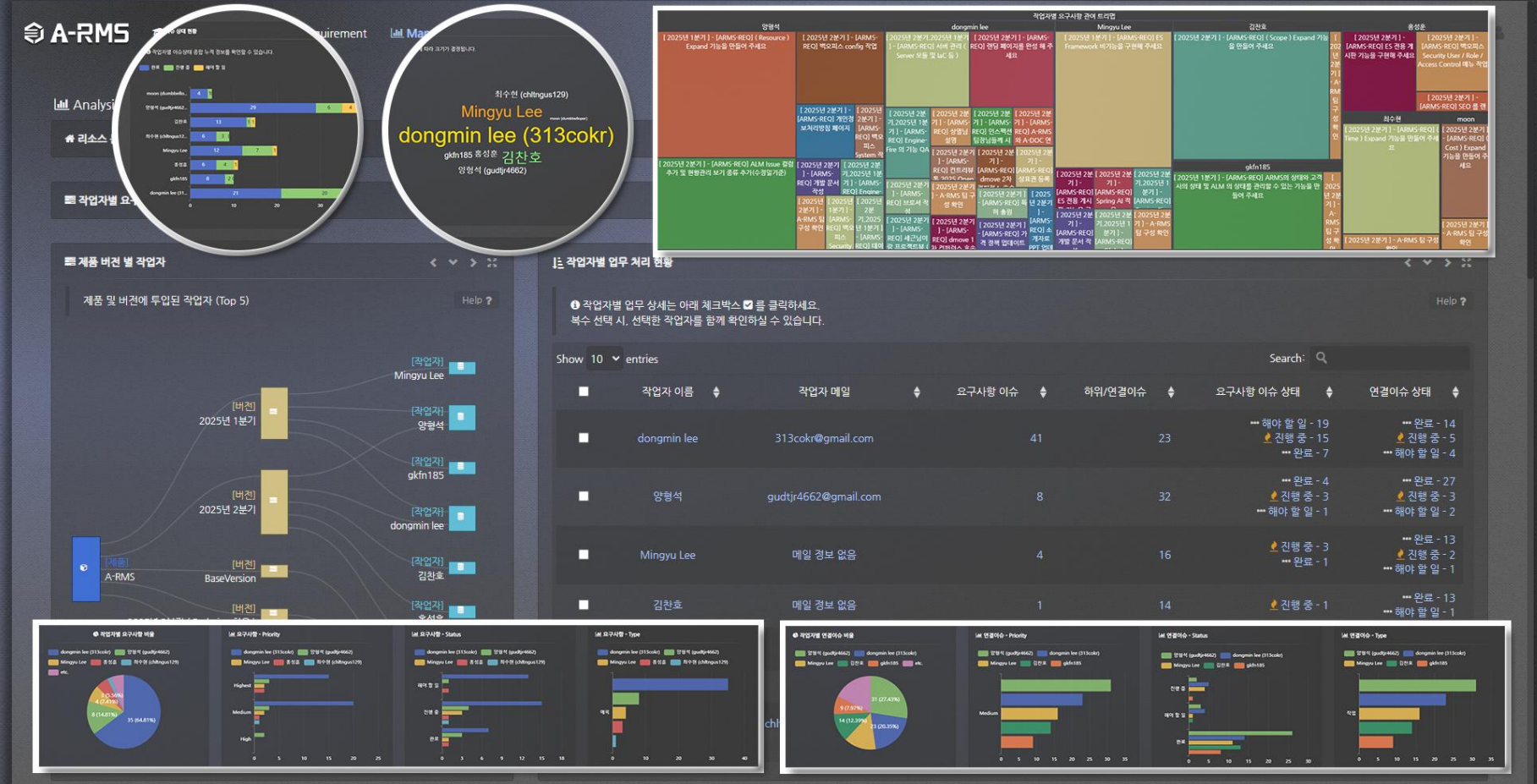
5.4 비용 분석 Cost

6. SWOT

7. ADOC



자원 분석 ( Resource ) 은 ALM의 할당자를 수집하여 이슈 기반으로 분석된 퍼포먼스를 측정 및 통계 기능을 제공합니다.





- ## 1. Dashboard

## 2 Product ( Service )

## 2.1 제품 관리

## 2.2 버전 관리

### 3 ALM ( Jira, Redmine, ...)

### 3,1 서버 관리

### 3.2 연결 관리

## 4. Requirement

## 4.1 요구 관리

## 4.2 현황 관리

### 4.3 간트 차트 ( Gantt )

#### 4.4 칸반 보드 ( Kanban )

## 5. Analysis

## 5.1 일정 분석 Time

## 5.2 범위 분석 Scope

### 5.3 자원 분석 Resource

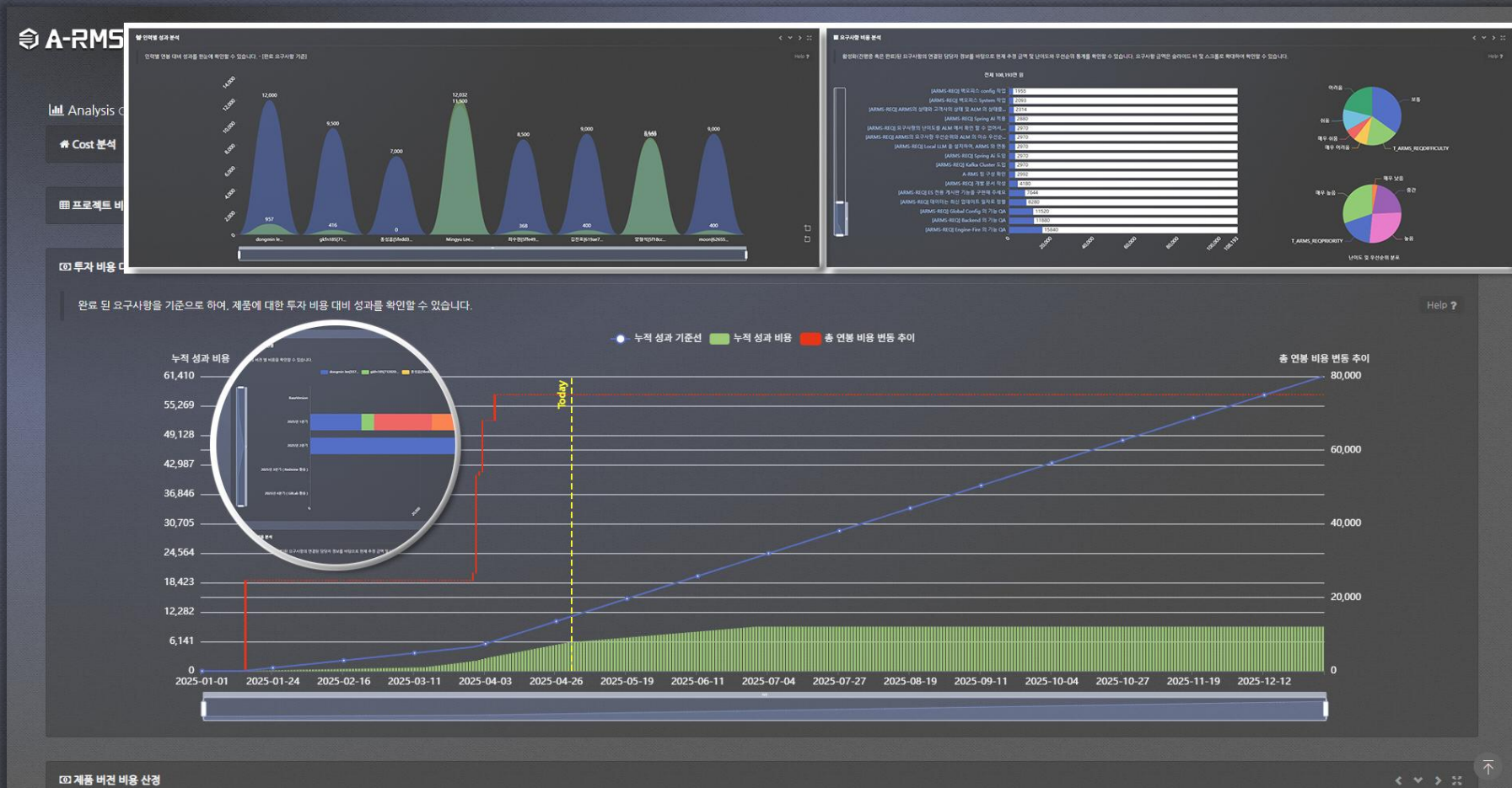
## 5.4 비용 분석 Cost

## 6. SWOT

## 7. ADOC



비용 분석 ( Cost ) 는 제품(\*서비스) 별, 버전 별, 요구사항 별 비용을 역산하여 계산된 결과를 제공해 드리는 기능입니다.



#### ㉑ 제품 버전 비용 산정



1. Dashboard

2 Product ( Service )

2.1 제품 관리

2.2 버전 관리

3 ALM ( Jira, Redmine, ... )

3.1 서버 관리

3.2 연결 관리

4. Requirement

4.1 요구 관리

4.2 현황 관리

4.3 간트 차트 ( Gantt )

4.4 칸반 보드 ( Kanban )

5. Analysis

5.1 일정 분석 Time

5.2 범위 분석 Scope

5.3 자원 분석 Resource

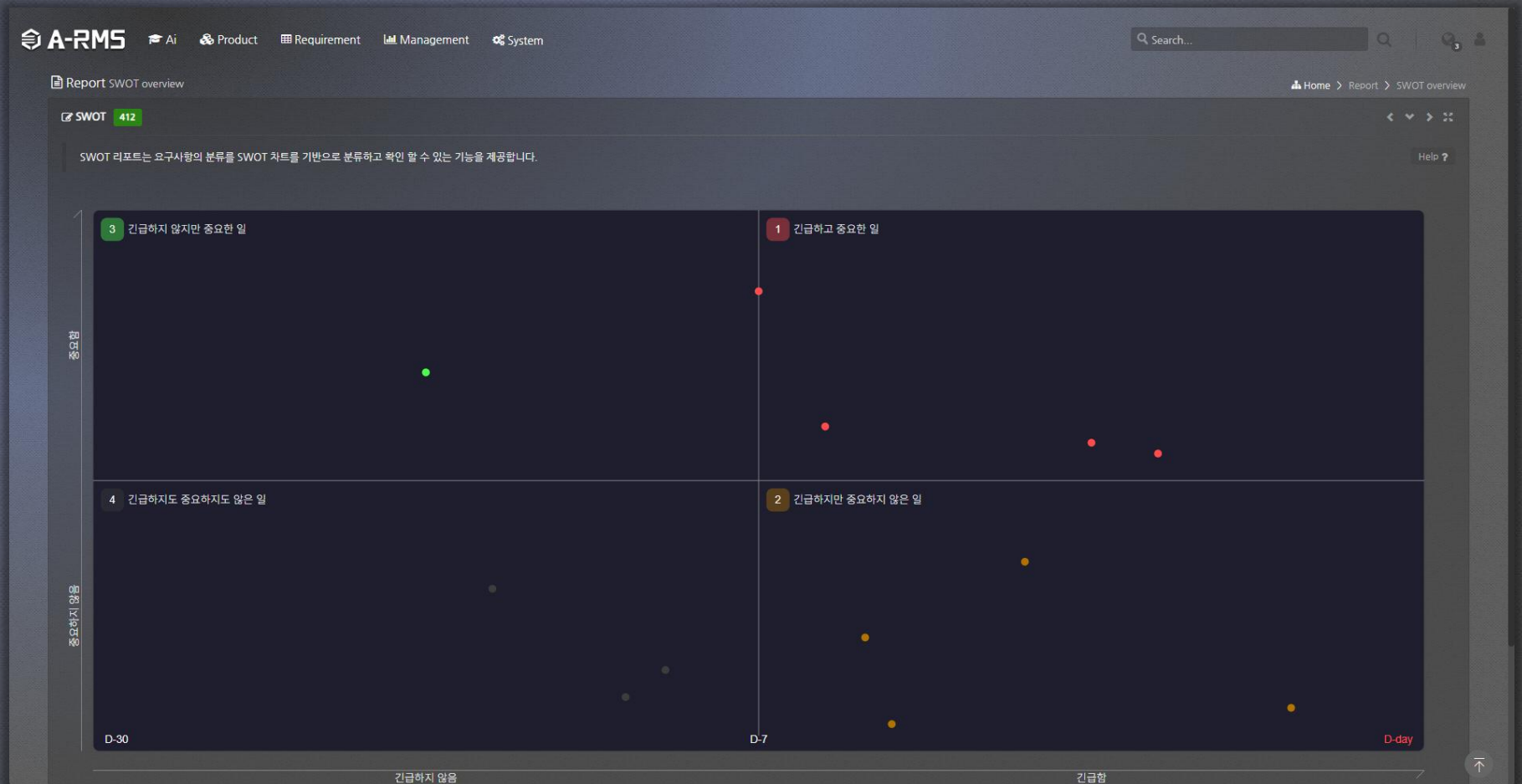
5.4 비용 분석 Cost

6. SWOT

7. ADOC



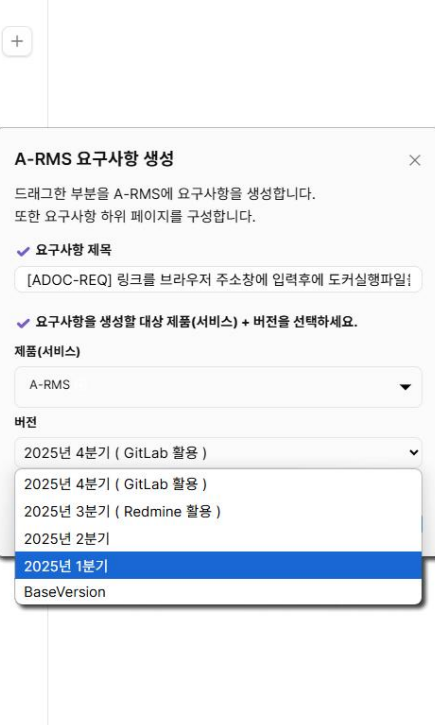
SWOT 리포트는 요구사항의 분류를 SWOT 차트를 기반으로 분류하고 확인할 수 있는 기능을 제공합니다.







