빌드업 특강 - 4차 산업시대의 SW의미

| 장성택 강사님

* ‘프로젝트’ 란? : 수주업종에서 업무를 진행하는 대표적인 방식
* 전 세계의 GDP의 23%는 프로젝트를 통해 생성된다. -> 오늘날 필연적인 업무방식
* 수주업의 개념 : 고객이 발주한 사업을 타 회사들과의 경쟁을 통해 획득하고 이를 수행하는 business를 영위하는 업종을 의미함

‘사업 기획 및 발주’ -> ‘제안’ -> ‘계약 및 협상’ -> ‘프로젝트 수행’

(수주를 위한 경쟁이 치열함!)

* 사업 발주 및 제안 프로세스

1. 프로젝트 타당성 검토(Feasibility Study)

: **Why?** **What?** (사업계획 – 사업범위, 일정, 적용기술, 예산) **Benefit!**

1. 제안평가(경쟁입찰 방식)

* 평가 주체
* 사업을 발주한 고객,
* 제안서를 객관적으로 평가할 수 있는 외부 전문가 등 평가위원(공공)
* 평가 대상
* 기술평가: 제시된 실행안(기술) -> 제안서, 요약서, 발표(PT) 자료
* 가격평가: 제안가격 -> 제안가(입찰가)
* 평가 요소
* 공식요소: 제시된 수행방안(기술)의 적합성 및 차별성, 가격의 적합성
* 비공식 요소: 고객/평가위원의 사업자에 대한 호감도, 친밀도, 관계 등

1. 사업참여 의사결정

* 사업 수행 가능 여부, 매출 규모와 이익률, 고객과의 전략적 관계 등을 고려하여 결정
* 수주 검토사항:
* 현실적인 ‘수주 확도’ 는 어느 정도인가? (솔직히 될까 안될까?)
* 성공적인 수행이 가능한가? (기술, 관리, 위험 검토)
* 매출 규모와 이익율은 어느 정도인가? (100억짜리 사업 1개 > 10억짜리 사업 10개)

(\* IT 업계의 평균 이익률: 5% - 그만큼 실패 확률이 높다! )

- 고객과의 관계 유지를 위해 전략적으로 필요한가?

(고객과의 관계 유지를 위해 (돈이 안돼도)울며 겨자먹기로 하는 일도 있다는 소리)

* 프로젝트 관리
* IT 프로젝트 유형은 크게 **컨설팅(IT Consulting)**, **시스템 구축(System Integration)**, **운영 및 유지보수 (IT Outsourcing)**으로 구분됨.
* **컨설팅** (\* 컨설팅 비용 > 개발자 단가 (최고 2배에서 20배) -> 객관성 확보 및 정치적인 목적으로)
* 기업의 중장기 IT 전략 수립
* 단위 IT 사업(프로젝트) 기획
* **시스템 구축**
* 시스템 개발, 구축 및 통합,
* 기반 인프라 구축 및 고도화
* **운영 및 유지보수**
* 구축된 시스템의 안정적인 운영
* 시스템 기능/품질 개선 및 유지보수

(\* SI와 SM의 차이점)

|  |  |
| --- | --- |
| **SI** | **SM** |
| * 새로운 시스템에 대한 사업 위주 * 신기술 적용 및 기존 시스템과의 연계통합   (상대적 난이도 ↑)   * 고객 변경 빈번 * 이직이 상대적으로 용이(풍성한 resume) | * 기구축(legacy) 시스템에 대해 다룸 * 시스템 고도화 및 개선   (상대적 난이도 ↓)   * 고객이 고정되는 경우가 많음 |

* IT 프로젝트
* IT프로젝트 규모
* 소형 프로젝트

: 0명 이내 수행인원, 수행기간 6개월 이내 프로젝트 (예산규모 ~10억)

* 중형 프로젝트

: 00명 정도 수행인원, 수행기간 12개월 이내 프로젝트 (50억 ~ 100억대)

* 대형 프로젝트

: 000명 이상 수행인원, 수행기간 2년 이상 프로젝트 (최소 1000억)

‘차세대 프로젝트’ 라는 이름이 많이 붙음

→ 프로젝트 규모가 작을수록 성공확률은 높아지며, 프로젝트 규모가 커질수록 성공확률이 현저히 작아진다.

(‘차세대’ 란 이름이 붙은 프로젝트 중 우리나라에 성공한 프로젝트는 없다:

기술적 난이도 높고, 연계 프로세스도 많고, 관련 인력도 너무 많다 -> 기한초과, 예산초과, 품질충족X)

* 프로젝트 현실
* 전세계 IT 프로젝트의 28%는 중도 하차하며 오직 26%만이 예산과 기한 내에 완료된다. //The Standish Group

(Scope Creep, Poor Requirement Gathering, Unrealistic planning and scheduling, Lack of Resources

* IT 프로젝트의 특징
* 프로젝트 진행상황을 가시적으로 정확히 판단하기 어려움 (Resource ~ Output? )

(코드를 완성시켰으면 100% 진행? -> 0%가 될 수도 있다.)

