|  |
| --- |
| ㈜ 에이블맥스 |
| AM the Method |
| AbleMax를 위한 SW방법론 |

|  |
| --- |
| 링크투 주식회사  2017-7-11 |

목차

[I. AM the Method의 소개 3](#_Toc488220688)

[1. 도입 배경 3](#_Toc488220689)

[2. 프로세스 구조 4](#_Toc488220690)

[3. 수행 플랫폼 6](#_Toc488220691)

[4. 수작업에 의한 업무수행 환경 6](#_Toc488220692)

[5. 표준 산출물 양식 및 참조 서식 7](#_Toc488220693)

[6. 공통 가이드 8](#_Toc488220694)

[II. 착수 9](#_Toc488220695)

[1. 정의 9](#_Toc488220696)

[2. 작업구성도 10](#_Toc488220697)

[3. 작업수행을 위한 준비 10](#_Toc488220698)

[4. 작업 11](#_Toc488220699)

[III. 계획 12](#_Toc488220700)

[1. 정의 12](#_Toc488220701)

[2. 작업 구성도 13](#_Toc488220702)

[3. 작업수행을 위한 준비 13](#_Toc488220703)

[4. 작업 14](#_Toc488220704)

[4.1. 프로젝트 수행계획 14](#_Toc488220705)

[4.2. WBS(Work Breakdown Structure) 14](#_Toc488220706)

[IV. 분석 15](#_Toc488220707)

[1. 정의 15](#_Toc488220708)

[2. 작업 구성도 15](#_Toc488220709)

[3. 작업수행을 위한 준비 15](#_Toc488220710)

[4. 작업 16](#_Toc488220711)

[4.1. 요구사항 명세 16](#_Toc488220722)

[4.2. 논리 데이터 모델링 17](#_Toc488220723)

[5. 참고사항 17](#_Toc488220724)

[5.1. 논리 데이터 모델을 위한 참고 17](#_Toc488220725)

[5.2. 요구사항 명세를 위한 프로토타입 18](#_Toc488220726)

[V. 설계 20](#_Toc488220727)

[1. 정의 20](#_Toc488220728)

[2. 작업 구성도 20](#_Toc488220729)

[3. 작업수행을 위한 준비 20](#_Toc488220730)

[4. 작업 21](#_Toc488220731)

[4.1. UI 설계 21](#_Toc488220735)

[4.2. 물리 데이터 모델링 21](#_Toc488220736)

[5. 참고사항 23](#_Toc488220737)

[5.1. 상세 물리 모델링 방법 23](#_Toc488220739)

[VI. 구현 25](#_Toc488220740)

[1. 정의 25](#_Toc488220741)

[2. 작업구성도 25](#_Toc488220742)

[3. 작업수행을 위한 준비 25](#_Toc488220743)

[4. 작업 26](#_Toc488220744)

[4.1. 프로그램 개발 26](#_Toc488220755)

[4.2. 단위테스트 26](#_Toc488220756)

[4.3. 테스트 계획수립 26](#_Toc488220757)

[4.3.1. 인수테스트 계획수립 26](#_Toc488220758)

[4.3.2. 통합테스트 계획 수립 27](#_Toc488220759)

[4.3.3. 단위테스트 계획 수립 28](#_Toc488220760)

[VII. 테스트 29](#_Toc488220761)

[1. 정의 29](#_Toc488220762)

[2. 작업구성도 29](#_Toc488220763)

[3. 작업수행을 위한 준비 29](#_Toc488220764)

[4. 작업 30](#_Toc488220765)

[4.1. 통합 테스트 30](#_Toc488220771)

[4.2. 인수 테스트 지원 30](#_Toc488220772)

[VIII. 종료 31](#_Toc488220773)

[1. 정의 31](#_Toc488220774)

[2. 작업구성도 31](#_Toc488220775)

[3. 작업수행을 위한 준비 31](#_Toc488220776)

[4. 작업 32](#_Toc488220777)

[4.1. 프로젝트 종료보고 32](#_Toc488220778)

# AM the Method의 소개

## 도입 배경

AM the Method는 에이블맥스 만을 위한 소프트웨어개발 방법론이다. AM the Method는 정보통신산업진흥원(NIPA)의 SW공학현장적용사업의 결과이며, 컨설팅 작업은 링크투에서 진행하였다. 방법론이 필요한 이유는 프로젝트 성공율을 높이기 위함이며, 소프트웨어 개발을 사람에게만 의존하는 한계를 극복하기 위함이다.

SW프로젝트의 성공을 개발자의 실력에 의존하는 것은 한계가 있다. 아무리 실력이 있는 개발자라도 병이 나거나 이직을 할 수 있으며, 남아있는 동료들은 프로젝트를 마무리하기 위해 많은 희생을 해야 한다. 문제를 해결하는 방법은 자체적인 방법론을 갖추고 정해진 절차에 따라 SW를 개발하는 것이다. 방법론을 보유한다는 것은 CMMi 인증을 받거나, SP(Software Process)인증을 받기 위한 목적도 있겠지만, 안정적인 개발업무 수행과 품질이 보장될 수 있는 SW를 창출하기 위함이다. 물론 각종 인증을 받기 위한 가장 기본적인 요건은 체계적인 프로세스에 의해 SW를 개발하는 것이기도 하다. 대부분의 중소기업에서 체계적인 SW개발 프로세스를 자체적인 방법론으로 정립하여 관리한다는 것은 쉽지 않다. 이는 별도의 방법론 담당자를 필요로 하는 일이며 추가적인 비용이 발생하기 때문이다.

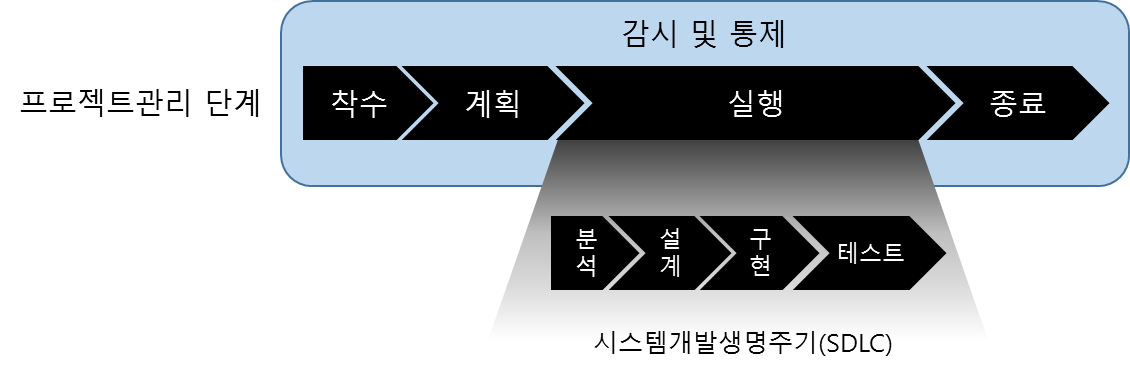
우리나라 중소규모 SW기업의 경우 별도의 방법론 담당자를 지정하거나 품질관리 업무를 수행할 담당자를 선발하여 지속적으로 내부 품질관리 절차를 유지하는 것은 결코 쉬운 일이 아니다. 항상 촉박한 일정과 개발자 부족으로 주어진 과업을 수행해내기에도 급급한 실정에서 방법론을 담당할 전담인력을 유지할 수 없다. 하지만 조직적 차원에서 제대로 된 업무 프로세스가 무엇인지 고민하고 이를 정립하여 규정화해야 하는 것에 대한 필요성은 개발자와 관리자 모두 공감할 것이다. 현재의 품질수준을 높이고 세계로 진출하기 위해서는 방법론을 갖추거나 품질담당자를 별도로 지정하는 것이 필요할 수 밖에 없는 것이다.

체계적인 프로세스에 의해 SW를 개발하는 것이 필요하다는 것을 깨닫게 되면 SW방법론을 고려하는 것이 답이다. 품질전담인력이 없이도 내부적으로 SW개발 프로세스를 자체적인 방법론으로 갖추고 SW개발인력이 서로 지켜 나가기만 해도 많은 한계를 극복할 수 있다. 그리고 지속적으로 프로세스를 개선하여 SW품질의 제고를 위해 노력한다면 SW품질이 높아질 수 밖에 없다. CMMi 혹은 SP인증은 SW품질이 높아질 수 있는 체계를 갖추고 있음을 뒷받침하는 증빙일 뿐이다. 어떻게 하면 별도의 프로세스 관리인력 없이 체계적인 개발방법을 정립하고 이를 기준으로 SW를 개발할 수 있을 것인가? 방법은 표준적인 방법론을 자체적으로 도입하여 보유하는 것이다.

㈜링크투가 만든 LT the Method를 에이블맥스에 맞게 커스터마이징(customizing)한 AM the Method는 국제표준에 따른 SW개발프로세스를 실무적으로 쉽게 적용가능 하도록 중소기업을 위해 정립한 SW방법론이다. AM the Method는 ISO 21500에 입각한 프로젝트 관리방법과 ISO/IEC 12207에 의한 개발프로세스를 통합하여 관리와 개발이 동시에 관리가능한 프로세스를 정립하였으며 중소규모 개발업무에 초점을 둔 프로세스이기 때문에 최소한의 관리와 단순한 개발절차를 지향하였다. 이러한 AM the Method는 체계적인 프로세스를 수립하고 안정적인 SW개발이 가능하도록 한다.

