



**TECHNISCHE
UNIVERSITÄT
WIEN**
Vienna University of Technology

Diplomarbeit

**Versionierung im Zuge der Konzept- & Entwicklungsphase anhand eines
AutoCad Plug-Ins**

**ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Diplom-Ingenieurs / Diplom-Ingenieurin
unter der Leitung von**

Dipl.-Ing. Dr. Gabriel Wurzer

259

Institut für Architekturwissenschaften

eingereicht an der Technischen Universität Wien

Fakultät für Architektur und Raumplanung

von

Josef Öhreleder

0125794

Wien, September 2015

Abstract

Die Zusammenarbeit und Kooperation der unterschiedlichsten Akteure in der Architekturplanung wird von immer entscheidender Bedeutung, dafür braucht es entsprechende technische Hilfsmittel. Andere Wirtschaftsbereiche, welche auch durch planende Tätigkeiten geprägt sind, haben diesen Entwicklungsschritt bereits vollzogen und IT basierte Versionskontrollsysteme für sich nutzbar gemacht. Die Untersuchung, wie diese Systeme auch in der Architektur Anwendung finden und in deren Arbeitsweisen integriert werden können, ist Ziel dieser Arbeit. Durch die Analyse der Funktionsweise von Versionskontrollsystemen und der Betrachtung der etablierten Arbeitsweise in der Architektur, anhand eines exemplarischen Beispiels, werden die Grundlagen für die Entwicklung eines Tools geschaffen, welches die Zusammenarbeit zwischen Planern unterstützen und die funktionalen Möglichkeiten schon vorhandener Lösungsansätze erweitern kann. Mit Hilfe von Probanden wurde das Potential des entwickelten Tools, in Bezug auf seine Fähigkeit die Zusammenarbeit und gleichzeitige Bearbeitung von Plänen zu fördern, anhand eines Testprojektes überprüft. Durch die Beobachtung der Probanden während des Tests und der Analyse der Ergebnisse konnte festgestellt werden, dass Versionskontrollsysteme klare Vorteile für die Architektur bereitstellen können. Durch die Erkenntnisse aus dem Test werden erweiterte Funktionen, die sich während der Testphase als nützliche mögliche Verbesserungen herausgestellt haben, für die Arbeit mit einem Plug-In erarbeitet.

The collaboration and cooperation of various players in the architectural process is more and more of vital importance. For that it needs appropriate technical aids. Other industrial sectors, which are also influenced by planning activities, have already managed to go one step further than architecture and made IT based version control systems available for themselves. The investigation of how these systems could be used in architecture and how they can be integrated in the existing workflows, is the aim of this work. Through the analysis of the functionality with version control systems provide and the still prevailing workflow, based on a sample, the foundations for the development of a tool are created, which should support cooperation between different players and expand the possibilities in already existing solutions. With the help of volunteers, the potential of the developed tools to promote cooperation and simultaneous working, has been verified on a test project. By observing the subjects during the conduct and analysis of the result, it was found that the use of version control systems can provide advantages for architecture. From the experience made in the test advanced functions, which presented them self in the test phase as potential improvements, for the plug-in are presented and discussed.

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, welche mich bei der Erstellung dieser Arbeit immer beiseite gestanden sind. Besonderer Dank gebührt hierbei meinem Betreuer DI Dr. Gabriel Wurzer, welcher mich von Anfang an unterstützte und mir jederzeit mit neuen geistigen Anstößen zur Entwicklung meiner Arbeit verhalf.

Allen Korrekturlesern insbesondere meinem Bruder Andreas Öhreneder und meinem guten Freund Reinhard Egelkraut möchte ich für ihre Unterstützung und ihre konstruktive Kritik an meiner Arbeit danken. Dem Team aus Andrea Gödel, Florentina Gorschek und Jakob Rastl, welche sich netterweise bereitklärten meine Versuchskaninchen zu sein, sei hier ebenfalls ein spezieller Dank ausgesprochen.

Meiner geliebten Frau Lourdes, meiner Mutter und meiner Familie möchte ich an dieser Stelle für ihre andauernde Unterstützung während meiner gesamten Studienzzeit ebenfalls danken, ohne sie wäre ich nicht in der Lage meine universitäre Laufbahn mit dieser Arbeit zu einem vorläufigen Ende zu bringen.

Inhaltsverzeichnis

Abstract.....	3
Danksagung.....	5
1 Einleitung.....	11
2 Related Work.....	13
3 Engineering Document Management Systems.....	16
4 Versionskontrolle.....	18
4.1 Entstehung der Versionskontrolle.....	18
4.2 Unterteilung der einzelnen Systeme.....	19
4.2.1 Zentralisiert oder Dezentralisiert.....	20
4.2.2 Die Konfliktlösung.....	21
4.2.3 Momentaufnahme oder Änderungsaufzeichnung.....	23
4.3 Unterschiedliche Versionskontrollsysteme.....	25
4.3.1 RCS (Revision Control System).....	25
4.3.2 CVS (Concurrent Versions System).....	27
4.3.3 SVN (Apache Subversion).....	28
4.3.4 GIT.....	30
4.4 Voraussetzungen für diese Arbeit.....	32
4.4.1 Stabilität.....	32
4.4.2 Sicherheit / Zugänglichkeit.....	33
4.4.3 Wartung.....	34
4.4.4 Bedienbarkeit.....	34
4.4.5 unterstützte Daten.....	34
4.4.6 Anpassbarkeit und Erweiterbarkeit.....	35
4.5 Das gewählte Versionskontrollsystem.....	36
5 Die momentane Situation oder der „Status Quo“.....	38
5.1 Strukturen für eine nachvollziehbare Planung.....	39
5.1.1 Layerstruktur.....	39
5.1.2 analoge und digitale Ablagesysteme.....	40
5.1.3 Vorlagen für den Schriftverkehr.....	41
5.1.4 personelle Strukturen.....	42
5.2 Der Büroalltag.....	42
5.2.1 Initialisierung von neuen Projekten.....	42

5.2.2 Aufgabenstellung.....	43
5.2.3 Ressourcenplanung.....	43
5.2.4 Verteilung der Planungsaufgaben.....	44
5.2.5 Bearbeitung.....	44
5.2.6 Kommunikation.....	45
5.2.7 Überprüfung.....	46
5.2.8 Publizieren.....	47
5.2.9 Projektende.....	48
5.3 Fazit der aktuellen Arbeitsweise.....	49
6 Arbeiten mit Versionen.....	52
6.1 Die Akteure.....	52
6.2 Die Planungsarbeit revolutionieren.....	53
6.3 Von der Arbeit der Administratoren.....	54
6.3.1 Die Vorlagen.....	55
6.3.2 Benutzerverwaltung.....	56
6.3.3 Projektinitialisierung.....	57
6.4 Die Arbeitsschritte während eines Projektes.....	59
6.4.1 Öffnen eines Projektes.....	60
6.4.2 Änderungen überprüfen.....	63
6.4.3 Bearbeitungsphase.....	65
6.4.4 Bearbeitung abschließen.....	67
6.4.5 Daten publizieren.....	69
7 Bearbeitung während eines Beispielprojektes.....	71
7.1 Hypothese – Welche Ergebnisse werden erwartet.....	71
7.2 Parameter für den Test.....	73
7.2.1 Testteilnehmer.....	74
7.2.2 Aufgabenstellung.....	74
7.2.3 mögliche Arbeitsteilung durch Versionierung.....	75
7.2.4 Testablauf.....	76
7.3 Beobachtungen während der Bearbeitung.....	77
7.3.1 Entwurfsphase.....	79
7.3.2 Ausarbeitungsphase.....	81
7.3.3 Kommunikation auf einer neuen Ebene.....	84
7.4 Kommentare der Testteilnehmer.....	85
7.5 Überprüfung der Hypothese.....	87
8 Visionen für die Zukunft.....	89

8.1 Darstellung der Historie eines Projektes.....	90
8.2 parallele Entwicklung.....	91
8.3 erweitern und zusammenführen paralleler Vorgänge.....	93
8.4 unterschiedliche Projekte miteinander verbinden.....	94
8.5 Freigabeprozesse.....	96
8.6 erweiterte Synchronisation während der Bearbeitung.....	97
8.7 Verteilung und Management von Aufgaben.....	98
8.8 Versionierung für das gesamte Projekt.....	99
9 Fazit.....	102
9.1 Eine neue Art zu entwerfen?.....	102
9.2 Einsetzbarkeit in der Realwirtschaft.....	104
9.3 Vorteile bei der Arbeit.....	105
9.4 Einschränkungen bei der Arbeit.....	108
9.5 Ausblick in eine Zukunft mit Versionskontrolle.....	110
Literaturverzeichnis.....	112
Abbildungsverzeichnis.....	114

1 Einleitung

Wie jedes andere Berufsfeld muss auch die Architektur sich den Herausforderungen der partnerschaftlichen Zusammenarbeit stellen. Nicht nur durch die immer größere Vernetzung von Arbeitsprozessen, sondern auch durch die Herausforderungen des digitalen Zeitalters muss sich die Architekturarbeit von einer auf einzelne Akteure abgestimmten Arbeitsweise hin zu einer Zusammenarbeit aller Akteure entwickeln.

Speziell bei der Erstellung der architektonischen Darstellungen wie zum Beispiel Grundrisse, Schnitte, Details und anderen steht die Architekturarbeit erst am Anfang eines langen Weges zur effektiven Zusammenarbeit mit mehreren Beteiligten. Im Maschinenbau, in der Automobilindustrie und im Flugzeugbau werden seit Jahrzehnten Programme verwendet, welche es den einzelnen Akteuren im Planungsprozess erlauben, direkt über die zeichnerische Arbeit miteinander im planerischen Dialog zu stehen. Solche Vorgehensweisen erlauben eine orts- und zeitunabhängige Arbeitsweise. So ist es verschiedenen Akteuren an verschiedenen Orten und zu unterschiedlichen Zeitpunkten in der Projektzeit möglich alle mit dem Projekt verbundene Daten zu sehen, zu bearbeiten oder neu zu strukturieren.

Programme für Architekten bieten hierbei noch unzureichende beziehungsweise eingeschränkte Möglichkeiten. Dabei sind entweder die Abläufe zur Zusammenarbeit mehrerer Akteure noch unterentwickelt und nur durch erhöhten Organisationsaufwand umsetzbar. Andererseits ist eine Zusammenarbeit zwischen Akteuren an unterschiedlichen Standorten bisweilen nur durch unternehmenseigene Serversysteme oder sogenannte „Cloudlösungen“ sichergestellt. In beiden Systemen kommen zwar Backupsysteme zum Einsatz, jedoch kann ein Backup keinen genauen Änderungsverlauf aufzeichnen, sondern nur in einem definierten Intervall die Änderungen der Dateien in einem vorab definiertem System aufzeichnen.

In der folgenden Arbeit soll ein Weg aufgezeigt werden, welcher versucht, durch die Verwendung eines Versionskontrollsystems, die Abläufe während der Arbeit mit mehreren Beteiligten zu automatisieren beziehungsweise diese für die Anwender zu vereinfachen. Es soll gezeigt werden, dass durch die Verwendung von Versionskontrolle eine strukturierte und flüssigere Arbeit in der Architekturplanung möglich ist. Nicht nur werden alle Schritte durch ein solches System aufgezeichnet, es ermöglicht auch die

parallele Arbeit verschiedener Subbereiche miteinander, da jedem Beteiligten, zu jedem Zeitpunkt, der aktuellste fertig bearbeitete Stand jedes Teilbereiches der Arbeit an einem Projekt zur Verfügung gestellt werden kann. Durch die Aufzeichnung der einzelnen Schritte ist es möglich, die Entwicklungsgeschichte eines Projekts darstellen und rückverfolgen zu können. Bis dato ist dies über Backupsysteme nur bedingt möglich, welche immer nur einen Schnappschuss der Inhalte eines Projektes zu einem gewissen Zeitpunkt wiedergeben können. Ein Versionskontrollsystem kann im Gegensatz dazu den gesamten Verlauf der Änderungen über den gesamten Projektzeitraum wiedergeben. Dabei muss aber noch angemerkt werden, dass ein Versionskontrollsystem nie ein Backup ersetzen kann oder wird, da es nicht redundant funktioniert und somit nicht an verschiedenen Orten unveränderlich gesichert werden kann. Dies ist dem Umstand geschuldet, dass Versionskontrollsysteme nur „live“ funktionieren können. Jedoch ist es möglich einen spezifischen Zeitpunkt auf einem Backupsystem zu sichern und bei Bedarf diesen wiederherzustellen, wobei jedoch die gesamten, nach diesem Zeitpunkt entwickelten, Änderungen verloren gehen würden.

In der vorliegenden Arbeit wird zu Beginn ein Einblick in die Funktionsweise von Versionskontrollsystemen gegeben, sowie eine Beschreibung einiger ihrer bekanntesten Vertreter, um dadurch eine Grundlage für die Auswahl eines Systems zur Ausarbeitung einer funktionstüchtigen Programmerweiterung, kurz Plug-In, zu schaffen. Im nächsten Schritt wird die aktuelle Arbeitsweise in einem Architekturbüro untersucht. Dabei soll herausgearbeitet werden wo das Plug-In ansetzen und Verbesserungen beziehungsweise Erleichterungen in der Architekturarbeit anbieten könnte. Nachdem die momentane Situation analysiert wurde, wird das im Zuge dieser Arbeit entwickelte Plug-In und seine Funktionsweise beschrieben. Das Plug-In wird anschließend einem Test durch mehrere Probanden unterzogen. Abschließend werden mögliche Erweiterungen und Weiterentwicklungen vorgestellt, welche die Integration des Plug-Ins in die Architekturarbeit noch weiter vertiefen könnten. In der abschließenden Diskussion soll untersucht werden, in wieweit das Plug-In die erhofften und prognostizierten Fähigkeiten in die Architekturarbeit einbringen kann und mit welchen Vor- und Nachteilen durch einen Einsatz in der Projektarbeit zu rechnen ist.

2 Related Work

Pittioni¹ versucht in seiner Arbeit eine grundlegende Definition zu finden wie Kooperation zwischen unterschiedlichen Planern stattfinden kann und sollte. Dabei propagiert er den Austausch der neuen und aktuellen Daten aller Beteiligten mittels eines Server, welcher die Daten verteilt und verwaltet, so dass jeder Bearbeiter jederzeit und überall Zugriff auf diese haben kann. Um dies umsetzen zu können muss grundsätzlich über die Arbeitsweise der einzelnen Planer reflektiert werden um schlussendlich ein System etablieren zu können, in welchem die Entwicklung der einzelnen Modelle überwacht und die Ergebnisse allen Beteiligten zur Verfügung gestellt werden können. Mahler et.al.² versuchen in ihrer Arbeit eine synchrone

Multi-User Plattform für die Zusammenarbeit mehrerer Planer zu entwickeln. Dabei verwenden sie Versionskontrolle um die nötigen Modelle und Informationen für alle Planer aktuell bereit stellen zu können, zusätzlich schlagen Sie einen Prozess vor, welcher es ermöglicht das mehrere Personen eine Datei gleichzeitig bearbeiten können und diese Änderungen von anderen Beteiligten mitverfolgt werden können.

Im Gegensatz dazu schlagen Jaby und Hall³ in ihrer Arbeit vor, ein System zu entwickeln welches die unterschiedlichen Benutzer während der kooperativen Arbeit in allen Phasen des Entwurfes unterstützen kann. Zu diesem Zweck entwerfen Sie eine Methode um nicht nur Daten zwischen den Benutzern zu verteilen, sondern auch eine gemeinsame Kommunikationsplattform aufzubauen, welche von einem Projektleiter mit der Unterstützung eines Management Programms, welches die aktuelle Arbeit sichert, vor Datenverlust schützt und alle nötigen Daten bereitstellt, organisiert wird. Durch die gemeinsame Erarbeitung von Meilensteinen wird sichergestellt, dass alle Daten schlussendlich harmonisiert werden können. Hauschild und Hübler⁴ versuchen in ihrer Arbeit ein Programm zu entwerfen, welches ein Bauwerk Modell auf mehrere Computer aufzuteilen vermag und mittels eines Servers die Kommunikation zwischen den

1[Past, present and future issues with co-operating, seen from the engineering practice, Gernot Pittioni, 2001, in ACCOLADE - Architecture, Collaboration, Design, Delft University Press (DUP Science), ISBN 90-407-2216-1, S. 83-94, <http://cumincad.architexturez.net/doc/oai-cumincadworks.id-8c32>, Zugriff am 17.03.2015]

2[Synchronous Support and Emergence in Collaborative CAAD, Mary Lou Maher, John S Gero, Milad Saad, in U. Flemming and S. Van Wyk (eds), CAAD Futures 1993, North-Holland, Amsterdam, ISBN 0-444-89922-7, S. 455-470]

3[Beyond the Shared Whiteboard: Issues in Computer Supported Collaborative Design, Wassim M. Jabi, Theodore W. Hall, 1995, in Proceedings of CAAD Futures '95: The Sixth International Conference on Computer-Aided Architectural Design Futures, Singapore, ISBN 9971-62-423-0, S. 719-725, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.4506&rep=rep1&type=pdf>, Zugriff am 12.03.2015]

Teilnehmern sicherstellen kann. Dabei wird mithilfe eines Versionskontrollsystems die Bereitstellung der korrekten und aktuellen Entwurfsdaten sowie die definierten Phasen der Entwicklungen verwaltet und bereitgestellt. Hierfür ist es nötig ein Modell in partizipialen Modellen abbilden zu können und der deren Bearbeitung und Anzeige über Berechtigungen zu steuern, um eine geregelte Kooperation sicher zu stellen. Jeng⁵ versucht in seiner Arbeit mittels eines Web-basierenden Prozess-Tools noch einen Schritt weiter zu gehen, um die Komplexität der Koordinationsarbeit und der Zusammenarbeit weiter zu verringern. Dafür schlägt er ein System mit drei Ebenen Benutzerinterface, Server und Datenbank vor. Die Datenbank, welche alle Daten bereitstellt, wird durch einen Server erweitert, welcher diese auf die Nutzer verteilen kann. Das Benutzerinterface zeichnet die signifikanten Events auf und koordiniert beziehungsweise sendet diese an den Server, welcher die Informationen an alle Beteiligten verteilt. Pelton et al.⁶ versuchen ihre Arbeit eine Plattform zu entwickeln, welche über eine zentrale Versionskontrolle die jeweils benötigten Daten und Informationen an die verschiedenen Bearbeiter verteilt. Dabei stellen sie ein Datenmodell vor, mit welchem sie unterschiedlichste Systeme bedienen können. Die Plattform stellt jedoch im Gegensatz zum Vorschlag von Jeng sowie Hauschild und Hübler lediglich die benötigten Daten für jeden Nutzer zur Verfügung und sammelt diese wieder ein, um sie an die Versionskontrolle zurückzugeben.

Schneider et al.⁷ stellen in ihrer Arbeit fest, dass die Integration von unterschiedlichen Aspekten und Interessen der verschiedenen Beteiligten schon zu einem sehr frühen Zeitpunkt der Planung erfolgen muss. Hierfür untersuchen Sie aktuelle Building Information Modelling Konzepte. Sie stellen weiters fest, dass die kreative Gruppenarbeit ein hochdynamischer Prozess ist, welcher von derzeitigen Building Information

4[Entwicklung eines verteilbaren und kooperativ nutzbaren objektorientierten CAAD-Produktmodellierkerns, T. Hauschild, R. Hübler, 1997, in Internationales Kolloquium über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen, IKM, 14, 1997, Weimar, Bauhaus-Universität, <http://e-pub.uni-weimar.de/opus4/frontdoor/index/index/docId/447>, Zugriff am 16.03.2015]

5[Towards a Process-Centric, Asynchronous Collaborative Design Environment, Jeng, Tay-Sheng, 2001, in Proceedings of the Fifth Conference on Computer-Aided Architectural Design Research in Asia, 15-24. CAADRIA, National University of Singapore, ISBN 981-04-2491-4, S. 15-24, <http://cumincad.architexturez.net/doc/oai-cumincadworks.id-1a59>, Zugriff am 19.03.2015]

6[An Engineering Document Management System, Hannu Peltonen, Tomi Männistö, Kari Alho, Reijo Sulonen Helsinki University of Technology, 1993, http://www.researchgate.net/publication/2727482_An_Engineering_Document_Management_System Zugriff am 30.04.2015]

7[Design Versioning - Problems and Possible Solutions for the Automatic Management of Distributed Design Processes, Sven Schneider, Jörg Braunes, Torsten Thurow, Reinhard König, 2011, Computer Aided Architectural Design Futures 2011, Liege, Belgien, ISBN 9782874561429, S. 669-681, http://cumincad.scix.net/data/works/att/cf2011_p042.content.pdf, Zugriff am 30.04.2015]

Modelling Konzepten nur schwer wiedergegeben werden kann. Es braucht die Speicherung und Versionierung des gesamten Entwurfsprozesses. Da die aktuellen Konzepte meist lineare Systeme sind, welche kein paralleles arbeiten unterstützen, sondern lediglich die Zusammenarbeit simulieren können und die Darstellung von unterschiedlichen Alternativen nicht umfänglich unterstützen, schlagen Sie neue nichtlineare Zugänge vor. Dazu braucht es ihrer Meinung nach ein Versionierungskonzept, welches so universell wie möglich sein muss. Die gesamte Geschichte eines Models muss in ihr abgebildet werden können. Die Art der Zusammenarbeit die Möglichkeiten von asynchroner und synchroner Arbeit und ein nicht lineares Aufzeichnen der Entwicklung kann ihrer Meinung nach nur durch computergestützte Versionierung der Zusammenarbeit sichergestellt werden. In seiner Arbeit zeigt Wurzer⁸ auf wie Versionierung und im Speziellen die Verwendung von Versionskontrollsystemen einen positiven Einfluss auf die Arbeit aller Projekt Beteiligungen haben kann. Einerseits gibt sie den Planern die Möglichkeit schnell Alternativen zu erzeugen und andererseits kann sie durch die nichtlineare Aufzeichnung aller Varianten und Alternativen den Beteiligten ein Tool in die Hand geben, mit welchem einzelne Alternativen effizient vergleichbar und somit die Auswahl der nächsten Schritte vereinfacht werden. Diese Systeme können auch verwendet werden um mehreren Benutzern ein paralleles arbeiten zu ermöglichen und die fehleranfällige manuelle Versionierung und Verteilung von Daten zu automatisieren.

⁸[Versioning in early-stage hospital simulation, Gabriel Wurzer, 2013, in Proceedings of the International Workshop on Innovative Simulation for Health Care, ISBN 978-88-97999-26-3, S. 65-71]

3 Engineering Document Management Systems

Seit einigen Jahren halten sogenannte Engineering Document Management Systeme⁹ (EDMS) in der Industrie Einzug. Die Entwicklung von EDMS geht auf die Verwendung von Versionskontrollsystemen in industriellen Prozessen zurück. Wie schon in der Einleitung erwähnt, ist vor allem der Maschinenbau in der Entwicklung von Mechanismen zur besseren Zusammenarbeit der Architektur weit voraus. In den von industriellen Herstellungs- und Entwicklungsprozessen geprägten Bereichen der Wirtschaft begann die Integration von Versionskontrollsystemen in die Entwurfsabläufe. Die Entwicklung der EDMS wurde hierbei von den Anwendern vorangetrieben, bis auch die großen CAD Schmieden auf die Entwicklung aufmerksam wurden und die Systeme weiter entwickelten.

Die heute vorhandenen EDMS¹⁰ sind größtenteils kommerzialisiert und auf die einzelnen Basisprogramme der Softwarehersteller zugeschnitten. So verwundert es nicht wenn jeder große Computer-Aided Design (CAD) Hersteller sein eigenes EDMS entwickelt hat und Kooperationen einen Seltenheitswert haben. Zusätzlich zu dieser Produktbindung ist noch festzustellen, dass die momentan am Markt befindlichen EDMS stark auf ihre primären Anwendungsgebiete ausgerichtet sind. So unterstützen sie mittlerweile zu großen Teilen die Prozesse und Abläufe im Maschinenbau und anderen Sparten, welche schon früh auf die Verwendung und Integration von Versionskontrollsystemen in ihren Abläufen setzten. Dies hat den Effekt, dass andere Sparten, welche sich nun für die Verwendung solcher Systeme interessieren, sich anpassen müssen, um ein solches System auch effizient nutzen zu können.

Wie auch in der Architektur wird im Maschinenbau mit Versionen bei Plänen gearbeitet, jedoch wird nicht in alternativen sondern in linearen Entwicklungen gedacht. In der Architektur wird dagegen oft in Alternativen gedacht, viele Möglichkeiten vereinen sich zu einem Endergebnis. Die lineare Ausrichtung von Prozessen stellt sich bei EDMS dadurch dar, dass sie dem Anwender das Zurückkehren zu alten Zeichnungsständen

9[An Engineering Document Management System, Hannu Peltonen, Tomi Männistö, Kari Alho, Reijo Sulonen Helsinki University of Technology , 1993 , http://www.researchgate.net/publication/2727482_An_Engineering_Document_Management_System Zugriff am 30.04.2015]

10[Einführung eines Product-Lifecycle-Management-gestützten Änderungsprozesses in einem internationalen Großprojekt , Anneke Lühr, Hochschule für angewandte Wissenschaften Hamburg, 2011 , http://edoc.sub.uni-hamburg.de/haw/volltexte/2012/1556/pdf/Luehr_Anneke_110831.pdf Zugriff am 30.04.2015]

erlauben, dies aber unter Umständen nur einige wenige Schritte betrifft und die Bildung von Verzweigungen in Alternativen nicht erlauben. Alle weiter zurückliegenden Ereignisse werden einem Benutzer nicht angezeigt und müssen durch einen Administrator bereitgestellt werden. Bei einer Bearbeitung eines Altstandes wird dieser zur neuesten Version und kann nicht parallel zur Entwicklung geführt werden. EDMS sind mehr darauf fokussiert den momentan letztgültigen Stand der Arbeit zu verarbeiten, als dem Anwender die Möglichkeit zu geben, innerhalb des Projekts Alternativen zu entwickeln und diese wieder in das Hauptprojekt zu integrieren. Die Ideen des transparenten Versionsmanagements und des nachvollziehbaren Entwicklungsablaufes werden bei EDMS sehr gut umgesetzt. Anwenderseitige Adaptierungen sind jedoch nur im Bereich der Systemeinstellungen beziehungsweise in eingeschränkter Form über vom Hersteller bereitgestellte Schnittstellen für das Implementieren von eigenen Plug-Ins möglich.

Wie schon eingangs angedeutet, basieren EDMS im weitesten Sinne auf Versionskontrollsystemen, welche als deren Basis fungieren um darauf aufbauend die Verwaltung der einzelnen Dateien durchführen zu können. Allen EDMS ist dabei gleich, dass sie die neuesten Versionen von Dateien für jeden Bearbeiter zur Verfügung stellen und die für eine konfliktfreie Bearbeitung die nötigen Bearbeitungsrechte verwalten. Diese Funktionen sind jedoch nur ein geringer Teil der Möglichkeiten welche durch Versionskontrollsysteme bereitgestellt werden. Da die Systeme jedoch nicht als Open Source geführt werden, ist es schwer Aussagen über die genauen Abläufe in den Programmen zu machen, oder neue Funktionen zu definieren. Dies hat zur Folge, dass eine, wie schon erwähnt schwierige, Adaptierung auf andere Einsatzbereiche zusätzlich erschwert wird. Um eine Untersuchung der genauen Funktionsweise und der damit einhergehenden Anwendungsfreundlichkeit, sowie eine Weiterentwicklung durchführen zu können, muss ein neues System entwickelt konstruiert, welchem zumindest alle Funktionen eines EDMS zu eigen sind. Durch diese Neuarbeitung wird es erst möglich die Vorgänge zu verstehen und das noch schlummernde Potential der verwendeten Versionskontrollsysteme nutzbar zu machen. Aus diesem Grund wird in dieser Arbeit darauf verzichtet ein schon bestehendes System adaptieren zu wollen. Vielmehr soll versucht werden, dass ein grundsätzliches Verständnis für die Funktionsweisen aufgebaut wird, um dieses an eine Anwendung für die Architekturplanung anzupassen und neue Möglichkeiten zu erschließen.

4 Versionskontrolle

Im Folgenden Kapitel sollen die Grundlagen der Versionskontrolle näher beschrieben und betrachtet werden. Die Begrifflichkeiten in diesen Systemen, deren spezifische Funktionsweisen, sowie warum und weshalb ein System für die weitere Arbeit Verwendung gefunden hat sollen hier beschrieben werden.

Dabei ist die Frage: „Warum braucht es Versionskontrolle?“ eine der ersten Fragen überhaupt. Um diese grundlegende Frage zu beantworten, muss man sich kurz mit der Entstehung von Versionskontrollsystemen auseinandersetzen.

4.1 Entstehung der Versionskontrolle

Die organisierte Ablage einzelner Schritte in der Produktentwicklung ist in allen Arbeitsbereichen von Bedeutung. Durch diese organisatorische Aufgabe entwickelten sich vorerst analoge und manuelle Vorgangsweisen. Wie zum Beispiel die einfache hierarchische Ordnung aller Schritte in Papierordnern in einem Archiv. Als sich der Computer immer mehr durchsetzte, wurden die vormals analogen Systeme für die digitale Versionierung übernommen und immer mehr verfeinert.

Die ersten Versionskontrollsysteme (Version Control Systems, kurz VCS) wurden als unterstützende Technologie für Programmierarbeiten geschaffen. Bis heute finden VCS in der Informatik ihren größten Anwendungsbereich. Seit der zweiten Entwicklungsstufe, bei welcher die Systeme ausgefeilter und netzwerktauglich wurden, finden sie aber auch immer mehr Anwendung in anderen Aufgabenbereichen, da die Programme mittlerweile nicht nur textbasierte Dateien, sondern auch Dateien die nicht durch einen Texteditor lesbar sind, also Bilder, Vektorgrafiken, u.a., unterstützen.

Die Wichtigsten Eigenschaften wurden schon bei den ersten Systemen eingeführt und werden seit diesem Zeitpunkt immer weiter verfeinert, diese sind:

- *Reversibilität*

Es kann immer auf eine ältere Revision zurückgegriffen werden, sei dies aus Gründen der Fehlerbehebung, oder um Änderungen an einer oder mehrerer Dateien rückgängig zu machen. Auch können dadurch verworfene Entwicklungsideen wieder revitalisiert werden.

