

아키텍처 평가 - ATAM

2004. 05. 18



목차

- ATAM이란?
- ATAM 구조
- ATAM 평가 단계
- ATAM 산출물
- 요약

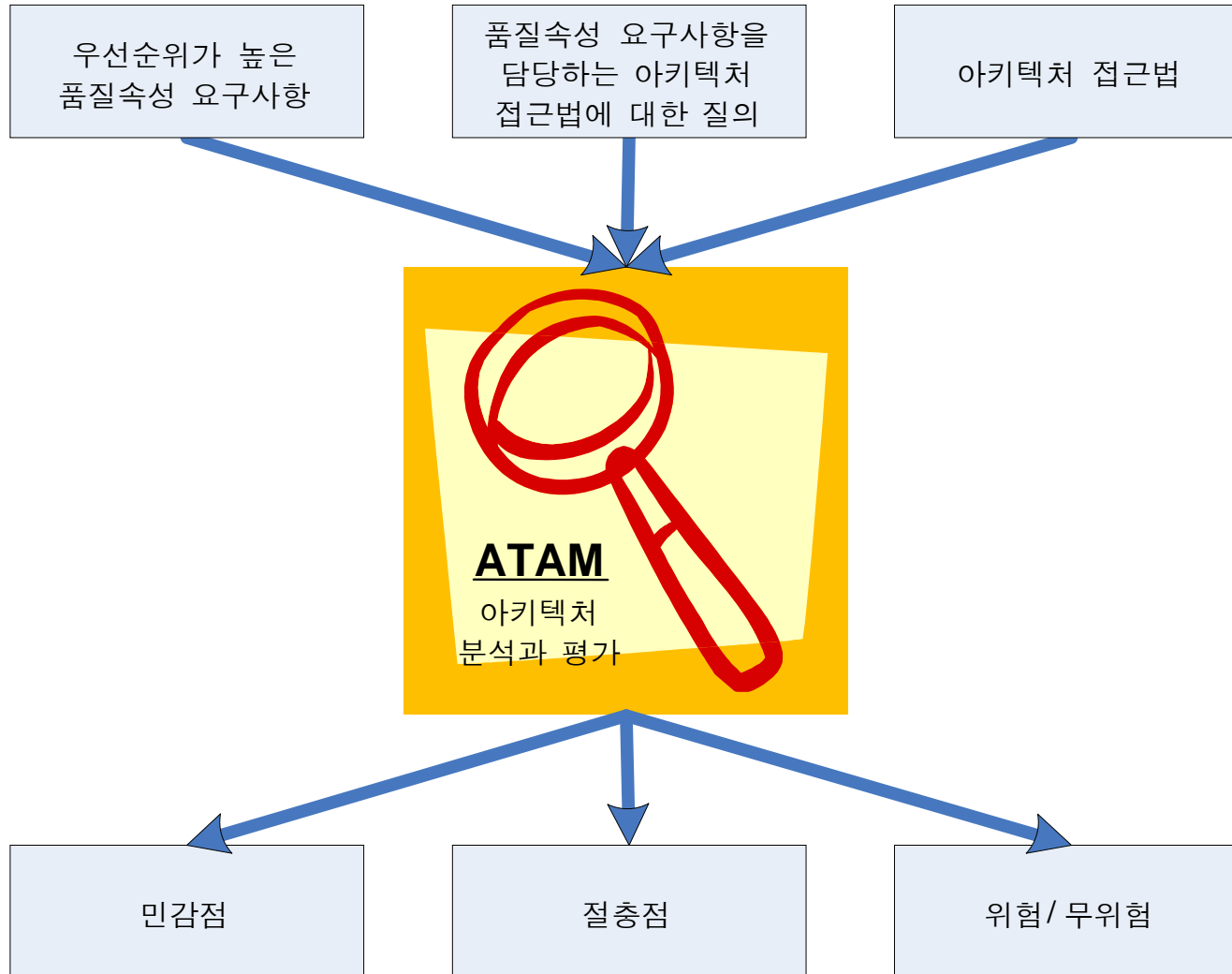
ATAM이란?

■ Architecture Trade-off Analysis Method

■ 특징

- 아키텍처가 품질속성을 만족시키는지 판단할 뿐만 아니라 품질속성들이 서로 어떻게 상충하면서 상호작용하는지(trade-off)까지 밝힌다.
- 분석 절차를 잘 정의하고 있다.
- 레거시(Legacy) 시스템 분석에도 적용할 수 있다.
- 아키텍처 스타일 (architectural style), 품질속성 분석, SAAM(Software Architecture Analysis Method)으로부터 많은 영향을 받았다.

ATAM이란?



