

어플리케이션 영향도 분석 솔루션 『DATAWARE™ AP#』 소개서

EN-CORE Inc. is "Total Data Service Provider." We, as one of the leading data scientists groups in Korea, are offering a wide-range of services such as consulting, education, solutions, and technology information based on data ranging from planning the corporation IT systems to analyzing, designing, developing, transitioning, managing, performing governance, and data servicing.

AP
dataware
impact
analysis

en-core co.,ltd.

06648 | 42, Seocho-daero 46-gil, Seocho-gu, Seoul, Republic of Korea

office +82 2 754 7301 | fax +82 2 754 7305 | web www.en-core.com





- 01. 회사 소개
- 02. 제안 배경
- 03. 솔루션 개요
- 04. 솔루션 구성
- 05. 솔루션 특징점
- 06. 솔루션 주요 기능
- 07. 도입 후 기대효과
- 08. 구축사례

01

회사 소개

Provide All-in-One Data Business Platform

엔코아는 데이터 컨설팅부터 솔루션 기반 시스템 구축, 데이터 분석 및 가공, 전문 교육 제공에 이르기까지 차별화된 데이터 토탈서비스를 제공하는 국내 유일한 “Total Data Service Provider”로서 500여 개사 약 800여개의 프로젝트를 수행한 데이터전문컨설팅 기업

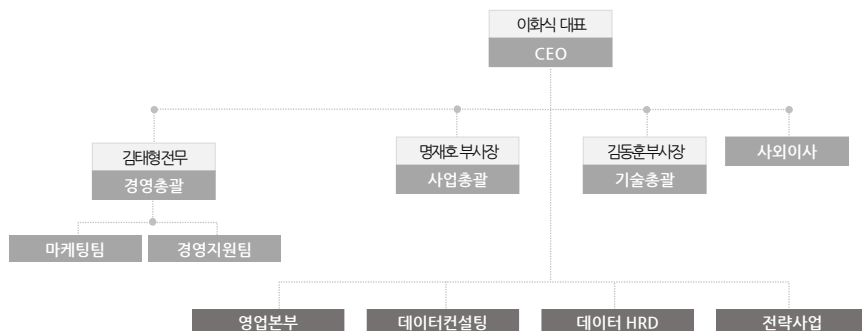
일반현황

일반현황	(주)엔코아	대표자	이화식
사업분야	데이터컨설팅/데이터솔루션/서비스/교육/도서출판		
주소	서울시서초구서초대로46길42 엔코아타워		
대표전화	02.754.7301	사원수	150명
회사설립연도	1997년 11월	그룹사	메타넷
소속협회	한국데이터산업진흥원/한국상용SW협회		

주요연혁

- 2019 국내 최대 독립 IT 서비스기업 - 메타넷 그룹 계열 편입
- 2018 아마존 AWS 레지스터드 파트너십 체결
- 2017 데이터 과학자 HRD센터 설립
- 2017 이화식 대표이사 ‘정보통신의날’ 대통령 표창 수상

조직도



주요사업

01

데이터
전문
컨설팅
서비스

02

데이터
End-to-
End
솔루션

03

빅데이터
분석, 활용,
가공

04

데이터
전문 인력
양성

05

DATAWARE™
EN-CORE Data
framework

데이터 솔루션 DATAWARE™

Provide All-in-One Data Business Platform

(주)엔코아는 기업 데이터 관리를 위하여 하나의 플랫폼으로 데이터 품질 극대화를 위한 데이터 통합 관리 솔루션 DATAWARE™를 제공합니다.

DATAWARE™는 기업 데이터의 분석, 설계, 운용, 관리에 이르기까지 기업 전반의 데이터 관리 체계를 지원하는 통합 솔루션으로 데이터 모델링, 메타, 품질, 흐름, 영향분석, 데이터이행, 운영 관리 등으로 구성되며 국제 웹 표준을 준수하고, 전자정부 프레임워크 호환을 기본으로 구성되어 있습니다.

On-Premises 환경이나 Cloud 환경에 최적으로 서비스 모듈이 구성이 되어 OS 및 브라우저에 종속되지 않고 서비스가 제공됩니다. 모든 솔루션은 플러그인 형태로 별도의 장착이 가능하며 하나의 통합 웹 기반의 관리 포털 형태로 제공되어 운영관리의 최적의 환경을 제공합니다.

데이터 End-to-End 솔루션

데이터 모델링 도구

데이터 이행 솔루션

클라우드 보안/마이그레이션

메타 데이터 관리 솔루션

데이터 포털 솔루션

오픈소스 DB 이관 솔루션

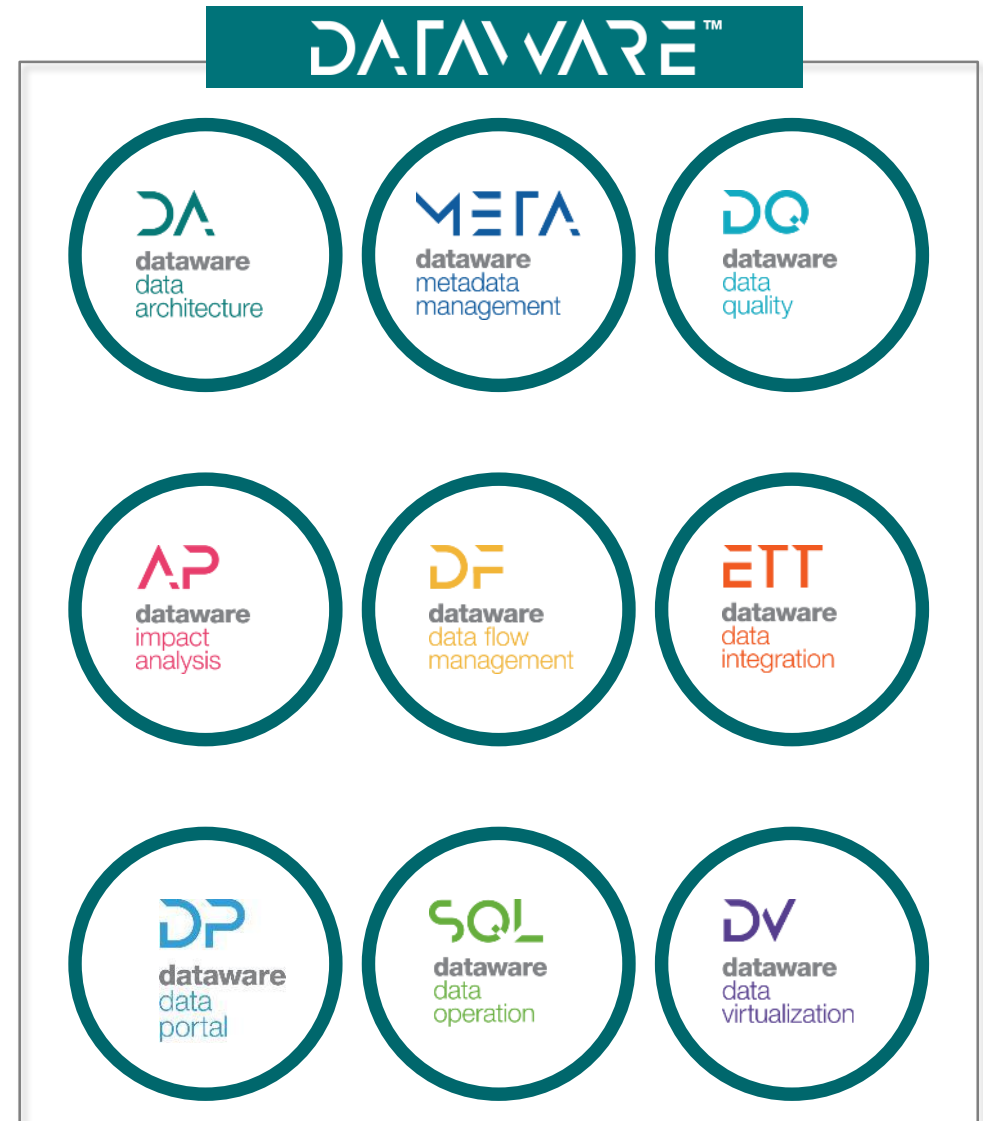
데이터 품질관리 솔루션

데이터 흐름 관리 솔루션

빅데이터 플랫폼

AP 영향도 분석 솔루션

데이터 운영관리 솔루션



02 제안 배경

현황 및 문제점



- 다양한 개발언어와 개발방법으로 인한 시스템 및 업무의 복잡성 증가
- 숙련된 개발자의 경험에 지나치게 의존하고, 비효율적인 Eye-Check 코드분석을 통한 어플리케이션의 관리로 유지보수 비용과 위험적 장애요소의 증가
- 잦은 기술인력의 변동 및 기술지식에 대한 미완적인 습득으로 개발자별 편차 발생
- 변경 프로그램으로 산출물 현행화 어려움
- 개발 및 테스트 시간 부족
- 프로그램 난이도 및 미사용 소스에 대한 식별 어려움