- *die Möglichkeit Daten mit Anmerkungen zu versehen*

Jede Revision kann mit Anmerkungen über die Änderungen an Daten und der Absichten hinter dieser versehen werden. Diese Anmerkungen sind eine zusätzliche Hilfe um die Entwicklung verfolgen zu können.

- *Aufzeichnung der Benutzerdaten zu jeder Revision*

Zusätzlich wird mit jeder Revision nicht nur die Änderungszeit sondern auch der verantwortliche Benutzer gespeichert. Mit diesen Informationen und der Änderungsevidenz können Bearbeiter schnell und unkompliziert mit dem Verantwortlichen einer Änderung in Kontakt treten, sollte dies nötig sein.

Jedes VCS können grundsätzlich in 3 Funktionsbereiche¹¹ unterteilt werden:

- Daten aus dem Versionsarchiv hohlen (check out)
- bearbeitete Daten wieder an das Archiv senden (check in oder commit)
- Daten aus einem Archiv kopieren

Alle Befehle, welche von einem VCS ausgeführt werden, können auf diese drei grundlegenden Befehle zurückgeführt werden.

Eines der Alleinstellungsmerkmale von VCS gegenüber anderen Programmen zur Datenverwaltung ist sicherlich, dass diese schon seit den frühesten Entwicklungsstufen die Anlage für Verzweigungen besitzen. Das heißt Änderungen an Dateien müssen nicht linear(1, 2, 3, 4, 5, ...) sein, sondern können sich zu „Was wäre wenn?“ Ästen verzweigen. Eine solche Verästelung hilft um alternative Lösungsstrategien durchspielen zu können und gleichzeitig die Hauptstrategie unberührt weiter verfolgen zu können. In weiterer Folge können diese Äste wieder in den Hauptast integriert, oder stillgelegt werden. Wird ein solcher Ast stillgelegt, heißt das aber nicht, dass er verloren wäre, denn er kann zu jeder beliebigen Zeit im Projekt wiederaufgenommen werden, oder nur zu Teilen in den Hauptast integriert werden.

4.2 Unterteilung der einzelnen Systeme

Die einzelnen VCS sind zwar von ihren Grundfunktionen gleich aufgebaut, sie unterscheiden sich dennoch grundlegend durch die Umsetzung und die Konzepte¹²,

¹¹[Understanding Version-Control Systems (DRAFT), Eric Raymond, 2008, <http://www.catb.org/~esr/writings/version-control/version-control.html>, Zugriff am 05.05.2015], Why use version Ccontrol?

¹² [Understanding Version-Control Systems (DRAFT), Eric Raymond, 2008, <http://www.catb.org/~esr/writings/version-control/version-control.html>, Zugriff am 05.05.2015]

welche den einzelnen Lösungen zu Grunde liegen. Jedes System hat dabei spezifische Eigenheiten und auch Gemeinsamkeiten im Vergleich zu seinen Vorgängern. Die Prämisse der Weiterentwicklungen ist dabei immer „Besser, Schneller und Sicherer“. So wirken manche Eigenschaften zwar ähnlich, entstanden aber aus unterschiedlichen Überlegungen und versuchen damit auch andere Ziele zu erreichen. Um ein VCS charakteristisch beschreiben zu können, muss man die drei wichtigsten Unterschiede gemeinsam betrachten. Nämlich:

- Wo wird gespeichert?
- Wie werden Konflikte gelöst?
- Was wird gespeichert?

4.2.1 Zentralisiert oder Dezentralisiert

VCS der ersten und der zweiten Generation sind um ein zentralisiertes Archiv angeordnet. Hierbei greifen alle Protagonisten auf ein Archiv zu und beziehen daraus ihre Daten um Änderungen daran zu erstellen und diese dann dem Archiv wieder zurückzugeben. Da hierbei das Archiv der Schnittpunkt aller Aktionen der Beteiligten ist, ist es auch fehleranfällig, da ein Fehler im Archiv sich auf alle Arbeitskopien auswirkt. Sollte das Archiv aus unterschiedlichen Gründen nicht reparierbar sein, gelangt auch die Arbeit zu einem abrupten Stillstand. Bei VCS mit zentralisiertem Archiv ist es besonders wichtig ein gutes und verlässliches Backupsystem zusätzlich im Hintergrund zu betreiben. Im optimalen Fall besteht ein solches Backup aus mehreren Spiegelungen des Archivs zu unterschiedlichen Zeitabschnitten, damit ein zerstörtes Archiv auch sicher wiederbelebt werden kann. Zusätzlich muss der Bearbeiter immer mit dem Archiv verbunden sein um neue Dateien beziehen oder zurückgeben zu können. In der ersten Generation musste der Benutzer gar auf dem selben Rechner arbeiten, während ab der zweiten Generation mit Archivservern gearbeitet werden konnte.

Ein dezentralisiertes System hingegen kann ohne die Verbindung zum Archivserver weiterbearbeitet werden. In einem solchen System besitzt jeder Bearbeiter eine komplette Kopie zur Version X des Archivs und ist somit unabhängig von der Konnektivität zu seinen Mitarbeitern. Dadurch dass jede beteiligte Person eine komplette Kopie des Archivs besitzt, ist er sogleich auch ein Backup für das gesamte Projekt. Dezentrale Systeme trifft man ab der dritten Generation an. Sie verbinden das lokale Versionsarchiv der ersten Generation mit dem auf Servern hinterlegten

Versionsarchiv der zweiten Generation. Durch den lokalen Ansatz wurde ein Geschwindigkeitsvorteil geschaffen, welcher sich aber bei der Zusammenführung der dezentralen Archive wieder relativiert, da dies eine wesentlich komplexere Konfliktlösung als bei zentralen Systemen erfordert.

4.2.2 Die Konfliktlösung

Jedes VCS verfügt über mindestens eine Art der Konfliktlösung, wenn zwei Personen die selbe Datei ansprechen und verändern wollen, da durch unkontrollierte Änderungen durch mehrere Bearbeiter die Änderungen der Vorgänger zunichte gemacht werden können. Um diesem Zustand, in welchem Änderungen verloren gehen würden, entgegenzuwirken, findet man in allen VCS mindestens einen von drei Mechanismen.

- *Locking – Bearbeitung für andere Benutzer sperren*

Bei dieser Methode sind die Dateien im Versionsarchiv als lesbar und nicht veränderbar angelegt. Will ein Bearbeiter eine Datei verändern, muss er dies dem VCS bekanntgeben, welches dann den „Lock“ an den Bearbeiter vergibt. Hat ein Bearbeiter einen Lock erhalten, hat er die exklusive Möglichkeit die betreffende Datei zu bearbeiten. Dies bedeutet er ist der Einzige mit den nötigen Rechten, um die betreffende Datei zu bearbeiten und eine veränderte Version an das Archiv zurückzugeben. Diese Methode ist vergleichbar mit der Vorgangsweise bei Datenverarbeitungsprogrammen, wie Textverarbeitung, Zeichenprogrammen und ähnlichen. Jedoch erwirbt der erste Benutzer hier die Exklusivrechte an einer Datei, unabhängig ob er Beobachter oder Bearbeiter ist und nicht wie bei den VCS, wo der Benutzer den Änderungswunsch bekannt geben muss, um diese Rechte zu erwerben. Dabei gleichen jedoch die Abfolge der Bearbeitungsschritten denen von Datenverarbeitungsprogrammen. Erst

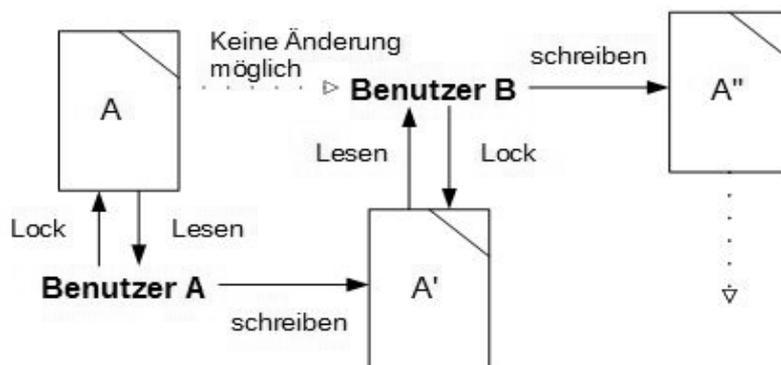


Abb. 1: Verfahren bei der Bearbeiten entsperren Funktion (eigene Abbildung)

durch die Implementierung neuerer Methoden zur Konfliktlösung wurde es möglich parallel eine Datei zu bearbeiten.

Diese Methode ist vorrangig bei den VCS der ersten Generation anzutreffen, nichts desto trotz findet man sie auch als zusätzliche Methode in anderen VCS-Generationen. Speziell wenn nicht textbasierte Dateien bearbeitet werden, ist die Bearbeitung durch den Lock-Modus zu bevorzugen.

- *Merge before commit*

Bei dieser Methode werden Änderungen vor der Übertragung ins Versionsarchiv mit anderen Änderungen vereint. Hierbei ist es möglich, dass mehrere Personen gleichzeitig Änderungen an einer Datei vornehmen. Wenn jemand seine Änderungen an das Versionsarchiv übergeben möchte, wird im Vorfeld überprüft ob seit dem Bezug der Datei Änderungen aufgetreten sind oder nicht. Sollte die Datei im Versionsarchiv seit dem Bezug durch eine neuere Version ersetzt worden sein, wird vom VCS versucht die beiden Versionen miteinander zu verbinden. Ist diese automatische Zusammenführung nicht möglich, wird der Benutzer aufgefordert diese Zusammenführung manuell vorzunehmen. Erst wenn alle manuell zu lösende Konflikte gelöst wurden erlaubt, das VCS die Datei in das Versionsarchiv zurückzugeben.

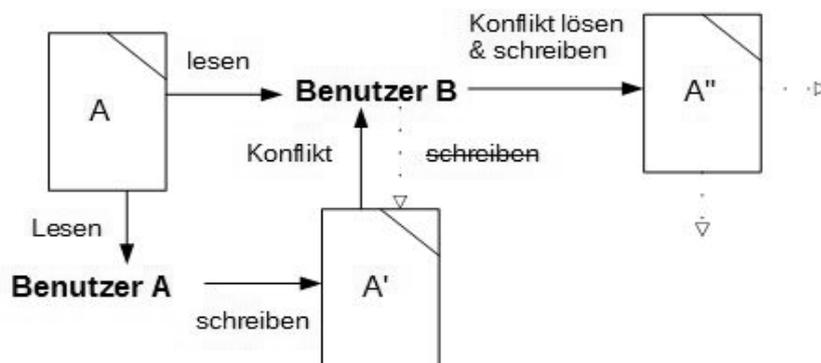


Abb. 2: Verfahren bei der merge before commit Funktion (eigene Abbildung)

Diese Methode findet man ab der zweiten Generation, dabei ist zu beachten, dass diese Funktion ausschließlich für textbasierte Dateien verwendet werden kann.

- *Commit before merge*

Bei dieser Methode werden Übertragungen an das Versionsarchiv generell nicht geblockt. Vielmehr wird ein eigener Zweig erzeugt, sollte die zurückgegebene

Datei, seit deren Bezug, im Versionsarchiv aktualisiert worden sein. Dadurch steht es dem Benutzer frei die dadurch entstandenen Zweige zu verschmelzen oder beide als Alternativen weiterzuführen. Der große Vorteil dieser Methode gegenüber der vorhergehenden ist, dass nicht schon vor Beginn einer Bearbeitung beschlossen werden muss, ob ein neuer Zweig eröffnet werden soll oder nicht.

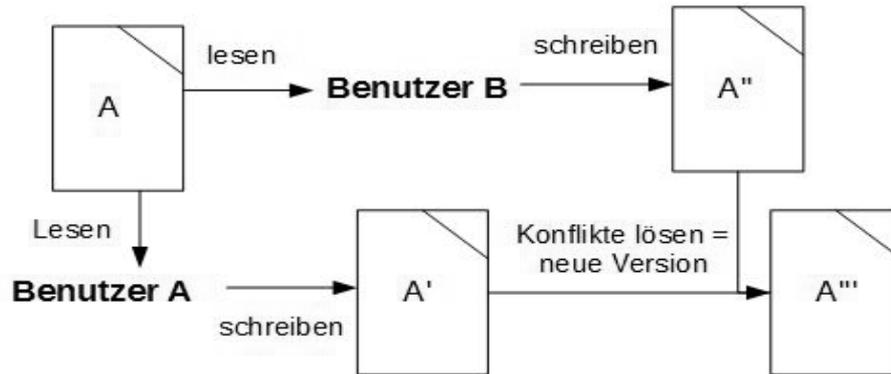


Abb. 3: Verfahren bei der commit before merge Funktion (eigene Abbildung)

Diese Methode trifft man erst in den neueren VCS der dritten Generation an, wie die vorhergehende Methode kann diese ebenfalls nur auf textbasierte Dateien angewendet werden.

4.2.3 Momentaufnahme oder Änderungsaufzeichnung

Bei VCS gibt es zwei Wege die Veränderungen im Projektverlauf darzustellen. Einerseits kann jede Revision als Momentaufnahme stattfinden andererseits als eine Aneinanderreihung von Änderungen aufgezeichnet werden.

In beiden Methoden werden Versionskommentare, Metadaten wie Benutzername und Zeit der Übergabe und andere direkt vor der Aktualisierung der Version an das VCS

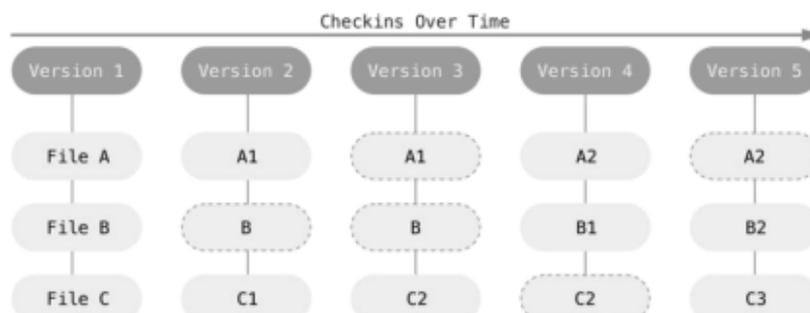


Abb. 4: Versionskontrolle mit Momentaufnahmen aus Pro Git second edition, Scott Chacon & Ben Straub, 2014, Seite 32

übergeben. Daher ist ein Unterschied beider Methoden für den Benutzer kaum ersichtlich.

Bei der Methode der Momentaufnahme werden die Abbilder der Dateien einer Revision gespeichert. Dies bedeutet, dass bei jeder neuen Version ein Abbild der gesamten Dateien der unter Versionskontrolle stehenden Dateien gemacht wird. Wenn eine Datei nicht verändert wurde, wird eine Verknüpfung, mit der zuletzt geänderten Version dieser Datei, anstatt einem Abbild gespeichert. So kann der Datenaufwand bei dieser Methode effizienter gestaltet werden. Ein großer Vorteil dieser Vorgehensweise ist, dass Rücksprünge in vorhergehende Versionen schnell durchgeführt werden können, da die Dateien in dem Zustand der Version eindeutig gespeichert sind.

Verwendet ein VCS eine Änderungsaufzeichnung, werden nicht Abbilder der Dateien aufgezeichnet sondern vielmehr die Unterschiede der einzelnen Dateien von Version zu Version. Dabei wird pro Version der Unterschied zur vorhergehenden gespeichert. Um eine Datei einer bestimmten Version bereitzustellen zu können, wird beginnend bei der

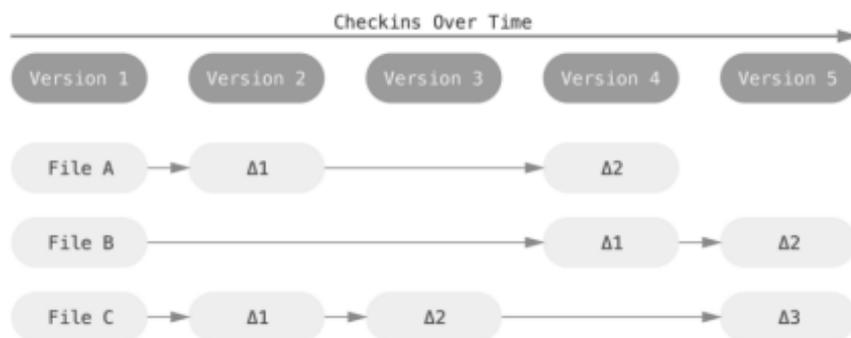


Abb. 5: Versionskontrolle mit Änderungverfolgung aus *Pro Git second edition*, Scott Chacon & Ben Straub, 2014, Seite 32

Version 1 bis zur angefragten Version die Datei zusammengestellt. Es wird bei dieser Methode nur einmal die gesamte Datei gespeichert. Dies geschieht, wenn die Datei dem Versionsarchiv hinzugefügt wird. Fortan werden nur noch die Änderungen gespeichert. Zusätzlich macht es eine Änderungsverfolgung mittels Änderungsaufzeichnung einfacher Namensänderungen, Dateibewegungen in einem Verzeichnisbaum und andere Operationen mit zu protokollieren. Dabei bleibt der Datenaufwand gering, jedoch braucht das VCS auch wesentlich länger eine Datei bereitzustellen.

4.3 Unterschiedliche Versionskontrollsysteme

Im Folgenden sollen kurz die bekanntesten VCS in der Reihenfolge ihrer Entwicklung vorgestellt werden. Die beschriebenen VCS¹³ fallen alle unter die Kategorie „Open-Source“, dies macht sie für diese Arbeit besonders interessant, da sie meist gut dokumentiert sind. Um die Entwicklung bis heute darzustellen, wird ein Programm aus der ersten Generation (RCS eines der ersten VCS überhaupt), zwei aus der zweiten Generation (CVS und Subversion) und die neueste Entwicklung im „Open-Source“-Bereich (GIT) vorgestellt.

4.3.1 RCS (Revision Control System)

Das Revision Control System¹⁴, kurz RCS, wurde in den frühen 1980er Jahren von Walter F. Tichy an der Prude Universität, West Lafayette – Indiana USA, geschrieben. Aus der Dokumentation geht hervor dass die Version 3 1983 veröffentlicht wurde Die Entwicklung davor ist leider nicht dokumentiert.

RCS basiert auf der Locking Methode und ist ein zentralisiertes VCS, welches auf einem lokalen System verwendet wird. Eine Möglichkeit RCS in einem Netzwerk zu verwenden war bei seiner Entwicklung noch nicht vorgesehen. Es war zwar nicht das erste VCS, stellt aber einen der großen Meilensteine der Entwicklung dar. Da es der Vorreiter der Änderungsverfolgung war, bei welcher nur die Unterschiede zur aktuellsten Version aufgezeichnet wurden.

In der ersten Entwicklungsstufe konnte das Programm nur mit textbasierten Dateien interagieren, in späteren Versionen war es zwar schon möglich mit Daten zu arbeiten, welche nicht durch einen Texteditor lesbar sind, jedoch musste dabei ein kompliziertes System an Befehlen ausgeführt werden. Heute findet RCS kaum noch Verwendung. Ein bekannter Anwendungsfall, wo es heute noch Anwendung findet, ist Wikipedia. Die Einflüsse von RCS auf die spätere Entwicklung von VCS sind bis heute in den unterschiedlichen Systemen immer noch ersichtlich. So sind die Grundsätzlichen Befehle, in Abwandlungen und Weiterentwicklungen, sowie das Grundvokabular immer noch in Verwendung. Sogar der Vorgänger von RCS verwendete in späteren Beschreibungen das Vokabular von RCS.

13 [Understanding Version-Control Systems (DRAFT), Eric Raymond, 2008, <http://www.catb.org/~esr/writings/version-control/version-control.html>, Zugriff am 05.05.2015]

14 [RCS—A System for Version Control, Walter F. Tichy Purdue University Department of Computer Sciences, 1985, <http://users.ninthfloor.org/~ashawley/rcs/tichy1985rcs/rcs.html>, Zugriff am 07.05.2015]

Eine weitere Innovation welche durch RCS eingeführt wurde, war die Möglichkeit Verzweigungen in der Versionshistorie einzurichten und diese mit spezifischen Namen zu hinterlegen. Wenn zum Beispiel eine Version 1.3 und 2.1 im Versionsarchiv existiert, kann ein Zweig bei der Version 1.3 mit dem Namen 1.3.1 erstellt werden. Dadurch wird die Möglichkeit geschaffen, Programmadaptationen zu bestimmten Entwicklungsschritten vorzunehmen. Eine Verzweigung an einer solchen Stelle ist deshalb nötig, da keine Version in die Historie eingeschoben werden kann. Durch die Verwendung eines spezifischen Klarnamens konnte diese Verzweigung einfacher gefunden und weiterbearbeitet oder mit der neuesten Version vereint werden.

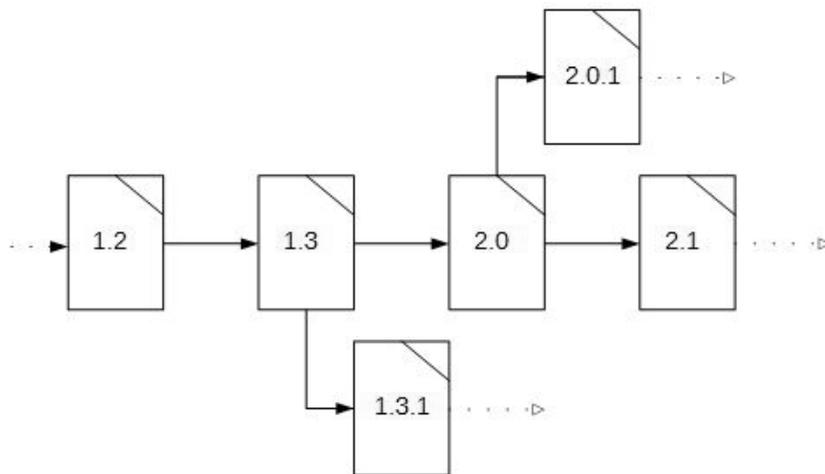


Abb. 6: Versionskontrolle in RCS – einzelne Dateien haben eigene Version (eigene Abbildung)

Um zusätzlichen Speicherbedarf zu vermeiden, werden bei Verzweigungen, nur die Unterschiede zur Ursprungsversion der Verzweigung gespeichert. Um die Datei in der Version 1.3.1 darzustellen, muss die Datei aus der aktuellsten Version (2.1) auf den Ursprung des Zweiges (1.3) zurück transformiert werden und dann die Unterschiede bis zur aktuellen Version (1.3.1) hinzuaddiert werden.

4.3.2 CVS (Concurrent Versions System)

Das Concurrent Versions System¹⁵ (CVS) startete, 1986 durch Dick Grune, vorerst als eine Sammlung von einzelnen Programmcode-teilen (Scripts), welche RCS einfacher und umfassender bedienbar machen sollten. In späterer Folge wurden die einzelnen Scripts durch Brian Berliner und Jeff Polk als eigenständiges Programm weiterentwickelt. In den frühen 90er Jahren des 20. Jahrhunderts wurde CVS schließlich durch den Programmierer Jim Kingdom netzwerktauglich und konnte dadurch auch standortunabhängig verwendet werden. CVS wurde in den folgenden Jahren weltweit zu einem Standardprogramm für Versionierung und Zusammenarbeit.

Die Vorteile von CVS gegenüber RCS waren vor allem die Möglichkeit mehrere Dateien gleichzeitig unter Versionskontrolle zu stellen. Zuvor wurde jede Datei für sich selbst unter Versionskontrolle gestellt. CVS konnte nun aber mehrere Dateien als ein Projekt betrachten und erleichterte somit die logistischen Herausforderungen bei der Entwicklung von komplexeren Projekten. Zusätzlich wurde die Idee des Sperrens, Bearbeitens und Entsperrens bei CVS durch ein automatisiertes Zusammenführungsverfahren ersetzt. Dadurch wurde die Arbeit an einer Datei durch mehrere Bearbeiter erleichtert, da nicht mehr darauf gewartet werden musste bis ein Bearbeiter seine Bearbeitung abgeschlossen hatte, sondern gleichzeitig an einer Datei gearbeitet werden konnte. CVS erledigte dann im Hintergrund die Konfliktlösung, respektive wurde der letzte Bearbeiter aufgefordert, Konflikte manuell zu lösen, welche nicht durch CVS gelöst werden konnten.

Eine große Schwäche von CVS ist, dass es direkt aus RCS hervorging und die einzelnen Dateien in einem Versionsarchiv als eigenständige Einheiten anspricht. Eine Version in CVS besteht aus mehreren Dateien, welche wiederum eigene RCS Versionen besitzen. Dadurch werden Addition, Löschung und Verzweigung sehr sensible Befehle. Besonders bei Verzweigungen ist diese Fragilität erkennbar. So gibt es bei CVS keine Möglichkeit die Konfliktlösung zwischen verschiedenen Zweigen aufzuzeichnen, was dazu führt, dass parallele Zweige nur unter erhöhtem Aufwand gewartet werden können und die Effektivität von Verzweigungen unter CVS fast nicht genutzt werden kann.

Die Einführung der „Merge before commit“ Konfliktlösungsmethode, die Netzwerktauglichkeit und die Umgestaltung des Kommandozeileninterface stellten die

15 [Open Source Development with CVS, 3rd Edition, Karl Fogel, 2003, ISBN 1-932111-81-6, http://cvsbook.red-bean.com/OSDevWithCVS_3E.pdf, Zugriff am 15.05.2015]

Meilensteine der Entwicklungsgeschichte von CVS dar. Das Kommandozeileninterface war so erfolgreich, dass nachfolgende Entwicklungen diese für sich adaptierten oder schlichtweg einfach übernahmen.

Durch die Komplexität der internen Zusammenhänge, welche sich durch das ohnehin schon komplizierte Erbe von RCS zusätzlich verstärkte, wurde CVS nur oberflächlich beziehungsweise nur in unregelmäßigen Abständen weiterentwickelt. Durch diesen Umstand wurde der Code noch unübersichtlicher und in weiterer Folge unverständlicher für die Entwickler. Diese Gründe führten schlussendlich zur Entwicklung des Nachfolgers Subversion.

4.3.3 SVN (Apache Subversion)

Das Credo des Subversion¹⁶ Projekts war, dass es eine überzeugende Alternative zu CVS darstellen sollte. Es sollte besser als seine Vorgänger sein und den Benutzern von CVS ein Umstieg möglichst leicht machen. Das SVN Projekt baut zwar auf den Grundfesten von CVS auf, sein Ziel war und ist es, nicht die selben Fehler und Fehlentwicklungen von CVS zu wiederholen.

Die Entwicklung von SVN begann Anfang 2000 unter der Schirmherrschaft von CollabNet. Das Projekt zog schnell eine große Gemeinde an Entwicklern an. Ein Gründe hierfür war vermutlich die Unzufriedenheit und die schlechten Erfahrungen mit CVS, welche man durch dieses Projekt beheben wollte. Schon nach einem Jahr Entwicklungszeit konnte SVN durch sich selbst weiterentwickelt werden. Dass heißt, CVS wurde durch die erste stabile Version von SVN als Versionskontrolle ersetzt. Seit 2010 ist Subversion Teil der Apache Software Foundation, eine ehrenamtliche Organisation von Entwicklern, welche an Open-Source Projekten arbeiten, sie ist eine Meritokratie, eine Gesellschaft in welcher nur aktive Mitglieder vertreten sind.

SVN übernahm viele Aspekte von CVS, führte aber auch einige Neuerungen ein. Unter anderem belebte es den Locking Modus aus RCS als eine Alternative zum „Merge before commit“ wieder, speziell da SVN Daten, welche nicht durch einen Texteditor lesbar sind, besser und umfänglicher unterstützt als seine Vorgänger, wurde diese Funktion wieder aufgenommen, denn auch weiterhin konnten VCS keine automatische Konfliktlösung für binäre Dateien durchführen. Bis heute kann die Konfliktlösung für

16 [Versionskontrolle mit Subversion: Für Subversion 1.7: (Übersetzt aus der Revision5039), Ben Collins-Sussman, Brian W. Fitzpatrick und C. Michael Pilato, 2011, <http://svnbook.red-bean.com/de/1.7/svn-book.pdf>, Zugriff am 07.05.2015]

binäre Dateien noch von keinem VCS gelöst werden. Eine weitere Neuerung zu seinen Vorgängern ist, dass globale Versionsnummern vergeben werden und nicht mehr jede Datei ihre eigene Versionsnummer hat. Die Versionen in Subversion beginnen bei 0 bis N und sind immer eine positive Ganzzahl. Eine Version stellt somit nicht nur eine Datei oder einen Teilbaum zum Zeitpunkt X dar, sondern das gesamte Archiv zum Zeitpunkt X. In CVS konnte man einer Version noch Klarnamen geben SVN versucht hingegen den Anwender dazu anzuhalten, für spezielle Fälle, zum Beispiel ein Meilenstein, einen neuen Zweig zu erschaffen. Die Vorteile liegen auf der Hand, denn ein neuer Zweig kann weiterentwickelt werden, eine Version mit Klarnamen hingegen ist per se unveränderlich. Da die Versionierung ein fortschreitender Mechanismus ist, kann daran in späterer Zeit keine Änderung mehr vorgenommen werden. Ein weiterer Effekt der globalen Versionsnummern ist, dass immer klar von einer Datei, einer Gruppe von Dateien oder des gesamten Archivs in der Version N gesprochen werden kann, da N immer alle Dateien zu diesem Zeitpunkt verbindet.

