

소프트웨어공학 프로젝트 계획서

2016920013 도형림

2018920006 김규희

1. 개요

1.1. 프로젝트 개요

(1) 프로젝트 명

- 본 프로젝트의 명칭은 “미스터 대박 디너 서비스 관리 시스템”이라 한다.

(2) 프로젝트 목적

- 본 프로젝트는 주어진 요구사항을 소프트웨어공학적으로 분석 및 모델링을 체계적으로 수행하여 신뢰도가 높고 확장성, 사용 편의성, 유지보수성이 뛰어난 서비스 구축을 목적으로 한다.

(3) 프로젝트 내용

- 고객 전용 인터페이스
 - 매장에서부터 제공되는 디너 정보 및 디너 스타일 열람 기능 제공
 - 고객 맞춤형 디너 메뉴 선택 기능 제공
 - 유저 친화적인 주문 인터페이스 제공
 - 과거 주문 내역 조회를 통한 간단 주문 서비스 제공

- 직원 전용 인터페이스
 - 고객으로부터 접수된 주문 정리 기능 제공
 - 고객 정보 관리 기능 제공
 - 배달 스케줄링 기능 제공

등의 기능을 통해 디너 배달 서비스를 보다 체계적이고 효율적으로 운영 및 관리함으로써 서비스 사용자(고객)에게는 안정적이고 일관된 서비스를 온라인 상으로 제공하며 서비스 관리자(직원)에게는 작업을 최소화 및 최적화하여 할당함에 따라 직관적이고 편리한 배달/고객정보 관리를 가능하게 한다.

(4) 프로젝트 기대효과

- 편리한 인터페이스를 통해 서비스의 접근성을 높여 서비스 사용량을 증대시키므로써 매출 상승 효과를 기대할 수 있도록 한다.
- 주문 접수, 배달 스케줄링, 고객정보 관리 등과 같은 기능을 통하여 인적 자원 분배를 효율적으로 수행함으로써 필수 인력 감소로 인한 인건비 감소를 기대할 수 있다.

1.2. 프로젝트 산출물

단계	종류	내용
프로젝트 계획	프로젝트 계획	프로젝트 계획서 WBS , CPM, 간트차트
설계	디자인 시안	각 뷰별 UI 디자인
설계	UML 모델링	유스 케이스 다이어그램 상태 다이어그램 액티비티 다이어그램

		시퀀스 다이어그램 클래스 다이어그램
프로토타이핑	서비스 프로토타입	서비스 필수 기능 포함한 프로토타입 모델
개발	웹 기반 서비스 시스템(MVC 패턴)	GUI 기반의 유저(고객/직원) 인터페이스 실시간 주문 처리, 실시간 배달 관리 등의 기능을 담당하는 애플리케이션 서버 유저(고객/직원) 정보, 과거 주문 내역, 메뉴 정보 및 스타일 등의 정보를 관리하기 위한 데이터베이스
보고서	보고서	검토 보고서 중간 프로젝트 보고서 최종 프로젝트 보고서

1. 정의, 약어

- 소프트웨어공학 : 소프트웨어를 분석, 설계, 개발, 운영, 유지보수 등 개발수명주기 전반에 걸친 계획·개발·검사·보수·관리, 방법론 등을 연구하는 분야
- 소프트웨어의 확장성
 - 컴퓨터 시스템에서 나중에 필요한 기능을 덧붙여 기능을 향상시킬 수 있는 능력
 - 컴퓨터 시스템에서 시스템의 부하가 증가할 때 대처할 수 있는 능력
- 소프트웨어의 신뢰도
 - 시스템이 어떤 기간동안 결함없이 정확하게 동작을 할 수 있는 능력을 말한다.
 - 시스템의 신뢰도는 요구된 기능을 정지한 데에 대한 손실의 크기에 따라서 예측된다.
 - 프로그램의 신뢰도를 측정하는 척도로서 테스트 결과에서 발생하는 오류의 개수를 측정하는 오류 성장 모델이 있다.

