

제129회 컴퓨터시스템응용기술사 해설집

2023.02.04

국가기술자격 기술사 시험문제

기술사 제 129 회

제 2 교시 (시험시간: 100 분)

분야	정보통신	자격 종목	컴퓨터시스템응용기술사	수검 번호		성 명	
----	------	----------	-------------	----------	--	--------	--

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 10 점)

1. CXL(Computer Express Link)에 대하여 다음을 설명하시오.

- (1) CXL 의 목적
- (2) CXL 의 3 가지 프로토콜인 CXL.io, CXL.cache 및 CXL.memory
- (3) CXL 의 디바이스 유형인 Type 1 디바이스, Type 2 디바이스 및 Type 3 디바이스

2. 데이터센터와 관련하여 미국통신산업협회(Telecommunication Industry Association) TIA-942 가이드라인에 대하여 다음을 설명하시오.

- (1) TIA-942 개요
- (2) TIA-942 품질등급

3. MIV(MPEG-Immersive Video)에 대하여 설명하시오

4. 리먼(Lehman)의 소프트웨어 진화 법칙에 대하여 설명하시오.

5. 정보시스템 하드웨어 규모산정지침(TTAK.KO-10.0292/R2)과 관련하여 다음을 설명하시오.

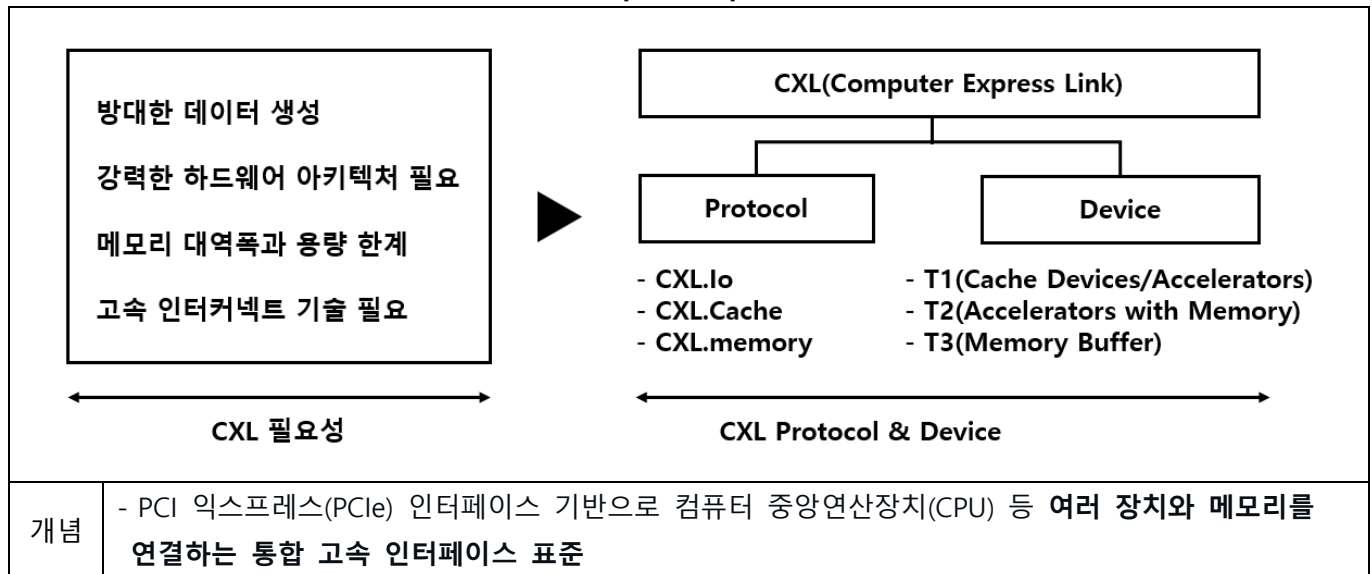
- (1) 규모산정 방법
- (2) 규모산정 대상
- (3) 규모산정 절차

6. 사용자 테스트와 관련하여 다음을 설명하시오.

- (1) 알파(Alpha) 테스트
- (2) 베타 테스트 (Beta Test)
- (3) 인수 테스트 (Acceptance Test)
- (4) 인수 테스트 프로세스(Acceptance Test Process)

01	CXL(Computer Express Link)		
문제	CXL(Computer Express Link)에 대하여 다음을 설명하시오. (1) CXL의 목적 (2) CXL의 3가지 프로토콜인 CXL.io, CXL.cache 및 CXL.memory (3) CXL의 디바이스 유형인 Type 1 디바이스, Type 2 디바이스 및 Type 3 디바이스		
도메인	CA	난이도	상(상/중/하)
키워드	CXL.io, CXL.cache, CXL.memory		
출제배경	최근 메모리 대역폭과 용량한계 극복을 위한 PCIe기반 CXL 부상에 따른 출제		
참고문헌	https://tech.gluesys.com/blog/2021/11/15/cxl_1.html https://semiconductor.samsung.com/kr/newsroom/tech-blog/expanding-the-limits-of-memory-bandwidth-and-density-samsungs-cxl-dram-memory-expander/ CCIX 연결망과 메모리 확장기술 동향(ETRI, 2022.02)		
해설자	소원반 소민호 기술사(제 119회 정보관리기술사 / mhsope@naver.com)		

I. PCIe를 대체할 인터커넥트 표준 기술, CXL(Computer Express Link)의 개요



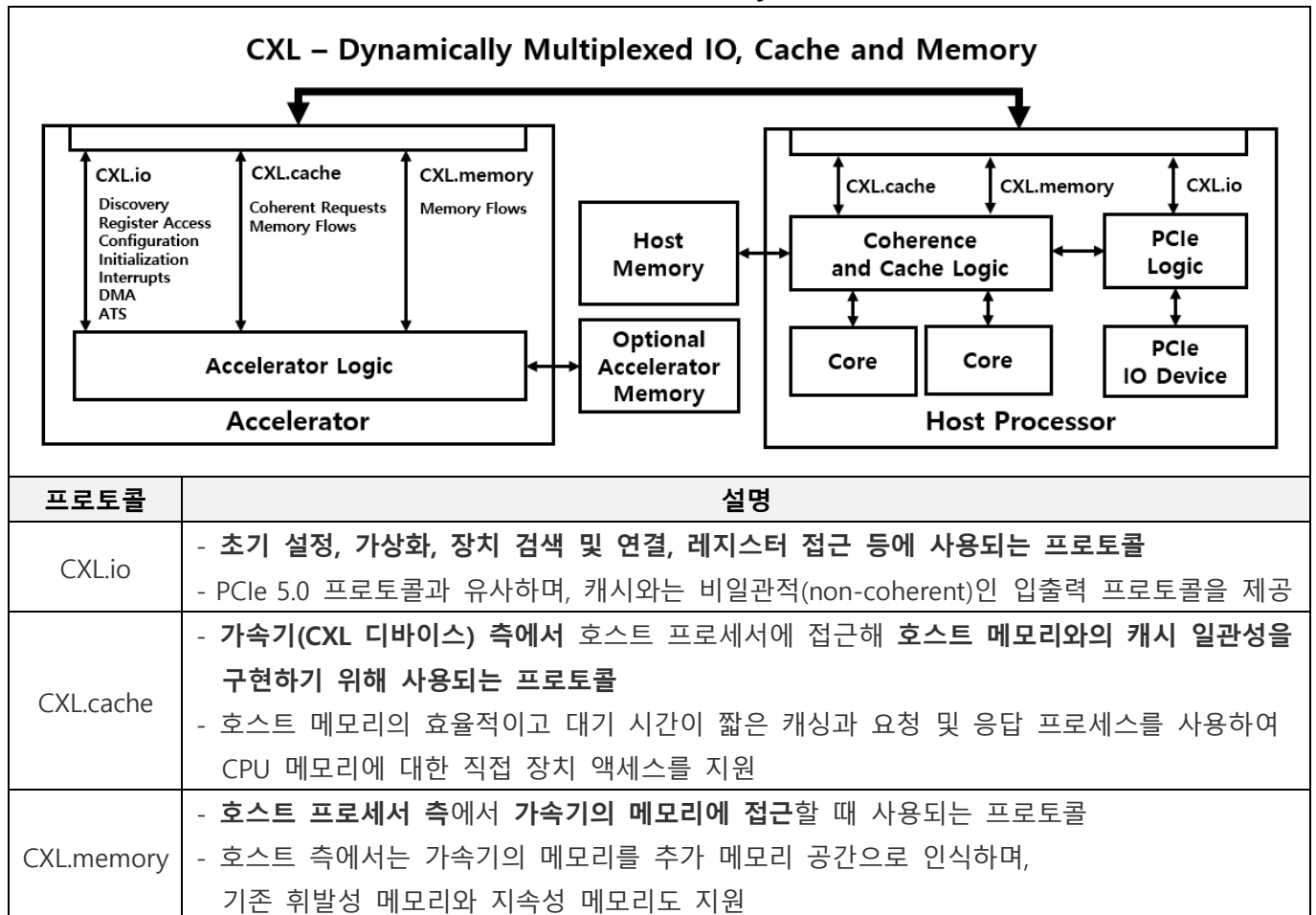
II. CXL의 목적 및 3가지 프로토콜인 CXL.io, CXL.cache 및 CXL.memory 설명