ADOC는 A-RMS 사용자만을 위한 문의사항, 설치 가이드, 릴리즈 노트, 콘텐츠 자료를 통합한 전용 게시판 허브를 제공합니다.



## 3. Docker 설치 (offline)

업데이트됨 2025년 4월 17일

백링크 · 1

정보

**3** 링크를 브라우저 주소창에 입력후에 도커실행파일을 로컬PC에 다운 받는다.

[https://download.docker.com/linux/static/stable/x86\\_64/docker-23.0.3](https://download.docker.com/linux/static/stable/x86_64/docker-23.0.3)

### 3.2 Docker 설치 파일 서버에 복사

Docker실행파일을 설치할 서버에 복사한다.

```
1 /*일반 접근 방식*/
2 scp docker-24.0.3.tgz [계정]@[서버주소]:[home디렉토리]
```

```
1 /*인증서파일 접근 방식*/
2 scp -i [인증서파일] docker-24.0.3.tgz [계정]@[서버주소]:[home디렉토리]
```

```
$ scp -i mad-dev.pem ./docker-24.0.3.tgz ubuntu@172.31.3.23:/home/ubuntu
docker-24.0.3.tgz      100% 66MB  5.9MB/s  00:11
```

### 3.3 Docker 실행 파일 압축 해제

Docker의 APT 저장소를 Ubuntu의 패키지 소스 목록에 추가하여, 이후 apt update 명령을 통해 Docker 패키지를 다운로드하고 설치할 수 있도록 설정합니다.

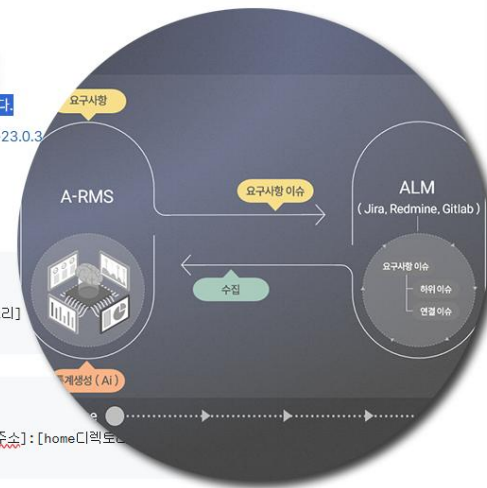


Table of Contents

- 3. Docker 설치 (offline)
- 3.1 Docker 실행파일 다운
- 3.2 Docker 설치 파일 서버에 복사
- 3.3 Docker 실행 파일 압축 해제
- 3.4 Docker 서비스 등록...
- 3.5 Docker 서비스 파일 ...
- 3.6 Docker 서비스 파일 ...
- 3.7 Docker 실행
- 3.8 Docker 서비스 등록



**Ai + ALM + RMS + PMS = A-RMS**

CUSTOMER OFFERING



분석 규모에 따른 구분 및 서버 구성에 대한 내용

항목	사양		
수집 분석 규모	사용자 100명 이하	사용자 300명 이하	사용자 300명 이상
OS	Docker Runtime 활용으로 서버 OS 상관 없음 (Linux 권장)		
서버 수량	1 EA ( Standalone )	1 EA ( Cluster 로 분리 가능 )	1 EA ( Cluster 필수 )
CPU	4 Core	8 Core	16 Core 이상
메모리	32 GB	64 GB	128 GB 이상
HDD	150 GB ( Log 추가 수집 시 : 300 GB )	250 GB ( Log 추가 수집 시 : 500 GB )	500 GB 이상 ( Log 추가 수집 시 : 1 TB 이상 )
설치 방식	Install Script 제공		
준비사항	ALM Issue Tracker ( JIRA, Redmine, Gitlab ) 별 API Token 권한 필요		
	Unix root 권한 / Docker Engine 설치	Linux sudo 권한 / Docker Engine 설치	Windows administrator 권한 / Docker Engine 설치
	ARMS 서버와 ALM Issue Tracker 서버 간에 각 API 방화벽 포트 Open 필요		
	API http port 80, https port 443	SSL 적용시 ARMS 인증서 사용 목적 : crt, pem 파일 필요	Cloud 적용시 ALB 활용 – SSL을 적용하더라도 인증서 제공 필요



라이선스 타입에 따른 설치 모듈 및 기능 스펙 정의

### Standalone 필수 모듈 ( 24 )

#### A-RMS MSA ( 8 )

nginx - Frontend App 서버 모듈

backend-core - API 처리 모듈

engine-fire - 수집 및 분석 모듈

global-config - MSA 설정 모듈

middle-proxy - 인증/인가 보안

ai - 내부 인공지능 데이터 추론 모듈

mysql - Main Database

redis - Memory Database

#### Support Utility ( 8 )

affine - Wiki

apachephp - Web Analysis

fluentd - Docker Logging

keycloak - SSO Server

ollama - Ai Server

gitea - Git Server

vault - 데이터 암호화 모듈

postgres - Sub Database

#### 검색 엔진 ( 8 )

es-coordinating - API 처리기

es-ingest - 데이터 전처리기

es-master-01 - 분산 처리기 1

es-master-02 - 분산 처리기 2

es-master-03 - 분산 처리기 3

es-data-01 - 분산 데이터 처리 1

es-data-02 - 분산 데이터 처리 2

es-data-03 - 분산 데이터 처리 3

### Cluster 필수 모듈 ( 4 )

#### Message Queue ( 4 )

Kafka00Service - 분산 큐 1

Kafka01Service - 분산 큐 2

Kafka02Service - 분산 큐 3

kafka-web-ui - 분산 큐 Admin

### Cluster 권장 모듈 ( 7 )

#### Monitoring ( 7 )

agent - 노드 모니터링 에이전트

cadvisor - 컨테이너 에이전트

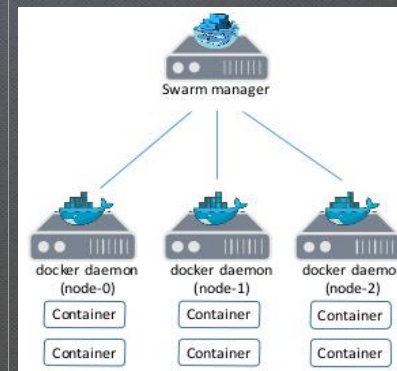
node-exporter - 호스트 모니터링

portainer - swarm 관리 콘솔

prometheus - 시계열 DB

grafana - 모니터링 웹 콘솔

kibana - 검색엔진 웹 콘솔





### A-RMS standard



제품·서비스 및 버전 관리  
ALM (\*Issue Track) 연결 관리



요구사항 관리 및 ALM 전파  
간트·칸반 활용 관리



( 일정, 범위, 자원, 비용 ) 분석  
전체 데이터 검색 활용

### A-RMS PREMIUM



\*Standard 모든 기능 포함



고성능 서비스 지원 ( HA )



연중 무휴 지원 ( Ticket Service )

### A-RMS Enterprise



\*Premium 모든 기능 포함



소프트웨어 수정 ( 통합 서비스 )



담당자 전담 관리

※ 300인 이상 솔루션 사용 희망 시 별도 협의



※ 추가 개발 별도 (M/M) , VAT 별도

요금제 구분		50명 + 5 Executive	100명 + 10 Exec	200명 + 20 Exec	300명 + 30 Exec	500명 + 50 Exec	1,000명 + 100 Exec
License	JIRA Standard Redmine Business GitLab Premium	6,000,000	12,000,000	24,000,000	36,000,000	60,000,000	120,000,000
	ARMS Standard	3,000,000	6,000,000	12,000,000	18,000,000	30,000,000	60,000,000
	JIRA Premium Redmine Platform GitLab Ultimate	10,800,000	21,600,000	43,200,000	64,800,000	108,000,000	216,000,000
	ARMS Premium	5,000,000	10,000,000	20,000,000	30,000,000	50,000,000	100,000,000
	JIRA Enterprise Redmine Enterprise	19,440,000	38,880,000	77,760,000	116,640,000	194,400,000	388,800,000
	ARMS Enterprise	9,000,000	18,000,000	36,000,000	50,000,000	90,000,000	150,000,000
Service	초기 구축, 교육비 별도	1,000,000	2,000,000	4,000,000	6,000,000	10,000,000	20,000,000



**Ai + ALM + RMS + PMS = A-RMS**

ROADMAP



질문 : 프로젝트 취소, 납기 지연, 비용 초과, 프로젝트 실패의 『근본적인 원인은 』무엇이라고 생각하십니까?



2025 개발 상황

구분	Scope 분석	Time 분석	Resource 분석	통합 분석		ALM 연동			SCM 연동 ( git , svn ... )
				Cost 분석	분석 Report	JIRA	Gitlab	Redmine	
제품*서비스	○	○	○	○	진행 중	○	예정	○	예정
버전	○	○	○	○	진행 중	○	예정	○	예정
단위 요구사항	○	○	○	○	진행 중	○	예정	○	예정

※ 일\*주\*월간 리포트 , Bull’s Eye Chart ( 예측 시스템 ) , 그룹 단위 비교 분석 시스템 , SWOT 분석 , 요구사항 현황

개발 영역

서비스 가능



고민 : 작업자의 퍼포먼스를 종합적으로 분석하여, 정량적인 절대 지표 및 상대 지표를 제공 할 예정입니다.

### 작업자 분석

제품\*(서비스) 기여도

▶ 회사의 이익 창출 기반 기여도 분석

버전 기여도

▶ 프로젝트 기반 기여도 분석

요구사항 관여도

▶ 팀 기반 성과 기여도 분석

ALM 이슈 데이터 기반 분석

▶ 정성적 평가가 아닌 정량적 평가 기반 분석

SCM 코드레벨 데이터 기반 분석

▶ 코드 레벨의 품질 및 QA 영향도 분석레벨 평가





고민 : 퍼포먼스 측정은 정량적 데이터를 기반하더라도, 절대적 기준을 마련하기 어렵습니다.

### 작업자간 비교 분석

경력 및 연봉 대비 작업자간 제몹\*(서비스) 기여도

▶ 회사의 이익 창출 기반 기여도 비교 분석

경력 및 연봉 대비 작업자간 버전 기여도

▶ 프로젝트 기반 기여도 비교 분석

경력 및 연봉 대비 작업자간 요구사항 관여도

▶ 팀 기반 성과 기여도 비교 분석

경력 및 연봉 대비 작업자간 ALM 이슈 데이터 기반 분석

▶ 정성적 평가가 아닌 정량적 평가 기반 비교 분석

경력 및 연봉 대비 작업자간 SCM 코드레벨 데이터 기반 분석

▶ 코드 레벨의 품질 및 QA 영향도 분석레벨 비교 평가





# A-RMS 로드맵 : ( 추가될 기능 ) 팀 분석

동민님,  
Resource 분석에서 세부 내용 설명이 없어 추가해 두었습니다.  
기존 : 팀원 구성에 대한  
수정 : 팀원 구성에 대한 업무 투입 현황 분석

## IV. Roadmap

고민 : 팀 단위의 분석 시 팀원의 데이터를 합산하여 팀의 성과 및 퍼포먼스를 측정합니다.