ATAM 구조

단계 Phase	협력관계 구축과 프로젝트 준비	평가				후속조치
		소개	조사와 분석	테스트	보고	
Phase 0						
Phase 1						
Phase 2						
Phase 3						

ATAM 구조

■ Phase0 : 협력관계 구축과 준비 국면

- 참가자 : 아키텍처 평가팀 리더/프로젝트 의사결정자
- 활동
 - 협력관계 구축
 - 평가팀 리더는 아키텍처 평가 방법을 간단히 소개한다.
 - 평가대상 조직은 아키텍처에 대한 자료를 제공하고 평가팀 리더는 프로젝트 진행을 결정한다.
 - 평가대상 조직과 평가수행 조직은 계약을 체결한다.
 - 정보열람 권리와 정보보호 의무를 확인한다.
 - 아키텍처 평가 비용과 이득을 평가대상 조직이 충분히 이해하도록 해야 한다.
 - 아키텍처 평가 준비
 - 평가팀 구성 : 팀을 구성하고 팀원의 역할을 결정한다.
 - Kick-off 미팅 : 팀원들이 자신의 역할을 이해하고 아키텍처 평가에 대한 지식을 공유한다.
 - Phase1 준비

ATAM 구조

■ Phase1 : 평가 국면 1

- 참가자 : 아키텍처 평가팀, 프로젝트 의사결정자
- 기간 : 1 일
- 활동
 - ATAM 평가 : 1단계에서 6단계를 수행한다.
 - Phase2 준비
 - Phase2 진행 여부 결정
 - Phase2 참여자 결정
 - 부족한 정보를 파악한다.
- 특징
 - 아키텍처 중심. 아키텍처에 대한 정보를 발굴하고 분석한다.
 - 아키텍처를 파악하는 데 초점을 맞춘다. 자세한 분석은 Phase2에서 수행한다.
 - 파악한 아키텍처를 기반으로 아키텍처 평가팀 구성을 최종 결정한다.
 - Phase1과 Phase2 사이에 2~3주 공백 기간을 두고 이 기간 동안 계속 아키텍처에 대한 정보를 발견하고 수집하고 만든다.

ATAM 구조

■ Phase2 : 평가 국면 2

- 참가자 : 아키텍처 평가팀, 프로젝트 의사결정자, 다양한 이해관계자들
- 기간 : Phase1이 끝나고 2~3 주 뒤에 2일
- 활동
 - ATAM 평가 : 1단계에서 9단계를 수행한다.
- 특징
 - 이해관계자 중심. 이해관계자의 관심사를 찾아내고 Phase1에서 Phase1을 통해 충분히 이해한 아키텍처를 분석하고 평가한다.

ATAM 구조

■ Phase1과 Phase2 비교

	Phase1	Phase2
참가자	이해관계자 2~4명 참가	이해관계자 10~15명 참가
특징	아키텍처 중심	이해관계자 중심
	문제 해결 중심	문제 중심
	분석 중심	검증 중심
목적	아키텍처 이해를 돕는다.	이해관계자들 사이의 상호 작용을 돕는다.
시나리오 용도	유틸리티 트리를 만들기 위 한 시나리오	유틸리티 트리를 검증하기 위한 시나리오
진행방식	다양한 비공식 방법	정규 회의

