

total

- 1주차

- ▼ 가설 설정

- 가설 x n = 리서치 (수집한 자료와 가설이 부합할 때까지)
- 이론이나 직관에 근거하여 가설을 수립하고, 이를 검증하기 위해 자료를 수집합니다. 그렇기에 가설을 clear하게 정리할수록 리서치 목표 또한 확실해집니다.

- ▼ 시장 분석

- 종류
 - 문제가설
 - XYZ 가설
 - 가장 위험한 가설

- ▼ 고객 분석

- 내가 타겟할 고객을 명확히 정의할수록, 서비스를 만드는 과정에서 방향을 잃지 않을 수 있습니다
- 고객 정의 : 내 서비스의 고객이 누구인가'를 명확하게 정의하는 과정
- 고객 Needs / Pain Points 분석 : 고객의 세부 Pain Point를 파악하여 각 Pain Point를 해결해줄 수 있는 Solution을 도출하기 위한 과정

- ▼ 사례 분석

앱 서비스를 기획하다보면 내가 생각한 아이디어는 이미 시중에 있는 경우가 많습니다. 경쟁 제품을 분석하여 차별화 포인트를 고민해내는 것이 중요합니다.

- 경쟁 제품 선정 기준 수립 : 경쟁 제품을 어떤 기준으로 선정할 것인가
- 경쟁 제품의 개요, 기능 목록, 한계 파악 : 위에서 정의한 문제를 해결하기 위해 택하고 있는 솔루션

- 2주차

- 생각하는 방법

- ▼ JTBD

- Jobs To Be Done: 고객이 제품을 통해 '얻을 수 있는 것'을 의미

- 기능적 차원, 사회적 차원, 정서적 차원으로 구분할 수 있습니다. JTBD는 제품의 본질을 정확히 이해하는 데 초점을 둡니다.

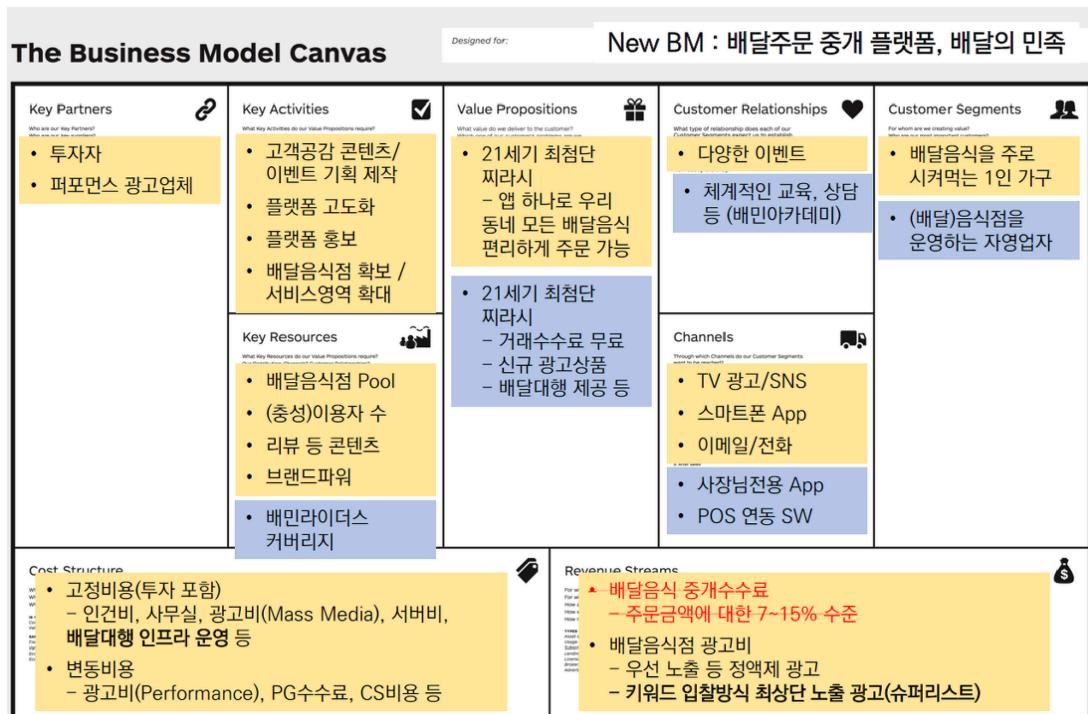
▼ PMF

- Product Market Fit : 나의 회사(제품)가 시장의 본질에 딱 맞는 경우를 의미합니다.
- 위 사진의 자물쇠에 열쇠가 꼭 들어 맞는 것과 같습니다. 열쇠로 자물쇠를 풀 때 중요한 점은 최소화된 기능으로 최적화된 모형을 모델링해야 한다는 점입니다.

