

ICT의 가치를 이끄는 사람들!!

128회

# 정보관리기술사 기출풀이 3교시

## 국가기술자격 기술사 시험문제

정보처리기술사 제 128 회

제 3 교시

분야	정보통신	종목	정보관리기술사	수험 번호		성명	
----	------	----	---------	-------	--	----	--

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 25 점)

1. 소프트웨어 모듈(Module)과 관련하여 다음을 설명하시오.
  - 가. 소프트웨어 모듈의 응집도와 결합도
  - 나. 소프트웨어 모듈의 fan-in 과 fan-out
  
2. 최근에는 웹기반 서비스의 증가와 모바일 퍼스트(Mobile First)에 따른 웹기반 서비스의 성능관리가 매우 중요하다. 다음에 대하여 설명하시오.
  - 가. 웹 성능저하 요인
  - 나. 프론트-엔드 관점에서의 웹 최적화 방안 6가지
  
3. 최근 시스템이 복잡해지고 안전(Safety)이 중요시되면서 다양한 위험분석 방법이 적용되고 있다. 다음을 설명하시오.
  - 가. 전통적 위험분석 기법인 FMEA(Failure Mode and Effects Analysis), HAZOP(Hazard and Operability Analysis)의 특징 및 한계점
  - 나. STPA(System Theoretic Process Analysis) 개념 및 위험분석 방법
  
4. 공공소프트웨어 사업의 계획단계에서 사업의 확정 및 사업 기간의 적정성 평가를 위한 검토항목과 사업수행 중 과업 변경에 대한 적정성 판단 평가 기준에 대하여 설명하시오.
  
5. 데이터 통합 및 마이그레이션 프로젝트에서 데이터 무결성 목표를 달성하기 위해서는 데이터들의 정합성을 확보하고 신뢰도를 높이는 일이 매우 중요하다. 다음의 내용을 설명하시오.
  - 가. 데이터 무결성(Integrity)과 정합성(Consistency)의 차이
  - 나. 데이터 값(Value) 진단 프로파일링(Data Profiling)의 중점 분석 관점
  - 다. 데이터 마이그레이션 검증 테스트 방법
  
6. 개인정보 보호를 위한 분산 인공지능(AI, Artificial Intelligence) 학습 모델인 연합학습(Federated Learning)에 대하여 아래 사항을 설명하시오.
  - 가. 연합학습의 동작 원리
  - 나. 연합학습의 주요 알고리즘
  - 다. 연합학습의 보안 및 프라이버시 보장형 기술

## 1. 소프트웨어 모듈(Module)과 관련하여 다음을 설명하시오.

문 제

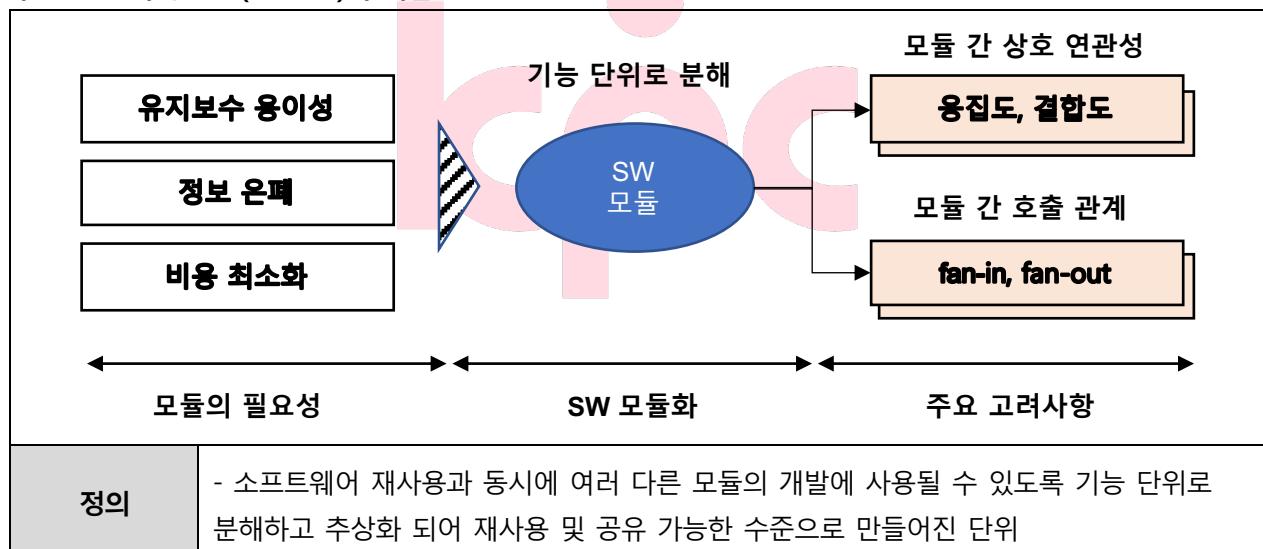
가. 소프트웨어 모듈의 응집도와 결합도

나. 소프트웨어 모듈의 fan-in 과 fan-out

출 제 영 역	소프트웨어공학	난 이 도	★★☆☆☆
출 제 배 경	- 운영, 유지보수, 재사용성을 고려한 소프트웨어 모듈에 대한 지식 확인		
출 제 빈 도	- 모의_2022.03, 합숙_2020.07		
참 고 자 료	<ul style="list-style-type: none"> <li>- 클린 아키텍처: 소프트웨어 구조와 설계의 원칙(로버트 C. 마틴 지음)</li> <li>- 정보통신용어해설(<a href="http://www.ktword.co.kr/test/view/view.php?m_temp1=2226">http://www.ktword.co.kr/test/view/view.php?m_temp1=2226</a>)</li> </ul>		
Key word	<ul style="list-style-type: none"> <li>- 응집도(우연적, 논리적, 시간적, 절차적, 통신적, 순차적, 기능적), 결합도(내용, 공통, 외부, 제어 스템프, 자료구조), 재사용, 복잡도, fan-in, fan-out</li> </ul>		
풀 이	백기영 (126 회 정보관리기술사)		

### 1. 재사용 및 공유 가능한 수준으로 만들어진 단위, 소프트웨어 모듈(Module)의 개요

#### 가. 소프트웨어 모듈(Module)의 개념



- 소프트웨어 모듈은 비용을 고려한 최적의 모듈 수를 산정하고 자료 추상화를 통한 정보 은닉을 구현.

#### 나. 소프트웨어 모듈(Module)의 원리

원리	설명
모듈과 비용의 관계	- 모듈수가 증가하면 인터페이스 비용이 증가함으로 최적의 모듈수 산정
모듈의 독립성	- 모듈 간의 관계에서 낮은 결합도와 높은 응집도를 고려한 SW 모듈 설계
자료 추상화	- 각 모듈 자료구조를 접근하고 수정하는 함수 내에 자료 구조의 표현 내역을 은폐
- 소프트웨어 모듈 간 독립성과 개발 편의를 위하여 <u>응집도, 결합도</u> 측면에서 SW를 설계하고 개발할 필요가 있음.	

## 2. 소프트웨어 모듈의 응집도(Cohesion)와 결합도(Coupling) 설명

### 가. 소프트웨어 모듈의 응집도(Cohesion) 상세 설명

구분	설명																		
정의	- 소프트웨어의 한 모듈이 해당 기능을 수행하기 위해 얼마만큼의 연관된 책임과 아이디어가 뭉쳐 있는지를 나타내는 정도																		
개념도	<p style="text-align: center;">“높은 응집도 관계”</p> <p>모듈 단위로 같은 목적의 기능이 응집</p> <p>A -&gt; C 호출</p> <p>B -&gt; C 호출</p> <p>모듈 내 기능</p>																		
단계	<table border="1"> <thead> <tr> <th>응집도</th> <th>단계</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td rowspan="7" style="text-align: center; vertical-align: middle;"> <span style="font-size: 2em;">↓</span> 낮음  높음(좋음)         </td> <td>우연적</td> <td>- 모듈 간 의미 있는 연관이 없는 기능으로 구성</td> </tr> <tr> <td>논리적</td> <td>- 유사한 성격의 처리 요소들이 한 모듈에서 처리</td> </tr> <tr> <td>시간적</td> <td>- 특정 시간에 처리되어야 하는 활동을 모은 모듈</td> </tr> <tr> <td>절차적</td> <td>- 모듈 내 구성 요소들이 순차적으로 수행</td> </tr> <tr> <td>통신적</td> <td>- 동일한 입/출력을 사용하여 다른 기능을 수행</td> </tr> <tr> <td>순차적</td> <td>- 모듈 내 한 활동의 출력값이 다른 활동의 입력값</td> </tr> <tr> <td>기능적</td> <td>- 모듈 내의 모든 기능이 단일 목적으로만 수행</td> </tr> </tbody> </table>	응집도	단계	설명	<span style="font-size: 2em;">↓</span> 낮음  높음(좋음)	우연적	- 모듈 간 의미 있는 연관이 없는 기능으로 구성	논리적	- 유사한 성격의 처리 요소들이 한 모듈에서 처리	시간적	- 특정 시간에 처리되어야 하는 활동을 모은 모듈	절차적	- 모듈 내 구성 요소들이 순차적으로 수행	통신적	- 동일한 입/출력을 사용하여 다른 기능을 수행	순차적	- 모듈 내 한 활동의 출력값이 다른 활동의 입력값	기능적	- 모듈 내의 모든 기능이 단일 목적으로만 수행
응집도	단계	설명																	
<span style="font-size: 2em;">↓</span> 낮음  높음(좋음)	우연적	- 모듈 간 의미 있는 연관이 없는 기능으로 구성																	
	논리적	- 유사한 성격의 처리 요소들이 한 모듈에서 처리																	
	시간적	- 특정 시간에 처리되어야 하는 활동을 모은 모듈																	
	절차적	- 모듈 내 구성 요소들이 순차적으로 수행																	
	통신적	- 동일한 입/출력을 사용하여 다른 기능을 수행																	
	순차적	- 모듈 내 한 활동의 출력값이 다른 활동의 입력값																	
	기능적	- 모듈 내의 모든 기능이 단일 목적으로만 수행																	

- 응집도는 높은 단계의 방향으로 설계하며 결합도는 낮추는 방향으로 설계하는 것이 좋은 소프트웨어의 조건.

### 나. 소프트웨어 모듈의 결합도(Coupling) 상세 설명

구분	설명
정의	- 모듈 내부가 아닌 외부의 모듈과의 연관도 또는 모듈 간의 상호 의존성을 나타내는 정도
개념도	<p style="text-align: center;">낮은 결합도 (A 모듈)</p> <p>낮은 결합도 (B 모듈)</p> <p>낮은 결합도 (C 모듈)</p> <p>호출</p> <p>호출</p> <p>호출</p> <p>호출</p> <p>B 모듈의 기능을 수정하려면 C, E, F 모듈의 확인 필요</p>

단계	결합도	단계	설명
	낮음(좋음)	자료	- 모듈 간 I/F로 전달되는 파라미터로 상호작용
		스탬프	- 모듈 간 I/F로 배열, 객체, 구조 등이 전달
		제어	- 제어 신호를 통해 다른 모듈의 이용
		외부	- 외부에서 입력된 데이터 포맷, 통신, 디바이스 I/F 공유
		공통	- 전역변수를 참조, 갱신하는 방식으로 상호 작용
	높음	내용	- 모듈 내부 변수, 기능을 다른 모듈에서 사용

- 응집도와 결합도는 모듈 간의 관계를 각각의 단계별로 정의할 수 있으나 모듈 간 호출 관계를 깊이(Depth)와 넓이(Widht) 파악하기 위해서는 fan-in, fan-out의 정량적 분석이 필요 함.