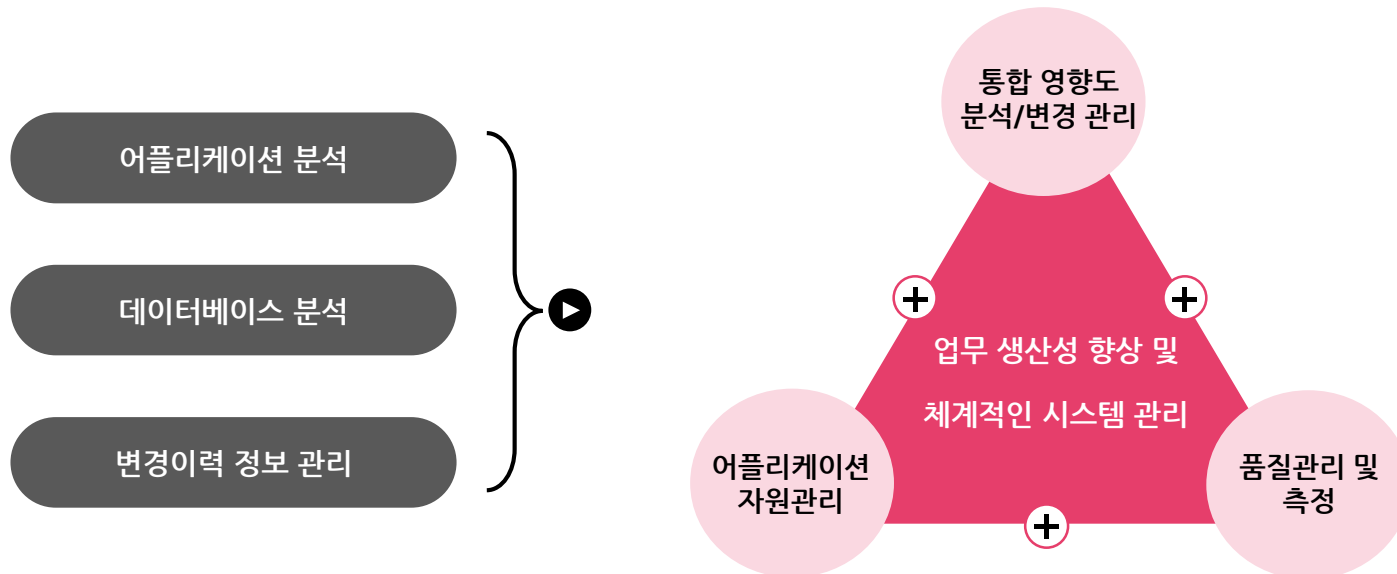


효과적인 개발 프로세스의 기반이 되는 시스템 필요

- ✓ 통합 영향도 분석 및 변경관리를 통한 장애 예방, 품질 향상, 프로젝트 관리를 위한 S/W를 도입하여 효과적인 개발 프로세스가 기반이 되는 시스템

어플리케이션 영향도 분석 솔루션 구축 필요성

시스템 변경에 대한 효과적인 영향도 분석 프로세스를 수립하여 **영향 범위 사전 인지 및 장애 발생 요인을 최소화** 하고, **IT운영 안정성과 운영 효율성 극대화**



03

솔루션 개요

“최고의 분석 엔진 성능과 안정성, 메타정보 활용 및 확장이 용이한 제품으로 관련 업계 1위“

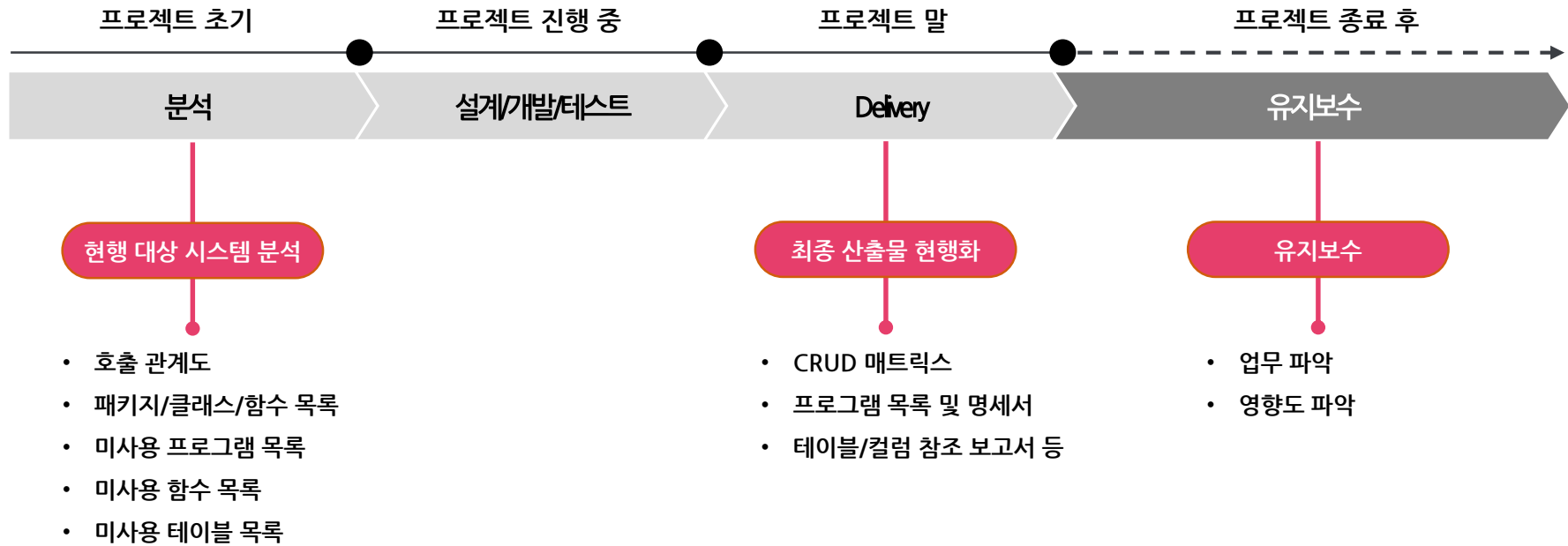


데이터 거버넌스, 어플리케이션 영향도 분석 시스템

- 데이터베이스와 어플리케이션에 대한 자동 분석을 통해 시각적인 구조/연관 분석 정보를 제공

시스템의 데이터베이스 카탈로그 정보와 다양한 언어의 어플리케이션 소스를 단일/통합 Repository에 자동으로 수집하고 파싱하여 이들 간의 정확한 구조 및 연관 정보를 제공하는 도구입니다. 이는 소스 및 데이터베이스 변경에 따른 위험 예방, AS-IS 분석 비용 절감, 업무 효율성 향상 등 어플리케이션의 변경 및 품질 관리와 효율적인 유지보수 체계를 지원하는 어플리케이션 거버넌스 솔루션입니다.

제품 활용 분야

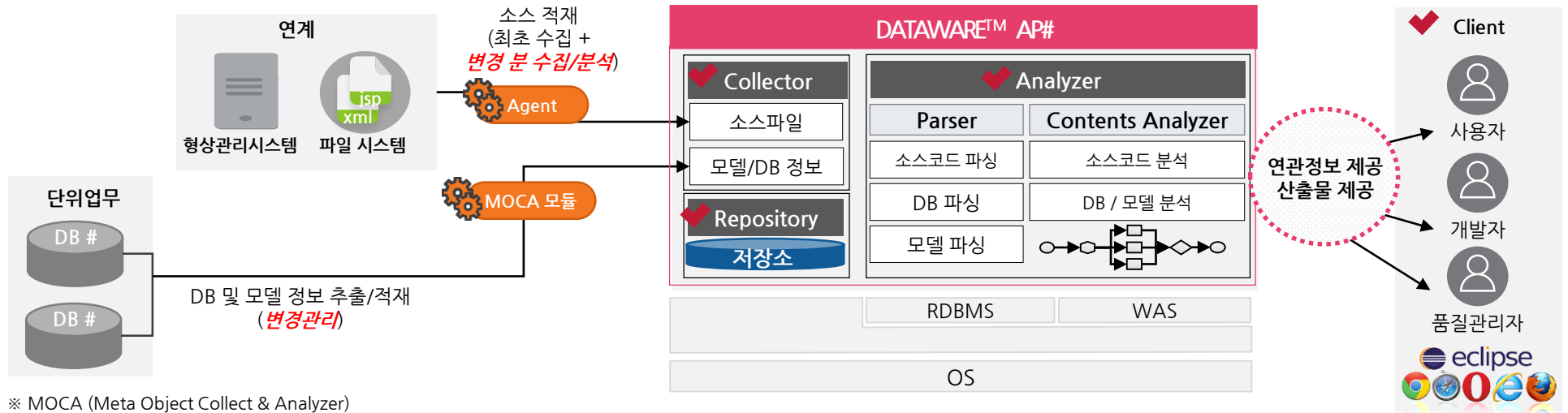


04

솔루션 구성

시스템 구성도 및 사양

DATAWARE™ AP#은 Collector, Analyzer, Repository, Client 로 구성되어 있으며 사용자는 WEB UI를 통해 보다 높은 접근성과 활용성을 제공