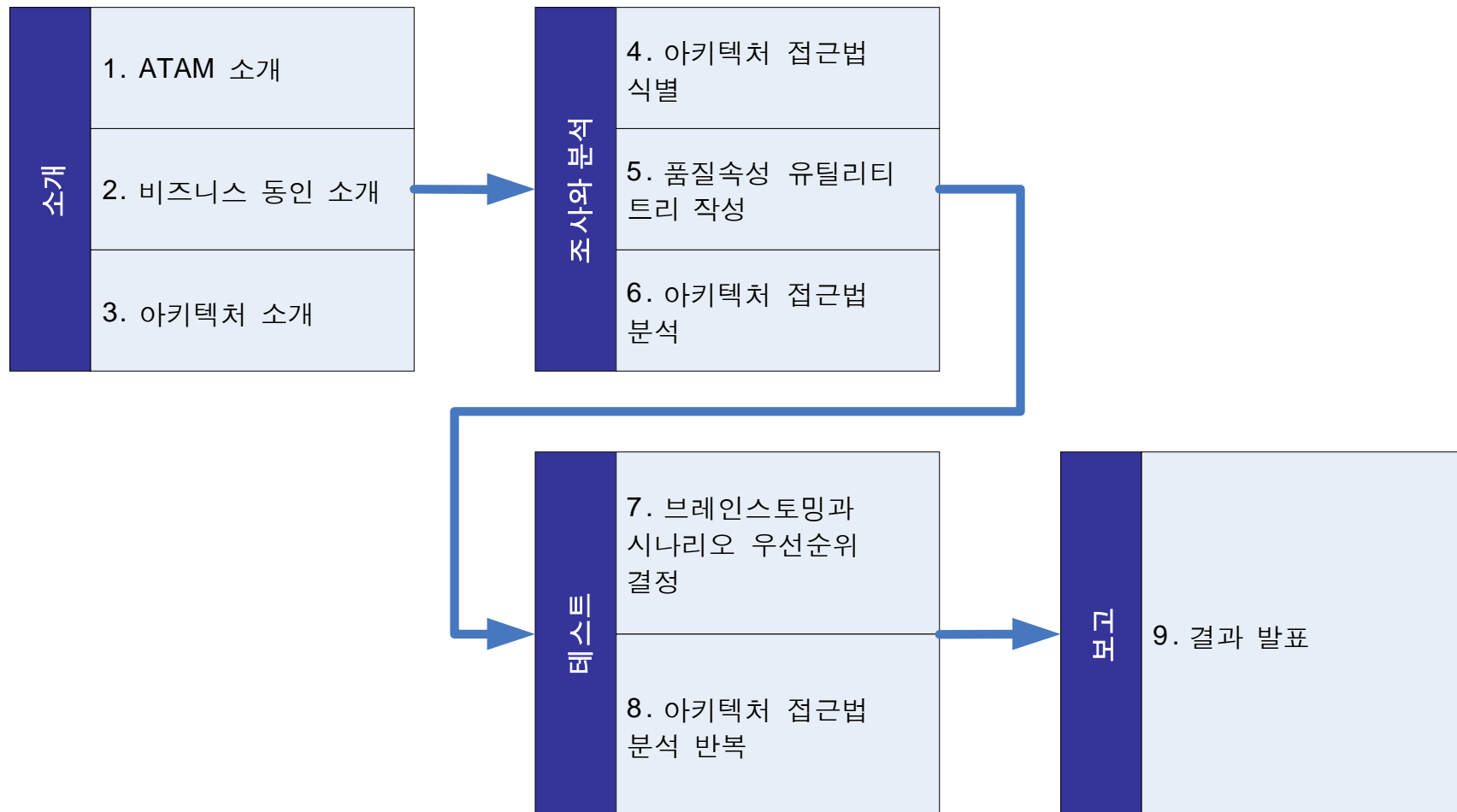
ATAM 구조

■ Phase3 : 후속조치 국면

- 참가자 : 아키텍처 평가팀, 아키텍처 평가 대상 조직
- 기간 : 1 주
- 활동
 - 최종 보고서 작성
 - 무엇을 평가하고 분석했나?
 - 분석하고 평가한 결과는?
 - 결론은 무엇인가?
 - 평가 방식 개선 사항 수집과 평가 만족도 측정을 위한 자료 수집
 - 평가팀과 아키텍처 평가 대상 조직으로부터 자료 수집
 - 개선점 수집, 비용 데이터와 이익 데이터 수집
 - 산출물 리파지토리 갱신
 - 다음 평가를 위해 현재 평가 결과를 저장한다.

ATAM 평가 단계

- ATAM 평가 국면은 다음과 같이 9단계로 이루어진다.



1. ATAM 소개

- 아키텍처 평가 리더가 이해관계자를 모아 놓고 아키텍처 평가 절차를 설명한다.
 - ATAM 단계들을 간단히 설명한다.
 - 기법을 설명한다.
 - 평가 산출물을 설명한다.

2. 비즈니스 동인(動因) 소개

- 이해관계자들과 아키텍처 평가 팀이 시스템이 태어나게 된 배경과 비즈니스 동인을 이해한다.
- 프로젝트 의사 결정권자(PM이나 고객)가 비즈니스 관점에서 시스템의 전체 모습을 설명한다.
 - 시스템의 가장 중요한 기능은 무엇인가?
 - 기술, 관리, 경제, 정치 문제 때문에 생긴 제약조건은 무엇인가?
 - 프로젝트와 관련있는 비즈니스 목표와 배경 지식은 무엇인가?
 - 주요 이해관계자는 누구인가?
 - 아키텍처에 영향을 미친 품질 속성인 아키텍처 동인은 무엇인가?

3. 아키텍처 소개

- 수석 아키텍트가 아키텍처 평가팀에게 아키텍처를 설명한다.
 - 기술 제약사항을 설명한다.
 - 시스템이 상호작용해야 하는 외부 시스템을 설명한다.
 - 품질속성을 만족시키기 위해 적용한 아키텍처 접근방식을 설명한다.
- 아키텍트는 다양한 뷰들 가운데 가장 중요하다고 생각하는 뷰로 아키텍처를 소개한다.
- 아키텍처 팀은 아키텍처 접근법을 찾아서 초안을 만들 수 있다.

4. 아키텍처 접근법 식별

- 아키텍트에게 직접 묻거나 단계 3에서 작성한 초안을 통해 아키텍처 접근법들을 찾아낸다. 하지만, 아직 아키텍처 판단들을 “분석”하지는 않는다.
- 아키텍처 접근법과 이 접근법에 따라 선택한 아키텍처 스타일(architectural style)을 찾아내야 한다.
 - 아키텍처 접근법과 아키텍처 스타일은 중대한 품질 요구사항을 예측할 수 있는 방법으로 달성할 수 있게 만드는 도구이다.
 - 아키텍처 스타일은 아키텍처 접근법을 선택한 근거를 제시하고 아키텍처가 따라야 하는 규칙을 제시하기 때문에 중요하다.
 - 아키텍처 스타일은 아키텍처가 따라야 하는 규칙이 될 수도 있다.

4. 아키텍처 접근법 식별

■ ABAS(Atttribute-Based Architectural Style)

- 특화된 아키텍처 스타일
- 특정 품질속성을 달성하는 방법까지 설명한 아키텍처 스타일
 - <예> 성능 ABAS : 프로세서에 프로세스를 할당하는 방법, 프로세스끼리 자원을 공유하는 방법, 프로세스 우선순위를 결정하는 방법 같이 성능이라는 품질속성을 아키텍처 스타일이 정한 구성요소들이 어떻게 달성하는지 설명한다.