가. CXL(Computer Express Link)의 목적

목적	설명
메모리 용량한계 극복	- CPU 코어 수 증가에 따른 메모리 용량 한계가 존재
고속 데이터 처리	- 기존 시스템의 데이터 처리 지연과 속도 저하, 확장성 문제를 해결
고대역폭 메모리	- 호스트 프로세서와 CXL 디바이스 간에 저 지연 고대역폭 메모리 접근 환경 구성
메모리 일관성	- 고대역폭을 통해 액셀러레이터와 CPU 간의 메모리 일관성 유지
초저지연 컴퓨팅 구현	- 워크로드 집약적인 고성능 컴퓨팅 환경에서 프로세서와 가속기 간 메모리 공유를 통해 초저지연 컴퓨팅을 구현
범용성과 확장성	- 서버 내 메모리 리소스의 범용성과 확장성을 제공하는 기술에 대한 표준 사양 및 적용 방안 제시
범용성	- PCIe 5.0을 지원하는 모든 시스템에서 Flex Bus2라는 고속 포트를 통해 PCIe나 CXL 기능들을 제공
벤더 종속성 탈피	- 이전의 프로세서들은 메모리 공유 수단이 한정되고 벤더 간 호환성 문제 존재

- CXL은 디바이스 간 메모리 일관성을 유지하면서 리소스 공유를 통한 고성능, 소프트웨어 스택의 복잡성 완화 및 전반적인 시스템 비용 완화를 실현할 수 있도록 지원
- 호스트 CPU와 액셀러레이터 및 메모리 확장 디바이스와 같은 연결된 디바이스 간에 효율적인 연결을 제공

나. CXL의 3가지 프로토콜인 CXL.io, CXL.cache 및 CXL.memory 설명



III. CXL의 디바이스 유형인 Type 1 디바이스, Type 2 디바이스 및 Type 3 디바이스

가. CXL의 Type 1 디바이스 설명

구분	설명
개념	- 별도의 메모리를 보유하고 있지 않은 CXL 디바이스와 호스트 프로세서가 연결된 형태
개념도	<p style="text-align: center;">Caching Devices/Accelerators</p> <pre> graph LR subgraph Host P[Processor] D1[DDR] D2[DDR] end subgraph Device AN[Accelerator NIC] C[Cache] end P <--> CXL.io AN P <--> CXL.cache AN AN --- C </pre>
구성	<ul style="list-style-type: none"> - CXL 디바이스의 캐시에서 호스트 프로세서의 메모리와 일관성을 구현하는 형태 - 장치 검색 및 구성을 위한 CXL.io와 호스트 메모리 캐싱을 위한 CXL.cache 프로토콜 활용
프로토콜	- CXL.io, CXL.cache
활용	- 자체 메모리가 없는 스마트 NIC와 호스트 간의 구성

나. CXL의 Type 2 디바이스 설명

구분	설명
개념	- 메모리와 연결된 CXL 디바이스와 호스트 프로세서가 연결된 형태
개념도	<p style="text-align: center;">Accelerators with Memory</p> <pre> graph LR subgraph Host P[Processor] D1[DDR] D2[DDR] end subgraph Device A[Accelerator] C[Cache] H1[HBM] H2[HBM] end P <--> CXL.io A P <--> CXL.cache A P <--> CXL.memory A A --- C A --- H1 A --- H2 </pre>
구성	<ul style="list-style-type: none"> - 검색 및 구성을 위한 CXL.io를 포함해, CXL 디바이스와 호스트가 서로 메모리에 접근하기 위해 CXL.cache와 CXL.memory 프로토콜을 사용 - CXL.memory를 통해 호스트 CPU에서 확인하고 접근할 수 있는 로컬 주소 공간 존재
프로토콜	- CXL.io, CXL.cache, CXL.memory
활용	- 주로 GPU나 FPGA와 같은 가속기를 활용한 고성능 컴퓨팅이 요구되는 구성

다. CXL의 Type 3 디바이스 설명

구분	설명
개념	<ul style="list-style-type: none"> - 호스트 프로세서가 메모리 풀에 접근하는 형태 - 호스트 프로세서가 CXL.memory를 통해 CXL 디바이스 메모리 캐시에 일관되게 액세스할 수 있는 메모리 확장 디바이스
개념도	<p style="text-align: center;">Memory Buffers</p> <pre> graph LR subgraph Processor P[Processor] end subgraph Memory_Buffers [Memory Buffers] subgraph DDRs D1[DDR] D2[DDR] end subgraph MB [Memory Buffer] M1[Memory] M2[Memory] M3[Memory] M4[Memory] end end P <--> CXL.io MB P <--> CXL.memory MB P --- D1 P --- D2 MB --- M1 MB --- M2 MB --- M3 MB --- M4 </pre>
구성	<ul style="list-style-type: none"> - 기존의 가속기와 연결된 형태가 아니기 때문에 CXL.cache를 통한 요청은 없고, CPU에서 메모리 풀에 접근하기 위해 CXL.memory 프로토콜을 사용
프로토콜	- CXL.io, CXL.memory
활용	<ul style="list-style-type: none"> - 주로 호스트의 메모리 버퍼나 SCM과 연결된 구성 - 메모리 용량 및 메모리 대역폭 확장에 사용

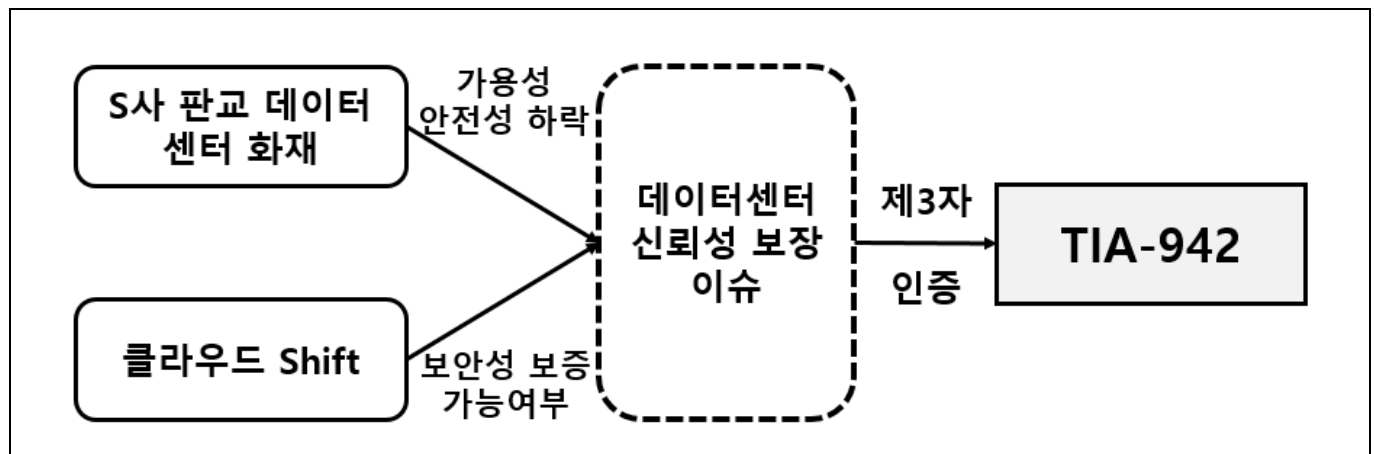
IV. 차세대 연결망 비교

비교항목	CXL	CCIX
토폴로지 지원	p2p	p2p, mesh, ring, etc.
대역폭	~128GB/s (x16)	~128GB/s (x16)
물리 계층	PCIe 5.0 PHY	PCIe 5.0 PHY
캐시 일관성 지원 여부	0	0
개발 주도 기업	Intel	ARM, Xilinx
구조	비대칭형 CCI 구조	대칭형 CCI(Cache Coherent Interconnect) 구조