### 3. 소프트웨어 모듈의 fan-in 과 fan-out 설명

#### 가. 소프트웨어 모듈의 fan-in 상세 설명

구분	설명																						
정의	- 모듈을 계층적으로 분석하거나 시스템 복잡도를 측정하기 위해 어떤 모듈을 호출하는 모듈의 수를 분석 하는 기법																						
개념도	<table border="1"> <tr> <td>A</td> <td>B</td> <td>C</td> </tr> <tr> <td>D</td> <td>E</td> <td>F</td> <td>G</td> </tr> </table> <table border="1"> <tr> <td>fan-in</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> </table>								A	B	C	D	E	F	G	fan-in	0	1	1	1	1	1	1
A	B	C																					
D	E	F	G																				
fan-in	0	1	1	1	1	1	1																
모듈 숫자 계산	- 자신을 기준으로 모듈에 들어오면 fan-in 으로 카운팅																						
고려사항	<ul style="list-style-type: none"> <li>- fan-in 이 높을 경우 재사용 설계가 좋음</li> <li>- 단일 장애점(SPOF) 발생 가능</li> <li>- 관리 및 테스트 비용 증가</li> </ul>																						

- fan-in, fan-out은 디지털 논리회로에서 논리 게이트에 연결될 수 있는 입력과 출력의 최대값을 의미.

#### 나. 소프트웨어 모듈의 fan-out 상세 설명

구분	설명							
정의	- 모듈을 계층적으로 분석하거나 시스템 복잡도를 측정하기 위해 어떤 모듈에 의해 호출되는 모듈의 수를 분석 하는 기법							

개념도	<pre> graph TD     A[A] --&gt; B[B]     A --&gt; C[C]     B --&gt; D[D]     B --&gt; E[E]     C --&gt; F[F]     C --&gt; G[G]   </pre>																						
	<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th><th>G</th></tr> </thead> <tbody> <tr> <td><b>fan-out</b></td><td>2</td><td>2</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>									A	B	C	D	E	F	G	<b>fan-out</b>	2	2	2	0	0	0
	A	B	C	D	E	F	G																
<b>fan-out</b>	2	2	2	0	0	0	0																
모듈 숫자 계산	<ul style="list-style-type: none"> <li>- 자신을 기준으로 모듈에서 나가면 fan-out으로 카운팅</li> </ul>																						
고려사항	<ul style="list-style-type: none"> <li>- fan-out이 높을 경우, 불필요한 모듈 호출 증가 위험</li> <li>- 단순화 가능 여부 검토 필요</li> </ul>																						

- fan-in, fan-out은 시스템 복잡도 보여주는데 이를 최적화 위해서 fan-in은 높게, fan-out은 낮게 설계해야 됨.

#### 4. 소프트웨어 모듈과 컴포넌트와의 관계

구분	모듈	컴포넌트
단위	<ul style="list-style-type: none"> <li>- 만들어진 산출물의 구현 단위</li> <li>- 구조의 단위, 기능의 단위</li> </ul>	<ul style="list-style-type: none"> <li>- 실행 중인 소프트웨어의 활동 단위</li> <li>- 런타임에 독립적으로 배포, 실행 단위</li> </ul>
특징	<ul style="list-style-type: none"> <li>- 시스템 상의 어떤 형태로 표현 가능</li> <li>- 소프트웨어가 동작하지 않을 수 있음</li> </ul>	<ul style="list-style-type: none"> <li>- S/W가 동작할 때부터 의미가 있음</li> <li>- 시스템이 동작하지 않으면 사라짐</li> </ul>
관계 설명	<ul style="list-style-type: none"> <li>- 1 개의 Server 와 10 개의 Client 가 있다면</li> <li>- 모듈 2 개: Server 1 개 + Client 대표 1 개</li> <li>- 컴포넌트 11 개: Server 1 개 + Client 전부 10 개</li> </ul>	

- 소프트웨어 재사용성을 높이기 위해 캡슐화, 정보숨김, 데이터 추상화를 해야되며 이를 위해 모듈과 컴포넌트 단위로 설계가 필요 함.

"끌"

#### 기출풀이 의견

- SW 모듈의 응집도, 결합도, fan-in, fan-out은 고전 토픽으로 많은 분들이 작성합니다. 차별화를 위해 충실히 키워드와 실무 적용사례 또는 설계 원칙 등 본인 만의 경험을 작성 하시됩니다.

## 문제

2. 최근에는 웹기반 서비스의 증가와 모바일 퍼스트(Mobile First)에 따른 웹기반 서비스의 성능관리가 매우 중요하다, 다음에 대하여 설명하시오.

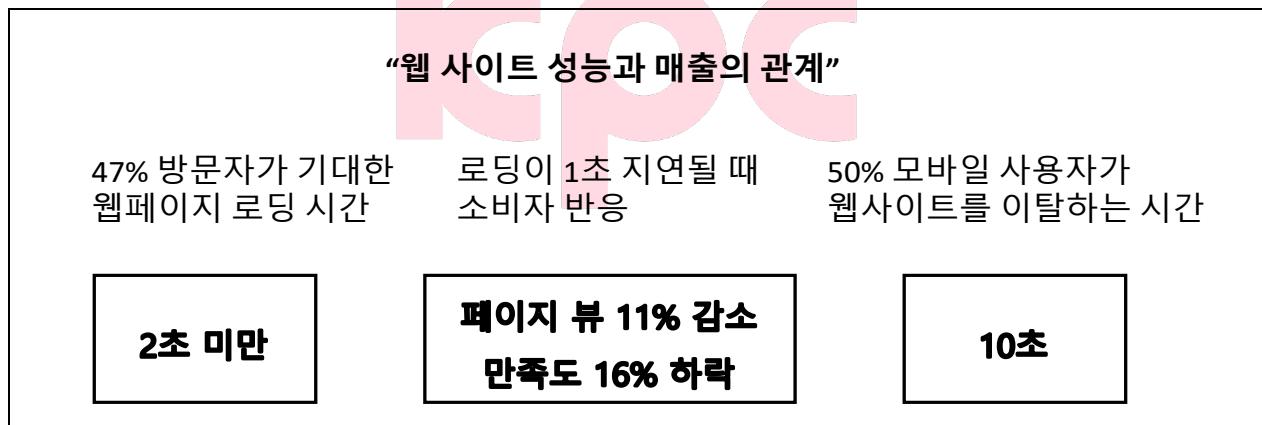
## 가. 웹 성능저하 요인

## 나. 프론트-엔드 관점에서의 웹 최적화 방안 6 가지

출 제 영 역	소프트웨어공학	난 이 도	★★☆☆☆
출 제 배 경	- 많은 서비스가 웹으로 제공되어 Web/WAS, Application 성능 관리가 다양한 측면에서 중요		
출 제 빈 도	미출제		
참 고 자 료	<ul style="list-style-type: none"> <li>- 최신 웹 개발을 위한 지침 제공 (<a href="https://web.dev/i18n/ko/why-speed-matters/">https://web.dev/i18n/ko/why-speed-matters/</a>)</li> <li>- 웹 성능 최적화 기법 웹에 날개를 달아주는 (강상진, 윤호성, 박정현 저)</li> </ul>		
Key word	- CDN, CSS, JS, 부하분산, 로드밸런싱, 캐싱, DNS, WebP, DOM, FCP, TTI		
풀 이	백기영 (126 회 정보관리기술사)		

## 1. 성능이 곧 매출, 웹기반 서비스 성능관리의 개요

## 가. 웹기반 서비스 성능의 중요성



- 웹기반 서비스의 성능은 고객의 만족도에 지대한 영향을 미치며 이는 곧 기업 매출과 사용자 가치와 직결 됨.

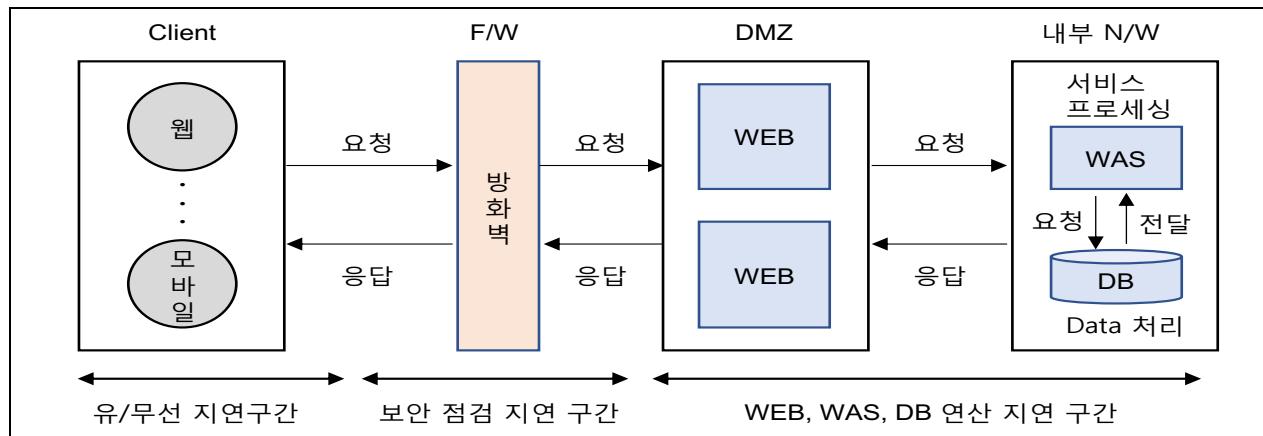
## 나. 웹기반 서비스 성능의 영향도

영향도	설명
사용자 유지 좌우	- 성능이 좋은 웹 서비스는 더 많은 사용자의 참여를 유도
전환율 향상 좌우	- 느린 사이트는 수익에 부정적인 영향을 미치고 빠른 사이트는 전환율을 높임
사용자 경험 좌우	- 웹 서비스가 로드되기 시작하면 사용자가 콘텐츠가 표시되기를 기다리는 시간
사람을 좌우	- 성능이 좋은 웹 서비스는 이를 사용하는 사람들에게 실질적인 이윤을 제공

- 웹기반 서비스 성능 향상을 위해 전 구간 별 상세 성능 진단을 실시하고 구간별 성능 지표 설정이 필요.

## 2. 웹 성능저하 요인 설명

### 가. 웹 성능저하 주요 구간



- 웹 성능 개선을 위해 client 부터 웹 서버/DB 까지 발생하는 성능 저하 요인을 상세히 도출 필요.

