

# 소프트웨어 아키텍트의 이해와 대응

정두현 / ㈜씨에스리

dhc97@naver.com

컴퓨터시스템응용기술사

정보시스템수석감리원

## 1. 소프트웨어 아키텍트의 필요성 증가

최근 인공지능, 모바일, 빅데이터의 활용요구가 증가함에 따라 솔루션을 이용하고 설치 및 실행만 하고 활용을 못하는 경우가 많이 발생하고 있다. 빅데이터는 수집된 정보를 통해 숨겨진 정보를 추출하고 이를 기반으로 의사결정을 돕는 정보를 실시간으로 제공 해야 한다. 하지만 대부분이 일반 RDBMS와 크게 다르지 않게 활용하고 있다.

인공지능의 경우에도 R&D형태로 준비하고 있으나 몇 개의 서비스 외에는 일반적인 패턴을 기반으로 한 스마트 하지 않고 단순 패턴에 따른 응답을 하는 형태로 기존 서비스 되던 패턴 분석에 따른 응답과 동일하다고 볼 수 있다.

이로 인하여 소프트웨어의 효율적인 설계를 통해 빅데이터, 인공지능 등의 본연의 목적에 맞도록 설계하고 프로세스를 구축하는 것이 매우 중요하다고 볼 수 있다.

## 2. 소프트웨어 아키텍트의 개념 및 역할

### A. 소프트웨어 아키텍트의 개념

- 인프라의 구성은 TA가 구성한다면 각 솔루션 오픈소스 미들웨어의 인터페이스와 정보의 교류 및 가치를 추구하는 청사진을 제시하는 담당자 라고 볼 수 있다.
- 대부분은 설계자라는 측면에서 다른 아키텍트와 다른 점이 없지만 소프트웨어라는 특징상 보이지는 않는 프로세스와 표준화를 주도하는 매우 중요한 파트 전문가임을 알 수 있다.

### B. 소프트웨어 아키텍트의 역할

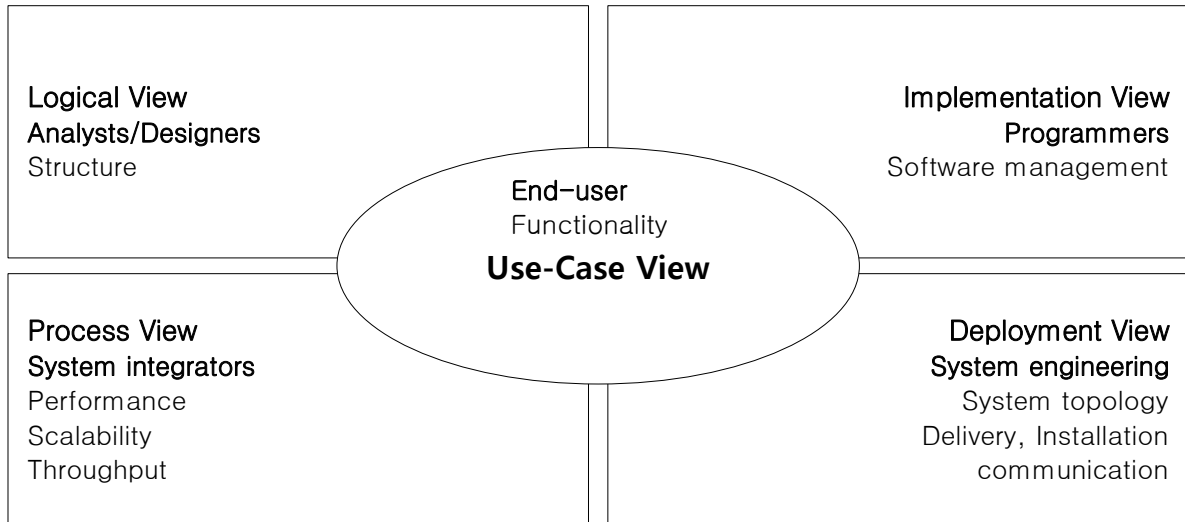
- 설계자: SW 및 솔루션간의 인터페이스 구성도를 그리고 업무에 맞게 배치
- Process: 솔루션, 서비스에 따른 작업과 데이터의 process를 설계
- 문화: 유연한 정보시스템의 활용을 위한 최적의 업무 패턴 확립
- 문화를 언급한 이유는 소프트웨어 아키텍트(SWA)는 눈에 보이지 않는 SW,솔루션등을 인터페이스로 연결 배치함으로 업무의 변화를 줄 수 있고 이것이 효율적이라면 문화적으로 지속적으로 개선하고 효율화를 통해 시간을 최소화 하고 남는시간을 통해 창의적인 발상을 할수 있는 기회를 줄 수 있는 문화적 뒷바침이 필요하다.