- MVC 패턴

- Model, View, Controller의 약자로 하나의 애플리케이션을 구성함에 있어, 구성요소를 3가지로 구분하여 애플리케이션의 개발 및 유지보수를 용이하게 해주는 디자인패턴의 한가지 종류이다.
- Model은 애플리케이션의 정보, 즉 데이터를 나타낸다. 데이터베이스, 상수, 변수, 초기값 등과 같은 데이터와 이러한 데이터를 가공하는 컴포넌트를 의미한다.
- View는 사용자로부터 데이터를 입력받거나 Model로부터 데이터를 받아서 출력하는 기능을 담당한다. HTML, CSS, Javascript와 같은 사용자 인터페이스 요소가 여기에 해당된다.
- Controller는 사용자가 데이터를 클릭하고 수정하는 이벤트들을 처리하는 부분으로, 위에서 언급된 Model과 View를 연결시켜주는 역할을 수행한다.

- UML

- 객체지향 설계를 위한 표준 언어로 소프트웨어 시스템의 산출물을 가시화, 명세화, 구축, 문서화하는데 사용된다.

- 유스케이스 다이어그램

- 시스템과 사용자의 상호작용을 다이어그램으로 표현한 것으로 사용자의 관점에서 시스템의 서비스 혹은 기능 및 그와 관련한 외부 요소를 보여주는 것이다.
- 사용자가 시스템 내부에 있는 기능 중에 어떤 기능을 사용 할 수 있는지 나타내며 유스케이스 다이어그램을 사용함으로써 고객과 개발자가 요구사항에 대한 의견을 조율 할 수 있다.

- 상태 다이어그램

- 객체지향모델에서 클래스의 이벤트에 의거한 작동을 보여주는 다이어그램이다.
- 상태의 변화로 인한 동작 또는 하나의 상태에서 다른 상태로 변화되게 하는 사건의 주어진 시간동안의 상태를 보여준다.

- 액티비티 다이어그램
 - 단계별 작업흐름을 그림으로 표현한 다이어그램을 뜻한다.
- 시퀀스 다이어그램
 - 시간 순서로 정렬된 객체 상호작용을 나타내는 다이어그램이다.
 - 시나리오 기능을 수행하는데 필수적인 객체들 간에 교환되는 일련의 메시지들과 시나리오에 수반되는 객체와 클래스를 표현한다.
 - 일반적으로 개발 중인 시스템의 논리적인 뷰의 유스 케이스 실현화와 관련된다.
- 클래스 다이어그램
 - 시스템의 클래스, 클래스의 속성, 동작방식, 객체간 관계를 표시함으로써 시스템의 구조를 기술하는 정적 구조 다이어그램의 일종이다.
 - 일반적인 개념적 모델링, 프로그래밍 코드로의 변환을 위해서 사용된다.

2. 자원 및 일정 예측

2.1. 자원

(1) 인력

- 도형림
- 김규희

(2) 비용

- 개발 비용 산정 모델 : COCOMO-81
- 산정 이유 : 규모를 기반으로 하는 수학적 공식으로 쉽게 개발 비용을 예측할 수 있다.
- 소프트웨어 개발 유형 : 유기형 모델
 - 50 KDSI 이하 소규모 소프트웨어
 - 기능 및 성능 요구사항과 인수 테스트 및 인터페이스 등의 명세가 덜 엄격함

- 일정에 대한 강제성이 적음
- 증명된 기술을 사용
- 예상 KDSI : 5
- M(노력승수) 계산 : 0.319

cost-driver	Rating	multiplier
제품 특성		
요구되는 신뢰도	높음	1.15
데이터베이스 크기	매우 낮음	0.75
제품의 복잡도	매우 낮음	0.70
컴퓨터의 특성		
실행시간의 제약	정상	1
주기억장치의 제약	정상	1
H/W, S/W의 안전성	높음	1.15
처리시간	높음	1.07
개발요원의 특성		
분석가의 능력	높음	0.86
응용 경험	높음	0.91
컴퓨터와의 친숙성	높음	0.9
프로그래머 능력	높음	0.86
프로그래밍 언어 경험	높음	0.95
프로젝트 성격		
소프트웨어 공학 원리의 사용	매우 높음	0.82
소프트웨어 도구의 사용	높음	0.91

요구되는 개발 일정	정상	1
------------	----	---

- MM : $2.4 \times 5^{1.05} \times 0.319 = 4.1$
- TDEV(개발기간) : 2
- FPS(필요 인력) : 2
- 인건비 : $4 \times 1,000,000 = 4,000,000$
- HW/SW 비용
 - AWS 요금
 - t2.large CPU 2개 메모리 8 GiB: 월 사용료 ₩77,000
- 장소 사용료
 - 사무실 대여, 월 사용료 ₩ 600,000