제품 구성	설명
Collector	- 모델, 데이터베이스 정보, 소스 파일 수집 및 추출 모듈 - 메타 데이터 수집 배치 작업을 위한 스케줄러 엔진
Analyzer	수집된 소스와 DB 정보, 모델 정보를 파싱한 분석 정보 및 연관 정보 분석
Repository	메타 분석 정보 및 관리 대상 데이터를 저장하는 저장소 역할
Client	분석에 따른 결과 조회 및 리포트 자동 생성 기능을 제공

구분	제안규격
H/W	CPU/Memory 4Core 이상 / 8Gb 이상
	HDD Portal 500Mb 이상, Repository 100 Gb 이상
S/W	OS OS Independent, HTML5 이상 지원 브라우저
	WAS Tomcat 7.0, Weblogic 10 이상, Jeus 지원
	JRE JRE1.7 이상
	RDBMS Oracle 11g R2 이상, Tiberio 6.0 이상, PostgreSQL Client 10 이상

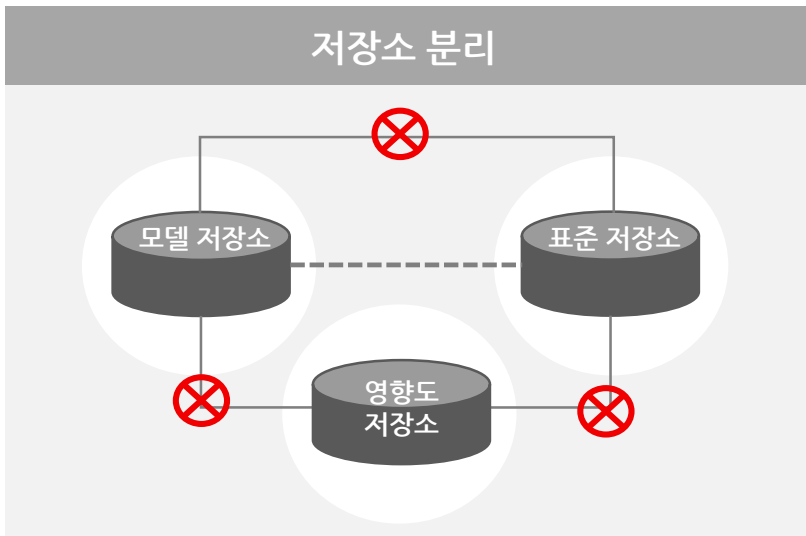
05

솔루션 특장점

솔루션 특징점

일반 시스템 구성

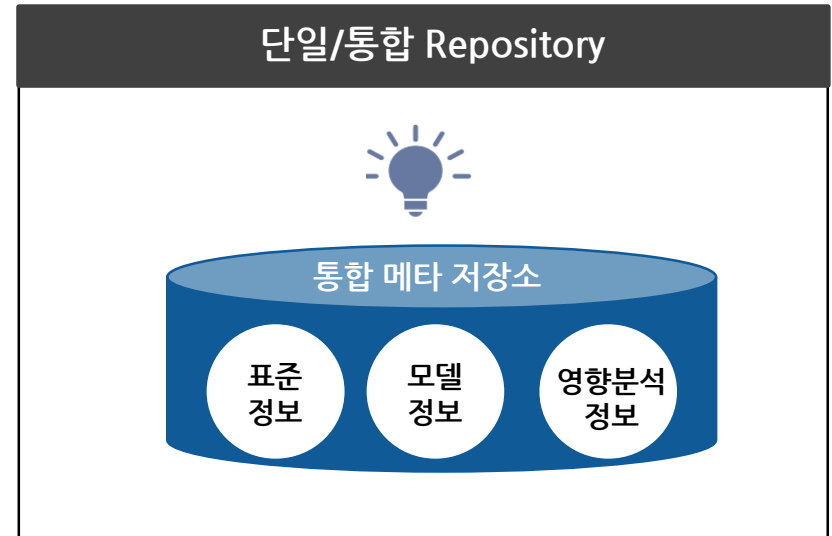
전사적 데이터 관리를 위한 기반 시스템 구축 시 필요한 Modeling Case Tool과 메타데이터 관리시스템, 어플리케이션 영향도 분석 솔루션은 일반적으로 이질적 벤더사로 독립적인 Repository로 구성되어 있음.



“이질적 벤더 간의 저장소 분리로 데이터 중복 관리 / 호환성 부족 / 실시간성 정보 교환 미흡 / 관리 포인트 증대”

제안사 시스템 구성

제안사는 국내 순수기술로 자체 Modeling Case Tool과 메타데이터 관리시스템, 어플리케이션 영향도 분석 솔루션을 개발하였으며 단일/통합 Repository로 구성되어 있음.

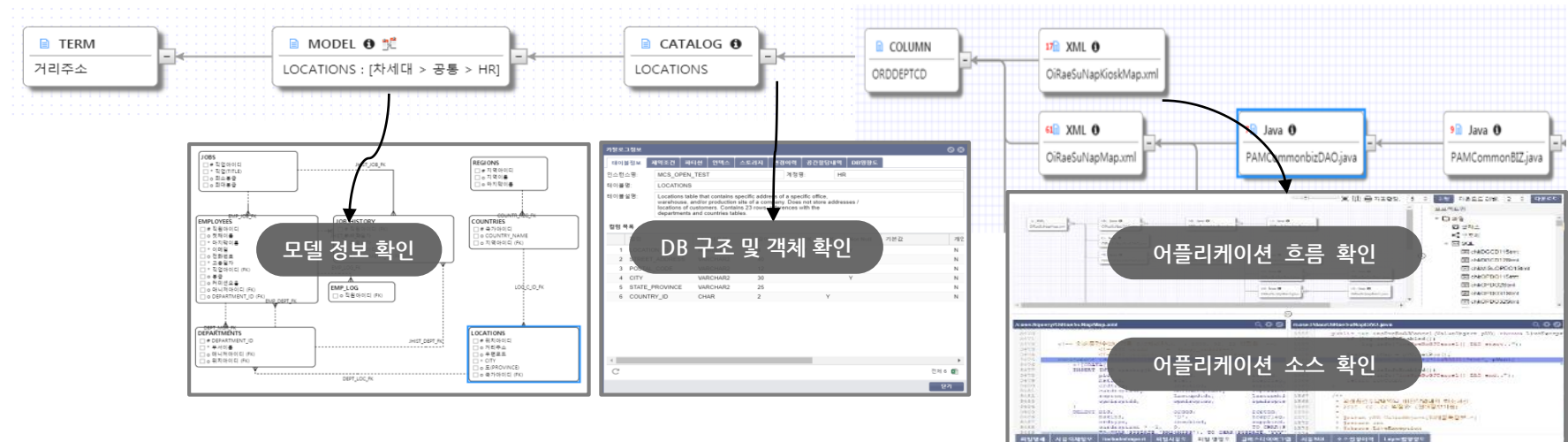
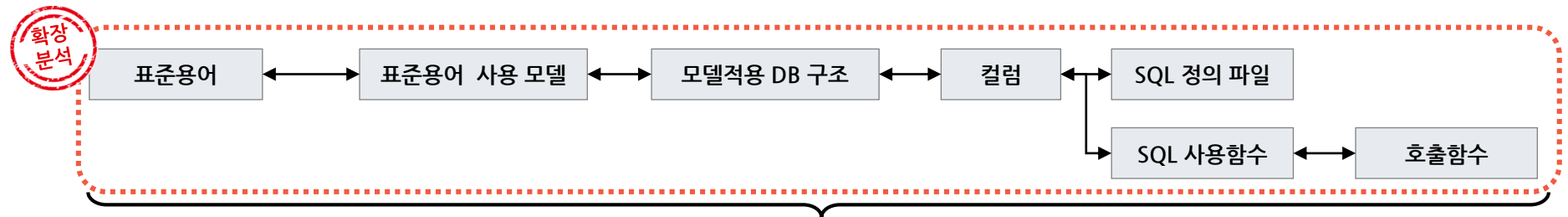