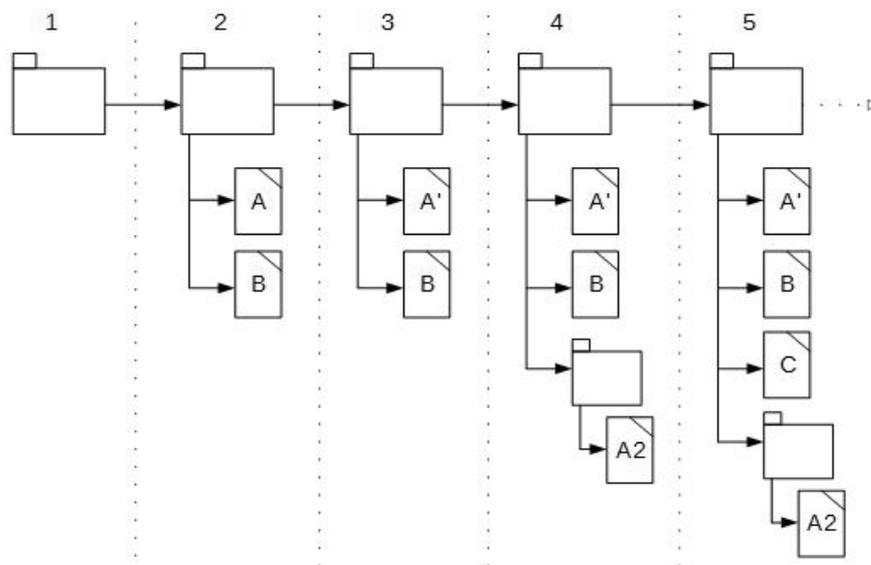


Abb. 7: Versionskontrolle in Subversion – das gesamte Archiv bildet eine Version (eigene Abbildung)

Durch die Einführung der „billigen Kopie“, eine Kopie von Dateien oder Ästen welche auf ihren Ursprung verweist und dadurch sehr effizient gestaltet ist, ist es möglich das vormals in CVS stiefmütterlich behandelte Verzweigen von Entwicklungsschritten wieder einfacher und weniger fehleranfällig zu gestalten. Das gesamte Archiv von SVN baut auf solchen „billigen Kopien“ auf, somit müssen nur noch geänderte Dateien aufgezeichnet werden und unveränderte Dateien verweisen auf ihre letzte Änderung.

Durch die verbesserte Handhabung von binären Dateien, ist SVN eines der ersten VCS, welches auch bei Nichtprogrammierern Verwendung findet. Auch die atomare Verhaltensweise, nur alle Änderungen werden übergeben oder keine, dürfte zur Beliebtheit zusätzlich beigetragen haben. SVN konnte innerhalb von wenigen Jahren CVS fast vollständig verdrängen und platzierte sich anstelle von CVS an der Spitze der zentralisierten VCS. SVN stellt auch den momentanen Endpunkt der Entwicklung der zweiten Generation dar, VCS der dritten Generation versuchen ganz neue Wege im Bereich der Versionskontrolle zu gehen.

4.3.4 GIT

Um die Entwicklung von GIT¹⁷ (engl. Dummkopf) zu verstehen, muss man wissen, dass um das Jahr 2005 eine Kooperation der Entwickler von Linux mit der Firma BitKeeper, einem dezentralisierten VCS der dritten Generation, scheiterte und BitKeeper die „free to use“ Lizenz für die Entwickler von Linux zurückzog. Dadurch waren plötzlich viele der Entwickler vom Projekt ausgeschlossen und es wurde nach Auswegen gesucht, um alle Entwickler wieder am Projekt zu beteiligen. Dazu wollte man ein eigenständiges Programm ohne die Verwendung von Code anderer Programme schaffen, welches vier Haupteigenschaften haben sollte. Schnelligkeit, simples Design, gute Vereinbarkeit mit nichtlinearer Entwicklung und es sollte große Projekte wie Linux bewältigen können.

Man versuchte Gelerntes aus alten VCS zu übernehmen und zu verbessern. GIT unterstützt besonders dezentrales und nicht lineares Arbeiten. Um dies zu bewerkstelligen, verwendet GIT ein ähnliches Verfahren der kontinuierlichen Momentaufnahmen wie SVN und führt fast alle Operationen lokal am Rechner des Benutzers aus. Lediglich um sich final mit allen anderen Bearbeitern abzustimmen braucht es eine Verbindung. Im Gegensatz zu seinen Vorgängern erstellt GIT für jede Version eine Prüfsumme, welche einzigartig ist. Dies erleichtert GIT die Zusammenführung der dezentralen Versionsarchive. Um das dezentrale System von GIT zu ermöglichen, verwendet es nicht nur 2 Instanzen, Archiv und Arbeitskopie, wie seine Vorgänger, sondern es wurde ein Zwischenschritt eingeführt, die sogenannte „Staging Area“. Man kann sich das ganze wie ineinander geschachtelte VCS vorstellen, wobei das gesamte Archiv zum Zeitpunkt X in die Arbeitskopie übertragen wird, in welcher gearbeitet wird. Änderungen werden in die „Staging Area“ übermittelt. Wenn der

17 [Pro Git second edition, Scott Chacon & Ben Straub, 2014, ISBN 978-1484200773, <https://progit2.s3.amazonaws.com/en/2015-05-02-7ddec/progit-en.478.pdf>, Zugriff am 07.05.2015]

Bearbeiter es wünscht, werden die Änderungen an das zentrale Archiv oder einen anderen Bearbeiter übermittelt. Zwischen all diesen Schritten gibt es eine Konfliktkontrolle und die schon beschriebene Hinterlegung der Versionen mit unveränderlichen und einzigartigen Prüfsummen. Somit ist es nicht nur möglich dezentral und ohne Serververbindung sondern auch asynchron mit dem Projekt zu arbeiten.

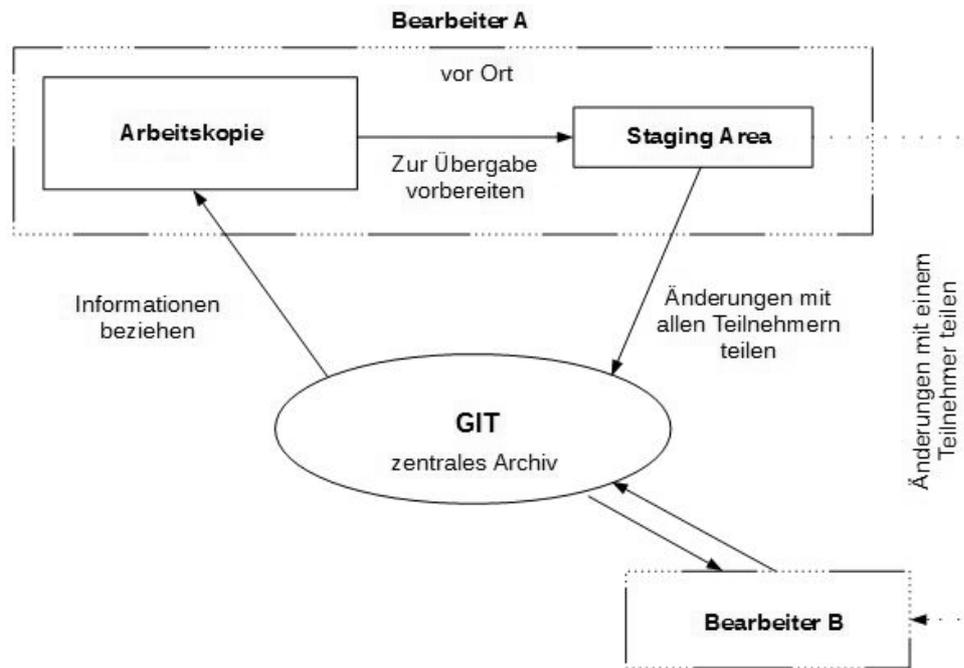


Abb. 8: Versionskontrolle mit GIT – Arbeitsablauf während der Bearbeitung (eigene Abbildung)

Jeder Benutzer ist sogleich auch ein Backup des Gesamtsystems zum Zeitpunkt X, da er das gesamte Abbild bei der Übertragung bekommt. Dies klingt nach einem riesigen Datentransfer, jedoch wird nicht immer das gesamte Projekt übertragen sondern immer nur die geänderten Informationen seit dem letzten Update. Durch die lokale Kopie des Versionsarchivs ist GIT in seiner lokalen Anwendung durch einen Benutzer deutlich schneller als alle seine Vorgänger, lediglich der Abgleich mit dem zentralen Archiv bedarf eines erhöhten Zeitaufwand. Dieser ist in der Regel bei dezentraler und asynchroner Arbeit nicht so häufig nötig, wie bei den zentralen Systemen der zweiten Generation.

GIT wurde grundsätzlich wie seine Vorgänger von Programmierern für Programmierer entwickelt, findet aber auch außerhalb dieser Kreise vermehrt Anwendung. In GIT gibt es durch die Dezentralität des Systems keine Möglichkeit mehr einzelne Dateien für die Bearbeitung zu sperren. Das kann bei binären Dateien zu großen Problemen führen, da

GIT wie auch seine Vorgänger diese Art der Daten nicht zusammenführen kann. Im Extremfall kann dies dazu führen, dass verschiedene Bearbeiter die selbe Datei verändern und später ein anderer Bearbeiter manuell sämtliche Änderungen wieder zusammenführen muss.

4.4 Voraussetzungen für diese Arbeit

Um ein geeignetes VCS für diese Arbeit zu wählen, muss dieses gewisse Eigenschaften aufweisen. Diese sollen im folgenden kurz diskutiert werden. Hierbei soll nicht nur die Eigenschaften der verschiedenen VCS betrachtet werden sondern auch ihre Verwendbarkeit beziehungsweise ihre Anpassbarkeit an die erforderliche Struktur des zu gestaltenden Plug-Ins.

Durch die Diskussion der Voraussetzungen, auch unter Einbeziehung der Voraussetzungen für das Plug-In, wird die Wahl eines Systems und der daraus resultierenden Effekte auf das Plug-In selbst erleichtert. Wie bei jedem Projekt ist es ursächlich, dass das Ausgangsprodukt und die darauf aufbauende Umsetzung gut ineinandergreifen und sich dabei bestmöglich ergänzen. Um dies zu gewährleisten wird auf 6 wichtige Aspekte eingegangen:

- Stabilität
- Sicherheit und Zugänglichkeit
- Wartung Bedienbarkeit
- unterstützte Daten sowie
- Anpassbarkeit und Erweiterbarkeit.

4.4.1 Stabilität

Da das VCS sozusagen das Gehirn des gesamten Planungsprozesses sein wird, ist es wichtig, dass das System stabil und zuverlässig läuft. Dafür braucht es unter anderem einen stabilen Server, eine stabile Verbindung zu den einzelnen Benutzern und ein gutes Backupsystem.

Stabilität bedeutet auch Kontinuität. Es ist zu empfehlen, das gesamte System redundant aufzubauen, um möglichen Ausfällen und äußeren Störungen entgegen wirken zu können.

Das zu wählende VCS darf Änderungen an einem Versionsarchiv nur annehmen, wenn diese vollständig übertragen werden. Da es sonst zu Fehlern kommen kann, welche das

gesamte Archiv gefährden könnten. Zusätzlich muss das System dem Plug-In die Möglichkeit bieten, über die Annahme oder nicht Annahme von Änderungen Informationen einzuholen, um einen erneuten Versuch zu starten, oder den Benutzer über Probleme in Kenntnis setzen zu können.

4.4.2 Sicherheit / Zugänglichkeit

Die Zugänglichkeit des Systems muss einerseits die Möglichkeit bieten offline nur im Firmennetzwerk erreichbar oder online über einen Server erreichbar zu sein. Die Online Variante ist speziell dann interessant, wenn das Archiv von mehreren Firmen, im Zuge einer Projektzusammenarbeit, bearbeitet werden soll.

Da sich das Versionsarchiv außerhalb der Firmennetzwerkes befinden kann, muss das VCS den Zugang durch Benutzernamen und Passwörter sichern können, um Unbefugten den Zugriff und somit die Veränderung im Archiv zu verweigern. Das Plug-In muss die Fähigkeit besitzen, die Kommunikation zwischen dem Versionsarchiv und dem Benutzer automatisiert abspielen zu können.

Die für die Sicherheit des Archivs nötigen Zugangsbeschränkungen werden auch für Arbeitsbeschränkungen, zum Beispiel wer ist Administrator des Archivs beziehungsweise wer darf was zum Archiv beitragen, verwendet. Das bedeutet das VCS oder das Plug-In müssen die Fähigkeit besitzen, unterschiedliche Zugangsberechtigungen parallel verwalten zu können.

Das VCS darf keine Löschung von Daten über das gesamte Archiv zulassen. Die Geschichte des Archivs soll immer konsistent bleiben. Eine Löschung in der Vergangenheit, muss die Bildung eines Astes zur Folge haben, welcher eine „alternative Realität“, in welcher eine Datei gelöscht wurde, erzeugt. Sonst wäre nicht nur die Konsistenz des Versionsarchivs sondern auch die Bearbeitbarkeit durch den Benutzer eingeschränkt. So könnte der Fall auftreten, dass nicht alle Benutzer mit einer Löschung einverstanden sind oder die zu entfernende Datei zu einem späteren Zeitpunkt wiederverwendet werden soll.

4.4.3 Wartung

Für die Administratoren muss die Möglichkeit bestehen auch ohne Benutzeroberfläche die nötigen Arbeiten am System durchführen zu können. Speziell wenn die Benutzeroberfläche ausfallen sollte, muss eine durchgängige Wartbarkeit gegeben bleiben.

Um die ununterbrochene Nutzbarkeit eines VCS zu gewährleisten, muss die Benutzeroberfläche getrennt vom VCS sein. Dies bedeutet, dass das VCS nur durch die Benutzeroberfläche angesteuert wird, jedoch systemkritische Operationen vom VCS selbst ausgeführt werden. Dadurch ist es möglich, dass alle anderen Beteiligten weiterarbeiten können, sollte bei einem Benutzer ein Problem mit dessen Benutzeroberfläche auftreten. Da sämtliche Operationen an das VCS delegiert werden, können, diese im Falle eines Fehlers durch einen Administrator in einen Nebenzweig verschoben werden und somit die Integrität des Hauptastes wiederhergestellt werden.

4.4.4 Bedienbarkeit

Die Bedienbarkeit durch die Benutzer ist stark an die Benutzeroberfläche des VCS und des Plug-Ins gekoppelt und korreliert mit der Einfachheit der einzelnen Befehle. So sollte das VCS über leichtverständliche Befehle verfügen und das Plug-In muss in der Lage sein zwischen Anwender und VCS effizient zu kommunizieren. Je weniger Befehle es gibt, desto besser ist ein Programm erlernbar. Einfache Befehle lassen sich in weiterer Folge zu komplexeren Befehlsketten verbinden. Dadurch wird es einem Anwender erlaubt, neue Abläufe und Anwendungsmöglichkeiten innerhalb der jeweiligen Programmgrenzen zu generieren.

Die wichtigsten und grundlegendsten Befehle, welche ein VCS verstehen und verwalten können muss, sind: neue Informationen hinzufügen, Änderungen empfangen, die neuesten Informationen bereitstellen, gespeicherte Informationen kopieren, verschieben, umbenennen oder löschen. Aus diesen 7 Grundbefehlen lassen sich in weiterer Folge eine Vielzahl von neuen Befehlen, auf Basis der Kombination der einzelnen Befehle, generieren.

4.4.5 unterstützte Daten

Da das Plug-In in der Basisversion Zeichnungs- und Layoutdateien verarbeiten soll, muss ein VCS, welches dem Plug-In zu Grunde liegt, binäre Dateien unter

Versionskontrolle stellen können. VCS können jedoch keine Unterschiede von binären Dateien darstellen, sowie keine selbstständige Konfliktlösung durchführen, daher müssen sie einen Mechanismus zur Bearbeitungssperre besitzen, um die Gefahr eines Konflikts während der Zusammenführung von gleichzeitigen Änderungen an einer binären Datei zu unterbinden.

Das Plug-In hingegen wird mit der Fähigkeit ausgestattet Unterschiede zwischen der zu öffnenden Datei und der zuletzt, durch den betreffenden Anwender, geöffneten Version darzustellen. Dabei soll das Plug-In aber nicht das Alleinige Kommunikationsmittel mit dem VCS bleiben. Es ist denkbar, dass nicht nur Zeichnungen und Skizzen in Zukunft unter Versionskontrolle gestellt werden, sondern ein Projekt in seiner Gesamtheit durch ein VCS kontrolliert werden wird. Das Plug-In wird zwar in einer Ersten Phase auf eine CAAD Software abgestimmt, um die Fragestellung nach dem optimalen Arbeitsablauf besser klären zu können. Denkbar ist, in weiterer Folge, eine Ausweitung des gesamten Projektes auf alle Bereiche der Planungsarbeit, um somit eine geschichtliche Darstellung des gesamten Projektes und all seiner Abläufe darstellen zu können.

Der ursächliche Grund warum sich diese Arbeit primär mit der Versionierung der Entwurfs- und Zeichenarbeit während eines Projektes beschäftigt ist, dass dieser Teil der Arbeit in einem Projekt, wohl durch die Variabilität von Entwicklungsschritten der komplexeste Teilaspekt für die Verwendung von Versionskontrollsystemen in der Architekturarbeit ist.

4.4.6 Anpassbarkeit und Erweiterbarkeit

Eine mögliche Erweiterbarkeit des Plug-Ins sowie die Möglichkeit dadurch das Plug-In an die jeweiligen Anforderungen verschiedener Benutzer anzupassen ist ebenso anzustreben, wie die Benutzerfreundlichkeit und die Unterstützung verschiedener Datensätze.

Um diese Erweiterbarkeit zu unterstützen, muss ein VCS so offen wie möglich gestaltet sein. Dies bedeutet, dass das System selbst angepasst werden kann, damit sich Anwendungsbereiche des Plug-Ins dadurch über die initial erschlossenen Bereiche ausdehnen und weiterentwickeln können. Diese Entwicklungsfähigkeit ist in einer Zeit der rasanten Veränderungen und Neuentwicklungen in der Softwarebranche wichtig, da nur so eine stetige Verbesserung erreicht werden kann und nicht technologischer Stillstand eintritt.

4.5 Das gewählte Versionskontrollsystem

Alle in diesem Abschnitt angesprochenen VCS entsprechen zumindest den meisten der geforderten Kriterien für die Ausarbeitung eines Plug-Ins zur Versionskontrolle während der Konzept und Entwicklungsphase.

So erlaubt das System RCS zwar die Verarbeitung von binären Daten, jedoch ist es auf eine lokale Anwendung beschränkt und wird auch von keinem Entwickler mehr weiterentwickelt. CVS erlaubt zusätzlich zu den Möglichkeiten in RCS auch eine Anwendung im Netzwerk. Die Entwicklung kam jedoch, nicht nur aufgrund vieler und nicht dokumentierter Eingriffe von unterschiedlichen Programmierern, zum Erliegen und wurde schon seit Jahren vollständig durch Nachfolger abgelöst. Aus diesen Gründen kommen die beiden Systeme RCS sowie CVS bei der Erstellung dieser Arbeit nicht zum Einsatz.

Im weiteren wurden noch das System Subversion und GIT diskutiert, welche beide gute Qualitäten für die gewünschte Ausarbeitung haben.

Subversion ist ein solides und stabiles Versionskontrollsystem, welches weitgehend auf den Konzepten von RCS und CVS aufbaut und diese weiter entwickeln konnte. So ist es mit Subversion möglich Dateien, welche nicht textbasiert sind, durch einen Benutzer zur Bearbeitung zu sperren und nach erfolgter Bearbeitung wieder für andere Benutzer freizugeben. Dabei wird anderen Benutzern nicht die Möglichkeit genommen den letzten Stand der Bearbeitung einzusehen, lediglich werden die Rechte zur Veränderung der Datei an einen singulären Benutzer vergeben, um dadurch Problemen durch eine Mehrfachänderung entgegenzuwirken. Weiters ist Subversion in der Lage in einem Netzwerk, mit zentralem Archiv, zu agieren und somit jedem Bearbeiter zu jedem Zeitpunkt die aktuellsten Daten zur Verfügung zu stellen.

Mit GIT kam eine gänzlich neue Idee der Verwaltung von Versionen auf. Diese basiert hauptsächlich auf der Idee, dass jeder Benutzer eines Archivs dessen gesamten Inhalt besitzt und diesen verändern kann, bis er alle Änderungen gesammelt an einen, mehrere oder alle Benutzer weiter gibt. Durch diese neue Form der Versionskontrolle wird das System im Vergleich zu allen Vorgängern deutlich beschleunigt. Lediglich bei der Übertragung der Änderungen an andere Benutzer muss man etwas längere Übertragungszeiten hinnehmen, welche jedoch für eine Verwendung als Basis für das Plug-In nicht als relevant zu betrachten sind. Ein weiterer positiver Aspekt der

Verwaltungsweise von GIT ist, dass jeder Bearbeiter sozusagen ein Backup des Gesamtarchivs besitzt und dieses bei Bedarf auch als solches verwendet werden kann. Andere Systeme müssen hierbei noch auf konventionelle Backupsysteme zurückgreifen. Auch die Verarbeitung von nicht textbasierten Dateien ist mit GIT möglich.

Obwohl GIT alle Ansprüche an ein Versionskontrollsystem für das Plug-In erfüllt, hat es einen entscheidenden Nachteil: sein revolutionäres Konzept zur Verwaltung von Versionen und die damit einhergehende Übertragung von Änderungen an alle beziehungsweise an einige Benutzer. Dadurch ist es für das System fast unmöglich, einzelne Dateien nur als durch einen Benutzer veränderbar zu markieren. Zusätzlich wird die Kommunikation zwischen allen Beteiligten, welche auf die aktuellsten Daten angewiesen sind erschwert. Der Vorteil den GIT durch diese Strukturen für andere Einsatzbereiche generiert, wird bei einer Anwendung für die Versionskontrolle von architektonischen Zeichnungen relativiert und könnte sich ins sogar negative umkehren. Zwar ist es GIT möglich ähnlich wie Subversion als Zentrales Archiv mit Sperrfunktionen zu funktionieren. Diese Möglichkeit degradiert GIT jedoch zu einem Subversion-klon, welcher zwar die selben Funktionalitäten wie Subversion aufweisen kann, jedoch, durch die nicht wie ursprünglich vorgesehen verwendete Systemarchitektur, seine Vorteile nicht mehr ausspielen kann.

Da Subversion ebenso wie das jüngere GIT immer noch weiterentwickelt und verbessert wird, ist in diesem Fall Subversion klarer Favorit, da es nicht nur alle geforderten Eigenschaften erfüllt, sondern auch noch Entwicklungspotential besitzt. Die gute und durchgängige Dokumentation, die klare Befehlsarchitektur, sowie die vielfältigen Möglichkeiten der Verzweigung und deren Zusammenführung sind gute Voraussetzungen, um Subversion als Basis eines Plug-Ins zur Versionskontrolle für die Architekturarbeit zu verwenden.

5 Die momentane Situation oder der „Status Quo“

Um ein Plug-In zur Versionskontrolle für die Architekturarbeit zu etablieren ist es nicht nur wichtig über die technischen Details zu entscheiden. Viel entscheidender ist es, dieses in die Abläufe der Produktion einzubetten und diese dadurch zu unterstützen und weiterzuentwickeln.

Es muss davon ausgegangen werden, dass ein Projekt nicht nur von einer Person bearbeitet wird, sondern von vielen verschiedenen Personen, mit teils unterschiedlichen Arbeitsweisen und beruflichen Hintergründen, bearbeitet werden wird. Daraus ergeben sich organisatorische und prozessorientierte Herausforderungen, welche in diesem Kapitel kurz beleuchtet werden sollen, um für die weitere Ausarbeitung des Plug-Ins die nötigen Grundfunktionen zu definieren.

Bei der Bearbeitung von Projekten bedarf es, ohne einer automatisierten oder unterstützenden Software zur Versionskontrolle, einiger wichtiger Grundvoraussetzungen, um einen reibungslosen Projektablauf gewährleisten zu können. So braucht es die genauen Definitionen der Vorgehensweisen bei der Erstellung von Dokumenten, einer klaren Ablagestruktur aller Dokumente und einer Kommunikationskultur. Dadurch werden erst die Strukturen geschaffen, welche eine klare und möglichst reibungsfreie Bearbeitung von Projekten ermöglichen. Jedem beteiligten Mitarbeiter wird dadurch die Möglichkeit gegeben, sich in den Projekten, die durch diese Strukturen getragen werden, zurechtzufinden und diese auch dementsprechend bearbeiten zu können.

Die Gesamtqualität eines Projektes hängt vor allem von der Arbeit jedes einzelnen Mitarbeiters und der Qualität der Zusammenarbeit zwischen den unterschiedlichen Bearbeitern, sowie der nachvollziehbaren Entscheidungen jedes Einzelnen ab. Zugrunde liegt allem ein gemeinsamer Grundkonsens wie, was, wann und warum gemacht werden soll. Dieser Grundkonsens wird als Strukturen oder auch Qualitätsmanagement bezeichnet. Ob mit oder ohne einem Programm für die Versionskontrolle, in jedem Fall müssen zuerst diese Mindeststandards festgeschrieben werden, um mit hoher Qualität produzieren zu können.

5.1 Strukturen für eine nachvollziehbare Planung

Solche Strukturen werden meist durch eine interne Layerstruktur, genau definierte Ablagesysteme für die Papierablage und Vorlagen für den Schriftverkehr erstellt. Häufig gibt es noch definierte Ablagestrukturen für digitale Dokumente. All diesen Vorlagen ist jedoch gemein, dass der einzelne Mitarbeiter auch in genau diesem System arbeiten muss, da sonst diese Strukturen korrumpiert werden.

Die Komplexität steigert sich mit der Zunahme der Akteure in einem System, wobei die Einfachheit von Strukturen deren Durchführung erleichtert und somit dem Einzelnen erlaubt, das jeweilige Projekt auf Grundlage der standardisierten Strukturen zu verstehen und in den Arbeitsablauf einzusteigen. Die Erstellung dieser Strukturen erfordert eine sorgfältige Analyse, wie viel Komplexität versus Einfachheit der einzelnen Strukturen nötig ist und wie die verschiedenen Strukturen untereinander verflochten werden.

5.1.1 Layerstruktur

Eine Layerstruktur sollte für das Büro und seine Aufgabenbereiche so einfach wie möglich gestaltet werden. In Wiener Planungsbüros wird oft die Layerstruktur der Stadt Wien verwendet. Doch gibt es auch Büros welche eine eigene Layerstruktur entwickelt haben.

Die Grundlage jeder Layerstruktur, sei es nun die Struktur wie sie von der Stadt Wien verwendet wird, oder eine Struktur welche von einem Planungsbüro ausgearbeitet worden ist, ist der Versuch eine Zeichnung so darzustellen, dass es dem einzelnen Bearbeiter oder Benutzer ermöglicht wird, die einzelnen Teile einer Zeichnung separat erfassen und bearbeiten zu können. Layerstrukturen werden meist eingesetzt um einerseits die Qualität und andererseits die Produktivität zu steigern. Diese Verbesserungen haben den ursächlichen Grund darin, dass durch eine Systematisierung der zeichnerischen Abläufe eine gleichbleibende Qualität sowie eine bessere Verständlichkeit der Zeichnungen geschaffen wird.

Ein architektonischer Plan ist grundsätzlich eine Anhäufung von Linien, welche durch die Verwendung einer Layerstruktur für den Anwender besser kontrollierbar wird. Die Strukturierung bietet neben, einer Struktur auch die Möglichkeit, zum momentanen Bearbeitungsstand, nicht benötigte Elemente vorübergehend auszublenden und

vereinfacht somit die Zeichnung. Durch diese Vereinfachung der Ansicht wird es dem Bearbeiter ermöglicht sich auf die momentane Aufgabe zu konzentrieren.

Wird die Struktur jedoch zu komplex, führt dies zu einem Unverständnis selbiger. Dieses Unverständnis gegenüber einer Struktur führt meist zur Nichtanwendung oder der Falschanwendung. Deshalb wird bei Layerstrukturen oft auf ein einfaches und verständliches Kürzelsystem zurückgegriffen. Durch die Verwendung von mehrstufigen Systemen, zum Beispiel WA_TRA_cs = Wand_tragend_geschnitten, können mehrere verschiedene Elemente in einzelne Gruppen zusammengefasst werden und erleichtern somit das bessere Verständnis der Struktur an sich.

Neben der Ordnung nach thematischen Gruppen (Wände, Öffnungen, Decke, ...) ist aber auch eine Ordnung nach organisatorischen Gruppen (Rohbau, Ausbau, ...) in vielen Layerstrukturen zu finden.

5.1.2 analoge und digitale Ablagesysteme

Genauso wie eine Layerstruktur muss auch eine Ablage strukturiert werden. Durch immer gleichbleibende Ablageverfahren und Logiken wird es den verschiedenen Akteuren in einem Projekt ermöglicht, unabhängig von einander, dem Projektverlauf folgen zu können. Zusätzlich erlaubt ein strukturiertes Ablagesystem im späteren Projektverlauf, die Projekthistorie aufrollen zu können. Dies ist insbesondere bei Projektende oder bei Problemen in der Planung von entscheidender Bedeutung. Ebenso können durch eine klare Struktur verschiedene Projekte miteinander verglichen werden. Dies hat zwar meist keinen direkten Einfluss auf ein laufendes Projekt, kann aber für zukünftige Projekte von Interesse sein, da dort Erkenntnisse aus vergangenen Projekten Einfluss nehmen sollen. Somit wird Langfristig an einer Qualitätsverbesserung gearbeitet.

Analoge Ablagestrukturen werden immer noch von vielen, wenn nicht von allen Büros, verwendet. Ihr großer Vorteil besteht darin, dass einzelne Dokumente direkt nach Erhalt beziehungsweise nach Erstellung in Papierform, zumeist in einem Ordner, abgelegt werden. Diese Ablagemethode hat jedoch einige gravierende Nachteile. Sie produzieren sehr viel Papier, einen enormen Archivierungsaufwand, braucht viel Raum und viel Zeit bei ihrer Erstellung. Zum Anderen muss sie in der heutigen Zeit von Computerarbeit und elektronischem Schriftverkehr.

Der heute schon obligatorischen digitalen Ablage, müssen nur die nicht digitalen Objekte wie Skizzen, Mitschriften aus Besprechungen, Plankorrekturen, usw. angehängt werden. Die Digitalisierung dieser noch immer analogen Arbeitsgänge wird von vielen Büros bereits standardmäßig im Nachhinein durchgeführt, um ein möglichst durchgängiges Bild eines Projektverlaufs auch in einem digitalen Abbild wiedergeben zu können.

5.1.3 Vorlagen für den Schriftverkehr

Schriftverkehrsvorlagen erleichtern im Büroalltag grundsätzlich die Abwicklung der Kommunikation und den Wissenserhalt während eines Projektes und darüber hinaus.

Für den Einzelnen erleichtern Vorlagen nicht nur das Erstellen sondern auch das Lesen von projektrelevanten Informationen.

