



Package and Release

패키지와 릴리즈

컴퓨터공학부
천세진

목 차

- 릴리즈 넘버링(Release Numbering)
- 릴리즈 브랜치(Release Branches)
- 릴리즈 안정화(Release Stabilize)
- 패키징(Packaging)
- 테스트링(Testing)과 릴리징(Releasing)
- 다수의 릴리즈 라인을 유지하기
- 릴리즈와 데일리 개발(Daily Development)



STABILIZING A RELEASE

STABILIZATION

- RELEASE BRANCH를 릴리즈가 가능한 상태로 만드는 과정
- Feature-Based Release
 - The last-minute feature rush
 - 많은 변화를 넣을 수록, 코드는 덜 안정화되고 더 많은 새로운 버그들이 출현
- Time-Based Release
 - 규칙적인 리듬에 새로운 릴리즈를 내놓는 것
 - 매 6개월마다, 버그와 특징에 관계없이 릴리즈



릴리즈 안정화 과정은 Saying “No”를 위한 메커니즘을 만드는 것

- 오픈소스 프로젝트의 트릭은 “NO”라고 말하는 방법을 고안하는 것
 - 실망이나 상처받는 개발자를 초래하지 않도록
 - 릴리즈에 들어갈 만한 변경사항을 막지 않도록
- 두 가지 전략
 - Release Owner에 의한 리더십
 - 변화에 대한 투표



<https://www.lianedavey.com/know-when-and-how-to-say-no/>

Dictatorship by Release Owner

- Release Owner에게 최종 결정을 위한 충분한 권한을 준다
- 릴리즈에 대한 일반적인 패턴
 - 이 변화에 대해 잘못된 것은 없다고 생각하지만, 우리는 그것에 대해 test할 시간이 충분하지 않아서 이번 release에 넣기에는 어렵다
- 프로젝트 리더와 동일하지 않아도 된다
 - 프로젝트 리더의 판단에 대한 평행추(counterbalance)



Voting on Changes

- 개발자는 어떠한 변화가 릴리즈에 투입될지에 대해서 투표
 - **변화를 배제**하는 것이 안정화의 가장 중요한 기능임
- 개발자의 서브그룹은 각 변화가 릴리즈로 가는 것에 대해 협력해야 한다
 - 많은 사람들이 각 변화에 리뷰 할수록,
개별 개발자는 변화에 대해 투표하는 것을 망설이지 않는다
- 일반적으로, Release Branch에 적용하는 투표 원칙은
 - 적어도 3명의 개발자가 그것에 대해 투표해야 한다
 - 그 중의 한명은 “no” vote (Vetoes, 거부권) 을 가진다



Managing Collaborative Release Stabilization

- 대다수의 의견이 무엇인지 알고 개인의 의견을 내는 것이 좋다
 - One-line statement, Justification, Notes, Votes

```
* commit b31910a7180fc (issue #49)
Prevent client/server handshake from happening twice.
Justification:
  Avoids extra network turnaround;
  small change and easy to review.
Notes:
  Discussed in https://.../msg-7777.html
  and other messages in that thread.
Votes:
+1: jsmith, kimf
-1: tmartin (breaks compatibility with some
        pre-1.0 servers; true, those
        servers are buggy, but why be
        incompatible if we don't have to?)
```

I am not saying you should always *do* what the majority says. But it might be good to *know and consider* what the majority wants (or at least the people who did vote).



Release Manager

- 1-2명의 사람이 릴리즈 과정을 주도함
 - Release Manager라고 일컫음
 - 변화에 최종결정이 있는 Release Owner와는 차이가 있음
- 주요 역할
 - 얼마나 많은 변화가 현재 고려 중에 있는지 추적(Under consideration)
 - 얼마나 많은 변화가 승인되었는지
 - 얼마나 많은 것이 승인될 가능성이 있는지



PACKAGING

Packaging

- Free 소프트웨어 배포의 전통적인 형태는 소스 코드
 - Compiled 혹은 Source form
- 패키징할 때, 주요 준수사항
 - Format
 - Name and Layout
 - Compilation and Installation
 - Binary Packages



- 소스 코드는 디렉토리 트리를 유지하는 표준 포맷 내로 제공되어야 함
 - Unix 및 Unix계열을 위한 TAR format
 - compress, gzip, bzip, bzip2
 - Window 계열을 위한 zip format
 - Javascript 프로젝트를 위한 Minified version
 - 소스파일도 함께 제공

Minification

■ Data compression 개념과는 차이가 있다

For example, the JavaScript code

```
// This is a comment that will be removed by the minifier  
var array = [];  
for (var i = 0; i < 20; i++) {  
    array[i] = i;  
}
```

```
for(var a=[i=0];i<20;a[i]=i++);
```

https://en.wikipedia.org/wiki/Minification_%28programming%29



Name and Layout

- 패키지의 이름: 소프트웨어 이름 + 릴리즈 번호 + archive 타입에 따른 적절한 suffix 로 구성됨
 - Scanley-2.5.0.tar.gz
 - Scanley 2.5.0 이라면
- 디렉토리 Top level에는
 - README, INSTALL, LICENSE/COPYING, CHANGES/NEWS 파일이 존재

```
Version 2.5.0
(20 December 2022, from branch 2.5.x)
http://scanley.org/repos/tags/2.5.0/
```

New features and enhancements:

- * Added regular expression queries (issue #53)
- * Added support for UTF-16 documents
- * Documentation translated into Malagasy, Polish, Russian
- * ...