### 팀 단위분석

제품\*(서비스) : Scope 분석

▶ 어느 제품(서비스)를 맡아서 진행하고 있는지

버전 : Time 분석

▶ 기간 내 요구사항 관리 상태 분석

팀원 : Resource 분석

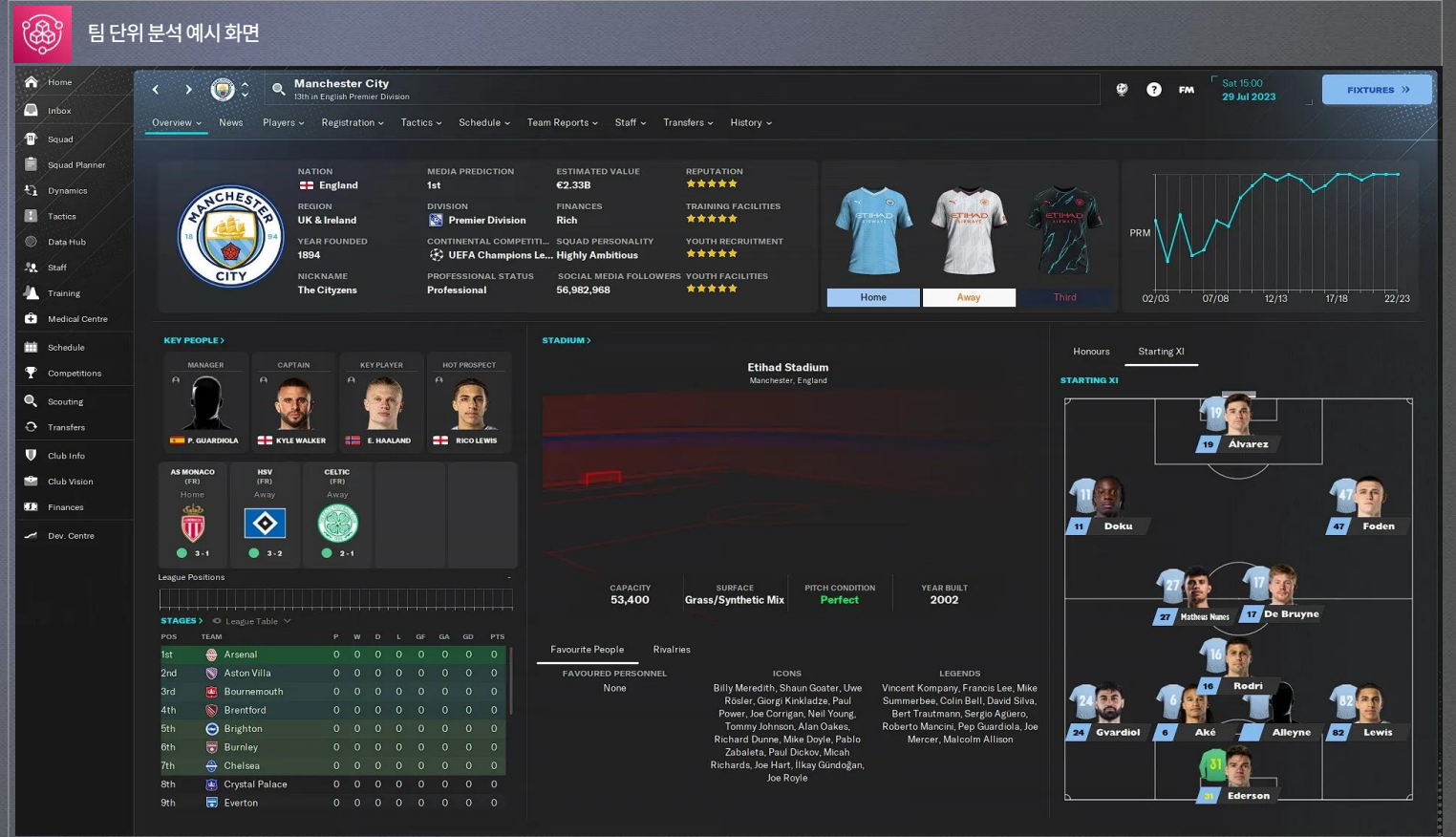
▶ 팀원 구성에 대한 업무 투입 현황 분석

ALM 기반 분석 : Performance 분석

▶ 정성적 평가가 아닌 정량적 평가 기반 분석

SCM & QA 기반 분석 : Quality 분석

▶ 코드 레벨의 품질 및 QA 영향도 분석레벨 평가





고민 : 팀 단위의 분석 시 팀원의 데이터를 합산하여 팀의 성과 및 퍼포먼스를 측정합니다.

### 실 or 그룹 단위 분석

제품\*(서비스) : Scope 분석

▶ 몇 개의 제품에 어떤 역할을 하는지

버전 : Time 분석

▶ 기간 별 역할의 추이 분석

실 or 그룹원 : Resource 분석

▶ 실 or 그룹 구성원 전체 Man Power 분석

ALM 기반 분석 : Performance 분석

▶ 제품의 요구사항 기반 : 기여도 분석

SCM & QA 기반 분석 : Quality 분석

▶ 코드 레벨의 품질 및 QA 영향도 분석





**Ai + ALM + RMS + PMS = A-RMS**

ARCHITECT & PLATFORM



### SW Product Line Engineering 방법론 채택

#### PLE 정의

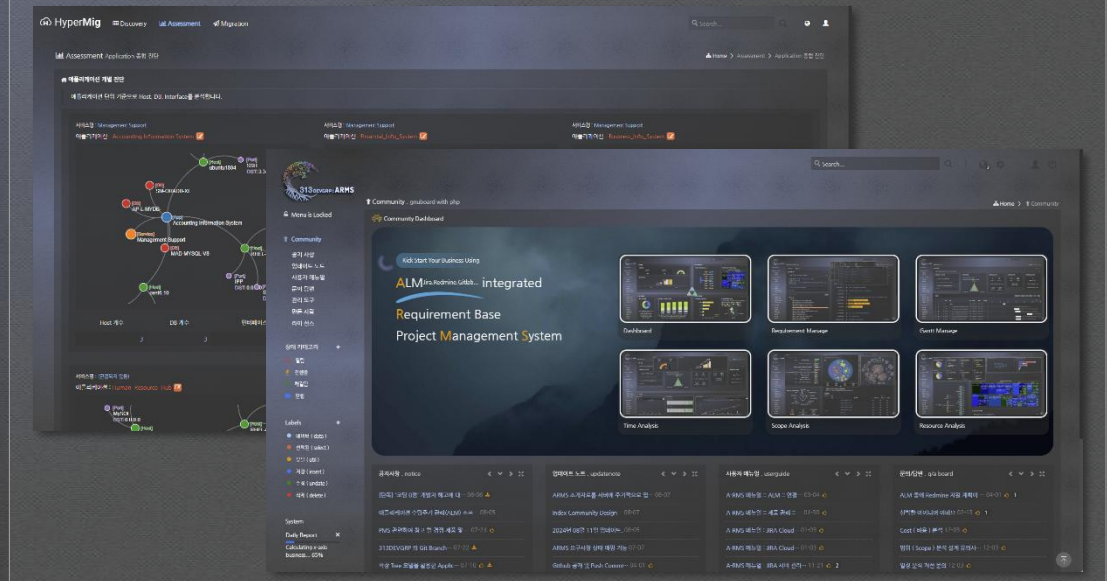
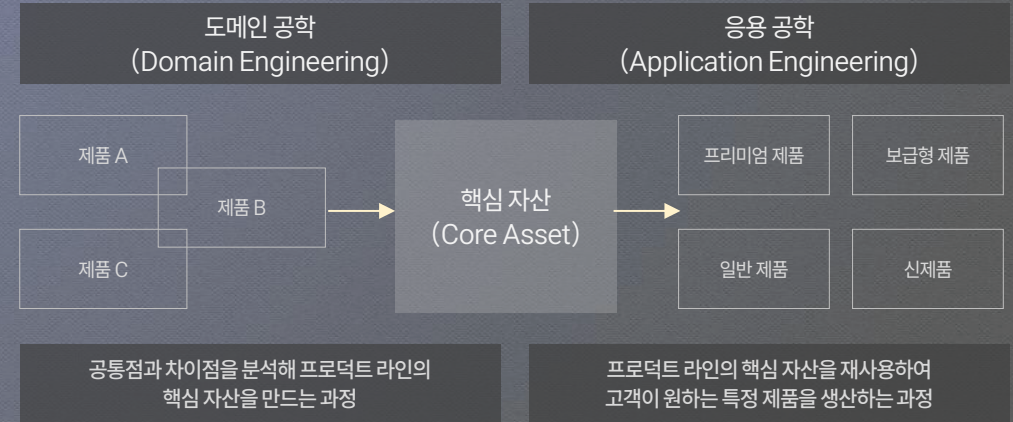
- ▶ 소프트웨어를 개발할 때 체계적인 재사용 기법을 적용함으로써 동일 영역에서 다양하게 특화된 소프트웨어를 신속하게 개발할 수 있는 효과적인 매커니즘을 제공하는 접근 방법

#### PLE 배경

- ▶ 컴포넌트 기반의 개발 자체가 재사용성을 높여주지 않음
- ▶ 재사용성을 고려해 주의 깊게 개발하지 않으면 컴포넌트를 재사용하기 어려움

#### PLE 구성

- ▶ 핵심자산(Core Asset) 개발  
: 여러 어플리케이션들의 공통적인 Feature을 핵심자산으로 모델링하고, 이를 재사용할 수 있도록 구체화하는 작업
- ▶ 어플리케이션 공학(Application Engineering)  
: 핵심자산과 통합하여 어플리케이션을 개발함
- ▶ 관리활동(Management)  
: 기술적 측면과 관리적 측면에서 핵심 자산 개발과 제품 개발 활동을 관리





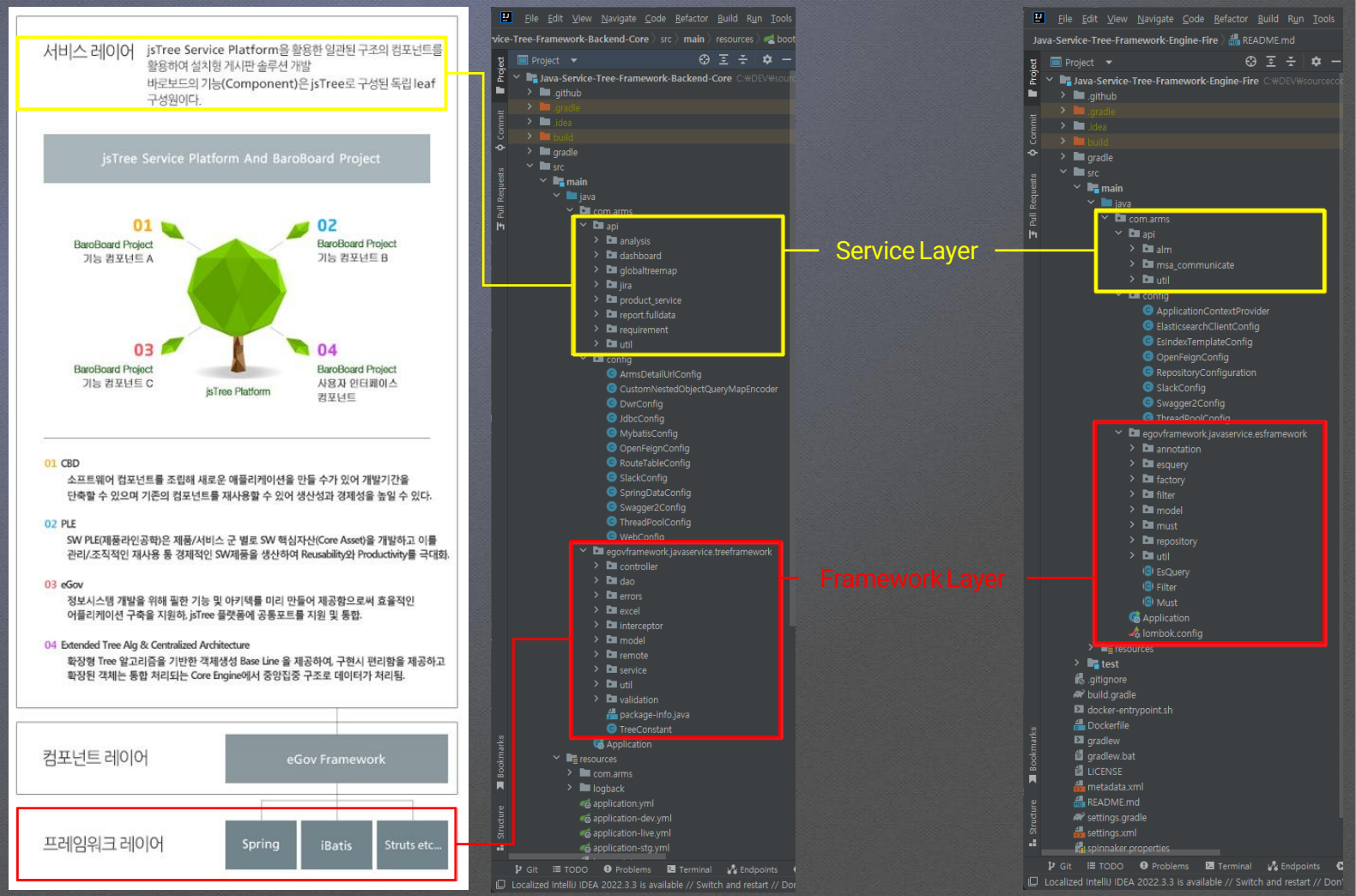
Service Layer = UTIL + Framework

### PLE ( Service Framework) 특징

- ▶ 같은 계열의 소프트웨어들에서 공통적인 특징들을 추출하여 고품질의 재사용 가능한 핵심 자산을 만들고 그 안에서 특화될 수 있는 부분을 가변적인 특성으로 모델링함으로써 특정 영역에서 어플리케이션을 개발할 때보다 높은 재사용성을 제공함
- ▶ 공통으로 유사한 기능을 가지고 있는 시스템의 집합으로 특정 영역의 요구사항을 만족하는 미리 구축된 핵심 자산으로부터 정해진 방식으로 개발
- ▶ 제품 계열 아키텍처의 재사용성을 최대한 높이도록 모델링하는 것이 중요QA 영향도 분석