## 프로세스 구조

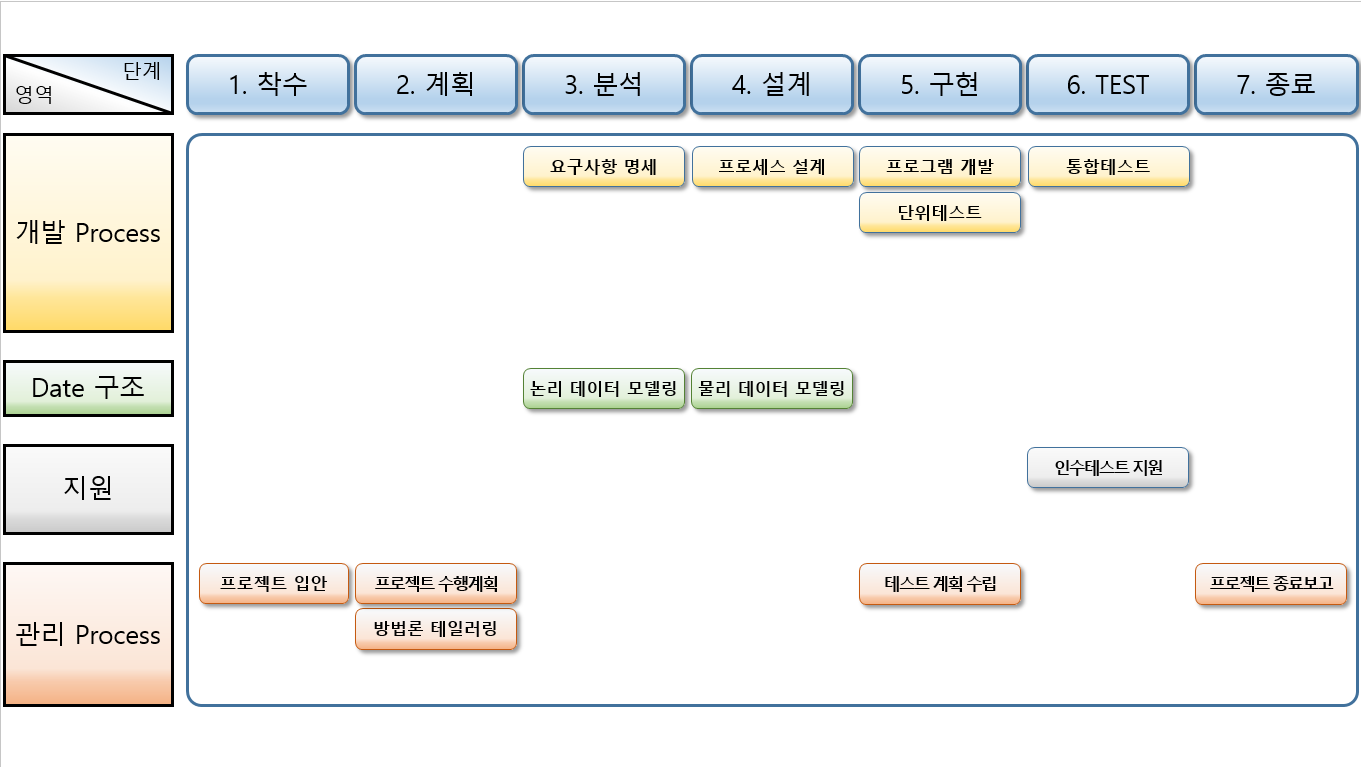
AM the Method의 수행단계는 PMBOK(Project Management Body Of Knowledge)와 SDLC(System Development LifeCycle)에 따라 시간적인 순서에 의해, 착수, 계획, 분석, 설계, 구현, 테스트, 종료의 7단계로 나누었다. 전체 수행단계는 ‘감시 및 통제’ 활동에 의해 관리된다.



단계별 수행업무를 상세하게 정의하기 위하여 ‘영역’을 구분하여 ‘개발 Process’, ‘Data 구조’, ‘지원’, ‘관리 Process’로 분리하였다. ‘개발 Process’ 영역이 일반적으로 말하는 SW개발 방법론에 속하는 내용이고 ‘관리Process’ 영역이 프로젝트관리 방법론에 속하는 내용이다. ‘지원’ 영역은 품질담당자 혹은 프로젝트의 사업관리 담당자가 주로 수행하는 활동일 수 있는데, ‘관리 Process’와 구분하여 분리하였으며, Data Architect가 주로 수행하는 ‘Data 구조’영역도 분리하였다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 단계  영역 | 착수 | 계획 | 분석 | 설계 | 구현 | 테스트 | 종료 |
| 개발process |  |  |  |  |  |  |  |
| data구조 |  |  |  |  |  |  |  |
| 지원 |  |  |  |  |  |  |  |
| 관리process |  |  |  |  |  |  |  |

단계와 영역에 속하는 수행업무는 작업(task)이라고 칭한다. 해당 영역과 단계별 ‘작업(task)’이 아래의 그림과 같이 나열되어 있다.



각 작업은 수행결과로 산출물이 작성된다. 산출물의 코드구성은 “단계\_영역\_작업별 표준 산출물 양식명\_날짜”를 조합한 코드체계를 가진다. 날짜는 지속적인 작업과정을 표시하기 위한 버전이지만, 최종 완료되기 전까지는 1.0전이 아니다. 작업이 완료되어 최종 완성버전이 되면, 날짜는 삭제하고 “v1.0”을 대신 입력한다. 1.0버전이라는 의미는 이후 변경관리를 필요로 함을 의미하며, 변경된 이력을 기록하고 버전을 하나씩 증가시키게 된다.

예) 분석\_개발Process\_요구사항명세\_20170707 -> 분석\_개발Process\_요구사항명세\_v1.0

변경사항이 반영된 산출물은 v1.1이 되고, 변경이력은 산출물 내부에 기록한다. 사소한 변경이 아니라, 구조적인 변경이라고 판단될 때에는 v2.0으로 새롭게 번호체계를 부여하기 되는데, 구조적 변경이라고 하는 것은 좀더 심각한 변경이 발생하여 별도의 시간과 노력이 투입되었음을 의미한다.

## 수행 플랫폼

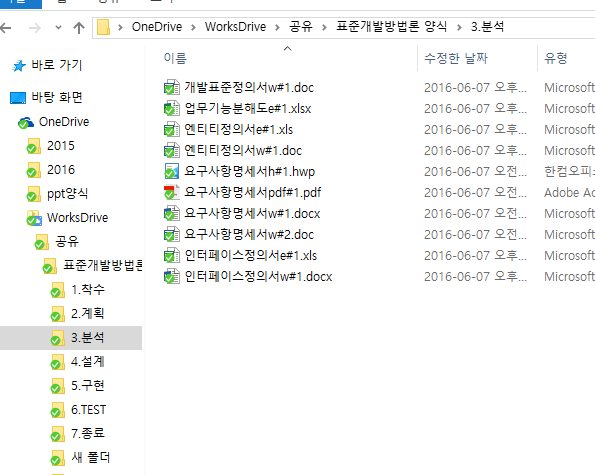
AM the Method에 의한 SW개발은 수작업으로도 수행이 가능하지만, 생산성을 높이고자 한다면 수행을 위한 플랫폼을 갖추는 것이 필요하다. 에이블 맥스의 프로젝트 진행은 SW개발과 프로젝트 관리를 동시에 지원하기 위해 아래와 같은 플랫폼 도구를 이용할 것을 권장한다.

|  |  |  |
| --- | --- | --- |
| **지원기능** | **도구명** | **구분** |
| 요구사항 관리 | 레드마인(RedMine) | 공개 SW |
| 형상 관리 | Git | 공개 SW |
| 테스트 케이스 관리 | 테스트 링크(TestLink) | 공개 SW |
| 지속적 통합 | 젠킨스(Jenkins) | 공개 SW |
| 테스트 자동화 | 소나큐브(SonarQube) | 공개 SW |

## 수작업에 의한 업무수행 환경

레드마인이나 Git등을 사용하기 어려운 상황이라면 산출물 관리용 서버를 별도로 두고 AM the Method에서 제공하는 산출물 양식을 활용하여 개발과 관리업무를 수행하는 것도 하나의 방법이다. 이러한 경우 업무수행은 개발자와 관리자가 공유하는 파일서버의 저장공간을 활용하여 적절한 홀더체계를 설정하고 산출물을 해당 홀더에 저장하면서 업무를 진행하게 된다. 다만 관리자의 입장에서 보안상 공유가 어려운 파일은 보안설정을 이용하여 접근통제를 할 수 있다. 프로젝트 수행결과를 관리하면서 문서 혹은 산출물의 버전의 관리는 수작업으로 이루어지지만 미리 정해진 규칙에 의해 관리하고 확인하게 된다. 홀더체계는 해당 영역별로 분리하거나, 단계별로 분리하여 설정할 수 있다.

레드마인이나 Git등을 사용하기 어려운 상황이라면 산출물 관리용 서버를 별도로 두고 AM the Method에서 제공하는 산출물 양식을 활용하여 개발과 관리업무를 수행하는 것도 하나의 방법이다. 이러한 경우 업무수행은 개발자와 관리자가 공유하는 파일서버의 저장공간을 활용하여 적절한 홀더체계를 설정하고 산출물을 해당 홀더에 저장하면서 업무를 진행하게 된다. 다만 관리자의 입장에서 보안상 공유가 어려운 파일은 보안설정을 이용하여 접근통제를 할 수 있다. 프로젝트 수행결과를 관리하면서 문서 혹은 산출물의 버전의 관리는 수작업으로 이루어지지만 미리 정해진 규칙에 의해 관리하고 확인하게 된다. 홀더체계는 해당 영역별로 분리하거나, 단계별로 분리하여 설정할 수 있다.