Bugfixes:

- * fixed reindexing bug (issue #945)
- * fixed some query bugs (issues #815, #1007, #1008)
- * ...

CHANGES 파일의 예



Name and Layout

- 릴리즈는 정적 참조 포인트에서 파일들임
 - Working copy 혹은 working files 단계가 아니어야 함
- 여러 개의 소스 패키지에서 마이너한 차이점은
 - CRLF(Carriage Return and Line Feed)와 LF간의 차이임
 - CRLF=Windows, LF=Unix계열
- 첫 글자가 소문자/대문자 여부에 관계없다
 - 다만 압축해제 시 생성되는 폴더 이름에는 주의할 것
- Pre-Releases
 - scanley-2.3.0-alpha1.tar.gz
 - scanley-2.3.0-alpha2.tar.gz
 - scanley-2.3.0-beta1.tar.gz
 - scanley-2.3.0-beta2.tar.gz
 - scanley-2.3.0-beta3.tar.gz
 - scanley-2.3.0.tar.gz



Compilation and Installation

- Unix계열 시스템에서, C/C++ 기반 프로그램은 다음과 같이 사용자에게 요구

```
$ ./configure  
$ make  
$ sudo make install
```

첫번째 명령어는 빌드 과정을 위한 환경에 대해 자동 감지
두번째 명령어는 소프트웨어를 빌드
마지막 명령어는 시스템에 빌드된 소프트웨어를 설치

- 오늘날, Docker 기반의 컨테이너 기반 환경에 의해 손쉽게 프로그램을 배치(Deploy)할 수 있음



Binary Packages

- Binary는 Compiled 라는 뜻보다, 패키지의 사전설정된 형태라는 뜻
 - 사용자가 소스 기반의 빌드 및 인스톨 과정을 생략하도록 함

Get Started with Docker

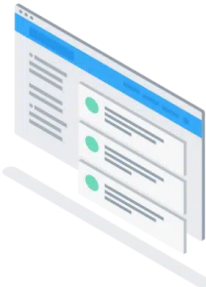
We have a complete container solution for you – no matter who you are and where you are on your containerization journey.



Docker Desktop

Developer productivity tools and a local Kubernetes environment.

Download for Windows



Docker Hub

Cloud-based application registry and development team collaboration services.

Signup

BuildKit

A set of tooling for building and packaging software using containers

[View on GitHub](#)

Compose

Define and run multi-container applications with Docker

[View on GitHub](#)

containerd

The industry-standard core container runtime, donated to the CNCF (Cloud-native Computing Foundation) with an emphasis on simplicity, robustness and portability

[View on GitHub](#)

DataKit

A tool to orchestrate applications using a 9P dataflow

[View on GitHub](#)

Docker CLI

The cli used in the Docker CE and Docker EE products.

[View on GitHub](#)

Docker Distribution

The Docker toolset to pack, ship, store, and deliver content.

[View on GitHub](#)

HyperKit

A toolkit for embedding hypervisor capabilities in your application

[View on GitHub](#)

InfraKit

A toolkit for creating and managing declarative, self-healing infrastructure

[View on GitHub](#)

Libnetwork

A native Go implementation for connecting containers

[View on GitHub](#)



Binary Packages

- 공식 소스 릴리즈에 binary 패키지의 base를 확실히 해야한다
 - 일부 개발자는 unstable한 코드를 가져오거나
 - 릴리즈 이후에 수행된 변화에 커밋 완료된 파트를 포함할 수 있다.
- Packager는 수정사항을 개발자에게 가져와서 그들의 계획에 표현하기를 독려한다.
 - Packager들은 주로 the best out-of-the-box experience 를 원한다



Best out-of-the-box experience(최상의 즉시 사용 경험)

- 개발자는 개발한 SW가 어떠한 버전인지 파악하길 원함
 - 일관된 버그와 호환성 보장에 대한 문제에 대해 고민
- 설치 또는 초기 설정 후 바로 사용하는 과정에서 사용자에게 제공하는 최고의 경험을 의미
 - 즉, 별도의 추가 설정이나 학습 없이도 사용자가 쉽게 사용할 수 있는 상태를 강조



**It's fine to disagree with packagers,
but don't flame them**

best out-of-the-box experience

<https://www.chainalytics.com/how-to-improve-out-of-box-experience-for-your-customer/>

주로 사용되는 분야

■ 소프트웨어(SW)

- 설치 후 바로 직관적으로 사용할 수 있는 UI/UX를 제공하는 경우
- 예: macOS, iOS, Ubuntu 등이 설정 없이도 깔끔한 사용 경험을 제공하는 경우

■ 하드웨어(HW)

- 제품을 개봉하고 기본 설정만으로 즉시 사용할 수 있는 경우
- 예: 애플 제품(아이폰, 맥북)은 박스에서 꺼내자마자 간단한 설정 후 바로 사용 가능

■ 서비스(클라우드, AI, 플랫폼 등)

- 초기 설정이 최소화되어 있으며, 사용자가 즉시 작업을 수행할 수 있는 경우
- 예: ChatGPT, Google Docs 등은 별다른 설정 없이 계정만으로 즉시 사용 가능