### Service Framework 란

: 비즈니스 코드를 개발하기 위한 일정한 요소와 규약





## Data Structure Base Framework

RDB 의 데이터 적재 구조체 지원

▶ Tree Data Struct 를 활용한 릴레이션 지원

Memory DB 의 데이터 적재 지원

▶ 고속 데이터를 지원하기 위한 Key - Value 타입의 데이터 구조체 지원

NoSQL Document 데이터 적재 지원

▶ 시계열 데이터 적재를 위한 검색엔진을 활용

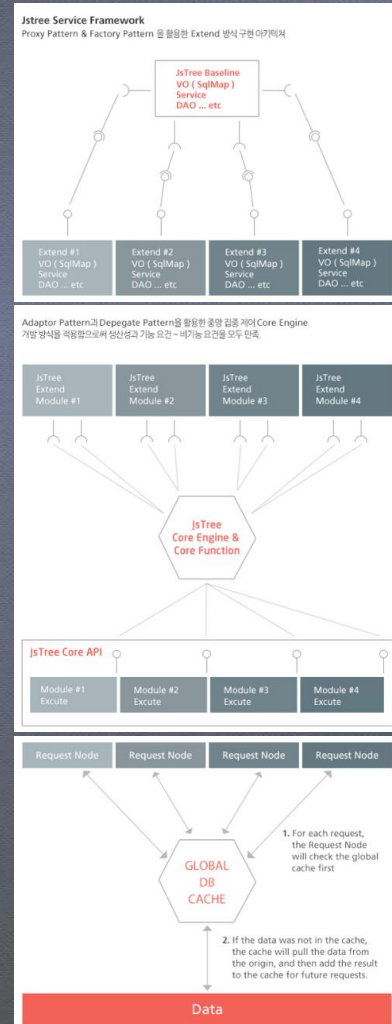
▶ 문서 구조의 단일 Entity 기반의 통계 데이터 활용

목적에 맞는 데이터 적재 방식 지원

▶ 데이터의 특성에 맞는 데이터 적재를 지원

▶ 적재를 위한 유틸 및 규약을 제공

▶ 일관된 비즈니스 코드를 유지





## PLE CODE 구성요소

### Core Asset

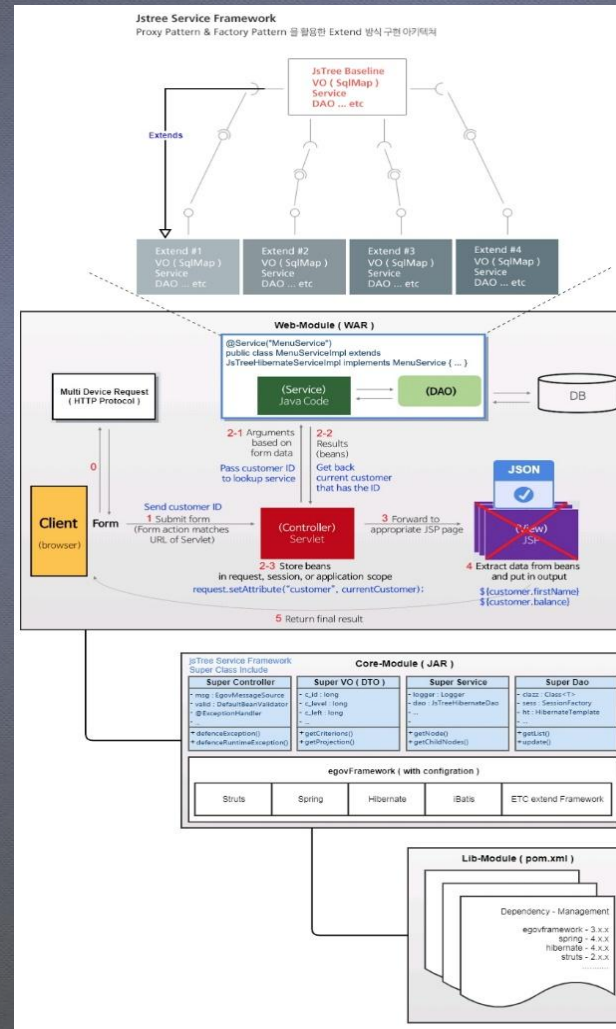
- ▶ 데이터 자료 구조에 기반한 프레임워크의 설정, 유틸 및 활용할 수 있는 API 를 제공하여 모델링을 단순화하고 재사용할 수 있도록 구체화된 코드 라이브러리

### Product Development

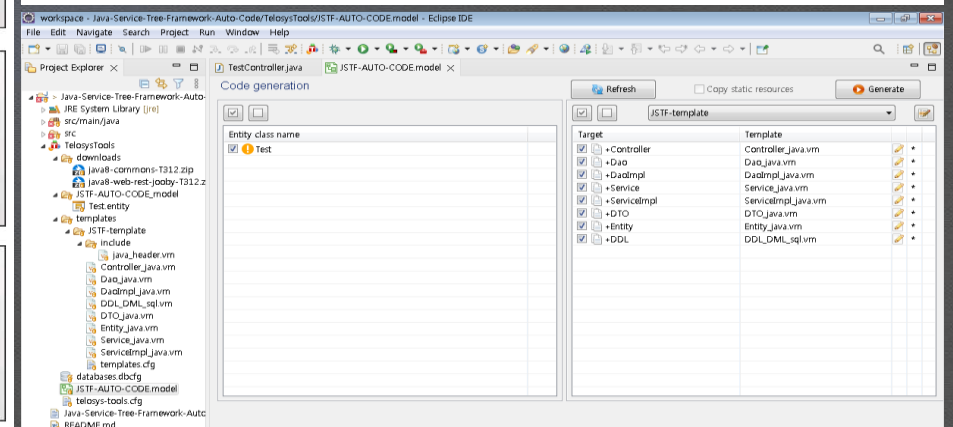
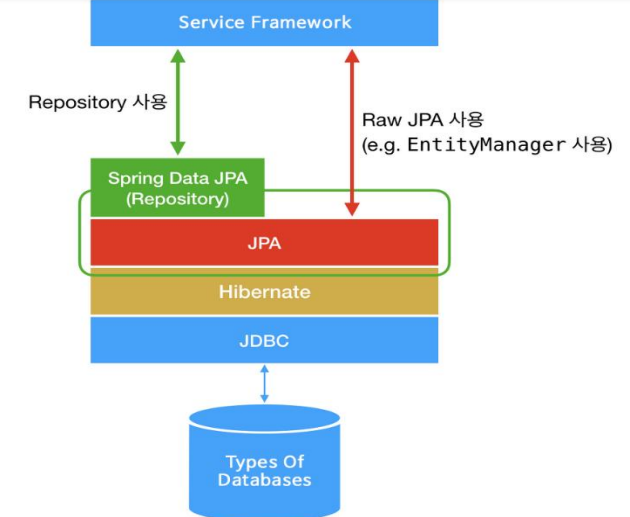
- ▶ Core Asset에서 제공하지 않는 목표 어플리케이션의 일부 기능을 모델링하며 Instance화된 Core Asset과 통합하여 어플리케이션을 개발

