* 데이터 기반 의사결정(Decision Science)
* 두 가지 형태: Data-driven / Data-informed

Data-driven은 데이터가 가리키는 대로 최적화, Data-informed는 과거 데이터를 참고하여 혁신

* 데이터로부터 통찰 얻기

1) 중요 지표를 데이터 기반으로 정의하고 시각화

2) 가설을 바탕으로 실제 데이터를 보고 확인: A/B Test

* 데이터 분석 사례

1) 중요 지표를 대시보드로 표시하기

2) 코호트(통계상으로 비슷한 인자를 보유한 집단) 분석으로 고객 이탈/잔존률 분석하기

3) 여러 마케팅 채널의 기여도 분석하기

* 데이터 분석가(Data Analyst)의 역할

1) Business Intelligence를 책임짐

중요 지표를 정의하고 시각화(Tableau, Looker, Superset)

비즈니스 도메인에 대한 깊은 지식이 필요

2) 회사 내 다른 팀의 데이터 관련 질문에 대답함

임직원들의 데이터 기반 결정 조력

반복적인 질문에 대한 대답 자동화, Self-Service화가 관건 – 대시보드, 교육

* 데이터 분석가(Data Analyst)의 Skillset

SQL, Data Modelling, ELT(dbt), Statistical Knowledge, A/B Test 지식과 경험,

지표 정의 능력, 시각화(대시보드) 역량, 비즈니스 도메인 지식

* 데이터 분석가(Data Analyst)의 Dilemma

많은 긴급한 데이터 관련 질문에 시달리고, 데이터 팀이 아닌 다른 현업팀에 소속되어 소속감이 불분명, 고과 기준 불명확해지는 경험을 겪을 수 있음

* 데이터 조직 구조와 그 중요성
* 중앙 집중형 구조

모든 데이터 팀원들이 하나의 팀에 소속

업무 우선 순위는 중앙 데이터 팀이 최종 결정

장점: 데이터 팀원들 간의 지식, 경험 공유가 쉬워지고 커리어 경로가 명확함

단점: 마케팅, 세일즈, CS 등 현업부서들의 필요가 반영되기 힘들고 응답 속도가 떨어져 불만족하게 됨

* 분산형 구조

데이터 팀원들이 각 현업 부서에 분산 소속

업무 우선 순위는 각 팀별로 결정

장점: 현업 부서들의 필요가 반영되기 쉽고 빠른 응답 속도로 만족함

단점: 데이터 업무 인원들 간 지식/경험 공유가 힘들고 데이터 인프라 비용 상승, 부서 간 데이터 공유가 어려워져 부서 간 업무 중복 발생 등의 문제가 생김

* 하이브리드형 구조

가장 이상적인 조직 구조로, 데이터 팀원들 중 일부는 중앙 데이터 조직에서 인프라 관리 업무를 수행하고 나머지는 현업 부서에서 데이터 분석 업무를 수행

데이터 팀원들 간 지식과 경험 공유가 쉬워지고 커리어 경로를 명확하게 유지할 수 있으면서도 현업 부서들의 데이터 분석 수요를 감당할 수 있음.

* 회사의 크기에 따라 데이터 조직의 형태가 달라짐

규모가 너무 큰 경우 전사적 Data Warehouse의 구성이 불가능.

조직별로 별도의 데이터 시스템을 갖추고 필요에 따라 통합

Data Mesh[[1]](#footnote-1): 각 조직이 가지고 있는 데이터를 카탈로그화해 공유하기 용이하도록 함

* 데이터 조직 구조 트렌드
* Cloud의 발전으로 분산형 구조에서 Data Mesh를 형성할 수 있음
* 데이터 조직 일과
* Agile Software Development

빠른 변화로 SW 요구사항 미리 알 수 없는 환경: 기존의 Waterfall 모델 사용할 수 없음.

1-3주 길이의 짧은 사이클(Sprint)로 기능을 하나씩 구현해나가는 방법.

Planning Meeting – Sprint 동안의 일정, 과제 계획

Daily Standup Meeting – 매일 짧게 만나서 경과보고

Demo Meeting – Sprint의 끝에서 성과 공유 및 토론

Agile/Scrum Board: JIRA, Swit, Clickup 등의 툴을 사용해 일정 공유

* 월요일: 지난 Sprint 리뷰 및 새로운 Sprint 계획

1) 지난 주 성과 시연 및 회고, 이번 Sprint 동안 사용할 수 있는 인력 점검

2) 일일 근무 시간산정(근무 시간의 30%는 유지 보수에 이용)

3) **On-call Engineer와 Analyst 지정: ETL 이슈 및 질문에 대한 답변과 반응 담당**

* 화요일: Daily Standup + 다양한 Meeting

1) 매일 5분 정도 모여서 이야기하는 Daily Standup

2) 데이터 팀 내부의 Meeting: Engineer – Analyst, Analyst – Scientist, Scientist – Engineer

3) 외부 현업 부서와의 Sync-up Meeting

* 수~목요일: Daily Standup + 중요 지표 리뷰, ML 모델 개발 리뷰

1) 대시보드를 보면서 중요 지표상의 변화를 살펴보기

2) 개발 중인 Machine Learning 모델에 관련하여 리뷰 수행하기. 특히 A/B 테스트 중인 모델에 대해서 지금까지의 성능을 리뷰하고 성공여부 결정하기