2.2. 일정

- 개발 기간: 2020.10.09 ~ 2020.12.02
- 시스템 오픈 예정일: 2020.12.03
- 전체 프로젝트 기간: 2020.10.09 ~ 2020.12.03
- 프로젝트 세부 진행 일정

	내용	10.09 ~ 10.15	10.16 ~ 10.22	10.23 ~ 10.29	10.30 ~ 11.05	11.06 ~ 11.12	11.13 ~ 11.19	11.20 ~ 11.26	11.27 ~ 12.03
계획	프로젝트								
	계획서 작성								
	WBS 작성								
	프로젝트 계획서 피드백								
분석	요구사항 분석								
	시스템 분석								

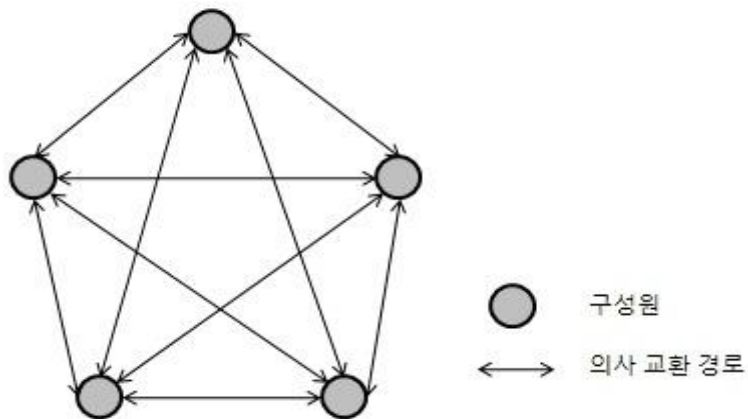
	개발 환경 설정								
	프로토타입 구현								
	프로토타입 피드백 및 수정								
설계	UML 모델링								
	데이터베이스 스키마 설계								
	UI 설계								
	애플리케이션 서버 설계								
	테스트 설계 및 기준 설정								
구현	UI 구현								
	데이터베이스 구축								
	애플리케이션 서버 구현								
테스트	시스템 테스트								
	성능 테스트								
	인수 테스트								
	테스트 피드백 및 수정								

	최종 보고서 작성								
배포	클라우드 웹 호스팅 서비스 기반 배포								

3. 조직 구성 및 인력 배치

3.1. 조직 구성

- 분산형 팀 구성 방식



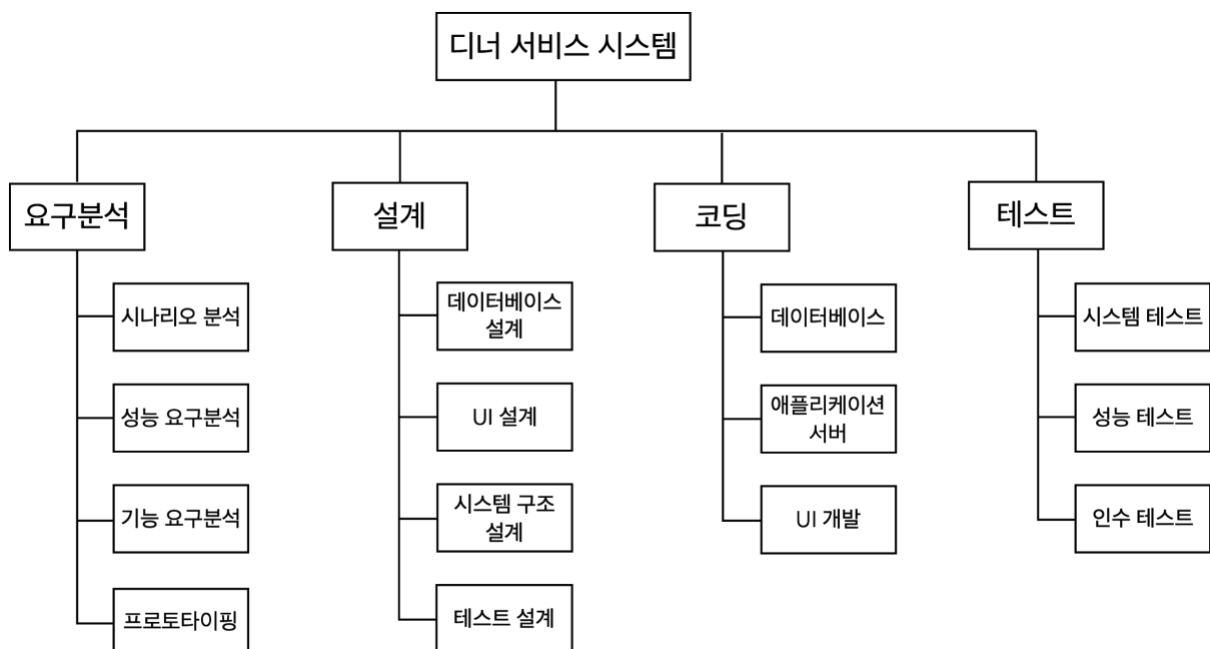
- 선정 이유
 - 의사결정을 민주주의식으로 하여 팀 구성원이 참여도와 작업 만족도를 높일 수 있다.
 - 구성원 간 동등한 책임과 권한을 갖는다.
 - 팀의 규모가 작기 때문에 빈번한 의사교류로 인한 생산성 하락과 책임감 하락과 같은 분산형 팀 구성의 단점의 영향을 받지 않는다.

