# 소프트웨어 아키텍처 설계

김정호 대표이사



### 강사 소개



• 이름 : 김정호

- 소속:(前) SKC&C 소프트웨어 아키텍트
   (前) 삼성전자 분석/설계 전문가 과정 강사 및 멘토
   (現) 삼성테크윈 아키텍트 양성 교육 강사 및 멘토
   (現) NIPA 소프트웨어 전문위원 (분석/설계 담당)
   (現) 제이유 컨설팅 대표이사
- 학력 : Carnegie Mellon University Master Software Engineering 전공 KAIST Ph.D 수료 - Software Architecture 전공
- 경력: 한화손해보험 차세대 시스템 소프트웨어 아키텍트 대신증권 차세대 시스템 - 소프트웨어 아키텍트 3D 가상세계 서비스 플랫폼 - 소프트웨어 아키텍트 네비게이션 플랫폼 - 아키텍처 컨설팅
- 저서: 소프트웨어 아키텍처 이론과 실제 (에이콘 2007, 번역)
   소프트웨어 아키텍처 문서화 (에이콘 2008, 번역)
   Software architecture reconstruction for extensibility (KCSE 2009, 논문)
   The method for verifying software architecture with FSP model (ICACT 2010, 논문)
   엔터프라이즈 애플리케이션을 위한 소프트웨어 아키텍처 개발 실무 (KOSTA, 2010, 저술)

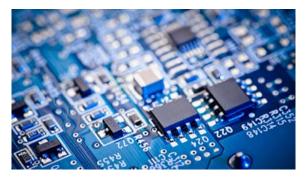
# 소프트웨어 품질

### 소프트웨어 개념

- ❖ Software VS. Program
  - Executable files(Source code)
  - Data
  - Documents?
- Software VS. Hardware
  - System
  - Which is more affected?
    - Software intensive system
    - Hardware intensive system

Software = Program + B





# 소프트웨어 특징 -I

- ❖ 보이지 않고 만져지지도 않는...
- ❖ 개발,관리의 어려움



## 소프트웨어 특징 - II

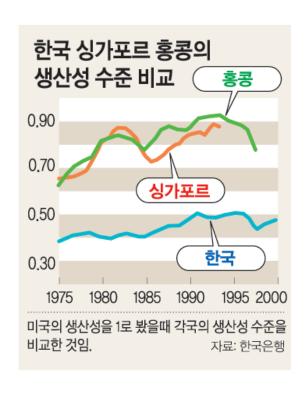
- ❖ 개발자에 따라 지대한 영향
- ❖ 설계와 구현의 모호함





### 소프트웨어 기술경쟁력

- ❖생산성과 품질
  - 인도는 왜 소프트웨어 기술경쟁력을 가지나?
  - 미국은 소프트웨어 기술경쟁력의 선진국인가?





# 소프트웨어 모델

- ❖모델?
- ❖모델링은 왜 하는가?

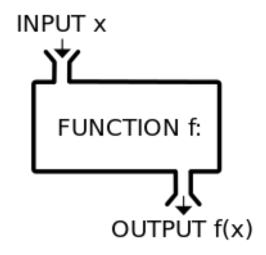


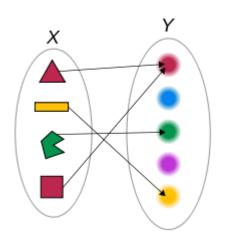


## 기능과 품질

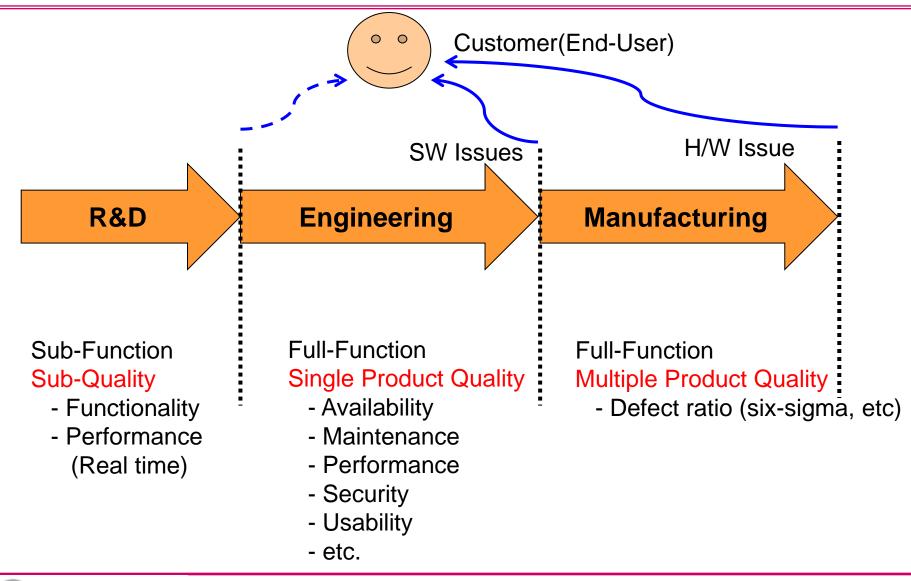
- ❖ 기능은 동사, 품질은 부사
- ❖뭐가더 중요?

- ❖ 소프트웨어 품질
  - Availability, Reliability
  - Performance, Efficiency
  - Maintainability, Modifiability
  - Security
  - Usability





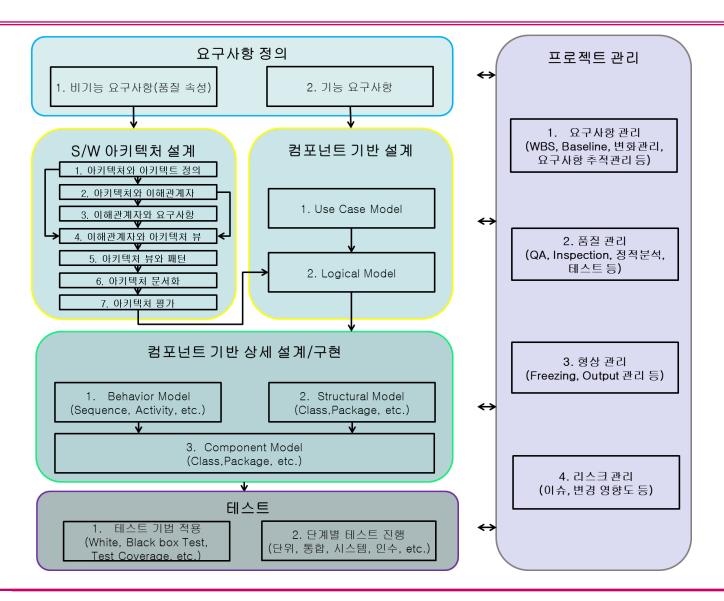
### 소프트웨어 품질



# 소프트웨어 개발 생명주기(SLDC)

$$f_1(x) = x - 1 = 0$$
  
sol)  $x = 1$ 

$$f_2(x) = x^2-2x-3 = 0$$
->  $(x-3)(x+1) = 0$ 
->  $x-3=0$  or  $x+1=0$ 
sol)  $x=3$  or  $x=-1$ 



### 소프트웨어 공학 적용 범위

- ❖ 소프트웨어 크기
- ❖ 소프트웨어 상용화

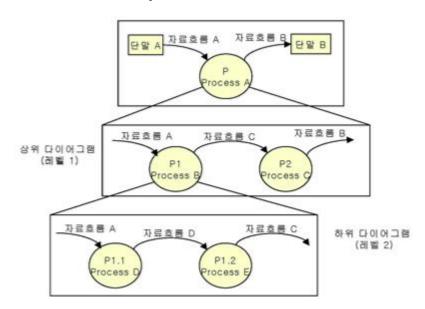


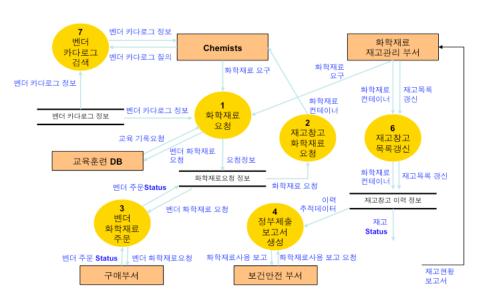


# 소프트웨어 개발방법론

### 구조적 방법론 모델링

- Functional Decomposition
  - Resolving a functional relationship into its constituent parts
- Data Flow Diagram
  - An intuitive way of showing how data is processed by a system
  - Examples





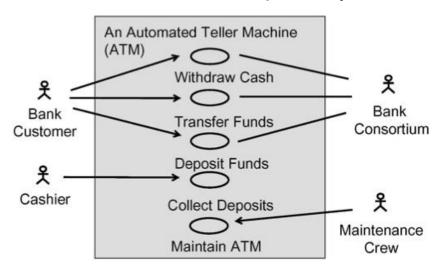
### 객체 지향 방법론 모델링

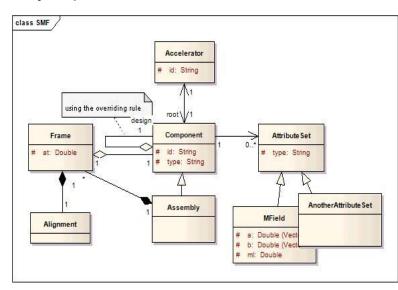
#### Use Case

- A scenario-based technique for requirements elicitation.
- Identify the type of interaction and the actors involved.

