

프로젝트와 산출물(폭포수)

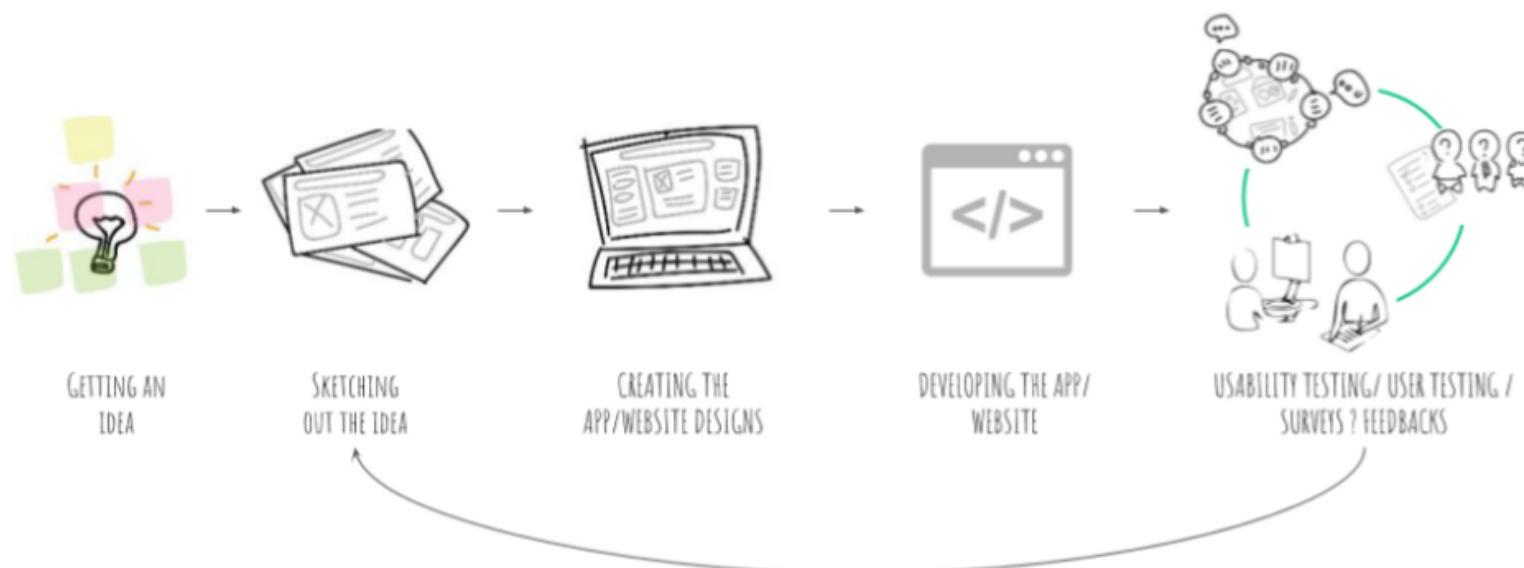
프로젝트 진행과 서비스 기획자

01

프로젝트 방법론의 이해

- 폭포수 방법론은 1950년대에 처음 언급된 소프트웨어 개발의 표준 방법

Waterfall development model





02

폭포수 방법론의 장단점

장점

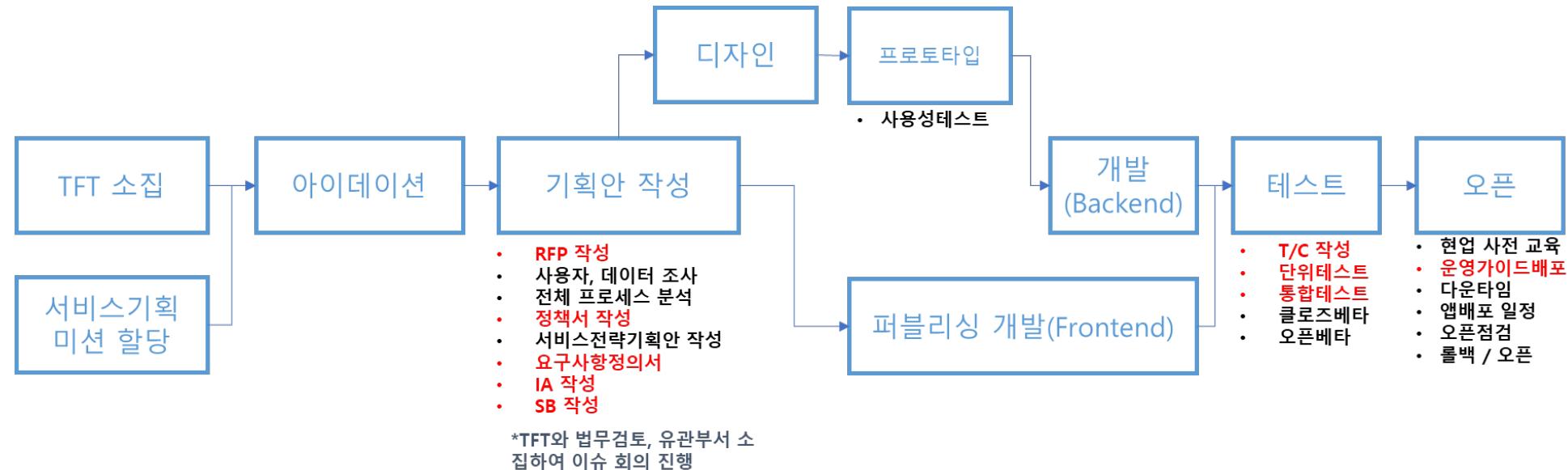
- 단계가 명확히 구분되어 있어서 정확한 진척률 확인이 용이하다
- 단계별로 책임 소재가 명확하다
- 프로젝트 인력에 산출물 기준이 명확하여 Push를 잘 하는 조직에서는 오히려 오픈 일정을 맞추기에 적합할 수 있음
- 산출물 문서가 명확하여 관리에 용이하다
- 외주 인력과 작업하기 용이하다