5. 품질속성 유틸리티 트리 작성

- 아키텍처 평가팀과 프로젝트 의사결정자가 모여 시스템에서 가장 중요한 품질속성 요구사항을 찾아서 우선 순위를 결정하고 정제한다.
- 품질속성 우선순위를 공유한다.
- 품질속성 요구사항 우선순위 정의서와 시나리오 (scenario)를 만든다.

아키텍처 평가팀은 이 우선순위에 따라 집중해서 평가할 부분을 결정한다.

5. 품질속성 유틸리티 트리 작성

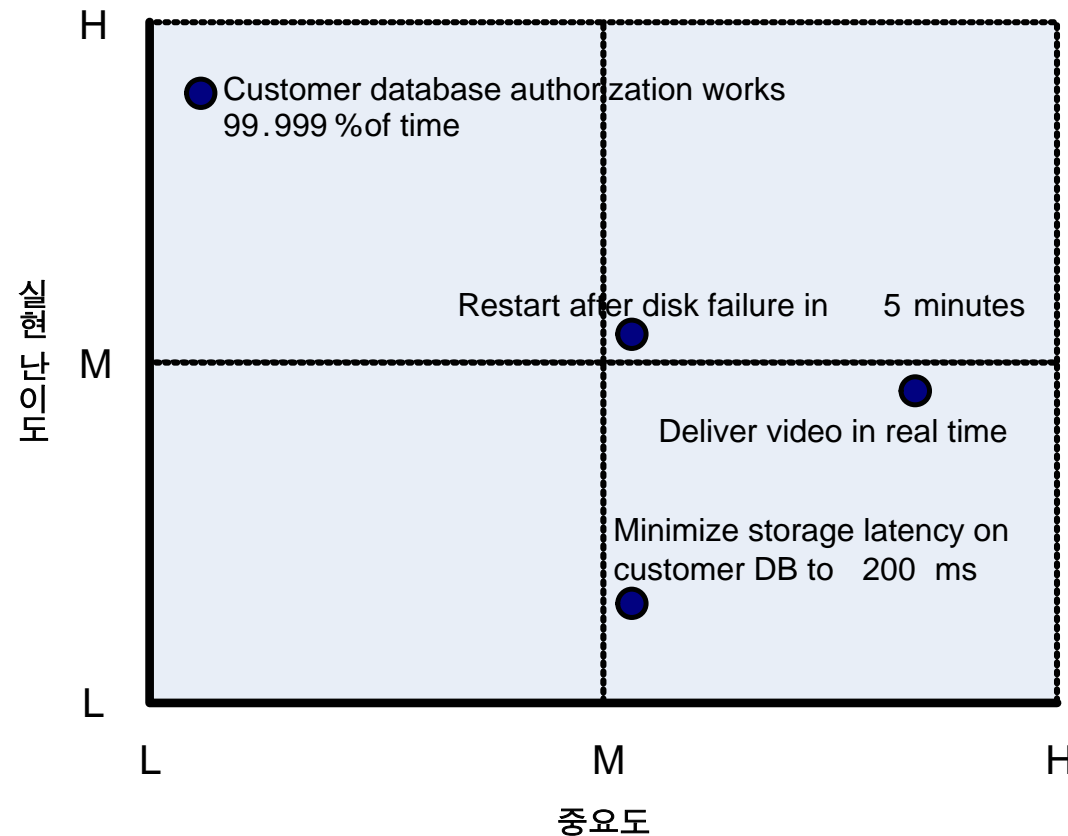
■ 유틸리티 트리

- 유틸리티란 시스템이 제공해야 하는 모든 품질이다.
- 유틸리티 → 품질속성 → 세분화한 품질속성 → 시나리오 순서로 트리를 만든다.
- 모든 시스템 이해관계자와 프로젝트 평가 팀이 시스템 필수 성공 요소를 한 눈에 파악하는 도구이다.
- 단계 2에 나온 비즈니스 동인을 품질속성 시나리오로 쉽게 바꿀 수 있는 메커니즘을 제공한다.
- 품질속성 요구사항들의 우선순위를 결정하는 데 도움이 된다.
- 품질 요구사항을 정확하게 정의해서 품질 요구사항을 정립한다.
- 시나리오의 우선순위 점수를 매긴다.

