

Smart Migration Solution

앱스마트워커

(AppSmartWorker)

2014.03



DAWIN ICT
information & communication



Contents

1. IT 환경의 변화
2. U2L 검토배경 및 목적
3. U2L 마이그레이션 시 고려사항
4. SM-Works의 구조적 이해
5. SM-Works의 특징
6. 룰 기반 변환 방안
7. 매핑 룰에 따른 변환 결과
8. IP Address 변환 방안
9. 수작업과 자동화 도구의 비교
10. 기대효과
11. 교육 및 기술 이전

IT 환경의 변화



IT환경의 변화는 비용절감, 효율성 증대와 비즈니스 민첩성의 가속화로 진화하고 있습니다. 또한 x86, Linux 기술의 성숙에 따른 Unix-to-Linux 에 대한 Needs가 증가하고 있으며, 클라우드 컴퓨팅의 등장과 활성화로 가상화 기술을 통한 저 활용 장비 통합과 유휴장비 재활용에서 근래에는 클라우드 인프라 기반으로 신규구축과 마이그레이션이 필수요소로 부각되어지고 있습니다.

기업 활동 환경 변화 요인

기업 경쟁력 확보에 총력

급속한 기술 변화

시스템 성능 요구 사항

신규 전략적 제휴

유지관리비용 절감

마이그레이션의 이슈사항

- 시스템 안정성 확보
- 비용산정 및 소요기간
- 기술확보
- 데이터 정합성 확보
- 비즈니스 프로세스 일관성
- 성능개선
- 신기술 수용
- 검증 방법

Customer Challenge

서비스 안정성
고가용성 확보

획기적 비용절감

성능 증가 및
기술만족도 향상

개발 / 유지보수
용이성

성공적
마이그레이션으로
최적화된
Infrastructure
구현

U2L 검토배경 및 목적



Unix-to-Linux 마이그레이션에 대한 필요성은, 과거처럼 Application Program과 Database를 중점적으로 고려하는 것이 아닌 시장환경과 경제상황이 악화되는 지금 시점에서는 유지비용이 많이 소요되는 낡은 서버부터 대상으로 검토한다는 점입니다. “UNIX 서버를 값싸고 성능이 뛰어난 Linux 서버로 마이그레이션”을 하므로 어플리케이션은 그대로 수평적 이동이 가능하고 상용 데이터베이스, 하드웨어 등을 완전한 Linux O/S 환경으로 교체함으로 획기적인 비용절감 효과와 서비스의 안정성과 고 가용성을 확보할 수 있는 최적의 방안이 될 것 입니다.