## 표준 산출물 양식 및 참조 서식

AM the Method 의 표준 산출물 양식은 MS오피스로 작성이 가능하도록 MS워드로 작성하는 DOC파일과 엑셀로 작성하는 XLS파일로 준비되어 있으며, 현장의 상황에 따라 커스터마이징하여 활용할 수 있다. 표준 산출물 양식은 소프트카피(softcopy)로 구성되어 있으며, 본 방법론에서 정의한 가장 기초적인 서식이다.

* + AM the Method 표준 산출물 양식

| 단계 | 영역 | 작업 | 표준 산출물 양식 | 테일러링 |
| --- | --- | --- | --- | --- |
| 1.착수 | 관리 Process | 프로젝트 입안 | * 별도로 정의하지 않음 |  |
| 2.계획 | 관리 Process | 프로젝트 수행계획 | * 별도로 정의하지 않음 |  |
| WBS |  |
| 3.분석 | 개발 Process | 요구사항 명세 | 요구사항 명세서 |  |
| Data 구조 | 논리데이터 모델링 | 데이터 분석서 |  |
| 4.설계 | 개발 Process | 프로세스 설계 | 프로세스 설계서 |  |
| Data 구조 | 물리 데이터 모델링 | 물리 데이터 모델 설계서 |  |
| 5.구현 | 개발 Process | 프로그램 개발 | * 별도로 정의하지 않음 |  |
| 단위테스트 | 단위테스트 결과서 |  |
| 관리 Process | 테스트 계획 수립 | 테스트 계획서 |  |
| 6.테스트 | 개발 Process | 통합테스트 | 통합테스트 결과서 |  |
| 지원 | 인수테스트 지원 | * 별도로 정의하지 않음 |  |
| 7.종료 | 관리 Process | 프로젝트 종료보고 | 프로젝트 종료보고서 |  |

## 공통 가이드

산출물을 작성하기 위한 양식의 레이블은 다음과 같이 정의한다.

**(양식-2.단계-영역)산출물명\_일자**

위의 양식을 활용하여 산출물 작성시, “양식”을 삭제하고 프로젝트번호를 입력하는 것이 필요하며, 산출물이 1.0버전이 되기 전에는 일자 뒤에 작성자 성명을 입력해서 관리하는 것이 필요하다.

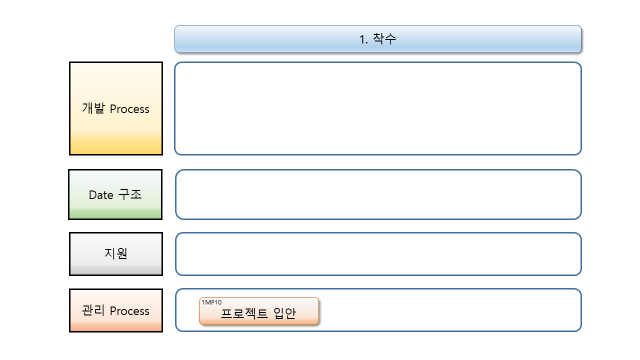
# 착수

## 정의

프로젝트를 입안하고, 의사결정자로부터 승인을 득하는 과정에 해당하는 것이 착수단계이다. 외부고객으로부터 프로젝트를 수주하려고 한다면, RFP(Request For Proposal)을 입수하여 제안에 참여할 것인지 결정하는 것이 착수단계의 주요 업무이다. 착수단계는 프로젝트 수행단계를 정의하고 이를 공인하는 과정으로 구성되며, 프로젝트의 목표를 식별하고 프로젝트관리자를 선정하여 권한을 부여하게 된다. 착수단계를 임하는 자세는 갑의 입장에 있는 발의담당자, 즉 사업PM의 입장과 을의 입장에 있는 수행책임자, 즉 개발PM의 입장이 다르다. 사업PM과 개발PM이 동일한 조직에 속하고 외부에 발주의뢰 하지 않는 인하우스(in-house)개발의 경우에는 갑과 을의 구분이 필요 없지만 같은 조직 내부에서도 요청자와 수행자의 차이는 있다.

대규모 프로젝트의 경우에는 대부분 갑의 사업PM이 RFP(Request For Proposal)을 통해 발주의뢰하고 전문 SI업체인 을의 개발PM은 제안서(proposal)를 작성하게 된다. RFP가 제시된 이후 제안서를 작성하여 제출하는 기간은 단 2주~4주 정도의 시간밖에 주어지지 않기 때문에 을의 개발PM은 상당히 짧은 시간의 착수단계를 가진다. 반면 갑의 프로젝트 발의 담당자는 프로젝트를 입안하여 RFP가 완성되는데 걸리는 시간이 많게는 1년이 걸릴 수도 있다. 내부적으로 프로젝트를 입안하여 결재를 받고 예산을 확보한 이후 구체적인 RFP를 완성하여 전문 SI업체에게 의뢰하는데 상당히 많은 어려움이 발생한다.

## 작업구성도



AM the Method는 갑과 을의 구분이 없이 작성되었지만, 이를 제대로 이해하는 사업PM과 개발PM이라면 훨씬 프로젝트의 성공가능성을 높일 수 있고, 이해관계자와 협력적인 의사소통을 통해 성공적인 프로젝트를 완수할 수 있다. 사업PM은 프로젝트를 입안하여 RFP를 생성하며, 개발PM은 RFP를 이해하고 이를 성공적으로 완수하기 위한 최적의 대안을 도출하게 된다. 본 방법론에서는 사업PM이 초기 프로젝트 입안 시 작성하거나, 개발PM이 프로젝트 수주를 위해 승인을 받기 위한 작업을 ‘프로젝트 입안’이라는 작업으로 정의하였으며, 산출물은 ‘프로젝트 개요서’라고 명명하였다. 프로젝트 개요서의 내용에는 프로젝트 범위, 산출물, 프로젝트 기간, 자원에 대한 기본적인 내용이 포함된다. 사업PM의 경우 프로젝트 개요서가 승인이 되면 RFP를 작성하게 되고, 개발PM은 RFP를 기초로 하여 프로젝트 참여결정 후 제안서를 작성하게 된다.

## 작업수행을 위한 준비

프로젝트 수행을 위한 구상이 필요하며 RFP를 접수하거나, 인하우스개발인 경우에도 프로젝트 착수를 위한 근거가 필요하다. RFP가 있다면 이를 충분히 숙지하여 고객이 원하는 시스템이 무엇인지 이해하여야 한다.

* 구축하고자 하는 정보시스템에 대한 이해
* 고객의 식별 및 이해관계 이해
* 프로젝트 착수의 근거 확보
* 착수 팀의 구성 및 승인
* 프로젝트 개요서 작성 및 확인

## 작업

프로젝트 입안 작업은 현장의 필요에 의해, 혹은 조직의 사업방향에 의해 발의될 수 있으며, SI수행업체라면 고객의 RFP를 참조하여 입안을 할 수 있다. 입안의 산출물로 프로젝트 개요서를 작성하고 의사결정자의 승인을 받는다.

프로젝트 개요서 서식은 별도로 준비되어 있는 방법론 양식을 활용하여도 되고, 별도의 양식이 없이 작성되어야 된다.

* **표준 산출물 양식: 별도로 정의하지 않음**

# 계획

## 정의

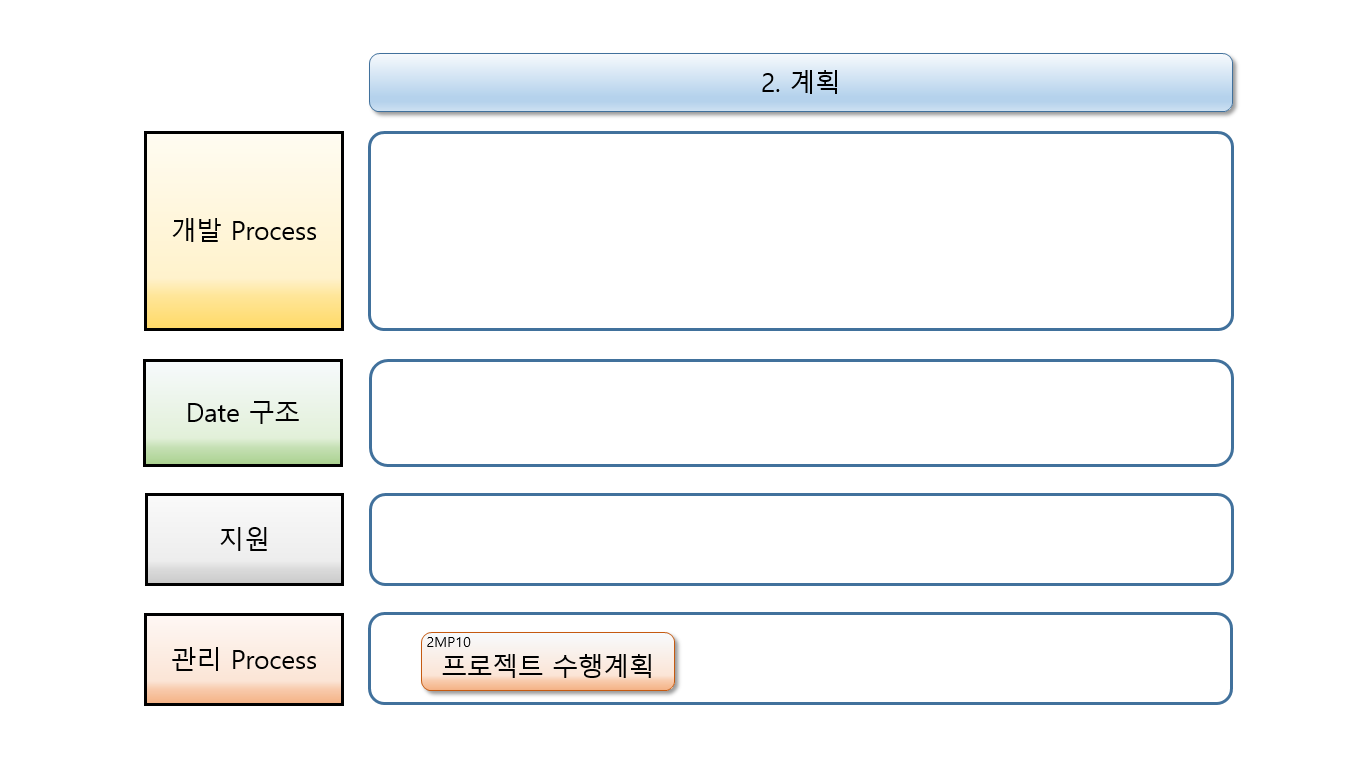
이미 착수단계에서 개요수준으로 정의된 프로젝트는 본 계획단계에서 구체화된다. 프로젝트에 관련된 활동들의 의존관계를 식별하고 관련된 위험을 계량적으로 분석한다. 가정과 제약사항들을 발굴하여 개략적인 해결방안을 계획한다. 최대한 상세한 계획을 수립하는 것이 좋겠지만 모든 것을 구체적으로 계획할 수는 없으며, 향후 발생이 예상되는 각종 이슈에 대하여 어떻게 해결할 것인지 프로세스를 설정하고 실행가능성을 제시하는 것이 필요하다.