5. 품질속성 유틸리티 트리 작성

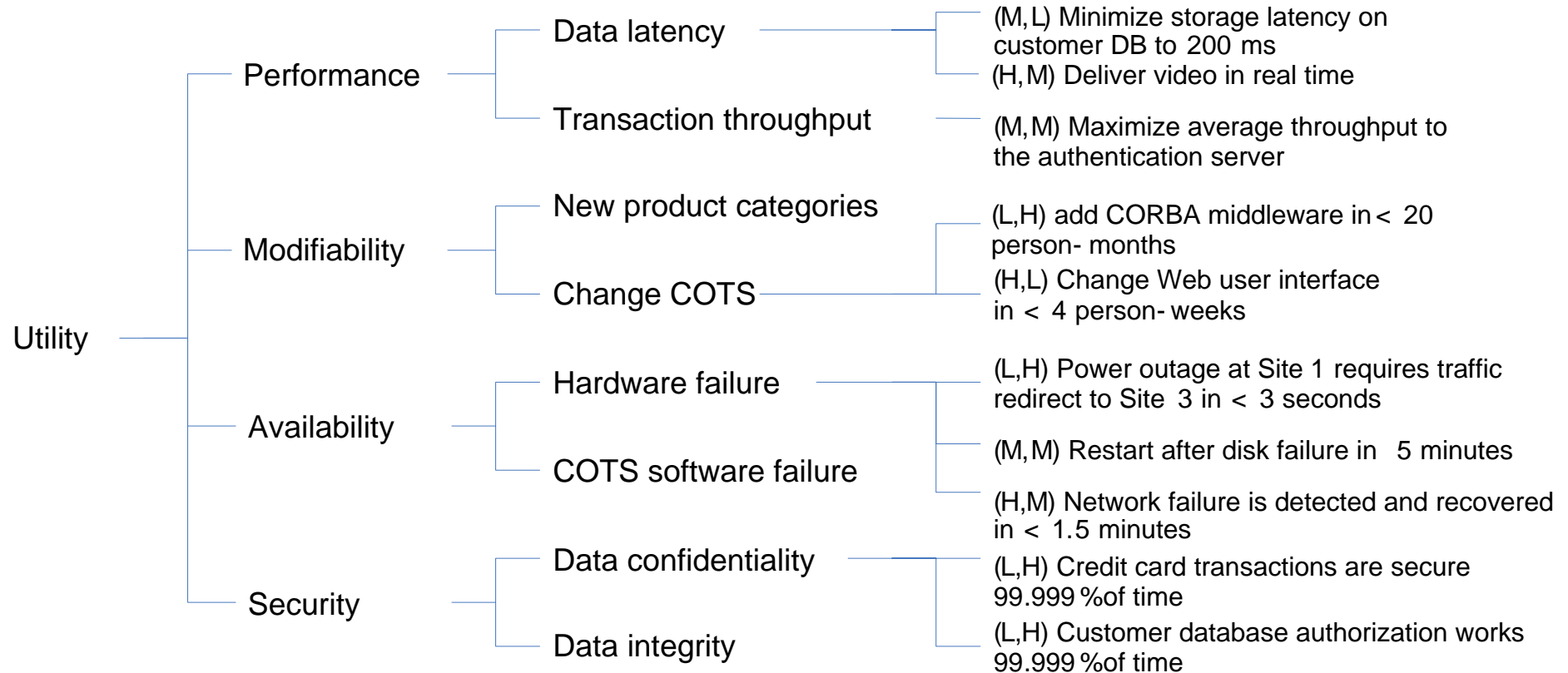
■ 유틸리티 트리

- 우선순위 결정 매트릭스 : 실현 난이도와 중요도를 기준으로 시나리오를 배치한다.



5. 품질속성 유틸리티 트리 작성

■ 유틸리티 트리 예제



5. 품질속성 유틸리티 트리 작성

- 시나리오란?
 - 품질속성을 찾아내는 방법으로 시스템과 이해관계자의 상호작용을 예를 들어 묘사하는 산문 형식의 글이다.
- 왜 시나리오 방식을 쓰는가?
 - 만들기 쉽고 이해하기 쉽다.
- 시나리오는 어떤 역할을 하는가?
 - 이해관계자의 관심사를 표현한다.
 - 품질속성 요구사항을 이해한다.
- 시나리오의 구조
 - 자극(Stimuli) : 이해관계자는 무엇을 하면 시스템과 상호작용이 시작되는가?
 - 반응(Response) : 시스템은 자극에 어떻게 반응하는가?
 - 환경(Environment) : 자극을 받았을 때 시스템은 어떤 상태인가?

5. 품질속성 유틸리티 트리 작성

- 시나리오의 종류 : 유스케이스 시나리오
 - 사용자가 완성 시스템을 목적을 가지고 이용하는 상호작용을 설명한다.
 - <예>
 - 사용자는 프로젝트 데이터를 다시 조회하지 않고도 여러 회계연도의 예산과 실제 비용 데이터를 비교할 수 있어야 한다. [사용성(usability)]
 - 만약, 데이터를 제공할 때 문제가 생긴다면 시스템은 데이터를 요청한 사용자들에게 전자메일을 발송하고 화면에 문제를 일으킨 상황을 표시한다. [신뢰성(reliability)]
 - 목표물이 다가올 때 탄도를 100 밀리초 안에 보정해서 포탄을 발사해야 한다. [성능(performance)]
 - 모든 유스케이스 시나리오는 이해관계자가 바라는 바를 반영한다.

5. 품질속성 유틸리티 트리 작성

- 시나리오 종류 : 성장(growth) 시나리오
 - 충분히 예견할 수 있는 시스템의 변경을 표현한다.
 - <예>
 - 새로운 메시지 종류를 리파지토리에 넣을 때 한 사람이 일주일 안에 처리할 수 있어야 한다.
 - 운영체제를 같은 종류로 바꿀 때 한 사람이 일년 안에 처리할 수 있어야 한다.
 - 데이터 양이 두배가 되더라도 1초안에 데이터를 추출할 수 있어야 한다.