검토 배경

오픈 소프트웨어와 상용 리눅스의 비교 시 유사

기존의 HP,Sun,IBM,Fujitsu 서버등에
오픈 리눅스 시스템이 탑재되어 운영

Oracle DBMS를 MySQL 또는 PG-SQL로 전환이
용이

기존 어플리케이션이 오픈 리눅스에서
Win-Back 가능

클라우드 컴퓨팅 및 대형 포털을 중심으로
오픈 소프트웨어를 적용하여 이미 검증 완료

대부분의 대기업들이 운영시스템의 상당 부분을
오픈 리눅스로 교체하고 있으며,
비용 절감효과를 50% 이상 실현

목적

Infra
Software
&
Application
Program

Linux OS, x86 서버의 서비스 안정성,
고 가용성 확보

표준화,최적화된 Infra기반 확보

하드웨어 구입 비용의 절감

저전력,저발열량으로 전력사용의절감

데이터베이스 구입비용 및
유지보수 비용의 절감

WAS 구입 비용 및 유지보수비용절감

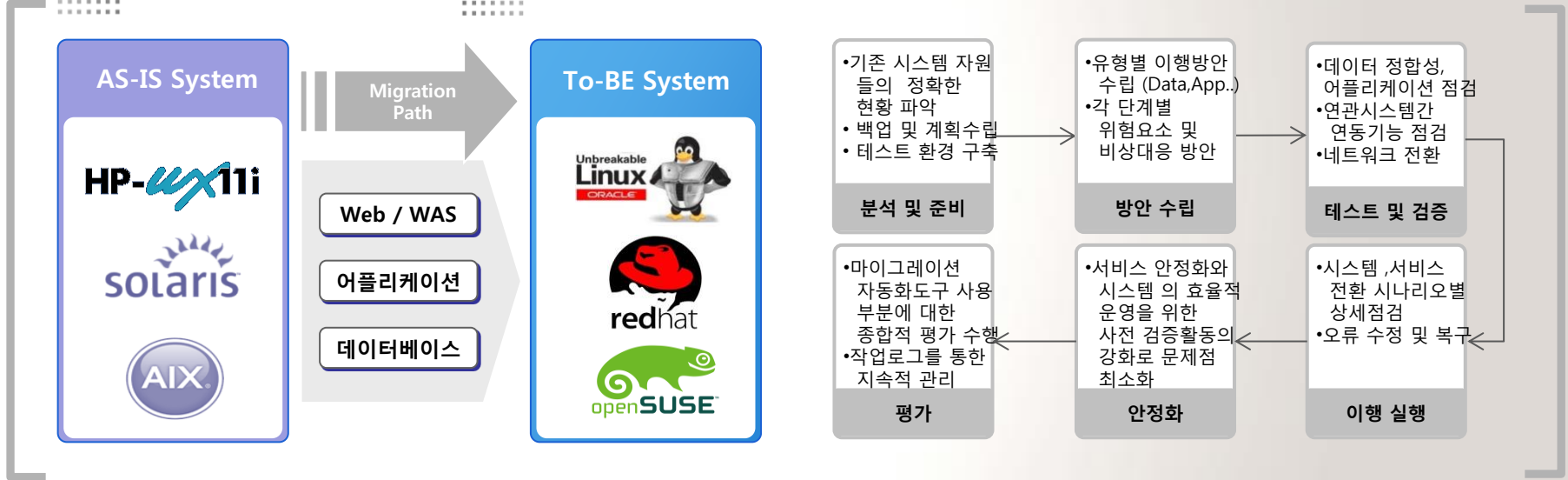
보안성 강화

획기적인
비용절감

U2L 마이그레이션 시 고려사항



일반적인 마이그레이션 절차



고려사항



기존 운영중인 응용프로그램의 소스프로그램이 없이 실행파일만 존재하는 경우

기존 사용중이던 패키지 소프트웨어를 신규 O/S에 설치가 불가할 경우

기존 운영중인 응용프로그램의 변경이 필요한데 개발자 지원에 제약이 있는 경우

기존 운영중인 응용프로그램에서 사용하는 라이브러리가 신규 O/S에서 지원되지 않는 경우