- 설계하는 방법

▼ BM

- Business Model
- 비즈니스 모델이란, 하나의 조직이 어떻게 가치를 창조하고 전파하며 포착해내는지를 합리적이고 체계적으로 묘사해낸 것입니다.
- key partners : 협업해야하는 사람들
- key activities : 핵심적인 목표
- key resources : 비즈니스를 시작할 때 필요한 모든 것(돈, 사람, 지적 자산 등등)
- value propositions : 사용자들에게 전하는 메인 기능
- customer relationships : 팬, 단골을 만드는 과정 (1명이 100번 들어오는 어플)
- channels : 어떻게 고객에게 다가갈 매체
- customer segments : 이 유저는 어떤 목표를 가지고 이 어플을 사용할 것인가?
- cost structure : 비즈니스에 드는 비용, 고정비용 및 변동비용
- revenue streams : 수익원



• 공유하는 방법

▼ One Pager

- One Pager는 제품 개발 및 프로젝트 진행에 대한 구성원들의 이해, 공감, 명확한 역할 부여를 위해 작성합니다.
- 산재되어 있는 아이디어 및 생각을 One Pager 작성 과정에서 모으고 구체화할 수 있습니다. 기록하는 과정에서 추상적인 내용들을 보다 명확하고 구체적으로 구현할 수 있게 됩니다.

▼ OKR

- OKR이란 조직 성과 관리 방법론 중 하나입니다.
- 좋은 OKR은 시간 제한이 있는 목표와 측정 가능한 마일스톤으로 구성됩니다.

▼ KPI

- **Key Performance Indicator: '핵심성과지표'**
- '성과 지표'는 '개인 또는 조직의 성과를 객관적으로 평가하는 기준, 척도'로 정의됩니다.
- **조직이나 회사의 목표하는 바를 달성하기 위한 지표를 수치화, 성공도 측정**

▼ MBO

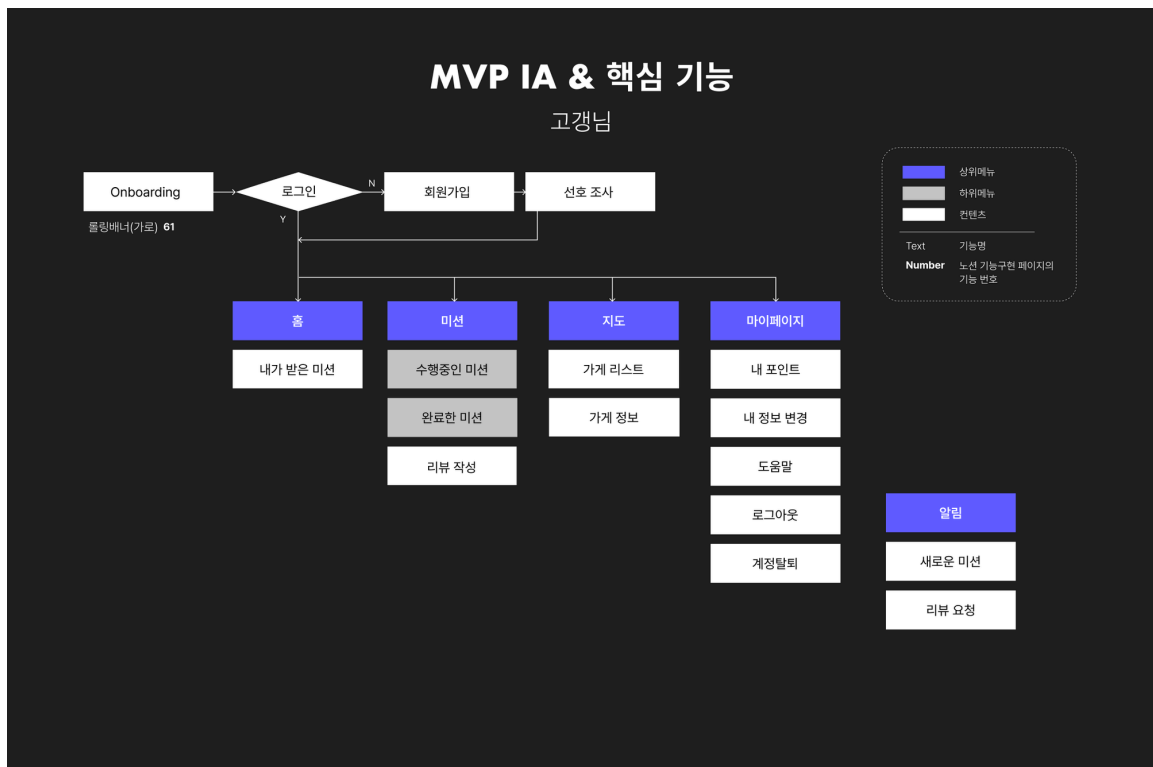
- **Management by Objectives : 목표에 의한 관리**