5. 품질속성 유틸리티 트리 작성

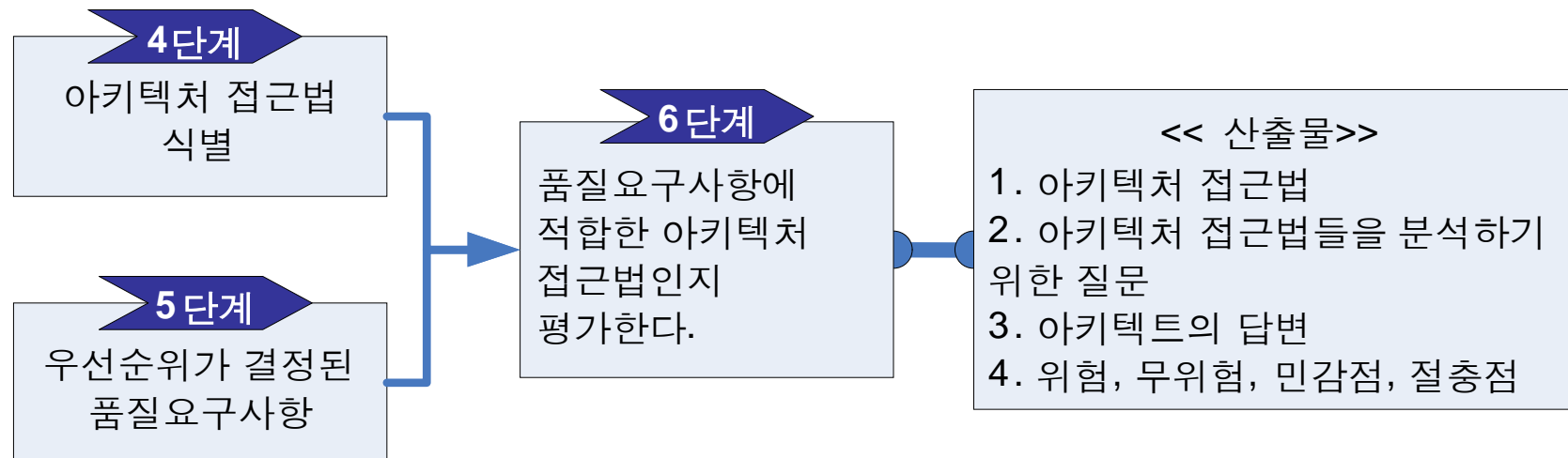
- 시나리오 종류 : 사전탐사(exploratory) 시나리오
 - 절대 일어날 것 같지 않은 시스템의 변경을 표현한다.
 - 시스템 한계와 정확하게 드러나지 않은 가정들을 찾아내는 것이 목표다.
 - <예>
 - 운영체제를 유닉스에서 매킨토시로 바꾼다.
 - 25년 된 제어 시스템을 재사용하여 차세대 항공기에 장착한다.
 - 시스템 가용성을 98%에서 99.999%로 높인다.

세가지 시나리오는

1. 시스템의 다른 측면들을 드러낸다.
2. 아키텍처를 평가하는 세가지 전략을 제공한다.
3. 이해관계자들이 아키텍처를 이해하고 서로 이야기할 수 있는 방법을 제공한다.

6. 아키텍처 접근법 분석

- 4단계와 5단계의 결과를 바탕으로 아키텍처 접근법이 품질요구사항에 적합한지 검사하는 단계다.
 - 품질속성에 대한 초보 분석을 수행할 수 있도록 관련 아키텍처 판단에 대한 충분한 정보를 수집한다.
 - 평가하고 있는 아키텍처 안에서 실제로 실현되는 아키텍처 판단들이 품질속성 요구사항을 달성하는 데 적절하다는 것을 확신하는 것이 이 단계의 목표다.



6. 아키텍처 접근법 분석

■ 산출물

□ 아키텍처 접근법

- 4단계에서 아키텍처 접근법을 모두 식별해야 한다. 만약, 빠진 것이 있다면 그 이유를 찾아야 한다.
- 아키텍트는 자신이 선택한 접근법의 구성요소, 구성요소 사이의 관계와 제약사항까지 정리해 놓아야 한다.

□ 아키텍처 접근법에 대한 분석 질의

- 경험을 미리 정리해 놓은 ABAS, 관련 서적, 이해관계자들의 경험을 토대로 질의를 만든다.

□ 분석 질의에 대한 아키텍트의 답변

□ 위험 / 무위험 (risk/nonrisk), 민감점 (sensitivity points), 절충점 (trade-off points)

- 이 세가지 항목은 5단계에서 찾은 품질속성 요구사항들과 관련있다.

6. 아키텍처 접근법 분석

- 아키텍처 접근법에 대한 분석 질의
 - 아키텍처 접근법을 자세히 이해할 수 있다.
 - 아키텍처 접근법이 가진 약점을 찾아낼 수 있다.
 - 아키텍처 접근법의 민감점과 절충점을 찾아낼 수 있다.
 - 다른 아키텍처 접근법과 관계(상호작용, 상충...)을 알 수 있다.