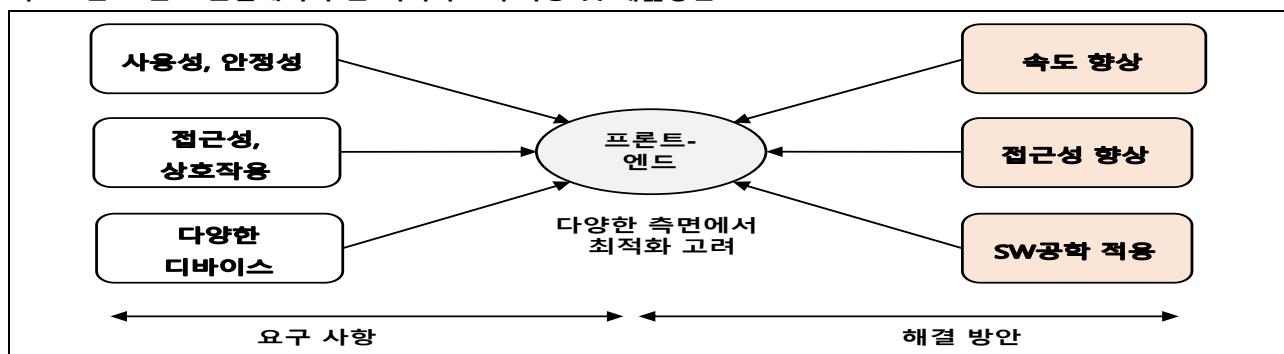
### 나. 웹 성능 저하요인 주요 구간별 상세 설명

분류	저하 요인	설명
유/무선 지연구간	- Client 요청 증가	- 웹, 모바일 등 다양한 사용자 요청 수
	- 대용량 미디어 파일 전송	- 동영상, 음성 등 큰 사이즈의 파일 업/다운로드
	- 캐싱 기술 미 적용	- 웹 브라우저 캐싱 미적용으로 매번 서버에 요청
	- CDN 미 적용	- 가까운 CDN 미 적용으로 원 서버가 있는 IDC로 요청
보안 점검 지연 구간	- IP, Port 점검	- Client의 접속 시도 IP와 Port를 점검하는 시간 소요
	- IDS, IPS 점검	- 침입차단, 침입발견 등 보안 장비에서의 시간 소요
	- 요청 내용 점검	- Client의 요청에 부정한 data가 있는지 점검
	- 반출 Data 점검	- 외부로 리턴되는 data에 기업 보안 위반 사항 점검
WEB, WAS, DB 연산 지연 구간	- 최적화되지 않은 CSS	- 캐스캐이딩 스타일 시트에 부정확한 마크업 사용
	- 부정한 코딩	- 개발자의 실수 또는 의도된 부정한 소스코드
	- 튜닝 되지 않은 DB Query	- 인덱스 미 활용, 불필요 데이터 조회 등 시간 소요
	- 서버 설정 문제	- 커넥션 풀, 메모리 사이즈 등 미숙한 서버 설정

- 사용자 입장에서 직접 체감할 수 있는 프론트-엔드 관점에서의 최적화 방안 수립 필요.

## 3. 프론트-엔드 관점에서의 웹 최적화 방안 6 가지 설명

### 가. 프론트-엔드 관점에서의 웹 최적화 요구사항 및 해결방안



- 프론트-엔드 관점은 사용자가 이용하는 디바이스에서 실행되는 성능 측면을 고려한 최적화 접근 필요.

## 나. 프론트-엔드 관점에서의 웹 최적화 방안 6 가지

분류	웹 최적화 방안	설명
웹페이지 로딩 최적화	JS, CSS 로드 시 블로킹 방지	- script 로드를 위한 script 태그는 DOM 생성 블로킹을 일으킬 수 있기 때문에 HTML 최하단에 배치
	JS, CSS 로드 사이즈 줄이기	- Webpack, Parcel 같은 번들러가 제공하는 기능을 이용해서 파일 사이즈를 줄임
	WebP 사용, 이미지 스프라이트	- 이미지 스프라이트는 여러 개 이미지를 하나로 생성 - CSS의 속성을 활용해 부분 이미지 사용
	레이저 로드(lazy load) 이용	- Html의 loading 옵션을 이용 - Intersection Observer를 활용해서 적시 이미지 로드
웹페이지 랜더 및 실행 최적화	JS 실행 최적화	- Web Worker를 이용해서 비동기적으로 계산
	레이아웃 변경 전략	- 강제 동기 레이아웃 피하기, 레이아웃 스래싱 피하기 - JS 보다는 CSS 애니메이션을 활용
	DOM 변화는 최대한 적게	- DocumentFragment 활용해서 한번에 붙이는게 유리
	FCP(First Contentful Paint) 및 TTI(Time to Interactive) 개선	- FCP(텍스트/이미지 그리기 시작), TTI(사용자가 행동을 취할수 있는 시점)을 개선하기 위한 전략 수립
공통 영역 최적화	도메인 분리	- 파일을 불러오는 도메인은 다르게 정의
	Service Worker 사용	- JS 코드나 스태틱 리소스들을 브라우저 내부에 캐싱해서 오프라인 상황에서도 사용

- 웹 서비스 성능 개선하기 위해서는 프론트-엔드 뿐만 아니라 백-엔드 성능까지 최적화가 필요.

## 4. 백-엔드 관점에서의 웹 최적화 방안

DNS	웹 서버	캐싱	S/W
<ul style="list-style-type: none"> <li>■ 서버 증설</li> <li>■ DNS 정보를 최대한 캐싱</li> </ul>	<ul style="list-style-type: none"> <li>■ N/W 출력, 대역폭 증설</li> <li>■ CPU, RAM 증설</li> <li>■ 프록시 서버 설정</li> </ul>	<ul style="list-style-type: none"> <li>■ CDN 활용</li> <li>■ 웹 콘텐츠 캐싱</li> </ul>	<ul style="list-style-type: none"> <li>■ Clean Code</li> <li>■ 부하 테스트</li> </ul>

- 웹 서비스 최적화를 위해 프론트-엔드 뿐만 아니라 백-엔드 및 프로토콜 최적화 등 3 가지 관점에서 접근 필요.

"끝"

## 기출풀이 의견

2. 프론트-엔드 관점에서 최적화는 다양함으로 가급적 경험 기반으로 작성하시는 것이 차별화할 수 있는 포인트입니다. 그리고, 4단락에는 백-엔드, 프로토콜 관점에서 최적화를 작성하면 좋습니다.

3. 최근 시스템이 복잡해지고 안전(Safety)이 중요시되면서 다양한 위험분석 방법이 적용되고 있다. 다음을 설명하시오.

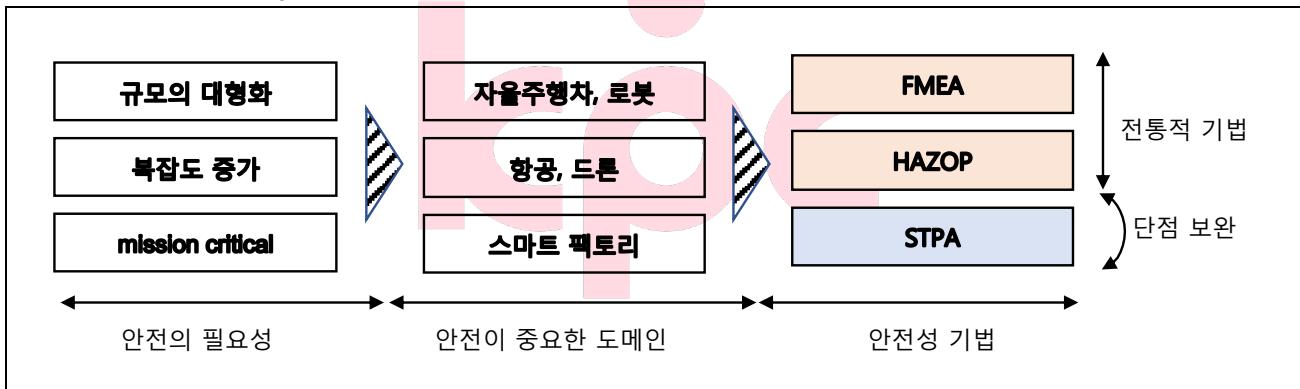
문 제

가. 전통적 위험분석 기법인 FMEA(Failure Mode and Effects Analysis), HAZOP(Hazard and Operability Analysis)의 특징 및 한계점

나. STPA(System Theoretic Process Analysis) 개념 및 위험분석 방법

출 제 영 역	소프트웨어공학	난 이 도	★★★☆☆
출 제 배 경	- 신 기술 분야에서 안전은 사람의 생명과 직결 됨으로 위험 분석이 무엇보다 중요 함		
출 제 빈 도	- 108 회_관리_2, 95 회_응용_1, 모의_2020.10		
참 고 자 료	- TTA: [SW 안전] STPA 를 활용한 위험분석 가이드( <a href="https://sw.tta.or.kr/notify/data_view.jsp?no=63">https://sw.tta.or.kr/notify/data_view.jsp?no=63</a> )		
Key word	- 전통적기법(FMEA, HAZOP, FTA), RPN, 이탈, 공정변수, 가이드워드, 시나리오 테이블, STAMP		
풀 이	백기영 (126 회 정보관리기술사)		

1. 시스템 안전(Safety)의 중요성 증대에 따른 위험분석 방법의 개요



- SW 의 규모와 복잡도가 증가함에 따라 기능적 실패를 만들어 내는 위험(Hazard) 요소들도 다양해지고 있음.
- 위험 유형이 점차적으로 사람의 안전/생명과 직결됨에 따라 위험분석 방법 또한 계속해서 고도화되고 있음.