### UML Model

 Types: class (package), object, use case, sequence, collaboration, statechart, activity, component, and deployment





Consulting Co.,LTD

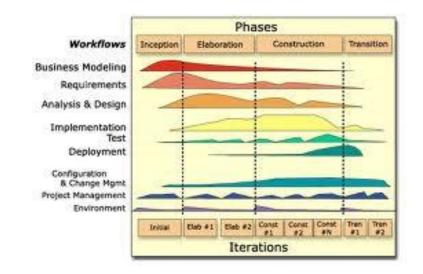
### **UP(Unified Process)**

#### ❖ RUP

- Use Case Driven
- Architecture Centric
- Iterative and incremental development

#### **❖** UML

- UML Notation
- UML Diagram
- RUP VS. UML
  - RUP is a description of a software development process
  - UML is a modeling notation
  - RUP can use UML, but the reverse is not true

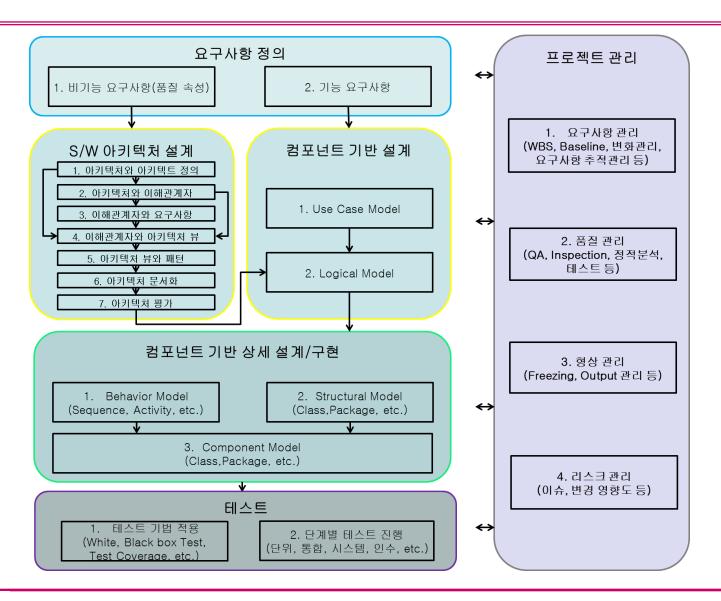




# 소프트웨어 개발 생명주기(SLDC)

$$f_1(x) = x - 1 = 0$$
  
sol)  $x = 1$ 

$$f_2(x) = x^2 - 2x - 3 = 0$$
  
->  $(x-3)(x+1) = 0$   
->  $x-3=0$  or  $x+1=0$   
sol)  $x=3$  or  $x=-1$ 



## Agile

#### **❖** XP

- Artifacts
  - User stories, CRC cards, prototype...
- 13 practices
  - Metaphor, whole team, planning game, small releases, customer tests, simple design, pair programming, test-driven development, design improvement, continuous integration, collective code ownership, coding standard, sustainable pace

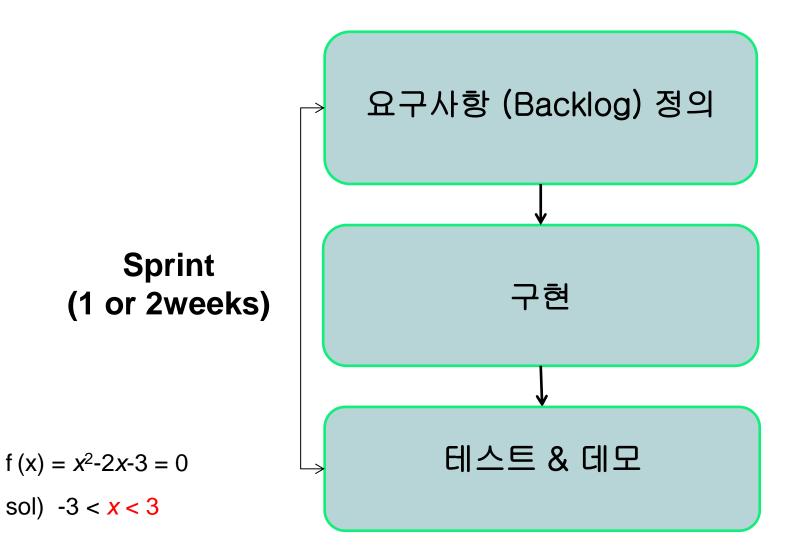
www.abacom.it

### Scrum

- Management approach using agile methodology
  - Daily Meetings, Sprint
  - Small Release
  - Back Log, Burn down chart



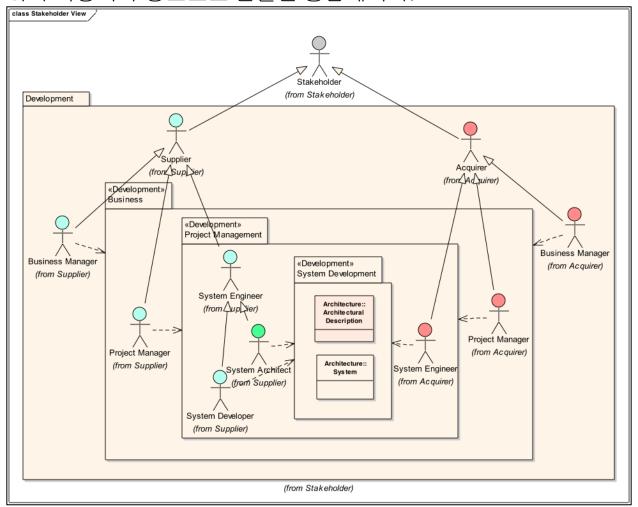
# 소프트웨어 개발 생명주기(SLDC)



# 소프트웨어 아키텍처 설계

# 시스템과 아키텍처

❖ 시스템은 하나 이상의 구성요소로 연결된 통합체이다.



\* Created by Sungwon Choi, Myoungji Univ.



### 기능과 품질 요구사항

- System
- Stakeholders
  - Concern(Needs)
  - Viewpoint
  - View
- Requirement
  - Functional
  - Non-Functional

<sup>\*</sup> Created by Sungwon Choi, Myoungji Univ.



