
애플리케이션 배포

2001020214_16v4

1

애플리케이션 배포환경 구성하기

2

애플리케이션 소스 검증하기

3

애플리케이션 빌드하기

4

애플리케이션 배포하기

1

애플리케이션 배포환경 구성하기

1-1. 애플리케이션 배포환경 구성

1

소스코드 빌드 과정의 이해

빌드는 소스코드를 실행할 수 있는 상태로 변환하는 과정을 말하며, **프로그래밍 언어의 유형에 따라 빌드 과정이 상이**하다

◆ 컴파일언어(ex. C, C++ 등)

기계어로 바로 변환되어 실행하기 때문에 실행속도가 가장 빠르고 보안에 유리하나 소스 변경시 마다 컴파일과정을 통해 빌드 작업을 수행하기 때문에 빌드 과정이 오래 걸린다(컴파일 결과 : exe)

◆ Byte Code 언어(ex. Java, C# 등)

컴파일의 결과물이 실행파일이 아닌 'class'라는 바이트 코드 파일로 생성되고 가상환경에서 실행하는 방식으로 빌드 (컴파일 결과:class)

◆ 인터프리터 언어=함수형 언어(ex. JavaScript, Python, Ruby 등)

한 줄씩 번역되어 실행. 컴파일 과정에서 메모리 소요가 적고 빠른 시간에 컴파일을 진행 (**컴파일 결과 파일 없음**)

1-1. 애플리케이션 배포환경 구성

2

애플리케이션 배포(Release) 환경

애플리케이션 배포는 개발자 또는 사용자가 애플리케이션을 실행, 테스트할 수 있도록 **컴파일된 프로그램**, 실행에 필요한 **리소스**(이미지, 환경설정파일 등)을 서버상에 적합한 위치에 이동하는 작업을 말한다. 웹 애플리케이션의 경우, Web Server, WAS(Web Application Server)를 동작환경으로 사용한다

- ◆ Web Server

정적인 리소스를 빠르고 안정적으로 처리한다

- ◆ WAS(Web Application Server)

동적인 처리를 수행하는 프로그램 실행 부분을 배포한다

1-1. 애플리케이션 배포환경 구성

3

애플리케이션 배포 단위

- ◆ 단순히 컴파일된 실행파일또는 byte Code를 복사하는 방식
- ◆ jar(Java Archive)
- ◆ war(Web Archive)
- ◆ ear(Enterprise Archive)