단점

- 앞의 단계에서 잘못된 기획이나 설계가 되었을 때 되돌리는 것에 굉장히 오랜 시간이 걸린다.
- 앞단계의 문서에 명시되지 않은 부분에 대해 누락되기 쉽고, 기획 변경 시 영향 범위가 넓다
- 테스트기간 전까지 산출물의 형태를 미리 보거나 테스트하기 어렵다

03

폭포수 방법론에서 기획자의 산출물





04

폭포수 방법론에 대한 오해와 하이브리드적 적용

폭포수 방법론을 배우는 것은 촌스럽다?

→ 애자일 방법론은 긴 시간 폭포수 방법론을 사용하며 문제를 개선하기 위해 등장한 방법론이기 때문에, 폭포수 방법론의 각 프로젝트 진행단계의 업무를 정확히 알지 못하는 상태에서 애자일 방법론만 학습하면 프로젝트 자체를 이해하기 어려워진다.

폭포수 방법론은 느리다?

→ 프로덕트를 출시하는 기간과 동일한 완성도를 요구하는 것은 다른 문제다.
실무자들은 단기간에 동일한 산출물을 만들려면 애자일방법론보다 폭포수 방법론이 더 빠를 수도 있다고 지적한다.
특히, 기존 시스템에 파악이 되지 않은 기획자와 개발자가 참여하는 프로젝트의 경우, 폭포수 방법론의 산출물 방식을 이용하여 명확하게 단계를 정리하고 기준을 마련해 주되 일부 개발 과정에서 애자일 방법론과 같은 회의와 공유를 중요시 하는 하이브리드 방법론을 사용할 수도 있다.

전통적인 프로젝트 방식인
폭포수 방법론도 여전히 많이 사용된다

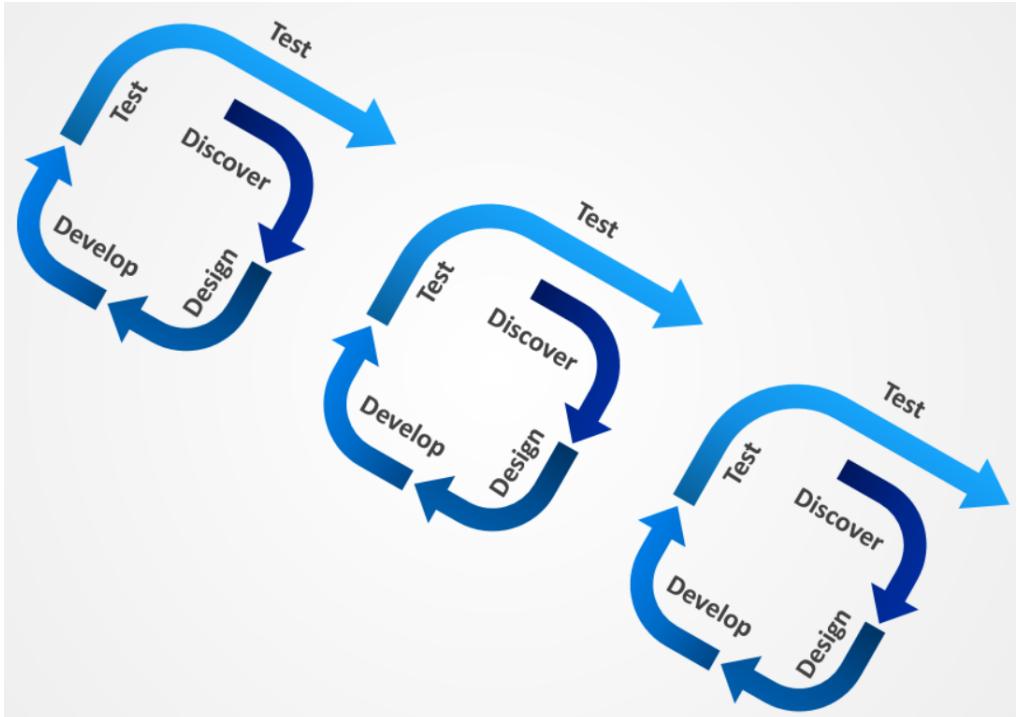
이번에는, 애자일 방법론에 대해 알아보자

프로젝트와 산출물(애자일)