대응방안



De-compiler 적용

패키지 소프트웨어의 원백

전문가의 가이드라인에 따른 자동화 변환 도구의 사용으로 정확성 제고

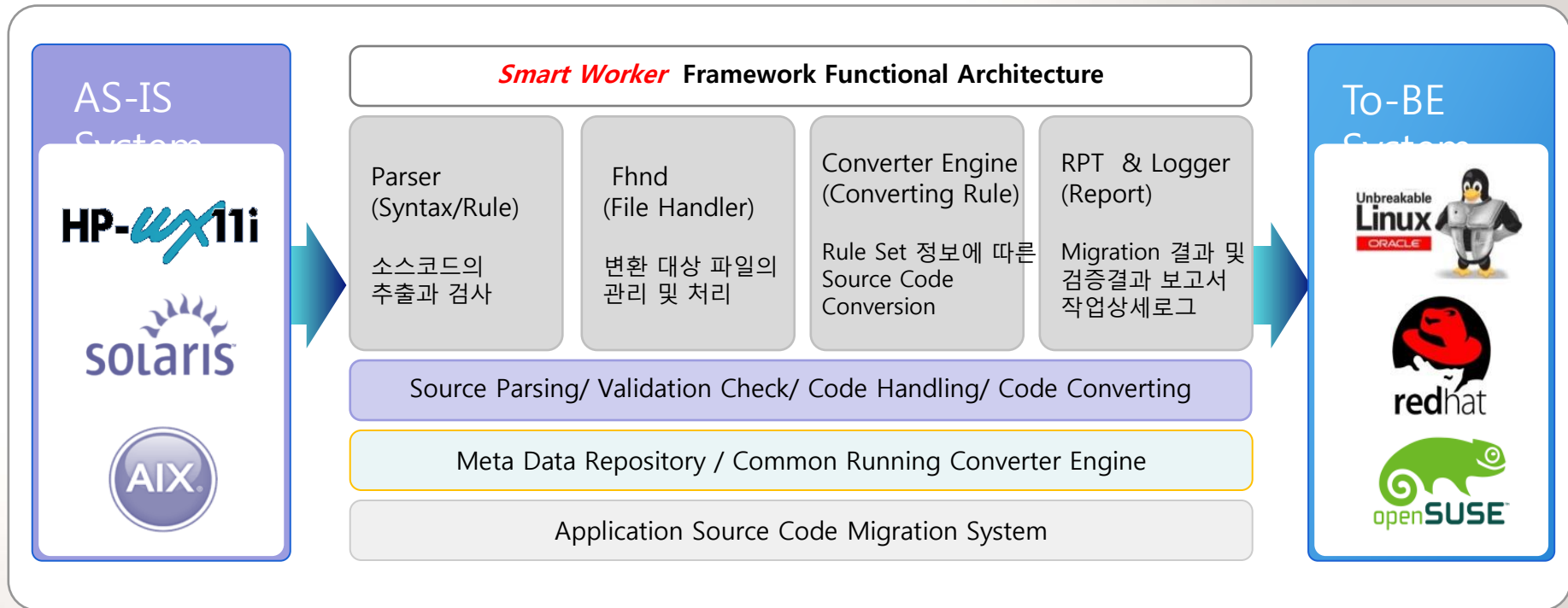
POSIX1 계열의 시스템 라이브러리와 유사

앱스마트워커의 구조적 이해

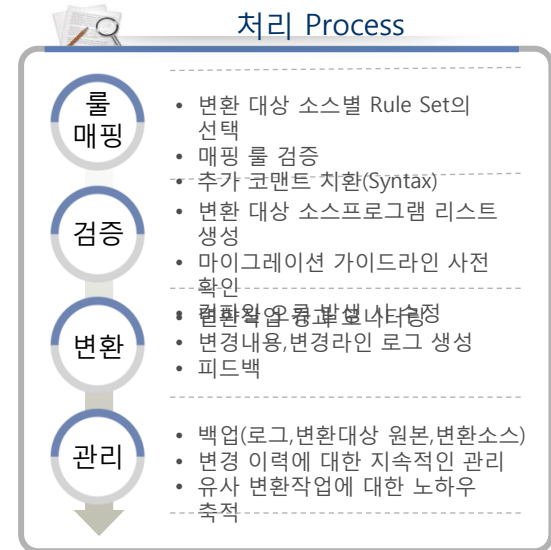
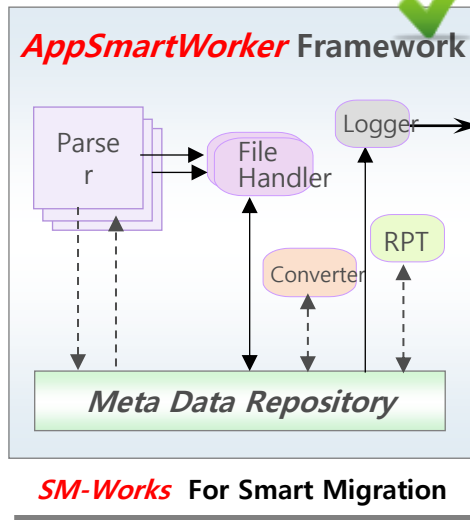


Smart Migration Solution SM-Works 는 다양한 시스템 운영환경에서의 마이그레이션 프로젝트를 경험하면서 축적한 노하우와 지식들을결집하여 개발한 마이그레이션 자동화 도구로서 Unix to Linux, H/W Vendor들의 Win-Back으로 인한 마이그레이션 필요에 유연하게 대응하고 비용과 전환 공수의 절감에 획기적인 효과를 가져다 주는 솔루션입니다.

AppSmartWorker Framework



앱스마트워커의 특징



비용 절감 측면

- 수작업에 대한 공수산정 어려움 극복
- 전체 마이그레이션 작업공정의 단축으로 비용절감
- Application Porting 기간 단축으로 전환공수 절감
- 비용분석을 위한 기초데이터 확보
- 다양한 마이그레이션 프로젝트 수행의 경험과 지식
- 수작업의 발생요류들을 사전에 방지
- 완벽한 변환기능의 안정적 제공으로 효율성 극대화
- Build환경과 변환작업이 수행되는 환경이 동일