2. 전통적 위험분석 기법인 FMEA(Failure Mode and Effects Analysis), HAZOP(Hazard and Operability Analysis)의 특징 및 한계점

가. 고장발생 회피 기법, FMEA(Failure Mode and Effects Analysis)의 특징 및 한계점

구분	설명
정의	<ul style="list-style-type: none"> <li>- 고장 형태 영향분석 이란 뜻으로 제품개발 및 공정 프로세스 상에서 발생 가능한 고장(Failure)과 이러한 고장으로 인해 야기될 수 있는 위험을 구조화하여 사전에 방지하는 방법</li> </ul>

개념도						
특징	<ul style="list-style-type: none"> <li>- 다른 위험분석 방법보다 서식이 간단</li> <li>- 적용 시점에 따라 3 가지 유형으로 분류: 제품(System), 설계(Design), 공정(Process)</li> <li>- 주요 지표로 RPN(Risk Priority Number)을 활용</li> <li>- <math>RPN = \text{중요도(심각도)} * \text{발생확률(발생도)} * \text{검출가능성(검출도)}</math></li> </ul>					
한계점	<ul style="list-style-type: none"> <li>- 논리성이 부족하고 각 요소간의 영향을 분석하기 어려움</li> <li>- 두가지 이상의 요소가 고장나면 분석이 곤란</li> <li>- 부품 수가 많아지거나, 고장의 영향이 크면 많은 시간과 노력이 소요</li> <li>- 원인과 결과 사이에 직접적인 연관관계가 없으면 복잡하고 방대하여 감당하기 어려움</li> <li>- 제품에 미치는 영향을 정량화 위해 균일한 근거를 제공할 수 있을 만큼 상세하지 못함</li> </ul>					
<ul style="list-style-type: none"> <li>- FMEA 는 알려진 원인들로부터 시작하여 가능한 영향들을 분석하는 반면, HAZOP 은 하나의 사건에서 출발하여 가능한 원인과 영향들을 분석하는 기법.</li> </ul>						
<b>나. 위험요소 조사 기법, HAZOP(Hazard and Operability Analysis)의 특징과 한계점</b>						
구분	설명					
정의	<ul style="list-style-type: none"> <li>- 공정에 존재하는 위험요인과 공정의 효율을 저해 할 수 있는 운전상의 문제점을 찾아내어 그 원인을 제거하는 방법</li> </ul>					
개념도	$\text{이탈} \\ (\text{Deviation}) = \text{공정 변수} \\ (\text{Process Parameter}) \times \text{가이드 워드} \\ (\text{Guide Words})$ <p>설계(정상운전조건)에서 벗어난 상태</p>					
특징	<ul style="list-style-type: none"> <li>- 변수의 질이나 양을 표현하기 위한 가이드 워드 사용. (없음, 증가, 감소, 반대, 부가, 부분, 기타)</li> <li>- 설계 의도를 벗어난 상태를 측정하기 위해 가이드워드와 공정변수 활용</li> <li>- 설계 단계에서 예기치 못한 위험을 발견하여 설계의 완성도를 높일 수 있음</li> <li>- 두개 이상의 기기가 동시에 고장나는 사고는 검토에서 제외(이탈은 1 대의 기기만 발생)</li> <li>- 체계적이고 조직적 분석이 가능</li> <li>- 경험이 풍부한 전문적 인력이 필요하며 관련 자료를 수집하는데 시간이 걸림</li> </ul>					

한계점	<ul style="list-style-type: none"> <li>- 부품 수가 많아지거나, 고장의 영향이 크면 많은 시간과 노력이 소요</li> <li>- 전문가의 역량에 따라 다루어지지 못하는 위험이 존재</li> <li>- 인적 오류나 설비 결함에 대해서는 다루어지 지지 않음</li> <li>- 원인과 결과 사이에 직접적인 연관관계가 없으면 복잡하고 방대 해짐</li> <li>- 제품 내 상이한 부품들에 발생하는 결함들 간의 다중 종속성이나 복잡한 상호 작용을 고려하기에는 곤란</li> </ul>
-----	---

- 기존 위험분석 방법의 한계점 발생 그리고 사고 양상을 보면 시스템뿐 아니라 여러 시스템 외적인 요인(사람, 정책, 환경 등)들이 결합해 발생함에 따라, 위험을 복합적 시각에서 분석하는 접근 방법이 필요.

### 3. STPA(System Theoretic Process Analysis) 개념 및 위험분석 방법 설명

#### 가. 시나리오 테이블을 이용, STPA(System Theoretic Process Analysis) 개념

구분	설명	
정의	<ul style="list-style-type: none"> <li>- STAMP를 기반으로 하는 위험분석 기법으로, 시스템 생명주기 전 과정에 걸쳐 존재하는 잠재적인 위험과 발생 원인을 시스템의 상위 수준에서 분석하는 새로운 기법</li> </ul>	
배경	<p>“사람, 정책, 환경 고려”</p> <p>방법론</p> <p>이론</p> <p>기존 방법론</p> <p>한계 발생</p> <p>STAMP 기반한 STPA</p>	
특징	상위 수준 분석	<ul style="list-style-type: none"> <li>- 시스템 전 생명주기 존재하는 잠재적 위험, 발생 원인</li> </ul>
	위험은 제어 문제에서 발생	<ul style="list-style-type: none"> <li>- 위험(Hazard)을 시스템과 시스템 또는 구성요소들 간 제어 문제(Control Problem)에서 발생함을 기본 전제</li> </ul>
	원인 시나리오 도출	<ul style="list-style-type: none"> <li>- 사고 정의에서 시작하여 원인 시나리오(Causal Scenario) 도출을 수행하는 하향식 분석 체계를 가짐</li> </ul>
	안전 제약사항 및 요구사항 도출	<ul style="list-style-type: none"> <li>- 안전 제약사항은 위험을 발생시키지 않도록 하기 위한 시스템 수준의 요건을 의미</li> </ul>

- STPA는 STAMP를 기반으로 하며 전통적 위험분석의 문제점을 보완하여 컴포넌트 중심의 접근 문제점을 해결.

#### 나. STPA(System Theoretic Process Analysis) 절차 상세 설명

구분	설명			
절차도	<p>1단계: 사고 및 위험 정의</p> <p>2단계: Control Structure 도식화</p> <p>3단계: Unsafe Control Action 도출</p> <p>4단계: 원인 시나리오 도출</p> <p>CA: Control Action, UCA: Unsafe Control Action</p>			

단계	상세	설명
① 사고 및 위험 정의	관련 사고 도출	- 시스템에 발생할 수 있는 사고를 정의
	위험 정의	- 대상 시스템을 선정하고 시스템 범위를 정의
	위험을 안전 제약사항으로 변환	- 정의한 각 위험 발생을 방지하기 위한 상태나 행동으로 정의
② Control Structure 도식화	제어 관계 따른 개체(컴포넌트) 식별	- Control에 관련한 컴포넌트를 정의 및 분석
	제어명령, 피드백, 프로세스 모델 등 도식화	- Control Structure는 Control loop 형태를 띠며 제어의 관점으로 주체, 객체, 제어, 반응으로 구성
③ UCA(Unsafe Control Action) 도출	4 가지 유형에 따른 UCA 도출	- UCA는 시스템의 위험을 유발할 수 있는 Control Action의 불안전한 형태를 의미
	4 가지 유형	1) Control Action 부재 2) 부적절한 Control Action 3) Control Action 제공 시간, 순서 4) Control Action 지속시간
④ 원인 시나리오 도출	Unsafe Control Action 발생 원인 도출	- UCA 발생 원인 분석 - CA 기반 직관적 원인 시나리오 도출

- 제4 차 산업혁명의 도래로 여러 분야에서 안전한 SW의 개발이 중요해지고 있음.

#### 4. 시스템 안전이 중요해짐에 따른, 위험분석의 시사점

##### 사회적 측면

- SW 안전 활동의 인식 제고
- 적극적 교육과 홍보 필요
- 충분한 검증에 대한 신뢰

##### 제도적 측면

- 프로젝트 의무 비용으로 합당
- 위험분석에 대한 합당한 대가 산정
- 안전검증 전문가 양성

- SW 안전이 확보될 수 있도록 위험원 분석 및 위험 평가에 대한 적극적인 홍보화 교육이 필요 함.

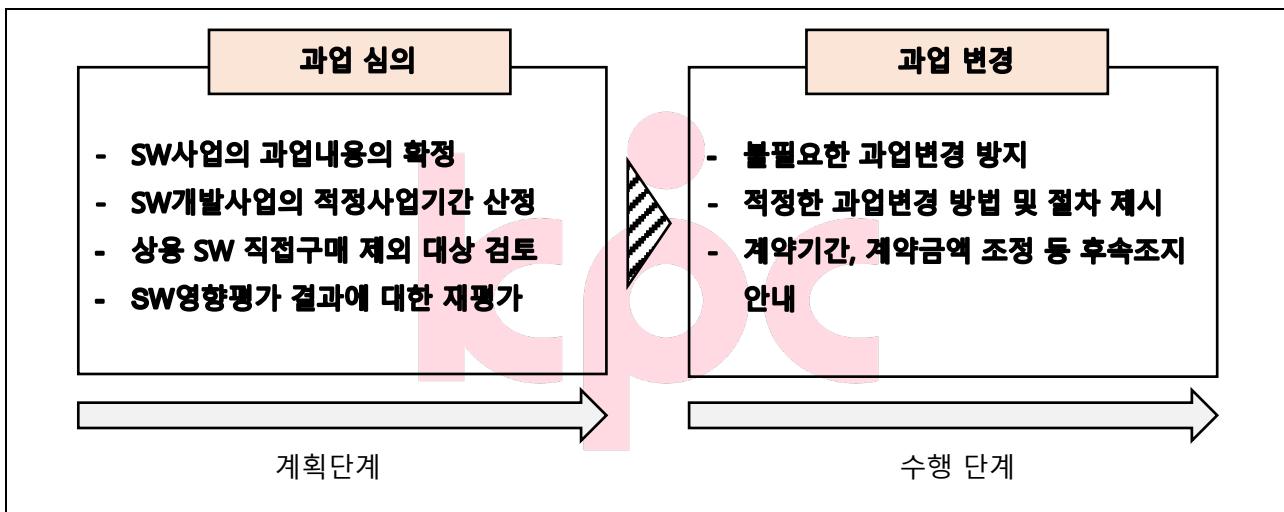
"끝"

#### 기출풀이 의견

3. 위험분석 방법론은 관련 업종의 종사자가 아니면 쉽지 않은 토픽입니다. FMEA, FTA, HAZOP, STPA 등 위험분석 방법론이 만들어진 이유를 잘 이해하고 SDLC 관점에서 접근하길 추천 드립니다.

문제 제	4. 공공소프트웨어 사업의 계획단계에서 사업의 확정 및 사업 기간의 적정성 평가를 위한 검토항목과 사업수행 중 과업 변경에 대한 적정성 판단 평가 기준에 대하여 설명하시오.
출제영역	소프트웨어공학
난이도	★★★☆☆
출제배경	- 공공 소프트웨서의 과업, 기간, 변경 등 적정한 평가기준을 가이드 기준으로 묻는 문제
출제빈도	- 합숙_2021.05
참고자료	- NIPA, 공공소프트웨어사업 과업심의, 과업변경 가이드
Keyword	- 과업심의, 과업변경, 과업심의위원회, FP 기반 SW 사업 적정, 사업기초자료, 유사사업자료, 과업변경(법령 제/개정, 정책/기술변화 등 기타 사유, 사업자와 합의한 경미한 과업)
풀이	백기영 (126 회 정보관리기술사)

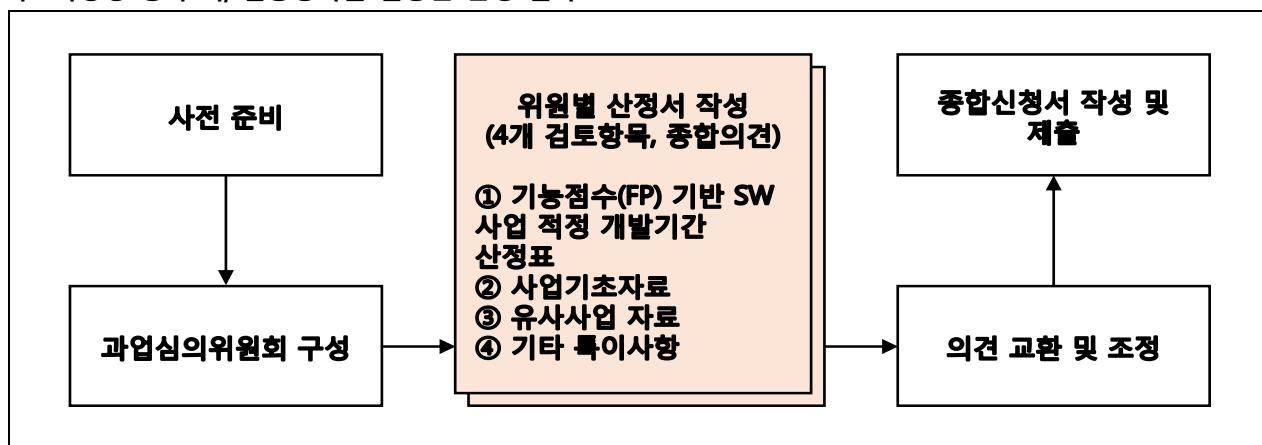
### 1. 공공소프트웨어 과업심의 및 과업변경 시, 적정성 평가의 개요



- 공공 소프트웨어사업 과업심의 가이드에 과업내용 확정 및 과업변경 시 적정성 판단 기준을 제시.