위험/무위험, 민감점, 절충점을 설명할 수 있는 기초 자료 제공

6. 아키텍처 접근법 분석

- 아키텍처 평가팀은 민감점, 절충점, 위험, 무위험을 식별해서 기록해야 한다.
- 모든 민감점과 절충점은 잠재 위험이므로 분석이 끝난 시점에는 모두 위험이나 무위험에 대응되어야 한다.
- 6단계가 끝나면 아키텍처 평가팀은 아키텍처의 중요한 모습들을 거의 다 파악하고 테스트 단계로 넘어간다.

6. 아키텍처 접근법 분석

■ 아키텍처 접근법 분석서 양식

아키텍처 접근법 분석서(Analysis of Architecture Approach)				
시나리오 번호 (Scenario #)	번호	시나리오 (Scenario)	유틸리티 트리에 나온 시나리오 문장	
품질속성 (Attributes)	시나리오와 관련된 품질속성			
환경 (Environment)	시나리오가 실행되는 조건과 시스템이 동작하는 환경에 대한 가정			
자극 (Stimuli)	시나리오가 시작되는 자극에 대한 정확한 설명(예: 함수호출, 시스템 장애, ...)			
대응 (Response)	자극에 대한 시스템의 정확한 반응(예: 응답시간, 수정 난이도, ...)			
아키텍처 판단 (Architectural Decision)	민감점 (Sensitivity)	절충점 (Trade -off)	위험 (Risk)	무위험 (Nonrisk)
시나리오로 표현한 품질속성에 해당하는 시스템 반응에 영향을 미치는 아키텍처 판단들	민감점 번호	절충점 번호	위험 번호	무위험 번호
...
...
근거 (Reasoning)	앞에 나온 목록에 등장하는 아키텍처 판단들이 시나리오가 표현하는 품질속성을 달성하는 데 공헌한다는 근거			
아키텍처 다이어그램 (Architectural Diagram)	앞에 나온 근거를 뒷받침할 수 있는 다이어그램			

6. 아키텍처 접근법 분석

■ 아키텍처 접근법 분석서 예제

아키텍처 접근법 분석서				
시나리오 번호	A05	시나리오	프라이머리 CPU에 발생하는 하드웨어 장애를 감지해서 복구한다.	
품질속성	가용성(Availability)			
환경	일반 운영 환경			
자극	CPU 가운데 하나가 장애를 일으킴			
대응	장애 CPU가 처리하던작업을 다른 CPU로 넘긴다 이때 99.9999% 가용성을 유지해야한다			
아키텍처 판단	민감점	절충점	위험	무위험
CPU백업	S2	.	R8	.
데이터채널을백업하지않음	S3	T3	R9	.
Watchdog	S4	.	.	N12
Heartbeat	S5	.	.	N13
Failover Routing	S6	.	.	N14
근거	<ul style="list-style-type: none"> ❖ 다른 하드웨어와 운영시스템을 도입해서 같은 원인으로 동시에 장애에 빠지지 않도록 한다 (위험 R8 참조) ❖ 최악의 경우라도 4초 안에 작업을 넘길 수 있다 ❖ Heartbeat 과 watchdog 방식을 쓸 때 2초 간격으로 장애를 감지할 수 있다 ❖ Watchdog은 간단하고 신뢰성이 증명되었다 ❖ 데이터채널을 백업하지 않으면 가용성 요구사항을 지키지 못할 수 있다 			
아키텍처 다이어그램	<pre> graph TD In(()) --> P[프라이머리 CPU OS1] In --> B[Watchdog을 가진 백업 CPU OS2] P -- "1초 간격 heartbeat" --> B P --> S[스위치 CPU OS1] B --> S S --> Out(()) </pre>			

7. 브레인스토밍과 시나리오 우선순위 결정

- 5단계에서 유틸리티 트리로 찾은 품질속성과 시나리오를 검증한다.
- 가능한 모든 이해관계자가 브레인스토밍을 통해 시나리오를 찾고 이 시나리오들의 우선순위를 결정한다. 우선순위가 높은 시나리오를 뽑아 5단계에서 만든 유틸리티 트리와 비교하여 아키텍처가 이해관계자의 의중을 얼마나 정확하게 반영했는지 판단한다.
- 이해관계자와 아키텍트 사이의 괴리 자체가 엄청난 위험 요소이기 때문에 이런 검증이 필요하다.

7. 브레인스토밍과 시나리오 우선순위 결정

■ 유틸리티 트리 방식과 브레인스토밍 방식 비교

	유틸리티 트리	시나리오 브레인스토밍
이해관계자	아키텍트, 프로젝트 리더	모든 이해관계자
참여 규모	평가자 : 2 ~ 3 명	평가자 : 5 ~ 10 명
일차 목표	품질속성 요구사항을 찾고, 구체화하고, 우선순위를 결정한다.	이해관계자들 사이의 원활한 의사소통을 통해 유틸리티 트리에서 도출한 품질속성 목표가 적합한 것인지 밝힌다.
접근법	Generic to specific : 품질속성로부터 시나리오를 도출	Specific to generic : 시나리오로부터 품질속성을 도출