프로젝트 계획단계에서 프로젝트 결과에 영향을 미치는 이해관계자들이 적절하게 참여하여 성공의 가능성을 높여야 할 필요가 있다. 이해관계자들은 프로젝트 계획 및 모든 하위 계획에 영향을 주는 기술 및 지식, 영향력을 가지고 있다. PM은 이해관계자를 잘 선정하고 적절하게 의견의 수렴이 가능하도록 의사소통을 관리해야 하며, 의사결정에 참여할 수 있도록 해야 한다.

계획단계에서의 산출물은 ‘프로젝트 수행계획서’와 ‘WBS(Work Breakdown Structure)’이다. 프로젝트 수행계획서는 프로젝트 전체의 진행을 위해 가장 중요한 문서이며, 만약 갑과 을이 프로젝트 수행에 대한 계약을 체결한다면 계약의 구체적인 수행내역에 대한 근거문서이기도 하다. 프로젝트 수행계획에는 PMBOK의 프로젝트 8대관리영역이 모두 포함되어 계획수립이 이루어지는 것이 좋겠지만, 프로젝트 규모에 따라 조정이 가능하다. 하지만 적어도 범위관리에 해당하는 요구사항 변경과 추적 계획, 예산관리에 해당하는 투입인력관리 계획, 일정관리에 해당하는 진척율 관리방법, 품질관리에 해당하는 프로세스 확인 및 테스트 전략은 반드시 포함되는 것이 좋다.

WBS는 전혀 새로운 형태의 작업을 필요로 한다. 보통은 “MS project”툴로 WBS를 작성하지만, MS project의 사용이 원활하지 않을 경우에는 엑셀로 작업을 하거나, “Open Project”라는 공개SW를 활용하는 경우도 있다. Open Project의 사용법은 MS project와 유사하지만 인터넷버전의 활용은 전혀 새로운 플랫폼의 사용을 가능하게 한다.

## 작업 구성도



## 작업수행을 위한 준비

프로젝트 개요서를 근거로 구체적인 계획을 수립하기 위한 자료를 수집한다. 회사의 기본적인 SW방법론이 있다면 테일러링을 통해 바로 산출물을 정의할 수 있지만 만약 방법론이 없다면 외부의 산출물을 참조하여 어떠한 산출물이 프로젝트 결과로 도출될 것인지 선택하는 작업을 진행하여야 한다. 이해관계자와 의사소통을 통해 프로젝트 계획에 필요한 내용들에 대하여 의견을 교환하고, 일부 이해관계자의 의견에 치우쳐 지나치게 많은 내용을 담지 않도록 하는 것이 중요하다.

* 객체지향 분석기법에 대한 이해
* BPMN에 입각한 프로세스 모델링 이해
* 개념, 논리 ERD에 대한 이해
* 형상관리 절차의 정립과 형상관리 도구의 설치
* WBS에 의한 일정관리 및 공수관리 기준
* 구축하고자 하는 정보시스템에 대한 상세한 이해
* 이해관계자 식별 및 정의
* 프로젝트 수행 팀의 구성 및 승인
* 프로젝트 수행계획서 작성 및 승인
* 범위추적 매트릭스 확인
* MS project 혹은 Open Project의 사용법 숙지

## 작업

### 프로젝트 수행계획

프로젝트 개요서, RFP, 제안서 등을 근거로 하여 구체적인 프로젝트 계획을 작성하게 되며, 아래의 목차를 참고하여 작성하면 된다. 프로젝트 수행계획서는 프로젝트 8대 관리영역에 대한 이해와 지식이 있어야 가능하며, 프로젝트 경험을 가지고 있는 PM이라도 새로운 프로젝트의 계획수립은 쉬운 일이 아니다.

* **표준 산출물 양식 : 별도로 정의하지 않음**

### WBS(Work Breakdown Structure)

MS project사용에 익숙한 PM은 WBS를 이미 작성한 경험이 있을 것이다. WBS는 단계와 액티비티, 그리고 태스크 형태로 작업을 분할하여 최종적으로 워크패케지(work package)단위로 구체화시킬 수 있어야 한다. 워크패케지는 2주이내의 작업이며, 누가 무엇을 수행하고, 수행의 결과로 어떤 산출물이 도출될 것인지 알 수 있도록 해야 한다. 투입인력과 시간을 통해 투입공수를 계산할 수 있으며, 이를 통해 인건비를 계산할 수도 있다. WBS가 구체적으로 작성되어 있어야 분석단계의 시작과 함께 진척율을 계산할 수 있다. 진척율 계산은 여러가지 방식이 있지만 가장 단순한 방식은 시간의 경과에 따라 산출물 확인과 함께 해당 업무의 수행이 완료된 것으로 가정하는 방식이다.

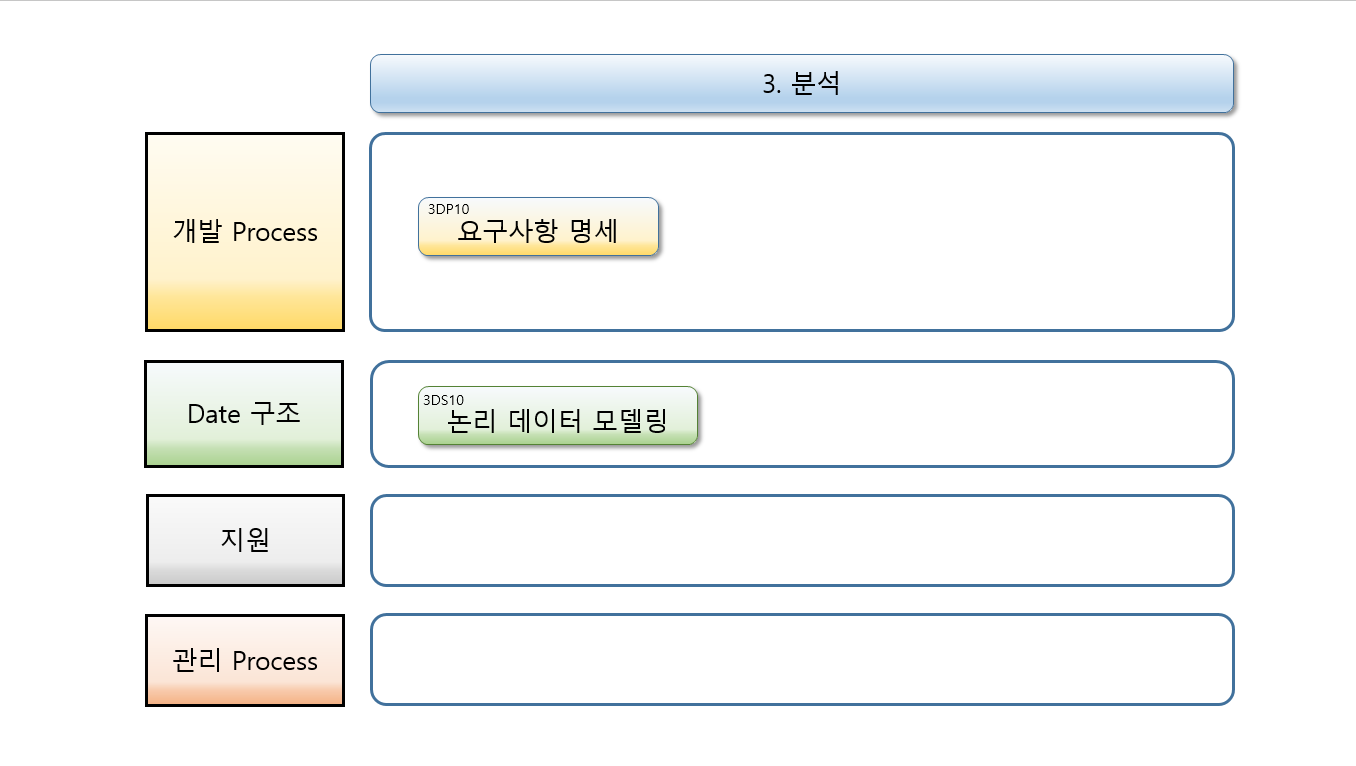
* **표준 산출물 양식 : WBS**

# 분석

## 정의

분석 단계에서는 표준에 입각하여 요구사항을 분석하고 Data의 논리적 구조를 모델링하며, 개발준비를 위해 형상관리와 개발표준을 준비한다. 관리를 위한 목적으로 요구사항 분석 이후 발생할 변경에 대비하여 변경관리 절차를 준비하고, 일정과 공수를 측정하여 계획 대비 실적을 확인한다.