VS

“단일/통합 Repository 기반으로 솔루션간 데이터 중복 최소화 / 유지보수 점점 단일화 / 실시간성 정보교환”

메타 데이터 영향도 VISUALIZATION

DATAWARE™ AP#은 자사 모델링 도구 및 메타데이터 관리시스템과의 단일/통합 Repository 기반으로 구성되어 있어 데이터 표준으로부터 데이터 모델, 데이터베이스, 데이터 활용, 어플리케이션 소스까지 전체 메타 데이터의 변경 영향도 정보를 시각적인 형태로 제공하여 통합 관리·운영 시 **장애 예방과 장애 대응할 수 있는 프로세스를 제공**



타사 제품 간 비교표

제안사는 풍부한 구축경험을 바탕으로 효율적인 시스템 구축 프로세스 보유 및 높은 수준의 제품 완성도를 유지하고 있으며 제품개발에 직접 참여한 전문개발인력을 통한 탁월한 문제해결능력과 유연한 커스터마이징 서비스를 제공

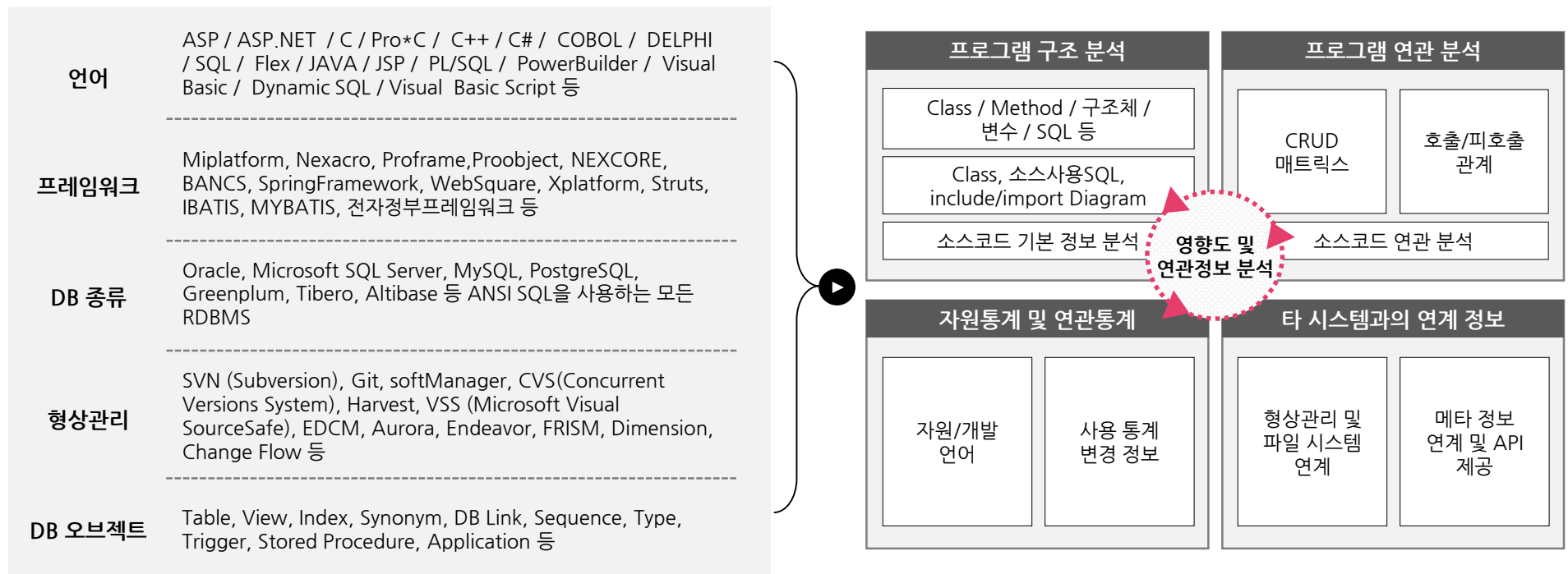
기능		DATAWARE™ AP#	C 제품	A 제품
시장점유율		▪ 중간	▪ 높음	▪ 낮음
타사 제품 대비 장점		▪ 분석엔진 성능 및 안정성 ▪ 확장 및 연계 용이성 ▪ 사용편의성	▪ UI(디자인)강점	▪ 상세분석 및 다양한 언어지원 강점
성능	분석 소요시간(10만본 소스기준)	▪ 1시간	▪ 7시간	▪ 2시간
	분석결과 조회 소요시간	▪ 평균 2~3초 내외	▪ 평균 10초 내외	▪ 평균 10초 내외
안정성 (자동 스케줄 및 장애 시 복구)		▪ 우수	▪ 보통	▪ 보통
클라이언트 환경		▪ 국제 웹 표준 지원	▪ C/S 환경으로만 제약	▪ 사용성 복잡
기술지원		▪ 우수(최적화 및 유지보수)	▪ 보통	▪ 미흡(최적화 X)
유연성 (시스템 연계)		▪ 우수(개발환경(IDE) 및 다양한 형상관리, 표준, 모델, 품질시스템 연계경험)	▪ 보통(자사제품 연계 : 올인원 솔루션)	▪ 미흡(외산)
라이선스 정책		▪ 서버 1식 + 분석대상 언어	▪ 상황 별 정책 변동	▪ 분석 추출 object 수

06

솔루션 주요기능

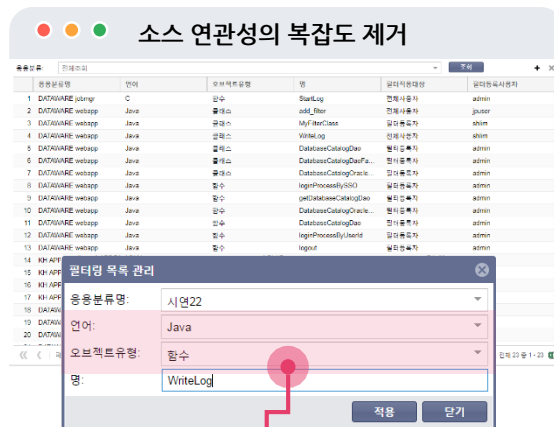
개발 환경 및 지원 언어

DATAWARE™ AP#은 다양한 언어의 어플리케이션 소스와 형상관리시스템, 프레임워크 등의 메타데이터들을 자동으로 수집 및 파싱하여 프로그램의 구조정보와 상호 연관정보를 시각화하고 체계화하여 보여주므로 신속한 어플리케이션의 이해와 상호 연관성을 분석할 수 있도록 지원

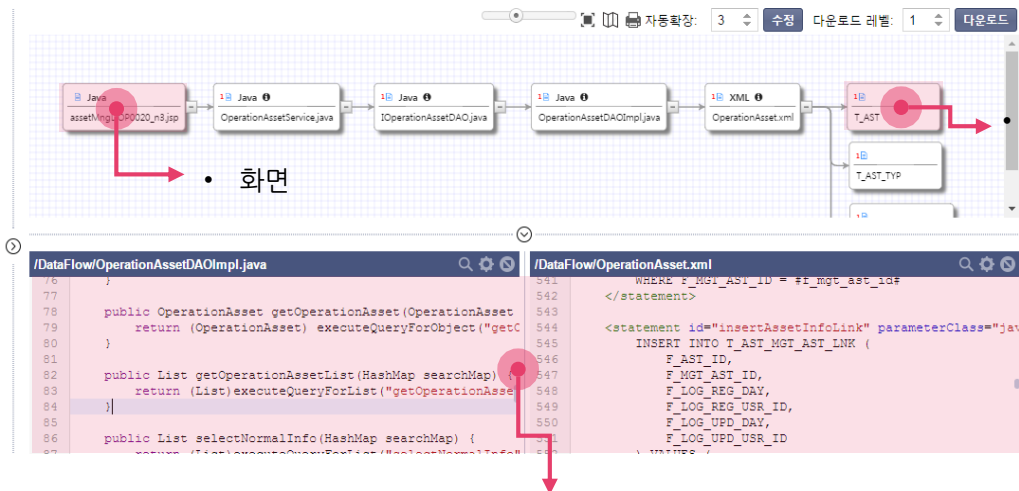


영향 / 연관 조회

- ✓ 필터링 기능을 통하여 불필요한 연관관계(오픈 소스/공통 모듈 등) 표시 제한 가능
- ✓ 어플리케이션 소스 간의 사용/참조관계와 어플리케이션과 데이터베이스 오브젝트와의 사용/참조관계를 다이어그램을 통해 시각화하여 보여줌
- ✓ 소스 변경 시 명확한 데이터 영향도 식별과 테스트 케이스 확인용



