DVCS

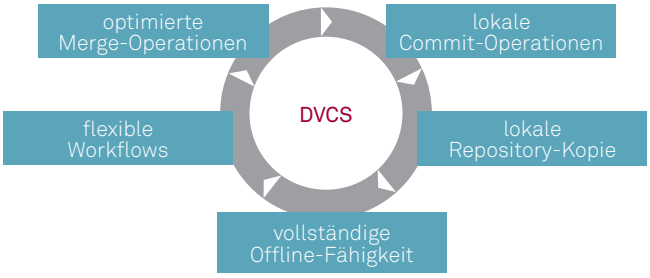
Distributed Version Control System

Version Control Systems (Versionsverwaltungssysteme, DVCS) sind schon lange ein unverzichtbarer Bestandteil jeglicher Art von Softwareentwicklungsprojekten. Reproduzierbarkeit von ausgezeichneten Softwareständen, Nachvollziehbarkeit von Änderungen und die damit einhergehende Risikominimierung und Effizienzsteigerung sind wichtige Erfolgsfaktoren, die durch ein VCS gewährleistet werden. Die neueste Generation, als Distributed Version Control Systems (verteilte Versionsverwaltungssysteme, DVCS) bezeichnet, bietet neben mehr Flexibilität verschiedene Vorteile, die gerade agile Projektvorgehen sehr gut unterstützen.

Definition

In Softwareentwicklungsprojekten nimmt die Versionsverwaltung eine zentrale Rolle ein und hat damit eine Reihe wichtiger Schnittstellen. Das Version Control System (VCS) dient als Quelle für das Build Management, muss das angestrebte Release Management technisch manifestieren, ist ein wichtiger Baustein der Team Collaboration und im Rahmen des Projektmanagement eng mit dem Issue Tracking verzahnt.

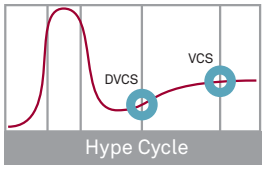
Ein VCS hat die Aufgabe, Änderungsverfolgung an Dateien und Dokumenten zu gewährleisten, durchgeführte Änderungen präzise zu identifizieren und jederzeit ältere Entwicklungsstände zur Verfügung zu stellen. Zudem können Entwicklungen in parallele Stränge aufgeteilt (branching), wieder geordnet und nachvollziehbar zusammengeführt (merging) werden. Die Verwaltung der Artefakte erfolgt dabei in einem



sogenannten Repository. Das wesentliche Merkmal einer verteilten Versionsverwaltung (DVCS) ist, dass jeder Nutzer eine Kopie des gesamten Repository lokal zur Verfügung hat, im Gegensatz zur zentralen Versionsverwaltung, bei dem es ein gemeinsam genutztes Repository gibt. Ein DVCS bietet ein vollständiges Offline-Arbeiten mit lokalen Commit-Operationen und unterstützt verschiedene Workflows, da Nutzer Änderungen entweder direkt untereinander austauschen können oder, wie mit einer zentralen Versionsverwaltung, Änderungen über ein zentrales Repository verteilt werden. Ein verbesserter Algorithmus zur Identifikation der verwalteten Dateien und die Verfügbarkeit aller Metadaten gewährleisten eine optimierte Konfliktlösungsstrategie bei Merge-Operationen.

Reifegrad

VCS sind auf Grund ihrer mittlerweile 40-jährigen Historie Commodity in jedem Entwick-



lungsprojekt. DVCS als aktuellste Ausprägung von VCS hat mittlerweile einen für den Enterprise-Einsatz erforderlichen Reifegrad erreicht und findet gerade in agilen Projekten immer mehr Verbreitung.

Marktübersicht

Die Vertreter der **ersten Generation** der VCS (Local VCS) wie Revision Control System (RCS) und Source Code Control System (SCCS) sind nicht über ein Netzwerk verwendbar, operieren immer nur auf einzelnen Dateien und erlauben keine parallel stattfindenden Änderungen.

Die Versionen der **zweiten Generation** der VCS (Centralized VCS) deren bekannteste Vertreter Concurrent Versions System (CVS) und Apache™ Subversion® (SVN) sind, sind als Client-Server-Systeme aufgebaut, mit einem zentralen Repository, erlauben Operationen auf mehreren Dateien und einem als „merge before commit" bezeichneten Nutzungskonzept.

Die **dritte Generation** der VCS (Distributed VCS), deren Hauptvertreter Bazaar, Git und Mercurial sind, zeichnet sich durch verteilte Repositories aus, verwaltet Änderungen in sogenannten Change-Sets und erlaubt Commit- und Merge-Operationen unabhängig voneinander („commit before merge").

Alternativen

Eine Alternative zu DVCS sind die Centralized VCS, die aktuell auch noch einen höheren Verbreitungsgrad besitzen. Auf eine Versionsverwaltung zu verzichten, stellt keine Alternative mehr dar. Für nicht-Text-Artefakte gibt es alternativ Document-Management-Systeme (DMS), die auch eine Versionierung beinhalten.

Referenzszenario

Im Enterprise-Umfeld wird auf Grund der organisatorischen Komplexität typischerweise eine am Centralized VCS angelehnte Repository-Hierarchie eingesetzt. Es gibt ein Master-Repository auf einem Server, mehrere lokale Entwickler-Repositories und ein (temporäres) Continuous-Integration-Repository auf dem Build-Server. Damit können die Vorteile des zentralen Ansatzes mit den Vorteilen eines DVCS vereint werden.

Business Impact

DVCS adressieren die Herausforderungen, die auf Grund der geografischen Verteilung von Entwicklerteams bei der verteilten Entwicklung entstehen. Weiterhin wird das Refactoring erleichtert, das wiederum der agilen Software-Entwicklung entgegen kommt. DVCS bietet schlussendlich eine Flexibilitäts- und Effizienzsteigerung und eine Risikominimierung.

Pro	Contra
Branching und Merging werden sehr gut unterstützt	Mehr Eigenverantwortung und Disziplin bei Entwicklern notwendig
Mehr Flexibilität	Hohe operative Komplexität der Werkzeuge
Effiziente Dateiablage	Dateien können nicht gelockt werden
Vollständige Offline-Fähigkeit mit lokalen Commits	Bedingte Authentifizierung (externe Hilfsmittel) und keine Autorisierung
Optimierte Konfliktlösung bei Merge-Operationen	Ungeeignet für große Binärdateien.

msg systems ag
Robert-Bürkle-Straße 1 | 85737 Ismaning/München
Telefon: +49 89 96101-0 | Fax: +49 89 96101-1113
www.msg-systems.com | info@msg-systems.com

Stand: September 2012
http://www.msg-systems.com/techrefresh
XT-140-1209-1