## 작업 구성도



## 작업수행을 위한 준비

다음 준비에 대한 검토 및 확인이 필요하다. 분석을 위한 충분한 준비가 되어있지 않은 경우, 교육과 워크샵을 통하여 프로젝트 참여인원들에 대한 공통의 분석역량 향상과 프로젝트에 대한 이해가 필요하다.

* 객체지향 분석기법에 대한 이해
* BPMN에 입각한 프로세스 모델링 이해
* 개념, 논리 ERD에 대한 이해
* 형상관리 절차의 정립과 형상관리 도구의 설치
* WBS에 의한 일정관리 및 공수관리 기준

## 작업



### 요구사항 명세

요구사항 명세를 제대로 정의한다는 것은 매우 어려운 일이며, 이는 방법론의 문제가 아니라 명세서를 작성하는 전문가의 경험과 식견에 좌우되는 문제이다. 명세서에는 기능적인 명세와 비기능적인 명세를 모두 정의할 수 있어야 하며, 비기능적 명세의 내용을 정의하기 위해 최종 시스템의 작동환경과 사용자의 입장을 고려하여야 한다.

* **표준 산출물 양식 : 요구사항 명세서**

#### 기능 요구사항 명세

* + - 작성된 ‘요구사항 정의서’에 대해 다음과 같은 사항을 고려하여 고객과 인터뷰 및 워크샵을 통하여 검토한다.
    - 요구사항간 일치성, 요구사항간 중복성, 구현 가능성, Test 가능성, 우선순위 등을 검토한다.
    - 검토한 내용을 토대로 프로젝트에 포함된 요구사항을 구체화하여 기능 요구사항 명세를 작성한다.

#### 비기능 요구사항 명세

* + - ISO 9126의 품질 특성을 참조하여 상세하게 작성하고 측정기준을 설정한다. (ISO 9126의 품질 주특성: 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성)
    - 본 프로젝트의 특성에 따라 중요하다고 판단되는 품질 특성을 선택한다
    - 비기능 요구사항 명세를 작성한다

#### 가정 및 제약사항

* + - 요구사항 도출, 설계 및 구현을 하는 데에 있어서 적용기술 / 도구 / 언어 / DBMS / 표준 / 법규 등 가정 및 제약 사항을 파악한다.

### 논리 데이터 모델링

엔티티와 관계를 정의하고, 속성을 확인하여 식별자를 확정하는 과정이다. 정규화를 통하여 새로운 엔티티가 생성되거나 새로운 관계들이 생성되고, M:N 관계들을 해소한다. 엔티티, 관계, 속성에 대하여 논리 데이터 모델을 검증하는 절차를 수행한다.

* **표준 산출물 양식 : 데이터 분석서**

#### 엔티티 정의

* + - 업무에서 관리하고자 하는 데이터를 바탕으로 후보 엔티티를 수집한다.
    - 데이터 요구사항을 분석해서 업무적인 활동에서 필요한 주체, 목적 대상 사건 등이 엔티티의 후보가 되며 이중에서 분명하게 업무적으로 관리 대상이 되는 것을 엔티티로 정의한다.
    - 실제 업무에서 보편적으로 사용하는 용어를 기준으로 전사에서 지정한 표준 단어 및 용어에 맞게 엔티티 명을 정의한다.

#### 속성 정의

* + - 개념 데이터 모델링 단계에서는 해당 엔티티의 특성을 가장 잘 나타내 주는 핵심적인 속성을 정의해서 엔티티 간의 관계를 정의하는데 활용한다. 예를 들어 고객의 경우 ‘주민번호’가 핵심 속성이 된다.

#### 관계정의

* + - 관계는 데이터 간에 존재하는 업무 관계를 나타내는 것으로 엔티티 간에 어떠한 업무 규칙이 있는지 확인을 한 다음 결정을 한다.

#### 엔터티 상세화

* + - 정규화, M:N관계 해소, 식별자 정의를 통해 논리 데이터 모델을 더욱 상세화한다.

## 참고사항

### 논리 데이터 모델을 위한 참고

#### 데이터 도메인 정의

도메인은 각 속성의 일반적인 특징을 나타내는 데이터의 성격을 분류한 것으로서, 청구금액 이라는 속성은 금액이라는 도메인에 속하고 금액도메인에 의해 보관되어질 데이터의 성격을 규제하게 된다. 따라서 도메인은 물리 설계 단계에 구현할 DBMS에 맞는 Data Type 으로 매핑되어 진다.

#### 표준 코드 정의

표준 코드 정의는 수집된 용어로부터 코드를 선별하여 현 코드의 코드값을 조사한다. 현 코드를 바탕으로 통합 요구 사항과 통합 필요성에 따라 통합 대상을 파악하고 표준 코드를 정의하고 현 코드와 매핑 설계를 한다.

#### 표준 용어 정의

표준 용어 정의는 단어, 도메인, 코드 표준이 정의되면 이를 바탕으로 표준 용어를 구성하고, 단어의 조합, 도메인 분류, 데이터 타입 길이, 코드값 등을 기준으로 해서 표준 적용이 업무적으로나 IT적으로 무리가 없는지 검토한다. 또한 검토 과정에서 누락된 단어, 도메인, 코드 등이 없는지를 확인하고 추가 보완 작업을 수행 한다.

### 요구사항 명세를 위한 프로토타입

추가적인 요구사항 도출 및 도출된 요구사항을 검토하기 위해서 프로토타입을 개발한다.  
프로토타이핑의 결과는 개발팀과 사용자의 의사소통을 개선하고, 보완해야 할 사항을 식별하는데 이용된다.

1. 프로토타이핑의 대상 범위와 구현 수준을 결정한다
   * UI 프로토타입 대상 식별 기준을 선정한다.
     + UI 프로토타입은 구축될 시스템의 전체 UI를 대상으로 하는 작업이 아니다.
     + 고객과의 의사소통 오류를 유발할 가능성이 높은 요구사항을 식별하고 해당 요구사항에 대해 선별적으로 UI 프로토타입 작업을 수행하는 것이 일반적이다.
     + 하지만 프로젝트의 특성에 따라 그 식별 기준을 다르게 할 수도 있다.
   * UI 프로토타입 대상을 식별한다.
     + 선정된 UI 식별 기준에 맞게 UI 프로토타입 대상을 식별한다.
     + UI 프로토 타입의 대상은 사용자가 사용할 화면 뿐만 아니라 보고서를 포함한 다양한 형태가 될 수 있다. 따라서 사용자 화면만을 UI 프로타입의 대상으로 식별해서는 안된다.
     + 로직, 화면 구성, 기술 구조 및 어플리케이션 구조, 처리 성능, 문서화 수준을 결정한다.
2. 프로토타이핑의 접근 방법을 선택한다.
   * + 수평적 대 수직적 프로토타이핑.
     + 진화적 대 비 진화적 접근 방법
3. 프로토타이핑의 구현 환경을 정의한다.
   * + 하드웨어, 소프트웨어, 개발 언어, 도구, 네트워크 환경, 데이터베이스
4. 결정된 구현 수준, 접근 방법, 구현 환경에 따라 프로토타입을 구축한다.
   * + UI 프로토타입 구현 방식을 선정한 후 이를 기준으로 UI 프로토타입을 구현한다.
5. 구축된 프로토타입에 대해서 고객과 함께 다음 사항이 목표대로 구현되었는지 검토한다.
   * + UI 프로토타입 검토 회의를 위한 범위, 참석자, 일정, 제반 환경 등을 준비한다.
     + 일정에 따라 참석자를 대상으로 프로토타입을 시연하고 시연 결과에 따라 제기된 쟁점을 조정한 후 문서화하고 이를 기반으로 요구사항을 정제한다
     + 기능 범위, 구현 수준, 접근 방법 등을 검토한다.
6. UI 프로토타입 검토 결과에 따라 요구사항을 정제한다.

