

정보시스템 프로젝트 관리

한은정 교수



학습 목표

1. 정보시스템 도입 프로젝트의 도전과제를 이해하고, 정보시스템 도입 전략의 수립 방법을 알 수 있다.
2. 전통적인 개발 방법론인 시스템 개발 수명주기(SDLC)를 이해하고, 다양한 정보시스템 개발 방법론을 비교할 수 있다.
3. 정보기술 도입에 따른 변화관리 필요성과 조직구조의 변화에 대해 이해할 수 있다.
4. 디지털 트랜스포메이션의 필요성을 이해한다

정보시스템 도입에 따른 도전과제

- ❖ 요구사항 결정의 어려움
- ❖ 요구사항의 잦은 변경
- ❖ 일정 및 예산관리의 어려움
- ❖ 변화하는 기술
- ❖ 규모의 비경제

정보시스템 도입 전략

- ❖ 정보시스템의 도입은 비즈니스의 우선순위에 따라 효율적이고 순차적으로 이루어져야 한다
- 1) 비즈니스 전략 계획: 비즈니스의 전반적 방향 설정
- 2) 정보화 전략 계획(ISP; information strategic planning):
비즈니스를 지원하기 위해 요구되는 정보시스템이 무엇인지 정의하고, 중장기적 관점에서 정보시스템 도입의 우선순위 결정
- 3) 정보시스템 전략 계획: 우선순위에 의해 선정된 개별 정보시스템의 구축과 운영에 관련된 상세 계획안

정보시스템 도입 전략

❖ 정보시스템 전략 계획의 구성요소

- 정보시스템의 기능 계획
 - 조직의 목표 달성을 위한 정보시스템 요구사항 결정
 - 예) 내부 업무 프로세스 통합을 위한 ERP 시스템의 기능 정의
- 정보기술 요소 계획
 - 정보시스템 구현을 위한 정보기술 인프라 요소의 이용
 - 예) ERP시스템 구현을 위해 통합 DB, 서버, 소프트웨어 개발
- 정보시스템 운영방안 계획
 - 정보시스템의 구축 및 운영을 위한 조직의 구성과 정보시스템 관리 지침
 - 예) ERP시스템 유지보수를 위한 내부 IT조직의 운영 방안

정보시스템 개발 관리

❖ 시스템 개발(systems development)

- 정보시스템을 개발하고 유지하는 과정
- 정보시스템 개발은 IT 전문가만의 기술적인 영역이 아니라, IT 전문가와 업무지식을 지닌 비즈니스 관리자의 공동 협업이 필요한 분야이다

❖ 시스템 개발 프로세스

- 시스템 개발은 여러 단계적 활동들로 이루어진 구조화된 문제해결 기법이다. 이러한 활동들은 시스템 분석, 시스템 설계, 프로그래밍, 테스트, 전환, 그리고 가동과 유지보수 등으로 구성된다.

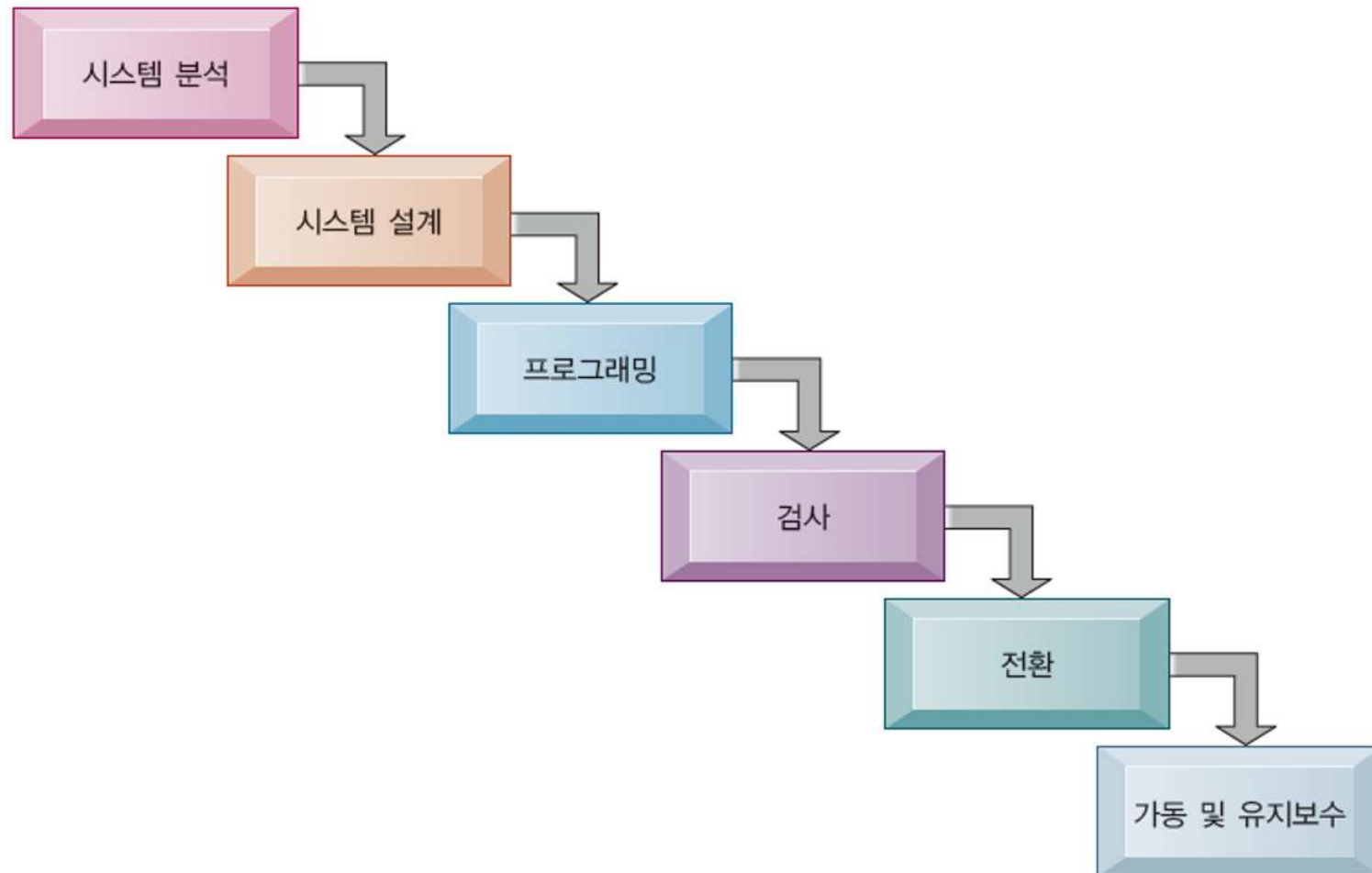
전통적 정보시스템 개발 방법론

❖ 시스템 개발 수명주기(SDLC; systems development life cycle)