7. 브레인스토밍과 시나리오 우선순위 결정

- 브레인스토밍 방식의 테스트
 - 브레인스토밍
 - 시나리오를 찾는다.
 - 5단계에서 찾기만 하고 분석하지 않은 시나리오가 좋은 출발점이 된다.
 - 우선순위 결정
 - 투표 : 브레인스토밍으로 찾은 시나리오를 대상으로 투표 (전체 시나리오 개수의 30%만큼 투표권을 부여)
 - 투표 결과에 따라 중요 시나리오 선정
 - 유틸리티 트리에 배치하여 5단계 결과와 비교
 - 일치
 - 품질속성은 일치하지만 새로운 시나리오
 - 새로운 품질속성을 발견 → 아키텍트와 이해관계자의 심각한 괴리

8. 아키텍처 접근법 분석 반복

- 아키텍처 평가 리더는 7단계에서 찾아낸 최우선 시나리오를 아키텍트가 처리할 수 있도록 지침을 제공한다.
- 아키텍트는 아키텍처 접근법이 시나리오를 실현할 때 어떻게 공헌하는지 설명한다.
- 6단계와 같은 활동을 반복한다.
 - 7단계에서 새롭게 찾은 시나리오에 알맞은 아키텍처 접근법을 찾아서 분석한다.
 - 7단계에서 새롭게 찾은 시나리오가 없으면 8단계에서 테스트만 한다.

9. 결과 발표

- ATAM 결과로 수집한 정보를 요약하여 이해관계자들에게 발표한다.
- 발표방법
 - 슬라이드를 활용하여 구두로 결과를 발표한다. 아키텍처 평가 리더는 ATAM의 모든 과정을 이해관계자들에게 상기시킨다.
 - ATAM 산출물을 설명한다.
 - 아키텍처 접근법 카탈로그
 - 시나리오와 우선순위
 - 유틸리티 트리
 - 위험/무위험, 민감점, 절충점
 - 아키텍처 질문
 - 최종 보고서를 제공한다.

9. 결과 발표

■ Risk Theme

- 위험을 비슷한 부류로 분류해서 모아 놓은 것
- Risk theme이 2단계에서 찾은 어떤 비즈니스 동인에 영향을 주는 지 밝힌다.
 - ATAM 시작과 최종보고가 일치하도록 하여 전체 평가의 완성도를 높인다.
 - 주의해서 관리해야 할 위험을 드러낸다.

9. 결과 발표

■ 아키텍처 문제 해결

- 아키텍처 평가는 해당 프로젝트에서 적용한 아키텍처 접근법을 근간으로 아키텍처를 분석하고 평가한 것이기 때문에 더 좋은 아키텍처 설계와 분석 방법이 있다면 추천한다.
- 하지만, ATAM은 문제 해결 방식을 제공해 주지는 않는다. ATAM은 아키텍처의 위험요소를 찾아낼 뿐이다. 찾아낸 위험요소는 다양한 방식으로 극복한다.

ATAM 산출물

ATAM 단계 \ ATAM 산출물	품질속성 요구사항 우선 순위 정의서	적용한 아키텍처 접근법 카탈로그	아키텍처 접근법에 대한 분석 질의서	품질속성과 아키텍처 접근법 관계 정의서	위험/무위험 정의서	민감점과 절충점 정의서
1. ATAM 소개						
2. 비즈니스 동인 소개	(1)				(2)	
3. 아키텍처 소개		★			(3)	(4)
4. 아키텍처 접근법 식별		★	★		(5)	(6)
5. 품질속성 유틸리티 트리 작성	★					
6. 아키텍처 접근법 분석		(7)	★	★	★	★
7. 브레인스토밍과 시나리오 우선순위 결정	★					
8. 아키텍처 접근법 분석 반복		(7)	★	★	★	★
9. 결과 발표						

★ : 산출물을 만드는 주요 단계

- (1) 비즈니스동인이 정제되지 않은 품질요구사항을 포함할 수 있다.
- (2) 비즈니스동인을 소개하면서 잘 알려진 위험을 찾을 수 있다.
- (3) 아키텍처가 찾은 위험도 있다.
- (4) 아키텍처가 찾은 민감점과 절충점도 있다.
- (5) 아키텍처 접근법마다 적용했을 때 나타날 수 있는 위험을 정의한다.

- (6) 아키텍처 접근법마다 자신의 민감점과 절충점을 정의하고 있다.
- (7) 4단계에서 미처 발견하지 못한 아키텍처 접근법을 찾을 수도 있다. 이렇게 찾아낸 아키텍처 접근법은 아키텍처 접근법 질의서에 영향을 미칠 수 있다.

요약

- ATAM은 아키텍처가 품질속성을 만족시키는지 판단할 뿐만 아니라 품질속성들이 서로 어떻게 상충하면서 상호작용하는지(trade-off)까지 밝힌다.
- ATAM은 보통 4 Phase로 진행된다. Phase 0에서는 아키텍처 평가를 준비하고 Phase 1과 Phase 2에서 실제 아키텍처 평가 작업을 수행하고 Phase 3에서는 후속 조치를 한다.
- 아키텍처 평가는 1.ATAM 소개 → 2. 비즈니스 동인(動因) 소개 → 3. 아키텍처 소개 → 4. 아키텍처 접근법 식별 → 5. 품질속성 유틸리티 트리 작성 → 6. 아키텍처 접근법 분석 → 7. 브레인스토밍과 시나리오 우선순위 결정 → 8. 아키텍처 접근법 분석 반복 → 9. 결과 발표 순서로 이루어진다.