

Fastcampus

Computer Science SCHOOL

Software Engineering

Beautiful Soup

Web Scraping with BeautifulSoup

```
$ pip install beautifulsoup4
$ pip install beautifulsoup4
Collecting beautifulsoup4
  Downloading beautifulsoup4-4.5.1-py3-none-any.whl (83kB)
    100% |████████████████████████████████| 92kB 153kB/s
Installing collected packages: beautifulsoup4
Successfully installed beautifulsoup4-4.5.1
```

Web Scraping with Beautiful Soup

```
$ pip list
DEPRECATION: The default format will switch to
the [list] section) to disable this warning.
beautifulsoup4 (4.5.1)
pip (9.0.1)
setuptools (20.10.1)
urllib3 (1.19.1)
```

Web Scraping with BeautifulSoup

```
>>> import urllib
>>> from bs4 import BeautifulSoup
>>> html = """
... <html><head><title>The Dormouse's story</title></head>
... <body>
... <p class="title"><b>The Dormouse's story</b></p>
...
... <p class="story">Once upon a time there were three little sisters; and their names were
... <a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
... <a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
... <a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
... and they lived at the bottom of a well.</p>
...
... <p class="story">...</p>
...
"""
>>> soup = BeautifulSoup(html, 'html.parser')
>>> print(soup.prettify())
```

Web Scraping with BeautifulSoup

```
import urllib
from bs4 import BeautifulSoup
html = """  
uglified html code  
"""  
  
soup = BeautifulSoup(html, "html.parser")
print(soup.prettify())
```

Web Scraping with Beautiful Soup

```
curl https://www.rottentomatoes.com
```

```
import urllib.request
from bs4 import BeautifulSoup

url = "https://www.rottentomatoes.com"
html = urllib.request.urlopen(url)
source = html.read()
html.close()

soup = BeautifulSoup(source, "html.parser")
print(soup)

table = soup.find(id="Top-Box-Office")
print(table)
```

Web Scraping with BeautifulSoup

```
all_tr = table.find_all("tr")
for tr in all_tr:
    all_td = tr.find_all("td")
    score = all_td[0].find("span", attrs={"class": "tMeterScore"})
    movie_name = all_td[1].a.text
    amount = all_td[2].a.text
    print(score, movie_name, amount)
```

Web Scraping with BeautifulSoup

```
>>> import urllib.request
>>> from bs4 import BeautifulSoup
>>> url = "https://www.rottentomatoes.com"
>>> html = urllib.request.urlopen(url)
>>> source = html.read()
>>> html.close()
>>> soup = BeautifulSoup(source, "html.parser")
>>> table = soup.find(id="Top-Box-Office")
>>> all_tr = table.find_all("tr")
>>> for tr in all_tr:
...     all_td = tr.find_all("td")
...     score = all_td[0].find("span", attrs={"class": "tMeterScore"}).text
...     movie_name = all_td[1].a.text
...     amount = all_td[2].a.text
...     print(score, movie_name, amount)
```

Web Scraping with Beautiful Soup

...

69% Sing

\$41.5M

95% Fences

\$10.2M

40% Why Him?

\$10.1M

16% Assassin's Creed

\$8.1M

12% Collateral Beauty

\$4.1M

73% Fantastic Beasts and Where to Find Them

\$4.0M

So, Let's Scrap Naver

Advanced Web Crawling

Selenium

Web Crawling with Selenium

```
$ pip install selenium
```

Web Crawling with Selenium

```
import os
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
```

Web Crawling with Selenium

```
ff_driver = webdriver.Firefox()
ff_driver.get("https://www.google.co.kr/")

query = ff_driver.find_element_by_id("lst-ib")
query.send_keys("스타크래프트")

ff_driver.find_element_by_name("btnK").click()
ff_driver.implicitly_wait(10)
```

Web Crawling with Selenium

```
RESULTS_LOCATOR = "//div/h3/a"
WebDriverWait(ff_driver, 10).until(
    EC.visibility_of_element_located((By.XPATH, RESULTS_LOCATOR))
page1_results = ff_driver.find_elements(By.XPATH, RESULTS_LOCATOR)
for item in page1_results:
    print(item.text)
```

Software Engineering

Software Engineering

Definition

Software engineering (SWE) is the application of engineering to the design, development, implementation, testing and maintenance of software in a systematic method.

--> 소프트웨어의 개발, 운용, 유지보수 등의 생명 주기 전반을 체계적이고 서술적이며 정량적으로 다루는 학문

Software Engineering

Why??