### 2. 계획단계에서 사업의 확정 및 사업 기간의 적정성 평가를 위한 검토항목 설명

#### 가. 적정성 평가 시, 검통항목을 활용한 산정 절차



- 공공 소프트웨어사업 특성을 고려하여 검토항목을 선정하고 필요 시 일부 내용을 수정할 수 있음.

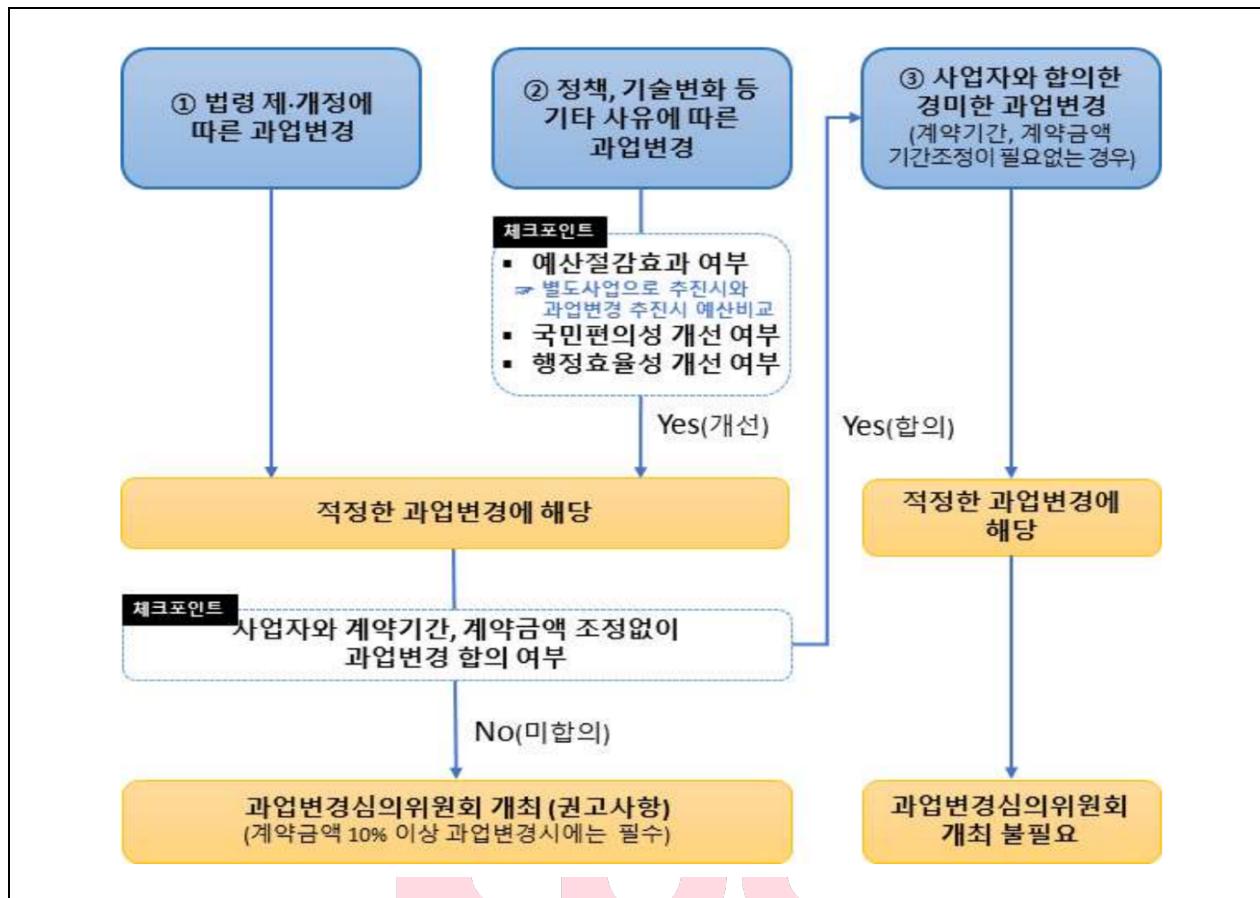
## 나. 적정성 평가를 위한 검토항목 상세 설명

분류	검토 항목	세부 활동
본 사업 문서 기반	① 기능점수(FP) 기반 SW 사업 적정 개발기간 산정표	1) 전체 기능점수 계산이 타당한지 검토(위원회) 2) 발주자가 적용한 1인 생산성이 타당한지 검토(위원회) 3) 1인 총 투입기간 산정 방법 설명(발주자) 및 타당성 검토(위원회) 4) 투입인력 수의 산정 기준 설명(발주자) 및 타당성 검토(위원회) 5) 추정 사업기간의 타당성 검토(위원회) 6) '소프트웨어 개발사업의 적정 사업기간 위원별 산정서' 작성 및 제출
	② 사업 기초자료	1) 대상사업 설명(발주자) 2) 대상사업 이해를 위한 논의(위원회) 3) 사업기초자료를 검토하여 적정 사업기간 추정(위원회) 4) '소프트웨어 개발사업의 적정 사업기간 위원별 산정서' 작성 및 제출
본 사업 외 문서 기반	③ 유사사업 자료	1) 대상사업 설명(발주자) 2) 대상사업 이해를 위한 논의(위원회) 3) 유사사업과 대상사업간 차이분석 실시(위원회) 4) 유사사업 자료를 검토하여 적정 사업기간 추정(위원회) 5) '소프트웨어 개발사업의 적정 사업기간 위원별 산정서' 작성 및 제출
	④ 기타 특이사항	1) 사업기간에 영향을 미치는 특이사항을 설명(발주자) 2) 특이사항으로 인해 사업기간에 영향을 줄 수 있는 추가 기간 산정/제출

- 계획단계에서 사업기간의 적정성 평가 후, 수행 단계에서 과업변경이 있을 경우 제도와 절차에 따라 진행 필요.

### 3. 사업수행 중 과업 변경에 대한 적정성 판단 평가 기준

#### 가. 과업 변경에 대한 적정성 평가 기준 절차도



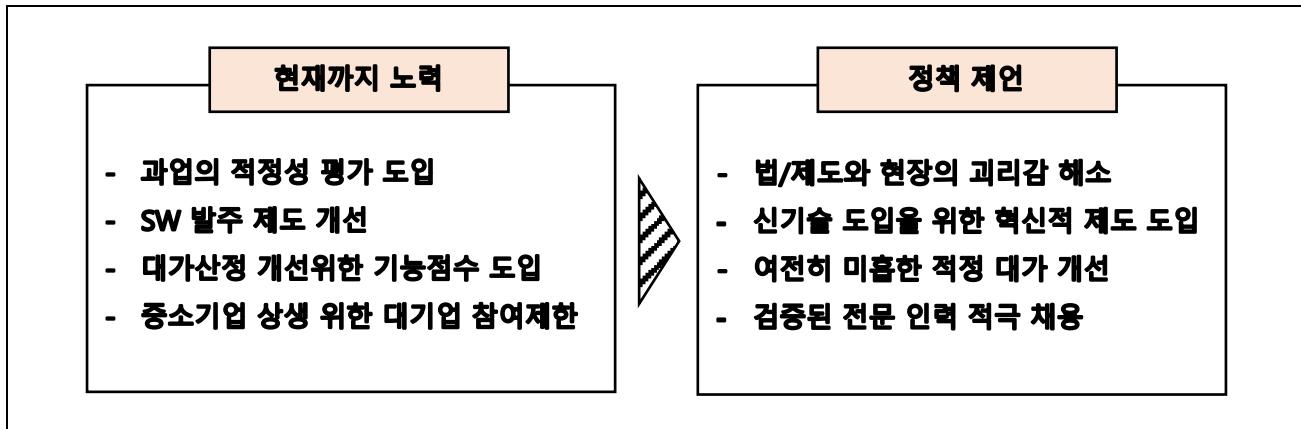
- 과업 변경 시, 적정한 과업변경 인지 불필요한 과업변경 인지 판단하여 검토 필요.

#### 나. 과업 변경 적정성 평가 기준 상세 설명

분류	평가 기준	설명
적정한 과업 변경 여부	법령 개정 등 제제도의 변경	<ul style="list-style-type: none"> <li>- 과업변경심의위원회 심의를 거치도록 권고</li> <li>- 과업변경시 계약기간, 계약금액 조정 등 후속조치 필요</li> </ul>
	기술적/정책적 환경변화	<ul style="list-style-type: none"> <li>- 과업변경시 계약기간, 계약금액 조정 등 후속조치 필요</li> <li>- 과업 변경이 향후 예산절감이 가능한 경우 대국민 편의성 행정 효율성이 개선되는 경우</li> </ul>
불필요한 과업변경 여부	수·발주자간에 사업비 조정 없이 가능하다고 합의한 경미한 과업	<ul style="list-style-type: none"> <li>- 과업변경심의위원회 심의가 불필요하며 필요시 계약기간 조정 등 권고</li> </ul>
	적정한 과업변경 요건에 해당되지 아니하는 과업	<ul style="list-style-type: none"> <li>- 법령 개정 등 제도 변화와 무관한 경우</li> <li>- 기술적 정책적 환경변화에 따른 과업변경이나 예산절감 효과 등이 없는 경우</li> <li>- 수/발주자간 합의 없이 발주기관이 일방적으로 정한 경우</li> </ul>

- 공공 소프트웨어 발전을 위해 정부 및 기관에서는 법/제도 개선을 지속적으로 개선해 왔음..

#### 4. 공공소프트웨어 발전을 위한 정책 제언



- SW 산업은 4 차 산업혁명 시대에 국가경쟁력을 견인할 수 있는 핵심적인 산업이고, 고용유발 효과가 제조업의 2 배에 달하여 실업 문제를 해결하는 방안이 될 수 있으므로 활성화될 필요가 큰 산업.

