# Software Architect

- 비용관리 (하)
- 일정 관리 (하)
- 요구 사항정의(하)
- 명세 수립 (중)
- 소프트웨어 설계 (상)
- 프로그래밍 (상)
- 개발 방법론 (상)
- 인프라 구축 (중)
- 테스트 방법 (중)
- 품질 관리 (하)

---

- 개발 서버 환경 구성
  - Source Version Control 선정; **GIT**, svn, mercurial, cvs,
  - 버전 컨트롤; **GitLab**, BitBucket, GitHub,
  - CI/CD; GitLab Runner, Jenkins
  - Issue Tracker, Project Management System; Redmine, Jira,
  - CMS; Confluence,
  - Deploy Repository; Nexus
  - 필요 서버 대 수, 서버 별 계정 정보 및 접속 방법(Tunneling, SSH, rlogin,...), DB 접속 방법 및 계정
  - 보안 관련 사항; VPN 접속, 보안 관련 프로그램(HiPass 등)
  - 아키텍처; 모놀리식, MSA
  - 레이어드 아키텍처,
  - Cloud/On-Premise
  - 외부 인터페이스; SMS, SNS, Mail, OpenAPI, Monitoring,
  - Docker, Kubernetes, ...
- 개발 지원
  - 프레임워크 선정; Spring Framework(+Spring Boot), Vue.js, React.js, Flask, Node.js 프레임워크, UI Framework, Editor, File Upload/Download, ...
  - 프레임워크 제공
  - 개발 툴 선정
  - 코딩 가이드
  - git 전략; 개발 및 운영 gitflow, gitlab flow, github flow
  - 공통 기능 개발 및 가이드
- 기타
  - 성능테스트
  - 아키텍처 정의서
  - 제안서
  - 시스템 구성도
  - 코드 인스펙션
  - 프레임워크 교육
  - IA(Information Architecture)
  - 소프트웨어 보안점검(모의 해킹)

# Application Architect

> 공통 로직, 개발 표준, 프레임워크 등 공통 업무 설계자

1. 프로젝트에서 사용될 기술 선택
    - ex)
2. 프로젝트에서 사용할 개발 방법론과 프레임워크 추천
3. 어플리케이션의 종합적인 설계와 구조를 제공
4. 어플리케이션 설계를 적절하게 문서화해야 함
5. 코드 작성 가이드라인을 수립
6. 인프라스트럭쳐 요구사항을 정의

- 설계시 어플리케이션 아키텍쳐 정의
- 개발 전, 프레임워크 선정 및 개발 가이드, CI/CD 환경 구축 지원
- 개발 중, 공통 코드 개발 및 지원
- 개발 후, 성능 테스트

# Technical Architect

> OS, WEB, WAS, DB 설치 등 하드웨어와 네트워크 구축 담당자

# Data Architect

> 데이터 표준, 구조, 품질, 마이그레이션 등 DB 설계자

# Business Architect or Enterprise Architect

> 기술적 관점이 아닌 비즈니스 관점의 프로세서 설계자

# Solutions Architect

> 프로젝트나 비즈니스 요구사항에 따라 개발 환경과 같은 솔루션에 대한 설계 담당. ex) DevOps 환경 컨설팅/구축 지원.

# Quality Assurance

> 산출물과 소스코드에 대한 품질을 보증하는 담당자

From:
https://theta5912.net/ - **reth**

Permanent link:
**https://theta5912.net/doku.php?id=public:computer:architect&rev=1652451050**

Last update: **2022/05/13 23:10**