Development vs Implementation

- Development
 - The process of analysis, design, coding and testing software.
- Implementation
 - The installation of a computer system or an information system.
 - The use of software on a particular computer system.

Trend of Software Engineering

- Acceleration of DevOps adoption
- Continued wave of everything natively mobile
- Greater demand for increased privacy
- Cloud computing will be a thing of the past
- integration with Web and Mobile App

DevOps

used to refer to a set of practices that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes.

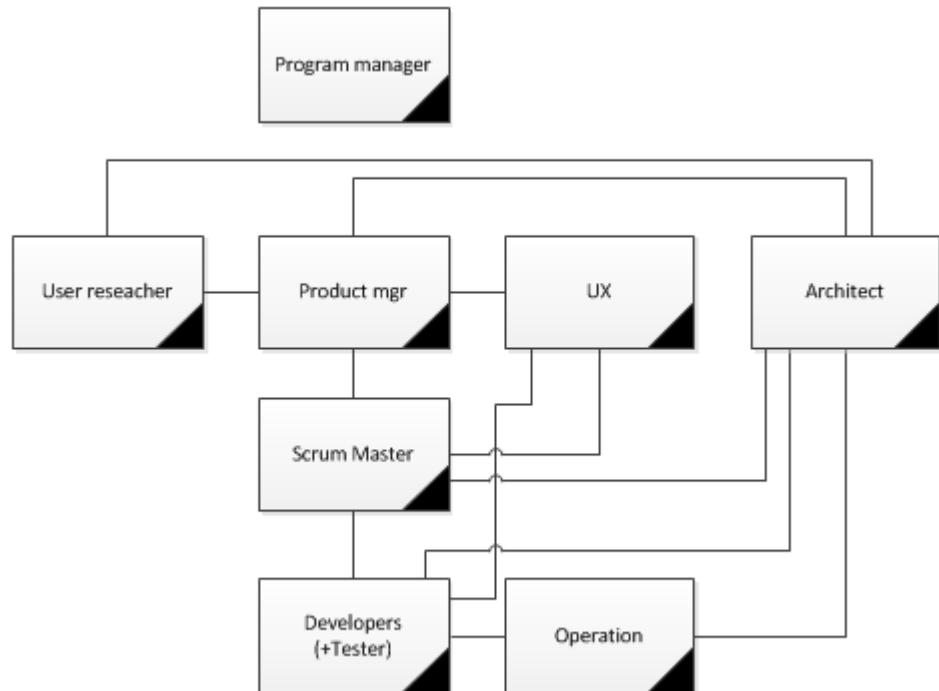
It aims at establishing a culture and environment where building, testing, and releasing software can happen rapidly, frequently, and more reliably.

DevOps

- 기존의 개발과 운영 분리로 인해 발생하는 문제들
문제 발생 -> 비방 -> 욕 -> 상처 -> 원인분석 -> 문제해결
- 좋은 소프트웨어를 위한 필수조건
 - 기획팀과의 원활한 소통으로 요구사항을 충실히 반영
 - 운영팀과의 원활한 소통으로 소비자 불만과 의견을 반영

