

일반적인 소프트웨어 개발 과정

한기용

목차

- How to Plan?
- How to Execute/Track?
- How to Manage Source Code
- How to Test
- How to Build
- 예제 하나
- 개발자용 메신저
- 속제

시작하기 전에

- 리눅스가 대세
 - 윈도우를 쓰는 회사는 거의 없음
- 오픈소스가 대세
 - 얼마전 마이크로소프트에서 C#을 오픈소스화 함
- SI vs. Product

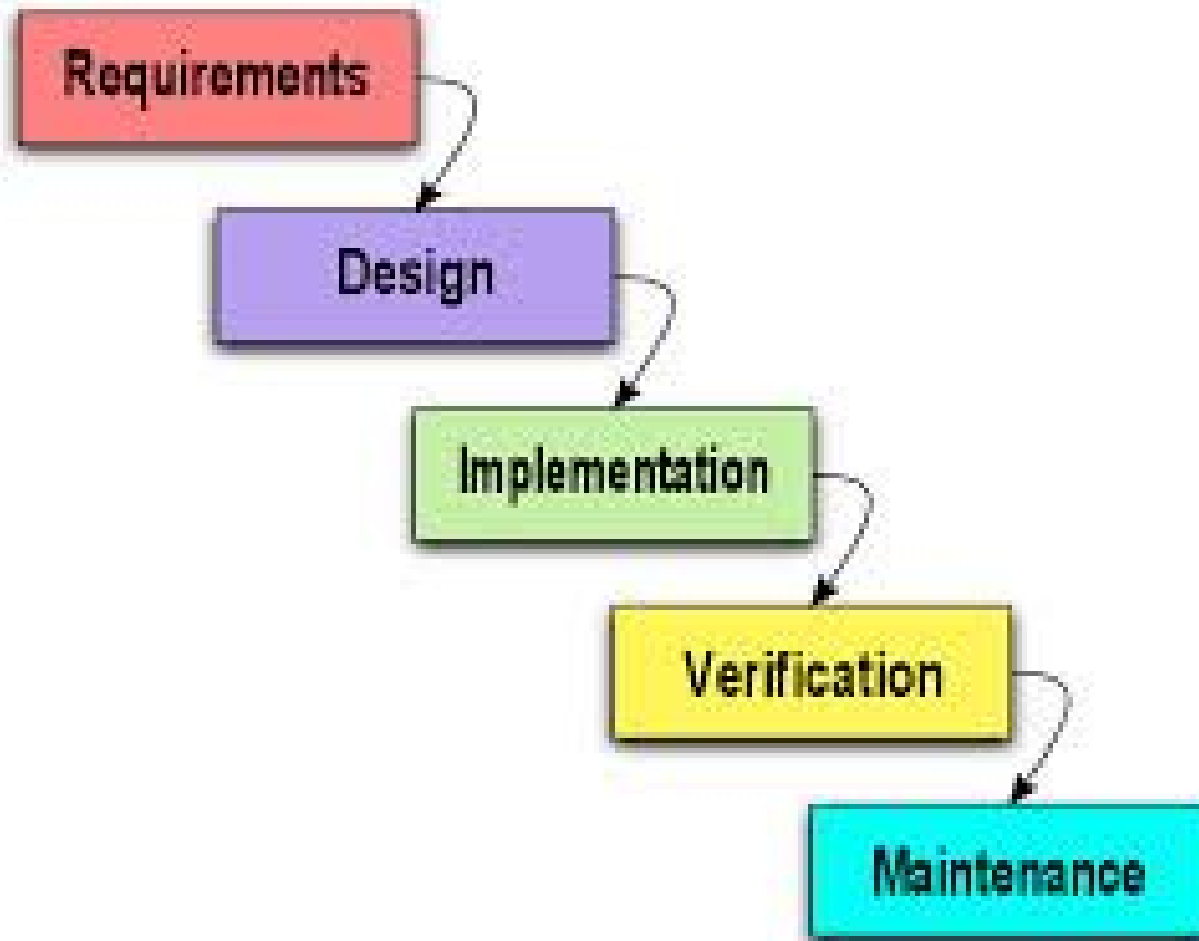
소프트웨어 개발 계획은 어떤 방식으로 하는가?