“끝”

02	TIA-942 가이드라인		
문제	데이터센터와 관련하여 미국통신산업협회(Telecommunication Industry Association) TIA-942 가이드라인에 대하여 다음을 설명하시오. (1) TIA-942 개요 (2) TIA-942 품질등급		
도메인	컴퓨터 구조	난이도	상(상/중/하)
키워드	Tier1, Tier2, Tier3, Tier4, CAB		
출제배경	최근 S사 데이터센터 화재 및 공공 클라우드 전환에 따른 안전성·보안성 인증 관련 해외 표준 지식 확인		
참고문헌	미국통신산업협회(Telecommunication Industry Association) ITA-942 가이드라인		
해설자	강남평일야간반 전일 기술사(제 114회 정보관리기술사 / nikki6@hanmail.net)		

I. 최근 데이터센터의 안전성 이슈 배경



- 최근 S사 판교 데이터 센터 화재 및 공공 클라우드 전환에 따른 데이터센터의 신뢰성 보장을 위한 국제 표준 검토 진행

II. 데이터센터 관련 인프라 안전 표준, TIA-942 개요

가. TIA(Telecommunication Industry Association)-942 개념

- 데이터 센터 인프라에 대한 최소 요구 사항을 지정하는 미국을 포함한 전세계 국가 표준
- TIA-942는 법적 효력이 있는 국제 표준과 같은 강제 사항은 아니지만 데이터센터에서 갖춰야할 제반 요소들의 기준을 규정하는 준거들을 명시

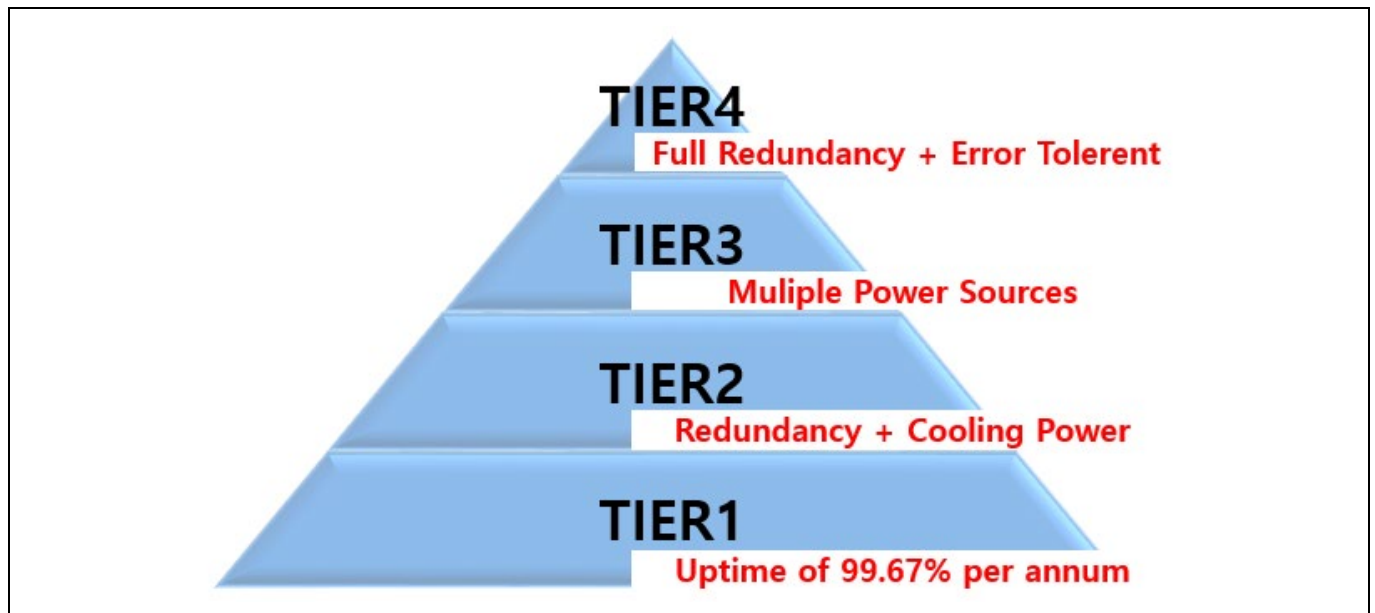
나. TIA 특징

항목	세부내용
선택사항	- 법적 효력이 있는 국제 표준과 같은 강제 사항은 아니지만 데이터센터에서 갖춰야할 제반 요소들의 기준을 규정하는 준거들을 명시
감사 증적	- TIA의 공식 인증을 통해 데이터 센터 운영자들은 내부 및 외부 최종 사용자들에게 각 데이터 센터들이 해당 비즈니스에서 요구되는 목표 레실리언스 등급을 충족한다는 사항에 대해 독립 감사를 받았다는 것을 증명
제3자 검증	- 데이터 센터를 검증할 수 있는 적합성 평가 기관(CAB: Conformity Assessment Body)을 확립할 예정

- TIA-942가 발표되기 이전에는 TIA-568(Commercial Building Telecommunications Wiring Standard) 이 대표적인 상업용 건물을 위한 네트워크 인프라의 표준으로 주로 사용

III. TIA-942 품질 등급 상세

가. TIA-942 품질 등급



나. TIA-942 품질 등급 상세 설명

항목	Tier I	TierII	TierIII	TierIV
데이터센터 가용성	99.67%	99.75%	99.98%	100.00%
연간 장애발생 시간	28.8	22	1.6	0.4
전력 및 냉방시설 이중화	N	N+1	N+1	2(N+1) or S+S
			동시 활성화	무 정지 상태
운영센터	필요치 않음	필요치 않음	필요함	필요함
보안시설 (로비에서 전산실까지)	일반 잠금장치	카드인식	생체인식	생체인식
백본 이중화	필요치 않음	필요치 않음	필요함	필요함
수평케이블링 이중화	No	No	No	(선택사항)
라우터 및 스위치 이중화	No	No	Yes	Yes
이중화된 액세스 프로바이더	No	No	Yes	Yes
2차출입 통제소	No	Yes	Yes	Yes
패치코드 꼬리표 장착	No	Yes	Yes	Yes
랙/케비넷의 지지대	No	바닥 지지대	풀 지지대	풀 지지대
데이터센터 기반 인프라	필요치 않음	필요치 않음	Yes	Yes