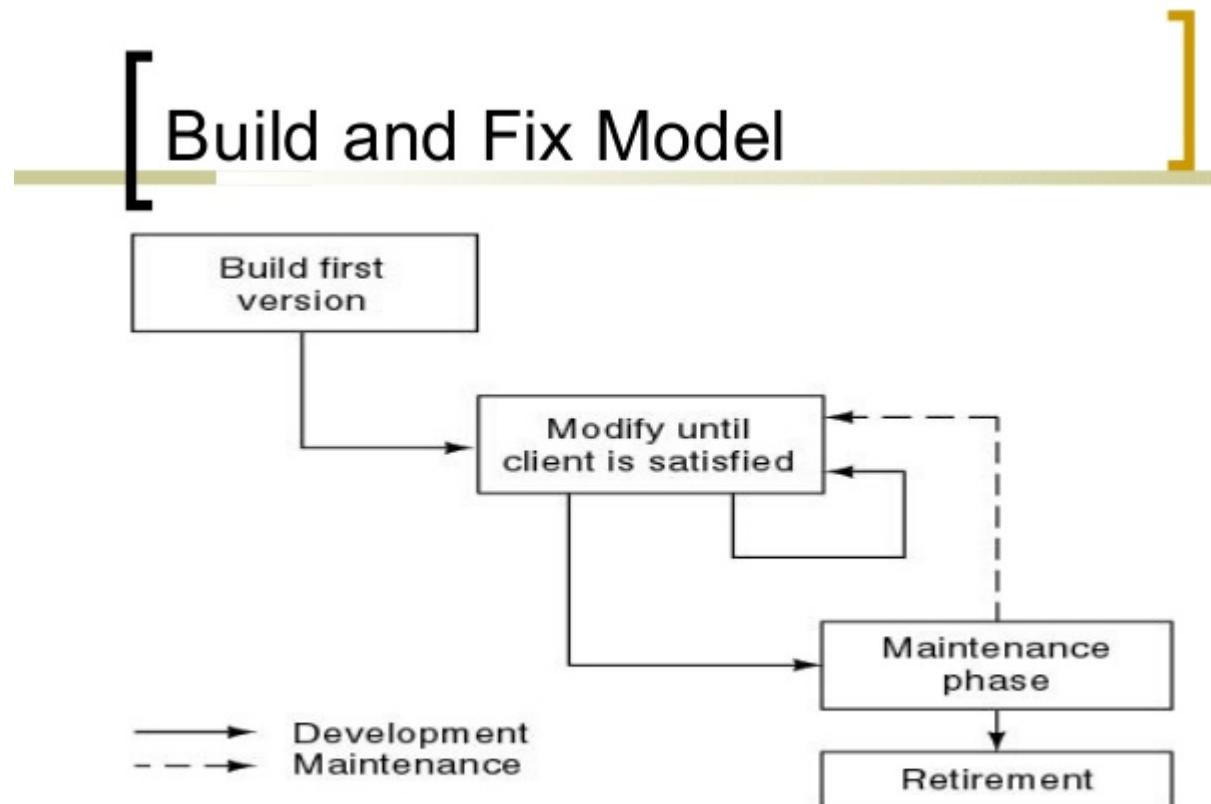
DevOps

운영과 개발을 통합하여 커뮤니케이션 리소스를 줄이고, 개발 실패 확률을 줄임과 동시에 보다 안정적인 서비스를 운영할 수 있음!!