3.2. 직무 기술

- 프로젝트 계획
 - 요구사항 분석, 파악 및 모델링 능력
 - 인간 중심 인터페이스 설계 능력
- 프로젝트 개발
 - 시스템 테스트 및 결과 분석 능력
 - 데이터베이스 설계 운용 능력
 - 서버 프로세스 설계 및 최적화 능력
 - 클라우드 웹 호스팅 기반 서비스 배포 능력
- 프로젝트 관리
 - 일정 계획표 활용 능력
 - 실행 가능한 대안 수립 및 평가 능력

4. WBS

4.1. WBS



4.2. CPM

작업		선행작업	소요기간(일)
A	시나리오 분석	-	1
B	성능 요구 분석	-	1
C	기능 요구 분석	A	1
D	프로토타이핑	C	5
E	DB 설계	D	3
F	UI 설계	C	3
G	시스템 구조 설계	E	5
H	테스트 설계	B, C	2
I	DB 구축	E	4
J	애플리케이션 서버 구현	I, G	15
K	UI 개발	F	15
L	시스템 테스트	H, I, J, K	2
M	성능 테스트	L	2
N	인수 테스트	M	4

도형림	10								시나리오 분석
	10								성능 요구 분석
	10								기능 요구 분석
		15	5						프로토타이핑
			30						DB 설계
		25	5						시스템 구조 설계
			5						테스트 설계
				30					DB 구축
					40	40	20		애플리케이션 서버 구현
							10	10	시스템 테스트
도형림 종합								5	성능 테스트
								5	인수 테스트
	30	40	45	30	40	40	30	20	275
김규희	10								시나리오 분석
	10								성능 요구 분석
	10								기능 요구 분석
		10	10						프로토타이핑
		20	10						시스템 구조 설계
		20	20						UI 설계
			5						테스트 설계
				10	30	40			UI 개발
							15	5	시스템 테스트
								5	성능 테스트
김규희 종합								5	인수 테스트
	30	50	45	10	30	40	15	15	235

5. 기술관리방법

5.1. 변경 관리

- 코드 변경
 - 형상관리 툴을 통하여 관리한다.
 - 관리 프로그램: git (github)
 - 소스코드의 변경 이력과 복원 등의 기능이 제공
- 사용자 요구사항 변경
 - 고객으로부터 변경 요청서 접수
 - 요구 변경사항 기술적 분석
 - 변경사항이 프로젝트에 미치는 영향 분석
 - 요구 변경사항 수용 여부 결정
 - 프로젝트 계획 변경을 필요로 하는 경우, 프로젝트 계획 수정
 - 변경사항의 실제 영향을 지속적으로 모니터링