안정성 측면

AppSmart Worker 특장점

성능 및 기능적 측면

- Low Latency Migration이 가능한 변환성능의 제공
- 반복 검증 및 개선을 위한 기초자료 제공
- 마이그레이션 유형별 테스트 및 디버깅 가능
- Point in time Recovery 기능으로 시점별 복구 가능
- 마이그레이션 리스크 최소화
- 마이그레이션 프로젝트 범위,필요기간 산정을 위한 기준데이터 제공
- 체계적인 룰 관리 및 작업내역에 대한 지속적 관리 가능
- 직관적 사용자 인터페이스 제공으로 개발자가 직접

운영 및 관리적 측면

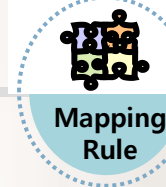
룰 기반 변환 방안



Mapping Rule Set의 내용

```
@<192.168.10.140><192.168.10.250>@  
,NULL,||memcpy(|| @<NULL><">@<memcpy><strcpy>@  
#include <stdlib.h>|| @<command><insertup><//앞라인 추가>@  
#include <stdlib.h>|| @<command><insertdn><//뒷라인 추가>@  
extern char *delete_func();|| @<command><delete>@  
char *sys_errlist[];|| @<command><comment><u2l>@  
,sys_errlist[^1] || @<sys_errlist[^1><strerror(^1)>@  
sprintf(^1,"^2",^3); ||  
@<ifstrcmp==><^1><^3>@<sprintf><S_sprintf>@  
snprintf(^1,^2,"^3",^4); ||  
@<ifstrstr==><^1><^4>@<snprintf><N_nprintf>@  
format_func("^1"^2); ||  
@<ifstrcnt!=><%^1><,^2>@<command><write><filename>@  
fpos_t ^1=0; || @<^1=0;><^1;>@<varsave><^1><$1>@  
=$1; || @<$1><$1.__pos>@
```

"C"
Application
Program
UNIX



Linu
X

AP변환 자동화도구

HP-ux11i



- 자동 Conversion
- Conversion 요소 확인
- Conversion 요소 위치
- IP Address 식별 및 변환
- 패턴 분석 및 매칭
- 작업 상세 로그
- 대상 소스 백업

수작업 변환



- 자동화 도구 이용 가능시
Rule Set 등록 유무 확인 후 매핑하여 변환 처리
- 자동화 도구 지원 불가능시
자동화 도구로 부터의 리스트를 참조하여
개발자가 수작업으로 전환
- 마이그레이션 작업진행 로그와 리포트의 참조
추가적인 Rule Set 등록으로 자동화로 전환

매핑 룰에 따른 변환 결과(예시)



AS-IS Source Program (예시)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>

#define IP_SET "192.168.10.140"

extern char *sys_errlist[];
extern char *delete_func();

long
main(long argc, char *argv[])
{
    char test=NULL;
    char aaa[10];
    char bbb[10];
    fpos_t lp_current = 0;

    memset(bbb ,NULL , strlen(bbb)); /* test */
    memcpy(test , NULL ,strlen(test));
    strcpy(test , NULL); //comment test
    gl_file_size = lp_current;
    printf("hello %d %s %n",errno, sys_errlist[ errno ]);
    sprintf(bbb, "hello %s %n",bbb);
    snprintf(bbb, sizeof(bbb),"hello %s %s %n",bbb,
test,ddd);
    format_func("count %d %s
% n" ,errno,strerror(errno),test);
    return 0;
}
```



범례

변환대상 매칭 패턴

To-BE Source Program (예시)

```
#include <stdio.h>
//앞라인 추가
#include <stdlib.h>
//뒷라인 추가
#include <string.h>
#include <errno.h>

#define IP_SET "192.168.10.250"

//u2l extern char *sys_errlist[];

long
main(long argc, char *argv[])
{
    char test=NULL;
    char aaa[10];
    char bbb[10];
    fpos_t lp_current;