- 순차적 단계를 거쳐 시스템을 개발하는 전통적 개발 방법
- 한 단계의 작업이 끝나야 다음 단계의 작업이 시작되는 폭포수(waterfall) 접근법
- 큰 규모의 정보시스템 개발 시에 개발 과정을 엄격히 통제하기 위해 주로 사용
- 각 단계의 산출물은 다음 단계의 기초 입력 자료가 된다

전통적 정보시스템 개발 방법론

❖ SDLC의 시스템 개발 절차



전통적 정보시스템 개발 방법론

1) 시스템 분석(system analysis)

- 시스템 분석은 조직이 정보시스템을 이용해 해결하고자 하는 문제를 분석해서 무엇을 해야 하는지를 파악하는 것이다.
- 정보시스템이 수행해야 하는 기능을 결정하여 이를 문서화한다
- 시스템 요구사항 정의 방법: 사용자 인터뷰/설문지, 비즈니스 문서 검토, 비즈니스 프로세스 분석, 데이터 정의

전통적 정보시스템 개발 방법론

2) 시스템 설계(system design)

- 요구사항 분석 단계에서 도출된 기능을 구현하기 위해 필요한 세부적인 구성요소를 설계하는 단계로서, 정보시스템 구현에 필요한 모든 기술적, 조직적, 인적 요소들을 포함한다.
 - ① 하드웨어의 사양 및 획득 방법 결정: 구매, 대여, 클라우드
 - ② 소프트웨어 획득 방법 결정: 기성 소프트웨어 도입, 기성 소프트웨어의 맞춤 수정, 소프트웨어 직접 개발
 - ③ 데이터베이스 설계: 관계형 DB 설계 및 물리적 데이터베이스 구현
 - ④ 프로세스(절차) 설계: 업무 수행을 위해 시스템을 사용하는 절차 정의
 - ⑤ 직무 설계: 사용자 직무명세 개발

전통적 정보시스템 개발 방법론

3) 프로그래밍(programming)

- 설계 단계에서 만들어진 시스템 명세서는 프로그래밍 단계를 거쳐 소프트웨어 프로그램 코드로 전환된다. 최근에는 많은 기업들이 새로운 시스템 개발을 위해 내부에서 프로그래밍 하기 보다는 상업용 소프트웨어 패키지를 도입하거나 외부에 아웃소싱을 맡기는 경우가 많다.

4) 테스트(test)

- 정보시스템의 정보처리 결과가 올바른지 테스트(test)하는 과정이 필요하다. 프로젝트 계획을 수립할 때 테스트에 소요되는 시간을 충분히 산정하여 테스트 데이터를 신중하게 준비하고 결과를 분석해야 하며, 필요한 경우 시스템을 수정해야 한다.

전통적 정보시스템 개발 방법론

5) 전환(conversion)

- 전환은 기존 시스템에서 새로운 시스템으로 변환하는 과정이다.
- 전환 방법: 일시 전환, 병렬전환, 단계적 전환, 파일럿 전환

전통적 정보시스템 개발 방법론

❖ 정보시스템 테스트의 유형

- 단위 테스트(unit test): 시스템 내의 각 프로그램에 오류가 없는지 개별적으로 테스트한다.
- 시스템 테스트(system test): 정보시스템의 기능을 총체적으로 테스트하는 것이다. 각 모듈이 전체 시스템에 제대로 통합되어 작동하는지, 시스템의 동작이 원래 설계했던 것과 차이가 없는지 등을 파악한다. 시스템의 수행 시간, 파일 저장 및 처리 능력, 최대 부하, 복구 및 재시동 능력 등을 점검한다.
- 인수 테스트(acceptance test): 시스템이 실제 운영 환경에서 사용될 준비가 되었는지 최종적으로 확인한다. 사용자에게 의해 시스템 운영 환경이 평가되고 관리자에 의해 최종 점검된다. 모든 관계자가 새로운 시스템의 사용을 승인하면 정식으로 인수된다.

전통적 정보시스템 개발 방법론

❖ 시스템 전환의 유형

- 일시 전환: 기존 시스템의 사용을 일시에 완전히 중단하고 새로운 시스템으로 대체하는 전환 방법. 가장 간단하고 비용과 시간이 적게 드는 방법이지만 새로운 시스템에 문제가 있을 경우 업무가 마비되는 위험이 있다.
- 병렬 전환: 새로운 시스템의 성능이 확인될 때까지 기존 시스템과 병행해 사용하는 유형으로, 가장 위험이 적은 방법이지만 번거롭고 비용이 만하이 든다.
- 단계적 전환: 새로운 시스템을 여러 개의 하위 모듈로 나누고 단계적으로 한 부분씩 가동하여 새로운 시스템의 적용 범위를 점진적으로 확대하는 방식. 가동하는 모듈이 다른 부분과 연관될 때는 복잡한 문제가 발생할 수 있다.
- 파일럿 전환: 새로운 시스템을 한 부서 또는 한 운영 단위 등과 같이 조직의 일부 영역에만 도입하고, 파일럿 버전이 문제없이 작동되면 나머지 조직에 대해서도 새로운 시스템을 사용하게 한다.