1-1. 애플리케이션 배포환경 구성

4

형상관리 시스템

소프트웨어 개발 및 운영/유지•보수에 필요한 문서 관리, 변경관리, 버전관리, 배포관리 및 작업 산출물에 대한 형상관리를 포함한다

◆ 형상관리의 범위

신규프로젝트 및 보안 개발, 업무 시스템의 운영/유지보수, 전산 설비 및 시스템 소프트웨어 등과 관련된 작업, 사용자 파일 관리 등을 포함

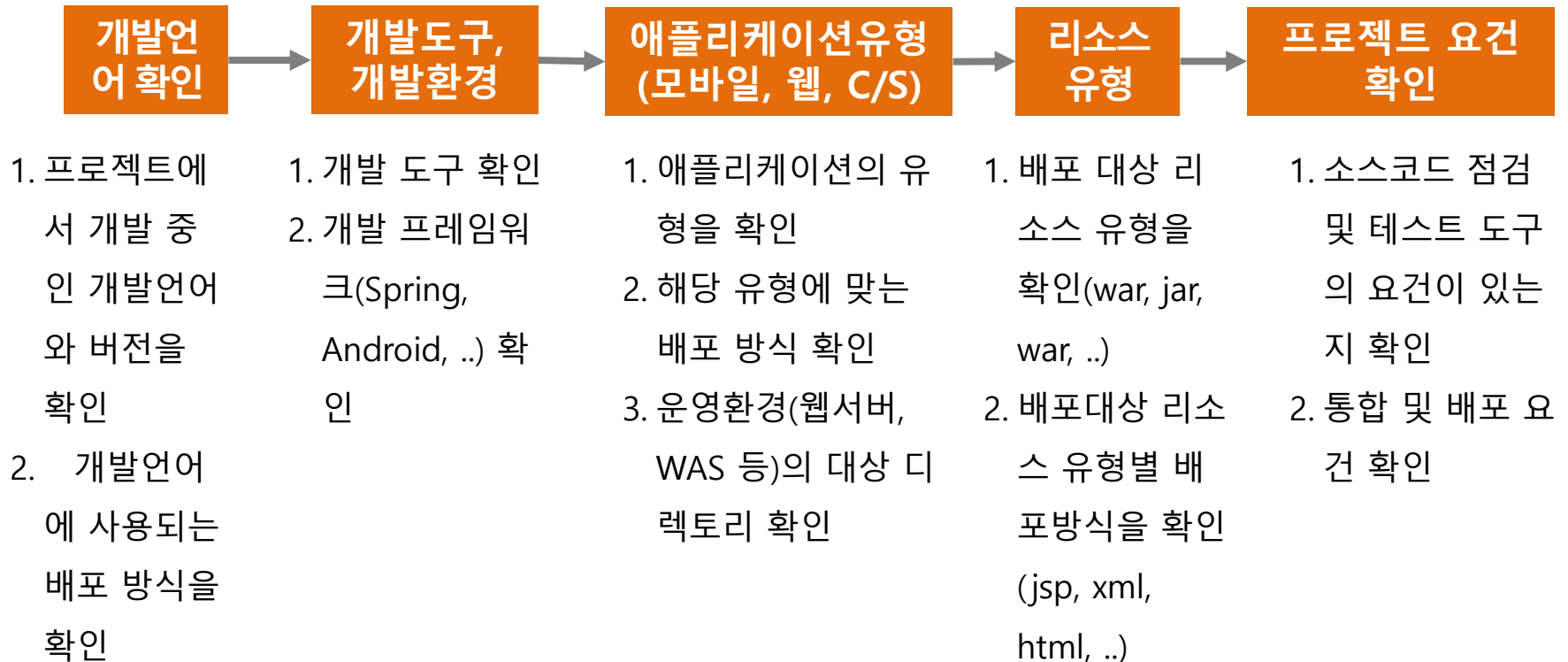
◆ 형상관리 관련 용어

- 형상관리 : SW 개발 전 주기 동안의 활동을 관리
- 형상항목 : 형상관리 대상 항목
- 기준선 : 기준이 되는 형상항목의 집합체
- 마이그레이션(Migration) : 개발 완료 시스템이 운영 단계로 전환될 때 관련 파일을 저장 공간(리포지터리)으로 이관시키는 작업
- 리포지터리 : 저장공간
- 워크플로 : 형상관리 절차
- 반출 : 형상항목을 변경하기 위해 저장공간(리포지터리)에 데이터를 전송 받는 것.
- 반입 : 저장공간에 데이터를 전송하는 것.

1-1. 애플리케이션 배포환경 구성

5

배포 환경 구성 순서



2

애플리케이션 소스 검증하기



2-1. 애플리케이션 소스 검증

1

소스코드 검증도구의 용도

(1) 정적 테스트 도구 사용 목적

정적 테스트 도구는 테스트하기 전에 코딩 오류, 성능 저하, 보안 취약점 등의 결함을 조기에 발견할 수 있도록 지원한다. 이렇게 함으로써 개발의 생산성을 향상시키고, 운영환경에서 프로그램의 품질 향상을 제고하며, 정량적인 품질 관리시스템을 구축하게 한다.

(2) 동적 테스트 도구 사용 목적

테스트 미수행 코드를 확인하고 분기(결정)문 등 특정 유형의 코드 구조가 충분히 테스트 되었는지를 확인하여 추가적인 테스트를 진행함으로써 애플리케이션의 안정성을 제고하고, 소스 품질 관리(통제) 활동을 할 수 있는 정량적인 품질 관리시스템을 구축할 수 있게 한다.