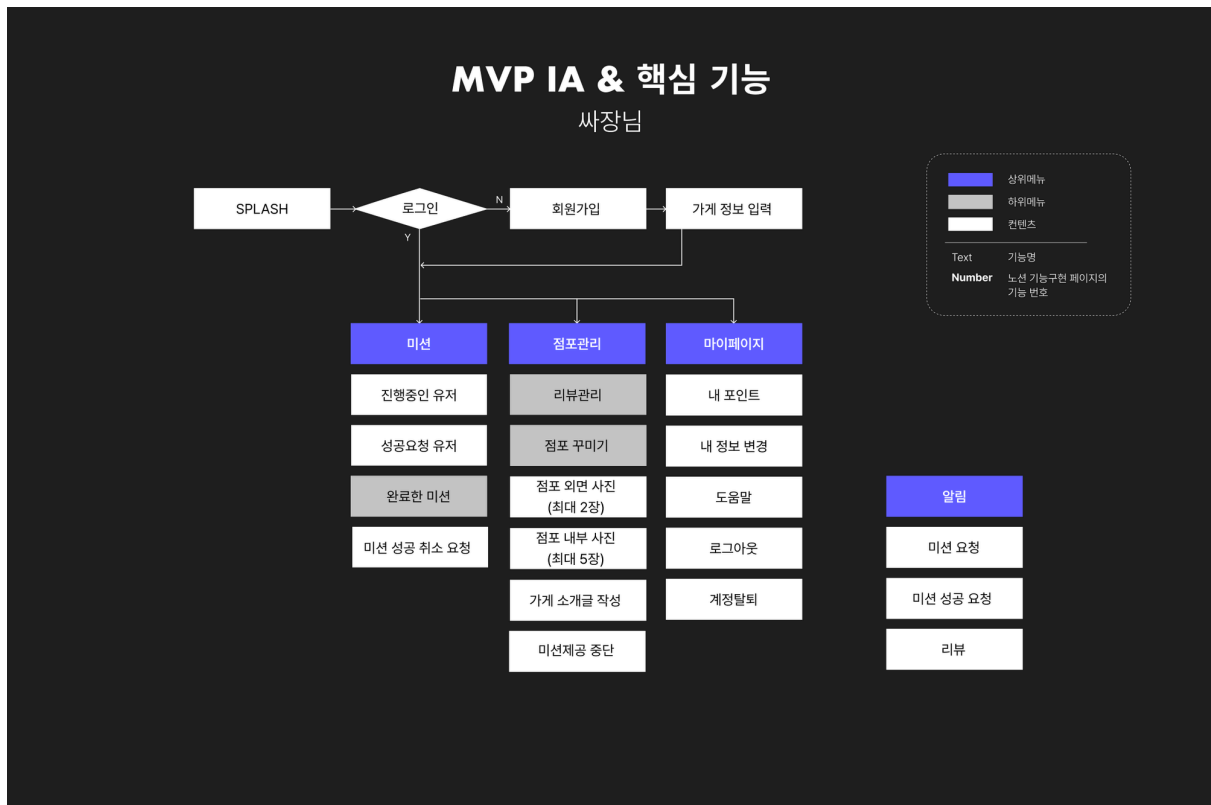
- 조직의 목표와 개인의 목표를 통합하여 종업원 각 개인을 중시하면서 동시에 기업의 실적을 향상 시키는 것을 목표로 하는 방법론입니다.
- 스스로의 객관적인 평가