전통적 정보시스템 개발 방법론

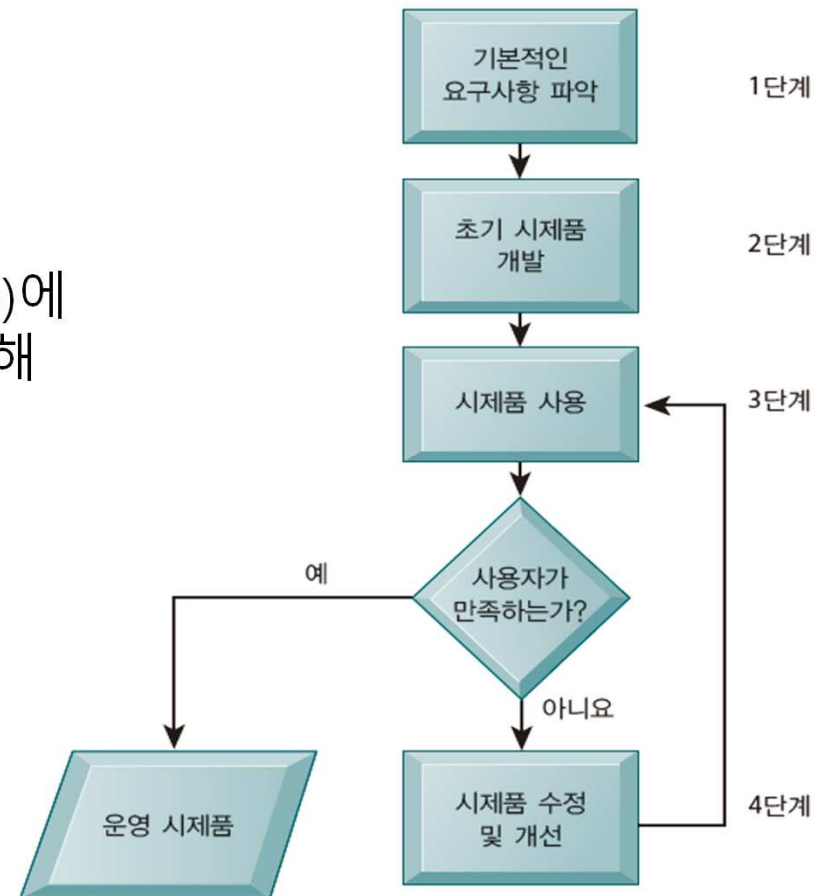
❖ SDLC의 문제점

- 개발 주기가 길어서 많은 비용과 시간 소요
- 비즈니스 요구사항 변화에 대한 유연성 부족
- 정보시스템 개발 과정에서 최종 사용자의 요구사항이 제대로 충족되었는지 확인 어려움
- 분석 중심의 개발 방법으로 각 단계마다 방대한 양의 문서가 생성된다
 - ❖ CASE(computer-aided software engineering) 도구: 정보시스템의 구조를 설계하기 위해 필요한 각종 차트와 다이어그램을 자동 생성을 지원하는 소프트웨어

다양한 정보시스템 개발 방법론

❖ 프로토타이핑(prototyping)

- 저렴한 비용으로 신속하게 실험 시스템을 만들어 사용자의 평가를 받는 접근법으로, 시제품(prototype)에 대한 사용자들의 의견을 받아 개선해 나가는 것이 목적이다.



다양한 정보시스템 개발 방법론

❖ 최종사용자 개발(end-user development)

- 최종사용자들이 기술 전문가의 도움을 거의 받지 않고 간단한 정보시스템을 개발하는 방법
- 최종사용자들은 사용자 친화형 질의어, 보고서, 웹사이트 개발, 그래픽 도구 등을 이용하여 데이터에 접근하고 보고서를 생성하면 간단하게 애플리케이션을 개발할 수 있다
- 장점: 최종사용자 개발 방식은 전통적인 시스템개발 수명주기보다 빠르게 시스템 개발을 수행할 수 있다. 사용자가 직접 참여하여 시스템을 개발하므로 만족도가 높아진다.
- 단점: 최종사용자 소프트웨어 도구는 대형 시스템이나 복잡한 데이터 처리 절차를 구현하지는 못한다. 또한 최종사용자 개발 방식은 조직의 IT부서 통제 범위를 벗어나서 외부에서 이루어질 수 있기 때문에 정보 유출 등의 위험이 있을 수 있다.

다양한 정보시스템 개발 방법론

❖ 구매 및 아웃소싱

- 최근에는 많은 기업들이 소프트웨어를 개발하지 않고 상용화된 소프트웨어 패키지나 클라우드 서비스로서의 소프트웨어(SaaS; software as a service)를 도입한다.
- 패키지화된 소프트웨어는 다양한 고객의 요구사항을 충족하기 어려운 경우가 많고, 조직 내부에서 설계 프로세스에 대한 완전한 통제권을 갖지 못한다. 대신 사용자의 요구사항을 최대한 커스터마이징(customizing; 고객화)할 수 있어야 한다.
- 기업이 내부적으로 정보시스템 개발을 위한 충분한 자원이 없는 경우 외부의 전문 개발 조직에 아웃소싱을 줄 수 있다.

다양한 정보시스템 개발 방법론

❖ 애자일 개발(agile development)

- 반복적이고 지속적인 피드백과 사용자 참여를 통해 시스템을 빠르게 가동시키는 것이 목적이다
- 큰 프로젝트를 작은 하위 프로젝트로 분할하여 시스템 기능을 개발한다. 각 미니 프로젝트는 마치 완전히 독립적인 프로젝트인 것처럼 개발하고 외부에 주기적으로 공개된다
- 요구사항을 명확히 함으로써 다음 반복 기간 내에 새로운 기능의 개선 또는 추가가 이루어진다.
- 테스트는 개발 과정 전반에 걸쳐 조기에 수시로 이루어진다.

다양한 정보시스템 개발 방법론

❖ 데브옵스(DevOps)

- 데브옵스는 신속하고 민첩한 개발 관행을 더욱 촉진하는 문화와 환경을 조성하기 위한 조직 전략으로 애자일 개발을 지향
- 데브옵스는 개발(development)과 운영(operations)을 의미하며, 애플리케이션을 개발하는 소프트웨어 개발자와 애플리케이션을 실행하고 유지하는 IT 운영 직원 간의 긴밀한 협력을 강조한다
 - 소프트웨어가 가동된 이후에는 운영팀이 시스템의 유지보수를 전담하여 운영상의 문제를 개발팀이 알지 못하는 경우가 있으므로 개발 그룹과 운영 그룹 간의 활발한 커뮤니케이션과 협업 분위기를 조성해야 한다
- 데브옵스는 기술의 민첩한 변화, 표준화된 프로세스, 보다 강력한 자동화 도구를 이용해 조직을 변화시키고, 이를 통해 더욱 빠른 소프트웨어 개발과 변경이 가능해진다