Requirements & Architecture Requirements «Aspect» Functional Non-Functional gross-cuts Requirements Requirements Serv ice Business Quality Functional Entity «Constraint» Component Quality constrains Quality Engineering Obj ect Quality Structure Instance Structure Type Architecture Sys te m Impleme ntation

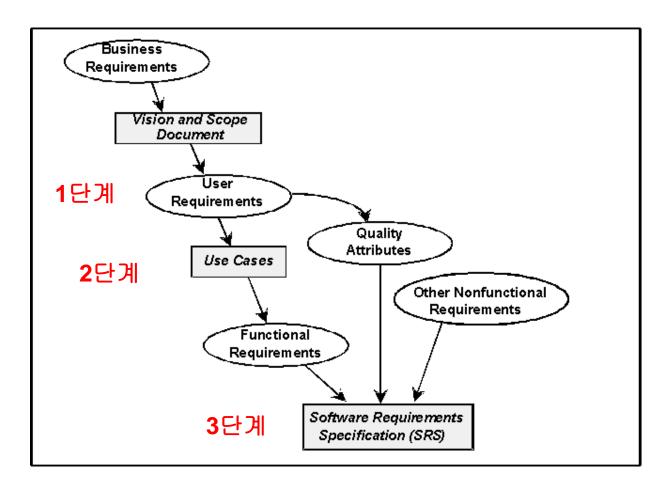
### 기능 요구사항 - 개요

### ❖요구사항 모델링 기법

- 구조적 분석(Structured Analysis)
  - Data Flow Diagram(DFD)
  - Entity Relation Diagram(ERD)
  - State Transition Diagram (STD)
- 유즈케이스 분석(Use Case Analysis)
  - Use Case Modeling (UC)
- 목표와 시나리오 기반 분석(Goal and Scenario Based Analysis)
  - Goal-Scenario Modeling(GS)

# 기능 요구사항 - 개요

## ❖소프트웨어 요구사항의 3단계



### 기능 요구사항 - 유스케이스

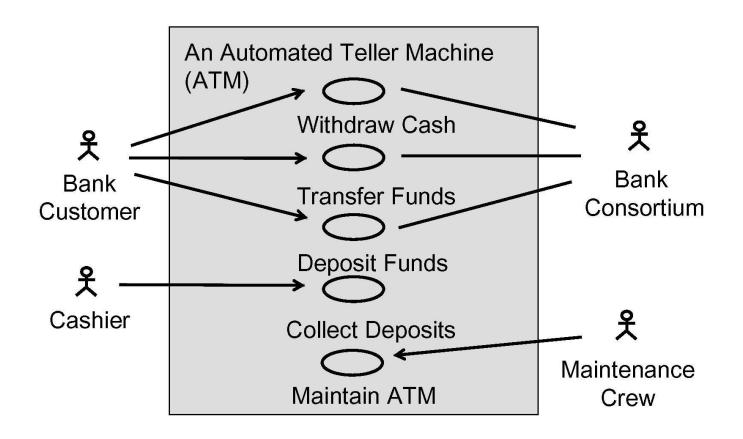
### ❖유스케이스 분석 기법

- 사용자 중심 요구사항 모델링
  - 유즈케이스와 액터 간 상호작용을 통해 시스템 분석
  - 사용자 비즈니스와 애플리케이션 영역을 이해
  - 작업 중심적 : Diagram 으로 요구사항 이해
- 유즈케이스는 각각 독립적
  - High Priority Use Case = High Priority Requirements
  - 단순한 유즈케이스 나열:전체적인 시스템 요구사항 이해 어려움
  - 비기능적인 요구사항은 표현이 어려움
- 표준 Graphic 표기
  - Communication 개선, 해석의 Risk 감소



## 기능 요구사항 - 유스케이스 예제

### ❖ATM 시스템 예제

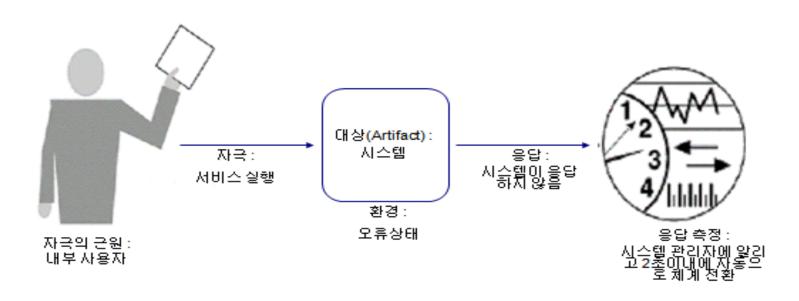


### 비기능 요구사항 - 개요

- ❖ <u>이해관계자</u>를 통한 비기능 요구사항 추출
  - 유스케이스 다이어그램에서 액터에서 시스템이 아닌 사람은 모두 이해관계자이다.
  - 이해관계자 중 액터가 아닌 사람도 있다.
    - 팀장, 기획자, 고객(사용자가 아닌), 유지보수자, 내부 개발자 등
  - 이해관계자의 이해를 중심으로 비기능 요구사항을 추출
    - 각각의 이해관계자를 중심으로 관심사항을 추출하여 비기능 요구사항 추출
- ❖ <u>프로젝트 환경</u>을 통한 비기능 요구사항 추출
  - 프로젝트 기간, 비용에 대한 이해후 이슈를 추출한다.
  - 프로젝트 조직(개발 인력의 부족, SQA 분리 등)에 대한 이해 후 이슈를 추출한다.
- ❖ 주의 사항
  - '<u>무엇을(What)</u>' 쓰는 것이지 '어떻게(How)' 를 쓰는 것이 아니다.
  - <u>시나리오 기반</u>으로 작성(상황을 명확히 표현)하는 것이 이해하기가 좋다.
  - 모호한 표현(Ex.<u>잘, 쉬운, 의도되지 않은, 유사 시 등</u>)을 쓰지 않는다.



## 품질 요구사항 - 작성 예제



❖ 내부 사용자가 시스템에 서비스 실행을 요구할 때, 시스템이 3초 이내에 응답하지 않으면, 재 요청을 3회 실시한다. 이 때에도 3초 이내에 응답하지 않으면 오류상태로 인식하고 시스템 관리자에 SMS으로 알리며, 2초 이내에 자동으로 응급 시스템으로 전환한다.

# 비기능 요구사항 - 작성 예제

품질₽	요구사항명	요구사항설명관	난이도	중요도
보안₽	시스템 <u>보안성</u> 확보~	군은 보안이 가장 중요하기 때문에 일반적인 외부인의 접속을 차단하는 기능 외에 전문의 본문, DB연동 데이터, 첨부파일 들의 암호화까지 고려되어야 한다?	상	상↵
성능	<mark>시</mark> 스템 성능 확보 <sup>⊋</sup>	중앙서버는 사용자가 많아 전문 수신량이 실시간에 처리할 수 있는 최대 건수인 시간당 10,000건을 초과할 경우 오류가 발생하는 것을 막기 위해 Queue에 저장해서 들어오는 순서대로 처리하고 예외적으로 우선순위가 높은 전문은 들어오는 순서에 관계없이 먼저 처리한다~	<mark>상</mark>	상₽
가용성₽	시스템 안정성 확보↔	중앙서버는 시스템이 <u>의도되지 않은 상황</u> 에서는 다운되지 말아야 한다₽	중	상₽
안정성₽	백업 기능지원↔	지역서버, 중앙서버 관리자가 데이터를 매주 일요일 새벽 0시에 백업 받을 수 있는 기능을 지원하고, 관련 로그(전문목록, 전문정보)를 남겨야 한다?	ĸ <mark>o</mark>	중위
신뢰성√ (유지보수)∉	복구 기능지원↔	<u>유사시</u> 백업 받은 데이터를 이용하여 문제가 발견된 시점을 기준으로 4시간 이내에 복구가 가능하여야 한다√		중
사용성₽	사용자 편리성₽	본문 작성 시 Text 외에 표, 그림 등의 객체를 삽입할 수 있어야 한다?	하	중리



### 비기능 요구사항 - 개요

- ❖ <u>이해관계자</u>를 통한 비기능 요구사항 추출
  - 유스케이스 다이어그램에서 액터에서 시스템이 아닌 사람은 모두 이해관계자이다.
  - 이해관계자 중 액터가 아닌 사람도 있다.
    - 팀장, 기획자, 고객(사용자가 아닌), 유지보수자, 내부 개발자 등
  - 이해관계자의 이해를 중심으로 비기능 요구사항을 추출
    - 각각의 이해관계자를 중심으로 관심사항을 추출하여 비기능 요구사항 추출
- ❖ <u>프로젝트 환경</u>을 통한 비기능 요구사항 추출
  - 프로젝트 기간, 비용에 대한 이해후 이슈를 추출한다.
  - 프로젝트 조직(개발 인력의 부족, SQA 분리 등)에 대한 이해 후 이슈를 추출한다.
- ❖ 주의 사항
  - '<u>무엇을(What)</u>' 쓰는 것이지 '어떻게(How)' 를 쓰는 것이 아니다.
  - <u>시나리오 기반</u>으로 작성(상황을 명확히 표현)하는 것이 이해하기가 좋다.
  - 모호한 표현(Ex.<u>잘, 쉬운, 의도되지 않은, 유사 시 등</u>)을 쓰지 않는다.