- 오브젝트별 필터링 대상 추가

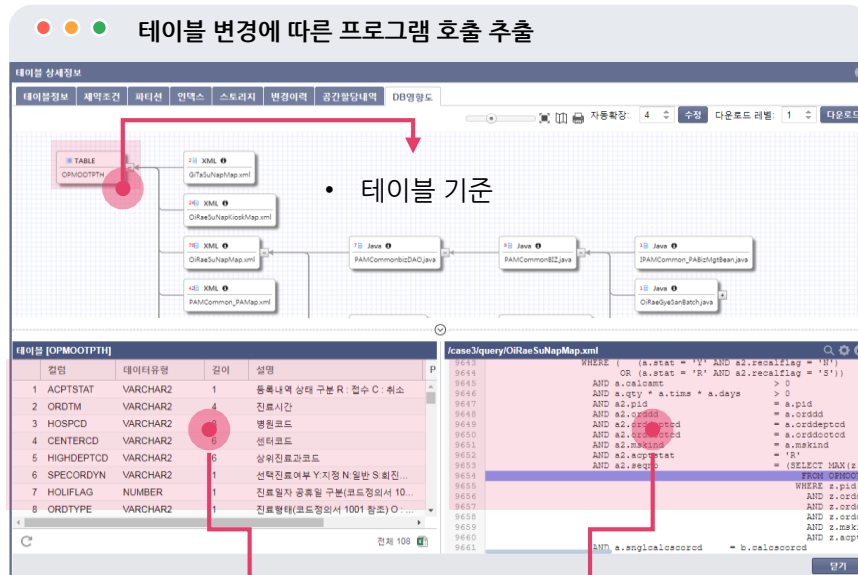


- 호출/피호출 소스 네비게이션

테이블

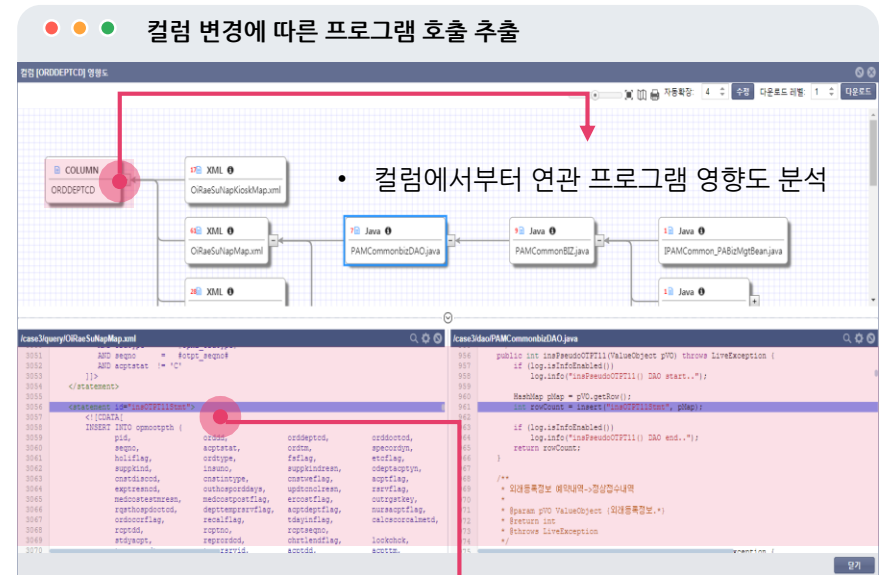
데이터베이스 (오브젝트) 분석

✓ 데이터베이스 오브젝트 기준으로 프로그램의 호출 관계 분석 정보를 제공



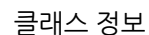
- 테이블의 컬럼(속성) 정보

- 소스 정보



- 조회된 다이어그램에서 노드 하나를 선택하면 하단 소스 뷰어에 호출/피호출 소스가 조회됨(색깔로 표시)

- ✓ 프로그램 소스 코드 파일 간의 'Include/Import' 관계 및 선택한 프로그램의 클래스 간의 상속 관계를 다이어그램 형식으로 확인할 수 있는 클래스 다이어그램을 지원



솔루션 주요기능

CRUD 매트릭스 자동 생성

- ✓ 프로그램-DB 객체 간의 CRUD (Create, Retrieve, Update, Delete) 관계를 매트릭스 형태로 제공
- ✓ 조회된 정보는 보고서로 출력하여 산출물로 활용

The screenshot displays the DATAWARE AP# interface. On the left, a '테이블 <-> 프로그램' (Table <-> Program) matrix shows relationships between tables and programs. The matrix has columns for programs like 'GIRaSuNapKioskMap.xml', 'OIRaSuNapKioskMap.xml', 'PAMCommon_PAMMap.xml', 'PAMCommon_PMTMap.xml', 'PAOGITaSuNapKioskMap.xml', 'SunSDataCopyMap.xml', 'TempMap.xml', 'WsonMaBachinVoonMap.xml', 'WsonMaSMSBachMap.xml', 'change_manage.ps', 'deletefileinfo.ps', and 'insertjobinfo.ps'. The rows represent tables: 'COMCDEPTM', 'COMCDSIDM', 'OPADCGMH', 'OPAMCARDH', and 'OPAMUNCOH'. The matrix cells contain 'R' (Retrieve), 'U' (Update), 'C' (Create), and 'D' (Delete).

On the right, a '프로그램 소스' (Program Source) window shows the source code for 'case3query/OIRaSuNapKioskMap.xml'. The code includes SQL queries and PL/SQL blocks. A red circle highlights a specific line in the code, which is linked to a table in the matrix below.

Below the matrix, a table titled 'OPAMCARDH' shows the table's structure. The table has columns: 'PID', 'ORDD', 'RCPTDD', 'RCPTNO', 'RCPTSEQNO', 'MSKIND', 'SEQNO', 'RCPTSEARKIND', 'STAT', 'HOSPFLAG', and 'ORDTYPE'. The table is linked to the 'OPAMCARDH' table in the matrix.

• 라인 정보

• 테이블 정보

1 CRUD Matrix 종류

- 업무 vs 테이블
- 업무 vs 테이블/컬럼
- 테이블 vs 프로그램
- 프로그램 vs 테이블/컬럼

2 CRUD Matrix 기능

- 정적 · 동적 SQL 자동 추출 및 분석

솔루션 주요기능

자원검색 분석

✓ 프로그램 또는 테이블 수준에서 어떤 자원을 사용하고 어떤 문자열이 있는지 확인할 수 있는 다양한 검색 기능을 제공

오브젝트 선택

문자열 검색

주석 선택

The interface displays a search results table with columns: Resource Name, Path, Operation, Table, File, Status, Count, and Action. The detailed view shows the source code of a Java file, with search results highlighted in blue. The search criteria are: Keyword: 'KH APPS', File: 'MA_FILE', and Search Scope: 'All Resources'.

1 자원 검색 종류

- 소스 객체 검색
- 소스 사용 SQL 검색
- 연관 (Depth 별) 검색
- 문자열/주석 검색

2 자원 검색 조건

- 파일 주석, 메서드 주석, 클래스 주석, CRUD, 프로그램과 DB의 오브젝트 이름, 타입, 키워드 등을 통해 사용자가 원하는 항목을 신속하게 찾을 수 있고 다양한 분석 정보에 접근 가능

프로그램 변경 이력 및 소스 비교

✓ 시점을 지정하여 변경/추가/삭제 된 이력 정보를 조회할 수 있으며, 프로그램 간 소스 비교를 통한 개발·운영 관리의 효율화 증대

변경 이력

응용분류	등록일자	최종수정일	파일명	버전	경로	변경형식	등록...
DATAWARE APPS	2012-05-01	2012-05-01 12:00:00	exec_storedpr...	1	/home/impact/src/impact...		
DATAWARE jobmgr	2012-05-01	2012-05-01 12:00:00	jobmgr.c	1	/home/impact/src/impact...		
DATAWARE webapp	2012-05-14	2012-05-14 10:03:11	jobmgr.c	3	/home/impact/src/impact...	UPDATE	
KH APPS	2012-05-14	2012-05-14 08:56:05	jobmgr.c	2	/home/impact/src/impact...	UPDATE	
DATAWARE	2012-05-15	2012-05-15 00:50:08	exec_storedpr...	2	/home/impact/src/impact...	UPDATE	
DATAWARE	2012-05-15	2012-05-14 10:05:16	jobmgr.c	4	/home/impact/src/impact...	UPDATE	

변경 형식: 선택

검색어:

조회:

• 변경 형식 정보

• 다양한 검색 조건

소스코드 비교

응용분류	최종수정일	등록일자	등록시간	파일버전	경로	변경형식
DATAWARE APPS	2012051410...	2012-05-15	09:45:14	4	/home/impact/src/impactMana...	UPDATE
DATAWARE jobmgr	2012051410...	2012-05-14	19:03:19	3	/home/impact/src/impactMana...	UPDATE
DATAWARE webapp	2012051408...	2012-05-14	18:01:39	2	/home/impact/src/impactMana...	UPDATE
DATAWARE	2012050112...	2012-05-01	12:00:00	1	/home/impact/src/impactMana...	