Protokolle werden meist schon mittels eigens dafür programmierten Lösungen erstellt und können dadurch den Protokollverlauf klarer darstellen als wenn diese manuell erstellt würden. Da vorangegangene Punkte und Zukünftige Punkte unter den verschiedenen Protokollen verknüpft werden können.

Der einzige Nachteil dieser Lösungen besteht eigentlich nur darin, dass sie eine Einschulung auf das jeweilige System voraussetzen und zwischen mehreren verschiedenen Systemen noch nicht verknüpfbar sind. So können Protokolle aus einem Programm nur mit Abstrichen in ein anderes übernommen werden.

Der große Vorteil solcher Programme besteht allerdings darin, dass Protokolle wie Planungsbesprechungen, Baubesprechungen, Telefonnotizen und der Gleichen in einem System erfasst werden können und somit die Ablage und Aufbereitung für den weiteren Planungsverlauf erleichtern, da ein durchgängiger Protokollverlauf gewährleistet werden kann.

5.1.4 personelle Strukturen

Gerade bei der Abwicklung von Projekten mit mehreren Personen ist es für jeden Beteiligten wichtig, die Hierarchie und somit auch seine direkten Ansprechpartner zu kennen.

Nur durch eine klare Entscheidungsstruktur ist es Beteiligten möglich Entscheidungen treffen zu können, welche den Projektfortschritt dienlich sind. Da durch eine klare Struktur ersichtlich wird wer welche Entscheidungen treffen darf und soll, weiß auch jeder Beteiligte welchen anderen Beteiligten er bezüglich einer Aufgabe oder Frage ansprechen kann beziehungsweise soll.

Generell wird dies in Büros mit mehreren Mitarbeitern durch sogenannte „Keyuser“ oder auch „Fachspezialisten“ umgesetzt. Hierbei handelt es sich um Mitarbeiter welche spezielle Kenntnisse in den jeweiligen Fachbereichen abdecken.

Für einzelne Projekte wird hierbei teils extern auf diese „Fachspezialisten“ zurückgegriffen, andererseits gibt es für die einzelnen Projekte wieder eigene „Projektspezialisten“, welche sich mit Detaillösungen für ein Projekt befassen und diese auch behandeln. Diese „Projektspezialisten“ werden zumeist von einem Projektleiter unterstützt, welcher, bei großen Projekten, einen Gesamtprojektleiter unterstellt wird.

5.2 Der Büroalltag

Im Folgenden soll nun ein Projektablauf in einem mittelgroßen Planungsbüro beschrieben werden, um die Komplexität der Aufgabe Planung darzustellen und auch deren Schwachpunkte offenzulegen, um im weiteren Verlauf eben diese Schwächen genauer beleuchten zu können.

5.2.1 Initialisierung von neuen Projekten

Wenn ein neues Projekt initialisiert wird, werden zu aller erst die nötigen Projektstrukturen geschaffen. Dies beinhaltet das Anlegen der Ordnerstrukturen in digitaler sowie analoger Form sowie die Vorbereitung der Hintergrundprozesse wie Planlisten, Übergabelisten, ...

Diese Arbeit übernimmt in den meisten Fällen der Projektleiter beziehungsweise ein von ihm angewiesener Mitarbeiter. Da die Abläufe zumeist standardisiert sind, das heißt

die Abläufe wurden vorab schon als Vorlage hinterlegt und werden bei jedem Projektstart ausgeführt, kann davon ausgegangen werden, dass das neue Projekt in der Ablagestruktur den meisten vorhergegangenen Projekten gleicht. Wenn sich natürlich die Vorlagen ändern, wird auch dieses Projekt anders strukturiert sein. Deshalb behilft man sich in solchen Fällen meist einer Adaptierung der Projektnummer – eine interne Nummer welche durchgängig von Projektstart bis Ende verwendet wird und nicht zwei Mal vergeben werden darf. Als Beispiel sei hierbei genannt ein Projekt 123-14, dieses verwendet noch alte Strukturen, wie Layerstruktur, Schriftverkehr, und Ordnerstrukturen. Ein Projekt mit der Nummer 0124-14 hingegen verwendet schon die neuen Strukturen. Durch die Nennung einer in diesem Beispiel vorangestellten Null ist den Mitarbeitern klar, dass bei diesem Projekt neue Strukturen verwendet werden.

5.2.2 Aufgabenstellung

Nach der Initialisierung eines neuen Projekts wird die Aufgabenstellung erarbeitet. Dies bedeutet nicht, dass erst zu diesem Zeitpunkt die Anforderungen der Auftraggeber übergeben werden, sondern, dass die Anforderungen der Auftraggeber insofern bearbeitet werden, um ein klares Anforderungsprofil zu erhalten. Hierbei findet unter anderem der Zeithorizont, die Ausarbeitungstiefe der jeweiligen Phasen sowie die qualitativen und quantitativen Ziele besondere Beachtung.

Zusätzlich werden aus älteren Projekten mögliche Vorlagen, Ideen, Planungen und ähnliches zusammengestellt, um Erfahrungen und schon erarbeitete Teillösungen für schon bekannte Problemfälle, welche auch in diesem neuen Projekt erwartet werden können, leichter zu bearbeiten und mögliche Doppelgleisigkeiten zwischen Projekten zu verhindern.

Erst nachdem ein Projektleiter ein genaues Bild der zu erledigenden Aufgaben und möglicher Vorlagen für ein neues Projekt hat, kann er eine Ressourcenplanung erstellen.

5.2.3 Ressourcenplanung

Die Erstellung einer Ressourcenplanung ist besonders wichtig, da aus ihr ersichtlich wird, wie viele Beteiligte in jeder Phase der Planungstätigkeiten gebraucht werden. Zusätzlich wird dadurch auch eine Projekthierarchie geschaffen. Dadurch fällt es in den weiteren Schritten einfacher den Fortschritt der Tätigkeiten und die Produktionszeiten besser überwachen und gegebenenfalls anpassen zu können.

Bei der Ressourcenplanung wird vorab grob ein Ablaufterminplan erstellt und dieser mit den dafür nötigen Mitarbeitern verbunden. Besonderes Augenmerk wird dabei auf die Auslastung der einzelnen Akteure verwendet. Dies ist nötig, um Mitarbeiter nicht in ihren Aufgaben zu überlasten und eine Grundausslastung der einzelnen Mitarbeiter sicherzustellen. Meist werden Mitarbeitern, welche die Funktion von Fachspezialisten inne haben noch ein oder mehrere Mitarbeiter zur Seite gestellt, um ihnen neben der Projektarbeit genügend Zeit zu geben, ihre Fachspezialistenfunktion auch weiterhin für alle Projekte ausüben zu können.

5.2.4 Verteilung der Planungsaufgaben

Die aus der Ressourcenplanung resultierenden personellen Strukturen werden den Mitarbeitern bei einem Projektstartmeeting dargelegt. Die einzelnen Akteure werden in den bisherigen Verlauf des Projektes eingewiesen und die Ziele und Vorlagen werden erläutert. Dadurch kann jeder Bearbeiter mit dem selben Wissensstand in seine Aufgaben einsteigen.

Diese Besprechungen werden meist in einem wöchentlichen Rhythmus wiederholt, um sicherzustellen, dass Projektleiter sowie die einzelnen Bearbeiter immer den selben Wissensstand über den Fortschritt des Projektes haben. Zusätzlich wird dem Team dadurch die Möglichkeit gegeben, mögliche Problemstellungen vorab zu erkennen und frühzeitig auf darauf reagieren zu können.

5.2.5 Bearbeitung

Nach der Verteilung der einzelnen Planungsaufgaben, beginnt jeder Mitarbeiter mit der Abarbeitung seiner Aufgaben. Besonderes Augenmerk ist hierbei auf den Aufbau der einzelnen Unterlagen im CAD System zu richten. Der Aufbau von neuen Unterlagen erfolgt immer aus der Vorlagenbibliothek des Büros oder aus der Vorlagenbibliothek des Projektes. Dabei ist von jedem Mitarbeiter die gleiche Vorgangsweise bei der Erstellung von Zeichnungen, welche untereinander verlinkt werden, zu achten. Bei Fehlern in diesem Frühstadium der Planungsarbeit ergeben sich zumeist nicht direkt Probleme, jedoch mit Fortschreiten der Arbeiten kommt es dadurch meist zu unnötigen Komplikationen. Die daraus entstehenden Probleme können im Nachhinein zumeist nur noch durch sehr aufwändige und zeitintensive Veränderungen oder Workarounds sowie ad hoc Lösungen für Teilprobleme gelöst werden.

Werden alle Vorgaben der Erstellung von Unterlagen eingehalten, kann auch ein anderer Mitarbeiter bei einer Teilaufgabe kurzzeitig einspringen ohne zusätzlich zum wöchentlichen Jour Fix eine besondere Einschulung in den Teilbereich zu brauchen, da die Struktur, in welcher gearbeitet wird, in allen Teilbereichen gleich ist und durch den Jour Fix die nötigen Informationen zwischen allen Teilnehmern ausgetauscht werden.

Während der Bearbeitungszeit muss der Bearbeiter, speziell bei Projekten, welche durch mehrere Bearbeiter des selben Aktionsfeldes, oder auch mehrerer Fachbereiche, zum Beispiel Architektur, Statik und Haustechnik bearbeitet werden, darauf achten, dass er die jeweils aktuellen Planungsunterlagen seiner Kollegen als Grundlage seiner weiteren Arbeit verwendet. Hierbei ist eine effiziente und schnelle Kommunikation und Aktualisierung der Planungsgrundlagen in einer Zeichnung unabdingbar. Kann es doch durch die Verwendung von veralteten Planungsgrundlagen dazu kommen, dass unnötige Problemlösungen ausgearbeitet und Arbeiten mehrmals ausgeführt werden.

5.2.6 Kommunikation

Eine effiziente und schnelle Kommunikationskultur innerhalb eines Planungsteams ist besonders wichtig und bedarf einer gewissen Selbstdisziplin der einzelnen Beteiligten. Nicht nur soll während der üblichen wöchentlichen Besprechungen ein Austausch von Änderungen erfolgen, sondern auch in der Zeit zwischen diesen Besprechungen. Sobald eine Neuerung der Planunterlagen erstellt wird sollte, schon eine Kommunikation stattfinden. Denn mit großer Wahrscheinlichkeit hat, beziehungsweise hatte, ein anderes Mitglied schon einmal mit einem ähnlichem Problem zu tun. Umplanungen welche sich aus architektonischen Gegebenheiten ergeben, sollten ebenfalls so schnell als möglich an alle Akteure verteilt werden und deren fachliche Meinungen eingeholt werden. Dies ist besonders wichtig, sollten diese Änderungen, oder neuen Ideen, Gewerke betreffen, welche durch Spezialisten wie Statiker, Haustechniker oder Bauphysiker geplant werden.

Auch im Rückschluss ist es von immenser Wichtigkeit, Änderungen oder neue Problemlösungen dieser Fachbereiche schnell in die eigene Arbeit zu integrieren. Da Bearbeiter welche einen Bereich bearbeiten, der sich mit dem Arbeitsbereich eines Mitarbeiters überlagert auf eine schnelle, effiziente und projektorientierte Kritik ihrer Planungen angewiesen sind, um Probleme schon im Vorfeld abzuwenden. Hierbei ist es wichtig, dass die verschiedenen Akteure den selben Planungsstand als Grundlage

verwenden. Diese Notwendigkeit führt dazu, dass fertige Unterlagen, welche mit anderen Akteuren geteilt werden, immer mit einer Versionsnummer beziehungsweise mit einem Datum versehen werden sollten. Nur dadurch wird sichergestellt, dass alle Beteiligten über die selben Dinge sprechen. Zusätzlich wird sichergestellt, dass zumindest bei diesen Konsultationen der aktuelle Stand von Planungsunterlagen verifiziert wird. Viele Missverständnisse während der Planungsphase von Projekten zwischen verschiedenen Fachrichtungen lassen sich vermeiden, wenn immer die aktuellsten Fortschritte der einzelnen Planungen ausgetauscht und in die eigene Planung eingearbeitet werden.

5.2.7 Überprüfung

Um Fehler schon während der Planung vorzubeugen wird oft darauf zurückgegriffen Pläne und Ergebnisse, durch von der Erstellung der Unterlagen unabhängigen Personen, meist sind es Kollegen, welche andernorts in das Projekt involviert sind, überprüfen zu lassen. Zu einer solchen Überprüfung braucht es nicht ausschließlich fachliche Kompetenzen sondern auch einen projektspezifischen Hintergrund, um Kritik nicht nur Fachlich sondern auch Projektspezifisch in eine Korrektur einfließen lassen zu können. Da vielen Fehlern, auch bei erfahrenen Beteiligten, oft die sogenannte „Betriebsblindheit“ zugrunde liegt.

Im täglichen Arbeitsgeschehen werden solche Überprüfungen meist am Bildschirm durchgeführt und funktionieren bei eingespielten Projektteams durchaus effizient, da jeder Beteiligte bei der Begutachtung neuer Unterlagen für seine weitere Arbeit die Arbeit der anderen Beteiligten kritisch hinterfragt. Dadurch werden Fehler früher als bei den analogen Korrekturmethode gefunden, da immer wieder von unterschiedlichen Personen die neuesten Daten gesichtet und hinterfragt werden können.

Natürlich sind solche Korrekturmechanismen nicht als Ersatz für die Korrekturlesungen am analogen Plan zu sehen. Bei der Korrektur von analogen Plänen werden meist auch Darstellungsprobleme und Defizite in der Ausdifferenzierung ersichtlich, da ein analoger Plan, welcher meist in Schwarz-Weiß oder Graustufen gehalten ist, nicht mehr die Hilfe von unterschiedlichen Layerfarben bietet und somit die Darstellung des Projektes in den Vordergrund hebt.

5.2.8 Publizieren

Nach Abschluss einer Planungsphase oder beim Versand von Planunterlagen an Dritte beziehungsweise an den Auftraggeber müssen, wie schon im Kapitel 5.2.6 „Kommunikation“ erläutert, Versionen angelegt werden, welche klar den jeweiligen Stand der Arbeiten widerspiegeln. So wird meist bis zu einer Freigabe durch den Bauherrn oder dessen Vertretern mit dem Vorabzug gearbeitet. Erst ab einer schriftlichen Freigabe wird der Vorabzug entfernt und der Plan das erste Mal mit einem Index, meist in alphabetischer Reihenfolge A, B, C, ... versehen. Der Vorabzug ist hierbei von rein rechtlicher Relevanz, da bis zu einer Freigabe der Planung durch einen Auftraggeber dieser Plan nur zur internen Verwendung vorgesehen ist und in Folge einen unfertigen Planungsstand darstellt. Ihm gegenüber steht der freigegebene Plan, welcher durch den Auftraggeber als fertiger Planungsstand gekennzeichnet wurde und in denn meisten Fällen nur noch an neue Gegebenheiten angepasst wird.

Um einen neuen Planstand herzustellen braucht es 3 Schritte.

- Als erstes wird die Datei, welche Grundlage des neuen Planstandes ist, in einem Archiv gespeichert und dann für eine Offlineverwendung vorbereitet. Dies bedeutet, dass die Zeichnung von nicht relevanten Elementen und Inhalten gereinigt wird um die Dateigröße zu minimieren.
- Im zweiten Schritt wird die Datei in einem unveränderbaren Dokumentenformat gespeichert. Ziel dieses Schrittes ist es, dass die Zeichnung wie sie auf Papier gedruckt wird, auch als digitale Datei zu speichern. Meist ist dies auch die Datei welche an den Plotter gesendet wird, da die Layouteinstellungen hierbei überprüft werden können
- Dannach wird noch eine E-Transmitt-Datei angelegt, um jedem Empfänger die Möglichkeit zu bieten, die Datei in der gleichen Weise reproduzieren zu können. Eine E-Transmit-Datei ist eine komprimierte und vereinfachte Sammlung aller für eine Reproduktion wichtigen Dateien, wie zum Beispiel alle verknüpften Zeichnungen, die Stifteinstellungen für den Drucker, verlinkte Bilder, andere Dateien und natürlich auch die Zeichnung an sich.

Zusätzlich zur Erstellung von Planständen muss parallel eine Planliste geführt werden. Diese Liste gibt Auskunft über die Pläne eines Projektes, deren Index, Verfasser, Verfassungsdatum, Freigabestand, Maßstab, Format und den digitalen Ablageort.

5.2.9 Projektende

Wird ein Projekt beendet, muss es für die Archivierung vorbereitet werden. Einerseits müssen alle Verknüpfungen innerhalb eines Projekts von absoluten zu relativen Verknüpfungen aktualisiert werden. Da sonst nach einer Archivierung des Projekts keine der Verknüpfungen mehr auf die richtigen Daten verweist und somit das Archiv praktisch nicht mehr nutzbar ist.

Andererseits werden am Ende eines Projekts diejenigen Problemlösungen welche mit größter Wahrscheinlichkeit wieder gebraucht werden, in ein eigenes Archiv dupliziert, um für spätere Projekte eine Wissensdatenbank aufzubauen. Auch werden nach Beendigung von Projekten Projektdatenblätter erstellt, welche schlussendlich ein Inhaltsverzeichnis des Projekt- und Wissensarchivs bilden. Durch die Datenblätter ist sichergestellt, dass folgende Bearbeiter sich in die jeweiligen Archivprojekte schnell einlesen können, um einerseits schnell Vorlageprojekte für Neuprojekte zu finden und sich andererseits bei einer Reaktivierung von Projekten schnell wieder einarbeiten zu können.

Um die Datenblätter für solche Anwendung aufzubereiten, muss der Ersteller einerseits einen kurzen Abriss der Projekthistorie erstellen und eine Aufstellung der wichtigsten Kenndaten des Projektes hinzufügen. Wichtige Informationen unter anderem die jeweiligen Bearbeiter, ob und welche Daten in die Wissensdatenbank Einzug gehalten haben und Abbildungen des Projektes, sollten in einem Projektdatenblatt ebenfalls nicht fehlen, da dadurch die Weiterverwendung erleichtert wird.

Die Datenblätter finden nicht nur für die Archivverwaltung und -einsicht Verwendung, sondern werden auch für neue Projektakquisition herangezogen.

5.3 Fazit der aktuellen Arbeitsweise

Durch die Betrachtung der aktuell im Projektgeschehen oft angewendeten Arbeitsweisen fällt klar auf, dass der einzelne Mitarbeiter eine nicht zu vernachlässigende Flut von Strukturen beherrschen und auch anwenden muss, um ein Projekt durchgängig nachvollziehbar zu machen. Natürlich kann ein positiver Projektabschluss nicht durch eine durchgängige Nachvollziehbarkeit und Dokumentation alleine erreicht werden, hierzu braucht es noch viele andere Faktoren. Doch kann eine klare Dokumentation den Ausgang eines Projektes und folgende Projekte positiv beeinflussen und unterstützen.

Durch die Vielzahl und Komplexität von Vorgängen, welche eingehalten werden müssen, um eine solche Dokumentation zu erzeugen, kommt es meist zu Lücken in der Dokumentation. Diese Lücken entstehen vor allem dann, wenn die Bearbeiter nicht die vorgelegten Vorgehensweisen einhalten oder diese zugunsten von Zeitersparnis oder aus Unwissenheit verändern. Meist kommt auch nur der Faktor Mensch zum Tragen, wenn Daten falsch oder unzureichend gespeichert werden. Einmal gespeicherte Daten gehen zwar nicht verloren, können aber nur schwer wieder gefunden werden, wenn deren Ablageort nicht bekannt ist. Die häufigsten Fehler hierbei sind: die Ablage in falschen Ordnern, oder das Kreieren eigener Ordnerstrukturen, welche für spätere Bearbeiter oder sogar für am Projekt beteiligte Kollegen, nicht nachvollziehbar sein können. Dadurch kann es zu Doppelgleisigkeiten, nicht reproduzierbaren Ergebnissen, versteckten „Zeitbomben“ und vielen anderen Problemen kommen, welche den Projektfortschritt stören oder im Nachhinein ein Projekt von einem Erfolg in ein Desaster verwandeln können.

Auch die Ressourcenplanung ist bei der beschriebenen Arbeitsweise ein Problem, da ein Bearbeiter immer nur eine Zeichnung bearbeiten kann. Diesen Umstand versucht man, durch die Hintertür über verschiedene Zeichnungsebenen, zum Beispiel in AutoCad als „xref“ bezeichnet, zu umgehen. Die Erstellung solcher mehrlagigen Verknüpfungen ist sehr anspruchsvoll und muss am Anfang sehr genau geplant werden. Die ist sehr fehleranfällig, wenn diese Verknüpfungen unsachgemäß durchgeführt werden. Eine Aufteilung der Arbeit zwischen den unterschiedlichen Beteiligten und Gewerken findet dadurch oft nur zwischen verschiedenen Geschoßen beziehungsweise zwischen verschiedenen Darstellungsformen wie Aufriss, Grundriss, Schnitt, Detail, ... statt.

Die Pflege von verknüpften Zeichnungen stellt ein weiteres Problem dar. Wenn ein Bearbeiter einem anderen Bearbeiter dessen Stand seiner Planungen zur Weiterverwendung übersendet, muss dieser Stand in den Referenzeigenschaften der Zeichnung ausgetauscht werden. Dabei kommt es nicht selten vor, dass ein Bearbeiter zwar den Eingang der Daten dokumentiert und diese auch digital ablegt, jedoch vergisst, die neuen Daten in den Eigenschaften auszutauschen, oder dieses aufgrund von Zeitmangels zu einem späteren Zeitpunkt plant durchzuführen. Da nun der Dateneingang dokumentiert ist, glauben andere Bearbeiter diese seien ebenfalls schon in den Zeichnungen aktualisiert worden und planen mit einer überholten Unterlage weiter. Wie so oft kommt es später im Projektverlauf dann zu einem bösen Erwachen, wenn klar wird, dass auf einer veralteten Planung aufgebaut wurde. Besonders ausgeprägt kann dieses Problem werden, wenn auf jeder Seite diese Aktualisierung, aus welchem Grund nun auch immer, nicht durchgeführt wurde.

Das Teilen von Informationen stellt hierbei einen wichtigen Punkt der Projektarbeit dar. Durch das Erstellen von Versionen lassen sich Informationen an jeden Akteur, in einer für alle Beteiligten verständlichen Art und Weise, verteilen und dokumentieren. Wie bei allen Vorgängen während eines Projektes ist es wichtig, die richtigen Arbeitsschritte und deren Reihenfolge einzuhalten. Diese Arbeitsschritte erfordern Zeit, welche investiert werden muss. Da viele Projekte meist immer unter Zeitdruck stehen, werden die nötigen Schritte des Öfteren in adaptierter oder gar unzureichender Form ausgeführt, um Zeit zu sparen. Der jeweilige Ersteller erspart sich jedoch nur kurzfristig Zeit, denn sobald Informationen von oder über die Dokumentation benötigt werden, kommt es durch eben diese Adaption der Vorgehensweise, dazu, dass mehr Zeit aufgewendet werden muss, um die nötigen Informationen zu beschaffen.

Ein digitales Korrekturlesen durch andere Mitarbeiter ist mit der aktuellen Arbeitsweise sehr zeitaufwändig, da immer das gesamte Dokument überprüft werden muss, für einen Überprüfer ist es nicht immer möglich direkt zwischen älteren und neuen Inhalten zu unterscheiden. Aus diesem Grund stellt diese Art der Kontrolle eines Dokuments, nur für den Ersteller selbst, einen Zeitgewinn gegenüber der Analogüberprüfung dar. Da der Ersteller selbst jedoch zumeist eben durch die ausgiebige Beschäftigung mit dem Dokument eine gewisse „Betriebsblindheit“ entwickelt, muss aus qualitätsfördernder Sicht ein Dokument immer analog, vom Ersteller und anderen Personen, überprüft werden. Dieser Arbeitsschritt ist sehr zeitintensiv und behindert mindestens eine weitere Person an der Bearbeitung ihrer eigenen Agenden. Aus diesem Grund wird eine analoge

Überprüfung so selten wie, möglich durchgeführt. Dies kann wiederum zu einem sehr großen Zeitaufwand führen, da Fehler erst verspätet erkannt werden und gegebenenfalls

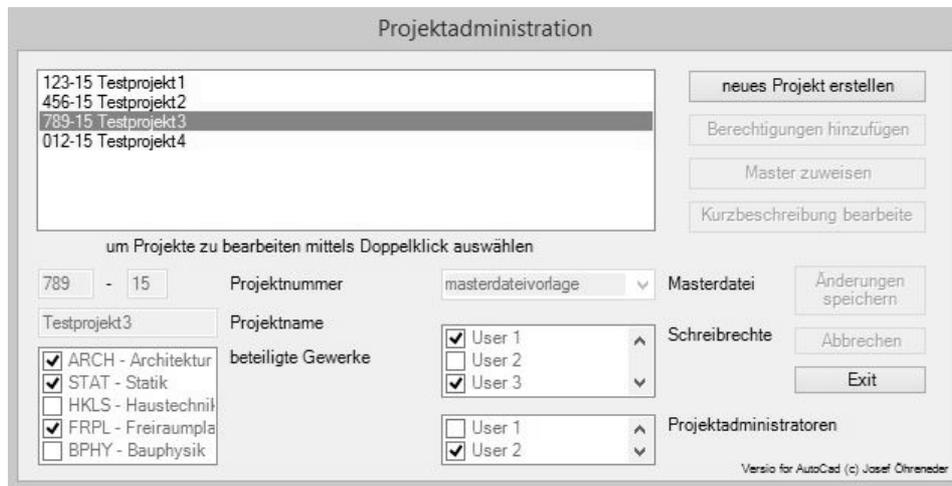


Abb. 9: Maske zur Projektinitialisierung (eigene Abbildung)

nicht nur in einem Dokument zu korrigieren sind. Dadurch kommt es wiederum dazu, dass man in Rückstand zum Zeitplan kommt und dadurch andere ebenfalls zeitaufwändige Arbeiten nicht mehr mit der Nötigen Konzentration und Kontrolle ausführt. All dies führt in einen Teufelskreis aus dem ein Projektteam nur sehr schwer wieder auszubrechen vermag.

Aus allen vorangegangenen Aussagen kann klar eine Kernaussage getroffen werden: Die Qualität von Projekten ist sehr stark von dem Wissen über Strukturen, deren Anwendung und deren Durchsetzung anderen Akteuren gegenüber abhängig. Die Umsetzung solcher Strukturen erfordert jedoch Zeit, welche bei vielen Projekten leider fehlt. Dies ist mitunter ein Grund, dass Strukturen nicht, beziehungsweise nur in sehr aufgeweichter Form Anwendung finden und zu den in diesem Kapitel angesprochenen Problemen führt.

6 Arbeiten mit Versionen

Ziel dieser Arbeit ist es, die im vorhergehenden Abschnitt beschriebenen Strukturen zu automatisieren und somit einem Projektteam wieder Zeit zu geben, sich auf die Projektarbeit konzentrieren zu können. Aus diesem Grund wurde ein Plug-In entwickelt, welches unter Zuhilfenahme von Versionskontrollsystemen die Strukturen während der Zeichenarbeit fest in den Arbeitsablauf zu integrieren versucht. Ziel des Plug-Ins ist es, dem Projektteam die aufwändige Arbeit der Dokumentation der Zeichenarbeit und die Erstellung von Projektvorlagen sowie die Generierung einer Wissensdatenbank zu erleichtern. Die Parallelarbeit an Projektinhalten sowie die Erstellung von verschiedenen Varianten wird von der Erweiterung ebenso unterstützt, wie die Möglichkeit Änderungen seit der letzten Betrachtung darzustellen und dadurch auf diese Änderungen schneller reagieren zu können.

6.1 Die Akteure

Um das Arbeiten mit Versionen umsetzen zu können braucht es unterschiedliche Akteure. Die einzelnen Akteure sollten Spezialisten ihres Bereichs sein.

Auf der einen Seite braucht es Administratoren, welche die Strukturen im Hintergrund erstellen, warten und erweitern. Die Administratoren sorgen somit für einen reibungslosen Ablauf der Projektarbeit. Ihre Kernaufgaben sind klar die administrativen Bereiche der Projektarbeit. Doch dürfen sie nicht ausschließlich mit der Administration beschäftigt sein, sondern sollten auch in kleineren Projekten zur planerischen Tätigkeit eingesetzt werden. Durch den Einsatz in den verschiedenen Ebenen der Projektarbeit können die Administratoren sehr viel effizienter die Wartung und Verbesserung der Strukturen durchführen, da sie selbst die Probleme miterleben und nicht nur auf die Anregungen der Projektbearbeiter angewiesen sind. Denn nur wenn Strukturen mit Projekten wachsen, sich anpassen lernen oder sich verbessern, können Büros sich entwickeln und somit Wettbewerbsfähig bleiben.

Auf der anderen Seite stehen die Anwender des Plug-In's. Sie verwenden die Erweiterung während ihrer Projektarbeit. Durch die im Hintergrund ausgeführten Strukturen müssen die Anwender diese nicht mehr selbst ausführen, selbst das Verständnis der Strukturen, welche durch die Erweiterung verarbeitet werden ist nicht

mehr unbedingt nötig, jedoch immer noch hilfreich. Dadurch ist es den Anwendern möglich, sich ihren Kernaufgaben zu widmen und die gewonnene Zeit dort zu investieren. Dadurch sollte nicht nur der Arbeitsablauf intuitiver werden, sondern auch der Workflow verbessert werden können.

6.2 Die Planungsarbeit revolutionieren

Durch die Verwendung eines Versionskontrollsystems wird es ermöglicht, dass eine automatisierte Dokumentation von Planungsschritten aufgezeichnet wird. Dadurch werden sämtliche Änderungen, die in dieses Versionskontrollsystem eingespielt werden, im Projektverlauf einsehbar, veränderbar oder können weiterentwickelt werden. Diese Schritte sind dadurch nicht mehr geradlinig angeordnet, sondern es kann direkt auf ältere Daten zurückgegriffen, oder zwischen Zeitabschnitten gesprungen werden. Die aktuellen Daten bleiben in einem parallelen Zweig erhalten und gehen nicht wie bei einer linearen Verarbeitung verloren. Durch die Verwendung eines Versionskontrollsystems wird es nun ermöglicht in einer Parallelentwicklung mehrere Projektideen zu entwickeln, sie direkt am Bildschirm zu vergleichen und die gewünschten Inhalte der verschiedenen Ideen in einem Paket zu bündeln.