HOW TO PLAN

소프트웨어 개발의 어려움

- 요구조건은 계속해서 변화한다
- 디자인시 모든 문제를 미리 알 수 없다
- Water-Fall 모델은 소프트웨어 개발에 부적합
 - 요구조건 -> 디자인 -> 개발 -> 테스트 -> 릴리스 -> 유지보수
 - 대규모 프로젝트일수록 디자인에 걸리는 시간이 길어지며 그 기간사이에 요구조건이 바뀔 수 있으며 실제 개발에 들어가면 생각치 못했던 문제들로 디자인을 변경해야할 수 있음

Water-Fall 모델



애자일 개발(Agile Development)

방법론 등장

- 아는 만큼 보이는 만큼 만들어가자!
 - 매 사이클마다 바로 쓸 수 있는 기능을 구현
- 짧은 사이클이 특징 (보통 2주)
 - 이를 보통 스프린트(Sprint)라고 부름
- 매 사이클마다 다음 작업을 반복
 - 작업별로 우선순위 결정 (Backlog Prioritization)
 - 보통 PM(Product Manager)들이 수행
 - 이번 사이클에 일할 작업 결정 (Planning)
 - 매일 Standup 미팅 (5분 – 10분)
 - 마지막날 Retrospective & Demo 미팅

애자일 스크럼(Agile Scrum) 모델

The Agile: Scrum Framework at a glance

Inputs from Executives,
Team, Stakeholders,
Customers, Users



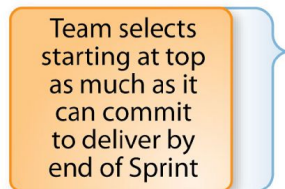
Product Owner



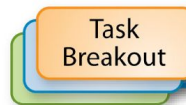
The Team



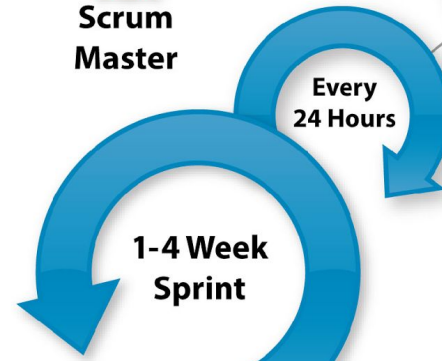
Product Backlog



Sprint Planning Meeting



Sprint Backlog



Sprint end date and team deliverable do not change



Scrum Master



Burndown/up Charts



Every 24 Hours



Sprint Review



Finished Work



Sprint Retrospective

소프트웨어 개발의 진척을 어떻게 관리하는가?

HOW TO EXECUTE/TRACK?

흔히 사용되는 툴

- JIRA, Trello, ...
- JIRA
 - 프로젝트 관리를 위한 전반적인 기능을 제공하는 툴로 호주의 Altassian이란 회사에 의해 개발
 - Agile Scrum 관리 툴, SVN (소스코드 컨트롤), Wiki (협업 문서툴) 등등
 - 많은 수의 회사들이 JIRA를 프로젝트 관리를 위해 사용
- Trello
 - 애자일 스크럼 관리를 위한 툴로 JIRA에 비해 훨씬 더 직관적인 단순한 인터페이스를 제공

Team Scrum Board

Plan Work Report Board ▾

SPRINT: Sprint 3 ▾ QUICK FILTERS: Product UI Server Only My Issues Recently Updated ⤴

To Do

+ TIS-28
↑ Research options to travel to Pluto



5

🔴 TIS-8
↑ Requesting available flights is now taking > 5 seconds



In Progress

+ TIS-27
↑ Add Phobos and Deimos Tours as a Preferred Travel Partner



8

🔴 TIS-10
↑ Bad JSON data coming back from hotel API



+ TIS-25
↑ Engage Jupiter Express for outer solar system travel



5

+ TIS-20
↑ Engage Saturn Shuttle Lines for group tours



3

In Review

🔗 TIS-58
♥ Add feedback button to the plugin sample code



💡 TIS-45
↑ Email non registered users to sign up with Teams In Space



2

Done

🔴 TIS-9
🚫 After 100,000 requests the SeeSpaceEZ server dies



💡 TIS-16
↑ Establish relationship with local office supplies company



3

🔴 TIS-7
↑ 500 Error when requesting a reservation



💡 TIS-11
↑ Register with the Mars Ministry of Labor



2

소프트웨어 소스 코드를 어떻게 관리하는가?