프로젝트 진행과 서비스 기획자

01

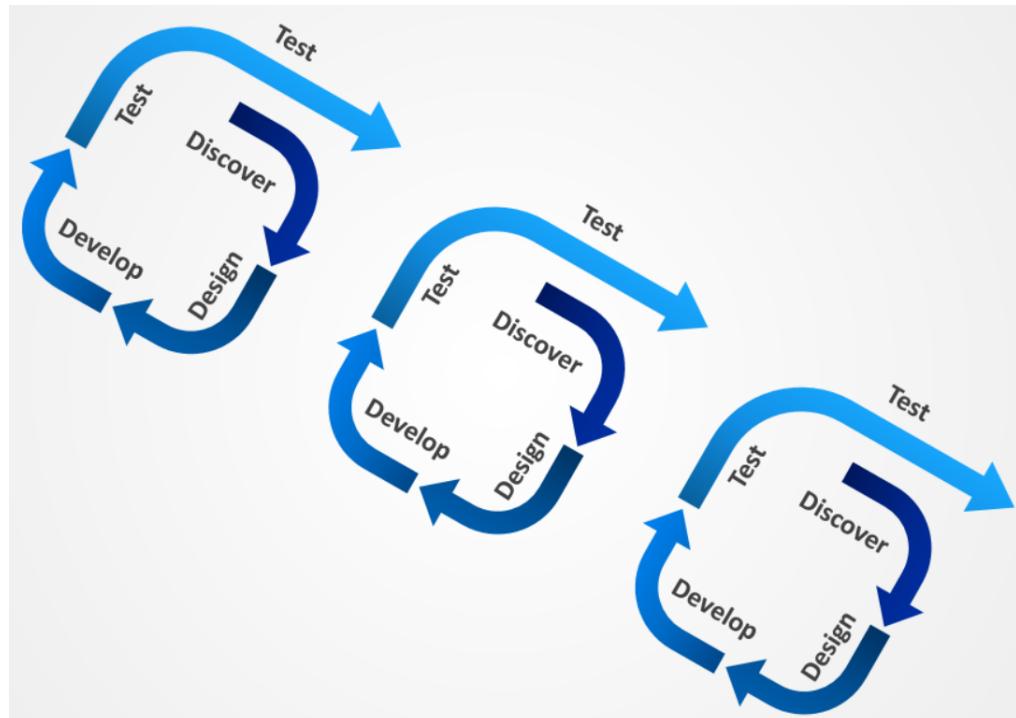
프로젝트 방법론(애자일)의 이해



- 2001년 켄트 벡을 중심으로한 개발자 17명이 애자일 방법론을 선언하여 촉발
- 기존 폭포수 방법론의 선형적인 프로세스가 매우 빠른 IT 시장에서 적합하지 않다고 판단하여 등장한 대안적 방법론
- 사전 문서 준비보다 실제 개발을 작은 스프린트 단위로 잘라서 실제 개발과 테스트를 통해 프로덕트의 기획과 검토를 여러 번에 걸쳐서 하는 방법론

01

프로젝트 방법론(애자일)의 이해



- 애자일의 4가지 핵심 사상

- 1) 절차와 도구를 넘어선 **개성과 화합**
- 2) 종합적인 문서화를 넘어선 동작하는 **소프트웨어**
- 3) 계약과 협상을 넘어선 **고객과의 협력**
- 4) 계획 준수를 넘어선 **변화에의 대응**

단순히 프로젝트 방법론을 넘어선 사상적인 변화!