"끝"



#### 기출풀이 의견

4. 공공소프트웨어 사업 관련한 법과 제도, 가이드 등 자주 출제되고 있습니다. 최신 버전의 내용으로 작성이 필요하며, 이러한 문제는 정답이 있기에 정확이 작성하시는 것이 중요합니다. 마지막 단락에는 기술사로서의 제언을 작성해주는 것도 채점자에게 좋은 인상을 줄 수 있습니다.

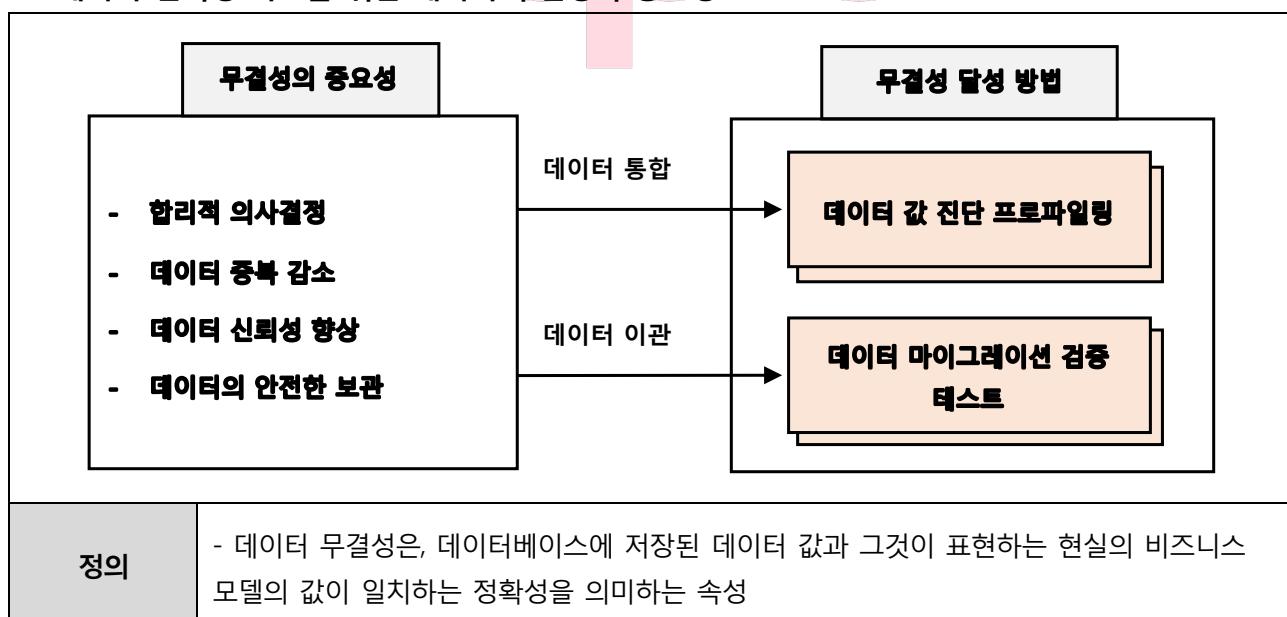
5. 데이터 통합 및 마이그레이션 프로젝트에서 데이터 무결성 목표를 달성하기 위해서는 데이터들의 정합성을 확보하고 신뢰도를 높이는 일이 매우 중요하다. 다음의 내용을 설명하시오.

## 문제

- 가. 데이터 무결성(Integrity)과 정합성(Consistency)의 차이  
 나. 데이터 값(Value) 진단 프로파일링(Data Profiling)의 중점 분석 관점  
 다. 데이터 마이그레이션 검증 테스트 방법

출제영역	데이터베이스	난이도	★★★☆☆
출제배경	- 데이터 무결성과 신뢰성 확보 방안은 고전 토픽이지만 정형 데이터에서는 매우 중요한 토픽으로 이를 정확히 알고 있는지 묻는 문제		
출제번호	- 83회/관리/1		
참고자료	<ul style="list-style-type: none"> <li>- softwaretestingportal.com 의 데이터 마이그레이션 튜토리얼 동영상</li> <li>- 인공지능 학습용 데이터 품질관리 가이드라인</li> </ul>		
Keyword	- 신뢰성도, 프로파일링 분석(누락값, 값의 허용범위, 허용 값, 문자열 패턴, 날짜, 유일값, 구조), 데이터 확인, 마이그레이션 런 테스트		
풀이	백기영 (126회 정보관리기술사)		

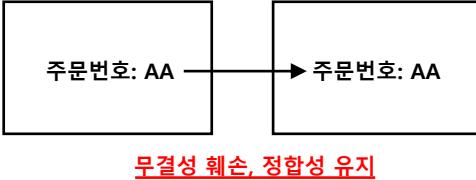
## 1. 데이터 신뢰성 확보를 위한 데이터 무결성의 중요성



- 무결성을 달성하기 위해서는 기본적으로 데이터의 일치성을 유지하는 정합성을 최우선 고려 필요.

## 2. 데이터 무결성과 정합성의 차이 및 데이터 값 진단 프로파일링의 중점 분석 관점 설명

### 가. 데이터 무결성과 정합성의 차이 상세 설명

구분	무결성		정합성			
개념도	조건) 올바른 주문번호: 11  <b>무결성 유지, 정합성 유지</b>		조건) 올바른 주문번호: 11  <b>무결성 훼손, 정합성 유지</b>			
개념	<ul style="list-style-type: none"> <li>- 데이터 값이 정확하고 완전한 상태</li> </ul>		<ul style="list-style-type: none"> <li>- 데이터의 올바른 유무와 상관없이 데이터들의 값이 서로 일치</li> </ul>			
공통점	<ul style="list-style-type: none"> <li>- 데이터의 값이 일관되게 유지되는 성질</li> <li>- 무결성을 유지하기 위해서는 필히 정합성도 같이 유지</li> </ul>					
차이점	<ul style="list-style-type: none"> <li>- 무결성은 올바른 데이터가 일관되게 유지되지만 정합성은 틀린 데이터도 일관되게 유지</li> <li>- 무결성은 데이터 품질의 필수 요소, 정합성은 무결성을 위한 속성</li> </ul>					

- 정형 데이터를 다루는 RDBMS에서 무결성은 데이터 품질을 보장하는 가장 큰 목표.

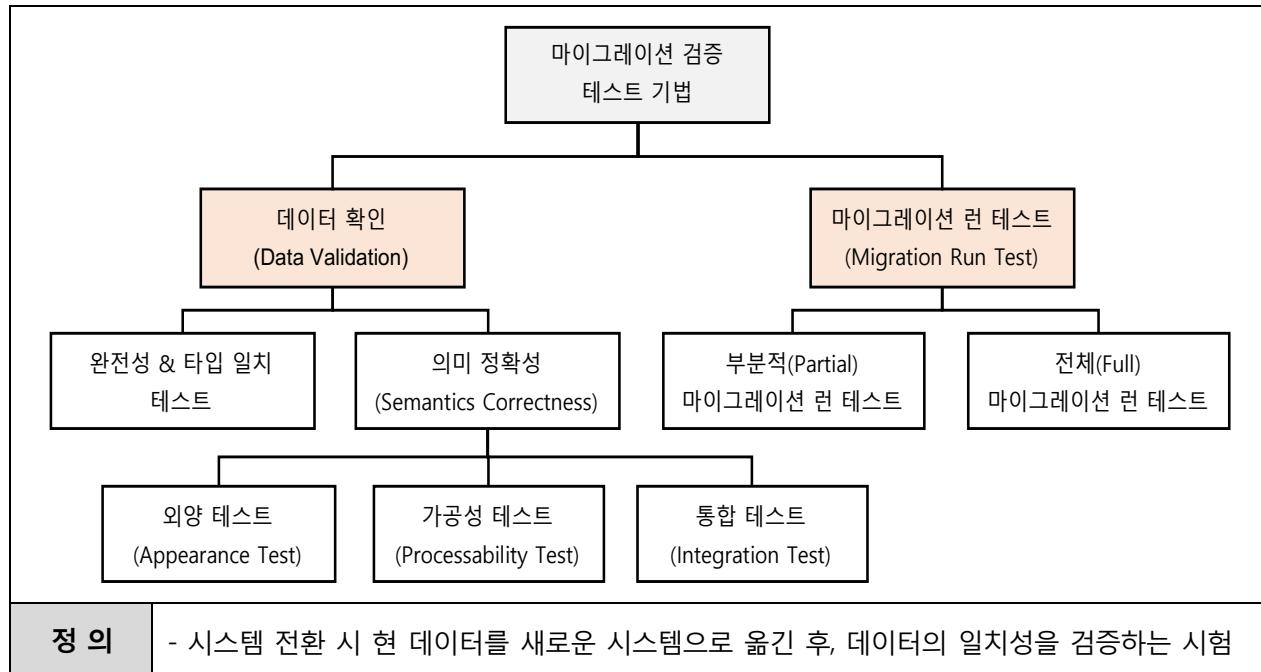
### 나. 데이터 값 진단 프로파일링의 중점 분석 관점 상세 설명

구분	설명	
정의	<ul style="list-style-type: none"> <li>- 메타데이터(설계상)와 실제 데이터(DB)에 대하여 통계적 분석을 시행해 데이터 품질 문제를 확인하는 기법</li> </ul>	
중점 분석 관점	누락 값 분석	<ul style="list-style-type: none"> <li>- 반드시 존재해야 하는 값의 누락이 발생된 컬럼을 도출</li> </ul>
	값의 허용범위 분석	<ul style="list-style-type: none"> <li>- 컬럼의 속성 값이 갖는 범위내에 실제 값이 존재하는지의 여부를 파악하고 오류 유형을 도출</li> </ul>
	허용 값 목록 분석	<ul style="list-style-type: none"> <li>- 해당 컬럼의 허용값 목록(코드)에 포함되지 않는 값을 분석</li> </ul>
	문자열 패턴 분석	<ul style="list-style-type: none"> <li>- 컬럼 속성 값의 특성을 각 컬럼 속성마다의 일정한 패턴형식을 미리 도출하여 분석</li> </ul>
	날짜유형 분석	<ul style="list-style-type: none"> <li>- DBMS에서 제공하는 DATE 유형을 사용하는 경우와 문자형 날짜패턴을 적용하여 값을 갖는 경우 분석</li> </ul>
	유일값 분석	<ul style="list-style-type: none"> <li>- 업무적 의미에서 유일해야 하는 컬럼에 중복이 발생되었는지의 여부를 파악</li> </ul>
	구조 분석	<ul style="list-style-type: none"> <li>- 구조 결함으로 인한 일관되지 못한 데이터를 발견하는 분석 기법</li> </ul>

- 데이터 통합 후 데이터 프로파일링을 실시하고 데이터 이관 후에는 검증 테스트를 통해 무결성 확보.