HOW TO MANAGE SOURCE CODE?

소스 버전 컨트롤이란?

- “개발자들이 자신이 개발하는 소프트웨어의 소스코드에 발생하는 변경사항들을 관리할 수 있도록 해주는 프로그램”

왜 소스 버전 컨트롤을 사용하나?

- 코드 백업의 역할 수행
- 코드에 생기는 변경 사항들을 쉽게 추적 가능(예를 들어 에러시 이전 버전으로 Rollback)
- 두 사람 이상이 공동 개발시 코드의 공유와 변경이 용이
- 최근 시스템들은 코드리뷰도 지원

어떤 버전 컨트롤 소프트웨어들이 있 나?


- CVS (Concurrent Version System)
- SVN (SubVersionN)
- **Git/Github**
 - 가장 인기있는 버전 컨트롤 소프트웨어. 웬만한 오픈소스 소프트웨어들은 Github상에 존재
 - 바로 우리가 사용할 것이 GitHub
 - GitHub == Git + Bug Tracking (Issue) + Wiki
 - 2일차에 자세히 다뤄볼 예정

코드 리뷰

- 주니어 개발자나 새로 온 개발자들을 트레이닝시키는 최선의 방법
- 단점은 리뷰를 해야하는 사람들이 이미 바쁜 사람들이라는 점
 - 스프린트 플래닝시 이를 고려하여 태스크 할당

좋은 코드 리뷰 방법

- 코드 리뷰를 요청하는 이
 - 요청시 되도록이면 조금씩 자주 요청. Unit test와 같이 요청하면 최상
 - 주석을 최대한 추가하고 무슨 이유에서 뭘 하려고 하는 것인지 설명
- 코드 리뷰를 하는 이
 - 코딩 스타일에 대한 것보다는 코드 자체에 대해 이야기
 - 충분히 시간을 들여 도움이 되는 리뷰를 제공
- 코드리뷰에 편리한 툴 사용

 octocat:master ... hubot:master Edit

Create pull request

Discuss and review the changes in this comparison with others.

?

1 commit

1 file changed

0 commit comments

1 contributor

Commits on Sep 10, 2014

hubot

Change description of the repository

8b7afff

Showing 1 changed file with 1 addition and 1 deletion.

Unified

Split

2 README.md <> View

...	...	@@ -1,6 +1,6 @@
1	1	### Well hello there!
2	2	
3		-This repository is meant to provide an example for <i>*forking*</i> a repository on GitHub.
	3	+This is my fork of the octocat/Spoon-Knife repository.
4	4	
5	5	Creating a <i>*fork*</i> is producing a personal copy of someone else's project. Forks act as a sort of bridge between the original repository and your personal copy. You can submit <i>*Pull Requests*</i> to help make other people's projects better by offering your changes up to the original project. Forking is at the core of social coding at GitHub.
6	6	

코드 리뷰 예

```
def compute_average(numbers):  
    sum = 0  
    for number in numbers:  
        sum += number  
    return sum/len(numbers)
```

소프트웨어를 어떻게 테스트하는가?

HOW TO TEST?

Test Driven Development

- 개발시 테스트를 어떻게 할 것인지부터 생각
- 코드 구성 자체가 테스트에 편리하게 되는 효과가 발생
 - 이는 또한 자신이 만드려는 기능에 대해 더 생각하게 하는 부수적인 효과가 있음
 - 코드 자체가 잘 구성되어 있지 않으면 테스트가 불가능

어떤 테스트들이 존재하는가?