02

애자일 방법론의 장단점

장점

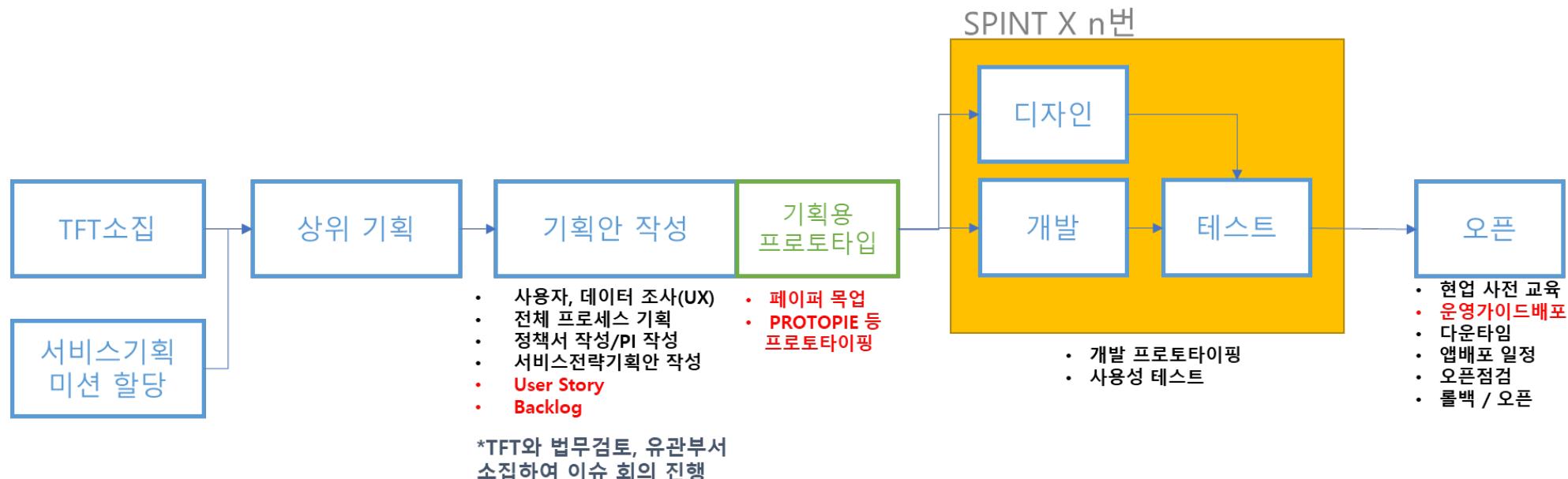
- 문서화 시간을 단축하고 개발된 산출물을 빠르게 만나볼 수 있다
- 중간 단계에서 변화하는 기획을 수용하여 빠르게 변화할 수 있다
- 개발자와 디자이너의 자유도가 높아 창의적인 결과를 가져올 수 있다
- 과정 내내 팀원들과 많은 협의를 통해 최고의 산출물을 만들어낼 수 있으며, 개발자와 디자이너와의 끈끈한 협업이 가능하다.

단점

- 복잡한 프로젝트에 참여자의 수준이 제각각이면, 전체적인 진척률과 수준을 관리가 어렵다
- 참여자들이 애자일 사상에 대한 공감이 없는 상태에서 출발하면 상호간의 R&R 문제로 번지기 쉽다
- 팀과 밀접하게 움직여야 하기 때문에 애자일 전문가 역할이 필요하고 전체적인 사상과 팀워크 관리가 필요하다

03

애자일 방법론에서 기획자의 산출물





04

애자일 방법론에 대한 오해와 하이브리드적 적용

애자일 방법론은 세련된 것이다?

→ 애자일 방법론은 긴 시간 폭포수 방법론을 사용하며 문제를 개선하기 위해 등장한 방법론이기 때문에, 폭포수 방법론의 각 프로젝트 진행단계의 업무를 정확히 알지 못하는 상태에서 애자일 방법론의 도구만 학습하는 것은 의미가 없다. Jira나 Trello를 쓸 줄 아는 것만으로는 애자일 사상을 이해했다고 말할 수 없다.

애자일 방법론은 빠르다?

→ 개발 결과물을 받아보는 시점 자체로는 빠르다고 할 수 있지만, 프로덕트를 만들기 위해서 이러한 스프린트를 여러 번 거쳐야 한다. 스프린트간 검토 과정에서 지나치게 많은 기획 변경이 일어나고, 개발자와 디자이너가 너무나 많은 실험을 반복한다면 애자일 방법론도 느릴 수 있다. 협력과 많은 대화를 통해 이러한 부분에 대한 상호간의 이해가 선행되어야 한다.

애자일 방법론은 팀 내에
사상적인 공감대 형성이 필수다

프로젝트 협업자와의 커뮤니케이션에 대해서 알아보자.

협업자와의 커뮤니케이션 (현업 요청자)

프로젝트 진행과 서비스 기획자



01

요청자, 현업에 대한 이해

현업? (現業)

- 프로덕트 개발 진행 후 서비스를 운영하면서 직접 사용하거나 활용해야하는 업무자를 이르는 업계의 슬랭
- 마케팅담당자, 영업담당자, CS 처리자 등등 프로덕트의 개선을 요청하거나 관계가 있는 모든 관련자
- Admin 서비스의 내부 사용자에 해당

*이 강의는 인하우스 기획 입장에서 진행



02

현업의 요청의 종류와 협의

- **현업 요청의 유형**

- 1) 업무적 목표를 위한 시스템적 의견 요청
- 2) 업무 중 시스템의 불편 사항 접수
- 3) 타사 서비스 캡쳐에 그림까지 있는 형태
- 4) 와이어프레임 수준의 UI와 설명까지 있는 형태

Cf. 외주 기획의 입장이라면 *RFP(request for proposal)*나 요구사항정의서의 형태로 수령



02

현업의 요청의 종류와 협의

웹기획 오퍼레이터 서비스기획자

Needs < Purpose

What < Who, How



02

현업의 요청의 종류와 협의

- **현업과의 협의를 위한 Tip!**

1. 요청된 UI는 수단일 뿐! 근본적인 목적을 파악하여 대안을 제시하자
2. 꼭 개발을 해야만 하는 것이 정답은 아닐 수 있다
목표를 위해 기존 시스템으로 할 수 있는 것을 안내해 줄 수도 있다
3. 요청된 사항이 전체 프로덕트의 비즈니스적 관점이나 고객 관점에서 충돌이 되는지
판단해야 한다. 요청사항에 대해 제로베이스에서 검토해야 한다.