### 아키텍처 전략 - 패턴

- 1. 소프트웨어 아키텍처 패턴
- 1) 다양한 상황에서 소프트웨어 아키텍처 수립하는 방식을 정형화한 것을 소프트웨어 아키텍처 패턴이라고 한다.
- 2) 가장 일반적인 아키텍처 패턴은 아래 표와 같다.
- 3) 아키텍처 패턴은 지속적으로 생성되고 있으며 POSA(Pattern Oriented Software Architecture) 시리즈(1 ~ 5권)에 자세한 내용이 나와있다.

패턴 명	설명	아키텍처 예제
Layer 패턴	가장 일반적으로 사용하는 아키텍처 패턴으로서 subtask 들을 그룹으로 묶어 사용 허가 관계를 표시하는 패턴이다. 모듈의 재사용성을 높여 유지보수성이나이식성에 좋은 패턴이다.	Component, 3.1 Layer 3 Component, 3.2 Component, 3.3  Component, 3.1 Layer 2 Component, 2.3 uses Component, 2.3  Component, 1.1 Layer 1 Component, 1.2 Component, 1.3
Broker 패턴	외부에 분산된 컴포넌트를 호출하려고 할 때 클라이언 트 request의 분석하여 서버 컴포넌트에 전달하고 그 결과값을 전달하는 역할을 하는 패턴을 브로커 패턴이 라고 한다. 보안이나, 안정성을 높일 수 있는 패턴이다.	Client-side Proxy pack_data unspeck_data uns
MVC 패턴	모델, 뷰, 컨트롤 세개의 컴포넌트로 어플리케이션을 구분한 패턴으로 사용자 인터페이스를 가지고 있는 많은 어플리케이션에 사용된다. 모델은 기능과 데이터를 가지고 있고 뷰는 사용자의 화면 표시를 지원한다, 컨트롤러는 이들 과의 관계를 가지고 사용자 이벤트나 모델의 변화를 감지하여 모델과 뷰에 전달하는 역할을 한다. 뷰와 모델 사의 사이의 일관성을 갖게하여 변경용이성, 기능확장성을 지원한다.	Oraphical Display (Encapsulation of Deplay Choices)  Model (Encapsulation of Deplay Choices)  User- Interface (Encapsulation of Internation)  Events (Encapsulation of Internation)

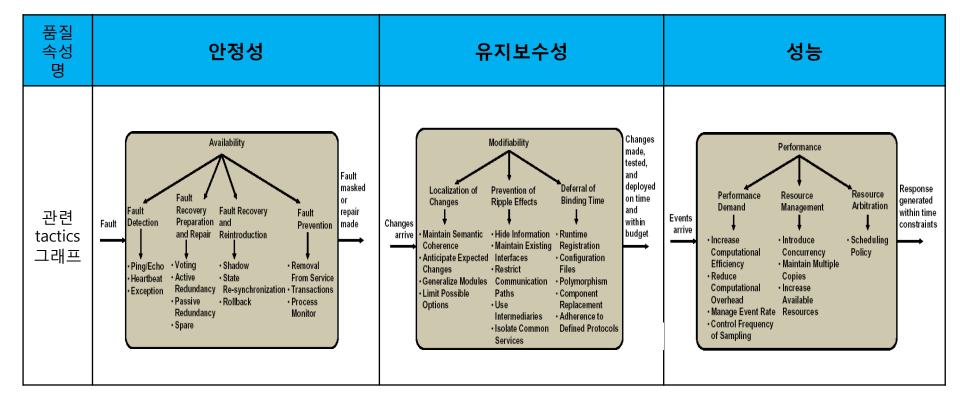
### 아키텍처 전략 – 스타일

- 1. 소프트웨어 아키텍처 스타일
- 1) 소프트웨어 아키텍처 수립하는 수단으로 아키텍처 스타일을 사용하기도 한다. 아키텍처 스타일은 아키텍처 패턴 중에 Runtime view, Module view, Allocation view에 자주 사용되는 패턴을 표준화 한 것이다.
- 2) 각종 뷰를 작성할 때 참조할 수 있는 아키텍처 패턴으로서 Call & Return style, Layer style, Uses style, Decomposition style이 많이 쓰이고 있다. (첨부 1 참조, 소프트웨어 아키텍처 문서화 책 참조)

뷰	아키텍처 스타일	설명
	Data Flow	데이터 이동에 목적을 둔 Style (Ex. Batch sequential, pipes & filters)
	Call & Return	S/W 요소들의 호출을 특성화 한 Style(객체지향에서 사용하는 스타일)
Runtime view	Interacting process	Event Based로 system이 움직이는 Style (Ex. Publish-Subscriber system)
	Data-Sharing	Data Storage를 중심으로 연동하는 Style
	Decomposition	모듈을 단계별로 분리하여 모듈을 파악하는 스타일(UML의 aggregation style)
Module view	Generalization	부모, 자식 모듈과의 관계(UML의 inheritance)
	Uses (Layer style)	모듈 간의 사용성을 표현한 스타일(UML의 association 관계) (Layer style은 Layer 패턴과 같음)
	Decomposition	Software 요소와 infra system(H/W, N/W)과의 관계를 나타낸 스타일
Allocation view	Implementation	소프트웨어 모듈과 파일 시스템(디렉토리)과의 관계를 나타낸 스타일
VICW	Work assignment	소프트웨 모듈을 개발하는 사람, 그룹가 개발할 모듈과의 관계를 나타낸 스타일

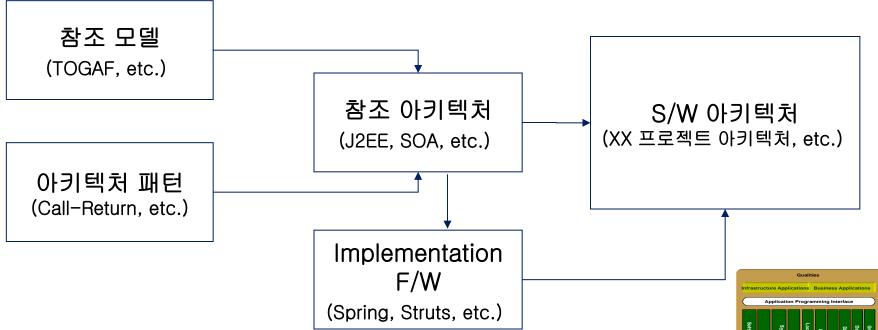
### 아키텍처 전략 – 전술

- 1. 소프트웨어 아키텍처 수립 전술(tactics)
- 1) 소프트웨어 아키텍처에 요구되는 품질 속성을 만족시키는 특정 기법의 모음을 tactics라고 한다.
- 2) 아키텍처 수립 전술은 Computer science 적인 기법과 소프트웨어 구조로 문제가 포함되어 있다. (패턴, 스타일과의 차이점)
- 3) 주로 소프트웨어 아키텍처를 수립하고 부족한 품질 속성을 높이는데 사용된다.
- 4) 보다 자세한 사항은 '아키텍처 이론과 실제' 4장에 나와있다.





### 아키텍처 전략 - 참조 아키텍처



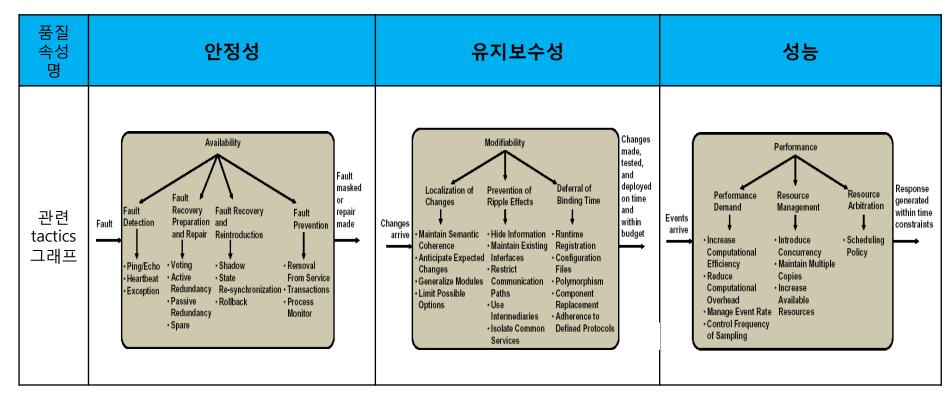
- ❖ 참조모델
  - 특정 도메인의 기능을 요소 별로 구분하여 놓은 것
- ❖ 참조 아키텍처
  - 참조 모델과 아키텍처 패턴을 결합하여 특정 품질을 항상 시켜 놓은 아키텍처
- Implementation F/W
  - 참조 아키텍처를 기준으로 프레임워크으로 개발 된 상용 제품