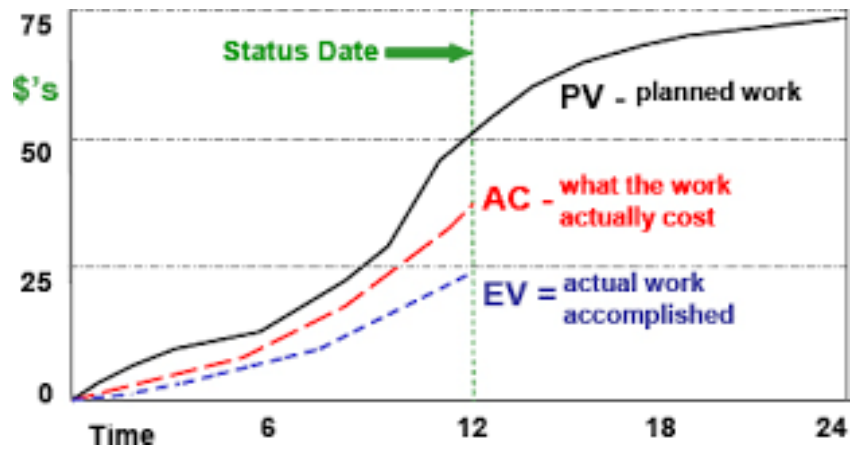
5.2. 위험 분석

단계	위험	발생 가능 성	영향 도	대응 방안
분석/ 설계	요구사항 및 분석 오류	중	중	프로토타입 구현체를 통해 사용자로부터 피드백 수용
구현	일정 지연	중	하	팀원들과의 정보공유 및 진척사항 관리, 업무 효율성 및 부하 관리

				각 개인의 피로도 파악 대처
	잘못된 기술 사용	하	하	프로젝트 계획 단계에서 철저한 조사와 검토를 통해 관련 기술의 적합도를 판단 및 대체 기술 마련을 통한 빠른 대처 실시
테스트 /이행	테스트 시나리오 오류	하	중	테스트 시나리오 재검증 및 수정
	성능 테스트 미달	중	중	시뮬레이션, 튜닝 및 관련 기술 도입
기타 위험요 소	산출물 오류	중	중	각 단계별 산출물 검토 및 수정
	비즈니스 지식 부족	하	중	소프트웨어 개발 전 사전조사 및 현업 종사자의 자문 수용
	개발 인력 부족	중	하	일정 재조정, 업무 환경 개선 및 프로토타입 재사용률 증가 등을 통한 작업량 축소

5.3. 비용 및 진도 관리

- 어닝 밸류 분석 방법

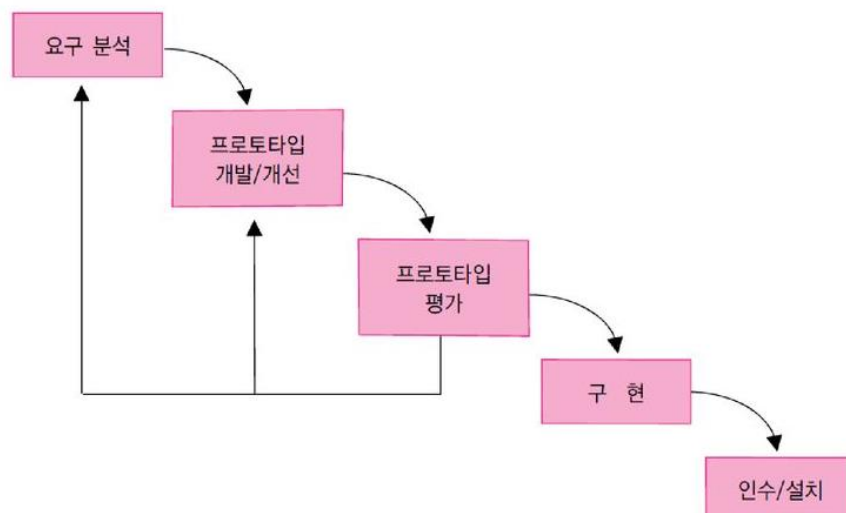


- 비용과 일정을 통합된 방법으로 모니터링 가능 → 전체적인 진척 상황 이해
- 계획된 노력, 실제 진척도, 실제 노력을 금전적 가치로 측정
- 계획 단계와 실행 단계를 비교하여 진행 상황의 추이를 확인, 미래의 비용과 일정 예측 가능

6. 표준 및 개발 절차

6.1. 개발 방법론

- 프로토타이핑 모델



- 구현 전 프로토타입 모델을 먼저 만듦으로써 사용자의 피드백을 통해 추후 시행할 구현 단계에서의 시행착오를 줄일 수 있다.
- 비교적 짧은 기간에 소규모의 프로젝트를 소규모의 인원으로 개발을 진행하기 때문에 다른 개발 프로세스보다 더 적합하다.