## 3. 소프트웨어 아키텍트의 핵심 요건

### A. 다양한 관점의 대응

- SWA는 다양한 관점측 사상을 가지고 있어야 한다. 어떤 신기술이나 설계가 최고일수는 없지만 최적으로 설계하게 된다.
- 아래 4+1 View는 아키텍트라면 누구나 중요한 이슈로 논리적,구현, 배포, 프로세스, 사용

자 측면으로 나누어 관리가 필요한 부분이다.



- 각 View별 상세내용을 확인해보면

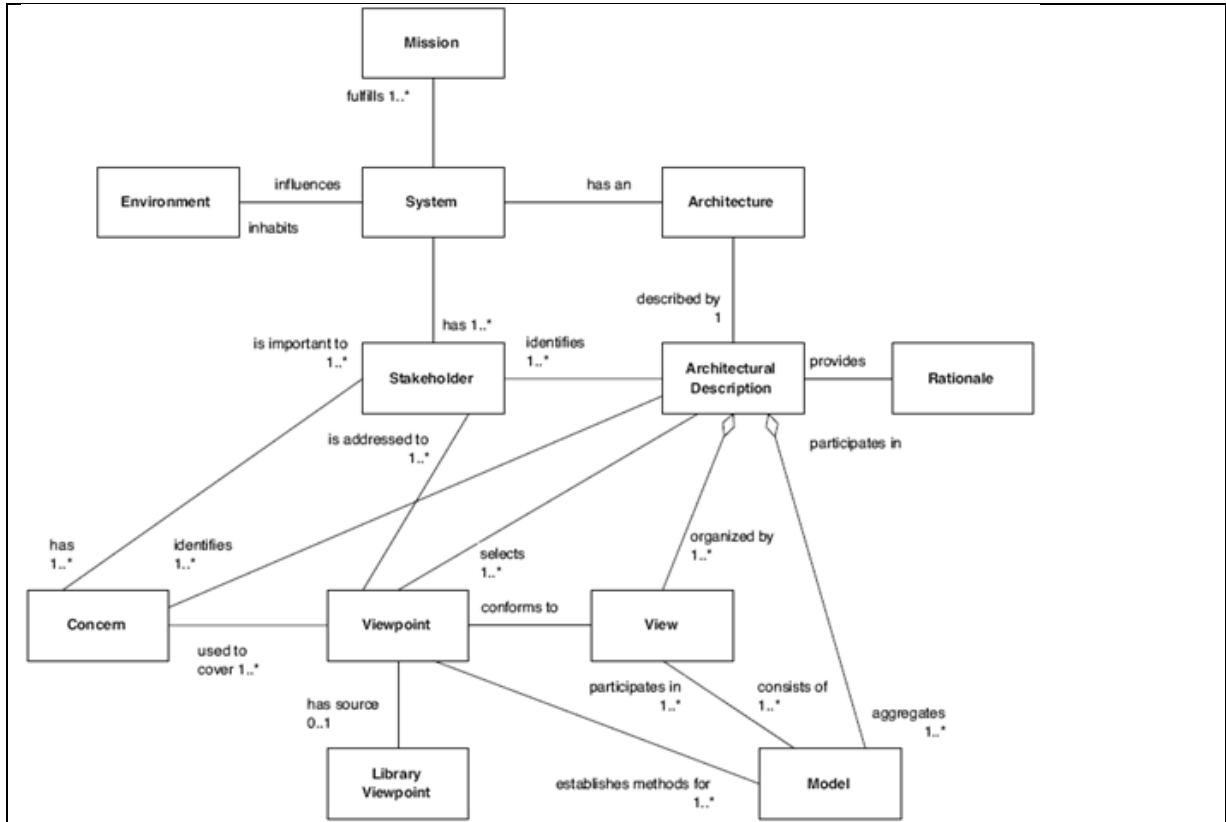
구분	내용
Usecase View	<ul style="list-style-type: none"> <li>· 사용자 관점 (End-User)</li> <li>· 요구사항 분석 통한 시스템 기능 명세</li> </ul>
Logical View	<ul style="list-style-type: none"> <li>· 설계자 관점 (분석가, 디자이너)</li> <li>· 전체 레이어의 구성과 중요한 구성요소 기술 (Structure)</li> </ul>
Process View	<ul style="list-style-type: none"> <li>· 시스템 통합자 관점 (System Integrator)</li> <li>· Performance, Scalability, Throughput 등 비기능적 요구사항 기술</li> </ul>
Implementation View	<ul style="list-style-type: none"> <li>· 구현자/개발자 관점 (Programmer)</li> <li>· 독립적으로 실행되는 컴포넌트와 이들 간의 상호관계 정의</li> </ul>
Deployment View	<ul style="list-style-type: none"> <li>· 시스템 엔지니어 관점 (System Engineer)</li> <li>· 물리적인 노드 상에 프로세스를 적절하게 분산, 배치</li> <li>· System Topolog, Delivery, Installation-Communication</li> </ul>