인하우스 환경에서,
현업과 기획자는 갑과 을이 아닌
협업의 관계여야 한다

디자이너와의 커뮤니케이션에 대해서 알아보자.

협업자와의 커뮤니케이션 (디자이너)

프로젝트 진행과 서비스 기획자



01

디자이너에 대한 이해

디자이너의 정의

UI 디자이너? UX 디자이너?
비주얼 디자이너? 프로토타이퍼?

비주얼 디자이너의 핵심 키워드

레이아웃 그리드 여백

자리수 레벨 톤앤매너

포인트 컬러 룩앤파일



02

디자이너 업무의 이해

*비주얼 디자이너의 업무 순서

1) 드래프트

배리한다

2) 기본 시안과 Variation

3) 프로토타이핑

4) 스타일 가이드 작성

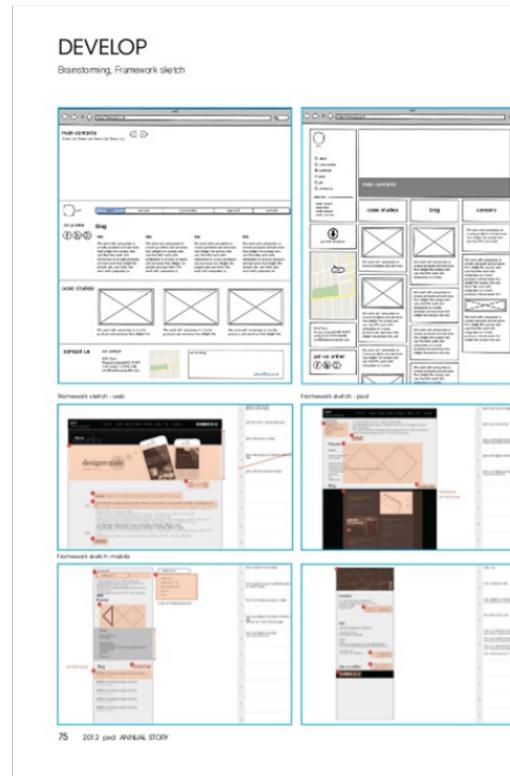


Cf. 무드보드 : 컬러나 분위기를 정하기 위한 이미지 모음

02

디자이너 업무 - 01 드래프트

SB의 기본 설계 요소를 기준으로 전체적인 디자인 컨셉과 방향을 만들어 보는 여러 개의 시안

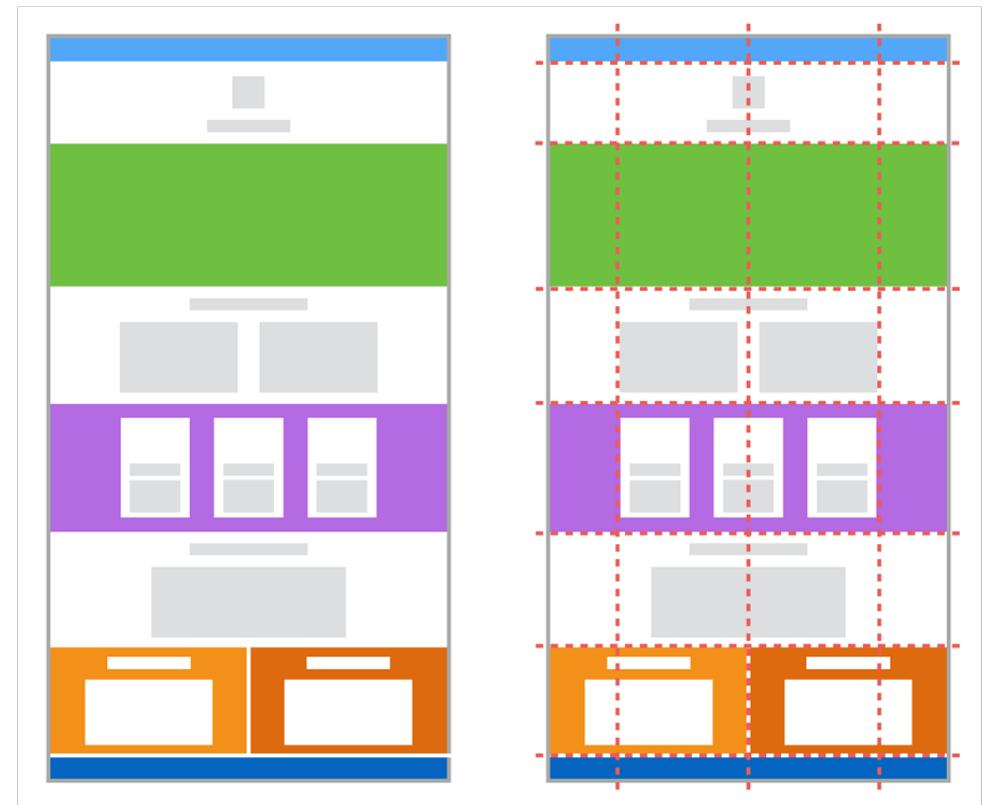


www.shutterstock.com · 572023633

02

디자이너 업무 - 02 기본 시안의 검토

기획자의 검토 : 의도대로 되었는지, 개발이 불가능한 디자인은 아닌지 기획자 검토가 필요
디자인 디렉터의 검토 : 그리드, 레이아웃 검토, UI 요소 -> '운영요소'의 가이드화