Software Development Life Cycle

Build-fix Model

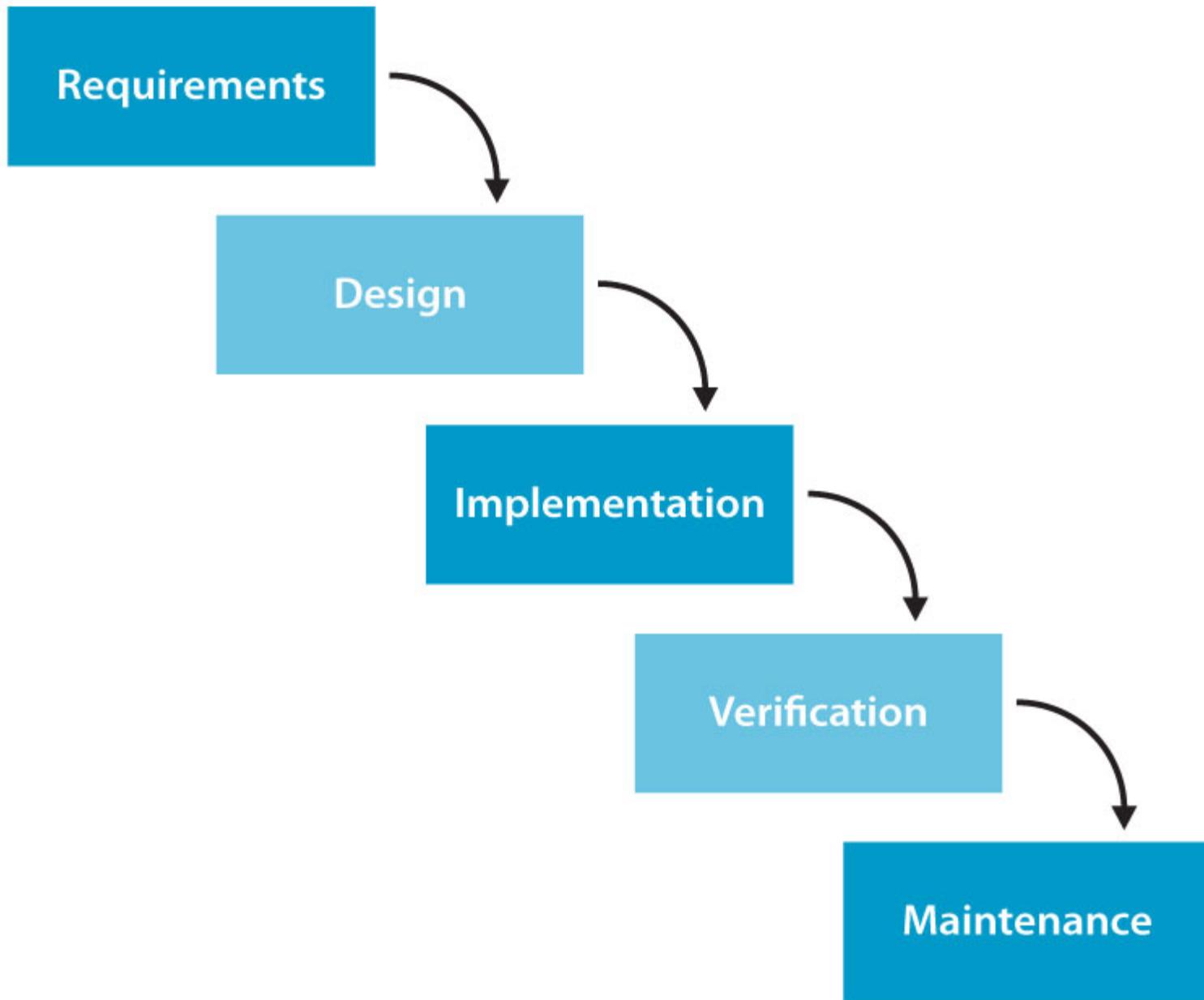


Build-fix Model

설계없이 일단 개발, 만족할 때까지 수정

시작이 빠름

계획이 정확하지 않음, 개발 문서가 없고 진행상황 파악이 힘듦



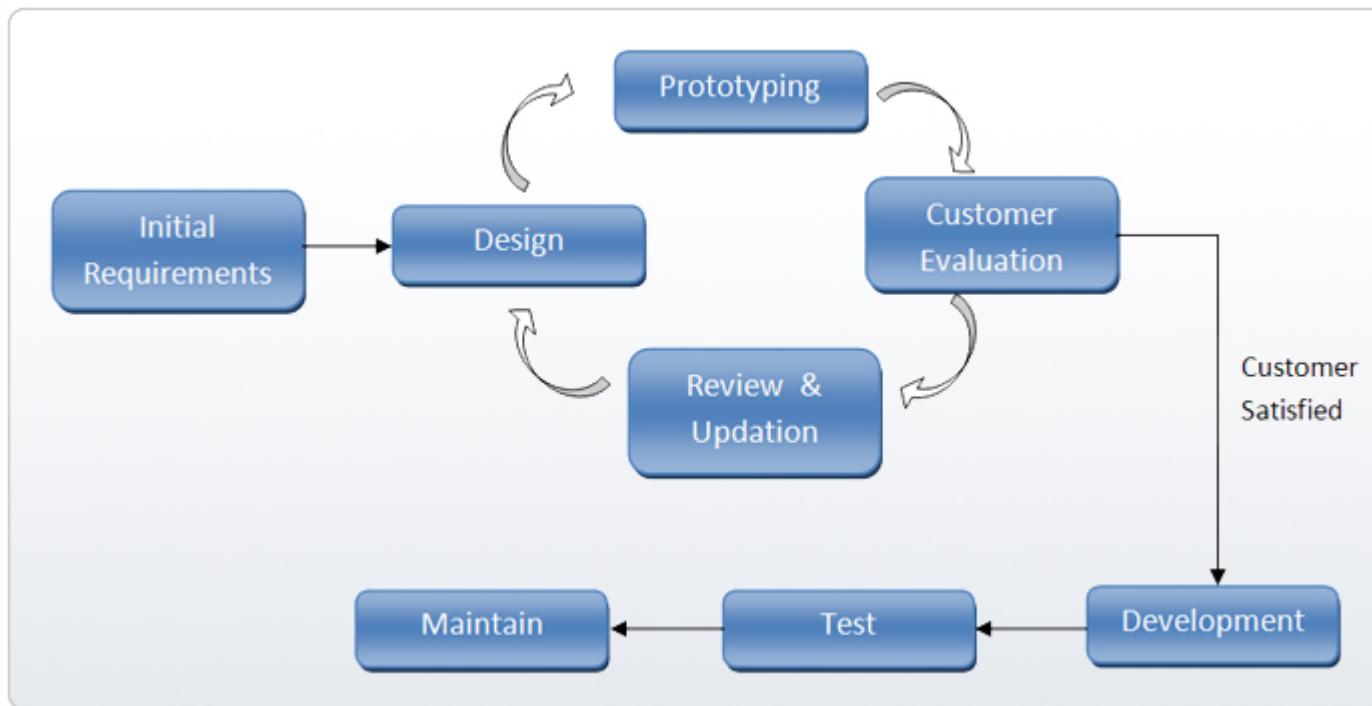
Waterfall Model

순차적인 개발 모델, 가장 많이 사용됨

정형화된 접근 가능, 체계적인 문서화 가능