- TIA는 데이터 센터의 설계 디자인과 구현 가이드라인을 기준으로 4개 계층(Tier)으로 구성

IV. TIA-942 표준화 동향

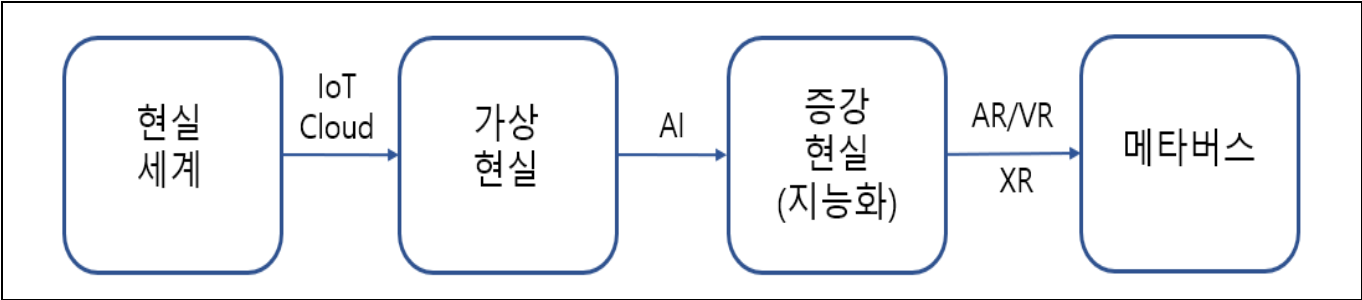
표준	승인 및 변경 사항
TIA-942-B	- 2017년 TIA-942-B 데이터센터 케이블링 표준(Data Center Cabling Standard) 개정을 승인
TIA-942-A	- 데이터센터 패브릭을 처리하는 TIA-942-A(Add)1을 통합 - 균형 꼬임 쌍선(Twisted pair cable)의 허용 유형으로 카테고리 8을 추가하고 카테고리 6A 균형 꼬임 쌍선의 권장 사항을 카테고리 6A와 그 이상으로 변경

- TIA-942 표준은 BICSI(Building Industry Consulting Services International)-002와 관리 표준인 TIA-606과 주로 함께 사용됨

“끝”

03	MPEG-Immersive Video		
문제	MIV(MPEG-Immersive Video)에 대하여 설명하시오		
도메인	디지털서비스	난이도	상(상/중/하)
키워드	TMIV 인코더, TMIV 디코더, 아틀라스, 텍스처, 기하정보, 현장감, 몰입감		
출제배경	메타버스를 위한 XR 비디오 표준 관련 확인		
참고문헌	몰입형 비디오 압축을 위한 스크린 콘텐츠 코딩 성능 분석_방송공학회논문지 2020.11 MIV 표준기술 개발 동향_주간기술동향 2022.3.1		
해설자	정상 기술사(제 12X회 정보관리기술사 / jeongsang_pe@naver.com)		

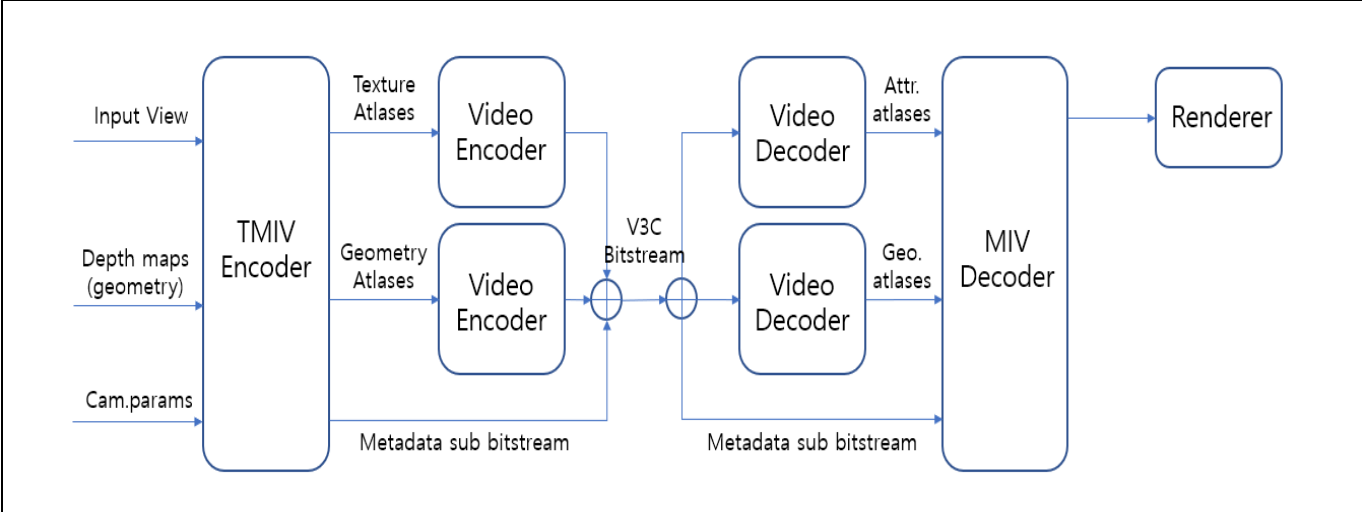
I. 메타버스 구현을 위한 MIV 기술 부각 배경



- 최근CG 기반의 게임이나 비현실적 공간으로 표현되는 메타버스에 현실 공간을 표현 하기 위한 기술 개발의 필요성이 대두됨에 따라 실사 기반으로 현실감과 몰입감을 동시에 지원할 수 있는 이머시브 비디오 기술의 활용도가 지속적으로 증대

II. MIV(MPEG-Immersive Video)의 개념과 구성요소

가. MIV(MPEG-Immersive Video)의 개념



개념	- 전통적인 비디오 코딩에서의 인코딩 효율 향상과 더불어 입출력 인터페이스, 압축 처리를 위한 코덱의 개수 등을 나타내는 픽셀율(pixel rate) 증가를 억제하고 비교적 적은 수의 2D 비디오 코덱을 통해 사용자의 인터랙션에 따라 최적의 주관적 화질을 재현한 기술
----	---

나. MIV(MPEG-Immersive Video)의 구성요소

구분	핵심 기술	설명
Encoder	- TMIV Encoder	- 다수의 텍스처(Texture) 및 기하정보(Geometry)를 이용하여 공간상의 중복성을 제거한 후 아틀라스(atlas)라고 불리워지는 새로운 인코딩 포맷을 생성
	- Video Encoder	- 아틀라스(atlas)를 기반으로 HEVC 또는 VVC 비디오 코덱을 통해 압축 전송이 가능함
Decoder	- MIV Decoder	- 후처리와 중간시점 영상 합성을 통해 시청자의 움직임에 따라 동적으로 끊어짐 없이 뷰포트(viewport)영상을 재생 - 메타데이터와 비트스트림 구조가 주로 표준화됨
	- Video Decoder	- 비트 스트림 데이터를 영상 디코더로 변환하여 아틀라스(atlas) 형식으로 MIV decoder로 전달

- MIV는 부복호화 표준 기술 및 TMIV 인코더 기술 존재

III. 이머시브 비디오 부호화를 위한 MIV 표준기술

가. MIV 부복호화 표준 기술

구분	핵심 기술	설명
인코더	- 푸루닝(pruning)	- 기하정보를 이요해서 모든 소스 시점의 픽셀들을 3D 공간상으로 역투영, 중복성 제거 하는 방식 - 깊이정보만 정확하다면, 단말에서 뷰포터 렌더링에 필요한 픽셀만 추출
	- 패치 패킹	- 다수의 시점영상들을 기하정보들을 통해 공간상에 매핑하고 소스 카메라의 위치에서 보여지는 픽셀들을 패치 형태로 추출, 아틀라스 영상포맷으로 변환
단말측	- 아틀라스 패치 점유지도	- 메타데이터를 파싱하여 픽셀별로의 패치정보를 기록
	- 랜더러(Renderer)	- 디코딩되어 분리된 패치들로부터 시청자가 바라보는 방향에 해당 하는 중간시점의 영상을 합성, 뷰포트 영상으로 제공

- 부복호화 표준기술에서 세부적인 TMIV 인코더의 기술로 아틀라스 생성

나. MIV의 TMIV 인코더 기술

구분	핵심 기술	설명
입력	- 속성정보	- 기하정보와 칼라정보, 투명 정보등으로 구성
	- 엔티티 맵(entity map)	- 객체기반의 부호화를 위해 필요
	- 시점파라미터(View parameter)	- 내부/외부 카메라파라미터 정보로 구성
인코더	- 그룹 인코더	- 전방위 360 공간에서 획득된 텍스처와 기하정보를 공간적으로 분할하고 공간별로 인코딩하여 아틀라스 영상 생성 후, 각각의 비디오 인코딩 진행
	- 인페인팅(inpainting)	- 다시점 영상 합성에서 난해한 과정의 하나인 폐색영역을 채우는 기술
	- 모드(mode)	- MIV 모드 : 푸루닝을 허용한 패치들을 포함 - MIV 뷰(View) 모드 : 푸루닝 연산 없이 모든 시점영상으로부터 공간 영상 재생에 필요한 몇 개의 기본시점을 선택하고 인페인팅된 배경과 함께 패킹