    memset(bbb ,NULL , strlen(bbb)); /* test */
    strncpy(test , "" ,strlen(test));
    strcpy(test , NULL); //comment test
    gl_file_size = lp_current._pos;
    printf("hello %d %s %n",errno, strerror(errno));
    S_sprintf(bbb, "hello %s %n",bbb);
    N_snprintf(bbb, sizeof(bbb),"hello %s %s %n",bbb,
test,ddd);
    format_func("count %d %s
% n" ,errno,strerror(errno),test);
    return 0;
}
```



범례

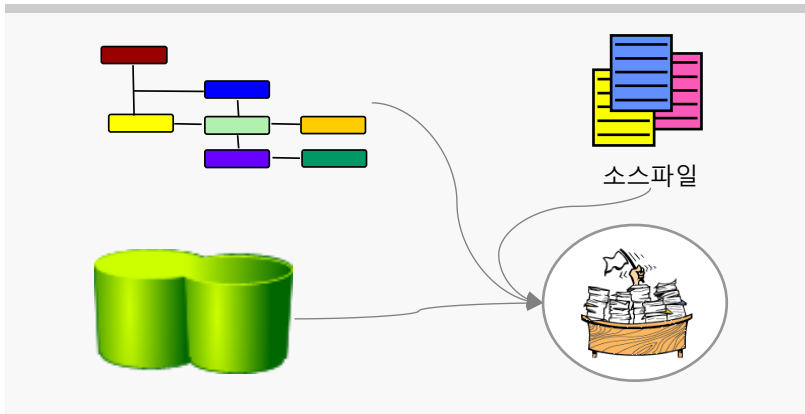
매핑 룰에 따른 변환결과

IP Address 변환 방안



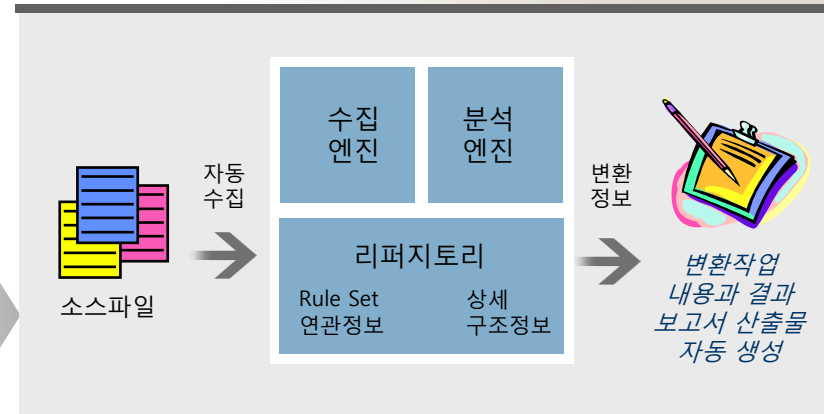
일반적인 마이그레이션 프로젝트의 분석단계에서 기존 환경의 소스 하드 코딩된 IP Address의 식별 및 변경이 필요한 경우가 빈번히 발생. 신규 IP Address로 변경이 필요한 경우 소스파일의 IP Address를 자동으로 식별하여 신규 IP Address로 변경하고 어플리케이션 소스의 IP Address의 변경 내역을 확인함으로써 어플리케이션의 유지보수성 향상을 지원합니다.

일반적인 방법



- 개발자/설계자의 수작업 확인 및 담당자별 검증
- 소스 파일, 데이터 베이스, 설계 문서 등
- 실제 시스템과 산출물 사이의 차이 발생
- 모든 개발소스 검증을 수작업으로 수행함에 따른 시간 지연 및 누락 발생 가능

앱스마트워커 솔루션 적용



- 프로그램 소스들의 IP정보를 자동으로 수집 / 식별하여 정의된 신규 IP정보로 변경 후 관련 내역을 산출물로 제공
- 자동화에 따른 신속성 및 정확성 제고
- 산출물 자동 생성을 통해 향후 시스템 개발을 위한 참고자료로 활용

기능
특징

- 자동화를 통한 신속성 및 정확성 제고
- 프로그램 소스 및 데이터베이스 관련 IP Address 정보를 자동으로 수집 / 식별하여 수작업시 발생 가능한 시간지연 및 누락 발생가능성을 사전에 차단
- 산출물 자동 생성을 통해 향후 시스템 개발을 위한 참고자료로 활용

수작업과 자동화 도구의 비교



항목	수작업	자동화 도구
마이그레이션 방법	<ul style="list-style-type: none"> 일반적인 개발방법론에 따른 대부분의 수작업 처리 	<ul style="list-style-type: none"> 개발방법론의 적용 및 자동화 도구 이용
처리방법	<ul style="list-style-type: none"> 경험에 기반한 단순 방법 처리 	<ul style="list-style-type: none"> 지식기반 Rule Set에 기반한 자동처리 방법