**TOGAF TRM** 

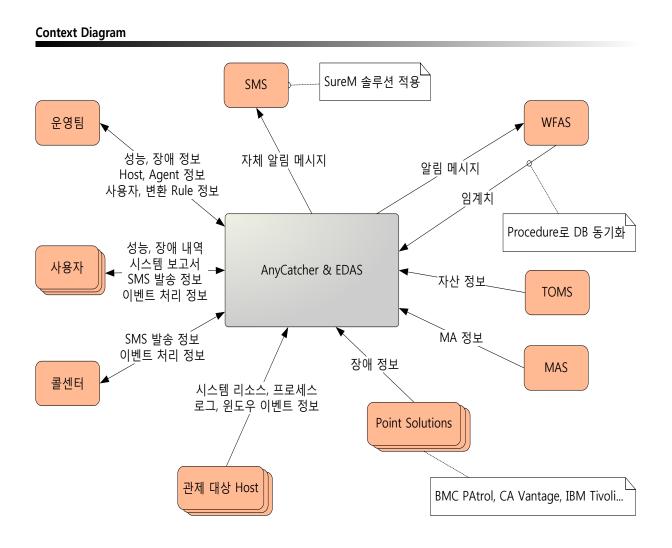
### 아키텍처 전략 – 전술

- 1. 소프트웨어 아키텍처 수립 전술(tactics)
- 1) 소프트웨어 아키텍처에 요구되는 품질 속성을 만족시키는 특정 기법의 모음을 tactics라고 한다.
- 2) 아키텍처 수립 전술은 Computer science 적인 기법과 소프트웨어 구조로 문제가 포함되어 있다. (패턴, 스타일과의 차이점)
- 3) 주로 소프트웨어 아키텍처를 수립하고 부족한 품질 속성을 높이는데 사용된다.
- 4) 보다 자세한 사항은 '아키텍처 이론과 실제' 4장에 나와있다.





# 아키텍처 뷰 - 작성 예제



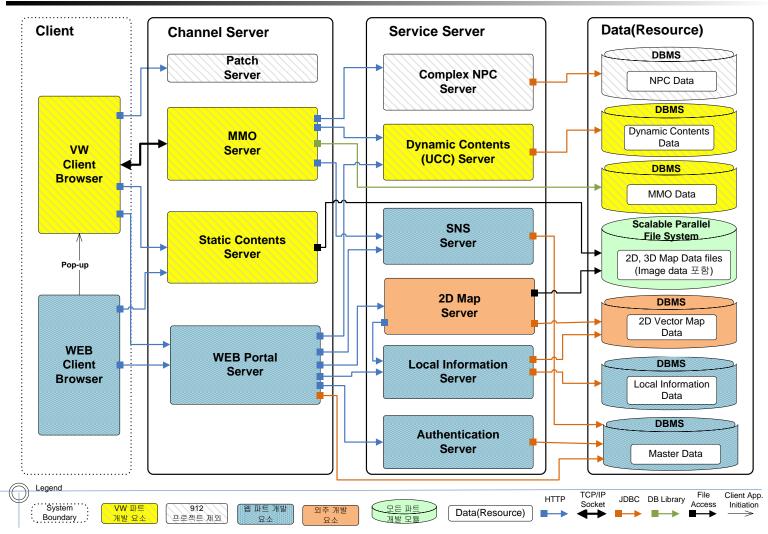


	Module Views				C&C Views		ation ws
Stakeholder	Decomposition	Uses	Class	Layer	Various	Deployment	Implementation
Project Manager	s	s		s		d	
Member of Development Team	d	d	d	d	d	s	s
Testers and Integrators		d	d		s	s	s
Maintainers	d	d	d	d	d	s	s
Product Line Application Builder		d	s	0	s	s	s
Customer					s	0	
End User					s	s	
Analyst	d	d	s	d	s	d	
Infrastructure Support	s	s		s		s	d
New Stakeholder	х	х	x	х	х	х	х
Current and Future Architect	d	d	d	d	d	d	s

Key: d = detailed information, s = some details, o = overview information, x = anything Source: Adapted from [Clements 03].

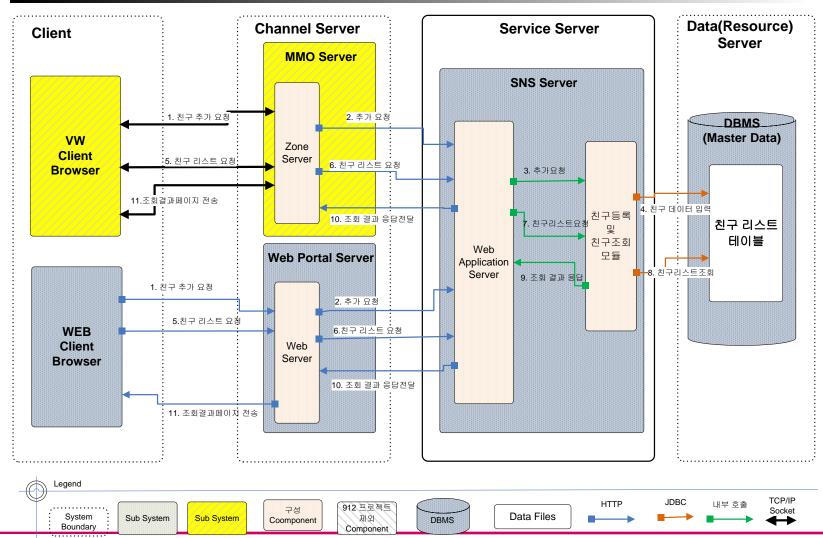


#### Runtime view – High Level Software Architecture

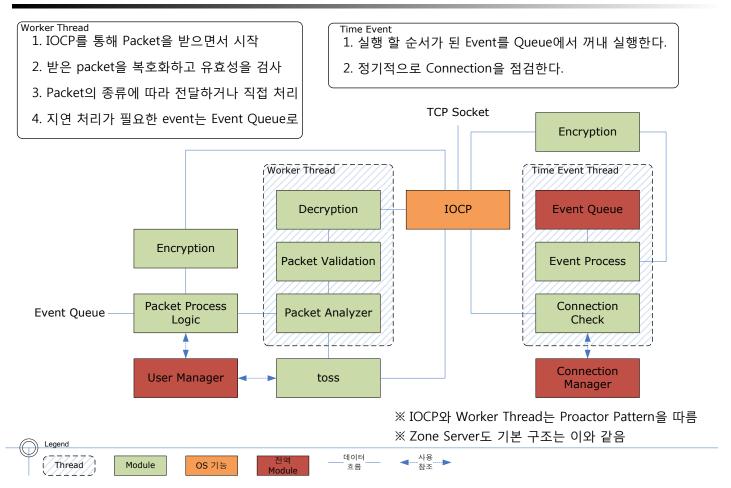




#### Runtime view - SNS 친구 추가, 리스트



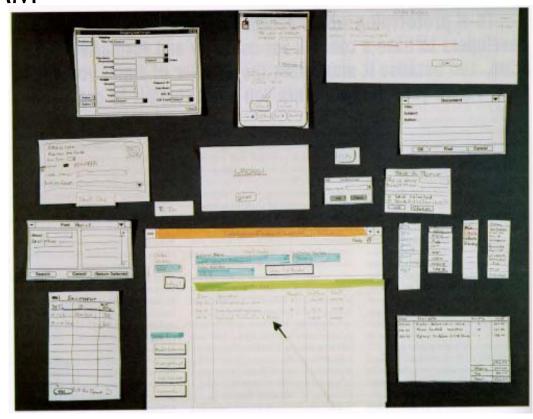
#### Front End의 데이터 흐름





### 아키텍처 검증

- Inspection
  - Peer-review
  - 아키텍처 평가 ATAM
- ❖ Formal Methods
  - Petri-net
  - FSP
- Prototype
  - Paper
  - Core
- Simulation



### 아키텍처 평가 – 개요

#### ❖ 아키텍처 평가가 필요한 이유

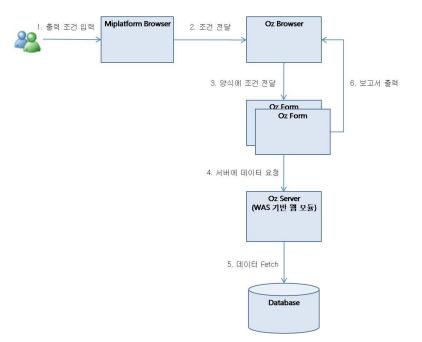
- 아키텍처 뷰가 요구사항(Architectural Drivers)를 만족하는지 바로 확 인해야 한다.
  - 개발 중이나 이후에 아키텍처 문제가 발견되면 큰 문제가 될 수 있음 눈덩이 효과
- 아키텍처는 이해관계자의 이해를 위한 대화 도구로 사용되는 것이 일 반적
  - 아키텍처에 문제가 있는지 확인할 방법이 없음
  - 개발자들의 신빙성이 적을 수 있다.