직전 단계가 완료되어야 진행 가능

Prototype Model



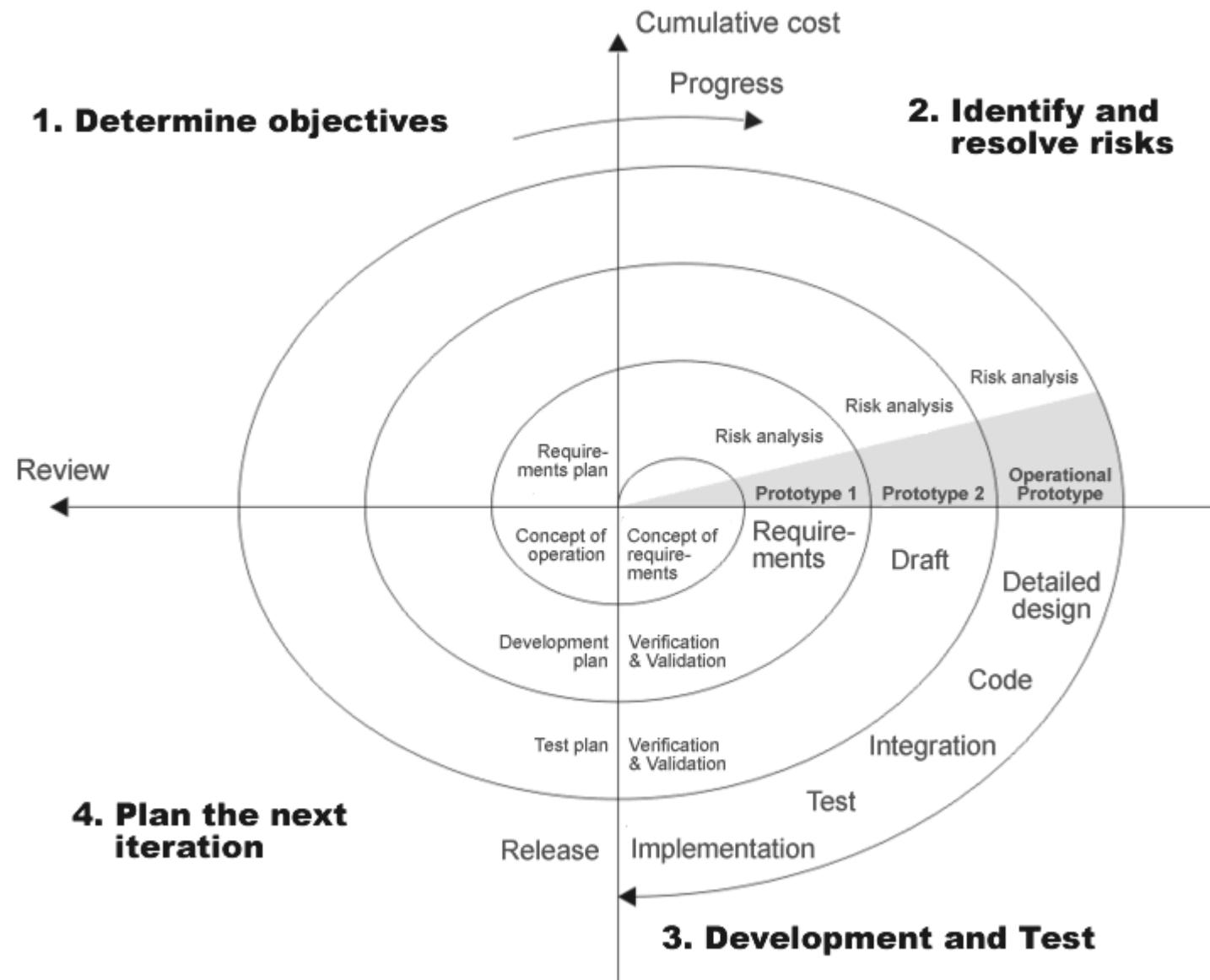
Prototype Model

고객 요구사항을 적극적으로 반영하는 모델

빠른 개발과 고객 피드백을 빠르게 반영할 수 있음

대규모 프로젝트에 적용하기 힘듦

Spiral Model



Spiral Model

대규모 or 고비용 프로젝트

프로젝트의 위험요인을 제거해 나갈 수 있음

각 단계가 명확하지 않음

이외에도..