2-1. 애플리케이션 소스 검증

2 코드 인스펙션

정적 테스트의 가장 일반적인 유형

◆ 성능개선

애플리케이션의 성능에 영향을 미칠 수 있는 코드를 점검하는 Rule이다. 메모리 누수, 미사용 변수, 메소드 여부 등을 확인하여 메모리를 낭비하는 코드를 식별한다.

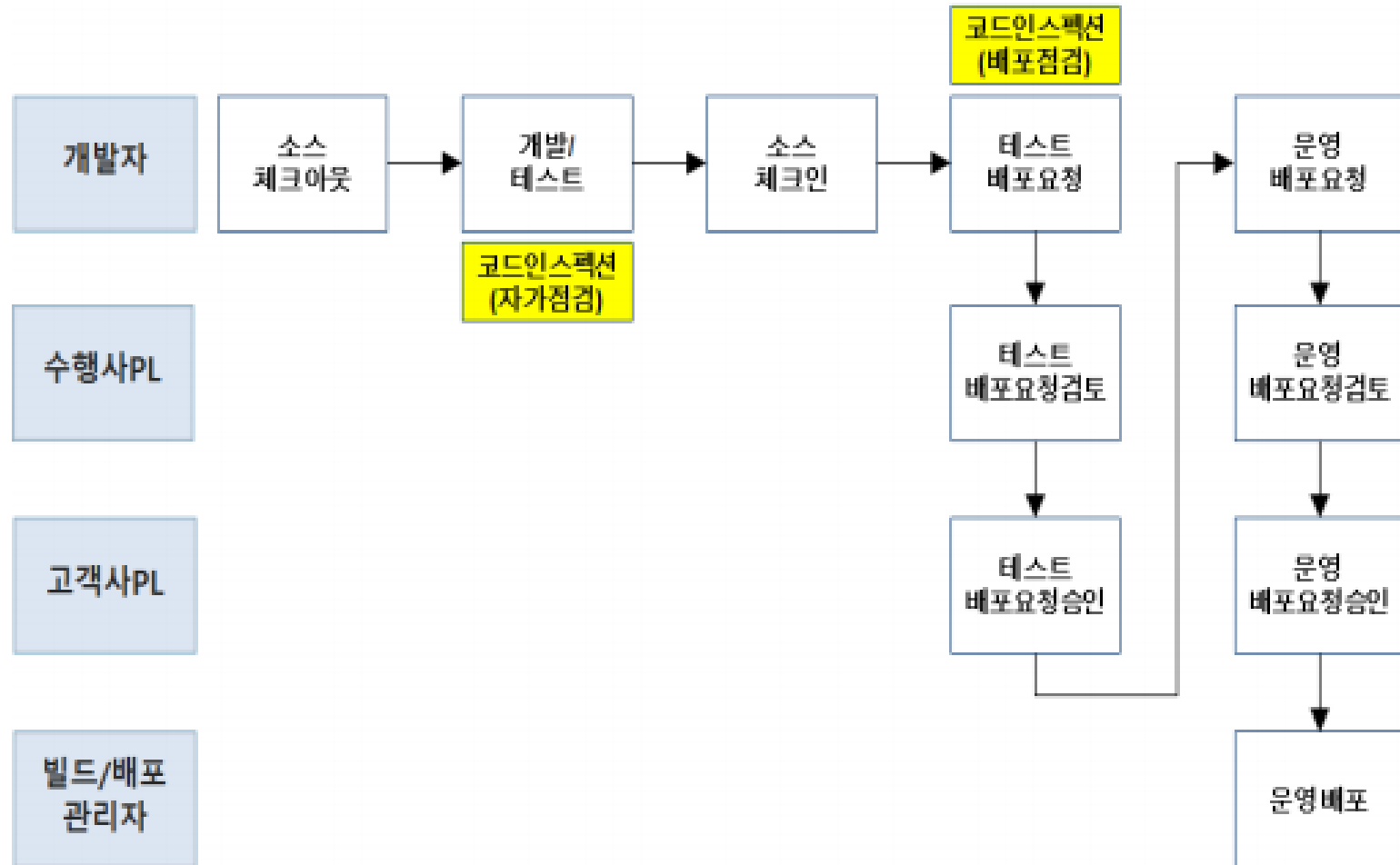
◆ 코드 작성 규칙

개발언어에서 사전에 정의된 작성 규칙 또는 프로젝트 내에서 정의된 프로그램 명명 기준 여부를 점검하는 Rule이다. 작성 규칙을 미준수한 코드 내역을 추출하여 소스 코드의 가독성을 향상시킨다. 코드 작성 규칙은 정규표현식(특정 규칙을 가진 문자열의 집합을 표현)으로 표현.

◆ 에러발생 가능성

애플리케이션 동작 중 에러 발생 가능성이 있는 코드를 점검하는 Rule이다

2-1. 애플리케이션 소스 검증



[그림 2-2] 애플리케이션 배포 절차 (예시) (코드 인스펙션 점검 단계 추가)

2-1. 애플리케이션 소스 검증

3 테스트 프레임워크(동적 분석 도구)의 구성

◆ 테스트 코드

테스트 코드 작성 및 자동화된 운영환경을 구성한다. 빌드 도구와 연계하여 빌드 수행 시

테스트 코드를 동작시켜 자동화된 테스트 환경을 제공한다. 시간과 노력 절감.

◆ 테스트 저장소

테스트 수행을 위한 테스트 코드, 테스트 데이터, 관련 테스트 스크립트, 테스트 수행 결과를 저장, 관리한다.

3

애플리케이션 빌드하기



3-1. 애플리케이션 빌드

지속적인 통합 환경

애플리케이션 개발 과정 중 지속적으로 통합, 빌드, 테스트, 배포

빌드 도구

테스트 도구

코드 인스펙션

빌드 스케줄 관리 도구

테스트 커버리지 측정 도구

4

애플리케이션 배포하기



- 애플리케이션 배포 수행 순서