### Management

- ▶ 플랫폼 개발과 제품 적용/개발의 분리로 인하여 Template 코드 제공 유틸이 필요



## Business API CODE





### ALL-IN-ONE PLE Development Compose

#### ALM Env.

▶ 소프트웨어의 시작과 종료까지의 프로세스 전반을 미리 정의된 활용가능한 Tool Chain 제공

#### CICD Env.

▶ 빌드와 배포까지의 대상별로 ( OnPremise, Cloud ) 미리 정의된 Config 제공

#### Development Env.

▶ Service Framework 및 사용되는 Language 별 Package Lib 및 미리 정의된 아키텍처

#### IAC Env.

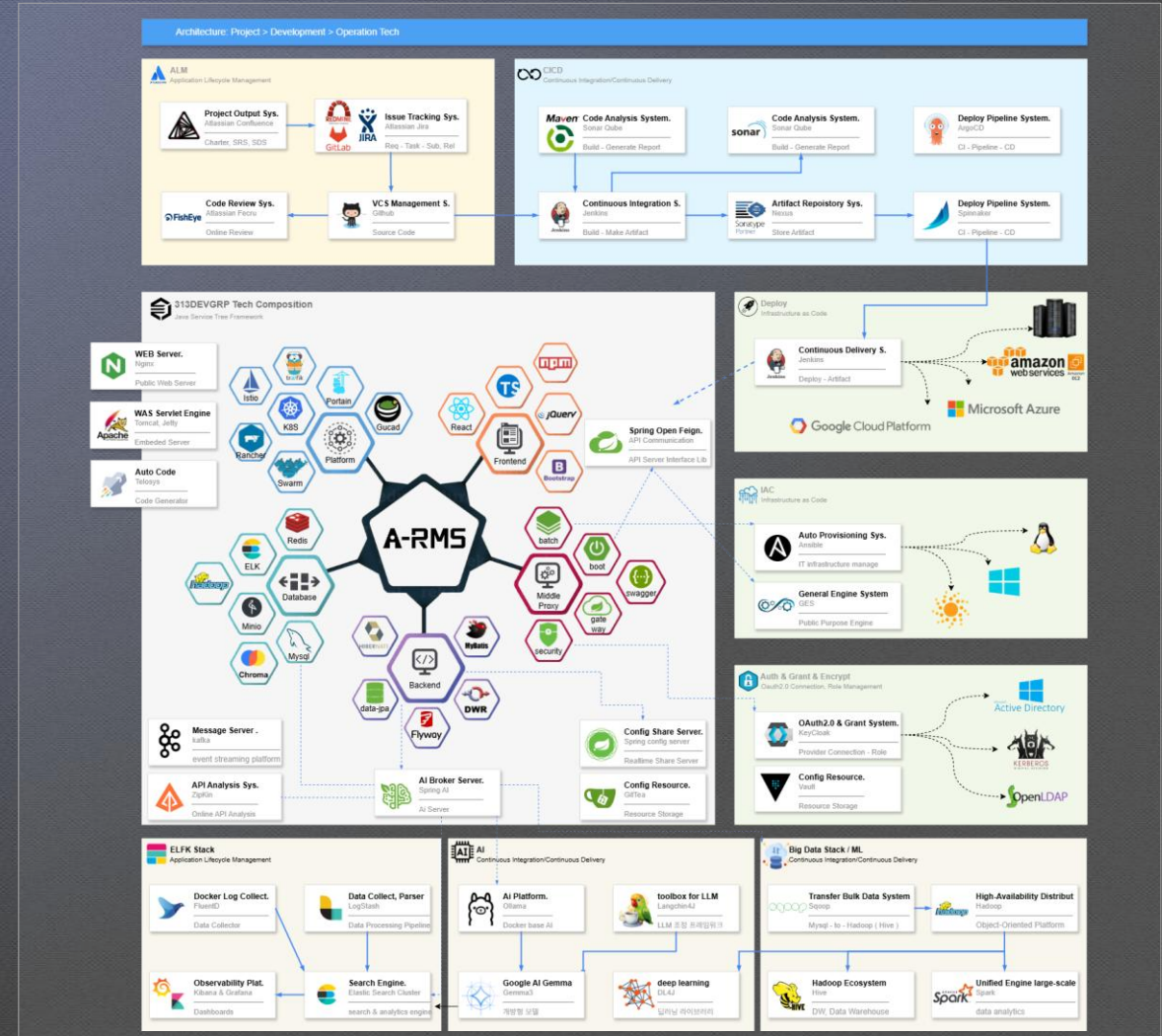
▶ 서비스 운용을 위한 대상을 관리하기 위한 미리 정의된 템플릿 제공

#### Auth N/Z Env.

▶ 인증 및 인가를 위한 레이어를 분리하여 확장성을 제공하고 SSO 및 OIDC 를 미리 제공

#### SearchEngine & BigData Env.

▶ 시계열 데이터 ( 모니터링 ) 및 빅데이터 분석 환경을 미리 정의하여 유틸 및 환경 제공





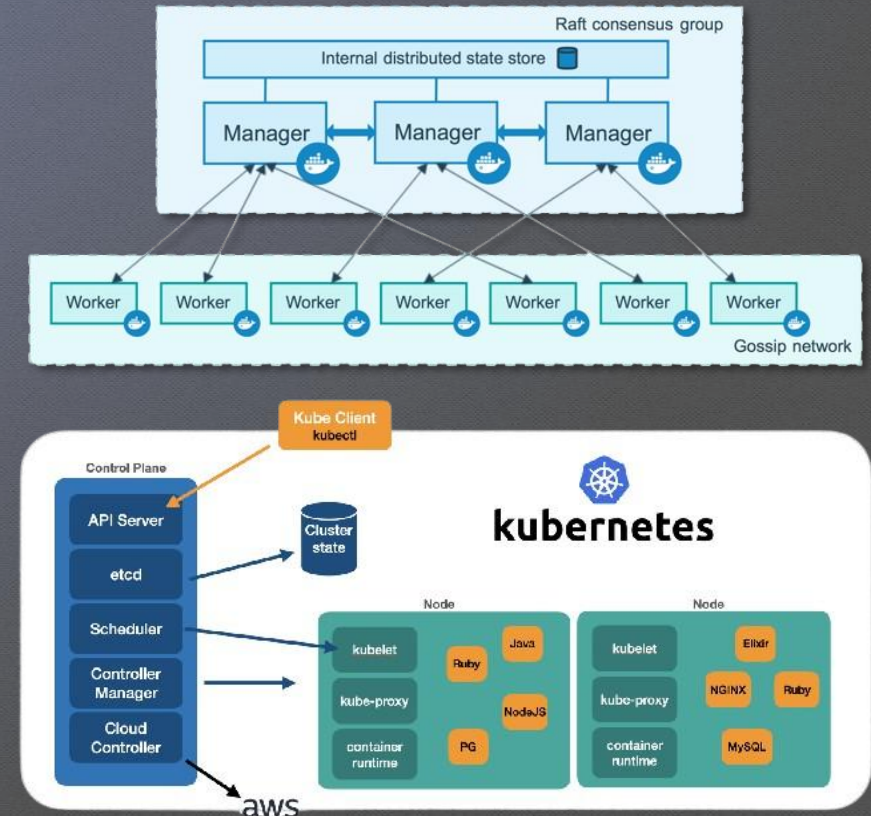
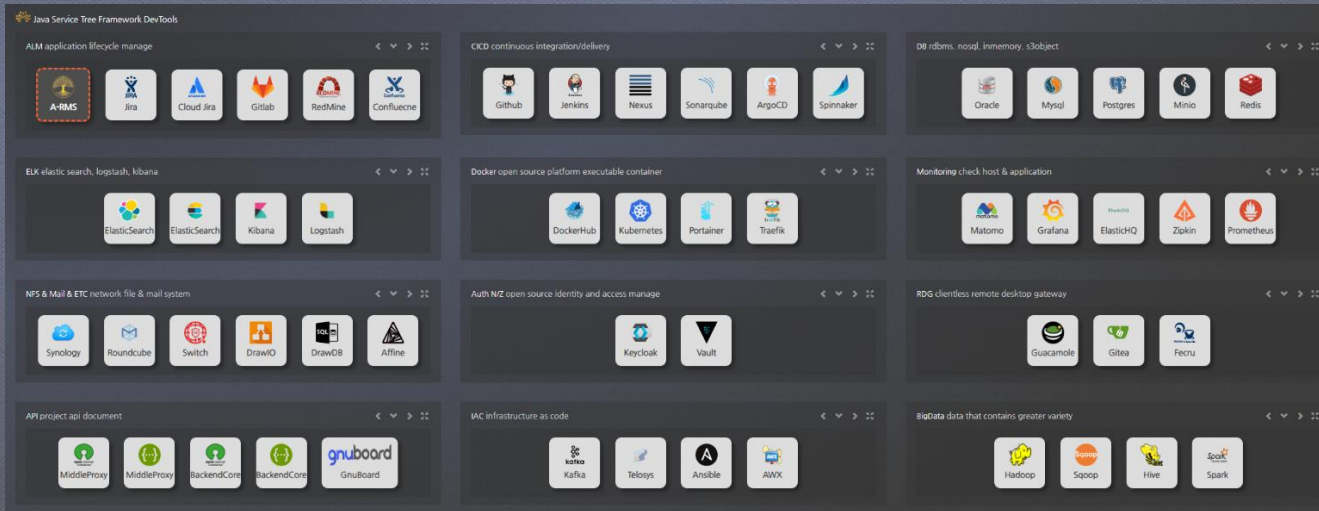
PaaS 및 SaaS : 온프레미스 및 클라우드 동시 대응을 위한 플랫폼 제공

## All-IN-ONE Solution = A-RMS

진정한 멀티 클러스터 환경의 운영 환경 까지 제공합니다

▶ 규모 및 환경에 맞는 구성과 DEVOPS 구성 APP를 동시에 제공

▶ 최적화 및 검증된 설정을 제공





## PLE Effect

### PLE 효과

- ▶ 핵심자산을 사용하여 소프트웨어의 생산성을 높이고 재사용성을 증대
- ▶ 개발비용 감소를 통한 개발 효율성 향상 및 시장 적시성 향상
- ▶ 컴포넌트의 공급이나 조립과 같은 기술적이면 뿐만 아니라 변화에 신속히 적응할 수 있도록 자산을 관리 필요

#### 개발 용이성

- 패턴 기반 개발과 비즈니스 로직에만 집중한 개발 가능
- 공통 기능은 프레임워크가 제공

#### 시스템 복잡도의 감소

- 복잡한 기술은 프레임워크에 의해 숨겨짐
- 미리 잘 정의된 기술 셋 적용

#### 이식성

- 플랫폼 비의존적인 개발 가능
- 플랫폼과의 연동은 프레임워크가 제공

#### 품질보증

- 검증된 개발 기술과 패턴에 따른 개발
- 고급 개발자와 초급 개발자의 차이를 줄여줌

#### 운영 용이성

- 변경이 용이함
- 비즈니스 로직 / 아키텍처 파악이 용이

#### 개발 코드의 최소화

- 반복 개발 제거
- 공통 컴포넌트와 서비스 활용

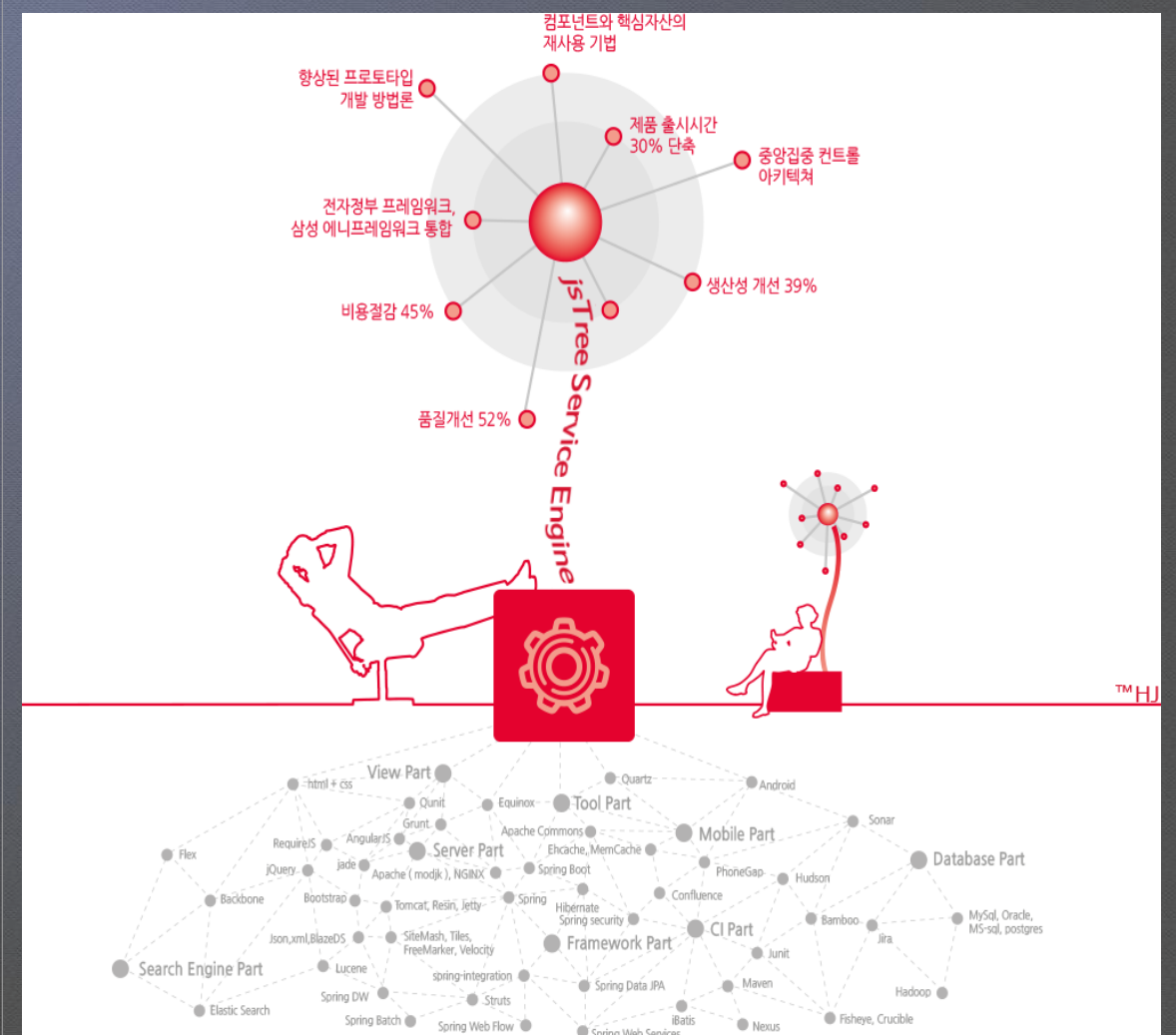
#### 변경 용이성

- 잘 구조화된 아키텍처 적용
- 플랫폼에 비의존적

#### 설계와 코드의 재사용성

- 프레임워크의 서비스 및 패턴의 재사용
- 사전에 개발된 컴포넌트의 재사용

### 프레임 워크 (Framework)





**Ai + ALM + RMS + PMS = A-RMS**

SALES OFFERING



### [ 1. Pain Point ]

영업 :

최근 업무 효율화 때문에,  
모니터링이 강화되지 않았나요?

담당자 :

업무량은 늘어나는데 인력은 줄고,  
현황 파악할 시간이 부족해요.

### [ 2. Value (Time/Scope/Resource/Cost) ]

영업 :

A-RMS는 Jira 연동만으로 업무 현황을 자동 시각화합니다.

#### Time

일정 지연 구간 자동 분석  
→ 마감 위험 사전 인지

#### Scope

요구사항-이슈사항 자동 연결  
→ 업무 흐름 단절 방지

[ 일반 담당자 ] 관점에서



#### Cost

팀 운영 효율 근거로 비용 설득 가능  
→ 상위 보고에 활용

#### Resource

리소스 과다/과소 투입 자동 탐지  
→ 부족한 인력 요청의 근거 제공

### [ 3. CTA (Call To Action) ]

영업 :

Jira 하나만 연동해보세요.  
30분 내 팀 현황과 병목이 보입니다.



### [ 1. Pain Point ]

영업 :

팀 성과, 수치로 객관화 하는데  
한계가 있지 않으세요?

관리자 :

진척률 정도인데, 실제 투입 대비 효과는  
수치화하기 어려워요.

### [ 2. Value (Time/Scope/Resource/Cost) ]

영업 :

A-RMS는 Jira 데이터로 투입 성과를 자동 분석해 수치화합니다.

#### Time

프로젝트 완료 예상 시점 예측  
→ 리스크 조기 대응

#### Scope

진척률, 우선순위, 생산성 시각화  
→ 임원 보고 근거 확보

[ 관리자 ] 관점에서



#### Cost

투입 대비 산출물 정량화  
→ 성과 기반 예산 요청 가능

#### Resource

팀원별 투입성과 비교 분석  
→ 효율성과 기여도 데이터 확보

### [ 3. CTA (Call To Action) ]

영업 :

현재 Jira 데이터 분석해드릴까요?  
실데이터 기반 보고가 달라집니다.



### [ 1. Pain Point ]

영업 :

투자 판단할 정량적 근거가  
애매하진 않으세요?

경영진 :

성과와 회수 시점이 안 보이니  
판단이 불확실합니다.

### [ 2. Value (Time/Scope/Resource/Cost) ]

영업 :

A-RMS는 매출 기여와 회수 시점을 실시간 계산해 투자 결정을 판단합니다.

#### Time

주요 과제 일정 리스크 추적  
→ 전략 일정 통제 가능

#### Scope

과제별 범위-성과 연계 분석  
→ 투자 우선순위 판단

[ 경영진 ] 관점에서

 **A-RMS**

#### Cost

ROI, BEP 시점 수치화  
→ 투자 판단 근거 확보

#### Resource

조직 단위 리소스 효율 분석  
→ 자원 최적화 전략 수립

### [ 3. CTA (Call To Action) ]

영업 :

지금 Jira 프로젝트 하나로  
투자 판단 근거를 확인해 보시죠.