다양한 정보시스템 개발 방법론

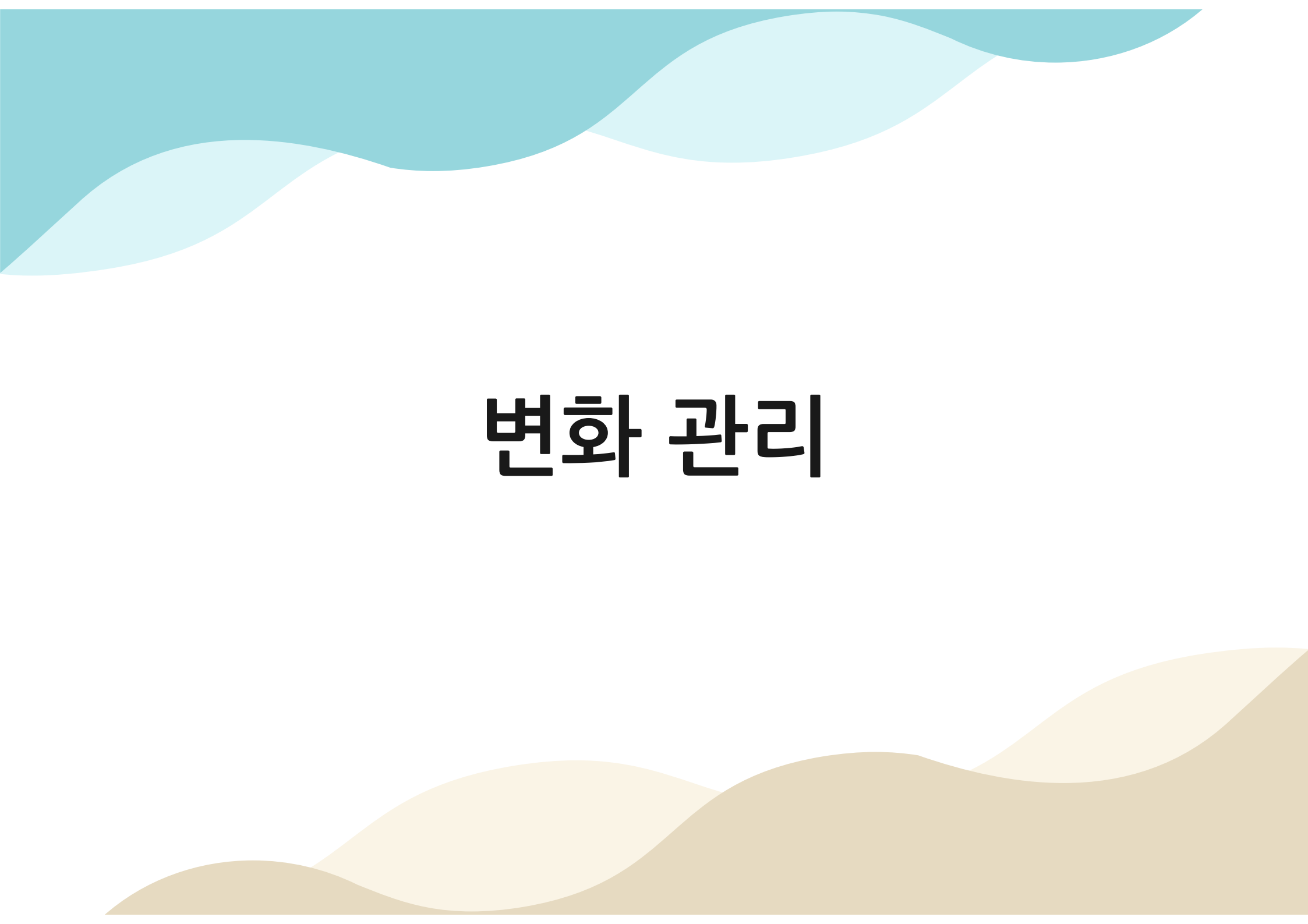
❖ 컴포넌트 개발(component-based development)

- 이미 만들어 놓은 소프트웨어 기능을 재사용하여 애플리케이션을 만드는 방식이다. 즉, 여러 컴포넌트를 조립하여 시스템을 개발한다.
- 최근에는 클라우드를 통해 다양한 컴포넌트가 공개되어 기업의 전자상거래 응용 프로그램 개발에 적극 활용되고 있다.

다양한 정보시스템 개발 방법론

❖ 모바일 서비스의 개발

- 모바일 웹사이트: 일반 웹사이트를 작은 모바일 기기에서 검색하기 용이하도록 콘텐츠와 네비게이션을 축소시킨 버전
- 웹 앱(web app): 모바일 기기 전용 기능을 갖춘 인터넷 지원 앱. 사용자는 모바일 기기에 별도의 앱을 설치하지 않고도 웹 브라우저를 통해 서비스에 접속할 수 있다.
- 네이티브 앱(native app): 특정 모바일 디바이스와 기기에서 실행되도록 설계된 독립형 애플리케이션이다. 네이티브 앱은 모바일 기기에 직접 설치된다. 네이티브 앱은 인터넷에 연결되어 데이터를 다운로드하고 업로드할 수 있으며, 인터넷에 연결되지 않은 경우에도 작동된다.
- 반응형 웹(responsive web): 컴퓨터와 모바일 디바이스 모두에서 구현 가능한 웹 설계로서, 접속 기기의 화면 해상도에 따라 웹사이트의 레이아웃이 자동으로 변경된다. HTML5를 사용하여 다양한 디바이스를 지원하는 웹서비스를 만들 수 있다.



변화 관리

정보기술과 변화관리

❖ 정보시스템 도입에 따른 문제점

- 사용자의 저항
- 조직문화와의 충돌
- 레거시 시스템과의 통합
- 새로운 시스템으로의 이동
- IT 조직의 운영

정보기술과 변화관리

❖ 변화관리(change management)란?

- 정보시스템 도입에 따른 변화를 조직 전체 차원에서 관리하는 것
- 변화관리의 주요 차원: 사람, 프로세스, 기술

❖ 변화관리의 필요성

- 정보시스템의 새로운 도입이나 변경은 조직에 큰 영향을 미친다
- 조직의 자원을 관리하기 위한 정보 접근 및 사용 방식이 변화된다.
- 조직 내부의 권한과 권력이 변화하면서 저항과 반대에 의해 좋은 시스템이 실패할 수도 있다.
- 따라서, **성공적인 시스템 구축은 면밀한 변화관리가 필요하다.**

변화관리 실행의 고려 요소

❖ 변화관리 에이전트

- 정보시스템의 구현 과정에서 시스템분석가는 변화 에이전트 역할을 한다.
- 시스템의 구성, 상호작용, 과업 활동, 그리고 조직의 구조적 변화 등을 재정립하며, 조직이 정보시스템에 의한 변화를 받아들일 수 있도록 촉진한다. *ex) 학습*

❖ 최종사용자의 참여

- 사용자-설계자 의사소통 격차: 사용자와 정보시스템 전문가 사이에 서로 다른 배경과 이해관계 및 의사결정 방식의 차이
- 최종사용자의 참여가 높을수록, 사용자 요구에 맞는 시스템이 만들어지고 사용자들이 시스템을 수용할 가능성이 증대한다.