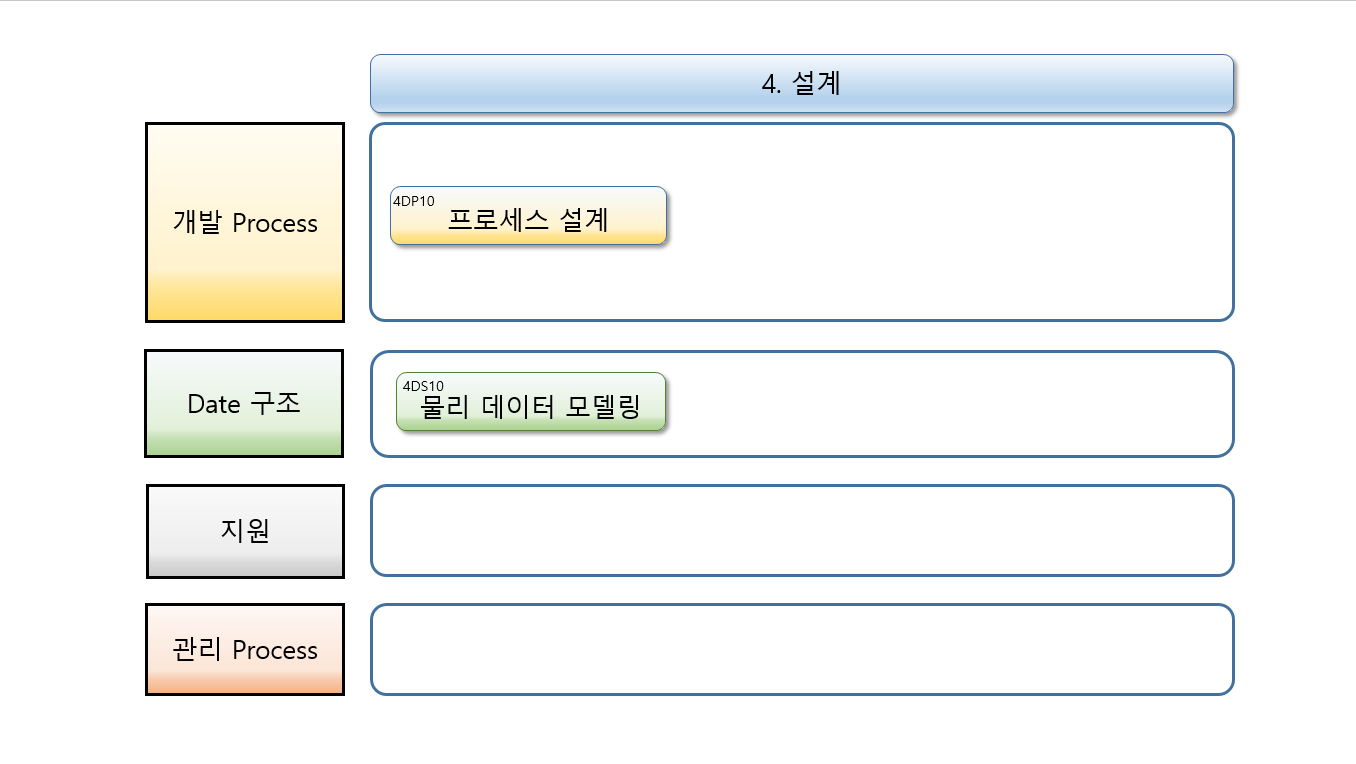
# 설계

## 정의

설계 단계에서는 개발자 관점의 충분한 구성과 세부 사항을 제시하여 고객의 요구사항이 어떻게 구현될 것인지를 구체적으로 나타 내어야 한다. 작성된 설계 산출물은 개발자가 시스템을 개발하기 위한 프로그램 개발과 단위 테스트 및 통합테스트 수행 시 활용된다.

분석단계가 “What”에 초점을 맞추었다면, 설계단계는 “How”에 초점을 맞추고 있으며, 정의된 문제에 대하여 솔루션을 제공하기 위한 방안을 구체적으로 강구하여야 한다.

## 작업 구성도



## 작업수행을 위한 준비

분석단계의 산출물을 검토 및 보완하여 설계 시 충분한 참고가 될 수 있어야 한다. 분석단계의 산출물은 프로젝트 참여인원과 공유 및 이해를 필요로 하며, 추가로 설계단계에 참여한 인원들을 위해 오리엔테이션을 실시하는 것이 좋다.

* + - 객체지향 설계기법에 대한 이해
    - BPMN에 입각한 프로세스 모델링
    - 물리 ERD에 대한 이해
    - 형상관리 절차에 따른 버전관리
    - 개발표준에 대한 이해

## 작업



### UI 설계

전체 프로그램 구성을 고려한 프로그램 목록을 도출하고, 도출된 프로그램 별로 코딩에 필요한 상세 로직을 설계하는 과정에서 UI(User Interface)를 설계한다. UI에 대한 목록을 프로그램에 맞게 도출하고, 도출된 프로그램 별로 코딩에 필요한 상세 로직을 설계한다. 프로세스와 UI에 대한 초안을 논리 데이터모델의 내용을 토대로 ‘UI 목록’을 작성한다.

* **표준 산출물 양식 : UI설계서**

##### 프로그램의 유형(온라인,배치,모듈)을 선정하고 프로그램과 관련된 화면과 테이블을 활용하여 UI에 연결한다.

##### 처리로직을 다음과 같은 항목이 포함되도록 상세하게 기술한다. : 정상/비정상 처리, 입/출력 데이터, 주요 계산식, 데이터 입력/조회/수정/삭제 조건 프로그램 초기화 로직, 분기조건, 출력 메시지 등

##### 프로그램의 일관성을 검토한다.

##### 시스템 전체적인 관점에서 프로그램 간의 커뮤니케이션, 데이터 액세스, 상세 설계의 수준을 점검한다.

##### 본 작업은 프로그램 설계를 직접 담당한 사람보다는 소프트웨어의 경험이 많은 팀내 다른 팀원이 수행하는 것이 바람직하다.

### 물리 데이터 모델링

물리 데이터 모델링은 논리 데이터 모델링에서 정의한 모델을 기준으로 논리 모델을 물리 모델로 변환하는 절차를 수행한다. 반 정규화, 저장공간 설계, 무결성 설계, 인덱스 설계 등 DBMS 의 물리적 요소를 적용한 물리 모델링을 수행한다.

* **표준 산출물 양식 : 물리데이터 모델 설계서**

#### 물리적 요소 조사 및 분석

##### 시스템 구축과 관련된 명명 규칙을 파악하여 물리 데이터 모델의 각 요소에 적용할 수 있도록 한다.

##### CPU, MEMORY, DISK, I/O Controller, Network 등 하드웨어 자원을 파악한다.

##### 운영 체제 및 DBMS 버전을 파악하고 적절하게 관리 되고 있는지 파악한다. ->특히 인스턴스 관리 기법 등에 대해서 파악하고 분석한다.

##### DBMS가 관리하는 파라미터의 종류와 관리 대상들을 정확하게 파악하고 정의한다.

##### 데이터베이스 관리를 위한 데이터 저장공간 관리 기법과 메모리 관리 기법 등과 관련된 파라미터들에 대해 고려한다.

#### 논리-물리 모델 변환

* 엔티티-테이블 변환

##### 논리 데이터 모델링 과정에서 정의된 엔티티, 속성, 관계들을 여러 가지 기준으로 물리 데이터 모델로 변환한다. 이 과정에서 엔티티 명에 해당하는 테이블 명을 생성하고, 속성 또는 관계에 해당하는 칼럼명을 생성하게 된다. 미리 생성된 데이터 표준(표준 용어, 표준 도메인, 표준 명명 규칙 등)을 따라 변환한다.

##### 서브 타입을 변환 할 때는 하나의 테이블로 통합, 여러 개의 테이블로 분할, 아크(ARC) 형태로 적용하는 3가지 방법으로 변환한다.

-> 각 테이블 변환 방법은 물리 데이터 모델링 가이드를 참조한다.

* 속성-칼럼 변환

##### 일반 속성을 전사적으로 미리 생성된 데이터 표준(표준 용어, 표준 도메인, 표준명명규칙 등)을 따라 변환 한다.

##### 논리 데이터 모델에서의 유일한 식별자(Primary Key)는 물리 모델에서 기본키로 변환한다.

##### 논리 데이터 모델에서 유일한 식별자 중에 해당 엔티티 자체에서 생성된 것 이외에 다른 엔티티로부터의 관계에 의해서 생성되는 관계 속성은 PK이면서 동시에 FK로 변환 한다.

* 관계 변환

##### 1:M 관계, 1:1 관계, 1:M 순환 관계, 배타적 관계 등으로 변환 한다. ->각 관계 변환 방법은 물리 데이터 모델링 가이드를 참조한다.

* 관리상 필요한 칼럼 추가

##### 예를 들어, 해당 데이터를 등록한 일자나 시스템 번호 등과 같은 칼럼을 추가한다.

##### 논리 데이터 모델에서 정의된 논리적인 데이터 타입을 물리적인 DBMS의 특성과 성능을 고려하여 최적의 데이터 타입을 선택한다.

## 참고사항



### 상세 물리 모델링 방법

#### 반 정규화

* 반정규화는 성능과 관리 효율을 증대시키기 위해 데이터의 일관성 및 정합성을 위배하는 위험을 내포하고 있는 과정이므로 충분한 검토를 통해 진행하도록 한다.
* 본 단계에서는 하나의 테이블을 수직 또는 수평으로 분할하는 과정을 고려한다.
* 수평 분할은 특정 테이블에 데이터가 너무 많이 있고 레코드 중에서 특정한 범위를 주로 엑세스하는 경우에 고려한다.
* 수직 분할은 조회위주의 칼럼과 갱신위주의 칼럼이 분명히 나뉘는 경우에 고려한다.
* 많은 양의 정보들이 자주 Group by, sum 등과 같이 집계함수를 이용해서 실시간으로 통계 정보를 계산하는 특정 통계 테이블을 위해 중복 테이블이나 중복 칼럼을 생성 할 수 있다.