변경: 1 | 현재 위치: 1

소스 비교 | 이전 | 다음

파일명: %검색어%

파일명 | 파일크기 | 최종수정일

deletejobinfo pc | 14.1 KB | 2012-05-14

exec_storedproc c | 11.7 KB | 2012-05-14

exec_storedproc pc | 2.3 KB | 2012-05-15

file_util c | 25.4 KB | 2012-05-14

global_var c | 890 bytes | 2012-05-14

insertjobinfo c | 382.7 KB | 2012-05-14

insertjobinfo pc | 126.4 KB | 2012-05-14

jobmgr.c | 44.8 KB | 2012-05-14

parserinfo c | 8.2 KB | 2012-05-14

parsermgr.c | 11.9 KB | 2012-05-14

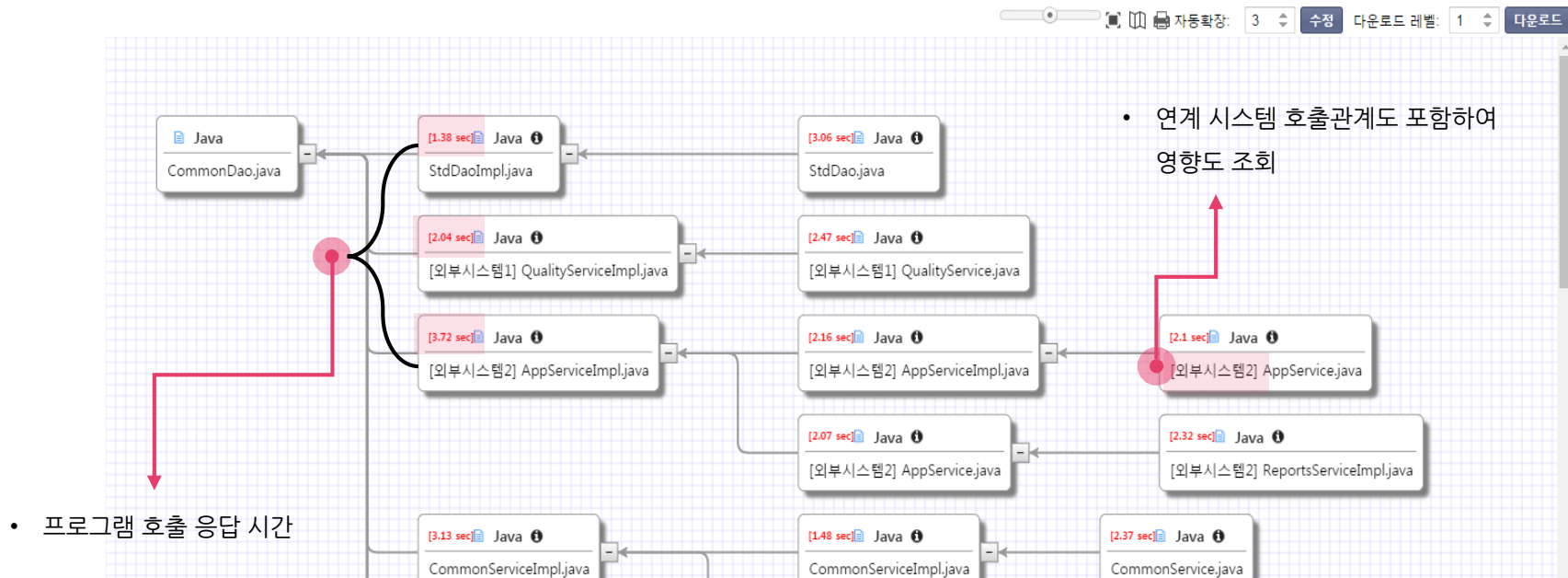
readparam c | 5.3 KB | 2012-05-14

변경 부분 표시(변경 전/후 소스)

• 변경 부분 표시(변경 전/후 소스)

프로그램 성능 데이터 표시

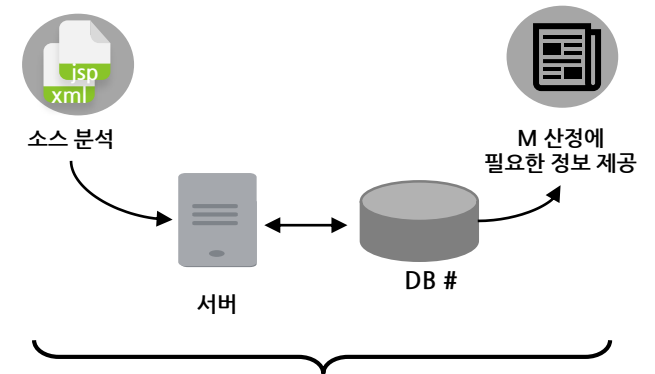
- ✓ 영향분석 데이터와 그와 매핑 되는 데이터인 관계 시스템 성능데이터 연계를 통해 화면 별 평균 응답시간 정보를 제공하며 그 외 기타 추가적으로 보고자 하는 연계 데이터 정보가 있을 경우, 협의와 커스터마이징을 통해 정보 제공이 가능하도록 지원



개발 M/M 산정

- ✓ 프로그램 소스코드를 수집 및 분석과정을 통하여 구축되는 정보를 토대로 프로그램 변경 시 개발에 필요한 M/M 산정에 필요한 정보를 보여줌
- ✓ M/M를 산정하는 기준이 될 수 있는 정보로는 코드 크기(LOC), 복잡도, 로직 Depth 등 정량적 분석 자료 제공

	파일경로	파일명	파일용량(...)	총라인	소스라인	주석라인	호출빈도	함수개수	최대복잡도	평균복잡도	최대 Depth	평균 Depth
13	/case3/action	ByeongSilJeongBoJoH...	6	146	107	39	73	2	1	1	0	0
14	/case3/action	ByeongWeonIJeongG...	7	176	119	57	55	4	1	1	0	0
15	/case3/action	DamDangJaSuNapJoH...	10	234	171	63	109	5	1	1	0	0
16	/case3/action	ERHwanJaGwanRiActi...	9	214	75	139	43	1	1	1	0	0
17	/case3/action	ERPcNfmAction.java	4	107	75	32	30	2	1	1	0	0
18	/case3/action	EuiSaBoChungJinRyoll...	3	78	48	30	17	1	1	1	0	0
19	/case3/action	EuiSaHoliMasterAction...	8	185	137	48	84	5	1	1	0	0
20	/case3/action	EuiSaJinRyollJeongGw...	14	322	230	92	162	8	1	1	0	0
21	/case3/action	GaToiWeonGwanRiActi...	8	218	157	61	64	5	1	1	0	0
22	/case3/action	GaeInGyeYakMiSuSuN...	53	1141	905	236	660	20	5	2.8	2	0.81
23	/case3/action	GiTaSuNapAction.java	21	497	309	188	165	10	5	3	1	0.9
24	/case3/action	GongTongJoHoiAction.j...	28	580	351	229	257	7	9	3.43	2	0.88
25	/case3/action	HabChoiBunChoiAction...	16	366	287	79	233	7	1	1	0	0
26	/case3/action	HallnCodeGwanRiActio...	8	212	153	59	92	4	6	3.75	2	1.05
27	/case3/action	HallnCodeSuigCodeG...	7	176	126	50	50	3	1	1	0	0
28	/case3/action	HlmsIfSampleAction.java	11	240	211	29	198	2	19	19	1	1
29	/case3/action	HomePageGwanRiActi...	20	439	396	43	255	10	1	1	0	0
30	/case3/action	HttpConHelper.java	2	84	82	2	6	2	1	1	0	0
31	/case3/action	HwanJaGiBonJeongBo...	62	1458	1106	352	826	33	1	1	0	0
32	/case3/action	HwanJaGiChoJeongBo...	5	143	103	40	46	2	6	6	1	1
33	/case3/action	HwanJaSangSeAction.j...	11	290	206	84	82	6	5	3.33	1	1
34	/case3/action	IfsKhnmMcMgrCallbacK...	3	83	44	39	0	7	1	1	0	0
35	/case3/action	IfsKhnmMcMgrStub.java	357	7255	6215	1040	883	21	1	1	0	0
36	/case3/action	lpWeonDeungRokActio...	59	1251	910	341	732	26	1	1	0	0



- LOC와 IF문의 Depth를 기준으로 복잡도 도출
- If 복잡도는 프로그램의 로직의 복잡도로서 프로그램 로직이 얼마나 복잡한가를 판단하는 기준