### 3. 데이터 마이그레이션 검증 테스트 방법 설명

#### 가. 데이터 마이그레이션 검증 테스트 체계도



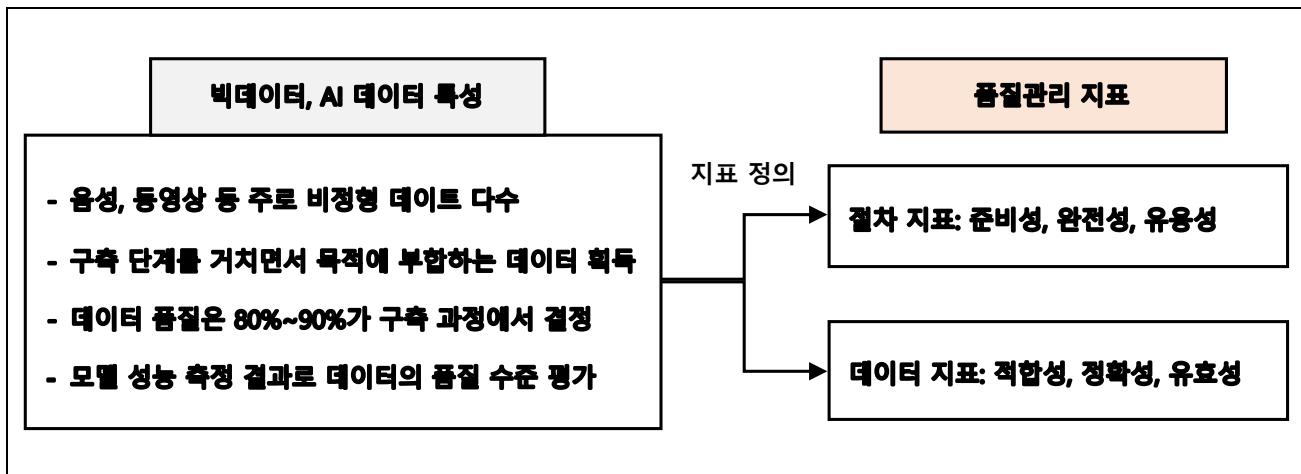
- 데이터 마이그레이션 검증 테스트는 레거시 또는 현 시스템의 데이터를 새로운 타겟 시스템으로 마이그레이션 하는 프로세스를 검증

#### 나. 데이터 마이그레이션 검증 테스트 방법 상세 설명

구분	상세 기법	설명
데이터 유효성 테스트 (Data Validation Tests)	완전성 테스트 (Completeness Tests)	- 타겟 데이터베이스에서 누락된 오브젝트를 식별
	외양 테스트 (Appearance Tests)	- GUI 수준에서 오브젝터의 외양에 주안점을 둠
	통합 테스트 (Integration Tests)	- 마이그레이션 후에 애플리케이션 간의 연결성 및 링킹이 제대로 작동하는지 확인
	프로세싱 테스트 (Processing Tests)	- 마이그레이션된 데이터를 처리하고, 타겟 애플리케이션과 불러온 데이터의 성공적인 연결성을 보장
마이그레이션 실행 테스트 (Migration Run Tests)	부분 마이그레이션 테스트 (Partial Migration Tests)	- 적은 수의 비즈니스 오브젝트를 마이그레이션 하여 시험 마이그레이션(trial migration) 속도를 향상
	전체 마이그레이션 테스트 (Full Migration Tests)	- 전체 데이터 세트로 모든 마이그레이션 프로그램을 실행하는 테스트

- 정형 데이터 관점에서 무결성은 주요 품질 지표이지만 빅데이터, 인공지능 관점에서는 비정형/반정형 데이터가 대부분 이기에 데이터 신뢰도 향상을 위한 다른 품질 지표가 필요 함.

#### 4. 빅데이터, 인공지능 측면에서 데이터 신뢰도 향상을 위한 품질지표



- 고품질의 빅데이터, 인공지능 학습용 데이터 확보를 위해서는 데이터의 품질관리체계를 확보하는 것이 중요.

"끝"



#### 기출풀이 의견

5. 정형데이터, RDBMS 에서 데이터 무결성을 통한 신뢰도 향상은 매우 중요한 개념입니다. 답안 작성 시, 구체적인 키워드와 더불어 실무 사례를 작성하시면 좋은 점수를 받을 수 있습니다.

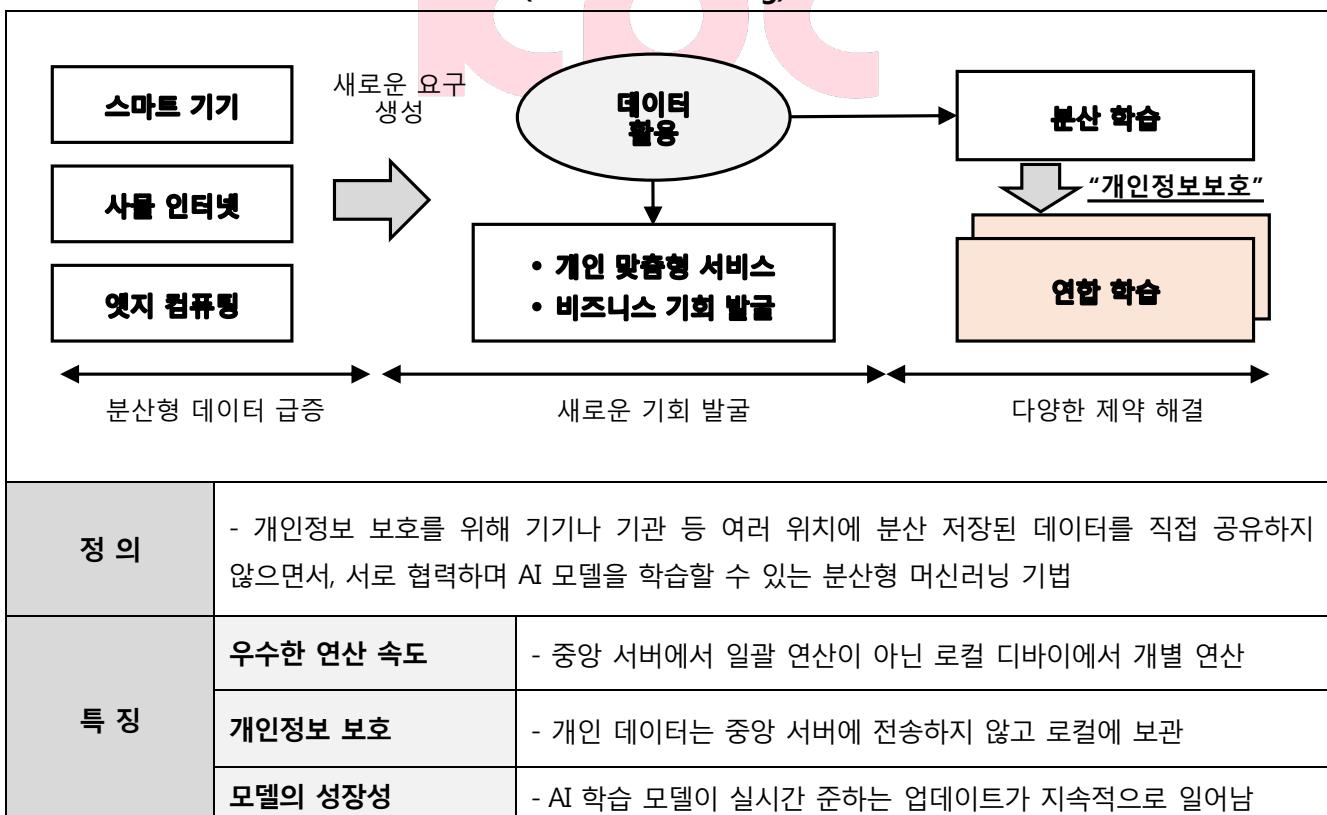
**6. 개인정보 보호를 위한 분산 인공지능(AI, Artificial Intelligence) 학습 모델인 연합학습(Federated Learning)에 대하여 아래 사항을 설명 하시오.**

**문제**

- 가. 연합학습의 동작 원리**
- 나. 연합학습의 주요 알고리즘**
- 다. 연합학습의 보안 및 프라이버시 보장형 기술**

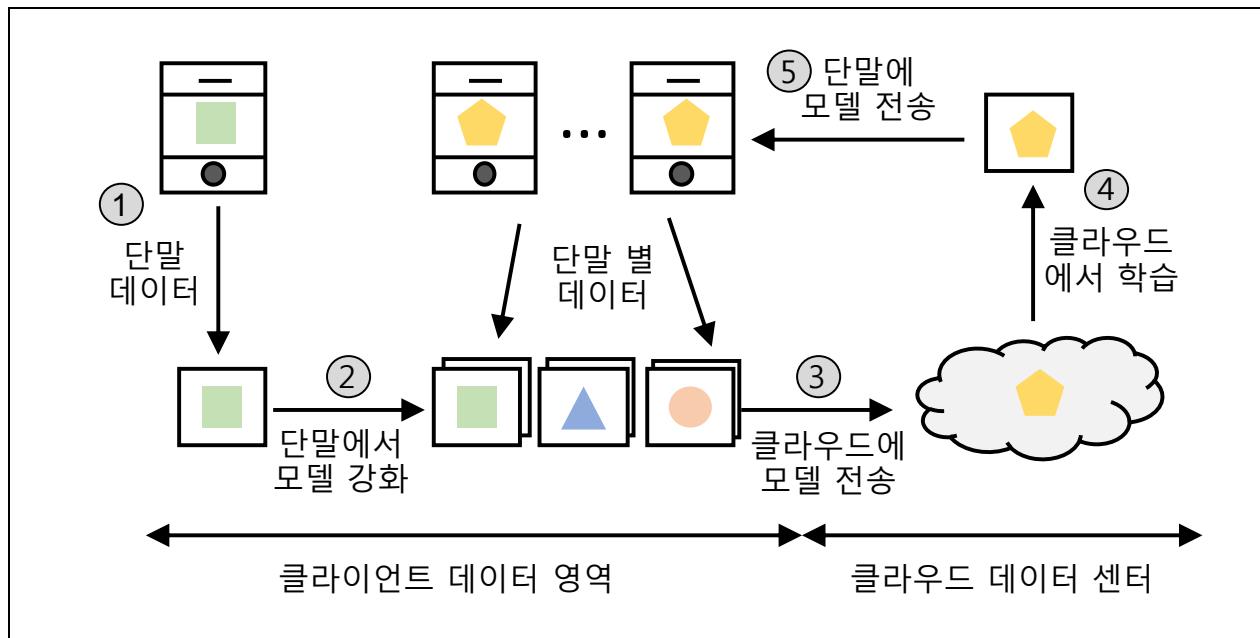
출제영역	인공지능	난이도	★★★☆☆
출제배경	- 분산형 환경에서 AI 학습 시, 성능과 개인정보 보호가 중요하여 묻는 문제		
출제빈도	- 모의_2021.12(공통 1), 모의_2021.05(공통 3), 합숙_2022.04		
참고자료	<ul style="list-style-type: none"> <li>- 연합학습 기술 동향 및 산업적 시사점 (기술정책 트렌드 2020-06)</li> <li>- Federated Learning in Mobile Edge Networks (<a href="https://jihyeonryu.github.io/2021-04-21-survey-paper2/">jihyeonryu.github.io/2021-04-21-survey-paper2/</a>)</li> </ul>		
Keyword	<ul style="list-style-type: none"> <li>- FedSGD, FedAvg, FedBCD, FedTrans, HierFaVG, 차등정보보호(Differential Privacy), 동형암호(Homomorphic Encryption), 안전한 다자간 계산(Secure Multi-Party Computation)</li> </ul>		
풀이	백기영 (126 회 정보관리기술사)		

**1. 분산 인공지능 학습 모델인 연합학습(Federated Learning)의 등장배경**



## 2. 연합학습의 동작 원리 및 주요 알고리즘 설명

### 가. 연합학습의 동작 원리의 상세 설명



- 단말은 생성한 로컬 AI 모델의 결과값(파라미터)을 압축 및 암호화하여 클라우드 서버로 전송, 개선된 글로벌 AI 모델은 다시 개인 스마트폰에 전송되어 기존 모델을 업데이트 함으로써 점점 예측력이 높은 로컬 모델이 만들어짐.