- Unit Test
 - 모듈의 특정 기능 (함수) 테스트
- Integration Test
 - 여러 모듈을 통합하여 하는 한 차원 위의 테스트
- UI Test
 - 요즘은 Selenium 등의 툴을 이용해서 웹페이지 자체의 기능을 테스트하는 것이 대세

테스트의 중요성

- 많은 회사들이 코드 변경의 일부로 Unit Test를 의무적으로 요구
 - 테스트가 없으면 아예 코드 체크인이 실패
- 테스트가 많을수록 잇점이 존재
 - 시스템의 안정성 증대
 - 나중에 Refactoring할 경우 혹은 새로 들어온 신입 엔지니어가 코드 수정할때 굉장히 편리
- 어떤 경우에는 테스트를 작성하기가 너무 힘든 경우들이 있음
 - 이 경우 태스크 플랜닝때 시간을 넉넉히 배당
 - 어떤 경우에는

소프트웨어를 어떻게 빌드하는가?

HOW TO BUILD?

빌드란?

- 자신(혹은 팀)이 개발한 소프트웨어를 최종적으로 출시하기 위한 형태로 만드는 것의 의미
 - 테스트가 빌드의 중요한 일부로 포함
- 참여 개발자들이 많을수록 이는 더 중요
- 개발이 끝나기 전부터 빌드를 하면 소프트웨어의 안정성 증대
 - Continuous Integration!

Continuous Integration이란?




- Software Engineering Practice의 하나
- 기본 원칙
 - 코드 Repo는 하나만 유지
 - 코드변경을 최대한 자주 반영 (적어도 하루에 한번)
 - 테스트를 최대한 추가
 - Test Coverage
 - 빌드를 계속적으로 수행 (자동화)
 - Commit Build vs. Nightly Build
 - 성공한 빌드의 프로덕션 릴리스 (자동화)

빌드 실패!

- 많은 회사들이 빌드 실패시 빌드가 다시 성공할때까지 코드 변경을 금지
 - 즉 빌드 실패는 모든 사람들을 잡아두는 족쇄
 - 그래서 어느 정도 조직이 커지면 빌드만 전담하는 엔지니어가 생김. 이 사람의 업무 중의 하나는 빌드 실패시 누가 주범인지 알아내는 것 ☺
 - 빌드실패시 가벼운 형태로 페널티 부여

Jenkins란?

- 오픈소스 CI 빌드 소프트웨어
 - CI와 관련한 모든 기능을 지원 (직접 지원하는 것은 아니고 플러그인의 형태로 지원)
 - Github, Git, SVN등 대부분의 소스 컨트롤 시스템과 연동
 - Jenkins 자체는 cronjob 관리 소프트웨어에 가까움
- 빌드된 소프트웨어의 배포 (릴리스)를 위해서도 사용가능
- 썬 마이크로시스템에서 Hudson이란 이름으로 시작
 - 오라클이 썬을 인수하면서 오픈소스화가 안됨. Jenkins라는 이름으로 재탄생

 [Back to Dashboard](#) [Status](#) [Changes](#) [Workspace](#) [Build Now](#) [Delete Project](#) [Configure](#) [Subversion Polling Log](#)

Build History [\(trend\)](#)

	#27	Jun 22, 2011 9:03:59 AM
	#26	Jun 17, 2011 9:23:27 AM
	#25	Jun 15, 2011 10:18:55 AM
	#24	Jun 15, 2011 10:16:00 AM
	#23	Jun 14, 2011 3:01:41 PM
	#22	Jun 14, 2011 2:57:39 PM
	#21	Jun 14, 2011 2:53:01 PM
	#20	Jun 14, 2011 2:05:07 PM
	#19	Jun 14, 2011 1:50:49 PM
	#18	Jun 13, 2011 3:00:34 PM
	#17	Jun 9, 2011 10:48:55 AM
	#16	Jun 9, 2011 10:48:01 AM
	#15	May 27, 2011 11:01:34 AM

Project Project B

 [add description](#)[Disable Project](#)[Workspace](#)[Last Successful Artifacts](#)