소스 Lifecycle 관리

- ✓ 운영 시스템에서 사용되지 않는 소스를 식별하여 관리자 또는 일반 사용자에게 정보를 제공하며 이를 통해 소스 Lifecycle 관리할 수 있음

미사용 테이블 보고서

스키마: (SERVER) ORCLUTF8, (INSTANCE) ORCLUTF8, (SCHEMA) PAMDBA

조회

보고서

테이블	애플리케이션	스키마
AAA	KH APPS, KH sample	PAMDBA
APPENDTEST	KH APPS, KH sample	PAMDBA
AUTOCONV	KH APPS, KH sample	PAMDBA
AUTOMAP	KH APPS, KH sample	PAMDBA
A_INTEGRITY_TEMP	KH APPS, KH sample	PAMDBA
A_SAVEUPDATES	KH APPS, KH sample	PAMDBA
A_SEL100RAND	KH APPS, KH sample	PAMDBA
A_SEL100SEQ	KH APPS, KH sample	PAMDBA
A_TENPCT	KH APPS, KH sample	PAMDBA
A_TINY	KH APPS, KH sample	PAMDBA
A_UNIQUES	KH APPS, KH sample	PAMDBA
A_UPDATES	KH APPS, KH sample	PAMDBA
BESUGA_TEMP	KH APPS, KH sample	PAMDBA
CARD_TEMP	KH APPS, KH sample	PAMDBA

«

<

페이지 1

/ 7

>

»

↺

Page size: 100

미사용 함수 보고서

보고서

프로그램경로	프로그램명	함수
/home/impact/sample/dataware/src	Ognl.java	substring
/home/impact/sample/dataware/src	Ognl.java	length
/home/impact/sample/dataware/src/encore/meta/a...	NodeExpander.java	DuplicationCallChe...
/home/impact/sample/dataware/src/encore/meta/a...	NodeExpander.java	has
/home/impact/sample/dataware/src/encore/meta/a...	NodeExpander.java	nextLevel
/home/impact/sample/dataware/src/encore/meta/a...	NodeExpander.java	print
/home/impact/sample/dataware/src/encore/meta/a...	CommonAspect.java	pushAfterRequest
/home/impact/sample/dataware/src/encore/meta/a...	MailAspect.java	sendMailForRequest
/home/impact/sample/dataware/src/encore/meta/a...	DatabaseMessage...	get18NSqlQuery
/home/impact/sample/dataware/src/encore/meta/a...	DatabaseMessage...	resolveCode
/home/impact/sample/dataware/src/encore/meta/a...	DatabaseMessage...	init
/home/impact/sample/dataware/src/encore/meta/a...	DatabaseMessage...	getMessage
/home/impact/sample/dataware/src/encore/meta/a...	MailConfiguration.java	mailAspect

미사용 프로그램 보고서

보고서

파일명	경로
change_manage.c	/home/impact/src/Impact/Manager/JobMgr
db_util.c	/home/impact/src/Impact/Manager/JobMgr
deletefileinfo.c	/home/impact/src/Impact/Manager/JobMgr
deletejobinfo.c	/home/impact/src/Impact/Manager/JobMgr
exec_storedproc.c	/home/impact/src/Impact/Manager/JobMgr
global_var.c	/home/impact/src/Impact/Manager/JobMgr
insertjobinfo.c	/home/impact/src/Impact/Manager/JobMgr
test.sh	/home/impact/src/Impact/Manager/JobMgr

분석 작업 모니터링

✓ 분석 작업에 대해 등록된 스케줄에 대하여 그 진행상황을 모니터링 할 수 있는 기능을 통해 사용자가 작업의 진행 상태를 쉽게 파악

• 작업에 대한
진행 상태

응용분류:

KH APPS

▼

조회날짜:

조회

응용분류명	작업유형	작업단계	작업시작일시	작업종료일시	작업결과	소요시간	오류메시지
KH APPS	재분석	연관관계분석	2018-06-21 11:31:33	2018-06-21 11:31:52	성공	00:00:19	
KH APPS	재분석	DB적재	2018-06-21 11:28:02	2018-06-21 11:31:32	성공	00:03:30	
KH APPS	재분석	파싱	2018-06-21 11:25:56	2018-06-21 11:28:02	성공	00:02:06	
KH APPS	재분석	연관관계분석	2018-06-20 11:31:33	2018-06-20 11:31:52	성공	00:00:19	
KH APPS	재분석	DB적재	2018-06-20 11:28:02	2018-06-20 11:31:32	실패	00:00:00	Execute Stored Procedure Failed
KH APPS	재분석	파싱	2018-06-20 11:25:56	2018-06-20 11:28:02	성공	00:02:06	
KH APPS	재분석	연관관계분석	2018-06-19 11:31:33	2018-06-19 11:31:52	성공	00:00:19	
KH APPS	재분석	DB적재	2018-06-19 11:28:02	2018-06-19 11:31:32	성공	00:03:30	
KH APPS	재분석	파싱	2018-06-19 11:25:56	2018-06-19 11:28:02	성공	00:02:06	
KH APPS	신규분석	연관관계분석	2018-06-18 11:31:33	2018-06-18 11:31:52	성공	00:00:19	
KH APPS	신규분석	DB적재	2018-06-18 11:28:02	2018-06-18 11:31:32	성공	00:03:30	
KH APPS	신규분석	파싱	2018-06-18 11:25:56	2018-06-18 11:28:02	성공	00:02:06	

• 실패 시 오류 확인

• 작업에 대한
상세 이력

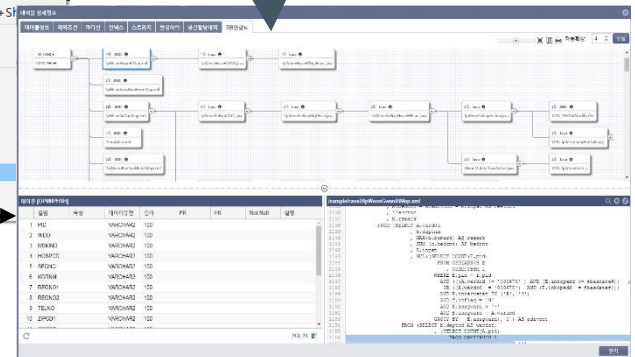
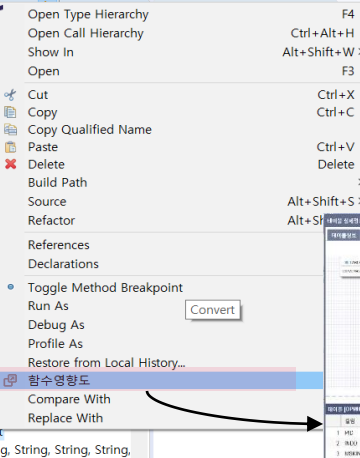
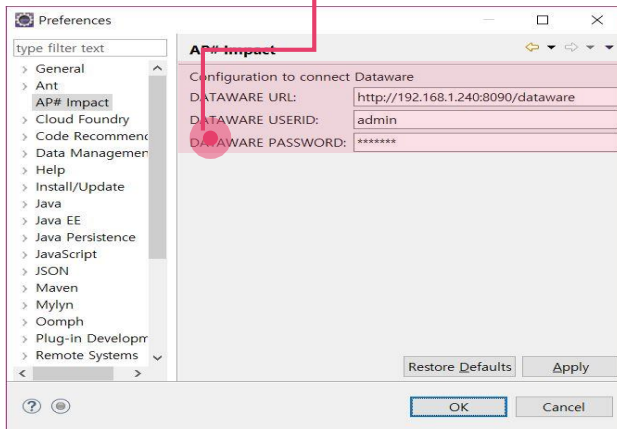
영향받는프로그램	경로	작업시작일시	작업종료일시	소요시간	건
Insert Xml Information		2018-06-18 11:25:20	2018-06-18 11:25:20	00:00:01	35077
Insert Sql Object Information		2018-06-18 11:28:21	2018-06-18 11:28:21	00:01:05	49646
Insert Logic Detail Information		2018-06-18 11:28:18	2018-06-18 11:28:18	00:00:00	0
Insert include/import Information		2018-06-18 11:28:18	2018-06-18 11:28:18	00:00:03	1920
Insert Logic Object Information		2018-06-18 11:28:16	2018-06-18 11:28:16	00:00:02	19613
Insert Source Object Information		2018-06-18 11:28:07	2018-06-18 11:28:07	00:00:09	34471
Insert File Information		2018-06-18 11:28:06	2018-06-18 11:28:06	00:00:01	281