* 투입인력(개발자) 의존도가 높은 속성을 갖음 (사람의 skill에 매우 종속적임 (자동화 수준↓))
* 초기에 사용자 요구사항은 명확히 정의하기 어려움(테스트시기 시연할 때야 구체적 파악)
* 대부분 일괄계약 형태로 진행하여 초기 정확한 규모 산정이 어려움
* IT 프로젝트 주요 실패요인 (Source: 감리협회 세미나, The Standish Group)

|  |  |
| --- | --- |
| 국내 주요 실패요인 | 국외 주요 실패요인 |
| * 고객의 잦은 요구변경 (21%) * 기술, 경험 부족 (20%) * 부정확한 기간, 비용 예측 (20%) * 고객, 현업의 참여 부족 (12%) * 영업 부서의 과욕 (5%) | * 불완전한 요구사항 (13.1%) * 사용자 참여 부족 (12.4%) * 자원 부족 (10.6%) * 비현실적인 기대감 (9.9%) * 경영층의 지원 부족 (9.3%) |

* 크런치 모드
* 크런치 모드는 프로젝트의 마감일(Deadline)을 앞두고 일정을 맞추기 위해 팀원들이 수면, 영양,

사회생활, 위생 문제 등을 포기한 채 고강도 근무체제를 유지하는 방식을 의미하는 용어

(게임 업계에서 알려진 용어(게임업계 크런치 모드 경험 84%))

* 게임 업계를 통해 알려졌지만 과거 IT 업계 전체에 만연해 있었던 프로젝트 수행 방식으로 현재 이를 개선하기 위한 사회적 분위기 조성과 노력이 지속되고 있음 (ex. 주 52시간 근무제도)
* 동/서양 프로젝트의 차이
* 서양의 문화는 사업 목표라는 개체 중심으로 프로젝트를 인식하여 계약을 체결하고 프로젝트 수행 중 발생하는 계약 범위 밖의 요구사항은 쉽게 거절할 수 있기 때문에 수행 및 관리가 비교적 용이함
* 동양권에서는 초기에 설정한 계약상의 프로젝트 목표도 중요하지만, 고객과의 관계 유지가 더 중요하기 때문에 고객의 요구에 따라 목표와 범위가 수시로 변동되어 수행 및 관리가 매우 어려움

“서양은 계약 중심의 프로젝트를, 동양은 계약 + 관계 중심의 프로젝트를 수행” //생각의 지도

* 프로젝트 관리 필요성
* 점점 복잡해지고 많은 이슈를 내재하고 있는 프로젝트의 성공적인 수행을 위해서는 체계적인 프로젝트 관리가 필수 사항
* 프로젝트 관리영역은 총 10개 영역으로 구성되어 있다. (PMBOK 지식영역)

범위관리, 일정관리, 원가관리, 품질관리 | 자원관리, 의사소통관리, 위험관리, 조달관리, 이해관계자 관리

(Core Area(목표, 4개), Facilitating Area(수단, 5개) + 통합관리(다양한 관리요소들을 적절하게 통합, 조정))

* 프로젝트 관리 핵심 요소 (Scope, Cost, Time)
* 과제 성격의 소규모 프로젝트에서는 범위, 일정, 품질에 초점을 맞추어야 함.
* 프로젝트 수행 방법론
* Waterfall 모델(traditional): 1->2->3 -- n <-> Agile 방법론: sprint1, sprint2, sprint3 …
* 기존 방식: 전체 프로세스가 분석 -> 설계 -> 개발 -> **테스트** -> 검증 (결과 확인이 느리다)
* Agile방법론은 sprint라는 일정한 고정기간의 소규모 cycle로 전체 프로젝트를 나누어 진행한다.
* Agile 방법론(SCRUM) 은 ‘고객 지향적’, ‘work smart’ 라는 어휘로 많이 지칭된다.
* 단, 고객 지향적이라는 말을 쓰고 있지만 현장에서는 별로 좋아하지 않는 방식

(‘우선순위’에 집착하는 Agile 방식. -> 요구사항 ‘추가’ 보다는 ‘소거’ 느낌이 들게 됨.

기존 waterfall 방식에서 기한이 늦춰지는 것에 대한 penalty만 부과하면 고객 측에선 손해 안 봄)

→ 통상 국내에서 Agile 방식은 고객이 없는 내부 R&D 프로젝트에서 사용되게 됨)

* 정리
* 프로젝트 착수단계:
* PM을 선입하고 팀원 별 명확한 역할과 책임을 정립해야 한다.
* 팀 내 협의 하에 프로젝트 Ground Rule(기본 규칙)을 정립

(회의가 많으면 일을 못하지만, 회의가 부족하면 일이 산으로 간다. )

회의방식: 논의사항, 이슈사항, 해결할 사항 -> 리더가 다시 적어서 메일로 재 배포(결국 최종보고서 化)

실제 현업에서 사용하는 방식과 최대한 유사하게 진행해볼 것

* WBS(Work Breakdown Structure)를 설계할 것

: 일정계획표, 업무 분장표를 작성 (Microsoft Projects, Microsoft Excel)

* 프로젝트 일정 계획 수립 시 고려해야 할 사항
* 개발 내역에 대한 기술적인 난이도를 사전에 평가해야 한다.
* 업무를 담당하는 팀원 각각의 능력을 고려한다. (fair한 프로젝트 업무 배정)
* 개발과정에서 예상치 못한 이슈가 발생하는 경우가 많다 -> 이슈 해결할 버퍼를 고려해서 일정 계획

(ex/ 완료일 기준으로 1주일 정도 유예기간을 만들어 놓는다. -> 기존 일정도 compact하게 진행하게 됨)

* 프로젝트 실행단계에서 고려해야 할 사항
* 기능, 데이터, UI, 아키텍처 요구사항 등을 고려한 체계적인 설계
* 불필요한 코드를 줄이고 테스트와 재작업이 쉬운 코드 구성
* 개발과정에서 무한 수정/변경 등의 삽질을 방지 -> 설계로서 해결 가능
* 설계로서 향후 수정 및 유지보수가 용이하여 운영 생산성이 향상될 수 있음

(이후 내용 교재필기로 대체)