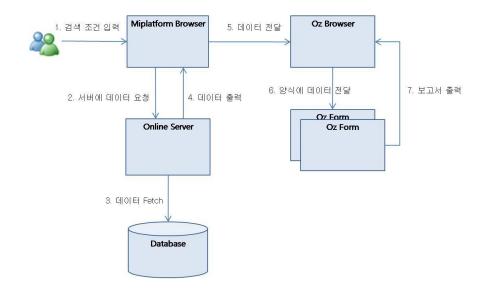
#### ❖ 아키텍처 평가 기법

- 시나리오 기반의 평가 기법
  - SAAM, ATAM etc.
- 프로토타입 평가 기법
  - 아키텍처의 일부 요소를 직접 구형하여 검증하는 방법
- 그외의평가기법
  - Simulation, Formal Methods, etc.



### 아키텍처 평가 - 작성 예제





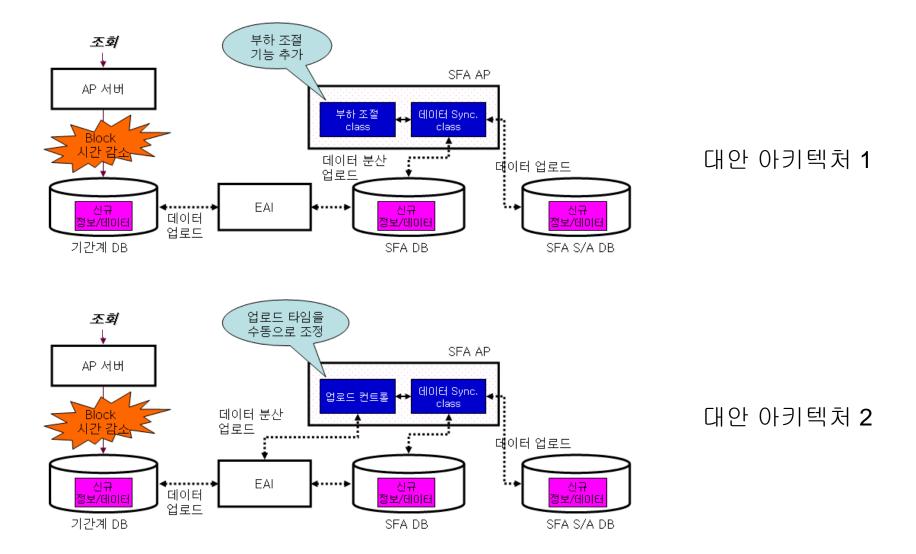
대안 아키텍처 1

대안 아키텍처 2

구분	장점	단점	적용
대안 1	마이플랫폼을 통해 데이터를 조회하지 않아도	마이플랫폼을 통해 조회를 이미 한 경우는 중	병행 적용
	보고서 츨력이 가능함	복해서 시스템에 부하를 주게됨	
	별도의 보고서 서버를 활용하므로 온라인 서버		
	에 부하를 주지 않음		
대안 2	이미 fetch한 데이터를 재사용하므로 보고서	Fetch한 데이터만 보고서 출력이 가능함; 즉,	병행 적용
	출력 속도를 향상할 수 있음	페이지네이션 등을 통해 검색 조건에 해당하	
		는 전체가 아닌 일부 데이터만 fetch된 경우 사	
		용할 수 없음	



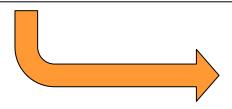
### 아키텍처 평가 - 작성 예제



### 아키텍처 평가 – 작성 예제

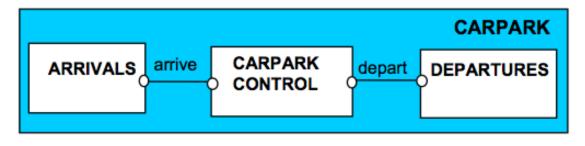
품질 요소	대안 아키텍처 1	대안 아키텍처 2
실시간성	+	_
성능	+	+
유지보수성	_	+

• 대안 아키텍처 리스크 기간계 사용자가 **SFA** 사용자의 정보를 실시간으로 사용할 수 없으므로 업무에 혼란을 줄 수 있음



리스크 대응책 마련

### 아키텍처 평가 - 작성 예제

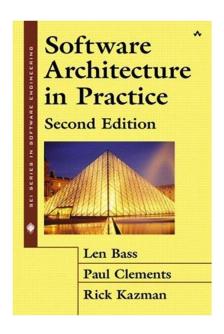


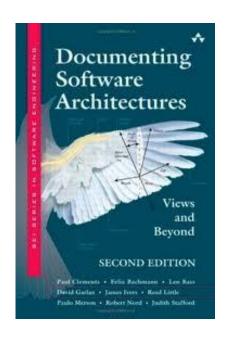
#### Trace:

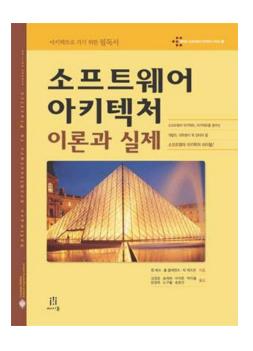
```
p.1.down
p.2.down
p.1.critical
p.2.critical
```



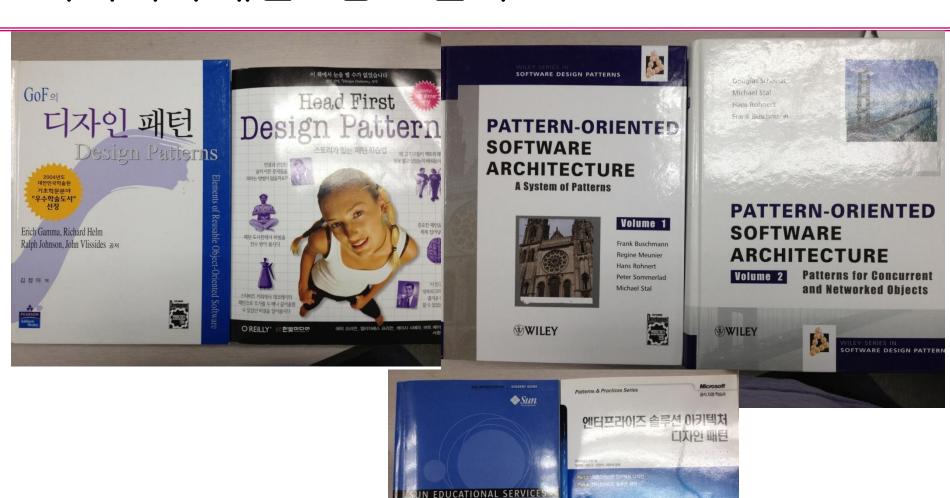
### 아키텍처 설계 - 참조 문서







### 아키텍처 패턴 - 참조 문서



J2EE ™ Patterns 2 SL-500

정보문회사 Microsoft

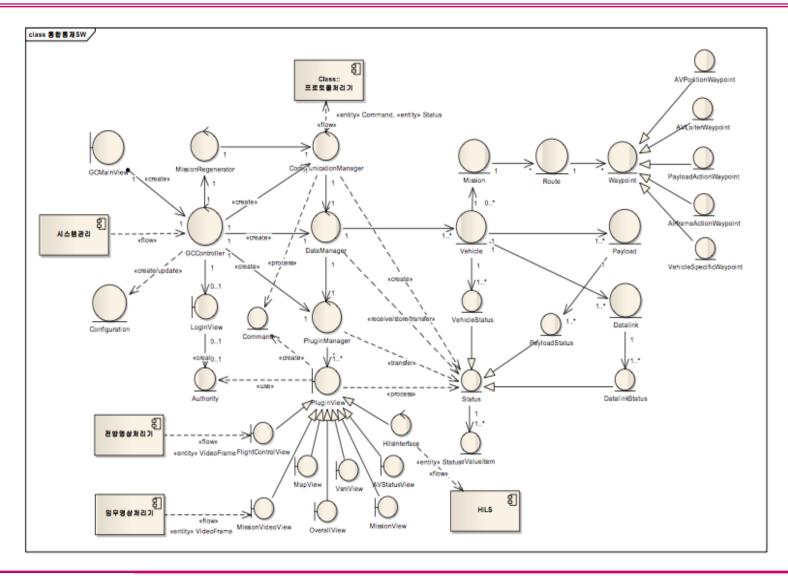
## 아키텍처 실전 사례

(강의 중 제공 예정)