### 나. 연합학습의 주요 알고리즘

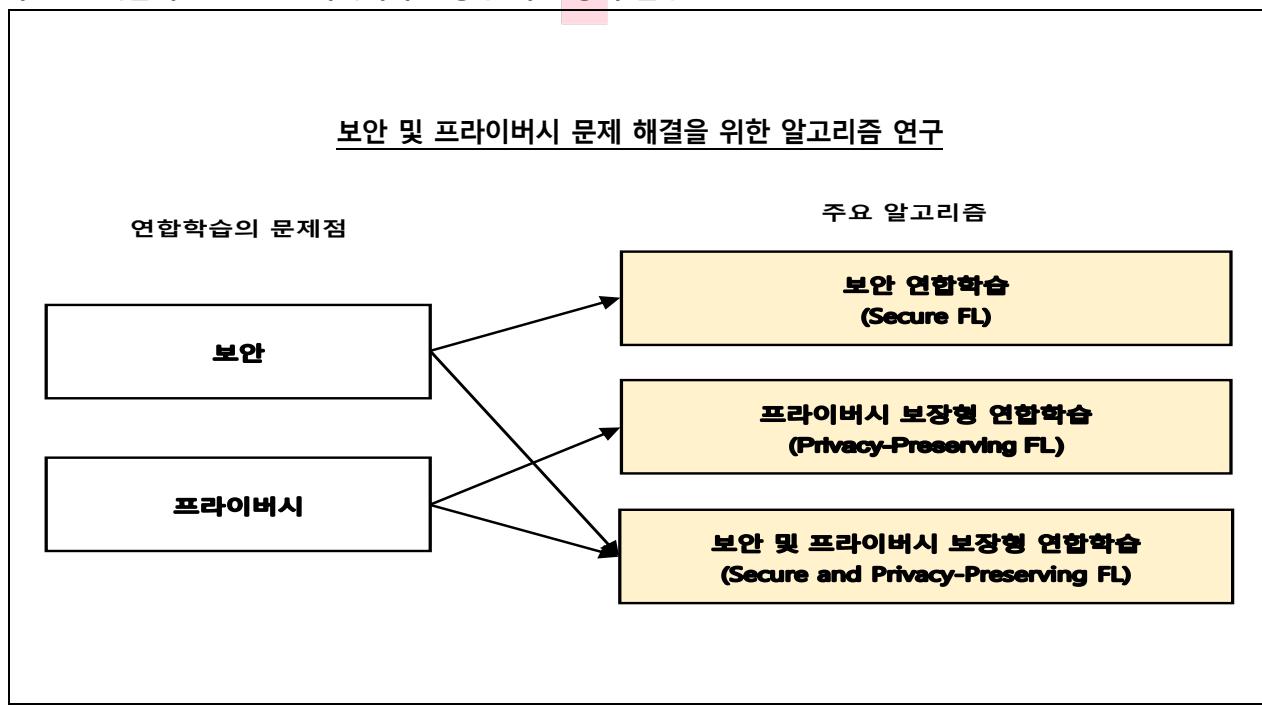
구분	설명		
주요 알고리즘 개념도	<p><u>주요 알고리즘: "FedSGD, FedAVG"</u></p> <p style="text-align: center;">서버</p> <p style="text-align: center;"><math>x_{new} = (x_1 + x_2 + \dots + x_n) / n</math></p> <p style="text-align: center;"><math>x_{old} = x_{new}</math></p> <p style="text-align: center;">1 Epoch</p> <p style="text-align: center;">Batch size      데이터셋      “FedAVG” K회 학습</p>		
주요 알고리즘	<b>FedSGD</b> <b>(Federated Stochastic Gradient Descent)</b>	<ul style="list-style-type: none"> <li>- 모든 참가자가 참여를 하며, 각 트레이닝 라운드마다 오직 하나의 pass만(1 step training)을 수행</li> <li>- 이는 마치 minibatch 사이즈가 각 참가자의 데이터셋 크기와 같은 것을 의미</li> <li>- 이것은 centralized DL에서 full-batch 학습과 비슷</li> </ul>	

	<b>FedAvg (Federated Averaging)</b>	- 각 단말이 일정한 횟수 K만큼 반복적으로 학습을 수행한 후의 파라미터 값을 서버로 전달하는 방식 - 각 단말에서 batch size 크기로 나눠서 학습하여 minibatch 효과를 줌으로써 글로벌 파라미터가 수렴에 이르는 시간을 단축
	<b>FedBCD (Federated Stochastic Block Coordinate Descent)</b>	- 각 참가자들은 global aggregation을 위한 통신 전에 multiple한 local update를 수행 - 게다가 각 통신마다 approximate 보정이 적용되면서 convergence가 보증
	<b>FedTrans (Federated Transfer)</b>	- 통신 비용을 줄이는 방법은 수렴 속도를 증가시키기 위해 학습 알고리즘을 변경 - transfer learning과 domain adaptation에서 공통적으로 사용하는 two-stream 모델을 사용
	<b>HierFaVG (Hierarchical FL)</b>	- edge server가 local model의 파라미터들을 집계 - 미리 정의된 수의 edge server 집계 수가 만족되면 edge server는 cloud와 통신하여 global model aggregation을 수행

- 최근 연구에 따르면 앞서 설명한 연합학습 알고리즘만으로는 프라이버시 보호로부터 완전히 자유로울 수는 없다는 사실들이 증명, 이러한 프라이버시 및 보안 문제를 해결하기 위해 최근 다양한 알고리즘 연구가 진행.

### 3. 연합학습의 보안 및 프라이버시 보장형 기술 설명

#### 가. 연합학습의 보안 및 프라이버시 보장형 기술 상세 설명



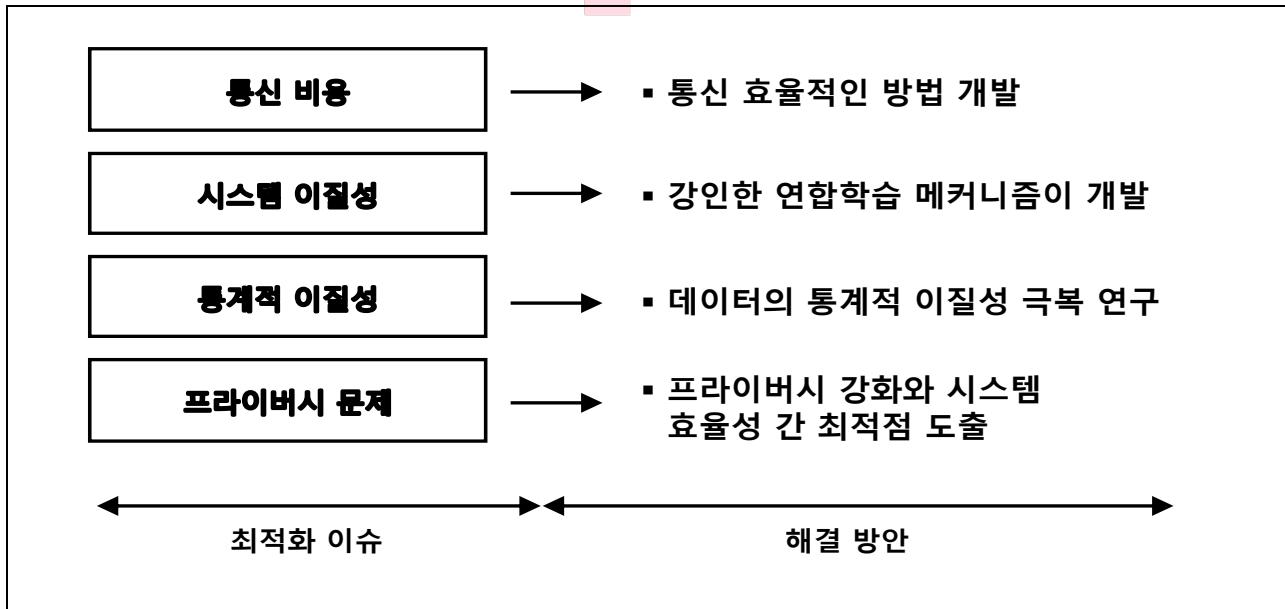
- 연합학습에서 개인 정보 보호를 위해 노이즈를 추가하거나 암호화된 결과로 연산하는 등 다양한 기술이 존재.

## 나. 연합학습의 보안 및 프라이버시 보장형 기술 상세 설명

주요 기술	개념도	설명
차등정보보호 (Differential Privacy)		<ul style="list-style-type: none"> <li>- 하나의 개인정보가 전체 자료에 추가로 포함될 때 증가하는 노출 위험을 '차등정보보호'라 정의</li> <li>- 원 데이터에 수학적 노이즈 추가</li> <li>- 파라미터에 노이즈를 추가하여 프라이버시 노출을 방지</li> </ul>
동형암호 (Homomorphic Encryption)		<ul style="list-style-type: none"> <li>- 암호화된 데이터를 복호화 없이도 연산할 수 있는 암호기술</li> <li>- 각 단말은 자신의 데이터를 동형 암호화</li> <li>- 동형 암호화된 값을 서버로 전달</li> <li>- 서버는 각 단말에서 전달 받은 암호값을 연산 후 복호화</li> </ul>
안전한 다자간 계산 (Secure Multi-Party Computation)		<ul style="list-style-type: none"> <li>- 동형암호와 유사</li> <li>- 각 단말에서 서버로 전달하는 원래의 값을 노출하지 않고 전체 합을 알 수 있음</li> <li>- 가장 대표적인 다자간 계산 알고리즘으로 Secure Aggregation</li> </ul>

- 연합학습은 기존 분산 학습과 유사한 개념이지만 이와 다른 분산 최적화 이슈가 존재하며 해결 방안이 필요.

## 4. 연합학습의 분산 최적화 이슈 및 해결 방안



- 분산 최적화 이슈 외 사이버 공격, 공정성(Fairness), 편향성(Bias) 문제 또한 고려해야 됨.

"끝"



### 기출풀이 의견

6. 연합학습은 최근 인공지능 분야에서 개인정보 보호를 위해 활발히 연구되고 있습니다. 연합학습의 등장배경, 필요성 부터 상세한 키워드를 작성하시면 좋은 점수를 받으실 수 있습니다.