1. 고객은 이런 고민이 있습니다

2. 기존 방식의 한계는 이렇습니다

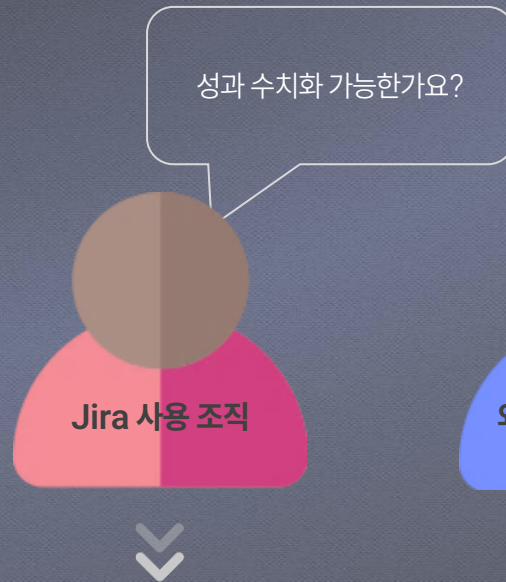
3. A-RMS가 해결합니다  
(Solution Value)

4. 이렇게 제안하세요  
(A-RMS Pitching Guide)

5. 경쟁 솔루션과 이렇게 다릅니다

6. 영업 활동을 이렇게 진행하세요  
(Sales Process)

☑ 주요 타겟 고객의 특성 및 진입 트리거 메시지



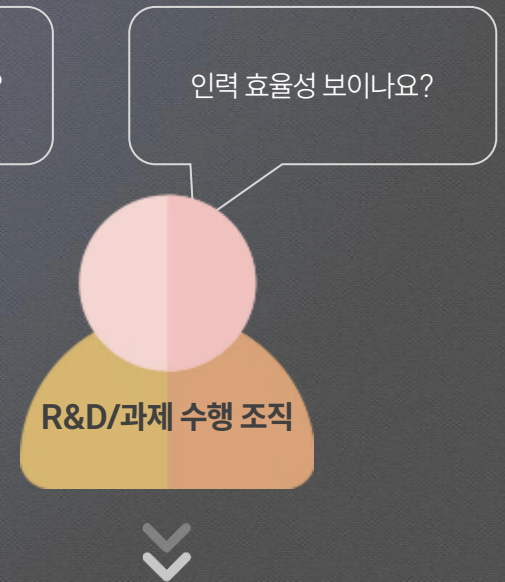
투입 대비 성과 측정 어려움



범위 관리, 변경 추적 어려움



일정 관리 복잡성, 지연 리스크



인력 효율, 자원 최적화 부족



1. 고객은 이런 고민이 있습니다

2. 기존 방식의 한계는 이렇습니다

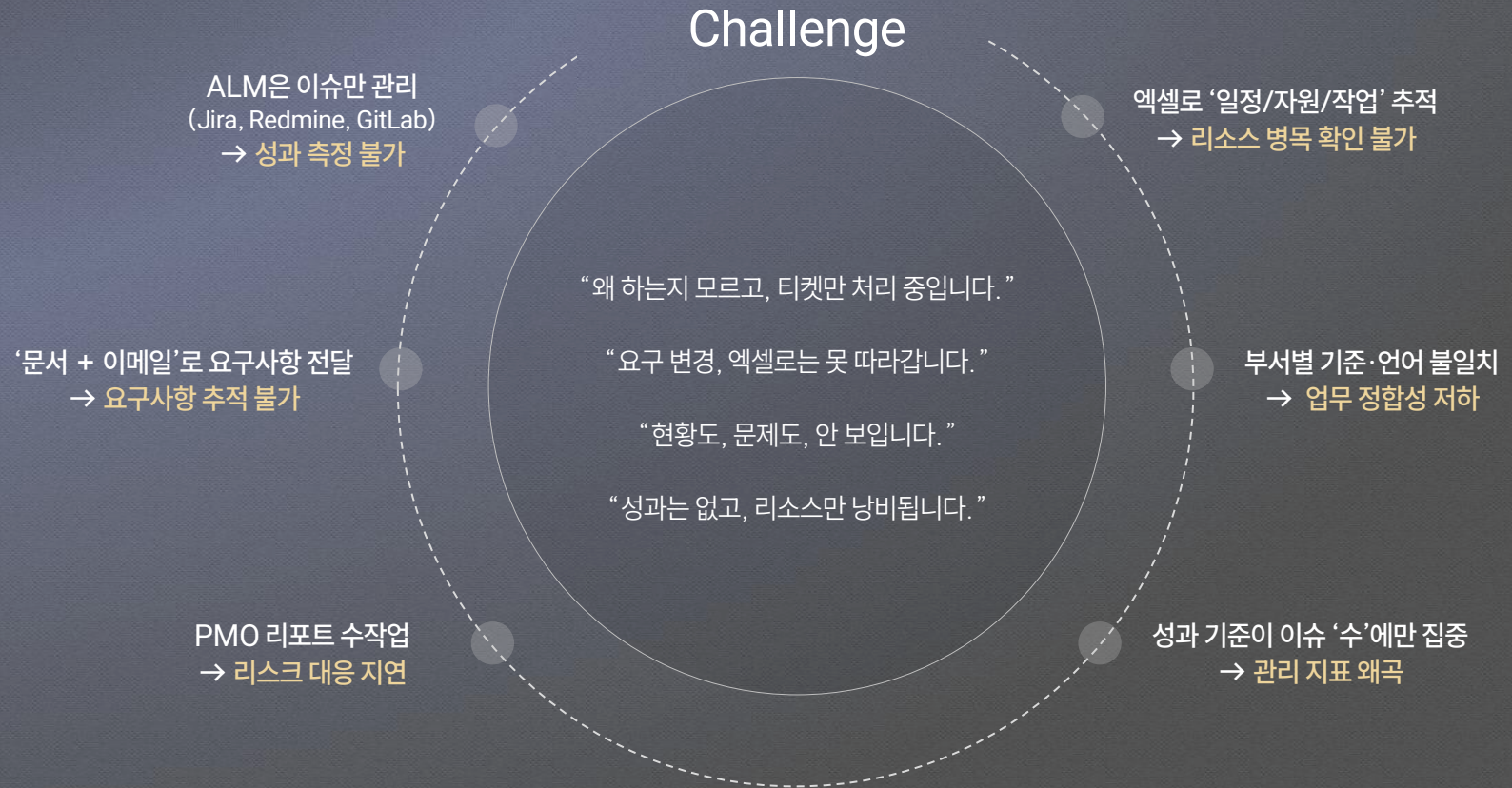
3. A-RMS가 해결합니다  
(Solution Value)

4. 이렇게 제안하세요  
(A-RMS Pitching Guide)

5. 경쟁 솔루션과 이렇게 다릅니다

6. 영업 활동을 이렇게 진행하세요  
(Sales Process)

☑ 기존 현행 방식의 본질적인 문제





1. 고객은 이런 고민이 있습니다

2. 기존 방식의 한계는 이렇습니다

3. A-RMS가 해결합니다  
(Solution Value)

4. 이렇게 제안하세요  
(A-RMS Pitching Guide)

5. 경쟁 솔루션과 이렇게 다릅니다

6. 영업 활동을 이렇게 진행하세요  
(Sales Process)

☑ A-RMS 가 제안하는 비즈니스 가치





1. 고객은 이런 고민이 있습니다
2. 기존 방식의 한계는 이렇습니다
3. A-RMS가 해결합니다  
(Solution Value)
4. 이렇게 제안하세요  
4.1 A-RMS Pitching Guide
5. 경쟁 솔루션과 이렇게 다릅니다
6. 영업 활동을 이렇게 진행하세요  
(Sales Process)

### ☑ A-RMS 제안 단계별 핵심 메시지 (1/2)

#### Hook 질문

“귀사의 **요구사항** 한 줄이 어디까지 했는지,  
**지금 당장 볼 수 있습니까?**”



요구사항 자동 연결  
(요구사항 - 이슈사항)

#### Why 공감

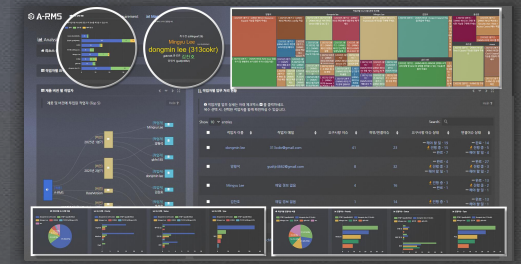
“**요구사항 가시성**이 없으니  
일정·재작업·비용이 계속 터지는 겁니다.”



변경된 요구사항의 영향도

#### Value 제안

문서와 시스템 사이 끊어진 선을 잇는  
단 하나의 규칙,  
“**작업자는 [요구사항]으로 일한다.**”



비즈니스 관점의 분석

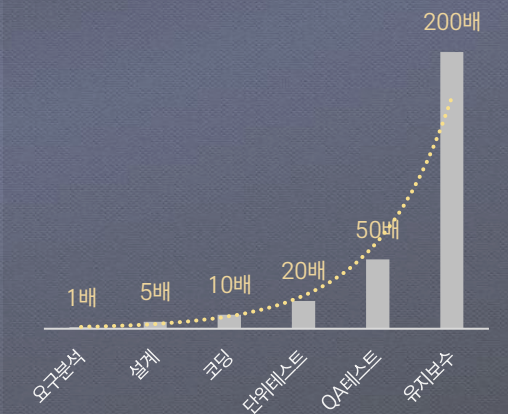


1. 고객은 이런 고민이 있습니다
2. 기존 방식의 한계는 이렇습니다
3. A-RMS가 해결합니다 (Solution Value)
4. 이렇게 제안하세요  
4.2 A-RMS Pitching Guide
5. 경쟁 솔루션과 이렇게 다릅니다
6. 영업 활동을 이렇게 진행하세요 (Sales Process)

### ☑ A-RMS 제안 단계별 핵심 메시지 (2/2)

#### ROI 근거

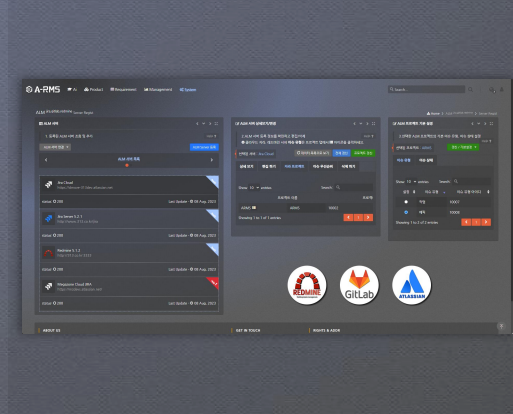
“초기에 놓친 요구사항 오류는  
유지보수 단계에서 200배의 비용이 듭니다.”



단계별 결함 제거 비용 10~13 배씩 증가  
(※ 출처 : Karl Wiegers)  
→ A-RMS 도입 시 초기 오류 조기 검출

#### Demo PoC

“지금 ALM 프로젝트 1개만 연결해 보시면,  
30분 안에 현황 및 성과를 보실 수 있습니다.”



ALM 데이터 성과지표로 변환  
(Time/Scope/Resource/Cost)

#### CTA

“1개월 Pilot 성과부터 보시겠습니까,  
아니면 지금 바로 계약 하시겠습니까?”



Ai 기반 분석 및 예측



- 1. 고객은 이런 고민이 있습니다
- 2. 기존 방식의 한계는 이렇습니다
- 3. A-RMS가 해결합니다 (Solution Value)
- 4. 이렇게 제안하세요 (A-RMS Pitching Guide)
- 5. 경쟁 솔루션과 이렇게 다릅니다
- 6. 영업 활동을 이렇게 진행하세요 (Sales Process)

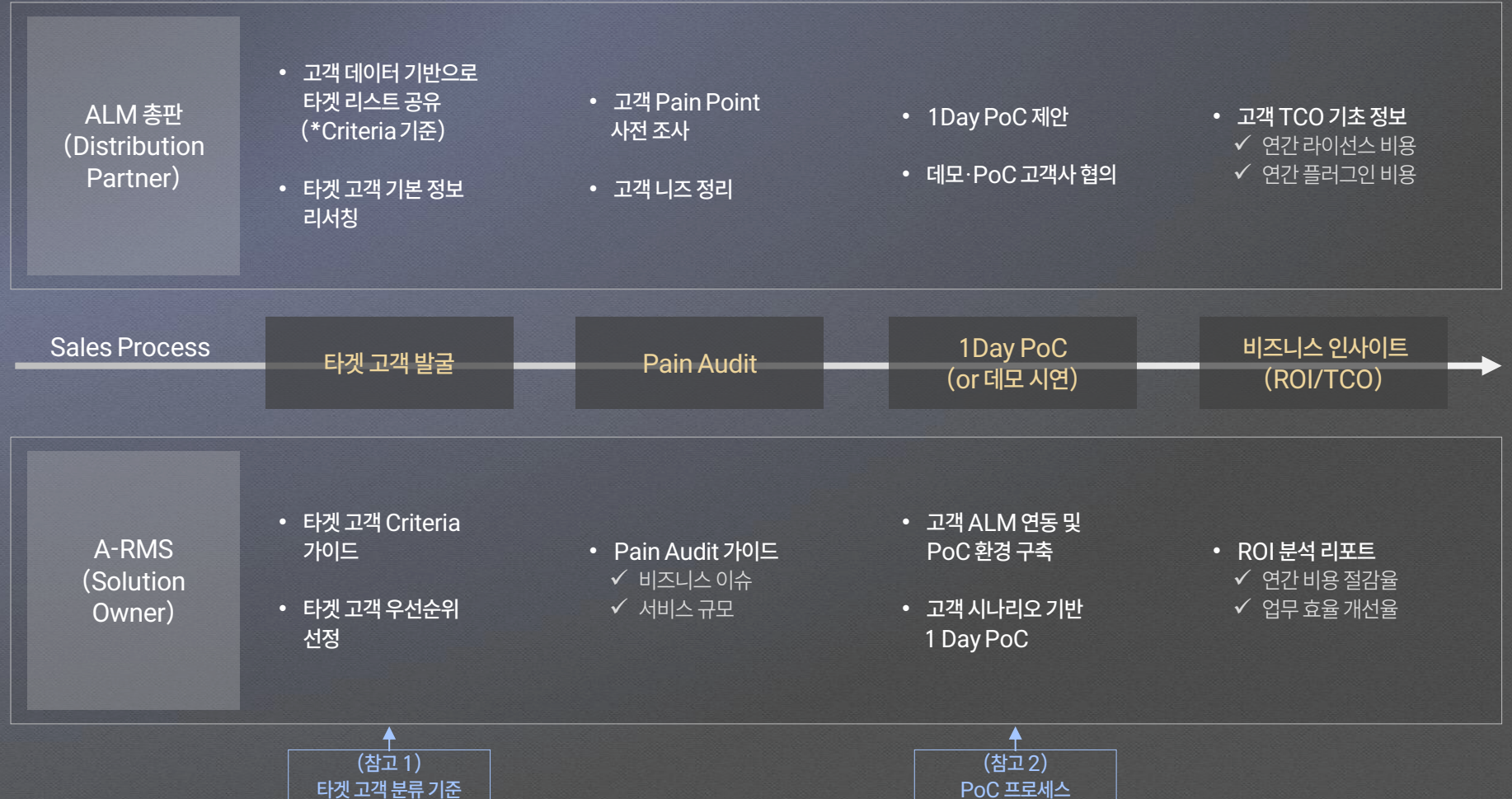
☑ A-RMS vs 경쟁 솔루션

항목	A-RMS	전통 ALM ( Jira 등 )	RM 툴 ( Jama 등 )	PPM 툴 ( Clarity 등 )
ALM 교체 여부	기존 ALM 유지 ( 연동 )	플러그-인 필요	별도 플랫폼	별도 플랫폼
요구사항 트레이스	제품·버전·이슈 Full Trace	제한적 ( Epic/Story 수준 )	가능	제한적
Time/Scope/Resource/Cost 분석	자동 분석 ( Dashboard )	수동적	제한적	제한적
Ai 예측 & Dev Performance	Ai 기반 진척 예측, 패턴 분석	없음	일부	없음
배포 형태	온프레미스	SaaS	SaaS	SaaS
총 소유 비용	★★★★☆ ( 기존 ALM 사용 )	★★★★★ ( 플러그-인 별도 비용 )	★★★★★	★★★★☆