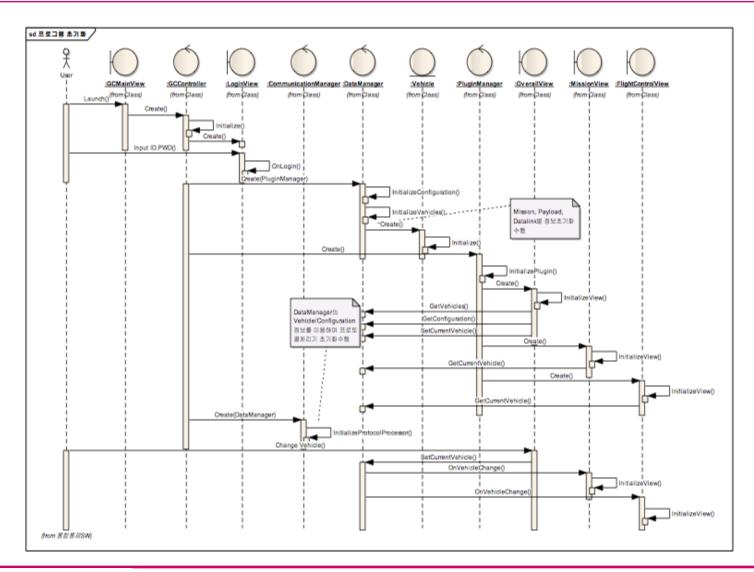
# Questions?

## **Appendix**

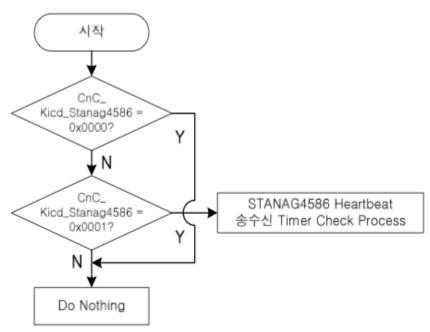
### 상세 설계 - 작성예제



### 상세설계 - 작성예제



### 상세설계 - 작성예제



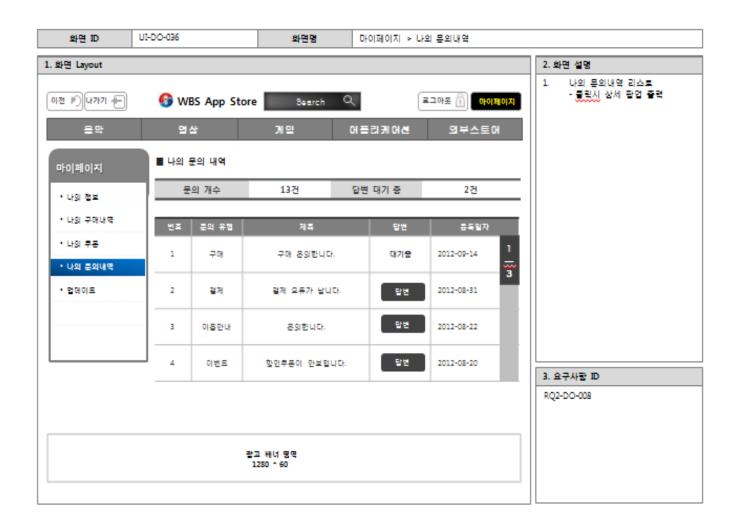
Kicd Mode or Stanag4586 Mode Branch	h Actions		
CnC_Kicd_Stanag4586_Mode == 0x0000?	Y	N	N
CnC_Kicd_Stanag4586_Mode == 0x0001?		Y	N
	1	2	3
Action1:			
Do Nothing			
Action2:			
SDD-FCS-CCCI-SFR-HBMGT_SRTBRCH 수행			
Action3:			

Do Nothing

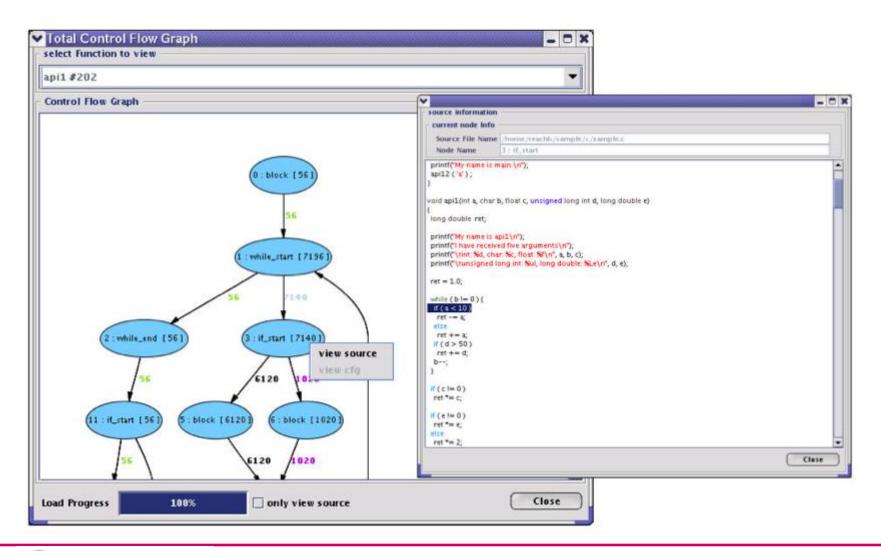
## 상세 설계 - DB 작성예제

현티티명	테이분명	속성명	현행병	월 명	DATATYFE	FE	H	결수역부	티고
바이어 회원	3T_BUYER	회원변호	MNO	회원변호	VARCHAR2	Ÿ		Y	'비'+시작(100000001)
		98D	MD	9181D	VARCHAR2				
		비밀번호	PASSWD	비밀번호	VARCHAR2			Y	
		0 5	NAME	이동	VARCHAR2				
		휴대전화변호	MIMOBILE	휴대전화변호	VARCHAR2				
		생년원일	BIRTHDAY	생년집일	VARCHAR2				
		IPIN 인증키	PIN_CERT	PIN 인증키	VARCHAR2				
		82	SEX	a <b>a</b>	CHAR				
		연필	AGE	g <u>a</u>	NUMBER				

### 상세 설계 – UI 작성예제



### 테스트 - 자동화 단위 테스트 및 커버리지 예제





## 테스트 - 통합 테스트 결과서 예제

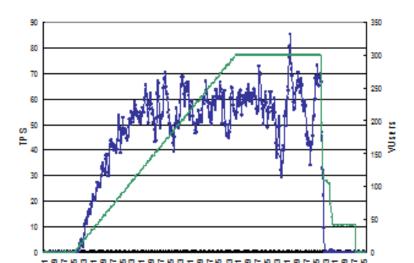
서브 시스템명₽	CanIf↔	업무 구분:	모듈 설정 및 조회:							
데스트 IDe	******									
케이스명~	UT_CanIf_CanIfPrivateConfiguration 레이스 순변의 1의 1의 CanIfPrivateConfiguration 컨테이너 생성과									
입력테스트 ID√	UT_CanIf_CanIfPrivateConfiguration-01	UT_CanIf_CanIfPrivateConfiguration-01₽								
예상 결과↔	CanIfPrivateConfiguration [LINK] 선택 시 CanIfPrivateConfiguration 컨테이너 가 생성되고 파라미터 조회/입력 화면으로 이동한다.₽									
실제 결과♡	- Caniff 모듈 에디터에 CaniffPrivateCon 미터 조회/입력 화면으로 이동하였다.  CanifDicCheck Selects whether the DLC check is supported.  > True: Enabled (default) > Fairs: Disabled    Fairs: Disabled     CanifDicCheck   CaniffDicCheck	CanifNumberOfTxBuffers Defined the number of L-PDU	elements for the transmit buffering. The Tx L-PDU an L-PDU once for each different L-PDU handle. L-PDUs to be used.							
패스 여부∂	₹7									
결함 내용도	T									