* 금요일: Daily Standup + Weekly Staff Meeting

1) 중요 지표와 전사적/부서별 목표 리뷰

ETL 성공/실패 비율, ML 모델 개발상황 및 테스트 진행상황 리뷰

2) 팀내 중요 인력 상황 점검하고 구인 진행상황 리뷰

3) 주간 사건사고 리뷰

4) 메인 프로젝트 리뷰: 내/외부 프로젝트

5) 팀 전체 상황 및 개인별 상황 업데이트

* 좋은 지표란?
* 지표(Metrics)

팀별/개인별로 정한 중요한 성과 목표. 데이터 문해력의 시발점

* KPI (Key Performance Indicators)

조직 내에서 달성하고자 하는 중요한 지표로, 정량적인 숫자가 선호됨

명확하게 정의되어야 하므로 지표 사전이 필요함

KPI의 수는 적을수록 좋음

잘 정의된 KPI를 통해 현재 상황을 알고 더 나은 계획을 수립할 수 있음

정량적인 지표이기에 **대시보드**를 통해 시간에 따른 성과 변화를 추적할 수 있음

* KPI의 기준

**실제 가치**를 정확하게 전달, 나타낼 수 있어야 함

**반복적으로 발생하는 가치**를 보여줄 수 있어야 함

예측하는 지표가 아니라 **최종적인 결과를 보여주는 지표**여야 함(후행지표)

유용한 피드백 메커니즘이 있어야 함: 의사결정에 사용할 수 있음

* 좋은 지표의 조건: Accessible, Actionable, Auditable (3A)
* KPI와 선행/후행 지표의 예시
* Controllable Input Metrics vs Output Metrics

예를 들어, 체중 감량이라는 Output을 내기 위해서는 식단이나 운동 등의 Input이 있어야 한다. 이처럼, 단순히 Output Metrics을 측정하는 것을 넘어 이에 영향을 줄 수 있는 Input Metrics에 집중하는 것이 중요하다.

* Input Metrics: Leading Indicator

Output Metrics를 움직이는 지표. 직접 통제 가능(Controllable)

예시) 제품 다양성, 가격, 편의성, 새로운 콘텐츠 등

* Output Metrics: Lagging Indicator

Input Metrics의 결과로, 통제 불가능(Uncontrollable)

예시) 판매량, 계약건수, 매출, 이익, 월간 활성 사용자 지표(MAU) 등

* KPI와 선행 지표의 예시

매출액이라는 KPI: 기존 고객 매출(recurring) vs 새로운 고객 매출(new)

매출 = 가격(P) x 판매량(Q)

P가 고정된 상황에서는 Q를 늘릴 방법을 찾아야 함.

그렇다면, Q에 영향을 주는 선행 지표는?

* 두 가지 중요한 KPI

일반적으로 매출이 훨씬 더 중요한 지표

(기존 고객 재구매 건수 / 신규 고객 구매 건수)

하지만 네트워크 현상이 중요한 비즈니스 도메인에서 매출액뿐만 아니라 서비스 사용자 수도 중요

서비스 사용 중인(Active) 고객의 정의가 중요: 유료? 무료? 공급자? 소비자?

* 시각화 대시보드 툴 소개
* Dashboard 혹은 BI Tool로 부르기도 함

KPI와 다른 지표, 중요한 데이터 포인트들을 데이터를 기반으로 보여주는 도구

결정권자들이 데이터 기반 의사 결정을 가능하게 함

현업 데이터 분야 종사자들이 데이터 분석을 용이하게 함

* Microsoft Excel, Google Spreadsheet: 사실상 가장 많이 쓰이는 시각화 도구
* Python: EDA(데이터 특성 분석)에 적합
* Google Looker, Salesforce Tableau, Microsoft Power BI, Apache Superset 등
* Google Looker

내부 고객뿐만 아니라 외부 고객을 위한 Dashboard 작성가능

굉장히 다양한 기능 제공(장점), 고가의 라이선스 정책(단점)

* Salesforce Tableau

학습에 시간이 많이 필요하지만 강력한 Dashboard 작성가능

다양한 제품군을 보유하고 있으며 무료 버전이 있다(장점)

* Looker와 Tableau 두 개의 도구가 많이 사용되는 추세.
* 중요한 포인트는 어떤 서비스를 사용하든 Self-Service Dashboard를 만드는 것.

1) 매번 사람의 노동으로 수정해야 하는 번거로움을 피하기 위해

2) 현업 인력들의 접근성을 높이기 위해: 데이터 민주화, 데이터 탈중앙화, 데이터 거버넌스

1. 2019년, Zhamak Dehghani가 처음 제안한 중앙 관리와 표준을 염두에 둔 데이터 분산 아키텍처로, 현재는 기술이라기보다는 조직 문화, 개념에 가깝다. <https://aws.amazon.com/ko/what-is/data-mesh/> (Data Mesh란 무엇인가요?) [↑](#footnote-ref-1)