- MIV 기술 표준은 지속적으로 현장감 , 몰입감 확보를 위해 진화 중임

IV. MIV 기술 표준화 동향

구분	설명
2022	- MIV IS의 발간을 목표로 적합성 시험 진행 중, 검증 시험 진행 - MIV edition-2를 위한 유저케이스 및 요구사항을 작성
2023	- 7월까지 MIV edition-2의 표준화를 완료, FDIS를 발간 예정 - MIV 콘텐츠 합성 및 부호화 기술을 포함하여 이종 객체미디어를 MIV를 통해 제공

- 현재 메타버스는 CG 위주로의 비디오가 제공되고 있지만, 다양한 환경에서 제작된 실사, CG 모델링 객체 등이 MIV edition-2를 통해 제공된다면 다양한 비즈니스 모델 개발이 활성화 될 것으로 기대

“끝”

04	리먼(Lehman)의 소프트웨어 진화 법칙		
문제	리먼(Lehman)의 소프트웨어 진화 법칙에 대하여 설명하시오.		
도메인	소프트웨어공학	난이도	하(상/중/하)
키워드	계속변경, 자가규제, 조직적안정화, 친근성유지, 피드백시스템, 지속적성장, 증가복잡도, 품질감소		
출제배경	시스템 내 소프트웨어 중요도 증가에 따른 소프트웨어 관리의 효율화를 위한 관련 법칙에 대한 지식 점검		
참고문헌	ITPE 기술사회 자료		
해설자	서경석 기술사(제119회 정보관리기술사 / akslemf@naver.com)		

I. S/W 변화 관리와 유지 보수의 핵심, 리먼의 소프트웨어 진화 법칙 개요

가. 리먼의 소프트웨어 진화 법칙의 정의

- 소프트웨어는 요구에 의해 계속적으로 변경되며, 변경에 따른 복잡성, 프로그램의 고유한 변경 추세, SW조직 생산성의 일관성, 소프트웨어 각 버전의 변화에 대한 일관성을 제시한 SW 변화의 원리

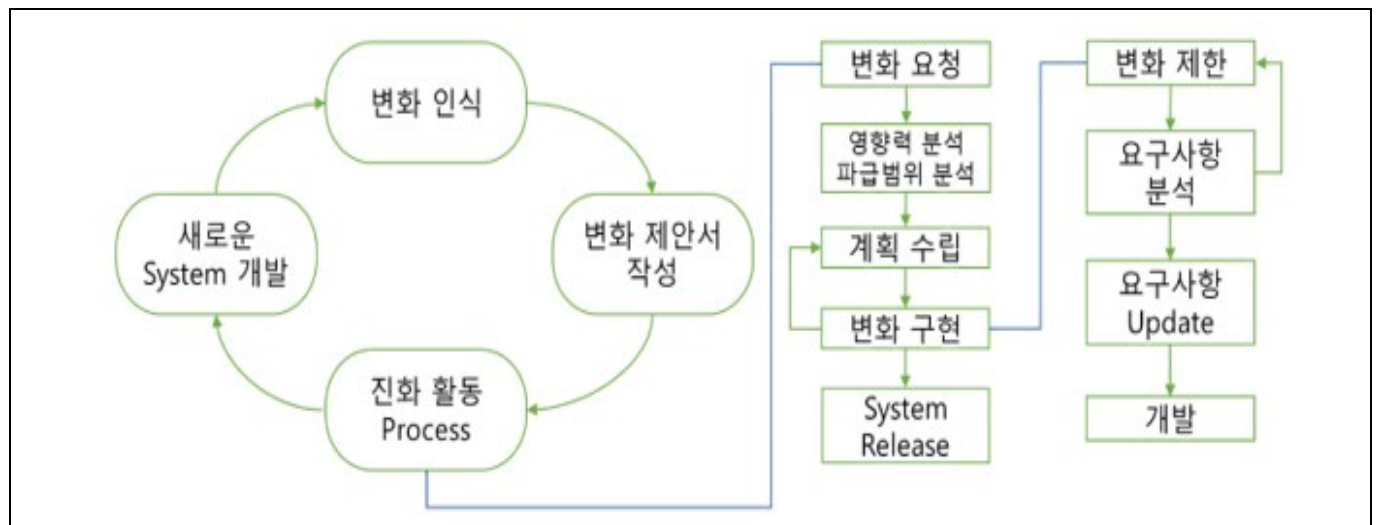
나. 리먼의 소프트웨어 진화 법칙의 분류

구분	관점	내용
S-type(Static)	- Specification And Solutions	- 정해진 명세에 따라 정확히 동작하는 S/W(계산기 프로그램)
P-type(Practical)	- Procedures	- 절차나 실행 입력 값에 따라 다르게 수행 되는 S/W(체스 게임)
E-type(embedded)	- Environment	- 실 세계의 환경적 상황과 밀접하게 연관되어 동작하는 S/W (Online trading Software, 세금 관련 시스템)

- 리먼은 E-type 소프트웨어에 대해 진화 법칙을 제시

II. 리먼의 소프트웨어 진화 법칙의 개념도와 상세 설명

가. 리먼의 소프트웨어 진화 법칙의 개념도



- 리먼의 소프트웨어 진화 법칙은 사용자 요구에 따른 SW의 계속적 변경 시, 변화 및 유지 관리 위해 활용

나. 리먼의 소프트웨어 진화 법칙의 원리 상세 설명

구분	세부내용	적용 방안
계속적 변경 (Continuing change)	<ul style="list-style-type: none"> - 소프트웨어는 계속 진화하며 요구사항에 의해 계속적으로 변경됨 - 소프트웨어는 자체적으로 갱신 불가하며 인간의 의지의 개입이 필요 - 다시 만드는 것보다 더 경제적이라고 판단되는 한 계속 변화 	<ul style="list-style-type: none"> - 요구관리프로세스 - 변경관리프로세스
복잡도 증가 (Increasing complexity)	<ul style="list-style-type: none"> - 변경이 가해질수록 구조는 복잡해짐 - 시스템의 복잡도는 이를 그대로 유지하거나 줄이고자 하는 특별한 작업을 하지 않는 한 계속 증가된다. 	- 3R
자가 규제 (Self-Regulation)	<ul style="list-style-type: none"> - 프로그램 별로 변경되는 사항은 고유한 패턴/추세가 있음 - 복잡성을 단순화 시키려는 인간 의지의 개입 	<ul style="list-style-type: none"> - 통제 프로세스 - 변경 영향도 분석
조직의 안정화 (Organizational Stability)	<ul style="list-style-type: none"> - 조직의 생산성이 조직 변화에 민감하지 않음 - 개인의 생산성 최적화가 팀의 생산성을 최적화 하는데 필수적이지 않음 	<ul style="list-style-type: none"> - 변경통제위원회(CAB) - 변경통제위원회(CCB)
친근성의 유지 (Conservation of familiarity)	<ul style="list-style-type: none"> - 소프트웨어의 각 버전의 변화는 일정함 - 소프트웨어는 규칙적인 수행결과와 추이를 보여주기 때문에 예측 가능 	<ul style="list-style-type: none"> - 형상관리시스템 - SVN, Git
지속적 성장 (Continuing Growth)	<ul style="list-style-type: none"> - 소프트웨어의 Lifetime 내내, 기능 내용은 사용자 만족도를 유지하기 위해 증가가 계속됨 	<ul style="list-style-type: none"> - CI/CD - DevOps
감소하는 품질 Declining Quality	<ul style="list-style-type: none"> - 소프트웨어는 엄격하게 관리 및 운영되지 않거나, 환경 변화에 적응 하지 않는다면 품질은 감소함(1996년 추가) 	<ul style="list-style-type: none"> - 리팩토링 - 회귀테스트
피드백 시스템 (Feedback System)	<ul style="list-style-type: none"> - 진화 프로세스는 다중레벨, 다중루프, Multi-Agent 피드백 시스템 수용. 중요한 제품개선 달성 위해 소프트웨어 피드백이 구성되어야 함 (1996년추가) 	<ul style="list-style-type: none"> - Baseline - CMDB