- 각 관점 별 요구하는 사항을 충족할 수 있도록 설계하는 것이 주 목적이다.

B. 다양한 이해관계자의 대응

- 또한 이해관계자에 따른 접근 방식의 필요성도 요구된다.

- 아래 그림은 IEEE1471로 Architecture Description이 생성되는 원리를 설명한다.



- 위의 그림에서 볼 수 있듯이 해당 미션과 환경, 이해관계자에 따라 설계가 이루어지고 이를 통해 시스템이 구성된다.
- 아키텍트의 가장 큰 요건으로는 밸런스를 맞춰야 한다. 즉 각 관점, 기능요구 비기능요구를 충족할 수 있도록 SW를 배치하고 솔루션을 선정하여 적절한 구성을 이루는 것이 중요한 요건이다.

#### 4. 최근 아키텍트의 이슈

##### A. 다양한 오픈소스

- 최근 Apache, Tomcat, MariaDB로 이전하는 사례가 증가하고 있다, 이에 따라 아키텍처의 변화가 매우 중요하게 자리잡고 있다.
- 구축 외에도 자동 배포 형상관리등의 오픈소스에 대한 관리 포인트가 증가되고 있고 이를 통한 설치 및 구성 교육에 대한 요구가 증가하고 있다.
- 오픈소스로의 이관에 대하여 대부분이 기능을 활용하는 측면과 비용절감이라는 측면에 집중되어 이들간의 연계성과 효율성 측면에 대한 이해가 부족하여 잘못된 요구사항을 제시하는 경우가 많이 있다.
- 오픈소스에 대한 정확한 이해와 POC, Pilot Test를 수행하여 장/단점을 파악하고 효과성 있는 연계 포인트를 찾는 것이 매우 중요하다.
- 이러한 오픈소스의 활용측면을 상세하게 설명하기 위하여 DevOps를 통해 설명하고자 한다.

##### B. DevOps의 도래

- 최근 DevOps에 대한 관심이 증가하고 있다. 이는 SI와 SM의 한계성을 인식하고 이를 해

결하기 위한 방안으로 제시되고 있다.

- 하지만 대부분이 도구나 솔루션을 사용할 뿐 내부적 사상과 업무의 변화에 대한 적용이 부족한 현실이다.
- 이에 따라 SWA측면으로 DevOps의 효율적인 적용을 위해 앞으로 제시하려고 한다.

## 5. 향후 과제

- 앞으로 약 10회에 걸쳐 SWA측면의 DevOps의 성공적인 도입방안을 제시하고자한다. DevOps는 그저 SM과 SI를 합친 조직이 아닌 오픈소스 솔루션과 개발문화 IT문화의 변화라는 측면으로 우선 도입이 쉬운 방식부터 설명하려고 한다.
- DevOps는 우선 Agile이라는 기본 베이스를 가지고 진행해야 한다.
- 주로 DevOps를 설명하겠지만 Agile과 연계된 도구를 제시할 예정이다.
- 앞으로의 기고를 통해 기업내의 조직내의 DevOps를 도입하여 효율적인 프로젝트 개발 및 운영 문화의 확립이 되었으면 합니다.

다음 회차 부터 DevOps의 SWA측면의 구축 방안을 점진적으로 진행할 예정입니다.

궁금한 사항은 [dhc97@naver.com](mailto:dhc97@naver.com)으로 메일 주세요