- 1) 애플리케이션 실행 환경에 대한 정보를 확인한다.

- 배포된 서버 IP, 디렉토리, 접근 계정 등

- 2) 애플리케이션 배포 절차에 따라 운영 환경에 적용한다(이행 의뢰서대로 이행되었는지).

- 3) 애플리케이션 배포 후 정상적으로 작동하는지 확인한다.

- 4) 애플리케이션 배포 결과 문제가 발생했을 경우 적용 내용을 이전 상태로 복원하고 오류 원인 확인한다.

- 5) 애플리케이션 운영 환경 전환(이관) 방안을 수립하고 애플리케이션 운영 전환을 수행한다.

- 팀프로젝트 배포 수행 순서

- 1) 애플리케이션 실행 환경구성하기.

- 배포된 서버 IP확인, 웹어플리케이션 서버 설치

- 2) 애플리케이션 배포

- ojdbc6.jar가 JRE System에 있을 경우 pom.xml에 추가

```
<!-- https://mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc6 -->  
<dependency>  
  <groupId>com.oracle.database.jdbc</groupId>  
  <artifactId>ojdbc6</artifactId>  
  <version>11.2.0.4</version>  
</dependency>
```

- 생성한 war 파일을 톰캣의 webapps폴더에 넣기
- Tomcat의 bin 폴더에 startup.bat실행하여 톰캣 구동

- 애플리케이션 배포 수행 순서

3) 애플리케이션 배포 후 정상적으로 작동하는지 검사한다

소프트웨어의 검사 방법 중 하나로 소프트웨어를 내부 구조나 작동원리를 모르는 상태에서 소프트웨어의 동작을 검사하는 방법을 이르는 것으로, 주로 올바르지 않은 입력을 조합하여 올바른 출력을 판별하는 방식 : **블랙박스 검사**