정확도	<ul style="list-style-type: none"> 휴먼에러의 가능성으로 정확도 미흡 	<ul style="list-style-type: none"> 100% (특이사항에 대한 정확한 보고 산출물 생성)
신뢰도	<ul style="list-style-type: none"> 작업과정의 불투명성으로 신뢰도 낮음 	<ul style="list-style-type: none"> 합리적이고 직관적인 작업진행과정에 대한 신뢰도 높음
위험도	<ul style="list-style-type: none"> 반복적인 수작업처리로 위험발생 요인 큼 	<ul style="list-style-type: none"> 매우 낮음. 자동화 방법과 툴에 의한 오류 처리
오류처리 방안	<ul style="list-style-type: none"> 모든 소스프로그램에 대한 Re-Compile방식의 처리 	<ul style="list-style-type: none"> 오류발생시 보정과 Rule Set의 보완으로 완벽하고 정확한 진행
처리속도	<ul style="list-style-type: none"> 속도가 매우 낮고 전체 공수산정의 어려움 	<ul style="list-style-type: none"> 신속하고 정확한 처리로 전체 프로젝트의 시간 단축
사후 관리	<ul style="list-style-type: none"> 오류 발생 가능성 대두 	<ul style="list-style-type: none"> 작업진행 및 결과에 대한 산출물을 통한 지속적인 관리 가능
환경 구성	<ul style="list-style-type: none"> 환경구성의 어려움 	<ul style="list-style-type: none"> 실 운영환경과의 완벽한 호환성 제공 및 동일한 작업환경 구성
History 관리	<ul style="list-style-type: none"> 개별작업별 관리의 어려움 	<ul style="list-style-type: none"> 마이그레이션 작업 전체의 체계적 관리 제공
산출문서	<ul style="list-style-type: none"> 문서작성의 어려움 	<ul style="list-style-type: none"> 변경 및 추가부분에 대한 코멘트 처리를 비롯한 마이그레이션 작업관련 산출물의 제공으로 작업의 투명성 제고와 사용자의 적극적인 참여를 통한 품질 높은 프로젝트의 진행
비용	<ul style="list-style-type: none"> 전체 공수산정의 어려움과 전문가에 대한 고비용 구조 	<ul style="list-style-type: none"> 작업 공수의 획기적인 단축으로 비용절감 극대화



HP, Solaris, AIX-Unix 에서 Linux OS 기반으로 Application 전환 사전 검증을 통한 Application Porting Guide 및 변환도구의 적용을 통해 유사 프로젝트 수행 시 Application 전환 비용 감소 효과를 가져올 수 있습니다.

