Software Release Management

Thomas Lampoltshammer
Bernhard Zechmeister

University of Salzburg
Department of Computer Sciences

June 11, 2011
Table of Contents

1 Introduction
   • Concepts of Software Release Management
   • Release Management (ITIL)

2 Problems

3 Tools
   • Subversion
   • Git

T. Lampoltshammer, B. Zechmeister
Software Release Management
Software characteristics

- software is dynamically changing
- continuous development cycle
- programming, testing and releasing
- maintenance costs
Version Control

- track changes
- restore older states
- revert unwanted changes
- multiple developers
- centralized or distributed
Issue Tracking

- structured overview
- bugs, features, requests
- manage responsibility
- monitor progress
- knowledge base
Quality Requirements

- evaluate targets for a release
- determine development states
- specific test plans
Release Process

- identification of requirements
- planning release
- release decision
- risk analysis
- quality control
- release creation
- release documentation
IT Infrastructure Library

- started in 80’s by British government
- open, non proprietary, best-practice quasi-standard
- document pool of successful stories
- no panacea or tool as such
- two main areas
IT Infrastructure Library - Service Delivery Set

- strategic support for existing IT environment
- service-level management
- finance management
- availability management
- capacity management
- continuity management
software release management

IT Infrastructure Library - Service Support Set

- operational support for existing IT environment
- accident management
- problem management
- configuration management
- change management
- release management
IT Infrastructure Library - Release Management

Figure: ITIL Release Management
Problems and Misconceptions
Cost and Value Functions - Customer

\[ C_{val}(update) = value(newFeatures) + value(removalOfWorkarounds) \]  

\[ Cost(update) = \{ \text{cost(downtime)} + \text{cost(training)} + \text{cost(updateEffort)} + \text{cost(lowFunctionality)} + \text{cost(paymentToVendor)} \} \]  

\[ C_{val}(update) > C_{cost}(update) \]
Cost and Value Functions - Vendor

\[
V_{\text{value}}(\text{newUpdatePackage}) = \begin{cases} 
\text{newCustomers} \times \text{priceNewRelease} + \\
\text{oldCustomers} \times \text{priceOfUpdate} + \\
\text{costReduction(support)} 
\end{cases}
\]

\[
V_{\text{cost}}(\text{newUpdatePackage}) = \begin{cases} 
\text{cost(development)} + \\
\text{cost(updateCurrentCustomers)} + \\
\text{cost(deliveryToCustomers)} + \\
\text{cost(packageCreation)} + \\
\text{cost(increasedSupport)} + \\
\text{cost(marketing)} 
\end{cases}
\]

\[
V_{\text{val}}(\text{update}) > V_{\text{cost}}(\text{update})
\]
Common Misconceptions

- customers want to stay up-to-date
- customers must stay up-to-date
- fixes can be postponed to the next major release
- workarounds must be avoided at all costs
- customers always want new features
- a quiet customer is a happy customer
- customers read release notes
Tool: Subversion
Subversion

- version control system
- free, open-source
- centralized
History of Subversion

- started in 2000 by CollabNet, Inc.
- replacement for Concurrent Version System (CVS)
- self hosting in 2001 (after 14 months)
- since 2010 top-level-project in Apache Software Foundation
Versioning Models

Figure: Lock-Modify-Unlock
Versioning Models

**Figure:** Copy-Modify-Merge 1/2
Versioning Models

**Figure:** Copy-Modify-Merge 2/2
Tags and Branches

- cheap copies
- tags are snapshots
- release branches
- feature branches

Figure: Revision Diagram
Managing Releases

- version numbering
- major.minor.patch
- stabilization process
Problems and Limitations

- single point of failure
- difficult merging of conflicting changes
- access control
- addressed by distributed systems
Tool: Git
Git

- version control system
- free, open-source
- distributed
initiated by Linus Torvalds
issues with Bitkeeper
used for Linux kernel development and a lot of other open-source projects
Git Index

**Figure:** Git - Index
Git Object Store

Figure: Git Objects
Figure: Git - Commits
Git Branches

**Figure:** Git Branches
Git Branches

Figure: Git Branches
Git Branches

Figure: Git Branches
Git Branches

**Figure**: Git Branches
Git Merges

**Figure:** Git Merges
Git Merges

Figure: Git Merges
Git Protocols

- local protocol
- SSH
- Git protocol
- HTTP/S
Problems and Limitations

- missing lock-modify-unlock support
- limited compatibility on Microsoft Windows
- lack of graphical user interface
Summary

- maintain overview
- saving costs
- common use of Subversion
- Git is developing fast (mysysgit, TortoiseGit, ...)
- Git as serious alternative
Thank you for your attention!