- RAD(Rapid Application Development) Model
- Iterative Development Model
- V Model
- Component Based Development

Software Development Process

UP(Unified Process)

- 도입(분석위주), 상세(설계위주), 구축(구현위주), 이행(최종 릴리즈)의 반복

XP(eXtreme Process)

- 스크럼 마스터가 주도적으로 프로세스를 주도하며, 고객과 개발자 사이의 소통을 중시함
- Product Owner와 Development Team, Customer로 역할에 충실

TDD

Test Driven Development

- 객체지향적
- 재설계 시간 단축
- 디버깅 시간 단축
- 애자일과의 시너지(사용자 중심적)
- 테스트 문서 대체
- 추가 구현 용이

Software Release Life Cycle

초기 개발단계부터 마지막 출시까지 주기를 나타냄

Testing and development period

Pre-alpha

- 테스트 이전까지 진행되는 요구사항 분석, 설계, 개발, 유닛 테스트를 포함
- 핵심 기능이 동작하기 시작한 상태

Alpha

- 소프트웨어 테스트를 시작하는 첫 단계
- 회사 내부 테스터를 통해 진행하며, 피드백을 통해 소프트웨어를 개선함

Beta

- 외부에 직,간접으로 오픈하여 테스트를 진행
 - 오픈 베타: 일반 유저에 모두 오픈하여 기능을 제공함.
 - 클로즈드 베타: 신청자 중 일부에 접근권한을 부여하고, 테스트를 진행함.

RC(Release Candidate)

- 정식 제품이 될 가능성이 있는 베타버전. 심각한 문제가 없다면 이들 중 하나가 정식 버전이 됨.

Release period

RTM(Release to Manufacturing)

- 소프트웨어를 유저에게 제공될 준비가 완료된 상태

GA(General Availability)

- 소프트웨어를 유저가 이용 가능한 상태

Agile Process

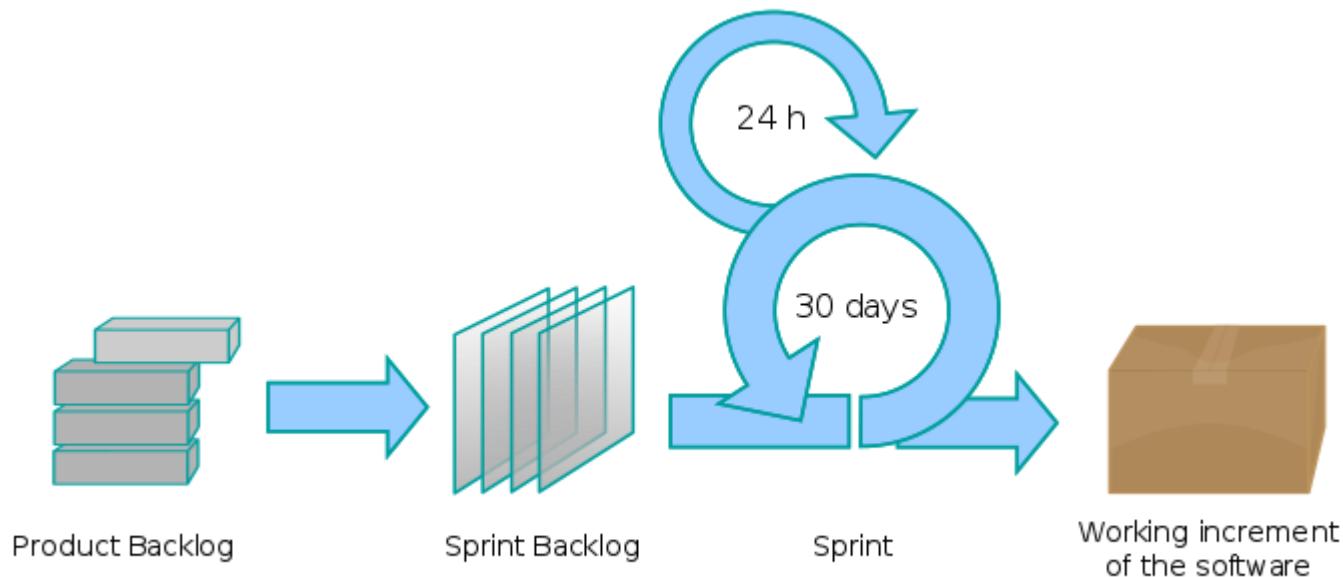
Scrum

- an iterative and incremental agile software development framework for managing product development