#### 저장 공간 설계

* 초기 사이즈, 증가 사이즈, 트랜잭션 관련 옵션, 최대 사이즈와 자동 증가 증을 고려하여 테이블 저장 용량을 분석한다.
* 데이터 증가 예상 건수, 주기, 로우 길이(Row Length)등을 고려하여 테이블 용량을 분석한다.
* 테이블 인덱스에 대한 크기를 기준으로 오브젝트 별 용량을 산정한다.
* 테이블 스페이스별 오브젝트 용량의 합계로 테이블 스페이스별 용량을 산정한다.
* 테이블 스페이스에 따른 디스크 용량과 I/O 분산 설계를 통해 디스크 용량을 산정한다.  
   -> 무결성 설계 가이드는 물리 데이터 모델링 가이드 참조

#### 무결성 설계

* 데이터 무결성은 실체 무결성, 영역 무결성, 참조 무결성, 사용자 저읭 무결성의 순서로 점검한다.  
  ->무결성 설계 가이드는 물리 데이터 모델링 가이드 참조

#### 인덱스 설계

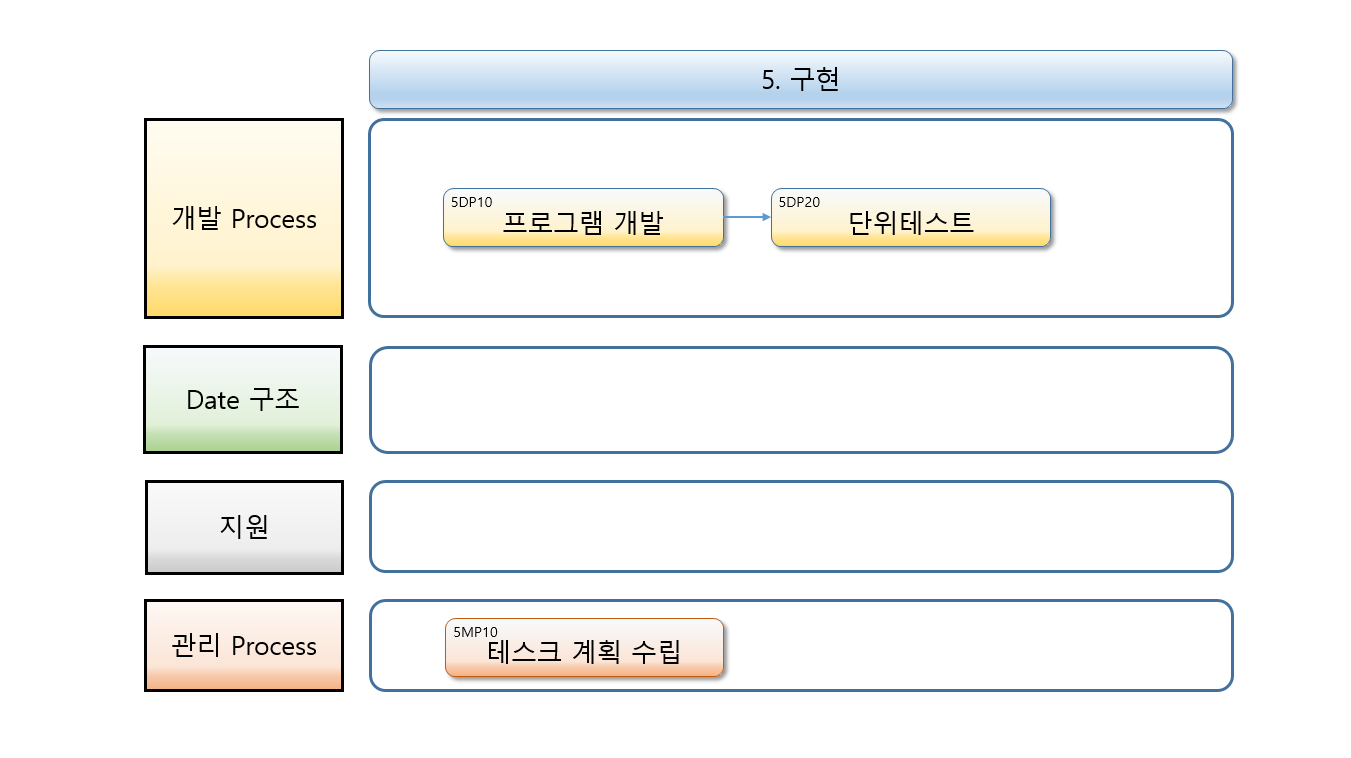
* 인덱스는 테이블에 접근하는 모든 경로를 수집하고 수집된 결과를 분석하여 종합적인 판단에 의해 결정한다.
* 반복 수행되는 접근 경로, 분포도가 양호한 칼럼, 조회 조건에 사용되는 칼럼, 자주 결합되어 사용되는 칼럼, 데이터 정렬 순서와 그룹핑 칼럼, 일련번호를 부여한 칼럼, 통계 자료 추출 조건, 조회 조건이나 조회 조건 연산자 등의 유형을 고려하여 인덱스 생성이 필요한 칼럼들에 대해 먼저 접근 경로를 수집한다.
* 수집 된 접근 경로 칼럼들을 대상으로 분포도를 조사한다.(설계 단계에서는 실제 분포도를 예측 할 수 없으므로 현재 시스템 데이터를 참고하거나 업무에서 예상한 상황을 고려하여 분포도를 예측한다.)분포도 조사에 의해 후보 칼럼을 선정한다.
* 인덱스 후보 목록을 이용하여 접근 유형에 따라 어떤 인덱스 후보를 사용할 것인지 결정한다. 누락된 접근 경로가 있다면 분포도 조사를 실시하고 인덱스 후보 목록에 추가 작업을 반복한다.
* 단일 칼럼의 분포도가 양호하면 단일 칼럼 인덱스로 확정하고 하나 이상의 칼럼이 조합이 필요한 경우는 인덱스 선정 가이드를 활용하여 인덱스 칼럼 순서를 결정한다.
* 설계된 인덱스를 적용하고 접근 경로 별 인덱스가 사용되는지 확인한다.

# 구현

## 정의

본 단계는 설계 단계를 통해 작성된 산출물을 기준으로 소스코드를 코딩하고 시스템에 설치하여 개발된 시스템이 기능/비기능 요구사항에 적합한지를 단위 시스템 수준에서 검증하는 단계이다.

## 작업구성도



## 작업수행을 위한 준비

설계단계 산출물을 검토하고 미비사항이 있을 경우 이를 보완하여 구현단계로 진입하는데 문제가 없도록 한다. 구현단계에 신규로 투입된 개발자 혹은 참여자를 위해 오리엔테이션이 필요하며 개발표준에서부터 설계단계까지의 진행경과를 전반적으로 설명할 필요가 있다. 아래의 산출물을 참고하여 구현 프로세스를 위한 입력물로 활용한다.

* 유스케이스 설계서
* 프로그램 목록과 프로그램 설계서
* 물리 데이터 모델
* 개발환경
* 테스트 계획서

## 작업



### 프로그램 개발

프로그램 설계서에 정의된 프로그램 사양을 만족시키는 프로그램을 개발하기 위해 소스코드를 개발한다. 개발된 프로그램은 프로젝트에서 정의한 개발 표준을 준수해야 하고, 테스트 환경에서 동작해야 한다. 개발된 프로그램은 테스트 단계에서 진행될 일련의 테스트 과정을 수행할 수 있게 준비되어야 한다. 산출물은 시스템에 저장된 소스코드이다.

* **표준 산출물 양식 : 별도로 정의하지 않음**

### 단위테스트

개발자가 스스로 개발한 소스코드에 대하여 단위테스트를 수행한다. 필요한 경우 동료 간의 검토활동을 통해 단위테스트를 수행할 수 있다. 현장에서 흔히 발생하는 문제는 충분한 단위테스트를 수행하지 않은 상태에서 통합테스트를 시작하게 되는 경우이다. 이는 개발자 스스로 본인이 개발한 프로그램에 대하여 충분한 테스트를 수행하지 않기 때문인데, 잦은 변경과 충분한 개발시간을 부여하지 않은 원인에 의해 개발에만 급급하게 되어 이러한 상황이 발생하는 경우가 많다.

* **표준 산출물 양식 : 단위테스트 결과서**

### 테스트 계획수립

고객의 비기능/기능 요구사항을 만족한다는 것을 보장하기 위한 인수테스트 활동을 계획한다. 프로그램 모듈간 통합이 제대로 이루어 졌으며, 설계 요구사항을 만족한다는 것을 보장하기 위한 통합테스트 활동을 계획한다. 프로그램 로직이 완전하며, 단위 모듈이 버그없이 수행된다는 것을 보장하기 위한 단위테스트 활동을 계획한다.

* **표준 산출물 양식 : 테스트 계획서**

### 인수테스트 계획수립

인수테스트는 고객이 수행하는 것이다. 이를 위해 고객이 테스트를 수행할 수 있도록 인수테스트 활동을 아래와 같이 계획한다.

* 인수테스트의 목적을 기술하고 수행할 테스트 종류와 절차를 수립한다.
* 테스트 전에 선행될 기준과 완료 기준 및 중단/재개 기준을 정의한다.
* 테스트 대상범위와 제외범위를 설정한다
* 테스트에 필요한 H/W, S/W 리스트를 기술한다.
* 테스트 조직과 역할을 정의하고 테스트 주요 활동에 대한 상세 일정을 수립한다.
* 테스트 데이터 생성 및 유지 절차를 기술한다.
* 테스트 수행 시 고려사항 및 산출물을 정의한다.

### 통합테스트 계획 수립

모듈간 통합이 제대로 이루어졌으며, 설계 요구사항을 만족한다는 것을 보장하기 위한 통합테스트 활동을 계획한다.

* 통합테스트의 목적을 기술하고 테스트 수행절차를 수립한다.
* 테스트 전에 선행될 기준과 완료 기준 및 중단/재개 기준을 정의한다.
* 테스트 대상범위와 제외범위를 설정한다.
* 테스트에 필요한 H/W, S/W 리스트를 기술한다.
* 테스트 조직과 역할을 정의하고 시스템 테스트 수행 주요 활동에 대한 상세 일정을 수립한다.
* 테스트 데이터 생성 및 유지 절차를 기술한다.
* 테스트 수행 시 결함 및 진척관리 계획을 수립한다.
* 테스트 수행 시 고려사항 및 산출물을 정의한다.
* 통합테스트 시나리오를 작성한다.

##### 테스트 계획서를 통해 테스트 범위를 이해한다.

##### 인터페이스 설계서를 통해 인터페이스 대상을 이해한다.

##### 아키텍처 정의서를 통해 시스템 전체 구성과 인터페이스 구간을 이해한다.

##### 인터페이스 발생 업무를 식별하고 테스트 시나리오를 문서화한다.

* 통합테스트 케이스를 작성한다.