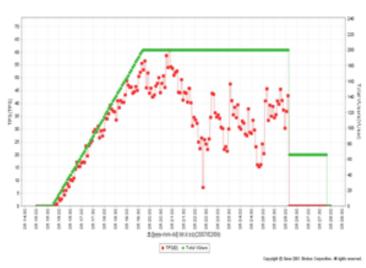


### 테스트 – 시스템 테스트(성능) 예제

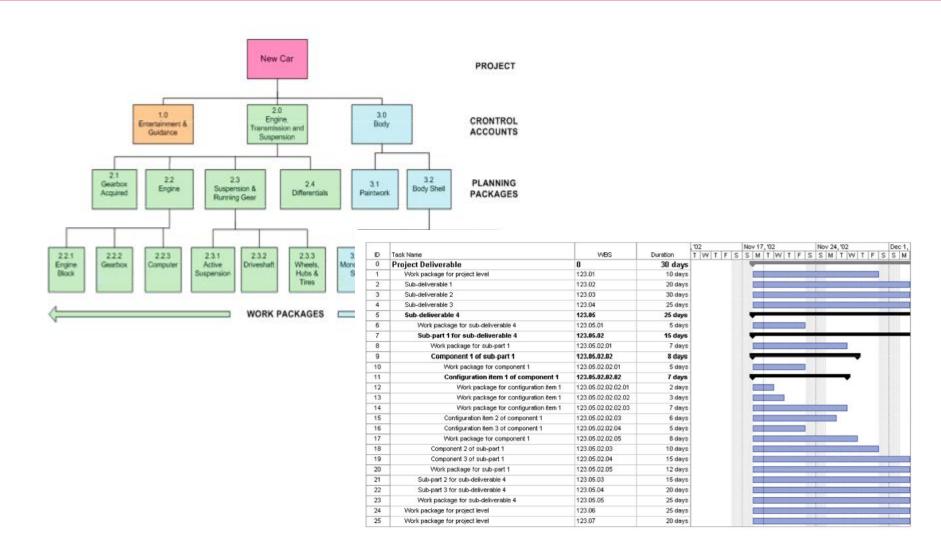
■ NCRM 업무 1차 테스트 결과 비교

1차테스트: ~ 30 TPS
 2차테스트: ~ 60 TPS



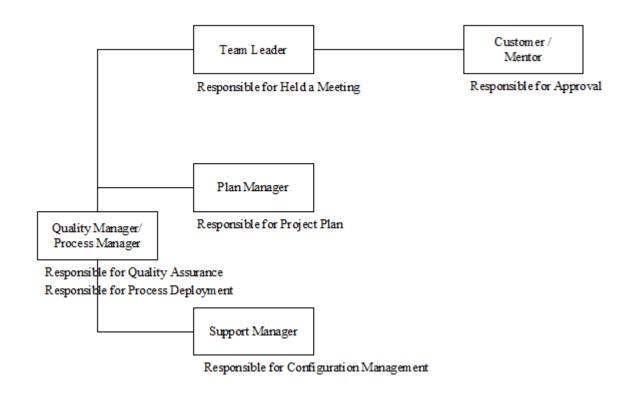


### 프로젝트 관리 – 요구사항관리 예제



### 프로젝트 관리 – 품질 관리 예제

- ❖네이밍 룰
- ❖코딩 룰
- ❖인스팩션
- ❖테스트 지원



## 프로젝트관리 – 리스크/이슈관리 예제

Risks No.	Description	1	Р	Т	Risk exposure	Coping strategies	Owner	Open Date	Status	Last review Date	Target Close Date
R-001	If we should translate all parts of TTCN standard? - TTCN standard has big and detail, we don't have enough time to translate all TTCN.	2.6	2.2		5.72	To mitigate this risk, we'll use estimation techniques such as use-case based estimation and function-point based estimation to anticipate the amount of work. The remaining parts of TTCN specification will be handled in different Studio Project.	Changsup	Oct.28.2004	Closed		Dec.17.2004
R-002	If the tool takes a long time to execute? - We don't know our developing tool doesn't have good performance	2	3.6		7.2	We will review the performance of the lex/yacc tool in this semester. At the end of this semester, we will estimate the time of running our target system using function point and test cases. On this estimation, we should decide on whether we are going on using these lex/yacc tool and we will reflect this prediction on the next iteration.	SeonAh	Oct.28.2004	Closed		Aug.30.2005
R-003	If our client ask us to use Bluetooth protocol for testing?  - We have to know the bluetooth spec also.	2.8	1.8		5.04	Bluetooth is not the scope of TTCN protocol, it is a target protocol for testing. We can't describe test-case for bluetooth because we doen't have domain knowledge of Bluetooth.	Shinyoung	Oct.28.2004	Closed		Nov.8.2004
R-004	JAVA tools selected for parsing might not be able to handle TTCN grammar; project might not be completed.	3	1	3	9	We prepare some cadidate parser tools such as JavaCC or SableCC, However it is impossible to test perfectly these tools, and spending time learning all these tools, Therefore using these candidate tools will be also a risk, If these tools also have a problem, we will change the codes into C language as	Miyul	Dev.28.2004	Closed		



### 프로젝트 관리 - 형상관리 예제

혐상변경신청서									
요참자	김영곤	소속	휴원	휴원 요청 및자 2012.09					
형상 항목	8DD-0	4-GUI	형상 번호 v1.0						
요청 내용	X 윈도무 시스템의 이식 구조 변경으로 인해 관련 사항 변경								
원인 및 해결 방안	사유: X 서버를 고신되 RTOS 상에 이식하는 것 보다 그래찍카드 내부의 Linux 상에 이식하는 것 이 쉽고 효율적이며, 설능과 시스템 경랑화 면에서도 이점이 있음 해결 방안: X 서버를 그래찍카드 내부에 이식하고 고신되 RTOS 상의 클라이언트와 PCI를 통해 통신하도록 설계 변경								
영향 평가 및 기타 고려사항	1. "TCP/IP 소켓 연돔" 인터페이스 제거								
	검토자	소속		검토 의견					
변경 요청 검토	길설호	현대트템	능 "TCP/IP 소켓 면동"	서 요청한 사항으로  이 몰필요함에 따라   대한 영향설 검토	인터페이스를 제				
-5 -5 -	하대언	MD8테크놀로지	요구사항 변경에 따 것으로 판단됨.	라 수점사항으로 변	경에 문제는 없음				
	설점훈	코츠테크놀로지	2차년도 Kick-off에	서 논의한 사항,형성	상변경 수용				
	김설관	지인정보기술	형상변경 수용						
숨인 여부	☑ 숨인 □기각 [	□보류	형상관리위원회	전체기관					
검토 결과	형상번경 승인, TCP/IP 소켓 연돔 인터페이스 삭제는 타 C8CI에 영향을 주지 않을 것이며, 향후 네르워크 스택의 인터페이스 요구사항 변경 필요								
	변경된 형	형상 항목	변경	내용	등록 밀자				
	8DD-0	4-GUI	스템 이식 구조 변경	영으로 인해 관련 내용	2012.11.15				



## Sprint 관리 - 예제

Iteration	C-I2					
Version	0.3					
Date	2013-11-04 ~ 2013-	-11-15 (2	?주)			
Goal	C-I1 에서 도출된 Class 들을 구현 구현하면서 전체적인 일정 산출					
완료 율	65%	전체	60			
		완료	39			
지연 율	28%	취소	4			
		C-I3	17			
긴급이슈	7건		20			