Die Struktur der Ablage in Versionskontrollsystemen kann man sich bildhaft als zwei Bäume vorstellen, wobei ein Baum am Kopf steht und die Äste mit den Ästen des anderen Baumes verbunden sind. Wobei nicht alle Äste eine Verbindung aufweisen müssen. Es kann vorkommen, dass Ideen im Laufe eines Projekts nicht weiterverfolgt

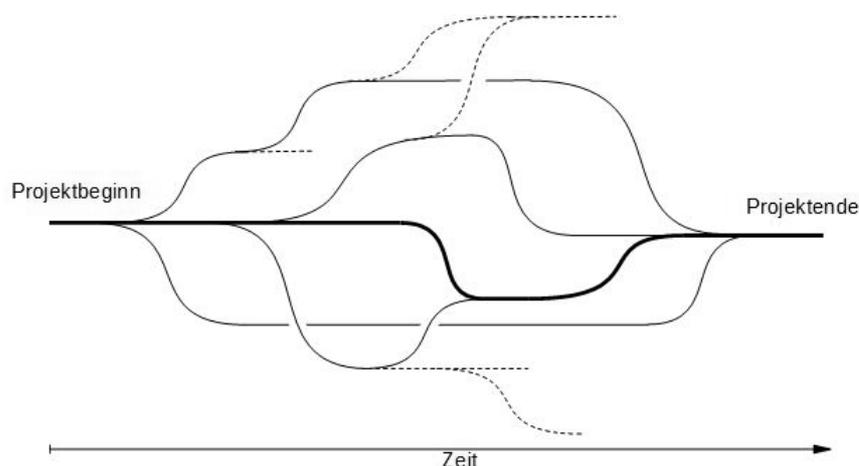


Abb. 10: Verzweigungen während eines Projektlaufes (eigene Abbildung)

werden und neue Inhalte hinzukommen, der Hauptstrang des Projektes hingegen schreitet bis zum Projektende immer weiter fort. So wie die Summe aller Teile ein Ganzes ergibt, braucht auch ein Projekt die Summe all seiner Ideen und Einflüsse. Bei

einer klassischen Arbeitsweise gehen Ideen, welche aufgegeben wurden meist unter und werden vergessen oder können nicht mehr dem richtigen Kontext zugeordnet werden. Da aber auch Irrwege und Sackgassen das Ergebnis eines Projektes beeinflussen, sind sie von unschätzbare Bedeutung für ein Projekt. Nehmen wir einmal an, dass eine Idee zu Beginn eines Projektes aufgegeben wird, da man annimmt sie würde in die falsche Richtung führen, jedoch stellt man nach einiger Zeit fest, dass Teile dieser Idee doch zu dem erhofften Ziel führen könnten. So kann der Zweig, oder der gewünschte Teilaspekt dieser Idee einfach, wieder in den Hauptast integriert werden. Durch die Wiederintegration eines Zweiges ergeben sich in der Struktur Verbindungen, welche klar den Ursprung des neuen Einflusses ersichtlich machen. Es ist dadurch nicht nur eine Dokumentation sondern auch eine digitale Karte des Projektes, welche alle jemals durchgeführten Änderungen darstellt und miteinander in Beziehung setzt.

Ähnlich einem Gedächtnis entstehen Verbindungen zwischen Ursache und Ergebnis, es entsteht also ein digitales Gedächtnis welches den verschiedenen Akteuren eines Planungsprozesses hilft den Verlauf der Planungen besser verstehen und leiten zu können.

6.3 Von der Arbeit der Administratoren

Der Administrator hat bei einer versionsbasierenden Arbeitsweise die Aufgabe, die im Hintergrund laufenden Prozesse zu erstellen und diese zu verwalten. Zusätzlich ist der Administrator ein sogenannter Superuser, durch seine umfassenden Kenntnisse des Systems hilft er den Bearbeitern bei technischen Problemen.

Ein Administrator soll aber keinesfalls nur administrative Aufgaben übernehmen, sondern wie auch die Bearbeiter selbst an der Projektarbeit teilnehmen. Nur durch die eigene Arbeit mit dem System, ist es dem Administrator möglich, Probleme vorab zu erkennen oder Probleme der Benutzer verstehen zu können. Durch dieses Verständnis des Systems auch von der Anwenderseite ist es den Administratoren auch möglich das System weiterzuentwickeln und die Grenzen der Softwareprogrammierung auszureizen.

Der Administrator koordiniert die Projektarbeit auf der Programmebene und arbeitet mit den Projektleitern Hand in Hand. Durch diese Synthese wird ein reibungsloser Ablauf der Projekte gewährleistet.

6.3.1 Die Vorlagen

Um mit dem Plug-In arbeiten zu können braucht es eine „Masterdatei“, ein „Masterlayout“, eine Layerstruktur und Blockdefinitionen.

Die Masterdatei ist die Basiszeichnung, in welcher die zu bearbeitenden Ebenen und die Hintergrundebenen geladen werden können. Sie stellt sozusagen das Gefäß der einzelnen Ebenen einer Zeichnung dar. Die schon heute verwendete Praxis der Verwendung externer Referenzen, kurz Xref, wird hierbei verwendet und in weiterer Folge als Ebene bezeichnet, diese repräsentiert eine eigene Datei und kann auch von Extern bearbeitet werden. Durch die Verwendung von Xrefs ist es möglich, Elemente einer Zeichnung so aufzuteilen, dass die einzelnen Elemente auf eigenen Ebenen angelegt und bearbeitet werden können. Die Aufteilung in solche Arbeitsebenen ermöglicht es mehreren Bearbeitern in einer Zeichnung aktiv zu sein. Diesen Umstand nutzt das Plug-In und verwaltet im Hintergrund die Schreib- und Leserechte für die einzelnen Benutzer. Zugleich erlaubt es die Aufteilung in Ebenen, sodass nur Elemente welche zur Arbeit nötig sind geladen werden müssen. Dadurch werden wiederum Ressourcen der Computer frei, die ein schnelleres und flüssigeres Arbeiten begünstigen. Die Erweiterung verwaltet und steuert die Eigenschaften und die Zuordnung der einzelnen Ebenen und erlaubt dadurch dem Anwender, ohne den Aufwand der manuellen Pflege der Verknüpfungen, zu arbeiten. Ein weiterer Vorteil dieser Vorgehensweise kann in der geringeren Fehleranfälligkeit der einzelnen Dateien gesehen werden. Bei großen und komplexen Zeichnungen kann es unter anderem zu Fehlern in der Verarbeitung kommen, die sogar zum Verlust der Daten führen können. Durch die Aufteilung einer großen Datei in mehrere kleine und den ständigen Austausch mit dem Versionsarchiv, kann es jedoch nie zu einem Kompletverlust kommen.

Die Layerstruktur und die Blockdefinitionen werden über das Plug-In aktuell gehalten. Hierbei wird die Funktion der Werkzeugpaletten in AutoCad genutzt. Sie erlauben es die Layerstruktur und vordefinierte Blöcke als Palette anzuzeigen und dadurch eine Zeichnung rein von nicht verwendeten Elementen zu halten. Die Informationen der Paletten werden von AutoCad in eigenen Dateien extern gespeichert und können dadurch problemlos aktualisiert werden.

Das Masterlayout funktioniert wie die Masterdatei ebenfalls als Gefäß für die einzelnen Ebenen, lässt aber im Gegensatz zur Masterdatei keine Bearbeitung der Ebenen zu, sondern ermöglicht lediglich das Publizieren der Ebenen.

Durch den Administrator wird auch festgelegt wie einzelne Ebenen benannt werden. Dies ist der Besonderheit geschuldet, dass alle Zeichnungen aus Ebenen bestehen. Diese Ebenen werden aber nicht wie üblich bei der Arbeit mit Xref in einer Zeichnung fix verlinkt, sondern stellen eigenständige Zeichnungen dar, welche je nach Wunsch übereinander gelegt werden können. Um diese freie Verwendung von Ebenen zu ermöglichen braucht es jedoch einen strikten Code für deren Benennung um jedem Benutzer zu ermöglichen auch die nötigen zusammenhängenden Ebenen zu finden.

Das Plug-In erlaubt in diesem Fall die Definition von vorangestellten Namen um eine Sortierung und Ordnung der Ebenen zu unterstützen. Hierbei kommen sogenannte Präfixe wie Zuordnung (Grundriss, Schnitt, Ansicht, Detail,...), Geschoß, alle Geschoße, Achse, und andere Unterscheidungsmerkmale zur Anwendung. Ziel ist es Ebenen zulassen zu können welche mit mehreren, bis dato, nicht in Verbindung stehenden Ebenen in Verbindung gestellt werden können. Zusätzlich können Auswahlsets, welche die Summe aller Ebenen druckbar oder nicht druckbar einer Zeichnung darstellen, zusammengestellt werden. Ein Auswahlset stellt somit eine Zeichnung im klassischen Sinne dar. Bei einer Auswahl durch einen Bearbeiter kann dieser zwischen Auswahlsets und den einzelnen Ebenen auswählen, welche Ebene er als Bearbeiter und welche Ebenen er als Betrachter öffnen möchte.

Die Benennung durch den Bearbeiter wird somit in einem ersten Schritt automatisiert, in einem weiteren Schritt gibt der Bearbeiter erst den eigentlichen Namen an. Vor der Erstellung kann dann noch die Zuordnung zu einem Auswahlset getroffen werden. Wird dies nicht gemacht generiert die Ebene ein eigenes Auswahlset. Dies muss geschehen, da sonst die Ebene beim Öffnen durch einen Benutzer mit der Funktion der Auswahlsets nicht aufscheinen würde.

6.3.2 Benutzerverwaltung

Um das Bearbeiten der einzelnen Ebenen und Vorlagen vor unbeabsichtigter Veränderung zu bewahren, werden mit dem Plug-In drei Benutzergruppen eingeführt. Die Administratoren, die Bearbeiter und die Betrachter.

Ein Administrator hat die Aufgabe, Vorlagen zu verwalten und aktuell zu halten, auch ist es seine Aufgabe neue Administratoren zu ernennen sowie neue Projekte und neue Zeichnungen anzulegen.

Jeder andere Benutzer ist entweder ein Bearbeiter oder Betrachter. Jeder Benutzer kann hierbei beim Öffnen einer Zeichnung selbst festlegen auf welchen Ebenen er Bearbeiter oder Betrachter ist. Der Unterschied zwischen Bearbeiter und Betrachter liegt in der Art und Weise wie Daten mit dem Versionsarchiv synchronisiert werden. Bei einer Bearbeitung wird im Versionsarchiv ein Marker hinterlegt, dass der Benutzer A die Rechte an einer Veränderung für sich beansprucht. Ein Betrachter synchronisiert eine schreibgeschützte Datei und hat somit keine Bearbeitungsrechte an der jeweiligen Ebene.

Im Fall dass Benutzer A eine Ebene als Bearbeiter öffnet, können andere Benutzer diese Ebene nur mehr als Betrachter öffnen. Mit dieser Vorgangsweise wird verhindert, dass Versionskonflikte entstehen. Wenn mehrere Fachbereiche an einem Projekt beteiligt sind, kann in der Benutzerverwaltung auch gesteuert werden, dass bestimmte Ebenen für andere Fachgruppen grundsätzlich nur als Betrachter offenbar sind.

Die Schaffung reiner Beobachter ist für ein Projekt besonders interessant, da ein Beobachter im Gegensatz zu einem klassischen Benutzer eine Zeichnung zur Bearbeitung nicht sperrt. Dies ist dann wichtig, wenn das Projekt von nicht direkt an der Entstehung der Zeichnungen beteiligten Partnern begleitet wird.

In der Benutzerverwaltung wird somit für jeden Beteiligten festgelegt welchem Fachbereich er zugeordnet ist, ob er ein reiner Beobachter ist (Sonderfall für Projektsteuerung, Baumanagement, usw.) und oder ob er zusätzlich auch als Administrator auftreten darf. Wenn ein Benutzer nur als Administrator auftreten darf, ist es ihm nicht möglich als Bearbeiter in Projekten tätig zu sein, da die Administratorrechte nur für die Administration des Plug-Ins von Relevanz sind.

6.3.3 Projektinitialisierung

Wird ein neues Projekt gestartet, werden als erstes die administrativen Grundlagen geschaffen. Erst wenn die Ressourcenplanung abgeschlossen ist, kann mit dem Plug-In ein neues Projekt initialisiert werden. Da schon zu Beginn des Projektes die ersten Projektbeteiligten bekannt sein müssen.

Die Initialisierung erfolgt mittels einer Eingabemaske, in welcher der Projektname, die Projektnummer, die verwendeten Masterdateien, die Berechtigten und die Kurzbeschreibung eingegeben werden. Das Plug-In kann daraufhin das Projekt im Versionsarchiv anlegen. Der Name und die Projektnummer werden nicht nur für die

reine Identifizierung eines Projektes benötigt, sondern werden mit der Kurzbeschreibung Teil des Projektdatenblatts. Das Datenblatt dient nicht nur zur Sammlung aller projektrelevanten Daten sondern auch für den Informationsaustausch. So kann ein unbeteiligter Mitarbeiter schon bevor er die Unterlagen eines Projektes sichtet, alle grundlegenden Daten des Projektes erfassen. Diese von Beginn an geführten Datenblätter von Projekten ermöglichen Mitarbeitern eine schnellere Suche nach bestimmten Inhalten.

Im nächsten Schritt werden die Masterdateien durch die Projektspezifika adaptiert. Hierbei wird speziell das Masterlayout mit Name, Projektnummer und Logo ausgestattet. Sollten spezielle Layer oder Blöcke zur Verwendung kommen, werden auch diese schon vorab eingestellt. Die Adaptierung kann auch zu einem späteren Zeitpunkt noch einmal ausgeführt werden, sollten Änderungen im Projektverlauf notwendig werden. Die erhaltenen externen Vorlagen für ein Projekt werden in das Ebenen System übersetzt und ebenfalls in die Versionshistorie aufgenommen. Weiters werden die ersten Ebenen der Zeichnung erstellt.

Nach dem Setup aller für den Projektstart benötigten Dateien, werden die Rechte an dem Projekt zugeordnet. Dabei werden die Teilnehmer, welche in der Teilnehmerverwaltung hinterlegt sind und dort auch ihren Fachgruppen zugeordnet sind für das Projekt freigeschaltet. Sobald das Projekt mit dem Versionsarchiv synchronisiert ist, werden die Projektbeteiligten durch den Administrator über die Aktivierung des Projekts und ihre Berechtigungen informiert.

Jeder Bearbeiter hat danach die Möglichkeit eine neue Ebene zu erstellen. Diese wird sogleich für seine Fachgruppe zur Bearbeitung freigeschaltet und mit dem Versionsarchiv synchronisiert. Nach der erfolgreichen Erstellung einer neuen Ebene, wird diese als neue Ebene im Bearbeitungsmodus geöffnet. Auch besteht die Möglichkeit schon gezeichnete Elemente aus anderen Ebenen oder Elemente aus den Projektarchiven in eine neue Ebene zu transferieren. Das Transferieren von schon bestehenden Elementen erleichtert die Reintegration von schon gelösten Problemstellungen aus anderen Projekten und Ebenen zusätzlich ermöglicht diese Vorgehensweise die Kombination von mehreren Ebenen oder Versionen zu einer neuen Ebene.

6.4 Die Arbeitsschritte während eines Projektes

Nach der Initialisierung eines Projektes kann mit der Projektarbeit begonnen werden. Die Projektarbeit beginnt wie bei der klassischen Projektarbeit mit der Sichtung der Unterlagen und den ersten Ideen und Entwürfen. Sobald jedoch der Computer für die ersten Zeichnungen benutzt wird, entfernt sich die Arbeit von einer klassischen Arbeitsweise, in der wild erste Entwürfe in ein CAD Programm übersetzt werden, später einer dieser Entwürfe weiterbehandelt wird und alle anderen Ideen über Bord geworfen werden, oder mit der Zeit in Vergessenheit geraten können. Durch die Möglichkeit der Versionierung können mehrere Ansätze gleichzeitig weiterentwickelt werden und zu jedem Zeitpunkt wieder aufgegriffen werden. Dadurch eröffnet sich dem Planer die Möglichkeit, alte Ideen einfach wieder weiterzubearbeiten oder Teile in neue Ideen übernehmen zu können. Bei klassischen Projekten geht dabei jedoch die Historie der Entwicklung meist nach jedem Arbeitsschritt verloren. Durch die Versionierung ist es nun dem Projektteam möglich, jeden einzelnen Schritt einer Entwicklung nachzuvollziehen. Dies bedeutet unter anderem auch, dass jede Idee und jede Lösung, auch wenn sie schlussendlich verworfen wurde, erhalten bleibt und für zukünftige Problemstellungen wiederverwendet werden kann. Dabei ist es gänzlich unwichtig, wo die Lösung oder Idee entwickelt wurde, denn durch das Plug-In kann sie einfach in das neue Projekt integriert und adaptiert werden.

Eine Arbeitsweise, welche auf Versionen basiert, zeichnet sich stark dadurch aus, dass sie zwischen verschiedenen Projekten, Versionen von Projekten und alternativen Projektentwicklungen kaum mehr unterscheidet. Das Projektergebnis ist somit nicht mehr wie klassisch ein Mix aus allen Erfahrungen eines Projektteams, sondern verbindet die Erfahrung und die Lösungen aus allen teilnehmenden Projektteams eines Versionsarchivs.

Im folgenden werden nun die einzelnen Funktionen der Erweiterung dargestellt.

6.4.1 Öffnen eines Projektes

Nach dem Start des Plug-Ins kann der Benutzer sein gewünschtes Projekt auswählen. Neben der Auswahl der Projekte findet sich ein Informationsbutton, welcher das Projektdatenblatt in einem Ansichtsfenster öffnet. Dieser Informationsbutton ermöglicht

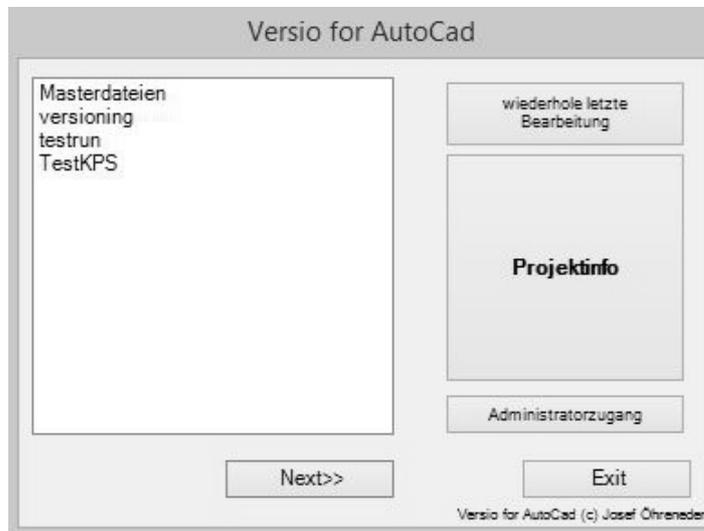


Abb. 11: Projektauswahlmaske des Plug-Ins (eigene Abbildung)

es einem Benutzer, schnell die jeweiligen Projektspezifika erfassen zu können und gegebenenfalls entscheiden zu können, ob dieses Projekt das Gesuchte ist. Ein weiterer Button am Ende der Projektauswahl erlaubt dem Benutzer, sofern er über Administratorrechte verfügt, die Vorlagen, die Benutzerdatenbank und die Projekte zu verwalten.

Nach der Auswahl des Projektes werden dem Benutzer alle Auswahlsets der Zeichnungen angezeigt, es besteht hierbei auch die Möglichkeit zur Auswahl einzelner

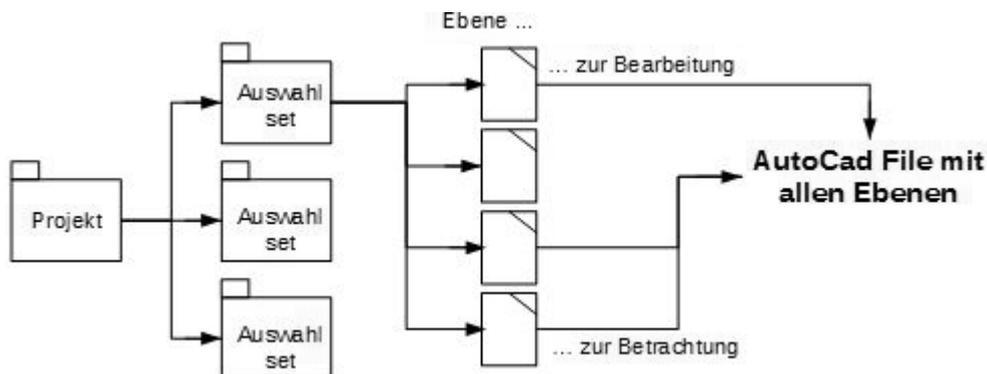


Abb. 12: Bearbeitung mit dem Plug-In starten (eigene Abbildung)

Ebenen unabhängig der Zugehörigkeit zu einem Auswahlsets zu wechseln. Bei der Wahl eines Auswahlsets kann der Benutzer, sofern er als Bearbeiter im Projekt hinterlegt ist, die zu bearbeitende und die zu betrachtenden Ebenen auswählen. Wird die Auswahl

ohne Auswahlset gewählt, erscheinen alle Ebenen, welche in einem Projekt existieren in alphabetischer Reihenfolge. Da die Ebenennamen mit einem Präfix ausgestattet sind, ist es dem Benutzer möglich die Funktionen und Inhalte der einzelnen Ebenen durch ihren Namen zu erkennen und sie so in seine Auswahl zu integrieren. Diese Vorgehensweise ist besonders dann interessant, wenn für eine neue Zeichnung noch kein Auswahlset besteht, diese aber schon existierende Ebenen mitbenützt. Aus dieser Auswahl kann ein neues Auswahlset generiert werden. Wird dies nicht sofort gemacht, kann dies auch während der Bearbeitung geschehen. Spätestens bei Beendigung der Arbeit wird von der Erweiterung ein neues Auswahlset generiert, sofern dies nicht durch den Benutzer erfolgte.

Nun kann der Benutzer die einzelnen Ebenen anwählen. Er hat die Möglichkeit eine Ebene als Bearbeiter, als Betrachter oder gar nicht auszuwählen. Ebenen welche gerade durch andere Benutzer zur Bearbeitung markiert wurden und Ebenen welche nur durch einen anderen Fachbereich bearbeitet werden können, können nur zur Betrachtung oder nicht geöffnet werden. Eine Bearbeitung ist bei solchen Ebenen ausgeschlossen, da es sonst zu Versionskonflikten kommen kann. Sobald mindestens eine Ebene zur Bearbeitung oder zur Betrachtung markiert ist, kann der Benutzer seine Auswahl laden und mit seiner Arbeit beginnen.

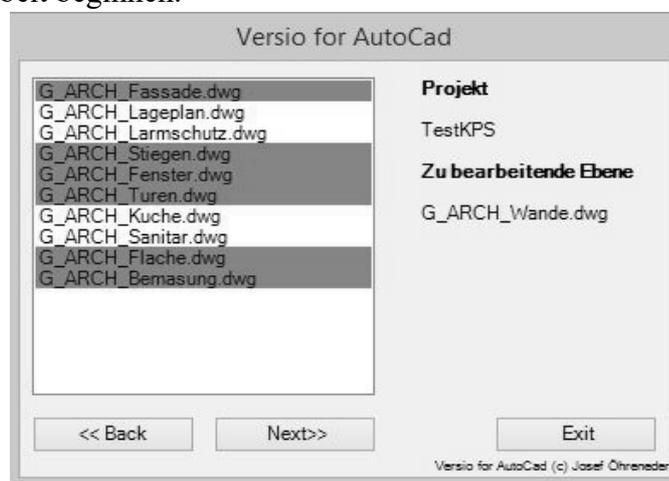


Abb. 13: „Ebenen zur Ansicht wählen“ Maske des Plug-Ins (eigene Abbildung)

Nach Auswahl der zu öffnenden Ebenen werden die betreffenden Ebenen aus dem Versionsarchiv geladen und an einem vordefinierten Ort lokal auf dem Computer des Benutzers gespeichert. Der Ort wird bei der initialen Installation der Erweiterung festgelegt und kann in weiterer Folge nur durch die Eingabe eines Passwortes in den Einstellungen der Erweiterung angezeigt werden beziehungsweise verändert werden. Dies ist dem Umstand geschuldet, dass die lokal gespeicherten Dateien aus dem

Versionsarchiv nicht durch den Benutzer verändert werden sollen, sondern eine Veränderung nur durch die Erweiterung durchgeführt werden soll. Sollte trotz dieser Vorkehrung eine Veränderung der lokalen Daten stattfinden, hat dies jedoch nur Auswirkungen wenn Ebenen zur Bearbeitung aus dem Archiv synchronisiert wurden. Alle anderen Dateien kommunizieren Änderungen nicht mit dem Versionsarchiv, diese werden sogar bei einem neuerlichem Öffnen neu aus dem Archiv geladen. Einzig die Änderungskontrolle kann durch eine Veränderung an den Dateien beeinträchtigt werden. Da die dafür nötigen Dateien nur lokal auf dem jeweiligen Computer beziehungsweise auf in dem jeweiligen Benutzerprofil gespeichert werden.

Nachdem alle benötigten Dateien vom Versionsarchiv lokal synchronisiert wurden, sucht das Plug-In nach den entsprechenden *.dxf Dateien im Unterordner Änderungskontrolle, das Dateiformat DXF ist eine im Gegensatz zum DWG Format, das verwendete Zeichnungsformat von AutoCad, quelloffene für Textverarbeitungsprogramme lesbare Schnittstelle für den Datenaustausch zwischen Produkten der Firma Autodesk und hat sich über die Zeit als Standardformat für den Datenaustausch verschiedener Hersteller etabliert. Sollten keine entsprechenden Dateien vorhanden sein, wurde die entsprechende Ebene noch nie mit diesem Benutzer synchronisiert und kann somit nicht auf Veränderungen überprüft werden. Findet das Plug-In jedoch eine entsprechende Datei, startet ein Textvergleichsprogramm und findet die Unterschiede der beiden *.dxf Dateien. Aus diesem Vergleich kann auf die Position von Veränderungen seit der letzten Synchronisation geschlossen werden. Diese Positionen werden nun in eine neue Ebene geladen und dort als Änderungswolke dargestellt.

Als einen letzten Schritt fügt das Plug-In die Ebenen zu den gewünschten Zeichnungen zusammen. Wobei alle zur Bearbeitung ausgewählten Ebenen als eigene Zeichnung geöffnet werden. Jede dieser neuen Zeichnungen enthält nun alle Ebenen als externe Referenzen, inklusive der neu erstellten Ebenen, welche die Änderungswolken enthalten. Dieses Verhalten ist dem Umstand geschuldet, dass die Referenzen einer Zeichnung nicht geöffnet werden können und dadurch jede zu bearbeitende Ebene als eigene Zeichnung geöffnet werden muss. Zusätzlich soll dieses Ladeverhalten die Bearbeiter von der Bearbeitung multipler Ebenen, durch längere Ladezeiten und größeren Ressourcenverbrauch, abhalten.

6.4.2 Änderungen überprüfen

Nachdem die gewünschten Daten geladen und geöffnet wurden, wird der Benutzer aufgefordert die Bereiche innerhalb der Wolken zu überprüfen, da dort Unterschiede zu der zuletzt für diesen Benutzer geladenen Ebenen vorhanden sind.

Wie zuvor schon kurz skizziert werden Änderungen zwischen der aktuellen Version und der zuletzt geöffneten Version mittels eines Vergleiches zwischen den jeweiligen *.dxf-Dateien durchgeführt. Warum werden nicht gleich die *.dwg-Dateien verglichen? Dies hat einen einfachen Hintergrund: DXF ist ein dokumentiertes Dateiformat welches speziell dazu dient Daten zwischen verschiedenen Programmen auszutauschen. Zwar kann es hierbei zu Informationsverlusten kommen, da spezielle Funktionen, welche in AutoCad zur Anwendung kommen, nicht durch den Syntax einer *.dxf-Datei wiedergegeben werden können. Dieser Informationsverlust stellt jedoch für ein Vergleichen von Veränderungen keine Probleme dar, da es egal ist ob Elemente als dynamischer Block oder eine andere spezielle Funktion in der Zeichnung innehaben, lediglich der Inhalt der Zeichnung ist für einen solchen Vergleich von Interesse und dieser wird in einer *.dxf-Datei gleichermaßen transportiert wie in einer *.dwg-Datei.

Dieser Vergleich wird für jede Ebene durchgeführt und jeweils als eigene Änderungsebene dargestellt. Stimmt eine Zeichnung mit der aktuellen Version überein wird keine Überprüfung durchgeführt und die nächste Ebene wird behandelt. Nach jeder

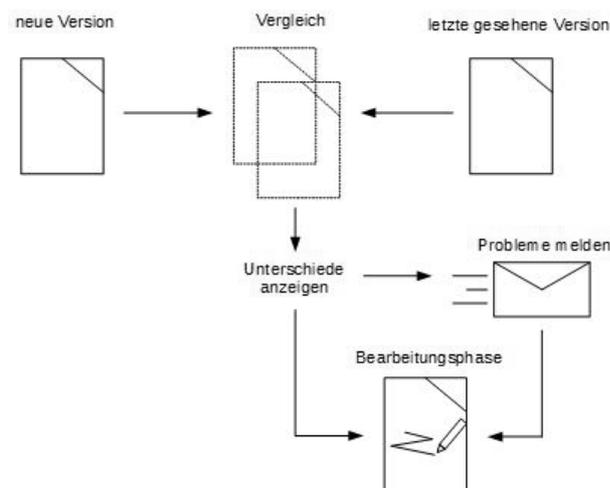


Abb. 14: Ablauf der Überprüfung auf Änderungen (eigene Abbildung)

Überprüfung wird dem Benutzer die jeweilige Änderungsebene angezeigt. Dieser hat nun die Möglichkeit die Änderungen zu überprüfen und gegebenen falls darauf zu

reagieren. Hat der Benutzer eine Änderung bestätigt, wird mit der nächsten Änderung fortgefahren.