• 3주차

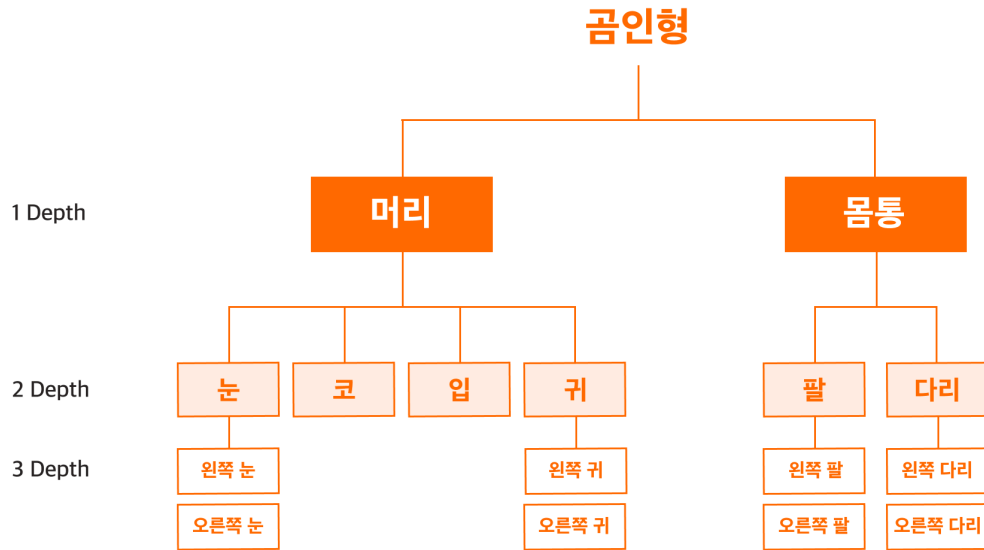
▼ IA

- Information Architecture
- 웹/앱 서비스를 구축할 때 메뉴의 정보 구조를 설계하고 정의하는 문서
- 화면들의 흐름을 시각화





- 주요 용어 depth :
 - 메인을 기준으로 0 depth로 시작하여, 서비스 화면의 흐름에 따라 1 depth, 2 depth, 3 depth 순서로 정의
 - 일반적으로 depth가 깊을수록 유저가 목표를 성취하기 위해 거쳐야 할 단계가 많음
- 구조
 - 계층 구조(트리형): 중첩되어 계속 depth가 깊어지는 경우를 표현



- 순차 구조(연속형): 탭과 같이 병렬적인 화면을 표현

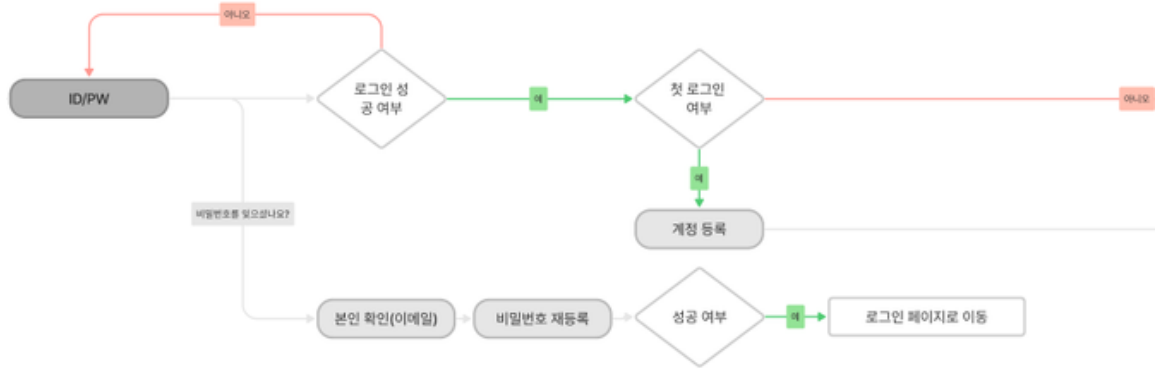
* 순차 구조(sequential structures)

순차 구조는 제한된 정보를 순차적으로 제공하여 사용자가 필요한 정보를 수집하도록 유도합니다. 이 방식은 특정 줄거리가 있는 정보를 전달하는 데 효과적이지만, 탐색 과정이 단조롭게 느껴질 수 있습니다. 또한 중간에 마지막 단계로 바로 이동할 수 없기 때문에 다른 방식의 구조와 함께 사용하는 것이 좋습니다. 이 방법은 보험 가입, 구독하기, 예약하기 시스템 등에서 많이 사용됩니다.