변화관리 실행의 고려 요소

❖ 경영진의 지원과 참여

- 경영진의 관심과 적극적인 지원은 사용자와 정보기술 서비스 직원 모두에게 긍정적으로 인식된다.
- 경영진의 지원은 시스템 성공을 위한 충분한 자금과 자원을 보장한다.
- 새로운 시스템의 도입에 따른 업무방식의 변화는 조직 개편으로 이어지며, 이때 경영진의 지원이 필수적이다.
- 경영진의 지원은 정보시스템 프로젝트 성공의 핵심 요소이다.

변화관리 실행의 고려 요소

❖ 공식적 계획과 통제 도구

- 대규모 프로젝트 계획을 문서화하고 통제하기 위한 도구 필요
- 프로젝트 계획의 문서화를 위해 일반적으로 사용하는 방법은 간트 차트와 PERT 차트이다. *둘다 시각화 도구.*
 - 간트차트(Gantt chart): 프로젝트 과업의 시작과 종료 날짜를 시각적으로 표현
 - PERT 차트: 프로젝트 과업들 사이의 상호 의존성과 과업의 순서를 표현

*가면
타스크 수행자* *자원에 인로되어야 하는 타스크 표현*

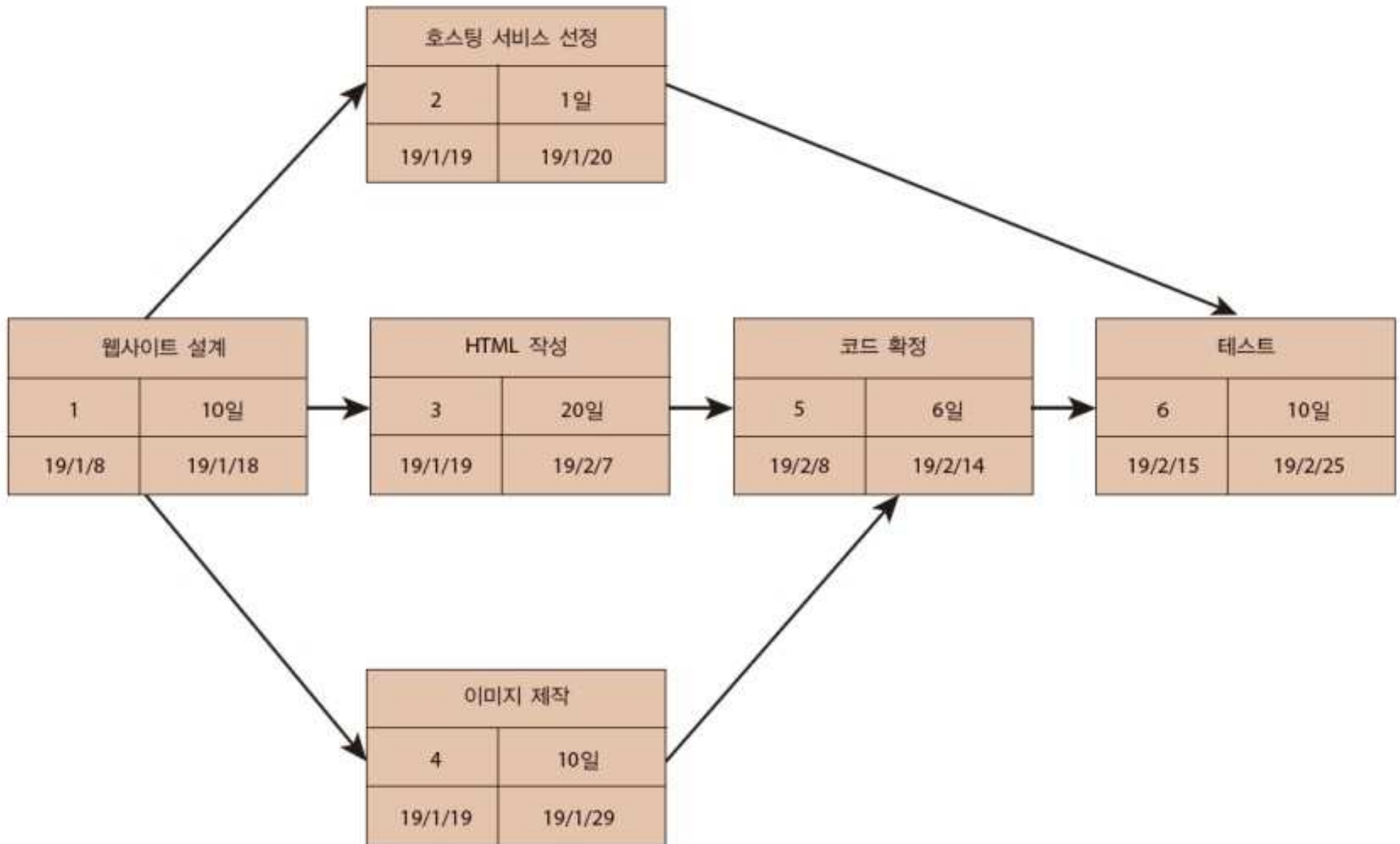
그림 14.4 간트 차트

이 표의 간트 차트는 과업, 일일 필요인력, 각 분야의 책임자와 더불어 각 과업의 시작과 종료 날짜를 보여준다. 자원 요약은 프로젝트 관리자들에게 프로젝트를 성공적으로 관리하기 위해서 프로젝트에 참여하는 개인별, 그리고 월별 필요한 인력의 총량을 요약하여 제공해준다. 여기서 설명된 프로젝트는 데이터 관리 프로젝트이다.