III. 리먼의 소프트웨어 진화 법칙에 따른 SW 라이프사이클 연장을 위한 유지 보수 분류

분류	세부	설명
목적	수정적 유지 보수	- 오류와 결함의 수정(corrective maintenance)
	완전적 유지 보수	- 기능 개선 (perfective maintenance)
	예방적 유지 보수	- 정기적인 유지보수(preventive maintenance)
	적응적 유지 보수	- 변화, 갱신의 적용(adaptive/porting maintenance)
대상	데이터 유지보수	- 데이터의 Conversion 등의 필요시 처리
	프로그램 유지보수	- 프로그램의 변경 및 오류에 대한 처리
	문서 유지보수	- 문서 표준의 변경이나 기타 필요시
	시스템 유지보수	- 시스템의 변경 및 장애에 대한 처리
시점	계획 유지보수	- 주기적인 유지보수
	예방 유지보수	- 미리 예방 차원에서의 유지보수
	응급 유지보수	- 긴급한 경우의 유지보수, 사후 승인 필요
	지연 유지보수	- 시스템에 대해 변경된 부분에 대한 추후 지원

- 소프트웨어 유지 보수 진행 시 목적, 대상, 시점 기준 적절한 관리 통한 SW 라이프사이클 효율성 확보

“끝”

05	하드웨어 규모산정지침		
문제	"정보시스템 하드웨어 규모산정지침(TTAK.KO-10.0292/R2)"과 관련하여 다음을 설명하시오. (1) 규모산정 방법 (2) 규모산정 대상 (3) 규모산정 절차		
도메인	소프트웨어공학	난이도	하(상/중/하)
키워드	수식계산법, 참조법, 시뮬레이션법, OLTP, WEB/WAS		
출제배경	컴퓨터시스템응용 126회 1교시 확장문제		
참고문헌	정보시스템 하드웨어 규모산정 지침(TTA.KO-10.0292/R2), ITPE 126회 기출풀이		
해설자	이상용 기술사(제 124회 정보관리기술사 / orangeday77@gmail.com)		

I. 정보화 투자 효율성, 하드웨어 규모 산정의 개념

구분	개념
용량관리	- 비즈니스 요구사항을 충족시키기 위한 현재와 미래의 용량 계획을 수립하고 비용과 용량의 균형을 맞추는 것
용량계획	- 개략적인 시스템 아키텍처와 응용 업무를 기반으로 시스템에 요구되는 성능 요구사항과 성능을 결정하기 위한 계획
규모산정	- 기본적인 용량과 성능 요구사항이 제시되었을 때, 그것을 시스템 요구사항으로 변환하는 것
하드웨어 규모산정	- 정보시스템 도입 시 기본적인 용량과 성능 요구사항이 제시되었을 때 그것을 시스템 도입을 위한 요구사항으로 변환하기 위한 산정하는 방법

II. 하드웨어 규모산정 방법

구분	개념	설명
수식 계산법	- 사용자 수 등 규모산정을 위한 요소를 토대로 용량 수치를 계산하고, 보정치를 적용하는 방법	- (장점) 규모산정의 근거를 명확하게 제시할 수 있으며, 다른 방법에 비해 간단하게 산정할 수 있음 - (단점) 보정치가 잘못되었을 경우 원하는 값과 많은 차이가 발생하며, 보정치에 대한 정확한 근거 자료 제시가 어려움
	예시	
	- CPU : 분당 트랜잭션 수 * 기본 tpmC 보정 * 피크타임 부하 보정 * DB 크기 보정 * 어플리케이션 구조 보정 * 어플리케이션 부하 보정 * 클러스터 보정 * 시스템여유율 - 메모리 : {시스템 영역 + (사용자당 필요메모리 * 사용자수)} * 버퍼캐쉬 보정 * 시스템 여유율 - 디스크 : 시스템디스크 = (시스템OS 영역 + 응용프로그램 영역 + SWAP 영역) * 파일시스템 오버헤드 * 시스템디스크 여유율 - 데이터디스크 = (데이터 영역 + 백업 영역) * 파일시스템 오버헤드 * RAID 여유율 * 데이터디스크 여유율	

참조법	<ul style="list-style-type: none"> - 업무량(사용자 수, DB 크기)에 따라, 기본 데이터를 토대로 대략적인 시스템 규모를 비교하여 비슷한 규모를 산정하는 방법 	<ul style="list-style-type: none"> - (장점) 기존 구축되어 있는 업무 시스템과 비교가 가능하므로 비교적 안전한 규모 산정 가능 - (단점) 계산에 의한 방법이 아닌 비교에 의한 것이므로 근거 제시 미약
	예시	
	<p>[경영지원-1,000명] CPU: 2P, 8Core Memory: 36GB Disk: 1TB</p> <p>500명 감원</p> <p>[경영지원-500명] CPU: 1P, 4Core Memory: 24GB Disk: 500GB</p> <p>- CPU: 40% (적정, 유지) - Memory: 70% (부족, 증설필요) - Disk: 400GB (여유, 축소 혹은 유지)</p> <p>- 1,000명 규모의 경영지원 시스템을 500명 규모의 경영지원 구축 시 참조모델로 사용</p>	
시뮬레이션법	<ul style="list-style-type: none"> - 대상 업무에 대한 작업부하를 모델링하고 이를 시뮬레이션 하여 규모를 산정하는 방법 	<ul style="list-style-type: none"> - (장점) 상대적으로 정확한 값을 얻을 수 있음 - (단점) 시간과 비용이 많이 소요
	예시	
	<p>[시뮬레이션 툴]</p> <p>H/W 사양 입력 사용자 수 용도 선택</p> <p>{ Web/WAS 서버 (파트너 관리 시스템) }</p> <p>{ CPU: 2P, 8Core Memory: 36GB Disk: 1TB }</p> <p>{ 전체사용자: 2,000명 동시사용자수: 300명 }</p> <p>[파트너관리 WAS] CPU: 1P, 6Core Memory: 24GB Disk: 500GB</p>	

- 국내 공공 부분 등에 적용하는 지침은 사업 입안 단계에서 규모산정이 가능 해야 하기에, 수식 계산법 적용되나, 산정 결과의 정확성과 객관성을 높이기 위해 이들 세가지를 혼합하여 사용 해야함

III. 하드웨어 규모산정 대상 설명

가. 하드웨어 규모산정 개념도

서버	CPU		시스템유형	
	메모리		OLTP	WEB/WAS
	디스크	시스템	○	○
		데이터	○	
			○	
스토리지		○	○	

구분	CPU			스토리지
	OLTP 또는 OLTP&배치 애플리케이션 서버	WEB 서버	WAS 서버	
성능 측정치	tpmC	max-jOPS ^a		IOPS
참조 성능 기준	TPC-C	SPECjbb2015		SPC-1

- 본 지침에서의 규모 산정 대상은 하드웨어로 PC나 기타 주변 장비가 아닌 서버 장비로 규정

나. 하드웨어 규모산정 대상 상세 설명

대상	산정 기준	상세 설명
CPU	- 해당 업무를 처리하기 위한 CPU 규모를 계산한 후, 적절한 성능을 지닌 서버기종 선정	- 산정 대상 시스템이 WEB이나 WAS 서버로 쓰이는 경우 CPU 산정방식 중 WEB/WAS 산정 기준을 적용하고, DB서버로 쓰이는 경우에는 OLTP 또는 OLTP & Batch 애플리케이션 산정 기준을 적용
메모리	- CPU 규모 산정에 따른 서버 구성 방안에 의거하여, 서버 별 시스템 S/W, 응용프로그램 등의 메모리 사용량 산정	- 구축하는 시스템별로 프로그래밍 언어나 스레드(thread)의 사용 등 다양한 방법에 의해 메모리 점유를 줄이기 위한 전략을 사용
디스크	- CPU 규모 산정에 따른 서버 구성 방안에 의거하여, 서버 별 OS, 시스템 S/W, DB의 데이터, DB의 아카이브(Archive) 및 백업 영역 등의 디스크 사용량 산정	- 디스크 규모 산정 시 가장 중요한 고려 요소는 데이터 백업 방안 - 백업 정책에 의해 디스크 요구량은 큰 차이를 가지기 때문에 데이터의 중요도를 고려하여 상황에 적절한 백업 정책을 수립할 필요
스토리지	- CPU 기준으로 산정된 서버 규모에 따라 필요한 스토리지의 규모 산정	- 소프트웨어 파트: 장치 관리, 볼륨 관리, 원시 파일(raw) 및 파일 시스템(file system) - 관리 소프트웨어의 운영 파트: 서버 하드웨어 플랫폼(server) 그리고 HBA(host bus adaptor) 및 파이버 채널(Fibre channel)과 같은 인터페이스 - 스토리지 장치: 캐시, 컨트롤러, 저장 장치