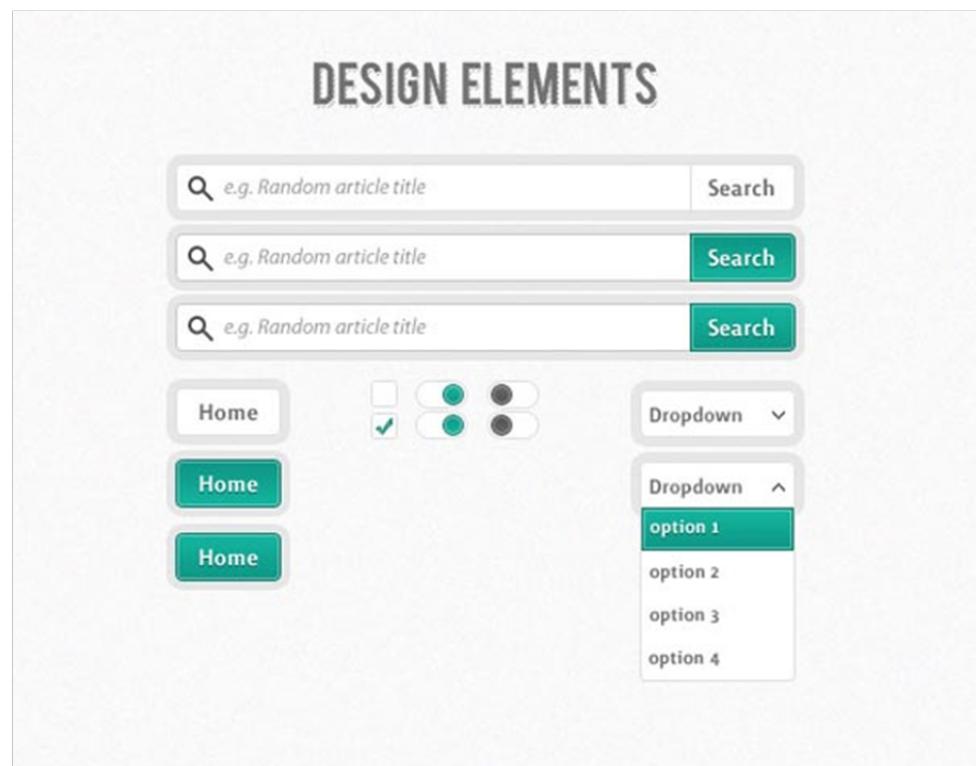




02

디자이너 업무 – 03 시안을 기본으로 한 Variation 작업

시안의 기본 케이스를 중심으로 여러가지 케이스에 필요한 여러가지 요소를 개별 디자인 정리



작게는

버튼 클릭시, 클릭 후

체크 시/ 미체크시

드롭다운박스 닫혔을 때, 열렸을 때

크게는