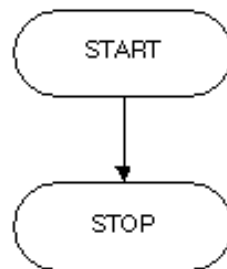


▼ Flow Chart

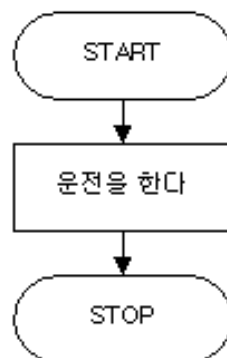
- 순서도, 흐름도
- IA 발전 ver, A보다 더 상세하게 분류가 되어있어, 화면 이동상의 논리 오류를 찾기에도 쉽습니다.



- 작성법
- 단말기호(Terminal) : 프로세스의 시작과 끝을 나타냅니다. 모서리가 둥근 박스 내 START, STOP으로 알려줍니다.



- 처리기호(Process) : 모든 처리를 표시합니다. 일상적으로 하는 일을 직사각형 안에 써 주면 됩니다.



- 판단기호(Decision) : 논리를 판단하여 참 또는 거짓의 분기를 작성합니다. '예'와 '아니오'라는 답변만 가능

- 4주차

- 워터폴 vs 애자일

- ▼ 워터폴

- 요구사항 정의(설계) → 디자인 → 개발 → 테스트 → 배포
 - 요구사항 정의(설계)
 - 고객의 문제를 정의하고 요구사항을 문서화하여 정리하는 단계
 - 어떤 작업이 필요한지, 필요한 리소스는 무엇인지, 우선순위는 무엇인지 등
 - 디자인
 - 제품을 설계하는 단계
 - 개발
 - 제품을 구현하고 만드는 단계
 - 테스트
 - 기능이 제대로 작동하는지 테스트하는 단계입니다.
 - 배포가 되기 전에 문제가 될 만한 버그, 오류를 찾아 수정
 - 배포
 - 장점
 - 맡은 업무가 명확
 - 현재 상황을 추적하기 좋음
 - 단점
 - 느린 속도
 - 시장 상황을 빠르게 반영x

- ▼ 애자일

- 워터폴 방법론의 문제로 인해 나타난 방법론으로, IT 기업에서는 당연시되고 있는 프로세스
 - 프로세스 : 스프린트1(디자인→개발→테스트):반복적 개발 주기 → 스프린트 2(디자인→개발→테스트) → ...

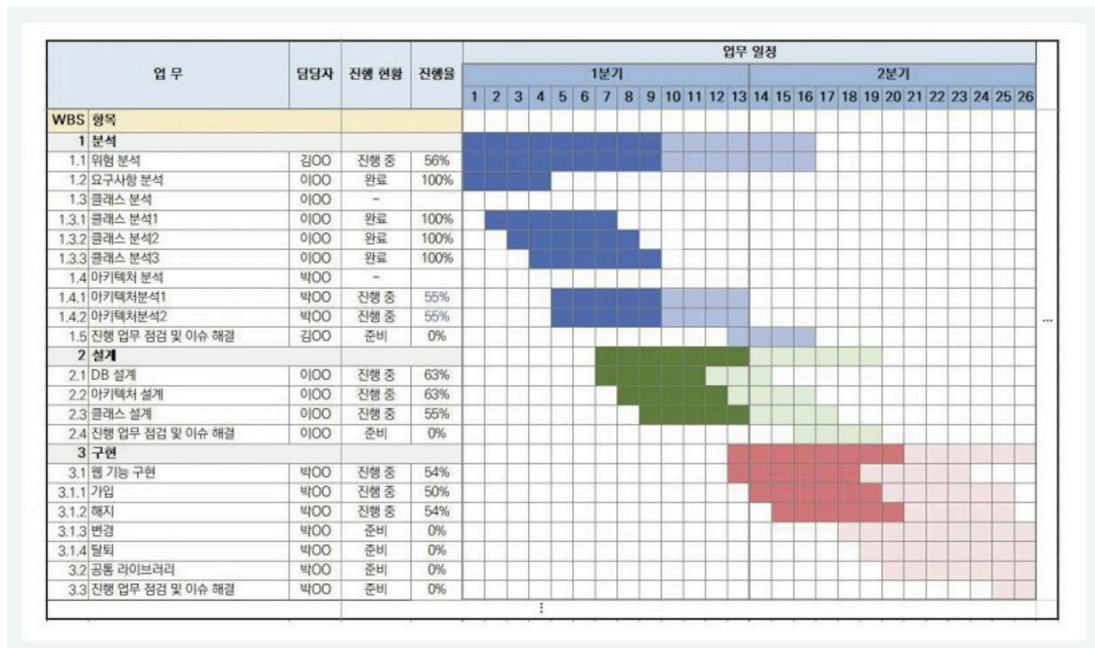
- 장점: 빠른 속도, 시장 상황을 빠르게 반영
 - 단점: 소통과 협업
- 업무 관리
 - ▼ Initiative
 - 여러 개의 Epic으로 구성
 - Epic들이 취합되어 이를 수 있는 넓은 의미의 목표
 - ▼ Epic
 - 여러 개의 Story로 구성
 - 완료하기까지 긴 시간이 필요하거나 몇 번의 스프린트가 요구되는 큰 업무 덩어리
 - ▼ Story