Scrum

- The product owner represents the stakeholders and is the voice of the customer, who is accountable for ensuring that the team delivers value to the business.
- The development team is responsible for delivering potentially shippable increments (PSIs) of product at the end of each sprint (the sprint goal).
- facilitated by a scrum master, who is accountable for removing impediments to the ability of the team to deliver the product goals and deliverables.

Sprint



Planning Poker

- 애자일 추정을 위해 사용하는 도구
- 모든 팀원이 한가지 과제에 대해 충분히 토론하고 작업시간을 추정하기 위함
- deck 구성
0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?, 무한대, 커피
- 점수는 단위 작업시간(8시간)을 의미함

Planning Poker

- 플레이방법
 - i. 추정할 과제를 가장 잘 아는 사람이 해당 과제에 대해 설명합니다.
 - ii. 다른 사람은 추정에 필요한 정보를 얻기위해 질문과 토의를 합니다.
 - iii. 각자 생각하는 이 과제의 점수를 보이지 않게 내려놓습니다.
 - iv. 점수를 공유하고 가장 낮은 점수, 가장 높은 점수를 낸 팀원이 이 점수를 낸 이유에 대해 설명합니다.
 - v. 모든 팀원이 같은 점수를 낼 때 까지 3~4의 반복

일정 추정 과제

1. Ethiopian Multiplication
2. stackoverflow python 검색결과 웹 스크래핑
3. Fizzbuzz

Pair Programming

- 시니어와 주니어가 한 팀을 이뤄 노하우를 전수하거나 같은 과제에 대해 충분한 논의를 함으로써 생산성 향상을 도모
- Navigator와 Driver가 한 팀을 이뤄 실시
- Navigator는 해당 과제에 대해 주도적으로 의견을 제시하고 Driver는 Navigator 가 지시하는 대로 작업하되, 이해되지 않는 부분이 있다면 이의를 제기
- 약속한 시간이 지나면 Navigator와 Driver의 역할 변경
- 과제를 해결할 때 까지 반복

Pair Programming

So, Let's Try!!

Code Review

검토사항

- 요구사항
- 설계요구 충족여부
- 과도한코딩
- 같은 기능
- 함수의 입출력
- 빌딩블록(API, 라이브러리, 자료구조, ..)
- 변수 사용전 초기화

Way of Developer

Developer



Developer

- 시스템 분석가의 요구에 맞게 컴퓨터 프로그래밍을 하거나 시스템 설계를 하는 사람

Difference between Developer, Programmer, Coder

- Developer: can design system
- Programmer: can design algorithm
- Coder: can produce code from pseudo code