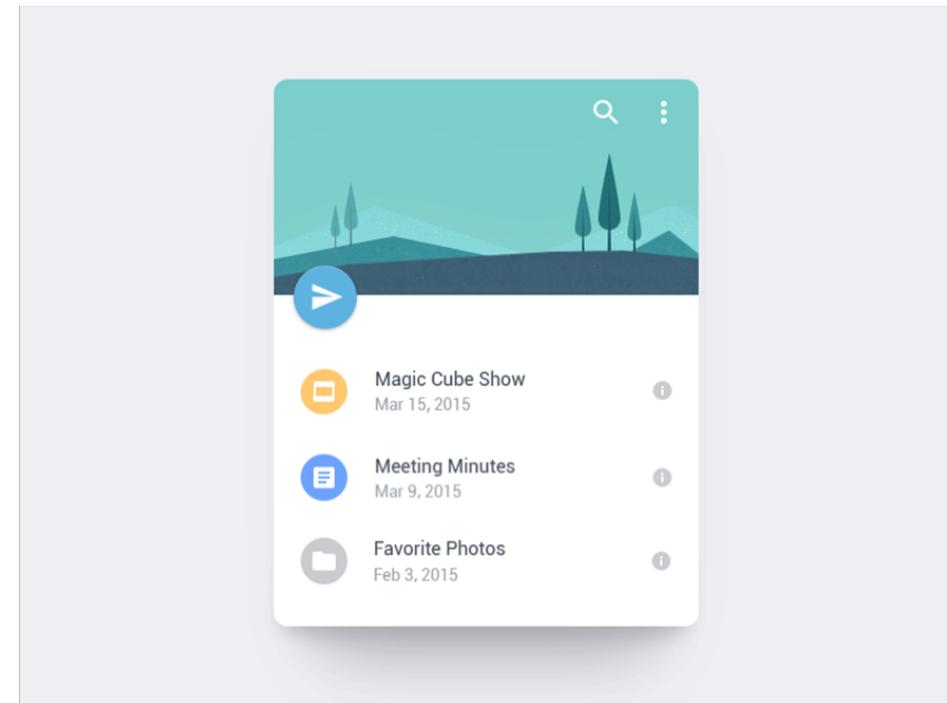
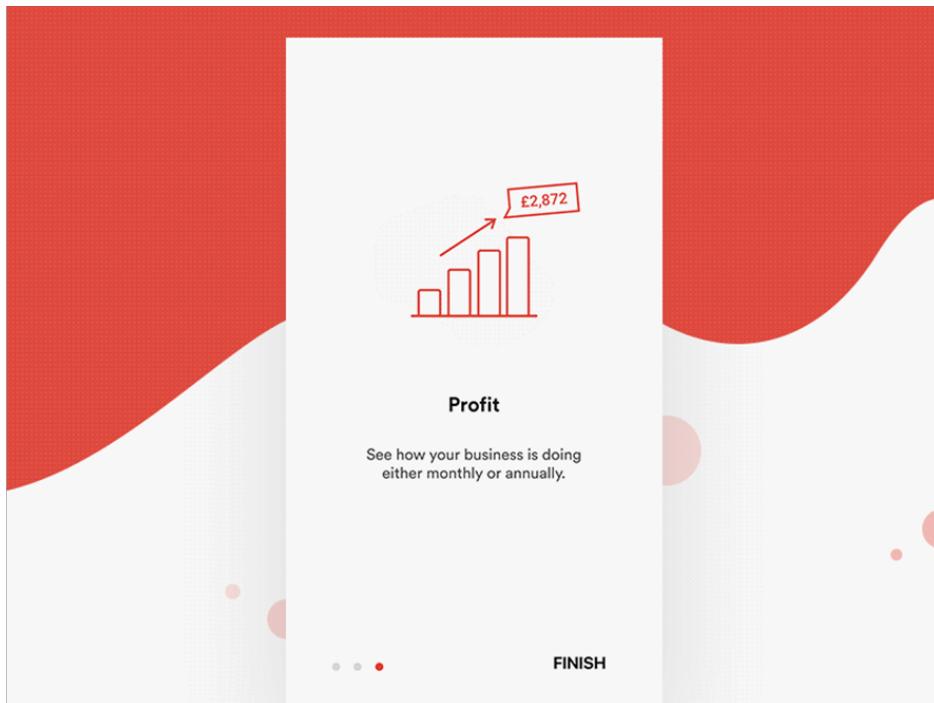
케이스 분기처리에 대한 시안



02

디자이너 업무 - 05 프로토타이핑

원하는 형태의 동작과 인터렉션을 프로토타이핑 툴로 구현한 것,
이를 통해 디테일한 인터렉션에 필요한 디자인 세부사항을 디자인 할 수 있다.