Wenn der Benutzer eine Änderung entdeckt, welche für seine weitere Arbeit ein Problem darstellt, oder Fehler beinhaltet, bestätigt er nicht die Änderungen zur Kenntnis genommen zu haben, sondern teilt dem Ersteller seine Ansichten mit. Hierfür verwendet das Plug-In den Benutzernamen des letzten Bearbeiters der betreffenden Ebene, um aus dem Adressbuch dessen E-Mail Adresse zu beziehen. Gleichzeitig wird der Benutzer aufgefordert den betreffenden Bereich zu markieren, woraufhin ein Screenshot des ausgewählten Bereichs erstellt wird. Sobald die E-Mail Adresse gefunden ist, wird eine neue E-Mail erstellt, welche an den letzten Bearbeiter adressiert ist. Als Betreff wird der Ebenenname und die Versionsnummer verwendet. Der Screenshot wird als Anhang in die E-Mail kopiert und der Benutzer wird aufgefordert, seine Kritik zu formulieren. Nach dem Versenden der E-Mail wird der Benutzer gefragt ob die Kritik versendet wurde. Wird dies bestätigt, kann der Benutzer nun als Bearbeiter seine Arbeit aufnehmen. Nach einer Bestätigung einer Änderungsebene wird diese wieder entfernt und die *.dxf-Datei der letzten Version durch die Datei der neuen Version ersetzt.

Während der Bearbeitungsphase wird in einem, durch den Benutzer zu definierenden, Intervall eine Versionsprüfung aller zur Betrachtung geöffneter Ebenen durchgeführt. Wird eine aktuellere Version im Versionsarchiv entdeckt, wird der Benutzer durch eine Nachricht darauf hingewiesen. Die Nachricht wird solange angezeigt bis der Benutzer diese bestätigt, jedoch kann er sie in den Hintergrund legen bis er seine momentane Tätigkeit abgeschlossen hat. Wird die Nachricht, dass eine neue Version einer Ebene vorhanden ist, bestätigt, wird die neue Version geladen und eine Änderungsüberprüfung durchgeführt. Diese muss nach Abschluss, wie schon zu Beginn, vom Benutzer bestätigt werden. Diese Vorgehensweise hat 2 entscheidende Vorteile: Einerseits hilft sie den verschiedenen Benutzern sofort nach öffnen einer Ebene die letzten Änderungen, seit dieser die Ebene zuletzt betrachtet oder bearbeitet hat, visuell anzuzeigen. Dadurch ist es den Benutzern möglich alle neuen Arbeiten in Ebenen schnell und effektiv erfassen zu können. Andererseits ist eine dauernde qualitative Prüfung der Arbeit aller Beteiligten gewährleistet, da jeder Beteiligte kurz einen Blick auf die letzten Änderungen werfen muss.

6.4.3 Bearbeitungsphase

Nach dem Öffnen der Ebenen und der Überprüfung kann mit der Bearbeitung begonnen werden. Während dieser Phase werden neue Inhalte in die Ebenen eingearbeitet. Das Plug-In führt nun, in einem vorab definiertem Intervall, eine Sicherung der Zeichnung durch.

Sollte das Plug-In beziehungsweise AutoCad abstürzen, werden die letzten aktuellen Datensätze neu geladen und auf Fehler überprüft. Ganz ähnlich dem Wiederherstellungsprozess von AutoCad erkennt das Plug-In automatisch, dass es nicht richtig beendet wurde und führt eine Wiederherstellung der letzten Arbeitssitzung durch. Hierbei unterdrückt das Plug-In den Wiederherstellungsprozess von AutoCad, um auf die durch das Plug-In gesicherten Daten zurückgreifen zu können. Dies ist möglich da das Plug-In erkennt, dass die Dateien nicht durch das Plug-In an das Versionsarchiv zurückgegeben wurden. So geht das Plug-In bei einer Wiederherstellung genauso vor wie bei einer Wiederaufnahme der Arbeit, solange die bearbeiteten Ebenen nicht mit dem Versionsarchiv synchronisiert sind. Dadurch ist einerseits eine Wiederherstellung nach einem Absturz, sowie ein Weiterarbeiten zu einem anderen Zeitpunkt möglich.

Grundsätzlich sollten jedoch nach einem Arbeitstag alle bearbeiteten Ebenen wieder an das Versionsarchiv zurückgegeben werden. Denn nur so ist es anderen Bearbeitern auch möglich, Änderungen, welche durch eine Anpassung einer Ebene hervorgerufen werden, auch in ihre Planung zu integrieren. Zusätzlich ist es ebenfalls ein sicheres Vorgehen, um im Krankheitsfall anderen Benutzern die Möglichkeit zu geben, an einer Ebene weiterzuarbeiten. Das Plug-In fragt deshalb vor einer Beendigung durch den Benutzer ab, ob die Bearbeitung abgeschlossen werden soll oder nicht. Wird die Bearbeitung nicht abgeschlossen, wird diese Entscheidung im Logfile aufgezeichnet und der momentane Stand der Arbeit gespeichert. Hierbei geht das Plug-In genauso vor wie bei der laufenden Sicherung und vermerkt in den Startoptionen des Plug-In, dass eine Bearbeitung bereits geöffnet ist und nimmt die Bearbeitung beim Neustart wieder auf, bis die Bearbeitung durch den Benutzer abgeschlossen wurde.

Wie die Projektbearbeitung durchgeführt wird bleibt vollständig dem Bearbeiter überlassen, lediglich die Hintergrundprozesse werden durch das Plug-In gesteuert. Sobald die Bearbeitung einer Ebene abgeschlossen ist und diese wieder in das Versionsarchiv synchronisiert und der Lock aufgehoben wurde, können andere Benutzer

die Ebene zur Bearbeitung öffnen. Durch einen Lock an der Datei im Versionsarchiv wird es ermöglicht, dass keine weiteren Personen eine Ebene bearbeiten können.

Wird eine neue Ebene für eine Bearbeitung benötigt, kann der Bearbeiter diese direkt über das Plug-In erstellen. Hierbei wird eine neue Ebene im aktuellen Auswahlset generiert. Die Vorlagen werden aus der Vorlagen-Ebene bezogen, die Namensgebung wird aufgrund der Einstellungen der Präfixe und der frei gewählten Benennung erzeugt. Nach der Erstellung der Ebene, wird diese mit dem Versionsarchiv erstmals synchronisiert, mit einem Lock für den aktuellen Benutzer versehen und als neue Zeichnung geöffnet. Die neue Ebene wird sodann auch in alle, zur Bearbeitung geöffneten Ebenen referenziert. Die neue Ebene wird ebenfalls in dem aktuellen Auswahlset hinterlegt und kann somit bei einer neuerlichen Bearbeitung direkt mit allen anderen elementaren Ebenen des Auswahlsets geöffnet werden. Sollte eine neue Ebene auch Zugehörigkeit zu anderen Ebenen besitzen, welche sich nicht im aktuellen Auswahlset befinden, muss diese über einen Dialog mit allen betreffenden Auswahlsets verknüpft werden. Diese Verknüpfung wird anschließend in den betreffenden Auswahlsets gespeichert. Die neue Ebene kann somit von allen Benutzern im jeweiligen Auswahlset abgerufen werden.

Neben der Möglichkeit eine völlig leere neue Ebene zu erstellen, werden dem Benutzer noch zwei weitere Möglichkeiten geboten den Prozess der Zusammenarbeit zu fördern. So kann er einerseits Teile einer zur Bearbeitung geöffneten Ebene als neue Ebene etablieren, oder diese Teile in eine schon bestehende Ebene übertragen. Dabei muss die Ebene jedoch zur Bearbeitung freigegeben sein und darf somit durch keinen Beteiligten unter Bearbeitung stehen. Wird eine komplett neue Ebene erstellt, folgt das Plug-In den Arbeitsschritten wie bei der Erstellung einer neuen leeren Ebene, jedoch wird diese nach der erfolgten Übertragung der zu verschiebenden Elemente, zur Betrachtung im momentanen Arbeitsprozess hinzugefügt und in die gewählten Auswahlsets aufgenommen. Danach kann sie von einem Anderen Benutzer zur Bearbeitung bezogen werden.

Eine neue Ebene wird nicht automatisch, bei jedem Benutzer der eines der betreffenden Auswahlsets geöffnet hat, geladen. Um eine neu erstellte Ebene auch betrachten oder bearbeiten zu können, müssen die Benutzer ihre momentane Arbeit erst abschließen und das von ihnen bearbeitete oder betrachtete Auswahlset neu öffnen. Um eine Änderung an einer Ebene anzeigen zu lassen, welche der Benutzer bereits zur Betrachtung

bezogen hat, wird ihm die Möglichkeit gegeben, zu überprüfen ob es seit dem Beginn seiner Arbeit etwaige Änderungen an Ebenen gegeben hat, Änderungen werden in diesem Fall als neue Versionen einer Ebene verstanden. Sollte dies der Fall sein wird es dem Bearbeiter ermöglicht, von ihm ausgewählte Ebenen durch die neueste verfügbare Version zu ersetzen.

Wenn ein Bearbeiter einen Lock besitzt, kann dieser nur durch den Bearbeiter gelöst werden. Da die gesamte bisherige Arbeit, seit dem Lock, auf dem lokalen Laufwerk des Bearbeiters gespeichert ist, muss erst die Bearbeitung abgeschlossen werden, um einen Verlust der Daten zu verhindern. Um also einen Lock ohne den Bearbeiter zu lösen, muss auf dessen Computer mit dessen Benutzerkonto zugegriffen werden und die Bearbeitung abgeschlossen werden. Erst danach kann ein anderer Bearbeiter die vormals gesperrte Ebene weiter bearbeiten.

6.4.4 Bearbeitung abschließen

Sobald die Bearbeitung abgeschlossen ist, wird der Synchronisationsprozess gestartet. Der Synchronisationsprozess wird in mehreren Schritten vom Plug-In verwaltet.

Zuerst werden alle geöffneten Ebenen nochmals lokal gespeichert. Dann werden alle Verknüpfungen in den geöffneten Ebenen gelöst und die Zeichnungen bereinigt, sodass kein unnötiger Datenballast in den Zeichnungen zurückbleibt. Durch das Lösen der Verknüpfungen, werden wieder die reinen Ebenen, wie sie im Versionsarchiv hinterlegt sind geschaffen. Nachdem die bearbeiteten Ebenen erneut gespeichert wurden, werden die Daten als *.dxf-Dateien lokal im Versionsvergleichsordner gespeichert. Dieser Ordner, wie schon im Kapitel 6.4.2 „Änderungen überprüfen“ erklärt, stellt die Basis der Überprüfung von Änderungen dar und beinhaltet immer die letzten bearbeiteten oder betrachteten Ebenen in ihrer *.dxf-Dateiform. Sobald das Plug-In den Stand der Ebenen lokal gespeichert hat, wird der Bearbeiter aufgefordert, der zu synchronisierenden Version noch einen Versionstext anzuhängen, welcher grob die Bearbeitung durch den Bearbeiter beschreibt. Als einen letzten Schritt speichert das Plug-In das Logfile, welches seit Bearbeitungsstart mitgeloggt wurde und sendet alle Informationen an das Versionsarchiv. Nach Bestätigung der Übertragung der Daten wird der Bearbeiter über die erfolgreiche Übertragung informiert. Alle lokalen Daten, ausgenommen die gespeicherten Informationen im Versionsvergleichsordner, werden gelöscht, da diese nun Teil des Versionsarchivs sind.

Wenn die Bearbeitung abgeschlossen wird, können vom Benutzer zwei Vorgehensweisen gewählt werden. Zum einen die momentane Bearbeitung abschließen und eine neue Bearbeitung starten und zum anderen nur die Bearbeitung abschließen.

Wird die erste Variante gewählt, startet das Plug-In nach beendigter Synchronisation, den Projektöffnen Dialog und das Prozedere wie es schon im Kapitel 6.4.1 „Öffnen eines Projektes“ beschrieben wurde. Bei der zweiten Variante schließt das Plug-In nach beendigter Synchronisation die aktuellen Zeichnungen und ermöglicht entweder eine neue Bearbeitung oder die Publizierfunktion zu nutzen. Da während der gesamten Zeit der Verwendung des Plug-Ins in einer Log Datei die einzelnen Schritte aufgezeichnet werden, ist es dem Plug-In möglich, bei einem neuerlichen Start des Programms zum letzten Schritt zurückzukehren. Dabei muss darauf hingewiesen werden, dass das Plug-In nicht automatisch die Zwischensicherung übernimmt, sondern diese immer noch dem Benutzer überlassen ist. Lediglich bei einem Pausieren der Bearbeitung wird der Fortschritt der Arbeit lokal gespeichert, um bei einem Neustart angezeigt zu werden.

Durch die Notwendigkeit eine Bearbeitung abzuschließen, bevor eine neue Aufgabe übernommen wird, kann es nicht dazu kommen, dass sich ein Bearbeiter in mehreren Ebenen als Bearbeiter eintragen kann und somit deren Bearbeitung für Dritte sperrt, diese jedoch selber gerade pausiert hat. Wäre diese Vorgangsweise möglich, könnte diese dazu führen, dass einzelne Bearbeitungen pausiert werden und die Rückgabe der Daten beziehungsweise die Fertigstellung des zu bearbeitenden Bereichs bis zur Anfrage eines anderen Bearbeiters verzögert werden, da der Sperreigner aufgrund anderer Bearbeitungen, auf diese Bearbeitung vergessen kann. Wird er jedoch gezwungen die Bearbeitung zu beenden um eine neue Aufgabe zu übernehmen, kann die betroffene Ebene auch von anderen Beteiligten weiterbearbeitet und die neuen Daten alsbald möglich synchronisiert werden. Diese Beschränkung von nur einer parallelen Bearbeitung wird nicht nur auf ein Projekt angewendet, sondern aus den vorher genannten Gründen auf die Gesamtstruktur. Somit ist nur eine Bearbeitung an einem Rechner durch einen Benutzer möglich. Sollte ein Benutzer an mehreren Rechnern arbeiten, kann er auch mehrere Bearbeitungen durchführen, jedoch je Rechner immer nur eine und nie die selbe. Das Plug-In lässt eine Bearbeitung nur zu, wenn die betreffende Ebene nicht gesperrt ist, egal von wem diese Sperre erstellt wurde.

6.4.5 Daten publizieren

Sollen Daten Publiziert werden muss erst eine derzeitige Bearbeitung abgeschlossen werden. Das heißt, alle Ebenen welche publiziert werden sollen, dürfen weder von einem Selbst noch von Anderen bearbeitet werden.

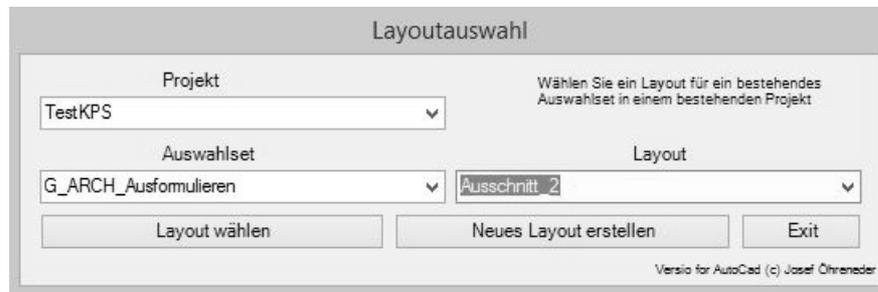


Abb. 15: Maske zur Auswahl des zu bearbeitenden Layouts (eigene Abbildung)

Über den Projektöffnen Dialog kann auch die Funktion Publizieren aufgerufen werden. Hierfür wird das zu publizierende Auswahlset gewählt. Sollten sich in diesem Auswahlset Ebenen befinden, welche noch in Bearbeitung sind, wird der Benutzer darüber informiert und auch von welchem Bearbeiter die jeweilige Ebene noch zur Bearbeitung geöffnet ist. Es ist nötig, dass keine Ebene des Auswahlsets unter Bearbeitung ist, da nur dadurch eine aktuelle Publikation möglich ist. Da jede Publikation zur Weiterverwendung durch Dritte verwendet werden kann, muss ein fertiger Arbeitsstand publiziert werden. Dies hat unter anderem rechtliche Gründe, da der Planverfasser für die Richtigkeit seiner Angaben haftet. Sollten also Teile eines Auswahlsets noch nicht den Erforderlichen Anforderungen entsprechen, könnte dies zu Schadenersatzforderungen führen.

Ist es dennoch erforderlich eine Publikation mit noch in Bearbeitung befindlichen Ebenen zu erstellen, wird im Gegensatz zu einer normalen Publikation auf allen Unterlagen, welche durch den Publikationsvorgang erzeugt werden, vermerkt, dass diese Publikation nur einen Zwischenstand der Planungsarbeit darstellt und keiner vollständigen Publikation des Standes der Arbeit entspricht.

Wird nun ein Auswahlset zur Publikation geöffnet, werden alle zugehörigen Ebenen durch den Benutzer gesperrt, damit die Ebenen in der Version der Publikation den selben der Publikation entsprechen. Auch wird das Layout des Auswahlsets wird gesperrt und als bearbeitbare Ebene geladen. Im Unterschied zu einer Bearbeitung wird bei der Publikation das Layout als zu bearbeitende Ebene geöffnet und die einzelnen zu

druckenden Ebenen mit dieser verknüpft. Nachdem der Benutzer den Änderungen zugestimmt hat, kann er das Layout anpassen und drucken. Jegliche Änderungen im Layout werden vom Plug-In erkannt und der Versionskontrolle übergeben. So ist es einem Benutzer auch möglich externe Daten in einem Layout anzeigen zu lassen und das Plug-In wird diese sodann unter Versionskontrolle stellen. Hierzu werden die Originalpfade eines externen Anhangs in den Arbeitsordner des Archivs kopiert und anschließend unter Versionskontrolle gestellt. Wird ein solcher Anhang verändert, kann er durch den Benutzer manuell neu geladen werden, das Plug-In erkennt einen neuen externen Anhang und stellt diesen wieder unter Versionskontrolle, dabei wird der vorhergehende Anhang mit dem selben Namen in der aktuellen Version ersetzt.

Sobald das Layout für die Publikation vorbereitet ist kann der Benutzer diese starten. Hierbei werden verschiedene Dateien erzeugt. Es wird eine *.pdf-Datei mit dem oder den vorhandenen Layouts, nach den gewählten Ploteinstellungen der Layouts, erstellt. Weiters wird eine E-Transmit-Datei erstellt, sowie die Layoutdatei an sich gespeichert. Am Ende des Publikationsvorgangs stehen dann also drei verschiedene Datensätze der selben Zeichnung zur Verfügung. Die *.pdf-Datei und die E-Transmit-Datei können nun per Plug-In an ein weiterverarbeitendes Programm gesendet werden, zum Beispiel einen PDF-Betrachter zum Plotten der Zeichnung oder an ein E-Mail Programm zum Versenden der Daten.

Wie mit der Publikation weiter verfahren wird liegt in der Verantwortung der Benutzers. Das Plug-In synchronisiert nach erfolgter Erstellung der Publikation alle 3 Datensätze mit dem Versionsarchiv und beendet den Publizierungsvorgang, gleich wie in Kapitel 6.4.4 „Bearbeitung abschließen“ beschrieben, als wenn eine Bearbeitung abgeschlossen würde.

7 Bearbeitung während eines Beispielprojektes

Um das Plug-In auf Herz und Nieren zu überprüfen, wird mit Hilfe einiger Testpersonen ein Testprojekt durchgeführt. Dabei wird einerseits die technische Umsetzung, als auch der Arbeitsablauf bei der Arbeit mit dem Plug-In, durch eine an die Realität angenäherte Situation, einer Überprüfung unterzogen.

7.1 Hypothese – Welche Ergebnisse werden erwartet

Durch die Verwendung von automatisierter Versionskontrolle kann, wie schon im Kapitel 6.2 „Die Planungsarbeit revolutionieren“ angesprochen, die Arbeitsweise während der Bearbeitung von Projekten positiv beeinflussen und nachhaltig verändert werden. Unter anderem wird dadurch auch eine effizientere Projektkommunikation ermöglicht, da jeder Bearbeiter, egal an welchem Ort er sich befindet, die selben Angaben und Grundlagen besitzt, wodurch sich Aufgaben besser verteilen lassen. Da ein Großteil der Organisationstätigkeiten vom Benutzer hin zum Plug-In verschoben werden, bleibt dem einzelnen Planer mehr effektive Zeit für die eigentlichen Planungsaufgaben.

- positive Beeinflussung der Arbeitsweise
- Verbesserung der gleichzeitigen Bearbeitung
- Verbesserte Fokussierung auf die Aufgaben
- aktuelle Daten immer und überall für jeden abrufbar
- optimierte Aufgabenverteilung und -abarbeitung
- Vermeidung von Flüchtigkeitsfehlern

Durch die Möglichkeit alle Zeichnungen auf beliebig viele Ebenen aufzuteilen, können mehrere Bearbeiter gleichzeitig an Plänen arbeiten und es kann eine Gleichzeitigkeit der Planungen verschiedener Gewerke stattfinden, bei welcher jeder Mitbearbeiter die aktuellsten Daten der anderen Bearbeiter zur Verfügung gestellt bekommt. Durch die Aufteilung der Zeichnungen in mehrere Ebenen entsteht nicht nur der Vorteil, dass eine Gleichzeitigkeit möglich wird, sondern auch dass Bearbeiter sich auf eine singuläre Aufgabe konzentrieren und nach deren Abschluss diese sofort mit allen Beteiligten teilen können. Diese Ebenen können dabei aus Themenbereichen oder räumlichen Einheiten bestehen. In jedem Fall stellen die Ebenen einen eingegrenzten

Themenbereich dar, welcher von einem Mitarbeiter auf einmal bearbeitet werden kann. Bisher finden alle Arbeiten meist nur an einer Datei statt, welche dadurch auch viele Aufgabenbereiche umfasst. Kann sich jedoch ein Bearbeiter auf genau eine Aufgabe konzentrieren, wird es ihm möglich effizienter zu arbeiten und vor allem qualitativ hochwertigere Ergebnisse zu produzieren.

Die Arbeitsweise mit einer automatisierten Versionskontrolle bietet dazu auch noch den Vorteil, dass Daten überall und zu jedem Zeitpunkt in ihrer aktuellsten Form vorliegen. Damit wird auch die Kommunikation zwischen den einzelnen Beteiligten aus verschiedenen Fachbereichen erleichtert, da jeder Mitarbeiter davon ausgehen kann, dass alle Beteiligten die neuesten Informationen besitzen. Somit muss sich ein einzelnes Teammitglied nicht mehr darum kümmern, dass alle Beteiligten auch die richtigen und aktuellsten Informationen besitzen. Diese Aufgabe wird vom Versionskontrollsystem für die Bearbeiter übernommen. Da sich die Projektmitglieder nicht mehr um die Verteilung kümmern müssen, haben sie mehr Zeit sich mit ihren Kernaufgaben zu befassen. Wobei in der selben Zeiteinheit mehr und intensiver an einem Projekt gearbeitet werden kann.

Durch diese Faktoren sollte sich die Aufgabenverteilung und -abarbeitung optimieren lassen, sowie die generelle Qualität von Ergebnissen gesteigert werden können. Letzteres ist dem Umstand geschuldet, dass durch die Begutachtung einer Ebene durch die anderen Teammitglieder, im Zuge der Prüfung oder Bearbeitung, die Fehlerquote deutlich gesenkt wird. Denn viele Flüchtigkeitsfehler entstehen, wenn ein Bearbeiter lange Zeit an ein und der selben Aufgabe arbeitet, ohne dass Kollegen diese kontrollieren. In diesem Fall stellt sich schnell eine sogenannte „Betriebsblindheit“ ein, welche durch eine ständige Querkontrolle durch Kollegen fast vollständig ausgeschlossen werden kann. Durch die ständigen Überprüfungen der neuen Änderungen durch Kollegen, sind die nötigen Änderungen meist nicht sehr umfangreich und können schnell durchgeführt werden. Sollte eine Überprüfung durch Dritte, wie in der heutigen Arbeitsweise üblich, jedoch erst relativ spät stattfinden, werden die Änderungen höchstwahrscheinlich komplexer und zeitaufwendiger sein, da sich aus einem einfachen Flüchtigkeitsfehler schnell eine Reihe von Folgefehlern ergeben können.

Wie die Aufteilung von Zeichnungen in mehrere Ebenen stattfindet, ist dabei jedem Projektteam freigestellt. Auch wenn Zeichnungen nur aus wenigen Ebenen, wie zum Beispiel nach Gewerken getrennt, bestehen sollten, ist weiterhin die Aktualität und die

gegenseitige Kontrolle gegeben, sowie ein gleichzeitiges Arbeiten aller Gewerke mit der selben aktuellen Information möglich. Unabhängig von den internen Strukturen, welche ein Projektteam verwendet, kann durch die Verwendung eines Tools zur automatisierten Versionskontrolle die Qualität des Ergebnisses und des Weges zu diesem verbessert werden. Durch die automatisierte Bereitstellung der aktuellsten Daten für alle Beteiligten, welche von einem Programm übernommen wird, werden die Mitarbeiter entlastet und können sich auf produktive Aufgaben konzentrieren. Dies sollte auch die Gesprächskultur innerhalb eines Projektteams positiv beeinflussen.

In der Überprüfung soll sich einerseits zeigen, dass sich durch die Verwendung von Versionskontrolle die Gleichzeitigkeit der Bearbeitung positiv auf ein Projekt auswirkt und andererseits die Bearbeiter dadurch besser ihren eigentlichen Teilaufgaben nachkommen können, wodurch die Kommunikation und Weiterentwicklung eines Projektes stark profitieren kann.

7.2 Parameter für den Test

Um die Hypothese zu überprüfen, wird das Plug-In einem Test unterzogen. Dabei soll die technische Umsetzung und die Theorie einer neuen Arbeitsweise geprüft werden. Mittels einer einfachen Aufgabestellung, mit welcher alle Befehle überprüft werden können und die Testteilnehmer zu einer neuen Arbeitsweise animiert werden sollen, wird sichergestellt, dass die zu testenden Vorgänge während eines Projektes abgehandelt werden können.

Der Arbeitsablauf, welchen die Testteilnehmer wählen, wird absichtlich nicht vorgegeben. Vielmehr sollen die Teilnehmer durch eine umfangreiche Einführung in das Plug-In und die damit möglichen Arbeitsabläufe animiert werden, den für sie besten Arbeitsablauf selbst herauszufinden. Hauptaugenmerk wird dabei auf die Durchführung der Aufgabe gelegt, da speziell die Arbeit und die Erfahrungen mit dem Plug-In getestet werden sollen. Den Probanden wird dabei nur die Aufgabenstellung und die Einschulung mit auf den Weg gegeben. Dadurch kann auch die Entscheidungsfindung, welche Arbeitsweise für diese Aufgabe angewendet werden soll, untersucht werden. Die Aufgabe wird dabei so gestaltet, dass die Teilnehmer sich nicht erst über eine Hierarchie einigen müssen. Schon durch die Verwendung des Plug-Ins und dessen Funktionalitäten sollte es den Probanden möglich sein ohne die Verwendung einer klassischen Hierarchie von Befehlsgeber und Befehlsempfänger auszukommen.

7.2.1 Testteilnehmer

Für die Durchführung des Tests werden drei Probanden ausgewählt. Die Anzahl der Teilnehmer erlaubt, dass sich während der Arbeiten die Zusammenarbeit zwischen den einzelnen Probanden gut entwickeln kann. Die Aufgabenstellung ist so gewählt, dass drei Bearbeiter diese leicht, in für alle Beteiligten annehmbarer Zeit abarbeiten und beenden können. Bei den Probanden handelt es sich um Architekten mit mehreren Jahren Berufserfahrung und besitzen Zusatzqualifikationen im Bereich der Bauphysik, örtlichen Bauaufsicht, der Projektleitung sowie der Projektentwicklung. Aufgrund ihrer Zusatzqualifikationen können die Teilnehmer ein großes Spektrum an möglichen Aufgabenbereichen abdecken und diese Erfahrung in die Überprüfung der Hypothese miteinbringen.

7.2.2 Aufgabenstellung

Um die Hypothese möglichst effizient und schnell zu überprüfen, ist die Aufgabenstellung darauf optimiert, die Zusammenarbeit mehrerer Personen und die Interaktion mit dem Plug-In in möglichst einfacher Art und Weise abbilden zu können.

Von den Teilnehmern soll dazu ein Regelgeschoß eines Hochhauses erstellt werden. Das Regelgeschoß soll durch Wohnungseinheiten mit Balkonen oder Loggien geprägt sein. Dabei soll von den Teilnehmern auch dargestellt werden wie eine mögliche Abfolge der anschließenden Geschoße aussehen könnte, zum Beispiel der Wechsel von Balkonen, Loggien und deren Orientierung. Um alle Teilphasen eines Projektes abzubilden, sollen

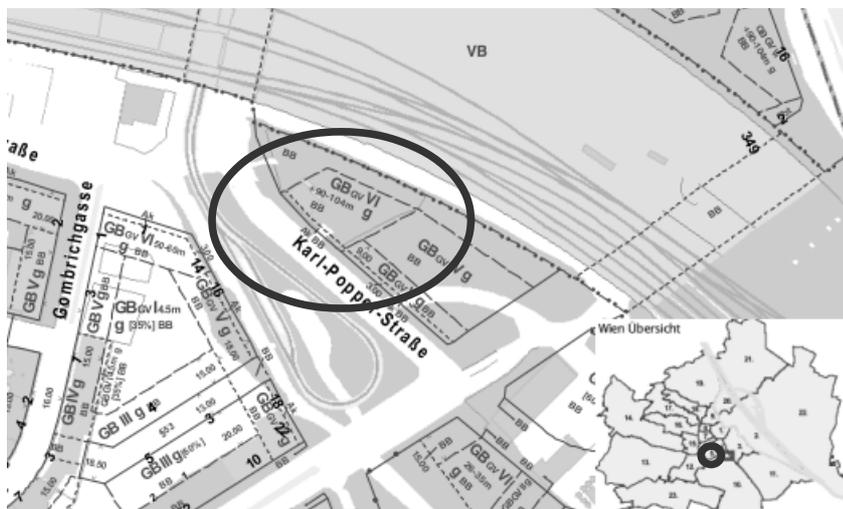


Abb. 16: Lageplan des Testprojektes in 1100 Wien, Ausschnitt aus dem Flächenwidmungsplan für Wien, <http://www.wien.gv.at/flaechenwidmung/public/start.aspx>, Zugriff am 30.06.2015

die Teilnehmer als ersten Schritt die Form des Grundrisses festlegen, um darauf aufbauend die Ausgestaltung vorzunehmen. Für die Aufgabenstellung wird ein definierter Bauplatz in Form eines Bebauungsplanes und dessen Bestimmungen den Probanden zur Verfügung gestellt wird.