- [project-b.war](#)
- [test-results.csv](#)
- [installer.exe](#)
- [System-Manual.pdf](#)
- [User-Manual.pdf](#)
- [Software Requirements Specification.doc](#)
- [Software Design Specification.doc](#)
- [Risk Analysis.doc](#)
- [Hazard Analysis.doc](#)
- [Release-Notes.xls](#)
- [Project-Plan.doc](#) 

[Recent Changes](#)

Permalinks

- [Last build \(#27\), 16 min ago](#)
- [Last stable build \(#27\), 16 min ago](#)
- [Last successful build \(#27\), 16 min ago](#)
- [Last failed build \(#24\), 6 days 23 hr ago](#)
- [Last unsuccessful build \(#24\), 6 days 23 hr ago](#)

예제 하나

Log에 국가코드 추가하기

1. 스프린트 플랜에서 웹서버 로그 파일에 국가코드를 추가하기로 결정
 - 그에 해당하는 태스크와 누가 할지 결정
2. 담당 엔지니어가 Github에서 그 기능이 들어갈 Branch 생성
3. 그 Branch에서 코드 수정하고 관련 유닛 테스트 생성/변경
4. 그 Branch를 Master로 합치기(Merge)위한 Code Review 요청 (Pull Request). 이 때 리뷰할 사람을 지정
5. 코드 리뷰어가 “Ship it!”라고 하면 Branch를 Master로 병합
 - 코드에 문제가 있을 경우에는 리뷰어와 계속해서 작업
6. 병합 후 Jenkins를 보고 빌드가 성공적으로 끝남을 확인
 - 빌드에 문제가 있으면 왜 문제가 있는지 해결
7. 프로덕션으로 릴리스

개발자용 메신저

개발자들만을 위한 메신저의 대두

- Slack, HipChat, ...
 - Slack의 Valuation이 \$1.12B
- 소스 컨트롤 시스템, 버그 트래킹 시스템, CI 시스템 등과 연동
- Python등으로 간단한 플러그인을 작성하여 기능 확장이 쉬움

Acme Sites

CHANNELS

api

changes

general

marketing

products1

random

sales

support

web

+3 more...

DIRECT MESSAGES

slackbot

david

don

eric

jessica

liz

myles

stewart

tara

+7 More...

kristel

online

#changes

Search

8

stewart1:05 PM

Im'ma tweeting that

<https://twitter.com/glitchlog/status/223600465462566912>

Glitch Change Log@glitchlog

12 new hairstyles & 19 new skin colors have been added to the Vanity. Taste the hairy rainbow! Forum post:

<http://www.glitch.com/forum/general/24360/>

Thursday, July 12th, 2012 7:11 PM

myles1:07 PM

Entering a street with a Qurazy for you now shows a pretty animated overlay where the growl used to go, for 10 seconds

liz1:09 PM

Fixed bug #9358. Piggies shouldn't get stuck walking in place any more.

bugbot1:10 PM★

[Bug #9358](#)

Title

Piggies are stuck walking in place

CreatorAssignee

mackenzieliz

StatusPriority

fixednormal

eric1:14 PM

@kristel, can you test on [dev.glitch.com](#) to see if coat sleeves are now working as expected?

kristel1:15 PM

indeed they are! thanks @eric!

숙제

1. GitHub에 자기 프로그램 하나 올리 기

- 계정 생성
- 업로드할 프로그램은 아무 것이나 상관없음
- 올린 후 그 링크를 다음 이메일로 보낼 것:
keeyonghan@hotmail.com
- 이후 조 작성 후 조별로 collaborator 등록

2. 조별로 Slack 셋업하기

- 조별로 한 사람이 slack을 싸인업
 - <https://slack.com/>
- 이후 팀원들을 초청
- 나중에 해볼 수 있는 일
 - Github과 Slack 연동
 - Jenkins와 Slack 연동

3. 다음 코드 작성하기

- 10진수가 주어졌을 경우 이를 16진수로 바꾸어 출력하기. 언어 상관 없음
- 작성 후 팀원 한 사람에게 리뷰 요청 (github 사용)

4. 숙제

- 영문 이력서 작성
- 학교 프로젝트나 개인 프로젝트들을 꼭 나열해볼 것