1. 고객은 이런 고민이 있습니다
2. 기존 방식의 한계는 이렇습니다
3. A-RMS가 해결합니다  
(Solution Value)
4. 이렇게 제안하세요  
(A-RMS Pitching Guide)
5. 경쟁 솔루션과 이렇게 다릅니다
6. 영업 활동을 이렇게 진행하세요  
6.1 Sales Process

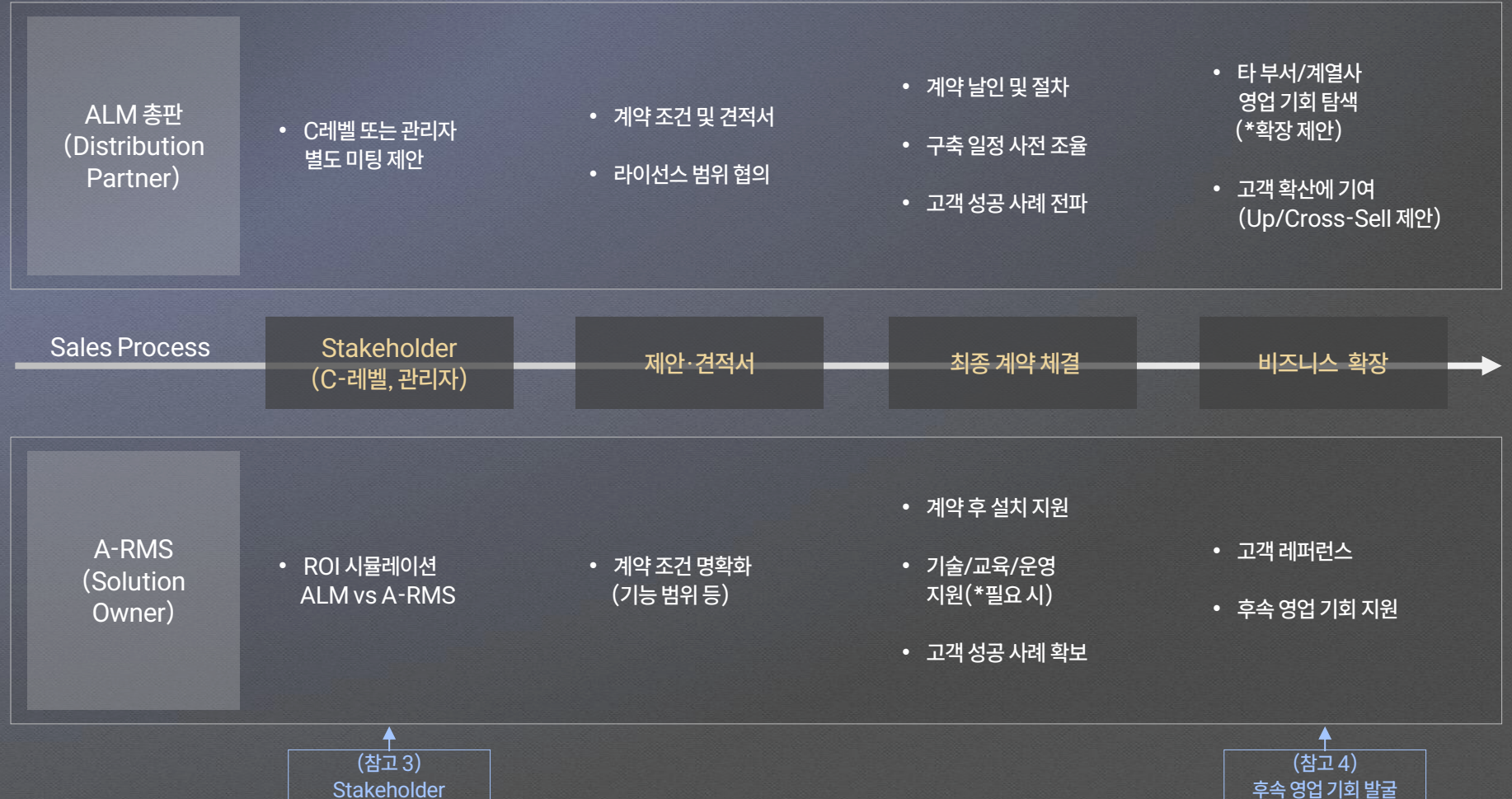
### ☑ ALM 총판 (Distribution Partner)와 A-RMS (Solution Owner) 영업 활동 프로세스 (1/2)





1. 고객은 이런 고민이 있습니다
2. 기존 방식의 한계는 이렇습니다
3. A-RMS가 해결합니다  
(Solution Value)
4. 이렇게 제안하세요  
(A-RMS Pitching Guide)
5. 경쟁 솔루션과 이렇게 다릅니다
6. 영업 활동을 이렇게 진행하세요  
6.2 Sales Process

### ☑ ALM 총판 (Distribution Partner)와 A-RMS (Solution Owner) 영업 활동 프로세스 (2/2)





1. 고객은 이런 고민이 있습니다
2. 기존 방식의 한계는 이렇습니다
3. A-RMS가 해결합니다  
(Solution Value)
4. 이렇게 제안하세요  
(A-RMS Pitching Guide)
5. 경쟁 솔루션과 이렇게 다릅니다

6. 영업 활동을 이렇게 진행하세요  
6.3 Sales Process

### ☑ (참고 1) 타겟 고객 분류 기준

분류 기준	세부사항
의사결정권자 정보	<ul style="list-style-type: none"> <li>• 관리자(팀장)</li> <li>• 임원급 이상</li> </ul>
회사 규모 (전체 인원)	<ul style="list-style-type: none"> <li>• 50명 이하</li> <li>• 50명~100명 미만</li> <li>• 100명~300명 미만</li> <li>• 300명~500명 미만</li> <li>• 500명 이상</li> </ul>
Jira 월간 사용자	<ul style="list-style-type: none"> <li>• 50명 이하</li> <li>• 50명~100명 미만</li> <li>• 100명~300명 미만</li> <li>• 300명~500명 미만</li> <li>• 500명 이상</li> </ul>
플러그인 추가 사용량	<ul style="list-style-type: none"> <li>• 월간 추가 비용</li> <li>• 연간 사용량 (* 플러그인 항목별)</li> </ul>



1. 고객은 이런 고민이 있습니다
2. 기존 방식의 한계는 이렇습니다
3. A-RMS가 해결합니다  
(Solution Value)
4. 이렇게 제안하세요  
(A-RMS Pitching Guide)
5. 경쟁 솔루션과 이렇게 다릅니다
6. 영업 활동을 이렇게 진행하세요  
6.4 Sales Process

### ☑ (참고 2) PoC 프로세스

단계	세부 설명
사전 인터뷰	<ul style="list-style-type: none"> <li>비즈니스 이슈 : 월간 사용자, 플러그인 비용, 주요 워크로드, 서비스 규모</li> </ul>
PoC 셋업	<ul style="list-style-type: none"> <li>ALM 연동 : 요구사항(A-RMS) – 이슈사항(ALM) 시스템 연결</li> </ul>
1Day 데이터 수집·검증	<ul style="list-style-type: none"> <li>통계 데이터 : Time, Scope, Resource, Cost (*고객 데이터 기반)</li> </ul>
분석 결과 리포트	<ul style="list-style-type: none"> <li>비즈니스 인사이트 : A-RMS 적용 시 예상 효과 및 ROI</li> </ul>



1. 고객은 이런 고민이 있습니다
2. 기존 방식의 한계는 이렇습니다
3. A-RMS가 해결합니다  
(Solution Value)
4. 이렇게 제안하세요  
(A-RMS Pitching Guide)
5. 경쟁 솔루션과 이렇게 다릅니다
6. 영업 활동을 이렇게 진행하세요  
6.5 Sales Process

### ☑ (참고 3) Stakeholder

Stakeholder	활용 자료	핵심 질문
C-Level (임원급 이상)	<ul style="list-style-type: none"> <li>투자 성과 분석</li> <li>ROI 시뮬레이션</li> </ul>	“투자 회수 기간은?”
관리자 (팀장급)	<ul style="list-style-type: none"> <li>프로젝트 범위·일정·리소스 통합 대시보드</li> <li>요구사항 흐름 추적</li> <li>리포트 및 통계 자동화</li> </ul>	“리소스 낭비 잡히나요?”
보안팀	<ul style="list-style-type: none"> <li>데이터 저장 위치/흐름도</li> <li>접근권한 관리</li> <li>설치형(고객사 서버)</li> </ul>	“데이터가 외부로 나가나요?”



1. 고객은 이런 고민이 있습니다
2. 기존 방식의 한계는 이렇습니다
3. A-RMS가 해결합니다  
(Solution Value)
4. 이렇게 제안하세요  
(A-RMS Pitching Guide)
5. 경쟁 솔루션과 이렇게 다릅니다
6. 영업 활동을 이렇게 진행하세요  
6.6 Sales Process

### ☑ (참고 4) 후속 영업 기회 발굴 - 타 부서/타 계열사 및 관계사

확장 대상	타겟 후보	추가 제안 포인트	실행 전략
타 부서 (조직 내 확장)	Jira 사용 여부, 협업 이슈	기존 적용부서와의 협업 문제 해결, 통합 가시성 확보	[계약 종료 고객 리스트] ↓ [고객 성공 사례 공유] ↓ [타 부서/계열사/관계사 리스트업] ↓ [사전조사: ALM 사용 여부] ↓ [데모 & 니즈 청취 미팅 제안] ↓ [적합 시 1Day PoC → 확장 계약]
타 계열사 (그룹사 확장)	그룹웨어/Jira 연동 환경	동일 ALM 또는 Jira 사용 시, 동일 환경 기반 도입 용이	
관계사·협력사 (외부 확장)	고객사와 협업 중인 시스템 (ex, 외주업체, SI 협력사)	요구사항 공유 및 외부 협업, 이슈 관리 필요성	



### 제품 이해

Q. Jira를 교체해야 하나요?

기존 사용 중인 ALM (Jira, Redmine, GitLab 등)를 그대로 연동해 사용합니다.

Q. 기존 데이터는 어떻게 마이그레이션하나요?

ALM 연동 즉시 데이터를 자동 수집하므로, 별도 작업 없이 사용 가능합니다.

### 도입/운영

Q. 도입 기간은 얼마나 걸리나요?

Standalone Pilot은 1일, 본격 확산 도입은 2-4주 내 완료됩니다.

Q. 유지보수나 기술 지원도 포함되나요?

도입 플랜에 따라 제공되는 지원 범위가 다르며, 자세한 사항은 별도 문의 부탁드립니다.

### 기술/보안

Q. 데이터 보안은 어떻게 보호되나요?

Vault 암호화, Keycloak SSO 등 다양한 보안 기능을 지원합니다.

Q. 리포트 자동화 기능이 있나요?

요구사항 기반으로 진척률, 리스크, 예측 지표를 자동 리포트(PDF, Excel, CSV)로 생성합니다.

### 라이선스 및 비용

Q. 라이선스 정책은 어떻게 되나요?

300명 이하 조직은 표준 요금제로 제공되며, 그 이상 규모는 조직 현황 및 사용 범위에 따라 맞춤 견적을 제공합니다.

Q. 기능별로 추가 과금이 있나요?

모든 라이선스 플랜에서 동일한 기능을 제공합니다. 단, 추가 모듈 도입 시 별도 비용이 발생할 수 있습니다.