HRIS가 결합된 계획	기 간	담당자	2018			2019												2020		
			10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
데이터 관리 보안																				
QMF 보안 검토/설정	20	EF TP																		
보안 오리엔테이션	2	EF JA																		
QMF 보안 유지보수	35	TP GL																		
데이터 입력 sec. 프로파일	4	EF TP																		
데이터 입력 sec. views est.	12	EF TP																		
데이터 입력 보안 프로파일	65	EF TP																		
데이터 사전																				
오리엔테이션 회의	1	EF																		
데이터 사전 설계	32	EF WV																		
DD prod. coordn-query	20	GL																		
DD prod. coordn-live	40	EF GL																		
데이터 사전 마무리	35	EF GL																		
데이터 사전 유지보수	35	EF GL																		
절차 수정과 설계 준비																				
업무흐름(기존)	10	PK JL																		
급여 데이터 흐름	31	JL PK																		
HRIS P/R 모델	11	PK JL																		
P/R 인터페이스 orient. mtg	6	PK JL																		
P/R 인터페이스 coordn. 1	15	PK																		
P/R 인터페이스 coordn. 2	8	PK																		
복자수당 인터페이스(기존)	5	JL																		
복자수당 인터페이스(새로운 방식)	8	JL																		
복자수당 소통 전략	3	PK JL																		
새로운 업무흐름 모델	15	PK JL																		
Posn. 데이터 입력 흐름	14	WV JL																		
자원 요약																				
에디스 파렐	5.0	EF	2	21	24	24	23	22	22	27	34	34	29	26	28	19	14			
우디 빈튼	5.0	WV	5	17	20	19	12	10	14	10	2							4	3	
찰스 피어스	5.0	CP		5	11	20	13	9	10	7	6	8	4	4	4	4	4			
테드 레우스	5.0	TL		12	17	17	19	17	14	12	15	16	2	1	1	1	1			
토니 폭스	5.0	TC	1	11	10	11	11	12	19	19	21	21	21	17	17	12	9			
패트리샤 크늘	5.0	PC	7	23	30	34	27	25	15	24	25	16	11	13	17	10	3	3	2	
제인 로튼	5.0	JL	1	9	16	21	19	21	21	20	17	15	14	12	14	8	5			
데이비드 홀로웨이	5.0	DH	4	4	5	5	5	2	7	5	4	16	2							
디아나 오닐	5.0	DO	6	14	17	16	13	11	9	4										
조앤 앨버트	5.0	JA	5	6			7	6	2	1				5	5	1				
마리 마르쿠스	5.0	MM	15	7	2	1	1													
돈 스티븐스	5.0	DS	4	4	5	4	5	1												
임시직	5.0	CASL		3	4	3			4	7	9	5	3	2						
케이지 핸디즈	5.0	KM		1	5	16	20	19	22	19	20	18	20	11	2					
안나 보먼	5.0	AB					9	10	16	15	11	12	19	10	7	1				
게일 로링	5.0	GL		3	6	5	9	10	17	18	17	10	13	10	10	7	17			
미지정	0.0	X										9			236	225	230	14	13	
공동부담	5.0	CO		6	4				2	3	4	4	2	4	16			216	178	
임시직	5.0	CAUL								3	3	3								
총 일수			49	147	176	196	194	174	193	195	190	181	140	125	358	288	284	237	196	12

그림 14.5 PERT 차트

소규모 웹사이트를 만들기 위한 간단한 PERT 차트이다. 프로젝트 과업들의 순서와 특정 과업의 이전 과업들과 이후 과업들과의 관련성을 보여준다.



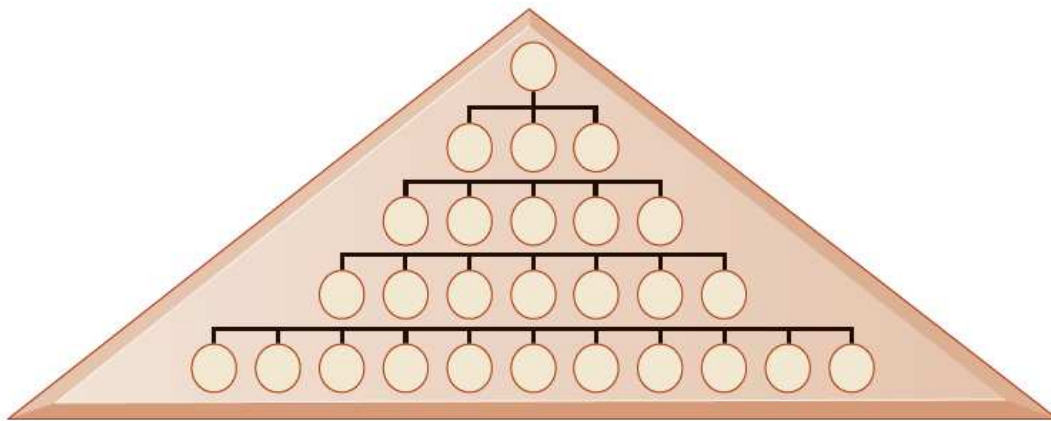
조직구조의 변화

❖ IT에 의한 조직 축소

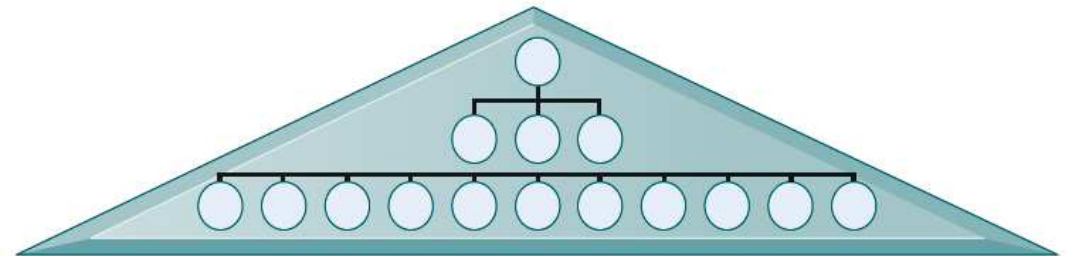
- 정보기술에 의해 정보의 전달 범위와 속도가 빨라지면서 일선 직원들에게 권한을 위임하게 되고 계층구조의 축소를 촉진
- 정보기술은 조직의 의사결정 권한을 하향 이동시킨다
- 빠른 의사결정으로 적은 수의 관리자만으로 관리가 가능해져 관리비용과 계층구조의 효율화

