**Aufgabe 1.**

**Versionsverwaltung**

|  |  |  |
| --- | --- | --- |
| Lokale Versionsverwaltung | Verteilte Versionsverwaltung | Zentrale Versionsverwaltung |
| nur eine einzige Datei versioniert | Ist als Client-Server-System aufgebaut, sodass der Zugriff auf ein Repository(Lager)  auch über  Netzwerk erfolgen kann | verwendet kein zentrales Repository  mehr |
| wurde mit Werkzeugen wie SCCS und RCS umgesetzt | Durch eine Rechteverwaltung nur berechtigte  Personen neue Versionen in das Archiv legen können | Jeder, der an dem verwalteten Projekt arbeitet, hat sein eigenes Repository und kann dieses mit jedem beliebigen anderen Repository abgleichen |
| die Versionen eines Dokumentes in der Datei des Dokuments selbst  speichern | **Einfache Wiederherstellung von Dateien, da jeder eine vollständige Versionshistorie bekommt** | Änderungen können lokal verfolgt werden, ohne eine Verbindung zu einem  Server aufbauen zu müssen. |
|  | Keine gleichzeitige Zusammenarbeit | Es kommt nicht zu einem Konflikt, wenn mehrere Benutzer dieselbe Version einer Datei ändern. Die sich widersprechenden Versionen existieren  zunächst parallel und können weiter geändert werden |
|  |  | Sie können später in eine neue Version zusammengeführt werden. Dadurch entsteht ein [gerichteter azyklischer Graph](https://de.wikipedia.org/wiki/Graph_(Graphentheorie)#Zyklisch/azyklisch) ([Polyhierarchie](https://de.wikipedia.org/wiki/Polyhierarchie)) |
|  |  | in der  Softwareentwicklung meist einzelne Features oder Gruppen von Features in separaten Versionen entwickelt und diese bei größeren Projekten von Personen mit einer Integrator-Rolle überprüft und zusammengeführt. |

Zentrale Versionsverwaltung

**Vorteile**

* Zugriff auf ein Repository auch über

Netzwerk erfolgen kann

* nur berechtigte

Personen neue Versionen in das Archiv legen können

* Parallel Betrieb ist möglich
* Prozesse können überarbeitet und auf neue Technologie angepasst werden
* Hohe Betriebssicherheit, da die Systeme nur während der Migrationsphase nicht zur Verfügung stehen

**Nachteile**

* Aufwändiger Ansatz, da parallel zwei Systeme betrieben werden
* Neben dem Aufbau des neuen Systems muss auch migriert werden

Verteilte Versionsverwaltung

**Vorteile**

* der Zugriff auf ein Repository(Lager)auch über Netzwerk erfolgen kann
* Sie erhalten stattdessen eine vollständige Kopie des Repositorys. Auf diese Weise kann, wenn ein Server beschädigt wird, jedes beliebige Repository von jedem beliebigen Anwenderrechner zurückkopiert werden und der Server so wiederhergestellt werden. Jede Kopie, ein sogenannter Klon (engl. clone), ist ein vollständiges Backup der gesamten Projektdaten

**Nachteile**

Lokale Versionsverwaltung

**Vorteile**

* die Versionen eines Dokumentes in der Datei des Dokuments selbst speichern

**Nachteile**

* wird oft nur eine einzige Datei versioniert

**Fremdwörter**

Repository=Lager,Depot