Als Areal für die Entwicklung des Regelgeschossgrundrisses wurde ein realer Bauplatz im Entwicklungsgebiet rund um den neuen Hauptbahnhof Wien gewählt. Die ausgewählte Fläche, welche an der Karl-Popper-Straße zwischen der Endstation der Straßenbahnlinie D und der Bahntrasse liegt, umfasst rund 1.400 m² Grundfläche, welche in geschlossener Bauweise mit einer Höhe von 47 bis 61 Metern bebaut werden kann. Die Bruttogeschossfläche darf dabei 35.000 m² nicht überschreiten. Bei der Ausnutzung der maximalen Gebäudehöhe von 61 Metern können auf diesem Bauplatz bis zu 18 Obergeschoße mit Raumhöhen zwischen 2,5 und 3 Metern entstehen. Die Probanden müssen selbst klären welche Erschließungsweise für den Regelgeschossgrundriss gewählt werden soll. Diese muss jedoch so ausgelegt werden, dass eine mögliche Erschließung von anderen Nutzungen, wie Büro- und Geschäftsräumen, ebenfalls ermöglicht werden kann, da die Fläche als gemischtes Baugebiet ausgewiesen ist und eine Nutzung zu Wohnzwecken erst ab einem Niveau von 15 Metern erlaubt wird.

Die Aufgabenstellung soll nicht die Einzelarbeit der Teilnehmer fördern sondern die Teamarbeit verbessern. Aus diesem Grund werden keine weiteren Einschränkungen oder Vorgaben getroffen. Da sich dadurch das Team selbst über die Arbeitsteilung einigen muss, kann auch überprüft werden, ob das Team auf das Angebot der neuen Vorgehensweisen bei der Zusammenarbeit eingeht, oder im Projektablauf in die klassische Bearbeitung eines Projektes zurückfällt.

7.2.3 mögliche Arbeitsteilung durch Versionierung

Wenn gleichzeitig an einem Projekt gearbeitet wird und diese Arbeit durch eine Versionskontrolle unterstützt wird, können Zeichnungen auf fünf verschiedene Weisen auf diese Arbeitsweise vorbereitet werden. Diese Möglichkeiten werden den Testteilnehmern vorgestellt, wobei auf eine Wertung verzichtet wird, da es sich bei diesen nur um Arbeitsvorschläge handelt und nicht um Arbeitsanweisungen.

- Teilung in Ebenen nach konstruktiven Gesichtspunkten
Hierbei werden Ebenen für tragende Wände und konstruktive Elemente,

nichttragende Wände, Öffnungen, Fassade, Schächte, Erschließung, Haustechnik und der Gleichen erstellt.

- Teilung in Ebenen nach Aufgabenbereichen

Als Aufgabenbereiche kann man hierbei Raumeinheiten oder einzelne Räume, die Erschließung, die technische Erschließung oder auch die Fassadengestaltung verstehen.

- Teilung in Ebenen nach Bereichen

Hierbei wird eine Zeichnung weder nach technischen noch Aufgaben geteilt, sondern regional geteilt. Eine Zeichnung wird hierfür in einzelne Sektoren unterteilt welche zusammen den gesamt Plan ergeben. Dabei enthält eine jede Ebene alle nötigen Elemente ihres Sektors.

- Mix der Teilung aus den Vorhergegangenen

Bei dieser Teilungsart werden die Ebenen einer Zeichnung je nach Bedarf erstellt. So können Ebenen zur tragenden Struktur, der einzelnen Nutzungseinheiten und der Erschließungen erstellt werden. Diese können zukünftig auch noch weiter aufgeteilt werden.

- keine Teilung in Ebenen einzelner Zeichnungen

Hierbei handelt es sich um eine Arbeitsweise, welche vornehmlich verwendet wird, wenn eine Zeichnung nur durch einen Benutzer bearbeitet wird. Sie erleichtert diesem die Bearbeitung, da er seine Zeichnung nicht mit anderen Benutzern zur Bearbeitung teilen muss.

7.2.4 Testablauf

Durch eine kurze Einschulung in die Funktionalitäten des Plug-Ins und die damit verbundenen Handlungsweisen, wie in Kapitel 6 „Arbeiten mit Versionen“ dargelegt, wird den Teilnehmern an dem Testprojekt die Aufgabenstellung näher gebracht. Daraufhin wird den Probanden, mittels eines eigens dafür zurechtgelegten Projekts, die Möglichkeit gegeben sich mit dem Plug-In und den möglichen Arbeitsweisen vertraut zu machen. Durch diese Phase des Ausprobierens der Funktionen und der Möglichkeiten des Plug-Ins können sich die Teilnehmer an die Funktionalitäten gewöhnen und machen erste Erfahrungen mit den möglichen Arbeitsweisen. Aufgrund dieser Erfahrungen werden Entscheidungen über die Art und Weise wie das Team zusammenarbeiten möchte und die Handhabe des Plug-Ins während des Tests erleichtert.

Nachdem alle Testteilnehmer sich mit der Arbeitsumgebung vertraut gemacht haben, wird das Testprojekt gestartet. In einem ersten Schritt ist das Team aufgefordert sich über die grundsätzliche Ausformulierung des Grundrisses Gedanken zu machen. Im weiteren sollte sich das Team über eine Vorgehensweise bei der Ausarbeitung der Aufgabe einigen und die Ausarbeitung beginnen. Zum Abschluss wird noch eine kurze Diskussion über den Test und seine Ergebnisse statt finden.

Bei ihrer Arbeit werden die Teilnehmer nur in technischer Hinsicht vom Autor unterstützt, jedoch nicht bei Fragen zum Projektfortschritt. Dadurch wird versucht, die Teilnehmer nicht durch die Meinung des Autors zu beeinflussen, damit mögliche Ergebnisse nicht verfälscht werden. Der Autor übernimmt also während des Testprojekts nur die Rolle eines Beobachters und technischen Betreuers. Durch diese Beobachtung der Probanden bei ihrer Arbeit und ihrer Kommunikation im Team während des Tests, sowie deren Kommentare bei der abschließenden Diskussion kann die Hypothese einer gründlichen Überprüfung unterzogen werden.

7.3 Beobachtungen während der Bearbeitung

Um während des Testprojektes flüssiger arbeiten zu können, versuchen sich die Probanden vorab an einem Probeprojekt. Hierbei erlernen die Teilnehmer die Handhabung des Plug-Ins und können Erfahrungen mit den vorgeschlagenen Maßnahmen zur Arbeitsteilung machen. Diese erste Phase des Tests ist nicht nur dafür gedacht, die Arbeit mit dem Plug-In zu üben, sondern ist sogleich der Test der Funktionen, damit während des eigentlichen Testlaufes das Auftreten von technischen Störungen weitestgehend ausgeschlossen werden kann. Ebenso ist diese Phase davon geprägt, dass sich die Probanden ein Bild der möglichen Bearbeitungsweisen, welche das Plug-In bietet, machen können, um eine gemeinsame Herangehensweise an die Strukturierung für das Testprojekt zu finden.

Da das Plug-In bis zu diesem Zeitpunkt noch nie durch unerfahrene Benutzer und gleichzeitig von mehr als einer Person bedient wurde, kommt es im Zuge dieser ersten Phase zu einigen Störungen, welche größtenteils schon während der Einarbeitung der Teilnehmer gelöst werden können. Nach Abschluss der Einarbeitungsphase wird mit den Probanden zusammen geklärt, welche technischen Probleme noch zu lösen sind, um einen erfolgreichen Test durchführen zu können. Speziell handelt es sich dabei um kleinere Maßnahmen die eine flüssigere Bearbeitung ermöglichen, sowie eine bessere

Kommunikation zwischen Plug-In, Benutzern und dem Versionskontrollsystem. In dieser Nachbesprechung wird auch die Vorgehensweise und Strukturierung der Arbeiten während des Testprojektes besprochen. Da alle Teilnehmer nun einen Einblick in die Möglichkeiten haben und diese selbst ausprobieren konnten, bedarf es nur einer kurzen Diskussion wie an die Aufgabenstellung heranzugehen ist und wie die Aufgaben während des Tests verteilt werden sollten.

Das Team einigt sich darauf, dass ein jedes Mitglied zu Beginn eine eigene Ebene bekommt, in welcher es seine Ideen für die Ausformulierung des Regelgeschossgrundrisses eintragen kann. Nachdem alle Teammitglieder ihren Entwurf gespeichert haben, wird eine kurze Besprechung klären welche Elemente und Ideen eines Entwurfs weiterverwendet werden sollen. Daraufhin sollen diese Elemente in eine gemeinsame Ebene übertragen werden, welche die Grundlage für alle weiteren Arbeiten darstellt. Sozusagen ein Grundentwurf welcher einer Ausdifferenzierung zu Grunde liegt und bei Bedarf noch adaptiert werden kann. Für die Aufteilung der einzelnen Ebenen einigen sich die Teammitglieder darauf, dass bei einer weiteren Ausarbeitung zuerst nur Ebenen entstehen sollten welche dem momentanen Bearbeitungsstand entgegenkommen. So wird vorgeschlagen, dass konstruktive Elemente, die Erschließung, die Fassade und die Wohnungstrennwände sowie die Innenwände eine eigene Ebene erhalten. Mit Fortschreiten der Arbeiten sollen dann je nach Bedarf neue Ebenen erstellt werden, wie die Sanitäreinrichtungen oder Schächte und technische Versorgungseinrichtungen.

Das Team möchte ebenfalls versuchen solange wie möglich aus den, zu Beginn erstellten, benutzereigenen Ebenen heraus zu arbeiten. Bei dieser Vorgehensweise sollen die erstellten Elemente eines Arbeitsschrittes in die entsprechenden Ebenen exportiert werden, um somit allen Teilnehmern beim Beginn der Arbeiten eine möglichst reibungsfreie Bearbeitung zu erlauben. Dadurch müssen die einzelnen Bearbeiter, gerade am Beginn wo viele neue Elemente erschaffen werden, nicht jedes Mal die Bearbeitung unterbrechen, um die Ergebnisse eines Arbeitsschrittes mit dem Team teilen zu können. Das Team verspricht sich von dieser Vorgehensweise gerade zu Beginn ein schnelleres Vorankommen der Bearbeitungstätigkeiten.

Nach Abschluss der ersten Phase werden die aufgetretenen Störungen sorgfältig und nachhaltig bereinigt und Anpassungen an die, von den Probanden vorgeschlagene, Vorgehensweise durchgeführt, um beim eigentlichen Test bereits einzufließen.

7.3.1 Entwurfsphase

Da den Beteiligten bereits die Aufgabenstellung bekannt ist und diese im Umgang mit dem Plug-In auch schon bewandert sind, starten alle Probanden mit der in der vorhergehenden Phase abgemachten Weise. Dabei erstellt jeder Proband seine benutzereigene Ebene, in welcher er mit der Darstellung seiner Ideen für die Grundrissform und Erschließungssystematik beginnt. Nachdem jeder Proband eine erste Version seiner Ideen gespeichert hat, beginnen die Teilnehmer die Ebenen der Kollegen unter ihre eigenen Entwürfe zu legen und entwickeln, durch den Einfluss der anderen Ideen, ihren eigenen Entwurf weiter. Dabei bewegt sich die Bandbreite der vorgeschlagenen Entwürfe von einer klassischen Blockrandbebauung mit Innenhof, einer Bebauung der zur Straße hin orientierten Grundstücksgrenzen in L-Form und einer T-Form welche zur Straße und zur Bahnlinie einen Freiraum belässt. Die Erschließung reicht von einem Mittelgang bis hin zu einem Laubengang welcher zu einem Innenhof geöffnet ist.

Durch den direkten Vergleich aller Entwürfe auf dem Bildschirm, entwickelt sich rasch eine Diskussion im Team, welche Elemente der Entwürfe weiterentwickelt werden sollen und wie man dadurch zu einem Gesamtentwurf gelangen möchte. Sobald sich die Testteilnehmer auf einen Gesamtentwurf geeinigt haben, beginnen die einzelnen Bearbeiter die Teile ihres Entwurfes an den Konsens des Gesamtentwurfes anzupassen. Aufgrund der andauernden konstruktiven Kommunikation während dieser Phase konnten sich alle Teilnehmer darauf einigen, dass jeder Entwurf Teile enthält welche weiter verfolgt werden sollen. So einigt man sich auf eine L-Form entlang der Straße mit Laubengang, welcher sich zur Bahnlinie hin zur entstehenden Freifläche öffnet und auf eine Lärmschutzwand, welche im Bereich des Bahndamms die Schallemissionen auf die Freifläche und den Laubengang eindämmen soll. An den Laubengang sollen ebenfalls Gemeinschaftszonen als Loggien anschließen. Zur Straße werden alternierend Balkone und Loggien den Baukörper auflockern.

Das Team erschafft zu diesem Zweck mehrere neue Ebenen, welche die Entwurfsteile der einzelnen Beteiligten aufnehmen sollen. Die Erstellung der Ebenen wird von einem Teammitglied übernommen, dabei werden Ebenen für die Gebäudeform und die zur Straße orientierten Balkone und Loggien, die Aufteilung der Wohnungen, die Erschließung mit den Gemeinschaftszonen, sowie für die Lärmschutzeinrichtungen gegen den Bahnlärm erstellt. Auch wird zu diesem Zeitpunkt auch das erste Auswahlset

„Entwurf“ erstellt, damit die Teammitglieder einen besseren Überblick über den Fortschritt des Entwurfes haben. Die einzelnen Probanden beginnen nun damit die relevanten Elemente in ihren Ebenen weiter zu Optimieren und anschließend in die neuen Ebenen zu exportieren.

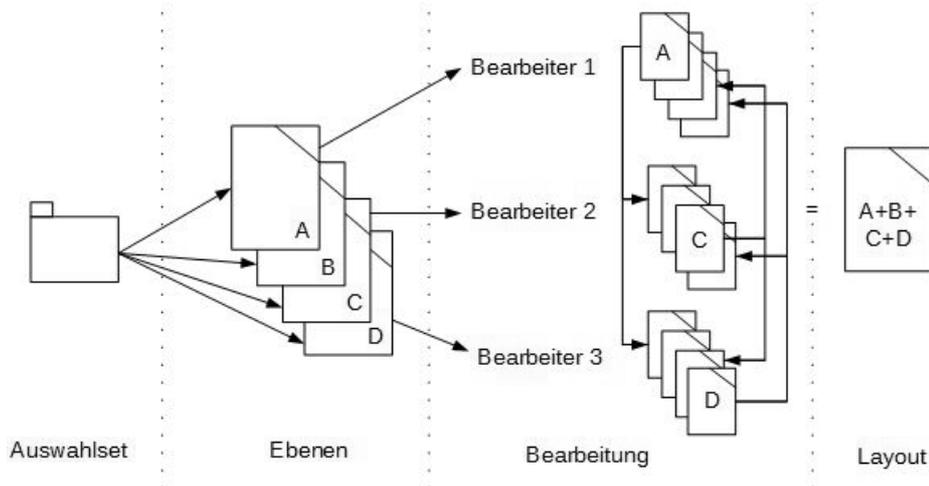


Abb. 17: der Arbeitsablauf bei der Arbeit mit unterschiedlichen Ebenen (eigene Abbildung)

Nach diesem Schritt werden die benutzereigenen Ebenen, entgegen der Annahme diese länger zu verwenden, nicht weiterverwendet, sondern in den neuen Ebenen weitergearbeitet. Dabei beginnen die Teilnehmer diejenigen Ebenen zu bearbeiten, welche vornehmlich durch ihre Planung beeinflusst wurden. Die Bearbeitung der Ebene, welche die Lärmschutzmaßnahmen enthält, wird in diesem Schritt vorerst hinten angestellt, da sich die Arbeiter auf die Fassadengestaltung, die Aufteilung der Wohnungseinheiten und die Erschließung konzentrieren. Die Bearbeitung der Fassade kann dabei nur in Teilschritten erfolgen, sodass der Arbeiter in den Bearbeitungspausen, in welchen er auf den Input des Bearbeiters der Wohnungseinheiten wartet, sich der Weiterentwicklung der Schallschutzwand und des Freiraumes widmen kann.

Da alle drei Probanden die Ebenen jederzeit aktualisieren können, ist zu bemerken, dass sie sehr schnell auf Entwicklungen in anderen Ebenen reagieren. Dabei wird zeitweise auf verbale Kommunikation ganz verzichtet, nur wenn Änderungen mit der eigenen Planung kollidieren wird kurz mit dem Ersteller erörtert wie dieses Problem beiderseitig gelöst werden kann. Es ist ebenfalls zu bemerken, dass die Arbeiter ihre einzelnen Aufgaben sehr konsequent durchführen und bei Beendigung dieser sofort die Änderungen an die Versionskontrolle übergeben, damit ihre Kollegen ebenfalls schnellstmöglich ihre Arbeit fortsetzen oder fertigstellen können. In gewissen

Situationen wird von den Teammitgliedern sogar um die Übermittlung von noch nicht vollständig erledigten Aufgaben gebeten, um den Arbeitsablauf noch zusätzlich zu beschleunigen. Durch die beschleunigte Arbeitsweise kann das Team in relativ kurzer Zeit den Entwurf abschließen und sich nun um die detailliertere Auseinandersetzung mit den einzelnen Bereichen des Regelgeschosses kümmern.

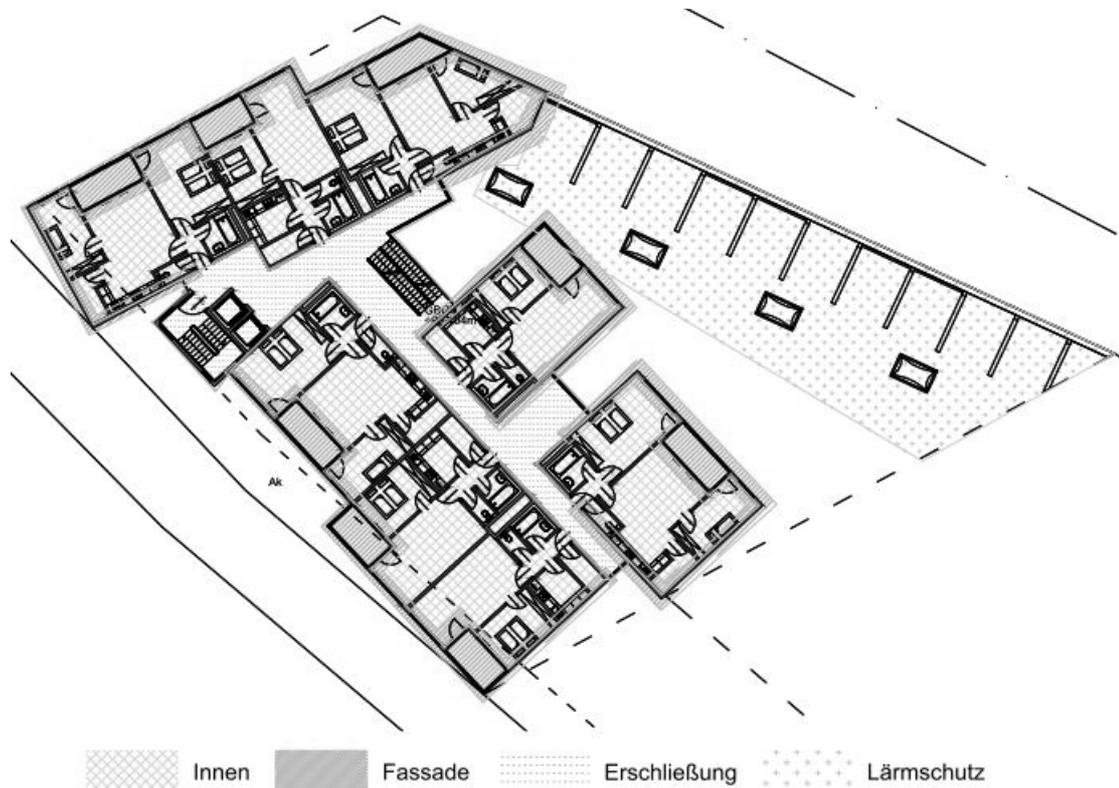


Abb. 18: Der Entwicklungsstand vor Beginn der Phase III mit eingefärbten Ebenen (eigene Abbildung)

7.3.2 Ausarbeitungsphase

In der dritten Phase des Tests ändert sich die Zusammensetzung der Ebenen noch einmal grundlegend. Auf Grundlage der Ebenen für den Entwurf erstellt das Team nun je nach bedarf neue Ebenen beginnend mit einer Ebene welche die Tragstruktur aufnimmt, einer Ebene für die Wohnungswände, einer Ebene für die Fassade, einer Ebene für den Laubengang und einer Ebene für die Stiegenhäuser inklusive der Aufzugsanlagen, sowie einer Ebene welche die Sanitäreinrichtungen und Ausstattungselemente enthält.

Die Probanden setzen in dieser Phase die etablierten und konstruktiven Methoden aus der letzten Phase fort und beginnen mit der Ausdifferenzierung der einzelnen Planungsbereiche. Schritt für Schritt wird dabei das Regelgeschoss weiterbearbeitet und der Endzustand immer deutlicher sichtbar. Wobei das Team zuerst die Innenwände, die Fassade und den Laubengang fertigstellt. Dabei kommt es kaum noch zu Schnittstellenproblemen zwischen den einzelnen bearbeiteten Ebenen, da die Entwurfsebenen als Unterlage die Orientierung aller Beteiligten erleichtern. Durch die Verwendung der Funktion „Wiederaufnahme der letzten Bearbeitung“ wird es den

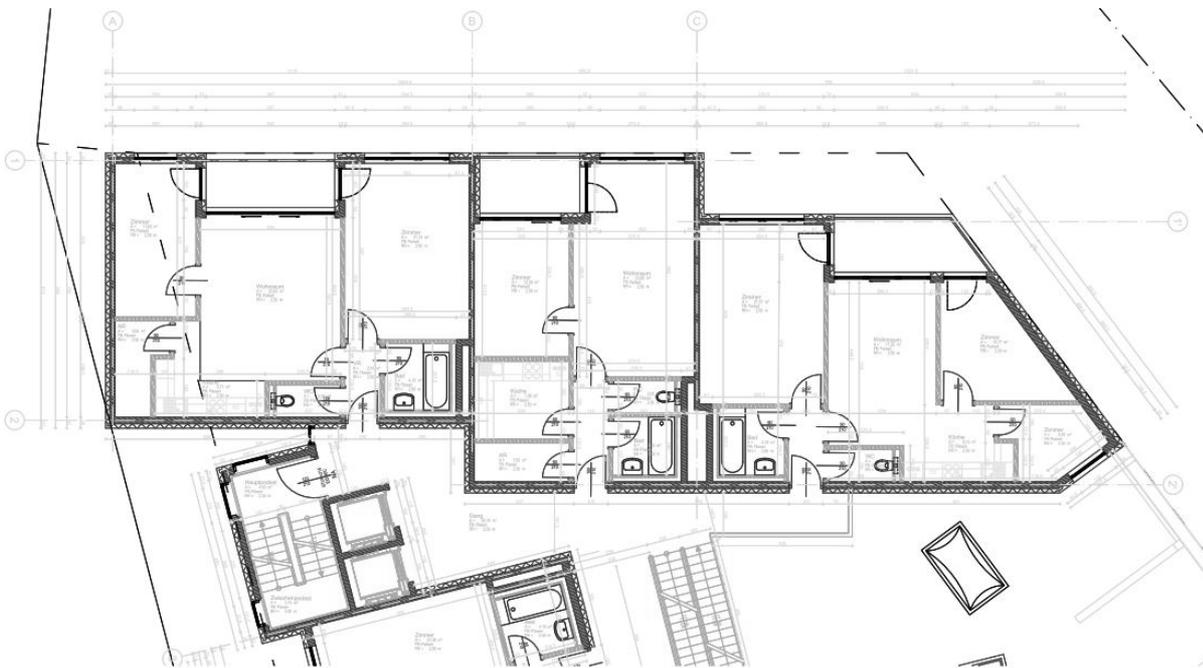


Abb. 19: Planausschnitt Endzustand nach Abschluss der Phase III mit eingefärbten Ebenen (eigene Abbildung)

Teilnehmern erleichtert Arbeitsschritte in kurzen und regelmäßigen Abständen miteinander zu teilen. Bei jeder Datenübergabe an das System informiert der Bearbeiter seine Teamkollegen über die Aktualisierung, sowie über seine nächsten Schritte. Wobei das Team darauf achtet, dass Aufgaben erst begonnen werden wenn die nötigen Vorarbeiten abgearbeitet sind. So wird ein Kollege darauf hingewiesen, dass er vorab eine andere Aufgabe erledigen soll, da für die angestrebte Aufgabe noch einige Elemente fertiggestellt werden müssen.

kommt es nur selten zu groben Missverständnissen, treten trotzdem welche auf lassen sich diese sehr schnell auflösen, da Änderungen sehr schnell erstellt und erneut besprochen werden.

Der gesamte Test in drei Phasen dauerte insgesamt 4 Abende, wobei die Phase II und besonders die Phase III durch intensive Arbeit der Probanden geprägt war. Große technische Störungen traten dabei nicht auf, lediglich in der Phase I, in welcher auch die technischen Grundlagen des Plug-Ins getestet wurden, traten einige Störungen auf, welche für die Durchführung der Phase II und III behoben werden mussten. Darunter fielen Probleme mit der Erstellung und Verteilung von Sperrern auf Dateien, generelle Probleme bei der Kommunikation zwischen Versionskontrollsystem und Benutzer und Probleme mit der Darstellung der auszuwählenden Ebenen. In der Phase II traten noch vereinzelt Probleme mit der Serverkommunikation auf, welcher den Zugriff zur Übertragung zeitweise verweigerte, dieses Problem konnte jedoch vor der Phase III gelöst werden.

7.3.3 Kommunikation auf einer neuen Ebene

Die Kommunikation zwischen den Testteilnehmern zeigte sich als äußerst konstruktiv besonders bei Abstimmungen zwischen den einzelnen Bearbeitungsschritten. Die Probanden standen während der Bearbeitung von Aufgaben und bei der Überprüfung der Tätigkeiten der Kollegen in regem verbalem Kontakt und konnten so effizient und rasch auf neue Inhalte und Ideen bei der Erstellung des Regelgeschossgrundrisses reagieren. Durch die Größe des Teams wurden Entscheidungen durch alle Bearbeiter demokratisch getroffen. Es musste niemand aus dem Team die Rolle als Projektleiter übernehmen, da allen Teilnehmern zu jedem Zeitpunkt die selben Informationen zur Verfügung standen und sich alle Teilnehmer im selben Raum befanden, dadurch waren Entscheidungen in kurzen Diskussionen möglich.

Für eine Anwendung in der Realwirtschaft heißt dies jedoch nicht, dass auf einen Projektleiter verzichtet werden kann, sondern lediglich, dass dieser sich mehr auf die Verwaltung, Verteilung von Aufgaben, Kundenkontakt und Prüfung der Ergebnisse konzentrieren kann. Und nicht mehr in der bisherigen Weise auch in den planerischen Entscheidungsprozess zwischen mehreren Teammitgliedern als Mittelsmann eingebunden werden muss.

Die Häufigkeit von Besprechungen steigen bei dieser Arbeitsweise zwar. Durch die kürzeren Abstände und durch den Umstand, dass jedes Teammitglied die aktuellen Entwicklungsschritte eines Projektes mitverfolgen kann, können die Besprechungen im Wesentlichen kürzer und konstruktiver gehalten werden. Dabei konnte, zumindest während des Tests, beobachtet werden, dass bei den kurzen Besprechungen weniger über Fehler gesprochen wurde, sondern die Gesprächsteilnehmer vielmehr über einen konstruktiven Weg zur Erfüllung des gemeinsamen Ziels berieten. Unterstützt wird diese Gesprächskultur noch zusätzlich durch die wiederholten Überprüfungen der Änderung durch die Teammitglieder beim Beginnen und Aktualisieren einer Bearbeitung, wobei etwaige Unstimmigkeiten und dadurch resultierende Änderungen in einem für alle Teilnehmer überschaubaren Maß geblieben sind. Aufgrund der Überschaubarkeit der Änderungen konnten diese in relativ kurzer Zeit umgesetzt werden. Durch die laufenden Überprüfungen ist zu beobachten, dass die Zahl der Unstimmigkeiten zwischen den einzelnen Bearbeitungen zurückgeht und die Frustration bei der Richtigstellung stark abnimmt. Schon nach nur kurzer Einarbeitungsphase scheint das System der gegenseitigen Kontrolle und Überprüfung der einzelnen Aufgaben zur allgemeinen Qualitätssteigerung beizutragen.

7.4 Kommentare der Testteilnehmer

Nach Abschluss der dritten und letzten Phase des Testprojektes findet eine Nachbesprechung mit allen Teilnehmer statt. Dabei wird über den Ablauf des Tests und seine Ergebnisse, sowie über das Plug-In und sein Potential diskutiert. Die Testteilnehmer sehen großes Potential im getesteten Plug-In und stehen einer Weiterentwicklung und auch einer Implementierung in die Architekturarbeit positiv gegenüber. Die Funktionen des Plug-Ins wurden von den Probanden als funktional und für die angewendete Arbeitsweise von Vorteil empfunden.

Die Möglichkeiten welche durch die Verwendung eines Versionskontrollsystems entstehen, werden bei einer Zusammenarbeit mit mehreren Beteiligten, insbesondere bei der Zusammenarbeit unterschiedlichster Gewerke, als positiv bewertet. Dabei wird der Umstand, dass in jedem Fall jeder Bearbeiter die letztgültigen Unterlagen zur Verfügung gestellt bekommt, besonders hervor gehoben. Die Bearbeitung einer älteren Version wäre zwar weiterhin möglich, muss aber vom Benutzer bewusst gewählt werden, dadurch ist es nahezu ausgeschlossen, dass Bearbeiter aus Unwissenheit an unterschiedlichen Entwicklungsständen des Projektes arbeiten, vielmehr arbeiten alle

gleichzeitig am selben Stand und können sich dadurch besser Koordinieren. Die Fähigkeit die gesamte Geschichte einer Planung digital aufzeichnen zu können und bei bedarf zurückzuspulen oder an einem beliebigen Punkt eine Alternative Entwicklung zu starten kann zu interessanten und neuen Arbeitsweisen und Herangehensweisen bei der Bearbeitung eines Projektes führen. Das dem Benutzer dadurch gleichzeitig die Pflicht der Pflege der digitalen Ablage, insbesondere der Erstellung und Verwaltung von Planständen und dem archivieren von Zwischenschritten, abgenommen wird, wird als zusätzliches Plus gesehen.