### Ai/커스터마이징

Q. Ai 기능은 바로 사용할 수 있나요?

Time/Scope/Resource/Cost 예측 기능은 기본 제공되며, 개인별/팀별 성과 분석은 2025년 하반기에 출시 예정입니다.

Q. 워크플로우 커스터마이징도 가능하나요?

워크플로우 확장, 사내 표준 리포트 양식 등 커스터마이징이 필요한 경우 별도 문의 부탁드립니다.

### ROI 및 효과

Q. ROI는 어떻게 측정하나요?

요구사항 연계 지표를 통해 투자 대비 성과를 측정합니다. (재작업 감소율, 일정 단축률, 리소스 효율 향상 등)

Q. 협업 부서(기획, QA, 외주 등)와 정합성도 향상되나요?

모든 작업이 요구사항 단위로 연결되므로, 협업 부서와 외주 업체를 포함해 정량적 기준으로 통합 관리가 가능합니다.

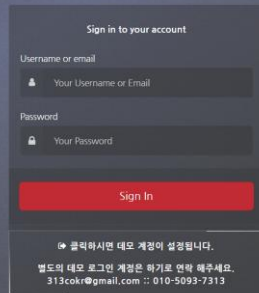




자세한 영업 지원 도구는 [313DEVGRP Sales Portal \(링크 연결\)](#) 에서 확인 부탁드립니다.

### Live Demo Site

웹 접속 후 라이브 데모



<https://www.a-rms.net>  
ID: admin / PW: qwe123

### Pitch Deck (PPT)

솔루션 프로파일·  
가치 제안 슬라이드



313DEVGRP Sales Portal  
↓  
ARMS\_313DEVGRP  
\_SolutionProfile  
파일 다운로드

### ROI Calculator (XLS)

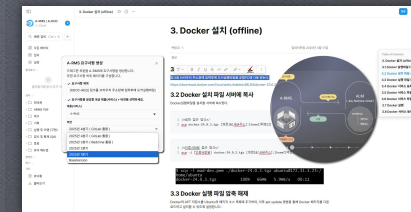
비용 절감 템플릿



313@313.co.kr  
메일 요청  
(2주 전까지)

### PoC Guide

1Day Pilot 설치  
스크립트·체크리스트



313@313.co.kr  
메일 요청  
(2주 전까지)

### Demo kit

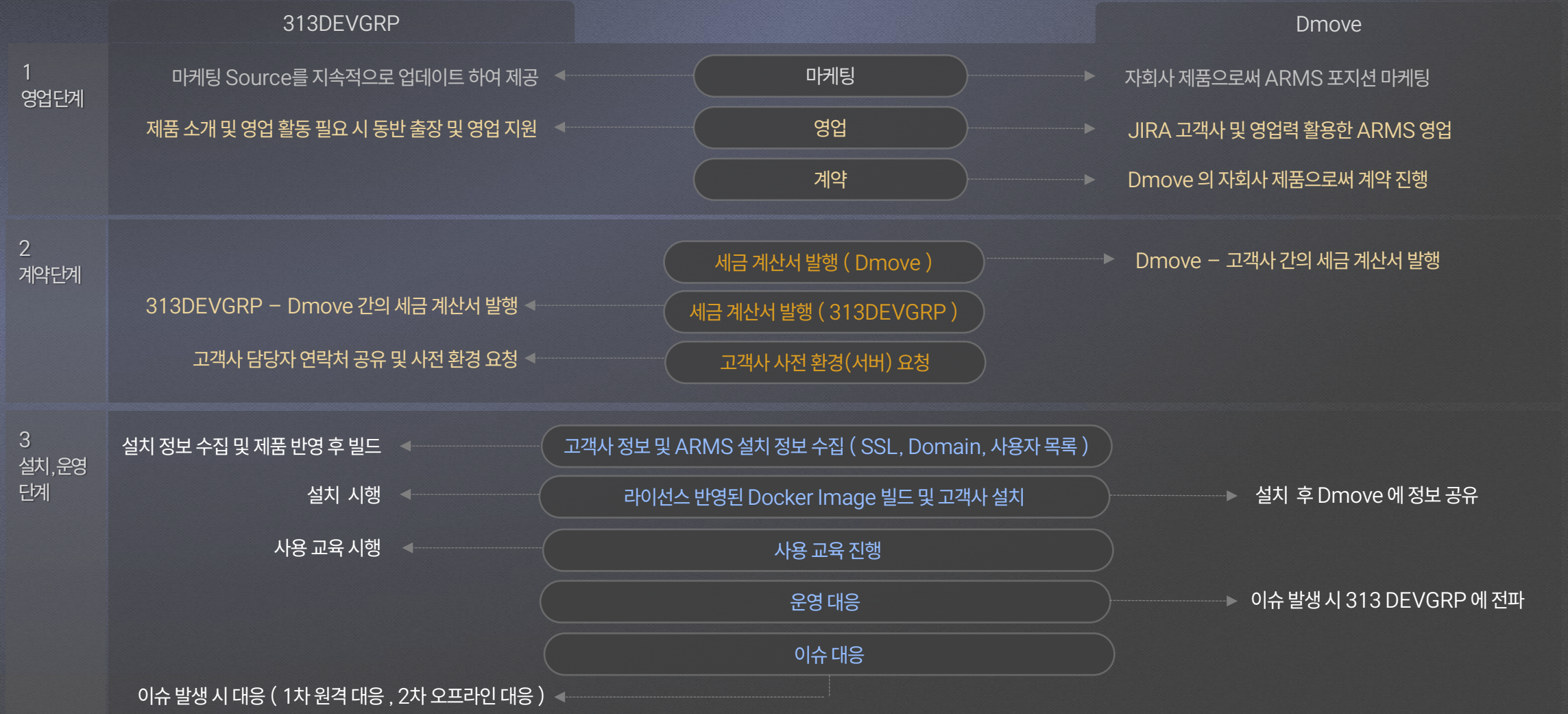
Demo 영상  
(Time, Scope, Resource, Cost)



Demo Data Set



313DEVGRP 와 Dmove 의 영업 부터 계약까지 정리

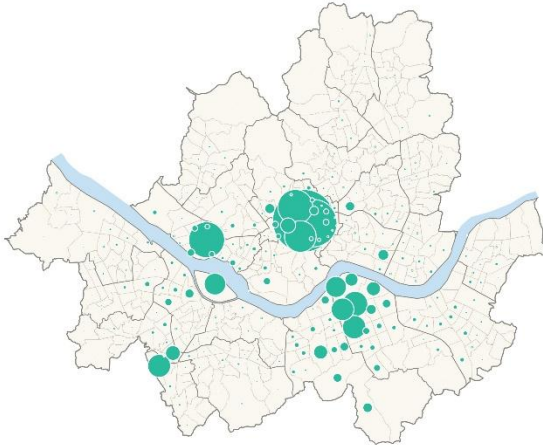
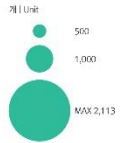




# ALM 시장 분석

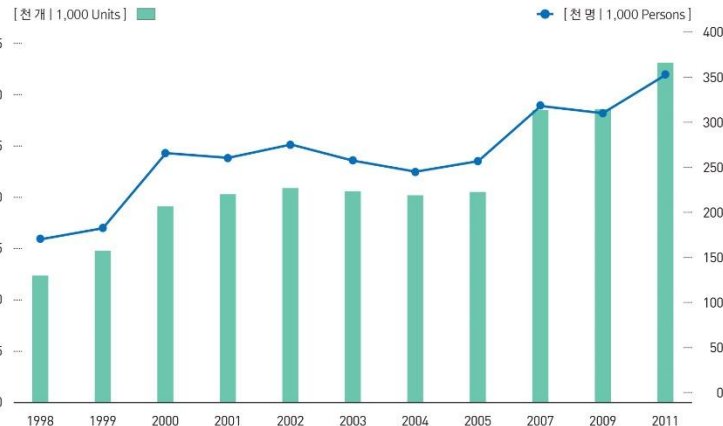
## ARMS - 성장 가능성 분석

그림 7-20. 동별 IT산업 사업체 수 2011  
Figure 7-20. Establishments in the IT Industries by Neighborhood(dong), 2011



그래프 7-12. IT산업 사업체 및 종사자 수 1998-2011  
Graph 7-12. Trends in the IT Industries, 1998-2011

■ 사업체 Establishments  
◆ 종사자 Workers

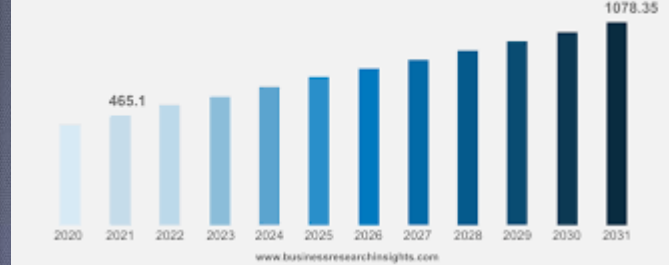


### And India Application Lifecycle Management (Alm) Software Market

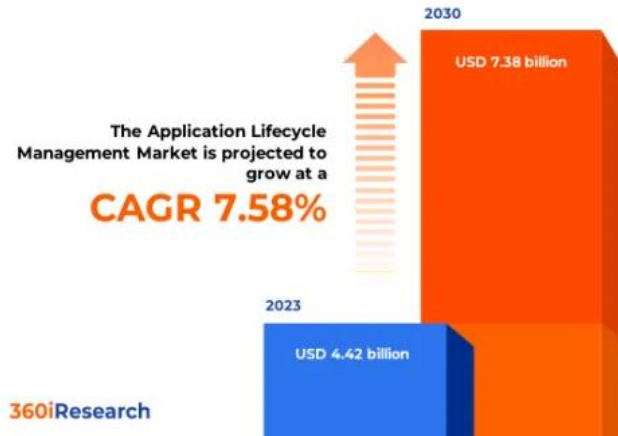
**Segment by Application** - Automotive and Transportation, Aerospace and Defense, BFSI, Energy and Utilities, Retail, Healthcare, IT and Telecom, Others  
**Segment by Type** - Single Function, Multiple Functions  
**By Company** - HP, Atlassian, Techexcel, IBM, Microsoft, Rocket Software, Enalean



### Global Application Lifecycle Management (ALM) Software Market Size, 2031 (USD Million)



### Application Lifecycle Management Market



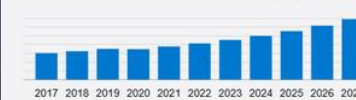
### GLOBAL APPLICATION LIFECYCLE MANAGEMENT MARKET 2021-2025



17000+ Reports covering niche topics. Read them at [technavio](https://www.technavio.com)

### Software Testing Services Market in ANZ 2023-2027

#### Market Size Outlook (USD Billion)



#### CUSTOMER LANDSCAPE

Drivers of price sensitivity  
Adoption lifecycle  
Importance in the customer purchase basket  
Adoption rates  
Key purchase criteria

**Key purchase criteria**

- Innovation
- Price
- Quality
- Relatability
- Regulatory Compliance
- Service

#### Drivers of price sensitivity

**Driver** **Impact**

Purchases are undifferentiated ☐

Purchase is a key cost to the buyer ☐

Quality is not important ☐

**Price Sensitivity** ☒

#### Adoption lifecycle

Innovators Early Adopters Early Majority Late Majority Laggards

17000+ Reports covering niche topics. Read them at [technavio](https://www.technavio.com)

2024년  
대한민국  
정보 통신업  
사업체 수  
127,974개

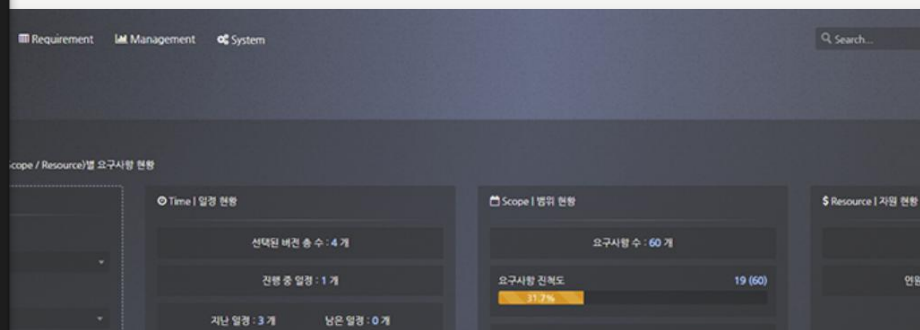


# smartPM

본질에 집중하세요. **A-RMS** 가 서포트하겠습니다.

A-RMS is ALM ( Jira, Redmine, Gitlab... ) integrated  
Requirement Base Project Management System

Get Start



## Build Your Business Properly



[www.a-rms.net](http://www.a-rms.net)

### | OFFICE HOURS

Monday – Friday  
09:00 – 20:00

### | OUR ADDRESS

313 B/D Dosan-daero, Gangnam-gu,  
Seoul, Republic of Korea 06021

### | GET IN TOUCH

(+82) 10 5093 7313

### | FOLLOW US

313@313.co.kr

회사에 깊게 뿌리내린 문제를 들여다 볼 수 있는 A-RMS.  
313DEVGRP 가 함께하겠습니다.





# THANK YOU

BUSINESS PRESENTATION

DEMO	: <a href="https://www.a-rms.net">https://www.a-rms.net</a>
ID	: admin
PW	: qwe123