1 U2L 전환 비용의 획기적 절감

- 인력기반의 마이그레이션과 비교하여 최소 2~3배 이상의 비용대비 효율성 제고

2 마이그레이션 작업의 생산성과 효율성 향상

- 이행시간 / 리스크를 최소화하고 반복된 검증작업으로 완성도 증진

3 Application 전환의 안정성과 신뢰성 보장

- In-House Application 개발환경 구축 및 기능테스트 수행

4 Unix-to-Linux 변환 가이드 및 자동화 도구 운영기술 전이

- U2L 전환시 예상되는 오류의 사전 검증 및 Application 전환 공수 절감

5 지속적인 기술적용 및 Infra 개선을 통한 경쟁력 유지

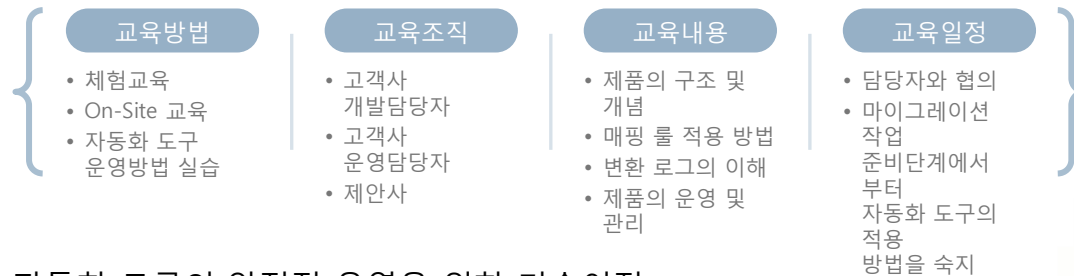
- 변환작업 전반에 대한 변경사항과 기준데이터를 분석 및 관리 가능

교육 및 기술 이전



교육 및 기술이전은 마이그레이션 프로젝트 수행 준비를 위한 초기의 교육훈련 및 문서, 기술자문등의 방법으로 자동화 도구의 운영과 다양한 마이그레이션 프로젝트의 진행중에 얻은 기술과 운영 노하우 일체를 대상으로 합니다. 본 교육과정을 통해서 프로그램 개발자 및 운영 관리자는 도입되는 자동화 도구의 특성을 이해하고 효율적으로 활용할 수 있도록 합니다.

교육훈련 개요

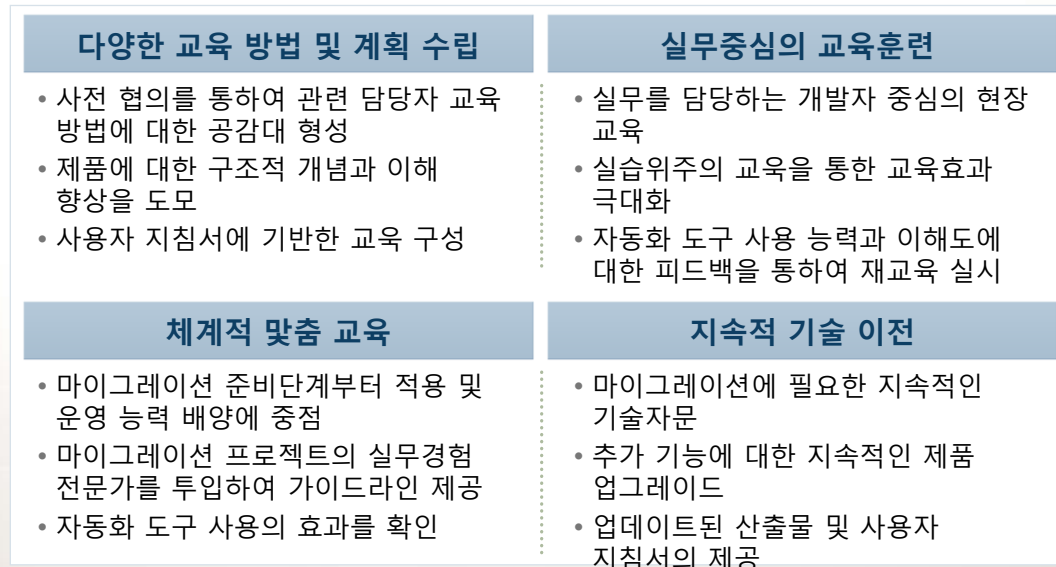


기대 효과



- ▶ U2L 프로젝트 수행 시 분석 및 계획 수립에 주도적 역할 수행
- ▶ 마이그레이션 이행 시 사전 분석 및 검증 능력 제고
- ▶ 마이그레이션 시나리오별 테스트 및 디버깅 능력 향상

자동화 도구의 안정적 운영을 위한 기술이전



- 1 마이그레이션 자동화 도구를 이용한 프로젝트 수행의 완벽한 이행을 위한 교육
- 2 마이그레이션 시 고객 담당자의 주도적인 참여와 수행을 통한 효율성 증대
- 3 U2L 마이그레이션 관련 노하우 공유
- 4 기술자문을 통한 기술이전 및 지원 활동

회사소개

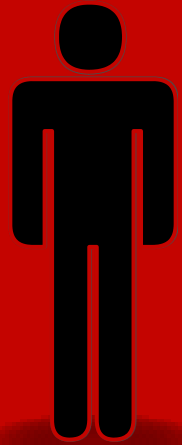


DawinICT (Dawin Information & Communication Technology)는 IT분야의 각종 시스템 진단 및 컨설팅, 플랫폼 마이그레이션, SI를 주 사업 영역으로 하고 있으며, 특히 플랫폼 마이그레이션 부분의 새로운 솔루션과 서비스의 경쟁력을 제고 시키기 위해 전력을 다하고 있습니다.



회 사 명	(주)다윈아이씨티	대표이사	김 성덕
주 소	135-860 서울시 강남구 대치동 889-5 상제리제센터 A동 1915호		
전화번호	070-4259-8117	e-Mail	sdkim@dawinict.com
종업원수	10명	설립연도	2010년 2월





Thank You

© 2013 DawinICT Corporation.
The information contained herein is subject to change without notice