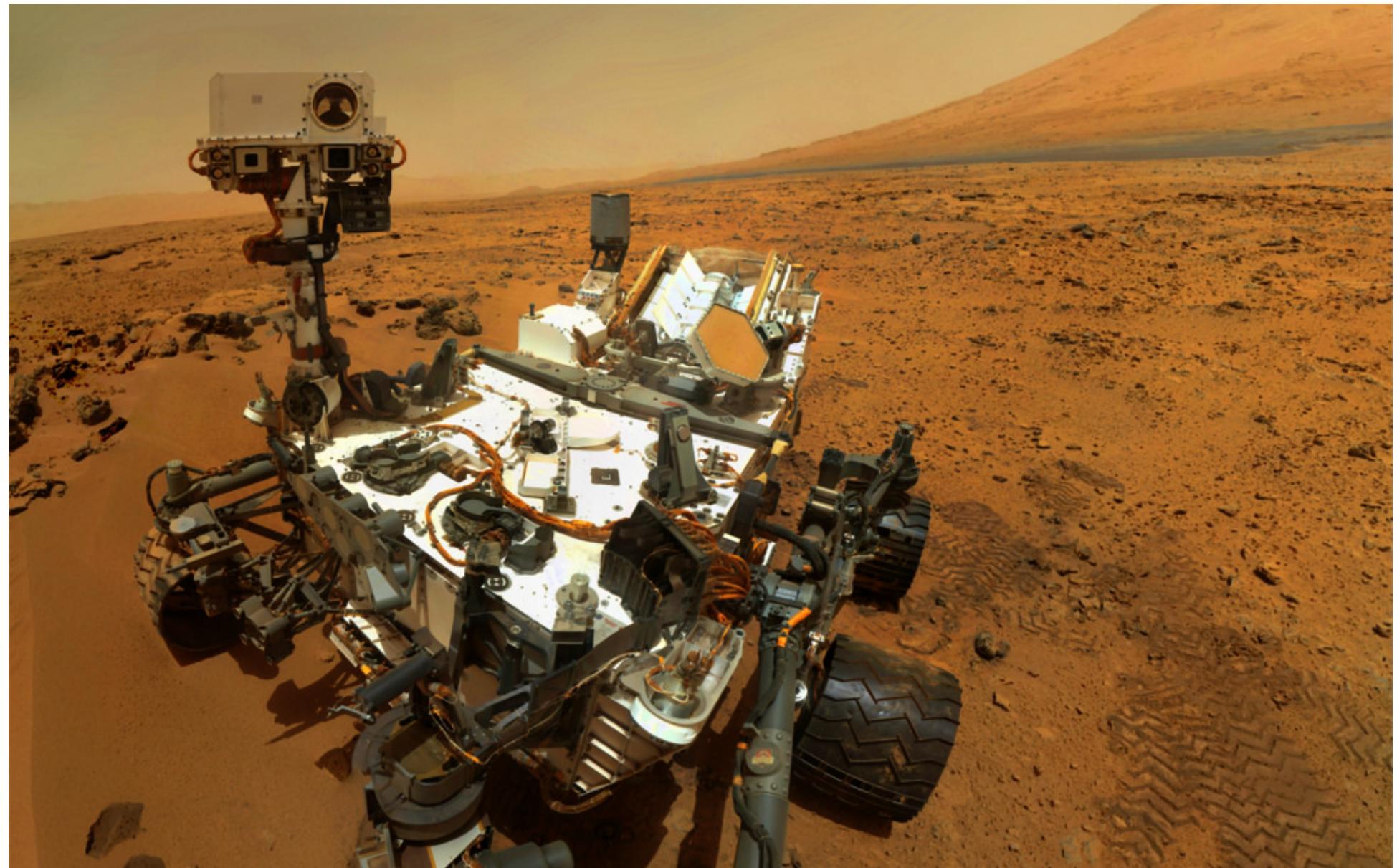
개발자가 갖춰야 할 덕목

Geekiness

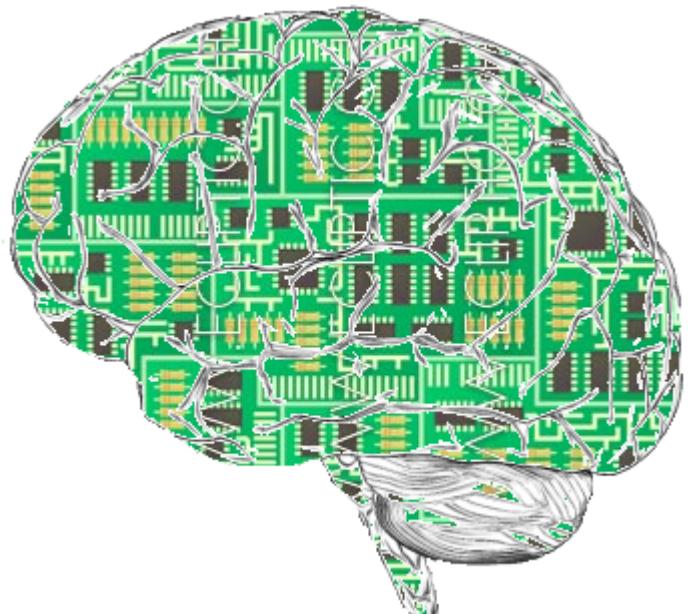


Geek.

Curiosity



Computational Thinking



개발자가 알아야 할 언어

개발자가 알아야 할 언어

A B C D E F G H I J
K L M P O Q R
S T V W X Z a b c
d e f g h i j k l m n o p q r s
t u v w x y z 0 1 2 3 4 5 6
7 8 9 ! " \$ % ? & * ()

개발자가 함께 일 할 사람





개발자가 함께 일 할 사람