##### 테스트 시나리오 별로 테스트 케이스를 작성한다.

##### 테스트 케이스 수행절차와 수행조건 및 테스트 종류별 입력 값(예: 성능 테스트 수행시 부하 정도)을 기술한다.

##### 스크립트 사용시 스크립트 명도 포함하여 기술한다.

* 통합테스트 데이터를 준비한다.

##### 통합테스트 케이스를 실행하기 위해 테스트 데이터를 준비한다.

##### 만약 테스트 데이터를 특정한 제품 환경에서 가져온다면, 동일한 환경에서 테스트를 하기 위해 필요한 모든 데이터 다운로드 스크립트를 작성한다.

### 단위테스트 계획 수립

프로그램 로직이 완전하며, 단위 모듈이 버그없이 수행된다는 것을 보장하기 위해서 단위테스트 활동을 계획한다.

* 단위 테스트의 목적을 기술하고 단위 테스트 계획, 설계, 실행 및 평가 단계에 따른 절차와 담당자, 산출물을 기술한다.
* 단위 테스트 전에 선행될 기준과 단위테스트가 완료되는 기준 및 단위 테스트 중단/재게 기준을 정의한다.
* 단위 테스트 케이스를 도출하기 위한 전략을 수립하고 점검항목을 기술한다.
* 단위 테스트 수행을 위한 교육일정을 수립한다.
* 단위 테스트 범위 및 범위 밖의 기능을 선정한다.
* 테스트 환경 및 테스트 조직과 역할을 정의하고 단위테스트 수행 주요 활동에 대한 상세 일정을 수립한다.
* 단위 테스트 수행 시 결함, 진척관리 계획을 수립하고 발생되는 산출물을 정의한다.
* 단위테스트 케이스를 작성한다.

##### 단위테스트 대상에 대한 테스트 케이스를 작성하며, 이때 케이스 내용, 사전조건, 테스트 데이터 예상결과를 작성한다.

##### 테스트 케이스는 값에 따른 분기와 업무 수행 경로로 고려하여 작성한다.

- 단위테스트 케이스를 작성한다.

##### 단위테스트 케이스를 실행하기 위해 테스트 데이터를 준비한다.

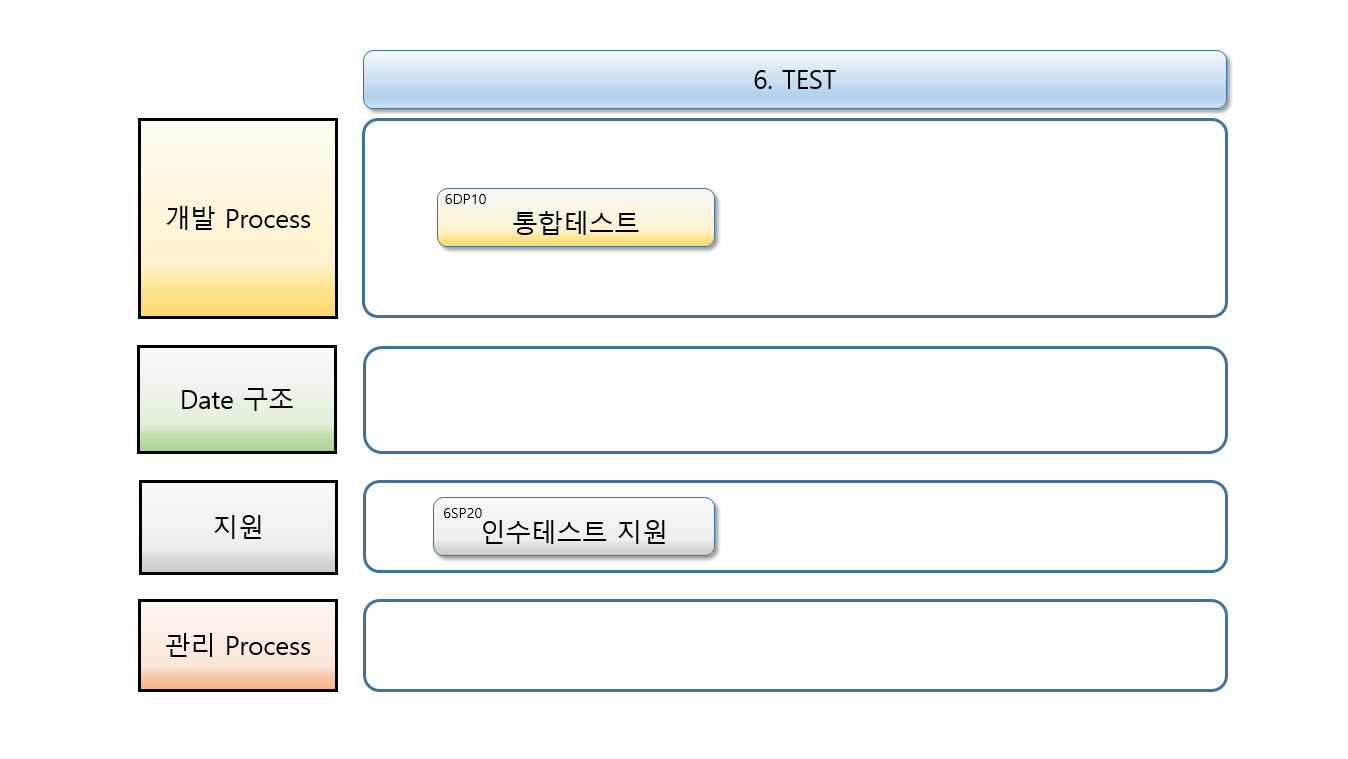
##### 만약 테스트 데이터를 특정한 제품 환경에서 가져온다면, 동일한 환경에서 테스트를 하기 위해 필요한 모든 데이터 다운로드 스크립트를 작성한다.

# 테스트

## 정의

지금까지의 단계가 프로세스 위주로 산출물에 대한 정적 분석을 통해 품질을 점검하였다면, 테스트는 실제 시스템을 작동시켜보고 이상이 없는지 확인하게 된다. 단위테스트에서는 소스코드의 정적 분석이 개발자 스스로에 의해 확인이 되고, 일부 동적분석을 통해 작동여부를 확인하였다면, 이제는 제3자적 관점에서 시스템을 직접 작동시키면서 프로덕트(product)위주의 품질관리를 하게 되는 과정이다.

## 작업구성도



## 작업수행을 위한 준비

구현단계에서 수립된 테스트 계획에 따라 작업을 수행한다. 테스트 환경은 이미 갖추어져 있을 것이며, 별도의 산출물로 정의하지 않았지만, 테스트 시나리오와 케이스를 준비할 필요가 있다. 테스트 시나리오와 케이스는 테스트방법론에 해당하여 본 AM the Method에서는 상세히 다루고 있지 않지만, 동적테스트를 정상적으로 수행하는 조직은 내부 자산으로 갖출 필요가 있다.

* 테스트용 데이터 확인
* 테스트 케이스와 시나리오
* 단위테스트 결과서
* 통합테스트 계획서, 인수테스트 계획서

## 작업



### 통합 테스트

전체 시스템의 통합이 완료될 때까지 단위 시스템 간의 연계성(Interface) 및 기능 요구사항들을 검증하고 하드웨어와 소프트웨어 구성 요소 간의 상호 작용을 테스트하는 활동이다. 통합테스트는 단위테스트를 수행했던 모든 프로그램을 포함해야 하며, 단위 시스템 간에 수행되는 인터페이스 기능과 통합적 기능수행에 초점을 맞춘다는 점에서 단위테스트와 구분된다.

* **표준 산출물 양식 : 통합테스트 결과서**

### 인수 테스트 지원

전체 시스템이 기능/비기능 요구사항을 만족하는지 검증하기 위해 하드웨어, 소프트웨어(시스템 S.W, 응용프로그램)를 포함한 테스트 활동이다. 시스템테스트는 운영 환경과 가장 유사한 환경에서 수행되어야 하며 설계 단계에서 계획된 시스템테스트 계획, 시스템테스트 시나리오 및 테스트 케이스에 의해 수행된다.

**표준 산출물 양식 : 별도로 정의하지 않음**

# 종료

## 정의

프로젝트 수행 결과를 공식적으로 수용하고 프로젝트를 마무리 하는 단계이다. 모든 활동을 종결시키고 완료된 산출물과 시스템 혹은 프로그램을 고객에게 인계한다. 종료단계에서는 계약이 종결이 이루어지고, 모든 받아야 할 것은 받고, 모든 주어야 할 것은 주어야 한다.

## 작업구성도



## 작업수행을 위한 준비

프로젝트의 종결처리를 위한 공식적인 절차가 필요하다. 고객이 원하는 시스템은 인수인계를 완료하여 제대로 가동되어 사용토록 하고, 협력회사 혹은 외부 개발인력에게는 대가를 지불한다. 내부적인 지식과 인력의 평가, Lessons Learned를 정리한다.

* 산출물 최종 품질점검 및 제출
* 고객의 인수
* 프로젝트 완료보고 준비
* 계약의 종결처리

## 작업

### 프로젝트 종료보고

프로젝트 종료보고 작업은 현장의 필요에 의해, 혹은 조직의 관행에 의해 발의될 수 있다. 종료보고회를 공식적으로 개최하는 경우도 있고, 내부 결재를 위해 보고를 필요로 하는 경우도 있다. 종료보고서의 내용은 프로젝트 착수부터 종료단계까지의 모든 내용이 포함될 수 있지만, 핵심적인 내용을 중심으로 작성하게 된다.

* **표준 산출물 양식 : 프로젝트 종료보고서**