엔드 유저의 관점에서 쓰여진 간단한 요구 사항
 - ▼ Task

Story를 완료하기 위해 작업해야 하는 단위
 - ▼ Sub Task

Task를 완료하기 위해 작업해야 하는 단위
- 일정 관리
 - ▼ WBS
 - 프로젝트 작업을 할 때, 해야 할 업무를 카테고리별로 구분하고 각각의 카테고리는 좀 더 세부적인 작업으로 나누어서, 일정 및 진행사항을 체크하는 기법
 - 구성 요소
 - 프로젝트 목적 및 요구사항 정의
 - 프로젝트 예상 기간
 - 필요 기술 정의
 - 조직별, 개인별 R&R 및 소요 시간
 - 잘 만드는 방법

- 모든 담당자들이 함께 결정하는 것이 좋습니다.
- Scope가 큰 업무부터 결정한 후 세부화하는 것이 좋습니다.



▼ 스크럼

- 마감기한 정해져있음
- 매일 스크럼 미팅 진행 , 짧은 시간, 효율성 위해 서서 미팅하기도 함 : 스탠드업 미팅
- 장점: 기한을 정해 주기적으로 미팅을 하므로 관리가 잘 됨, 긴장감을 높여야 할 때 효율적
- 단점: 돌발 상황이 생기면 스프린트별로 일정을 전부 수정해야 함 (기업에서는 스프린트 중 Hotfix가 발생하면 백로그로 쌓여 모든 스프린트 뒤로 미뤄질 때도 있습니다.)
- 핫픽스(hotfix) 또는 QFE 업데이트(QFE[Quick-Fix Engineering] update) 는 제품 사용 중에 발생하는 버그의 수정이나 취약점 보완, 또는 성능 향상을 위해 긴급히 배포되는 패치 프로그램

▼ 칸반

- 필요한 작업 목록을 우선순위와 종속 단계에 따라 나열하고, 팀이 동시에 해결 할 수 있는 최대 과제 수를 제한하는 방식
- 마감 기한보다, 프로젝트 목적의 정성적 달성을 중시

- 장점: 중요한 업무에 끊임없이 초점을 맞추고 디벨롭할 수 있음, 이미 존재하는 서비스, 완료된 프로젝트에 대한 지속적인 업무를 다루며 프로젝트의 유지, 보수에까지 관여하는 경우 효율적, 애자일 OKR을 이용한다면 고객의 요구사항과 서비스의 비전에 더욱 집중할 수 있음, Hotfix가 발생한 즉시 바로 Cue에 넣을 수 있음
- 단점: 업무 당 마감기한이 없는 경우에 전체 프로젝트가 딜레이될 위험 큼

▼ 번업

- 업무가 얼마나 완료되었는지 추적하기 위해 사용
- 완료된 업무를 기록하는 방식

▼ 번다운

- 완료되어야 할 업무의 양을 추정하는 방식입니다.

• 7주차

▼ 워킹 스켈레톤: Walking Skeleton

- 기본 아키텍처의 PoC(proof of concept : 기술검증) 버전
- 실제로 실행 가능하고 테스트도 병행 가능한 모델
- 최소 실행 가능 제품(MVP)의 기능 우선 순위를 정하는 데 사용

특징

- 사용자 스토리에 초점
- 필수 기능 구현에 중점을 두기 때문에 주요 기능이 완전하게 동작하는 제품의 형태 여야함
- 비즈니스 가치를 보여줘야함

장점

- 적은 소요 시간
- 실제 동작하는 MVP를 목적으로하여 핵심 필수기능에만 초점을 맞출 수 있음
- 사용자로부터 우선순위 결정에 대한 피드백을 신속하게 받을 수 있음