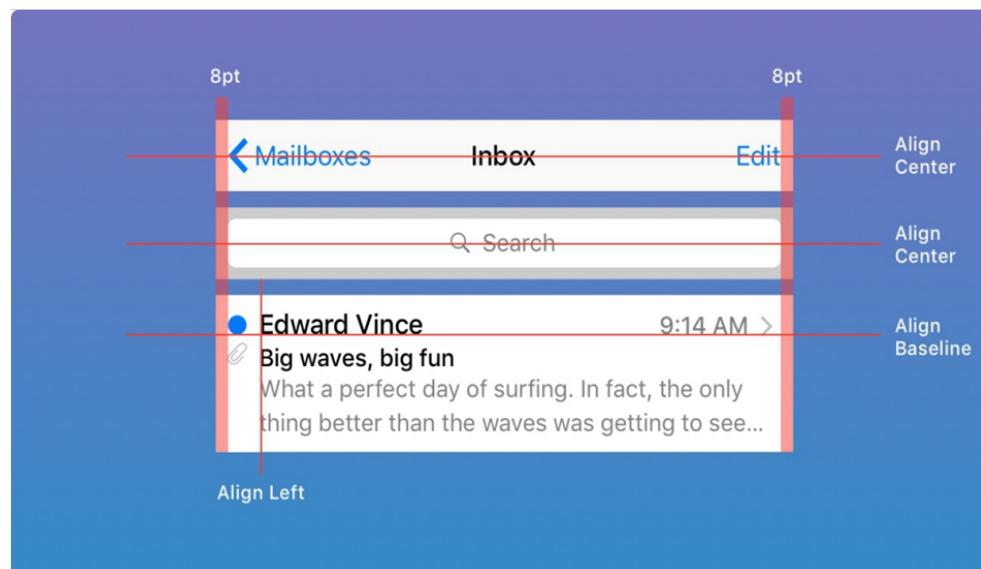




02

디자이너 업무 - 06 웹 스타일 가이드

퍼블리싱을 하기 위해 디자인에서 주는 가이드, 메인 컬러, 서브컬러, 여백, 폰트, 텍스트 레벨,



COLOR PALETTE

| | | |
|--|--------------------|---------|
| ● | CANDY APPLE | #e82525 |
| ● | CRIMSON | #c92820 |
| ● | COOL BLUE | #074f67 |
| ● | DEEP LAKE | #14354b |
| ● | GRAVEL | #565656 |
| ● | WET CEMENT | #778d8e |

TYPOGRAPHY

Header 1
Font: Montserrat Bold / Color #e82525

HEADER 2
Font: Montserrat Bold / Color #c92820

HEADER 3
Font: Montserrat Bold / Color #074f67

HEADER 4
Font: Montserrat Bold / Color #14354b

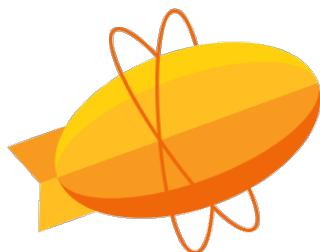
Header 5
Font: Montserrat Bold / Color #565656

This is Body copy. *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent eu semper ligula, nec fermentum odio. Aenean non blandit neque, ac accumsan nibh. Morbi elementum neque id sodales blandit. Morbi eget turpis in urna sodales pharetra. Aenean quis pulvinar lacus, sed lacinia sem.*
Font: Uluru Bookerville / Color #778d8e

This is an inline link
Font: Uluru Bookerville / Color #e82525 / Underline: dotted

03

디자인 업무의 효율



Sketch

- Variation 단축

Zeplin

- 스타일가이드
자동 생성

Invision

- 프로토타이핑에
Sketch 작업파일 연동



04

디자이너와의 커뮤니케이션 Tip

- 디자인은 디자이너가 전문가다
- 1) 비즈니스와 기획적 관점에서 의논하고, 취향으로 접근하지 마라
 - 2) 기획자는 컴퓨터가 아니라, 의논하고 논의하는 협업자다
 - 3) 모바일의 가변적인 환경을 이해하고 시안을 검토하고,
또 다른 협업자들의 입장에 대해서도 충분히 대변해 주어야 한다

디자인은 디자이너가 전문가다

개발자와의 커뮤니케이션에 대해서 알아보자.

협업자와의 커뮤니케이션 (개발자)

프로젝트 진행과 서비스 기획자



01

개발자에 대한 이해

SW 형상 관리 빌드 담당자 : 깃허브, SVN 등의 형상관리를 담당하고 소스 빌드 및 배포하는 담당자

TA(Technical Architecture) : 데이터, 애플리케이션 아키텍처를 지원하는 데 필요한 정보기술 인프라 요소 및 구조, 그리고 이들 간의 관계를 정의하는 아키텍처 담당자

DBA(Database Administrator) : 데이터베이스의 테이블 구조 및 인덱스, 쿼리 검수 등 관리자
AWS의 보급으로 점점 더 Data Science 영역의 업무가 요구됨.

시스템 프로그래머 : 대형 시스템의 내부에서 OS처럼 움직일 시스템을 설계하는 개발자(국내 많이 없음)

보안 프로그래머 : 정보보호와 방화벽, 네트워크 보안 등을 담당하는 프로그래머

웹 프로그래머 : JAVA, C++, python 등의 언어로 웹 어플리케이션을 개발하는 프로그래머
cf) 코더 : 전문적인 지식없이 단순 코딩을 하는 사람을 낮춰 부르는 말

임베디드 프로그래머 : IoT, 스마트폰 등 디바이스에 이식된 소프트웨어를 개발하는 프로그래머

***개발자라고 다 같은 개발자가 아니다!!**



02

개발자와의 커뮤니케이션 : 용어의 문제

- 지금 당장 개발 언어를 배우는 것이 도움이 될까?

기본 함수
배운 기획자

VS

프로그램
짤 수 있는 개발자

- 필요한 것은 커뮤니케이션 : 이슈를 해결하기 대화 속 전문 용어는 결국 회사마다 다르다
- 자사의 용어사전을 만들어서 어휘를 통일 시켜야 한다



02

개발자와의 커뮤니케이션 : 무시의 문제

- 기획자가 시스템을 잘 모르는 것은 당연한 일
- 모르는 것은 터놓고 배우고, 기획을 할 때부터 시스템적인 부분에 대해서는 물어보자
- 기획자는 비즈니스와 UX로 대화해야 한다
- 기획자는 Boss가 아니다

개발은 개발자가 전문가다

프로젝트에서 작성하는 산출물의 작성에 대해서 알아보자.