- 이러한 하드웨어 구성 분야는 여러 가지가 있지만 시스템 가격 및 성능 측면에서 가장 중요한 CPU, 메모리,

디스크, 그리고 스토리지 등 4가지 분야를 규모 산정 분야로 정의

IV. 하드웨어 규모산정 절차 설명

가. 하드웨어 규모 산정 절차도



- 4단계로 하드웨어 규모 산정을 진행함

나. 하드웨어 규모 산정 절차 상세설명

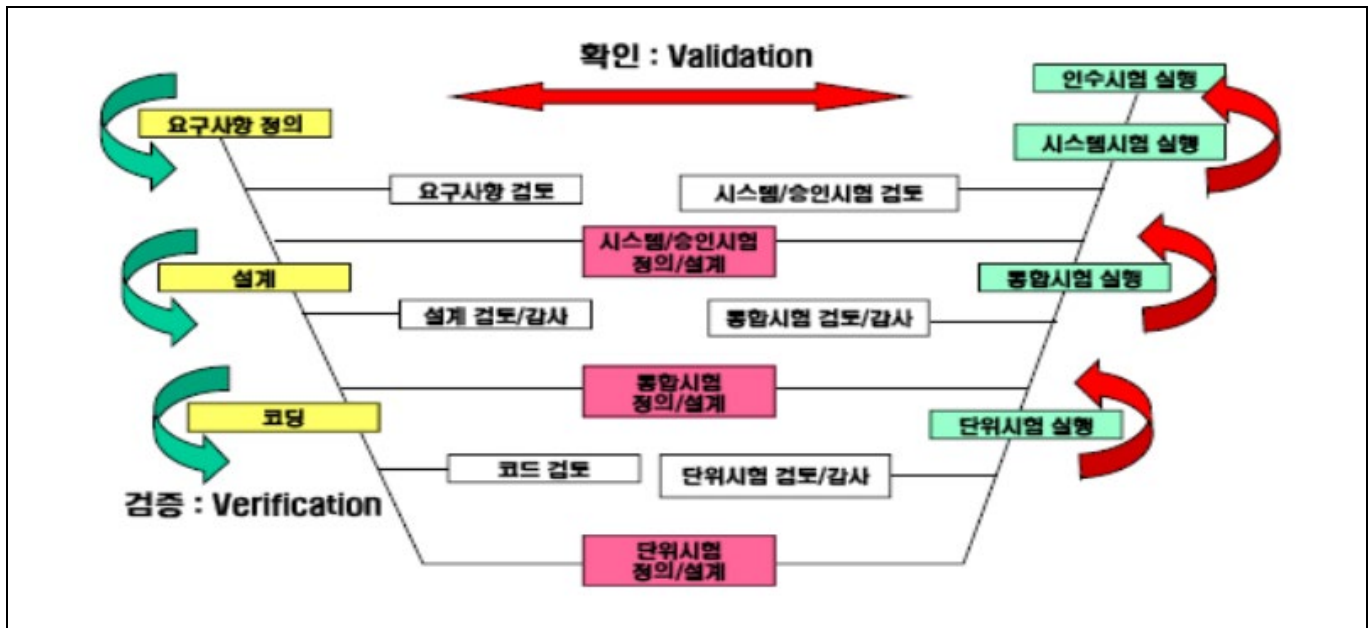
절차	활동	설명
[1단계] 구축방향 및 기초자료 조사	구축 방향 파악	- 전체 시스템에 포함되는 대략적 서버 수, 통신 환경 파악 - APP 아키텍처(2-계층, 3-계층), 업무 성격, 정보 흐름 파악
	기초자료 조사	- 공통/OLTP: 응답속도, 최대 사용자, 업무 내용 등 조사 - WEB/WAS: 시스템 용도, 서비스 형태, 사용자 수 등 조사
[2단계] 기초자료 및 업무분석	기준 부하 산정	- OLTP: 동시 사용자 수 및 분당 트랜잭션 수(TPM) 산정 - WEB/WAS: 동시 사용자 및 세션 수, 단위 오퍼레이션 산정
	업무 내용 검증	- 비즈니스 요구사항 포함, 트랜잭션 타입, 특성, 가중치 조사 - 온라인/배치 구분, 요구 시간, 처리 볼륨, 복잡성 분석
[3단계] 참조모델 결정 및 서버 규모산정	참조 모델 선택	- 참조모델1(WEB/WAS/OLTP): 단일 서버에서 모든 계층 처리 - 참조모델2(WEB/WAS, OLTP): 응용 계층과 DB 계층 분리 - 참조모델3(WEB, WAS, OLTP): 각 응용 계층과 DB 계층 분리
	요소별 규모산정	- 2단계에서 조사된 업무 분석 자료 기반 보정 계수 설정 - CPU, 메모리, 디스크 구성요소에 대해 각각 규모 산정 수행
[4단계] 참조모델 별 가중치 적용	참조모델1 가중치 적용	- WEB/WAS/DB 역할을 동시에 수행하므로 서버의 CPU 규모에 - 상대적 서버 가중치 2.1(WEB:0.4, WAS:0.7, DB:1) 적용
	참조모델2 가중치 적용	- WEB/응용서버, DB서버: WEB/응용서버의 경우 WEB, WAS의 역할을 동시에 수행해야 하므로 1.6(WEB:0.6, WAS:1) 적용 - WEB서버, 응용/DB서버: 응용/DB서버의 경우 응용과 DB서버 의 역할을 동시에 수행해야 하므로 1.7(응용:0.7, DB:1) 적용
	참조모델3 가중치 적용	- 별도의 서버 가중치 미적용 - WEB: 1, WAS: 1, DB: 1

- 일반적으로 규모 산정의 정확도는 규모 산정을 위한 기초 자료 확보, 표준화된 계산식과 절차, 산정 결과의 데이터베이스 저장 및 재활용 등의 4가지 요소에 의해서 결정

“끝”

06	알파 테스트, 베타 테스트, 인수 테스트		
문제	6. 사용자 테스트와 관련하여 다음을 설명하시오. (1) 알파(Alpha) 테스트 (2) 베타 테스트 (Beta Test) (3) 인수 테스트 (Acceptance Test) (4) 인수 테스트 프로세스(Acceptance Test Process)		
도메인	SW공학	난이도	중(상/중/하)
키워드	알파테스트, 베타테스트, 사용자 인수테스트, 운영상의 인수테스트, 계약 인수테스트, 규정 인수테스트		
출제배경	일반적인 이해하고 있는 인수테스트에 대하여 잘 숙지 되고 있는지 확인		
참고문헌	개발자도 알아야 할 소프트웨어 테스트 실무, 정부정보자원관리 적용가이드		
해설자	장건환 기술사(제 126회 정보관리기술사 / jkh556@naver.com)		

I. V-Model에서의 인수테스트



- (개념) 요구사항에 대한 Validation(확인), Verification(검증) 단계를 거쳐 요구조건에 부합하는지 확인하는 테스트
- 인수테스트의 비공식적인 테스트 과정 중 비정형화된 방식으로 테스트하는 알파테스트와 베타테스트를 수행함으로써 예측 못한 결함을 발견 가능