7. 검토 회의

7.1. 검토회 일정

검토 종류	차수	일정
프로젝트 계획 검토	-	20.10.16 ~ 20.10.17
프로토타입 검토	-	20.10.23 ~ 20.10.25
산출물 검토	1차	20.11.06 ~ 20.11.12
	2차	20.11.20 ~ 20.12.03
진행사항 검토	-	각 마일스톤 후

7.2. 검토회 진행 방법

- 검토회 일정에 맞춰 각 시기에 맞는 검토 기준을 적용하여 평가 및 후속 조치 예정

검토 종류	검토 기준	검토 결과물
프로젝트 계획 검토	프로젝트 일정이 현실적으로 진행될 수 있는가	프로젝트 계획 수정 사항
프로젝트 계획 검토	적절한 조직 구성 및 비용 산정, 위험을 고려하였는가	프로젝트 계획 수정 사항

프로토타입 검토	프로토타입의 기능이 최종 산출물의 기능을 충분하게 포함하고 있는가	중간 보고서
프로토타입 검토	요구된 서비스의 유스케이스, 시나리오를 수행할 수 있는가	중간 보고서
프로토타입 검토	프로젝트의 모델링이 충분히 효율적으로 이루어졌는가	중간 보고서
산출물 검토	개발자, 서비스 이용자, 향후 서비스 관리자 등 프로젝트 이해관계자들의 의견이 충분히 반영되었는가	최종 보고서
산출물 검토	프로젝트 산출물이 서비스의 요구사항을 충분하게 만족하는가	최종 보고서
산출물 검토	최소한의 필수 컴포넌트로 구성되어있고 컴포넌트 간의 상호작용이 충분히 효율적인가	최종 보고서
진행사항 검토	진행사항에 대한 검토가 정기적으로 수행되고 있는가	-
진행사항 검토	개발 프로젝트 진단 시기와 진행 상황 검토 시기가 일치하는가	-

7.3. 검토회 후속 조치

- 검토 결과물 확인 및 팀원 간 의사소통을 통해 문제 사항 재확인 및 시정 조치 실시

8. 개발 환경

- 운영체제: Windows, Mac OS
- IDE: Visual Studio Code, IntelliJ
- language: JavaScript, Java
- Framework: React JS, Spring boot
- 프로젝트 일관성 유지: Github, Google Drive

9. 성능 시험 방법

- 테스트 방법 : 페이지 로딩시간 측정
- 시험 전략 :
 - 실제 서비스의 운영환경과 동일한 환경 구축
 - 운영 시나리오 적용
 - 서비스 사용자 수를 증가시키면서 서버의 부하 정도 측정 및 서비스 장애 여부 검사

10. 문서화

- 설계 문서화
 - starUML 프로그램을 통하여 시스템의 구조를 시각화한다.
 - 설계 목적, 근거, 우선순위, 아웃라인, 상세사항 등을 포함한다.
 - 상위 정보는 구글 문서를 통하여 작성하며,
 - 검토회의 시 이슈에 관련된 설계 의사결정이 난 경우 문서화한다.
- 요구사항 및 변경사항 문서화
 - 자연언어 문서, 유스케이스, 사용자 스토리 또는 공정 명세서와 같은 다양한 형식 포함.
 - 구글문서를 통하여 작성한다.
- 코드 문서화
 - API 명세

11. 유지 보수

업무범위	기술지원 내용
예방 정비	시스템 운영 상태 점검을 위해 시스템 정기 점검 시간에 예방정비 실시
장애 정비	장애 발생 시 시스템 중단시간을 최소화 하기 위해 기술 인력을 신속 투입/ 해결하여 장애 해결
시스템 확장 및 이설	시스템 확장 및 이설, 타 기종으로의 교체 등이 발생에 따른 기술 지원 요청 시, 적극적인 의사소통 및 협조

12. 설치, 인수

- 서버 설치
 - git clone을 통해 코드 다운로드
 - jar 파일 빌드
 - 도커 이미지 파일 빌드 후 도커 컨테이너 실행
 - 도커 컨테이너로 프로세스를 관리함으로써, 환경 의존성을 낮출 수 있음
- 사업 종료 후 인수인계 실시
 - 단계
 1. 고객 요구사항 정의서 재분석
 2. 결과물에 대한 테스트 진행
 - 테스트 진행 시 하자보수 항목 정의 & 요청
 3. 사업자로부터 인수인계 진행(주요 프로세스 및 변경 사항 기준)

- 소스 중 리팩토링이 필요하다고 판단되는 부분 정리 및 요청

4. 인수 인계 종료 보고서 제출

- 상위 1-3단계 진행 내용에 대한 보고서

5. 운영 단계로 하자 보수 관리 및 개선 업무 진행

13. 참고문헌 및 부록

- 생능출판사, 소프트웨어 공학의 모든 것, 최은만