조직구조의 변화

❖ IT에 의한 조직 ~~축소~~ *많은 관리자 필요 X*



전통적 계층 구조 조직



관리층이 제거되어 축소된 조직

조직구조의 변화

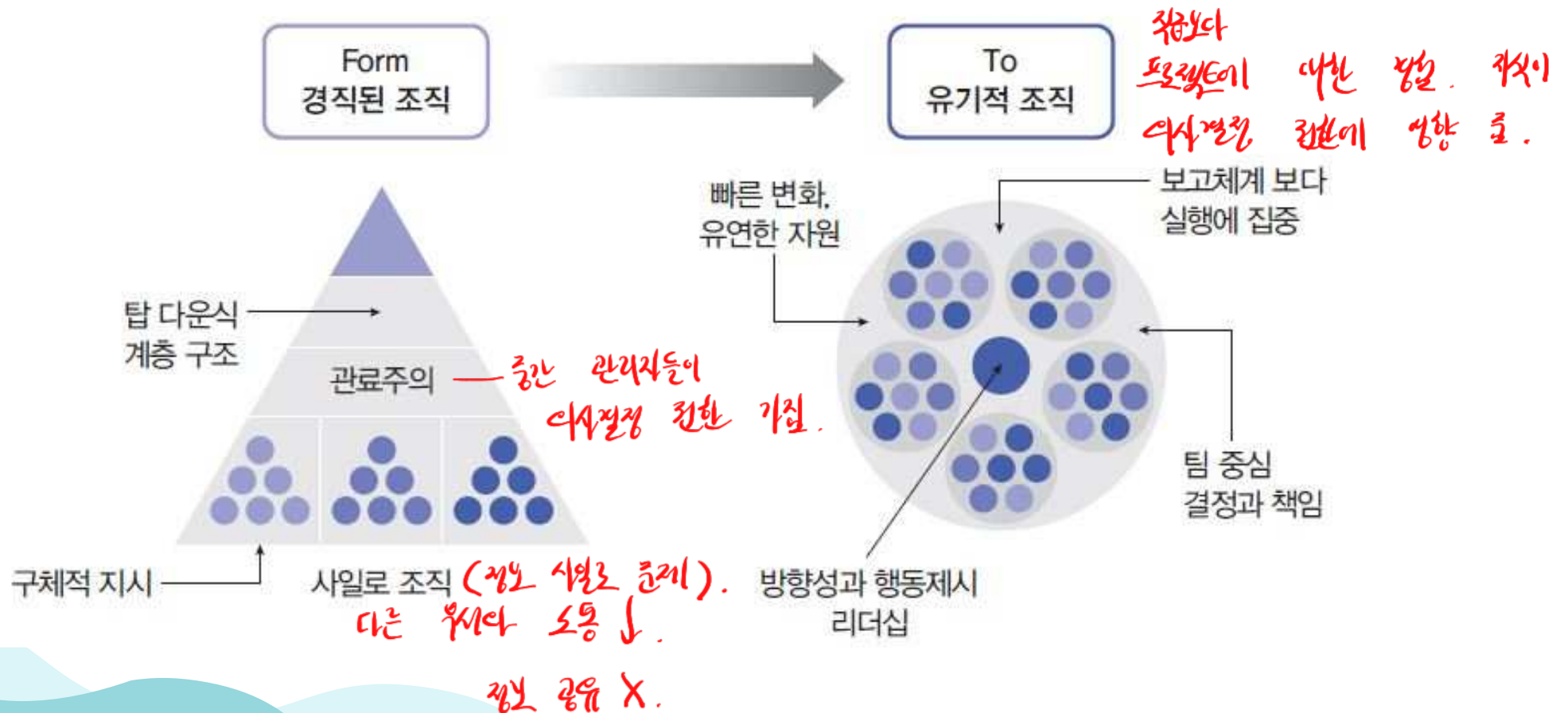
❖ 애드호크라시(adhocracy) 조직

- 조직 내 권한의 이동: 직위 중심 -> 지식과 능력 중심
- 전통적인 관료조직(bureaucracy)을 대체하는 새로운 조직구조 개념
- 상업화단식
뷰로크로라시는 수동적이고 고정적이며 수직적인 반면, 애드호크라시는 능동적이고 역동적이며 수평적이다.
- 애드호크라시의 예: 작업 태스크포스 팀, 프로젝트 팀, 매트릭스 팀
- 애드호크라시 조직은 정보기술을 이용해 한정된 기간에 프로젝트를 수행하고, 프로젝트가 끝나면 구성원들은 다른 프로젝트에 합류한다

프로젝트 관리 팀 구성.

조직구조의 변화

- ❖ 전통적 관료 조직 vs. 애드호크라시 조직
- 변화에 빠르게 적응 가능.



디지털 트랜스포메이션 (디지털 전환).

❖ 4차 산업혁명 시대의 새로운 변화관리 전략

- 사물인터넷, 빅데이터, 인공지능 기술의 발전으로 기업의 조직구조, 운영프로세스, 비즈니스모델 등에 대한 새로운 접근방식이 요구됨
- 디지털 기반 스타트업의 등장과 변화된 비즈니스 환경으로 지난 15년간 포춘 500대 기업의 절반이 사라짐 (디지털 기반 신규 기업에게 많이 빼앗김).
- 디지털에 익숙한 소비자들의 구매패턴이 변화하면서 고객 중심의 경험을 강화하기 위해 상호작용적이고 개인화된 고객관계 형성 필요

디지털 트랜스포메이션 - 이젠 꼭 필요한 역할자 (CIO) 필요

❖ 디지털 트랜스포메이션(digital transformation; DT; DX)

- 4차 산업혁명에 대응하기 위해 기업의 전략, 조직, 프로세스, 비즈니스 모델, 문화, 커뮤니케이션 등 비즈니스 전 영역에 걸친 기업 경영 시스템을 디지털 기반으로 변화시키는 전략
- “ICBM(IoT, Cloud, Bigdata, Mobile) + AI” 등 디지털 신기술로 촉발되는 경영 환경 상의 변화에 선제적으로 대응함으로써 현재 비즈니스의 경쟁력을 획기적으로 높이거나 새로운 비즈니스를 통한 신규 성장을 추구하는 기업 활동

디지털 트랜스포메이션의 성공 요인

(1) 비즈니스 전면에서 CIO의 역할 변화

(2) 유연한 조직문화

(3) 기업 내/외부 구성원간 협력과 지식 공유

내부 구성원 재교육,
외부 전문가 유입.

(4) 디지털 전환을 실행할 수 있는 인적자원

(5) 통합적인 디지털 비즈니스 전략 개발

(6) 데이터 활용 기반 구축

이제 기업이 보유하고 있는 데이터의 양, 질

→ 경쟁력.