• 작업에 대한 내역
정보 엑셀 다운로드

Eclipse AP# Plugin

✓ IDE 환경에서 어플리케이션 영향도 분석 기능을 사용하기 위해 전용 Eclipse Plugin 기술을 제공

영향도 정보 즉시 확인을 위한 접속 정보 세팅

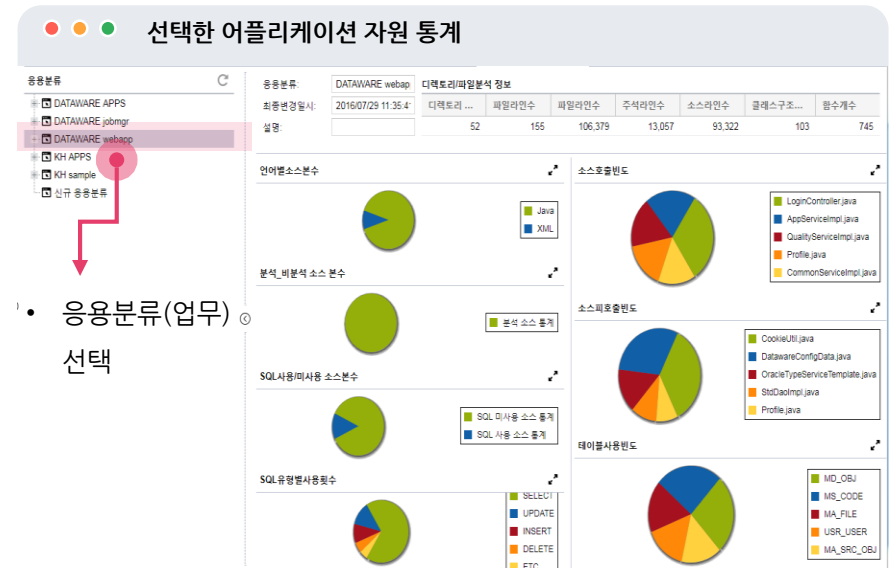


프로그램 분석 종합 정보

- ✓ 업무 내에 존재하는 소스의 총 본 수, 총 라인/주석 라인/소스 라인 수, 소스 객체의 개수, 언어별 소스 본 수, 사용SQL 정보 등 프로그램의 현황 및 통계 정보를 제공하며 이는 제공 S/W의 규모를 측정하는데 활용



- 전체 어플리케이션 자원 통계 조회 및 PDF, WORD, EXCEL 출력



- 선택한 응용분류(업무)에 대한 자원 통계 조회

- ✓ 프로그램 정의서, 테이블 정의서 등 주요 관리문서를 정의하고 자동으로 생성하여 개발자의 문서부담을 줄여줌

사용 테이블 보고서

[보고서](#)

프로그램 경로	프로그램명	테이블명	CRUD
/home/dft/APPS/Manager/comlib	db_util pc	MA_JOB_LOG	U
/home/dft/APPS/Manager/comlib	db_util pc	MA_JOB_LOG	C

사용 컬럼 보고서

[보고서](#)

프로그램 경로	프로그램명	사용데이터명	사용컬럼명	사용형태	라인
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	ACCT_ID	INTO 1,635
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	ACCT_ID	WHERE 1,739
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	AVAL_END_DT	INTO 1,628
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	AVAL_ST_DT	INTO 1,629
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	DB_ID	INTO 1,634
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	DB_ID	WHERE 1,738
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	INST_ID	INTO 1,633
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	INST_ID	WHERE 1,737
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	LINE_NO	INTO 1,630
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	MESSAGE	INTO 1,631
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	OBJ_ID	INTO 1,627
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	SVR_ID	INTO 1,632
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_ANAL_LOG	SVR_ID	WHERE 1,736
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_INST	DB_TYPE	SELECT 134
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_INST	INST_ID	WHERE 138
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_INST	SVR_ID	WHERE 137
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_INST	SYS_ID	SELECT 134
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_OBJ	ACCT_ID	WHERE 273
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_OBJ	DB_ID	WHERE 272
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_OBJ	DEL_YN	WHERE 269
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_OBJ	INST_ID	WHERE 271
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_OBJ	OBJ_ID	SELECT 258
/home/dft/AI	/home/dft/APPS/DBPars/MetaSharpMgr	select_source.pc	MD_OBJ	OBJ_ID	WHERE 275

<<
<
 페이지 1 / 22
 >>
>
 Page size: 100

현재 2131 중 1 - 100

07 도입 후 기대효과

도입 후 기대효과

분야별 기대효과

IT 관리

- 통합 영향도 분석/변경 시각화 정보 및 자동화 문서 제공으로 유지보수비용 절감
- IT 어플리케이션 개발 / 관리 정보의 유지
- Outsourcing에 대한 제어력 향상

프로젝트 관리

- 업무 변화에 대한 적응력 향상
- 업무 이해속도의 증가
- 프로젝트 통제력 강화

개발

- 다른 개발자의 소스코드 이해
- 전체적인 어플리케이션 Road Map 이해로 개발 생산성 향상
- 개발 및 유지보수 시 발생하는 비효율적인 반복작업의 제거

품질 관리

- 정의된 표준화 적용의 용이성 및 개발 표준에 대한 품질 관리 기능 강화로 어플리케이션 품질 향상
- 프로젝트의 위험요소 파악

08

구축사례

✓ CSR 및 개발환경(빌드 & 배포)과 연계하여 프로세스를 통합 구성한 사례를 보유하고 있으며 외부 시스템과의 인터페이스를 통한 메타정보 간의 영향도 분석을 통한 최적의 개발·운영 프로세스를 지원

사업명 : L통신 개발 영향도분석 자동화 구축 프로젝트

- 분석/설계단계에서 CSR을 기반으로 개발해야 할 소스에 대해서 미리 영향분석 수행
- 배포단계에서 배포대상 파일에 대해서 영향분석을 하지 않은 파일은 배포할 수 없도록 통제

▶ 품질점검정보 ※적합여부 구분: Y-적합, N-부적합, NA-측정대상 아님

구분	항목	적합여부	측정값
통시 점검	소스표준 준수수(CodePrism)	Y	직접 파일 개수 : 4, 예외처리 파일 개수 : 0
	개발자 테스트(TestLink)	N	테스트케이스 등록개수 : 0, 테스트 수행개수 : 0, Pass 개수 : 0, Fail 개수 : 0
	테스트수행 커버리지(TrueCoverage)	NA	
	3자테스트(Redmine)	N	미조직결합수 ; , 조직결합수 ;
	코드 복잡도(SONAR)	Y	
	코드 리뷰(Review board)	N	
	영향분석(AP#)	N	대상 개수 : 4, 완료 개수 : 2
스테이징 점검	회귀테스트 (UTAS)	NA	메시지 : 배포일 0-3일내 수행된 테스트가 없습니다.

C#로 배우는 딥러닝						
번호	이름	C#명	주제	언어	비고	강연회/강연장
190701	이정호	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.01
190702	C#190702_7674	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.02
190703	C#190703_1430	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.03
190704	C#190704_1403	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.04
190705	C#190705_86207	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.05
190706	C#190706_14497	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.06
190707	C#190707_19469	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.07
190708	C#190708_74693	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.08
190709	C#190709_4336	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.09
190710	C#190710_19979	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.10
190711	C#190711_21658	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.11
190712	C#190712_17323	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.12
190713	C#190713_21658	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.13
190714	C#190714_14404	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.14
190715	C#190715_15361	딥러닝을 위한 딥러닝 프레임워크	딥러닝	C#	이정호	2017.08.15

배포대상 파일 목록

[illegible]

구축사례

고객사 적용 사례

- ✓ CSR 및 개발환경(빌드 & 배포)과 연계하여 프로세스를 통합 구성한 사례를 보유하고 있으며 외부 시스템과의 인터페이스를 통한 메타정보 간의 영향도 분석을 통한 최적의 개발·운영 프로세스를 지원

case 2 : 빌드&배포 전 영향분석

사업명 : K은행 데이터품질관리체계구축 프로젝트

- 형상관리 솔루션(프리즘)과 연계하여 소스코드를 수정하기 전 변경대상 소스에 대한 영향분석을 미리 수행함으로써 함께 수정되어야 할 소스가 누락되지 않았는지 확인
- 소스 수정이 완료되고 변경된 프로그램이 배포되기 전 영향분석을 수행하여 함께 배포 되어야 할 프로그램이 누락되지 않았는지 확인



감사합니다.
Thank you!

EN-CORE Inc. is "Total Data Service Provider." We, as one of the leading data scientists groups in Korea, are offering a wide-range of services such as consulting, education, solutions, and technology information based on data ranging from planning the corporation IT systems to analyzing, designing, developing, transitioning, managing, performing governance, and data servicing.

AP
dataware
impact
analysis

en-core co.,ltd.

06648 | 42, Seocho-daero 46-gil, Seocho-gu, Seoul, Republic of Korea

office +82 2 754 7301 | fax +82 2 754 7305 | web www.en-core.com