단점

- 중요한 다른 기능 포함하지 않을 수 있음(전체 테스트에 제한)
- 첫 릴리즈까지 시간이 걸림

사용하는 경우

- 최소 실행 간으한 제품 출시에 성과를 기대할 수 있기 때문에 많은 추가 기능이나 부가적 비즈니스 가치를 지닌 상품을 릴리즈 할 때는 사용을 자제해야함

▼ 라이스: RICE

- Reach, Impact, Confidence, Effort

특징

- Reach : 도달 범위, 특정 기간 동안에 이 기능을 사용할 수 있는 사용자 수 반영
- Impact : 기능을 사용함으로써 어떤 영향을 받을지 측정, 0.25~3점으로 평가
- Confidence : 신뢰도 값, 이 기능이 사용자에게 얼마나 혜택을 주는지에 대한 추정 값, 주로 % 척도 사용
- Effort : 제품, 디자인 및 엔지니어링 팀이 소요한 시간

$$\frac{\text{Reach} \times \text{Impact} \times \text{Confidence}}{\text{Effort}} = \text{RICE SCORE}$$

장점

- 큰 그림을 볼 수 있게 해줌
- 의미있는 평가지표
- 사용자 만족도에 기여할 수 있음

단점

- 계산하는데에 많은 시간 필요
- 모든 데이터가 없는 경우 릴리즈를 연기하게 되는 결정을 만들 수도 있음

- 책임에 대해 명확하지 않음

| 사용하는 경우

- 여러 측면에서 제품을 종합적으로 살펴볼 수 있는 효율적인 방법이지만 MVP를 만드는 경우에는 어울리지 않음

▼ 카노: Kano

- 고객기반의 우선순위 지정 방법
- 프로덕트/서비스의 기능에 대해 사용자들의 만족도와 행동이 다르다는 사실에서 시작

| 특징

- must-be : 고객이 이러한 기능이 포함된 경우에만 제품의 의미를 두는 반드시 구현해야하는 기능(=MoSCoW의 Must Have)
- performance : 반드시 필요한 것은 아니지만 고객이 매우 가지고 싶어하는 경우, 고객의 요구 기대치를 예측하는 것과 밀접하게 관련
- attractive : 만족감을 더해주는 기능 (=MoSCoW의 Could Have)
- indifferent: 고객이 관심 없는 기능

| 장점

- 제품의 잠재적인 장점과 단점을 강조 → 시장 적합성 볼 수 있음
- 고객에 가치 기준으로 순위를 정하게 되므로 사용자의 요구에 맞게 조정할 수 있도록 도움

| 단점

- 특정 기능에 필요한 시간과 비용 고려하지 않음
- 결과 처리 및 추정에 노력과 시간 필요 → 릴리즈 시기 늦추고 프로덕트 팀 개발 집중도 떨어트림
- 고객의 의견과 지식에 의해 제한될 수 있음

| 사용하는 경우

- 초기 UX 에 대해 피드백이 필요하다면 효율적이지만 일반적으로 고객은 제품의 기술적 복잡성이나 다양한 종류의 어려움을 잘 모르기에, 이 변수를 최종 우선순위지정을 할 때 반영해야함

• 8주차

▼ 프로젝트 로드맵

로드맵이란 '어떤 일을 추진하기 위해 필요한 목표, 기준 등을 담아 만든 종합적인 계획. '종합적인 계획' 요소: '시간' '단계' '업무' 가 포함되어야 프로젝트를 매니징함에 있어 지장이 없이 프로젝트를 완수할 수 있음

▼ 프로젝트 백로그

제품 백로그는 **제품 로드맵 및 그 요구 사항에서 파생된 개발 팀에서 수행할 우선 순위가 지정된 작업 목록**

▼ 마일스톤

마일스톤은 **프로젝트의 주요 목표 지점**을 의미

단순히 일정상의 목표 날짜를 넘어, 프로젝트 진행 과정에서 중요한 성과를 달성하거나 특정 단계를 완료하는 지점을 의미하는데, 목표 지점까지의 여정 사이의 '경유지'를 생각해주시면 이해가 쉬울거라고 예상

▼ 칸반 보드

'프로젝트 진행 상황을 한 눈에 파악하는 시각적 도구'

작업의 진행 상황을 열(column)과 카드(card)를 사용하여 시각적으로 표현하는 프로젝트 관리 도구로, 각 열은 작업의 상태를 나타내며, 카드는 각 작업을 나타냄.