(7) 성과지표 설정 및 측정

(KPI: key performance index)

디지털 트랜스포메이션의 성공 요인

❖ CIO의 역할과 권한 변화

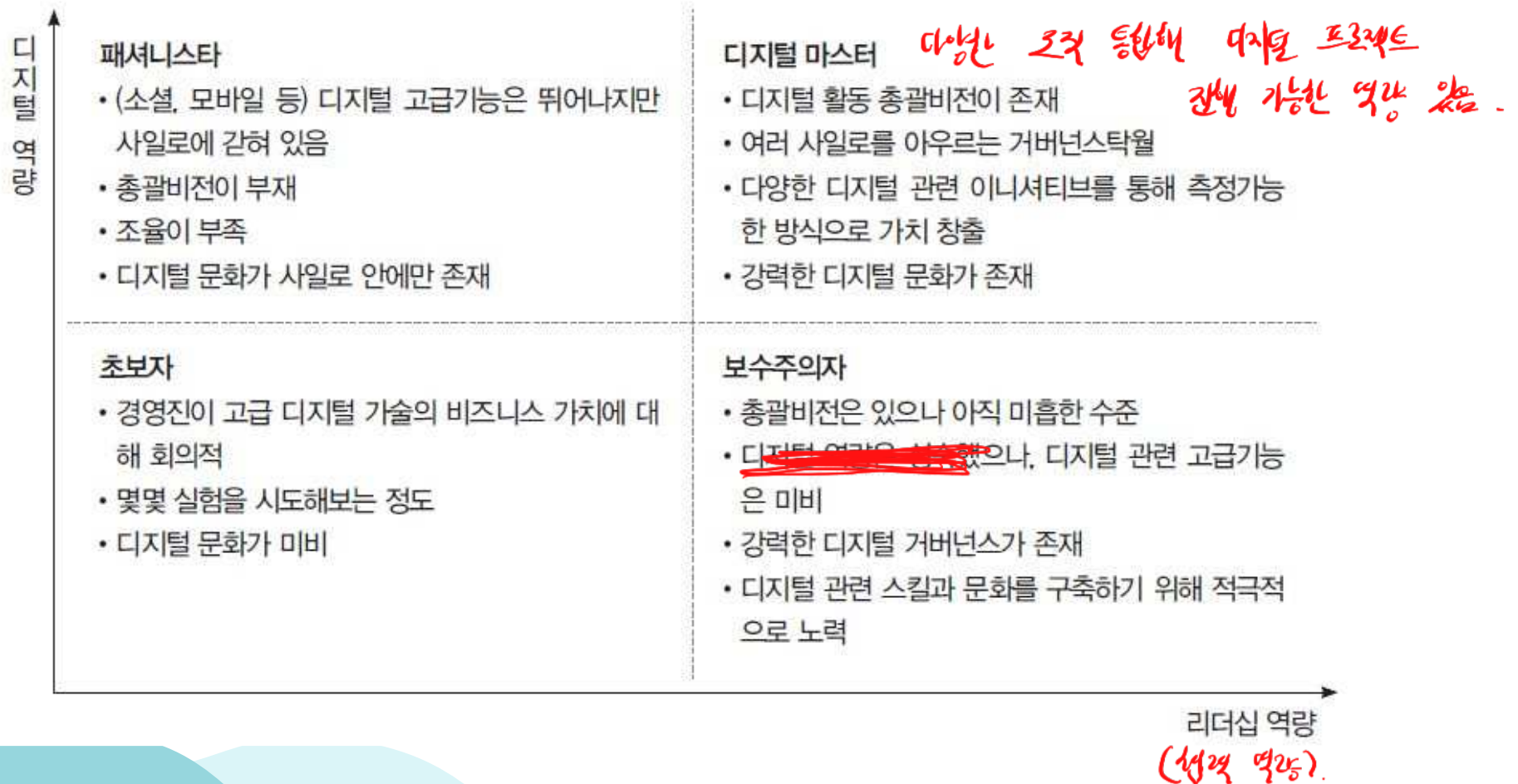
- 과거 CIO의 리더십은 내부 조직의 관리 위주였던 반면, 이제는 고객 접점 전면에 나서는 역할을 한다 *(재능·배경·관계)* *(디지털 전환 대응.)*

Old Mandate		New Mandate
공유 서비스 비용 센터 운영	→	새로운 비즈니스 모델 및 수익 창출
비즈니스 사업 결정	→	비즈니스 사업 결정을 이끌어냄
내부조직 중심 업무	→	고객 중심과 내부조직 파트너십
네트워크 주변 보안에 의존	→	'기본적인 보안'의 주도적 문화
한 덩어리로서의 구형의 컴퓨터 시스템	→	클라우드 우선, 모바일 우선, IoT용 설계
백업(back up) 데이터	→	모든 데이터를 보호하며 하나의 체계로 구축
자료를 비축	→	통찰력을 얻기 위한 영향력 있는 데이터 분석
사용자 정의 소프트웨어 프로젝트 주도	→	규칙 기반 자동화의 주도적인 문화

자료: Dell EMC(2018)

디지털 트랜스포메이션의 성공 요인

❖ 디지털 조직문화와 인적자원



사회적 디지털 트랜스포메이션

정보기술에 의한 노동시장의 변화

- Robocalypse: AI 로봇에 의한 인간 직업의 대체
- 플랫폼 경제 시대의 프리랜서 노동 시장의 확대
 - 구매자와 공급자를 연결하는 플랫폼이 다양하게 확산됨
 - 안정된 일자리는 줄어들고, 많은 노동력이 플랫폼 기반의 프리랜서로 대체됨
 - Gig Economy란? → 노동자들의 인식도 변화. (프리랜서 · 창업으로 자아 실현 경쟁으로)
 - 디지털 플랫폼을 통해 제공되는 독립형 일자리 생태계
 - 고용되지 않고 필요할 때마다 돈을 버는 플랫폼 기반 일자리
 - 예: Uber, Airbnb, Taskrabbit, Skillshare

Case 130년 제너럴일렉트릭(GE)의 디지털 트랜스포메이션

(산업용 대량 장비)

- 2011년 전통의 제조기업 GE는 소프트웨어 기업으로의 전환을 선언하고 산업인터넷(industrial internet) 개념을 제시

❖ 산업인터넷이란, 모든 산업 현장에서 쓰이는 장비, 기계에 인터넷을 접목하는 것으로, 모든 사물이 연결된다는 IoT(사물인터넷) 개념을 산업 분야에 적용해 만든 용어이다

품질관리 서비스

- 2015년 가전사업을 매각하고, 'Predix'라는 데이터분석 플랫폼을 개발하여 발전기, 항공기엔진 등의 이상징후를 모니터링하고 고장 가능성을 예측하는데 활용