II. 알파테스트와 베타테스트

가. 개발 조직내 고객 수행, 알파테스트

구분	설명	
정의	- 선택된 사용자에게 의해 개발 환경에서 수행되는 테스트 (공장 인수 테스트)	
세부설명	환경	- 개발환경에서 테스트
	목표	- 결함 발견, 부하검사, SW 신뢰도 향상,
	시점	- 시스템 테스트 단계 이후 또는 제품의 70~90% 완료 시기
	기간	- 많은 테스트 주기 수행으로 1-2주 동안 지속 (결함/문제 수에 따라 기간 조율)
	참가자	- 개발자, 테스터, 고객 또는 최종 사용자

나. 실 환경에서 고객 수행, 베타테스트

구분	설명	
정의	- 실 환경에서 사용자 혹은 잠재 고객에 의해 수행되는 테스트 (사이트 인수 테스트)	
세부설명	환경	- 사용자 환경에서 테스트
	목표	- 사용자에게 의한 제품 평가, 사용자 조건에 의한 테스트
	시점	- 알파 테스트 후 90~95% 완료 시기
	기간	- 1회 or 2회 주기로 약 4-6주 동안 지속 (피드백/제안에 따라 조율)
	참가자	- 고객 또는 최종 사용자

III. 개발된 시스템의 인수결정을 위한 확인과정, 인수테스트 개요

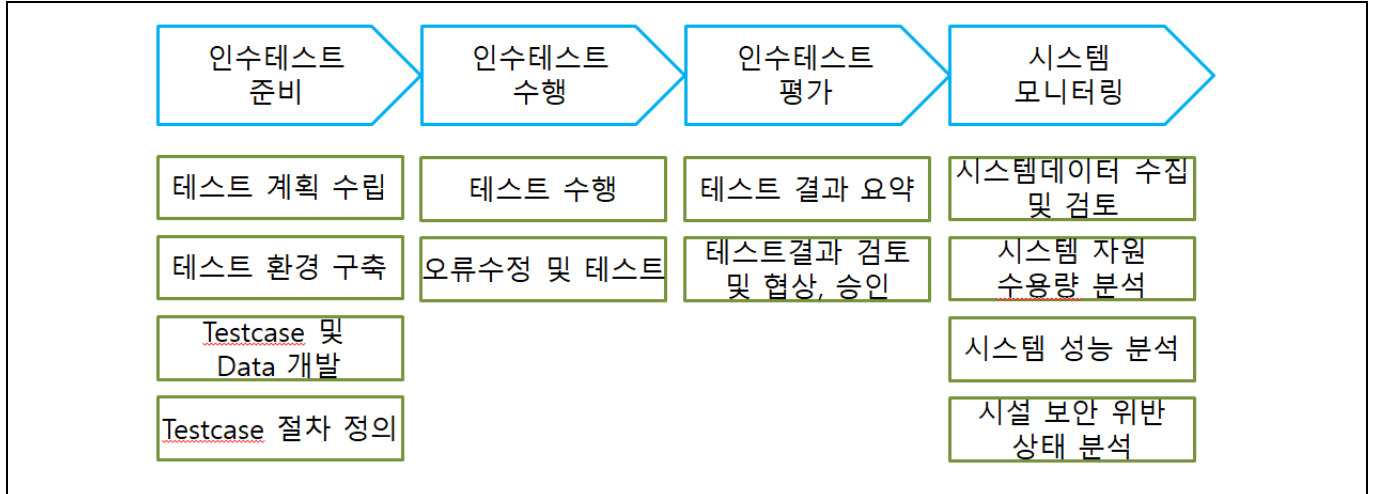
가. 인수테스트 설명

구분	설명		
정의	- 시스템이 인수조건을 만족시키는지, 사용자/고객 또는 다른 인가된 개체가 시스템을 인수할지 결정할 수 있도록 사용자의 필요, 요구, 비즈니스 프로세스를 고려하여 수행하는 테스트		
목적	확신	- 시스템이나 시스템의 일부 또는 특정한 비기능적인 특성에 대해 “확신”을 얻는 것	
	배포가능성 평가	- 결함을 찾는 것이 아닌, 시스템을 배포하거나 사용할만한 준비가 되었는지에 대한 평가	
	준수성 확인	- 계약 또는 규정에 대한 준수성을 확인함	
	고객 피드백	- 판매 또는 상용(COTS) 소프트웨어는 목표시장의 기존 고객이나 잠재적인 고객으로부터 피드백을 위해 실시함	
유형	공식적인 인수테스트	사용자 인수테스트	- 일반적으로 비즈니스 사용자가 시스템 사용의 적절성을 확인
		운영상의 인수테스트	- 시스템 관리자에 의한 시스템의 인수 시 수행하는 테스트 활동 - 백업/복원 테스트, 재난복구, 보안 취약성 점검
		계약 인수테스트	- 맞춤형-개발(Custom-developed) 소프트웨어가 계약상의 인수 통과 조건을 준수하는 확인
		규정 인수테스트	- 정부의 지침, 법률 또는 안전 규정 등 준수해야 하는 규정에 맞게 개발되었는지 확인

	비공식적인 인수테스트	알파 테스트	- 선택된 사용자에게 의한 개발 환경에서의 테스트
		베타 테스트	- 사용자의 사용환경에서 진행되는 테스트 - 사용자가 모든 문제를 기록하여 보고함 - 상용소프트웨어가 상업적으로 판매되기 전에 목표 시장의 기존 고객이나 잠재적인 고객으로부터 피드백을 목적으로 활용

III. 인수테스트 프로세스 설명

가. 인수테스트 프로세스 개념도



- 인수테스트는 정의, 계획, 도출, 수행, 협상, 인수 과정으로 수행

나. 인수테스트 프로세스 상세설명

단계	활동	설명	산출물
준비	테스트 계획 수립	- Test Basis 검토, 요구사항 식별 - 테스트 기법 할당	테스트 시나리오
	테스트 환경 구축	- 테스트 환경에 대한 HW/SW 구성 - 테스트를 수행할 환경을 구축	
	테스트케이스 및 데이터 개발	- 테스트에 필요한 테스트 데이터 생성 및 유지 절차를 정의	
	테스트케이스 절차 정의	- 테스트 요소 식별, 범위, 목표수준 정의	
수행	테스트 수행	- 시나리오에 따라 테스트 수행	테스트 수행 계획서
	오류 수정 및 테스트	- 결함 및 이슈 기록, 해결 절차 포함	
평가	테스트 결과 요약	- 결함, 수정된 기능 등 정량적 결과 도출	테스트 결과 보고서
	테스트 결과 검토 및 협상, 승인	- 기대 결과 비교하여 검토 - 협상에 따른 승인 or 거부	
시스템 모니터링	시스템 데이터 수집 및 검토	- 데이터 수집 및 실 운영 데이터 필요 여부 검토	시스템 테스트 보고서
	시스템 자원 사용량 분석	- 테스트시 자원 사용정도 분석	
	시스템 성능 분석	- 시스템에 대한 성능 부하	
	시설 보안 위반 상태 분석	- 제품에 대한 보안 위배사항 확인/분석	

IV. 인수테스트 주요 인수기준

인수기준	설명
기능성	- 단계 내 또는 단계 간의 문서 및 코드 일관성, 기능의 추적성, 논리의 적합성 검증 - 기능 평가 및 시험, 운용환경에서 기능 유지
성능	- 성능 요구의 타당성 조사, 정확한 모의실험 및 구현도구, 운용환경에서 성능분석
인터페이스 품질	- 인터페이스 문서화 / 복잡도 / 통합시험계획, 운용환경에서 인터페이스 시험
소프트웨어 품질	- 품질측정의 정량화, 소프트웨어 산출물의 인수기준, 문서화 및 소프트웨어 개발표준의 적합성, 운용시험의 품질기준
소프트웨어 안전성	- 안전 요구사항 확인 - 안전성 요구되는 시스템에서 부적절한 개발기법 및 기술의 삭제 - 안전위험을 방지하고 복구 할 수 있는 안전장치 및 안전코드 도입
보안성	- 보안 요구 확인, 보안 시험 계획, 보안의 공식 검증

“끝”



법적인 처벌을 받을 수 있습니다.