→. 칸반보드를 사용하면 프로젝트 진행 상황을 한눈에 파악하고, 작업의 흐름을 효율적으로 관리할 수 있음

• 9주차

▼ 역기획이란?

- 역기획은 이미 만들어진 프로젝트를 뜯어보며 " 왜 이걸 이렇게 만들었을까? " 라는 질문을 끊임없이 던지는 과정
- 시장에서 살아남은 서비스 분석, 성공 이유 알아내기 위함

- 기업의 전략과 사용자 니즈를 파악하고 난 뒤엔 개선안을 제시하는 것까지가 역기획

▼ 역기획의 5단계

1단계 : 기존 서비스 이해

- 서비스 특성 및 기능 분석 : 서비스 주요 기능, 사용자 그룹, 특징 파악
 - 사용자 피드백 분석 : 피드백 분석하여 서비스 강,약점 도출
 - 기존 서비스 평가하여 현재 상황 이해
 - 경쟁 환경 파악: 유사 서비스 동향 파악하여 시장에서 위치 확인 및 경쟁 우위 판단
- 종합적으로 분석하여 개선의 필요성 도출

2단계 : 기업 목표 전략, 특징 분석

- 기업 이해 : 기업의 비전, 목표, 전략, 핵심 가치 이해
 - 기업 문서 및 자료 분석 : 공식 문서, 보도 자료, 투자 발표 등 분석
 - 전략적 분석 :기업의 경영 전략과 특징 고려한 현재 상태 평가 및 개선의 필요성 도출
 - 전략 매핑 : 서비스의 기능과 특징을 기업의 전략과 매핑해 일치 정도 분석

3단계 : 데이터 가설 설정 및 검증

- 가설 수립: 데이터를 기반으로 가설을 수립하고, 가설이 검증 가능하도록 정의
- 데이터 수집 및 분석: 데이터의 통계적 또는 비교적 분석을 통해 가설을 검증 후 새로운 가설 정의

4단계 : 동일 전략에 대한 문제해결 방식 비교

- 경쟁사가 동일한 전략을 채택한 경우, 해당 전략의 문제점을 파악하고 문제해결 방식 비교
 - 사전 대비: 동일한 전략을 채택한 경우 발생 가능한 문제에 대비하여 사전에 대응 방안 마련

- 경쟁사 분석: 경쟁사가 동일한 전략을 채택한 사례를 조사하고, 해당 전략의 성공 및 문제점 파악
- 대응 전략 마련: 동일 전략의 문제점을 사전에 파악하여, 해당 문제에 대응할 수 있는 전략 마련

5단계 : 개선안 제시

- 기존 서비스 이해, 기업 전략 분석, 데이터 가설 검증, 경쟁사 비교를 종합하여 개선안을 도출합니다.
 - 사용자 중심의 개선: 사용자의 니즈와 피드백을 반영하여 실질적이고 실행 가능한 개선안 제시
 - 서비스 개선 포인트 도출: 분석 결과를 종합하여 개선이 필요한 부분 도출
 - 사용자 피드백 반영: 사용자의 피드백을 고려하여 개선안을 사용자 중심으로 설계
 - 실행 가능성 평가: 필요한 자원과 기술 등 고려하여 평가

▼ 기존 서비스의 이해와 분석

- 서비스 비전과 목표
 - 가치와 목표를 명확하게 정의
 - 해결하고자 하는 문제 및 혜택 명시하여 사용자에게 어떤 가치 제공할 것인지 이해
 - 회사의 비전과 목표 - 서비스 비전과 목표, 일치시키는 데 중점
- 서비스 수익 구조
 - 수익 모델 이해, 수익 창출하는 방식과 과정 분석
 - 수익 모델이 현재 비즈니스 전략과 연결된 것 파악
 - 서비스 수익성 높이기 위한 잠재 전략과 개선 방안 고려
- 사용자 프로필 및 행동 분석
 - 서비스 이용 사용자 특성 파악하고 사용자 프로필 작성
 - 사용자 행동 패턴, 선호도, 요구사항 조사해 그룹화
 - 사용자 서비스 어떻게 사용하는지 분석하여 UX 향상시키는 데 도움줄 수 있는 인사이트 도출

- 경쟁사 분석 및 시장 조사
 - 경쟁사 서비스 분석해 경쟁 우위와 약점 파악
 - 시장 동행과 경쟁 환경 조사해 시장 크기와 성장 가능성 파악
 - 경쟁사와 비교해 경쟁력 평가 및 개선 방안 모색