Trotz der Funktionalität der einzelnen Befehle des Plug-Ins wurde die Vorgehensweise bei der Übergabe von Daten an das Versionskontrollsystem als umständlich und eher schädlich für die kontinuierliche Bearbeitung einer Ebene empfunden. Trotz der Möglichkeit die letzte bearbeitete Ebene und die dafür ausgewählten Ebenen zur Betrachtung zu laden, muss immer noch die Bearbeitung beendet und wieder reinitialisiert werden. Die Probanden würden hierfür die Möglichkeit begrüßen, einen Stand der Bearbeitung während diese noch läuft an die Versionskontrolle übergeben zu können. Weiters wird eine Möglichkeit gefordert während einer Bearbeitung zusätzliche Ebenen zur Betrachtung hinzuladen und diese auch entfernen zu können. Da bei der Extraktion von Elementen in eine neue Ebene diese nur mit einem Neustart wieder geladen werden kann und sich auch während der Bearbeitung die Notwendigkeit eine zusätzliche Ebene anzeigen zu lassen ergeben kann.

Das Projektteam hat zwar das Benachrichtigungssystem, welches in das Plug-In integriert war, hauptsächlich aufgrund der kurzen Kommunikationswege beim Test, wo verbale Kommunikation einfacher war, nicht genutzt. Um jedoch die Möglichkeiten der Zusammenarbeit, gerade bei größeren Projektteams mit mehreren involvierten Gewerken, besser nutzbar machen zu können, wird von den Probanden empfohlen eine Adaptierung des Benachrichtigungssystem vorzunehmen. Dabei sollen die Benachrichtigungen nicht wie im getesteten Plug-In direkt an den letzten Bearbeiter versendet werden, sondern an eine Zentrale Stelle. Dieses zentrale Benachrichtigungssystem könnte zusätzlich mit der Möglichkeit verbunden werden neue Aufgaben zu erstellen und diese zu verwalten. Die Aufgaben könnten dann, wie bei einer To-Do-Liste, durch die nächsten freien Mitarbeiter übernommen, ausgeführt und bei der Beendigung dieser abgeschlossen werden. Speziell bei größeren Teams wäre so eine ähnliche Arbeitsteilung, wie sie das Testteam angewendet hat, möglich.

7.5 Überprüfung der Hypothese

Der Test konnte zeigen, dass sich durch die Möglichkeiten der gleichzeitigen Bearbeitung ein Projekt sehr gut entwickeln kann. Die Arbeitsweise unterscheidet sich dabei klar von einer klassischen Arbeitsweise, da vor Beginn der Arbeiten eine Ebene gewählt werden muss in welcher gearbeitet werden soll und durch die zu betrachtenden Ebenen während der eigenen Bearbeitung durch das Team verändert werden können. Dennoch funktioniert diese Arbeitsweise schon nach kurzer Einarbeitung und Umstellungsphase optimal und gibt den Beteiligten die Möglichkeit sich auf singuläre Aufgaben besser zu konzentrieren, sowie diese besser und schneller abzuschließen. Da die einzelnen Ebenen kleinere Aufgabenpakete umfassen können, wird der Zeitintervall, bis eine Änderung mit den Kollegen geteilt werden kann, stark verkürzt. Dies beeinflusst wiederum den Arbeitsfortschritt der Kollegen, welche dadurch schneller auf Entwicklungen in einem Projekt reagieren und entsprechend agieren können. Es war während des Tests zu beobachten, dass dies bei den Probanden auch zu einer effizienteren Kommunikation bei der Besprechung der Vorgehensweise führte.

Durch die ständige Gegenkontrollmöglichkeit konnten Flüchtigkeitsfehler und Schnittstellenprobleme zwischen den einzelnen Ebenen schneller beseitigt werden. Die Begutachtung von Änderungen in einer Ebene hilft nicht nur der Fehlersuche sondern trägt mit dazu bei, dass die Bearbeiter effizienter und flüssiger arbeiten können, da sie Neuerungen in Ebenen dadurch deutlich besser überblicken können und ihre eigenen Aufgaben besser an die geänderten Bedingungen anpassen können. Jeder Projektbeteiligte sieht immer die aktuellsten Ereignisse in der Planung und kann dadurch sehr viel effizienter und präziser Sachverhalte und weitere Schritte mit Kollegen diskutieren. Zusätzlich werden den Bearbeitern die Arbeitsschritte, welche benötigt werden um Planstände und Zwischenschritte zu erzeugen, sowie die Verteilung der aktuellsten Planunterlagen und deren Einpflegung in die aktuelle Projektarbeit, durch das Plug-In abgenommen. Dadurch wird den einzelnen Mitarbeitern mehr effektive Projektarbeitszeit zur Verfügung gestellt.

Durch die Verwendung von Versionskontrollsystemen stehen überall und zu jedem Zeitpunkt alle Daten in ihrer aktuellsten Form bereit und können von den berechtigten Bearbeitern weiterentwickelt werden, sowie von allen anderen Beteiligten eingesehen werden. Die Verwendung von unterschiedlichen Ebenen trägt zusätzlich zu einer besseren gleichzeitigen Bearbeitung bei und erweist sich speziell bei der Aufteilung von

Aufgaben als besonders Vorteilhaft, da dadurch die Möglichkeit entsteht mehrere Aufgaben gleichzeitig durch unterschiedliche Bearbeiter abarbeiten zu können. Die gesamte Projektarbeit lässt sich dadurch optimieren und effizienter gestalten, sodass ein Projekt flüssiger abgearbeitet werden kann. Wenn Mitarbeitern die Möglichkeit gegeben wird flüssiger arbeiten zu können, sowie administrative Aufgaben automatisiert ablaufen zu lassen, sollte sich dadurch das Arbeitsklima verbessern und die Gesprächskultur innerhalb des Projektteams positiv beeinflussen lassen.

Die Hypothese kann in ihren Hauptpunkten einer Überprüfung standhalten, in einigen Bereichen muss sie noch etwas geschärft werden, durch die kleine Testgruppe ließen sich einige Bereiche nicht nachhaltig überprüfen. So haben sich im Test die Kommunikation und die Arbeitsstrukturen so etabliert wie prognostiziert. Jedoch kann das Ergebnis nicht einfach auf größere Gruppen hochgerechnet werden. Es muss davon ausgegangen werden, dass vor der Verwendung eines solchen Systems, klare und einfache Strukturen geschaffen werden müssen. Mitarbeiter müssen in Kommunikation und Vorgehensweisen, welche von dem Plug-In gefördert werden, geschult werden, um unnötige Einstiegshürden schon vorab abzubauen. Wie sich während des Tests zeigte kann durch die Verwendung des Plug-Ins die Art und Weise einer Projektbearbeitung positiv beeinflusst werden. Die durch das Plug-In geförderten Arbeitsweisen haben das Potential die Arbeit im Bereich der Architekturplanung nachhaltig zu verändern und den Weg für eine Revolution der Arbeitsweise, wie nach dem Umstieg von Zeichenbrett zur Arbeit mit Computern, zu bereiten. Diese wird jedoch keine Technologische, sondern vielmehr eine Ideologische sein, von einer Einzelkämpfermentalität hin zur integrativen Zusammenarbeit aller Beteiligten als gleichberechtigte Partner.

Da das Plug-In innerhalb von einem dem Benutzer bekannten Programm aus operiert, müssen Anwender nicht auf ein neues Programm eingeschult werden, und können rein Anwendungstechnisch problemlos die Arbeit aufnehmen. Die Umstellung der Arbeitsweise auf eine Arbeitsweise mit mehreren Ebenen und die daraus resultierenden Möglichkeiten der gleichzeitigen Bearbeitung durch mehrere Nutzer, bedürfen jedoch einer umfassenden Einschulung und Vorbereitung der zukünftigen Nutzer. Das Plug-In nimmt dem Bearbeiter zwar die Verwaltung der einzelnen Zeichnungen und Ebenen ab, stellt aber erhöhte Anforderungen an die Kooperationsbereitschaft und die Konfliktlösungs- beziehungsweise Kommunikationsfähigkeiten eines jeden Projektbeteiligten.

8 Visionen für die Zukunft

Um die vorgeschlagenen Arbeitsweise mit einer automatisierten Versionierung vollständig in die Architekturarbeit zu implementieren, sollten noch zusätzliche Funktionen aus den Möglichkeiten eines Versionskontrollsystems etabliert werden. Die wichtigsten dieser Funktionen sind:

- optimale Darstellung der Historie sowie das Aufrufen von vergangenen Ereignissen
- parallele Arbeit an Alternativen und deren Zusammenführung zu einem Einzelergebnis
- Verknüpfung von unterschiedlichen Projekten, um Einzellösungen für andere Projekte zur Verfügung stellen zu können
- ein System welches die Freigabe einzelner Versionen durch die einzelnen Fachplaner erlaubt
- eine Möglichkeit Ergebnisse während der Bearbeitung zu teilen
- sowie die Integration aller Projektinformationen in die Versionskontrolle.

Allen diesen Funktionen ist gemeinsam, dass sie funktionell schon ein Bestandteil von Versionskontrollsystemen sind und nur die Etablierung eines Arbeitsablaufes und die Implementierung in das Plug-In nötig sind, um diese nutzen zu können. Dabei muss noch angemerkt werden, dass all diese Funktionen, sozusagen manuell auch ohne die Hilfe eines Plug-Ins benutzt werden könnten. Ein Plug-In bietet lediglich die nötige Vereinfachung der nötigen Abläufe, um neue Funktionen, sowie die schon im Plug-In umgesetzten, effizient, ohne großen Aufwand und ohne tiefgreifende Kenntnisse der Versionskontrolle, durch jeden Beteiligten durchzuführen.

Diese Erweiterungen ermöglichen einem Projektteam den Zugang zu Bearbeitungsmöglichkeiten, welche die Arbeit an Projekten zusätzlich unterstützen und verbessern können. So ist es möglich in der Zeitlinie eines Projektes zurück und nach vor zu springen, dadurch wird es Bearbeitern erleichtert aufgegebene Elemente oder nicht weiterentwickelte Elemente wieder in die Arbeit zurück zu holen. Damit zusammenhängend ist auch die Möglichkeit zu sehen, parallel unterschiedliche Zugänge zu erarbeiten und diese wieder miteinander zu fusionieren oder Entwicklungsschritte in andere Projekte zu integrieren.

Im folgenden sollen diese Erweiterungsmöglichkeiten kurz beschrieben werden, wie auch ihre Auswirkungen auf die Bearbeitungsprozesse während eines Projektes diskutiert werden.

8.1 Darstellung der Historie eines Projektes

Die Geschichte eines Projektes lückenlos zu speichern ist eine Sache, ebenso wichtig ist aber auch eine Möglichkeit bestimmte Ereignisse in dessen Geschichte darstellen zu können. Dies ist nicht nur wichtig um Ereignisse und Entscheidungen zu verifizieren, sondern kann auch dazu benutzt werden, Entwicklungen zu einem bestimmten Zeitpunkt neuerlich in die aktuelle Planung zu übernehmen. Eine leicht verständliche und nachvollziehbare Darstellung der Änderungsgeschichte ist eine Grundvoraussetzung, um die Möglichkeiten einer parallelen Entwicklung von Projektinhalten und Interaktion zwischen verschiedenen Projekten zu fördern.

Ein Versionskontrollsystem bietet hierzu die Möglichkeit alle Informationen, respektive die Information welche Person hat etwas geändert, was hat die Person geändert, wann wurden diese Änderungen an das Archiv übergeben, welche Änderungsanmerkungen wurden geschrieben und welche Dateien waren von dieser Änderung betroffen. Diese Informationen lassen sich für das gesamte Projekt, oder auch nur für Teilbereiche des Projekts und in unterschiedlicher Darstellungstiefe, sowie auch zeitlich begrenzt oder unbegrenzt ausgeben. Diese werden in einem zeilenbasiertem Textformat vom Versionskontrollsystem zur Verfügung gestellt und können dadurch relativ einfach in einer Datenbank abgebildet werden. Dadurch wird die Sichtung der gesamten Informationen über die Aktionen innerhalb eines Projektes unter Versionskontrolle deutlich erleichtert, da nun gezielt nach Schlagworten gesucht werden kann. Durch die Verwendung einer Datenbank zur Darstellung der Historie ist es möglich nicht relevante Ereignisse von vornherein auszublenden. Bestimmte Informationen, wie zum Beispiel das Erstellen einer neuen Ebene, sind für die Geschichte nicht unbedingt relevant, da es sich dabei um sogenannte Leereinträge handelt. Wie im Beispiel einer neu hinzugefügten Ebene, welche zu diesem Zeitpunkt noch keine Informationen enthält außer der Information, dass sie gerade erstellt wurde.

Durch die Verwendung einer Datenbank ist es somit möglich die nötigen Informationen über die Geschichte eines Projektes so darzustellen, dass eine Erfassung durch einen Benutzer mit Hilfe einer graphischen Oberfläche möglich wird. Dieser kann hierbei nicht nur Sortierung und Anzeigtiefe festlegen, sondern hat auch die Möglichkeit

historische Dateien zur Ansicht zu öffnen, diese in die aktuelle Planung oder in Alternativen zu integrieren oder auf Basis einer historischen Version einen Nebenzweig eröffnen und eine neue Richtung in der Entwicklung beschreiten.

8.2 parallele Entwicklung

Unter dem Begriff parallele Entwicklung wird die gleichzeitige Entwicklung von Teilaspekten oder des gesamten Projekts verstanden. Die Verwendung dieser Gleichzeitigkeit ist besonders dann von Interesse und von Vorteil, wenn noch nicht abschließend geklärt ist wie ein Teilbereich ausgestaltet werden soll oder wie ein spezielles Problem gelöst werden soll. Durch die Anfertigung von Alternativen kann den Entscheidungsträgern die Arbeit insofern erleichtert werden, dass diese aus verschiedenen Vorschlägen und Lösungen wählen können. Heutzutage wird diese Vorgehensweise nur im kleinen Maßstab angewendet, da eine Entwicklung von Alternativen sehr viel zeitlichen und administrativen Aufwand bedeutet. Durch die Verwendung eines Versionskontrollsystems lässt sich zumindest der administrative Aufwand minimieren, sowie die Erarbeitung einzelner Alternativebenen sehr gut in den Arbeitsablauf während eines Projektes einbauen.

Wenn heute eine Alternative geplant wird, muss für diese eigens eine eigene Unterstruktur aufgebaut werden und die Bearbeiter der Alternative müssen die nötigen aktuellen Zeichnungen immer wieder mit der Alternative aktuell halten, da sonst eine zu große Abweichung vom Hauptprojekt entstehen könnte und eine Alternative nur mehr schwer integriert werden kann. Bei der Verwendung eines Versionskontrollsystems kann hierfür ein neuer Zweig erstellt werden der entweder nur die Ebene enthält, welche zu einer Alternative ausgearbeitet werden soll, oder aber mehrere relevante Ebenen beziehungsweise das gesamte Projekt enthalten kann. So ist es möglich spezifisch angepasste Alternativen zu erstellen und diese in der für die Alternative nötigen Kommunikation mit dem Hauptprojekt zu halten. Einmal eingestellte Entwicklungen können einfach fortgesetzt werden, da die aktuell zuhaltenden Ebenen und Projektteile immer im aktuellen Stand angezeigt werden können.

Die Verzweigungen, welche durch einen Benutzer sehr einfach erzeugt werden können, funktionieren in sich wie das gesamte Versionsarchiv. So lassen sich Veränderungen speziell für einen bestimmten Zweig anzeigen und haben keinerlei Auswirkungen auf den Hauptast der Entwicklung. Dabei kann ein Zweig nur eine Ebene oder auch das gesamte Projekt beinhalten. Durch eine Zuweisung einer Ebene zu einem Nebenzweig

wird diese kopiert, sodass fortan zwei Bearbeiter unabhängig von einander an einem Lösungsansatz arbeiten können.

Die Herausforderung für das Plug-In stellt dabei die Zuordnung der Varianten zu deren Hauptdatei, sowie die Kombination der Varianten mit den restlichen Ebenen für eine Publikation dar. Um eine Variante zu erstellen, erhält ein Bearbeiter die Möglichkeit bei der Auswahl der zu Bearbeitenden Datei eine Variante dieser zu erstellen oder eine schon vorhandene Variante zu bearbeiten. Um einem Überblick über Varianten zu bewahren, werden bei der Erstellung einer Variante, der Name der Ursprungsdatei und die Variante in einer Datenbank gespeichert, hierbei wird wie bei den Auswahlsets beziehungsweise den Layouts vorgegangen. Diese sind bereits in einer projektbezogenen Datenbank hinterlegt und stellen die nötigen Informationen für die Verarbeitung bereit. Im Falle einer neuen Alternative wird für die Ursprungsdatei die dazugehörige Alternative und der in dieser Alternative enthaltenen Ebenen hinterlegt. Dies ermöglicht dem Plug-In für jede Ebene die vorhandenen Alternativen mittels einer Dropdown-Auswahl anzuzeigen wodurch der Benutzer eine Alternative auswählen kann. Wird eine Alternative gewählt welche mehr als nur eine Ebene umfasst, wird standardmäßig bei den zu betrachtenden Ebenen die Äquivalente der Originalebene in dieser Alternative angezeigt. Dadurch ist es dem Bearbeiter möglich die ausgewählte alternative Ebene zu bearbeiten und die entsprechenden Unterebenen aktuell anzuzeigen.

Wird während der Bearbeitung einer alternativen Ebene eine neue Ebene erstellt, wird diese standardmäßig der Alternative zugeordnet, sie kann jedoch auch einer anderen Alternative oder der Hauptentwicklungslinie zugeordnet werden. Ähnliches Verfahren wird beim Hinzufügen von Daten zu einer bestehenden Ebene angewandt. Somit wird einem Bearbeiter die Möglichkeit gegeben aus einer Alternative heraus, neue Daten im jeweiligen Entwicklungsstrang zu halten oder in einen Anderen zu integrieren.

Wenn Alternativen gedruckt werden sollen, werden die bestehenden Auswahlsets verwendet, dem Bearbeiter wird dabei, zusätzlich zu den Möglichkeiten ein Auswahlset und ein Layout zu wählen, die Möglichkeit gegeben einen Entwicklungsstrang auszuwählen. Für jede Entwicklungslinie wird das gewählte Layout, bei der ersten Verwendung, kopiert und fortan für diese Entwicklungslinie verwendet. Dadurch bleiben die Einstellungen des Layouts erhalten und es ist den Bearbeitern möglich nun den neuen Entwicklungsstrang im selben Layout wie bisher zu drucken, dies erleichtert

einerseits die Übersichtlichkeit bei der Betrachtung von mehreren Varianten, und es muss nicht ein gänzlich neues Layout erstellt werden.

8.3 erweitern und zusammenführen paralleler Vorgänge

Bei der Bearbeitung unterschiedlicher Varianten für ein Projekt, kann der Fall eintreten, dass zusätzliche Ebenen in die Variante, aus dem Hauptstrang des Projektes oder aus einer anderen Variante, integriert werden müssen. Dieser Fall tritt dann ein, wenn eine Variante sich mehr und mehr von der Entwicklung im Hauptprojekt entfernt oder in einer anderen Variante Elemente und Ebenen entwickelt wurden, welche in der gerade bearbeiteten Variante von Interesse oder Nutzen sind. Für diesen Vorgang wird den Bearbeitern die Option zur Verfügung gestellt einzelne Elemente in eine Variante zu exportieren oder eine gesamte Ebene zu kopieren. Dieser Vorgang erfordert, dass die zu exportierenden Daten und wenn Elemente in eine bereits existierende Ebene kopiert werden sollen, diese nicht durch einen Bearbeiter gesperrt und somit gerade bearbeitet werden. Ist diese Voraussetzung gegeben werden Ziel und Ursprung der zu exportierten Daten zur Bearbeitung durch den betreffenden Benutzer gesperrt und die Übertragungsfunktionen für diesen freigegeben. Bei der Übertragung von Elementen in eine andere Variante werden einem Bearbeiter nur diejenigen Elemente einer zu exportierenden Ebene angezeigt welche in der Ebene, welche die Elemente aufnehmen soll, nicht enthalten sind. Zur visuellen Hilfe wird dem Bearbeiter dabei die Empfängerebene als eingefärbte Hintergrundebene angezeigt.

Wird eine gesamte Ebene übertragen, kann der Bearbeiter nach der erfolgreichen Übertragung der Ebene eine neue Bearbeitung starten. Werden jedoch Elemente aus einer Ebene übertragen, muss der Benutzer vorerst die Ebene, welche diese Elemente aufgenommen hat, prüfen, ob die neuen Elemente nicht im Konflikt mit schon existierenden stehen. Danach kann auch hier eine neue Bearbeitung gestartet werden.

Es kann aber auch der Fall eintreten, dass zwei Varianten zu einer gemeinsamen Variante vereint werden sollen oder eine Variante mit dem Hauptprojekt zurückgeführt werden soll. Dabei wird grundsätzlich die selbe Vorgangsweise wie bei der Übertragung von Elementen zwischen zwei Varianten angewendet, jedoch mit einem Unterschied. So werden als erstes, für jede Ebene der zu integrierenden Variante, die Elemente durch einen Bearbeiter ausgewählt und in einer Sicherungskopie aufbewahrt. Sobald dieser Vorgang abgeschlossen ist, wird der Bearbeiter aufgefordert die Elemente jeder Ebene der Zielvariante die Elemente auszuwählen welche durch die Elemente aus der

Sicherungskopie ersetzt werden sollen. Dafür wird ihm die Sicherungskopie, der zu importierenden Elemente im Hintergrund eingefärbt angezeigt. Erst danach werden die Elemente aus der Sicherungskopie eingefügt. Wenn dieser Vorgang beendet wurde, wird der Bearbeiter aufgerufen mögliche Konflikte selbst zu lösen oder die Ebenen an andere Bearbeiter zu übergeben um die Konflikte zu lösen. Dabei wird solange ein Konfliktparameter für jede Ebene in der Dokumentation der entsprechenden Auswahlsets gehalten bis eine jede Ebene durch den Befehl Konflikte lösen und Konflikte gelöst angesprochen wurden. Solange Konfliktparameter in einem Auswahlset enthalten sind kann dieses weder gedruckt werden, noch ist eine normale Bearbeitung der Variante möglich. Dies wird deshalb ausgeschlossen, damit das Projektteam gezwungen ist Konflikte aus einer Zusammenführung zuerst zu lösen, bevor die normale Bearbeitung weitergeführt werden kann. Bei der Bearbeitung der Konflikte kann ein Bearbeiter lediglich die Ebene zur Konfliktlösung bearbeiten und alle anderen Ebenen werden, anders als bei der normalen Bearbeitung, zur Betrachtung geöffnet.

Sobald Varianten zusammengeführt wurden wird eine zusätzliche Version im Versionsarchiv gespeichert welche in der Datenbank hinterlegt, dass die Bearbeitung einer zu integrierende Variante durch einen Benutzer gestoppt wurde und die Variante wird zur Bearbeitung durch die Benutzer gesperrt. Ab diesem Zeitpunkt ist die Variante nur noch zur Ansicht offenbar und kann nicht mehr weiterentwickelt werden, da sie in eine andere Variante oder das Hauptprojekt integriert wurde. Zu diesem Zweck wird jedes Element welches in der Variante hinterlegt wurde, durch den Administrator zur Bearbeitung gesperrt. Soll die gestoppte Variante zu einem späteren Zeitpunkt wieder gestartet werden, kann die erfolgte Bearbeitungssperre durch einen Administrator oder den Projektleiter wieder aufgehoben werden und die Variante kann wieder wie vorher weiterentwickelt werden. Durch die Bearbeitungssperre scheint die Variante nicht mehr in der Auswahl der möglichen Varianten einer Ebene zur Bearbeitung auf und ermöglicht so eine bessere Übersichtlichkeit über die bearbeitbaren Versionen und das Hauptprojekt.

8.4 unterschiedliche Projekte miteinander verbinden

Innerhalb eines Projektes ist es, wie vorhin beschrieben, von Interesse verschiedene Entwicklungslinien miteinander zu verbinden. Ebenso kann es von Interesse sein Inhalte von verschiedenen Projekten miteinander zu verbinden, beziehungsweise Ebenen oder Elemente in ein anderes Projekt zu integrieren. Besonders nach Abschluss eines

Projektes, kann dieses nun zu einem Pool an Informationen, Lösungsansätzen und Inspirationen für ein neues Projekt werden. Aber auch während einer Projektlaufzeit können Situationen entstehen, in welchen in parallelen Projekten ähnliche Problemstellungen behandelt werden oder Lösungen für das selbe Problem schon gefunden wurden. Um die Bearbeitung aller Projekte zu beschleunigen ist es von Vorteil wenn die schon erstellten Elemente an andere Projekte verteilt werden können.

Grundsätzlich wäre es möglich, wie bei der Verwendung von Alternativen, verschiedene Ebenen aus verschiedenen Projekten in eine Bearbeitung zu laden, jedoch ist eine solche Vorgangsweise aus Gründen des Projektverlaufes nicht zu empfehlen. Bei einer Verbindung von Ebenen außerhalb von Projektstrukturen wird die Projekthistorie gestört, da Änderungen an einer entsprechenden Ebene in einem Projekt gespeichert werden und die restlichen Ebenen im bearbeiteten Projekt unter Versionskontrolle stehen. Dadurch wird es für die Bearbeiter unmöglich die Änderungen an einer Ebene welche aus einem anderen Projekt verbunden wurde verfolgen und kontrollieren zu können. Deshalb müssen alle Elemente aus einem Projekt extrahiert werden um in ein anderes Projekt integriert zu werden.

Für die optimale Verwendung dieser Möglichkeiten wird den Benutzern ein internes Projekt zur Verfügung gestellt, welches alle Ebenen und Elemente aus beendeten Projekten enthält, welche nach Ansicht der Beteiligten für einen späteren Zeitpunkt von Wichtigkeit sein könnten. Dadurch kann über den Verlauf von mehreren Projekten ein Archiv von Beispiel- und Detaillösungen entstehen. Dies erlaubt dem einzelnen Benutzer auf verifizierte Elemente zurückzugreifen und nicht alle Projekte nach Lösungen durchforsten zu müssen, um die gewünschten Informationen zu finden.

Um dieses Archiv zu füllen, können Mitarbeiter einzelne Elemente oder auch ganze Ebenen aus den Projekten extrahieren und in das interne Projekt übertragen. Bei dieser Übertragung müssen die Mitarbeiter einen Namen des zu erstellenden Eintrags angeben sowie eine kurze Beschreibung des Inhalts bereitstellen. Werden Elemente oder ganze Ebenen extrahiert, wird Aufgrund der Auswahl eine graphische Darstellung erstellt, welche dem Namen und der Beschreibung zugeordnet werden.

Für den Bezug einer Lösung aus diesem Archiv, kann ein Mitarbeiter während der Bearbeitungsphase auf diese Datenbank zugreifen und eine Lösung auswählen. Diese Lösung wird dann in seine momentane Bearbeitung integriert und kann ab diesem Zeitpunkt vom Mitarbeiter angepasst werden. Da ein solches Lösungsarchiv losgelöst

von anderen Projekten funktioniert, können enthaltene Lösungen weiterbearbeitet und erweitert werden, ohne dass ein anderes Projekt direkt davon beeinflusst wird. Diese Möglichkeit macht ein Lösungsarchiv, welches als Mittler zwischen den Projekten und Mitarbeitern steht, zu einem großen Gewinn für alle Beteiligten. Da es einerseits, bei richtiger Verwendung, verhindern kann, dass Arbeiten in unterschiedlichen Projekten doppelt gemacht werden. Andererseits stellt es eine gute Basis für den Aufbau beziehungsweise den Erhalt und Ausbau eines Wissensmanagements dar.

8.5 Freigabeprozesse

Da durch eine Versionskontrolle der Name, die Uhrzeit einer Handlung und ein Kommentar automatisch gespeichert werden, kann dieser Umstand auch zur Freigabe von Auswahlsets durch Fachplaner genutzt werden. So kann den Teilprojektleitern eines Projektes durch das Plug-In die Fähigkeit gegeben werden Auswahlsets der anderen Fachplaner freizugeben. Diese Freigabe kann durch schriftliche Anmerkungen und einer digitaler Unterschrift erfolgen. Die Anmerkungen werden dabei auf einer neuen Ebene für die jeweiligen Planer ersichtlich und können eingearbeitet werden. Die digitale Signatur entspricht hierbei dem Benutzernamen und seiner Autorisierung eine Freigabe durchzuführen. Dabei wird jedem Layout ein Attribut hinzugefügt, welches die Freigabe mit oder ohne Anmerkungen oder die Nichtfreigabe mit Anmerkungen hinterlegt. Durch dieses Attribut kann das Plug-In bei jedem Druckvorgang die jeweiligen Freigaben mit Datum und Person auf den Plänen automatisch vermerken. Dadurch ist eine Freigabe immer mit einer Version und einem Layout des Projektes verbunden und jeder Mitarbeiter kann sehen, für welche Version eines Layouts die Freigabe gilt.

Eine Freigabe erfüllt zweierlei Aufgaben, einerseits bestätigt sie die korrekten Angaben des betreffenden Gewerks in einer Planung eines anderen Gewerks, oder sie bestätigt, innerhalb eines Fachbereiches, den positiven Fortschritt einer Planung, durch den Projektleiter. Andererseits ist es die Genehmigung eines Planungsstandes durch den Auftraggeber. Beide Möglichkeiten erfordern mitunter jedoch Änderungen in der momentanen Planung. Durch die Möglichkeit diese erforderlichen Änderungen in der Versionskontrolle zu hinterlegen, wird es dem Projektteam erleichtert, diese korrekt einzuarbeiten. Bei der internen Freigabe wird das freizugebende Layout für den Prüfer zur Ansicht angezeigt. Dieser erhält eine Ebene in welcher er seine Anmerkungen hinterlegen kann. Sollte er das Layout ohne Anmerkungen freigeben wird lediglich ein Attribut für das Layout erstellt, welches das Layout, die Version, den Namen der Prüfers

sowie das Datum beinhaltet. Werden Anmerkungen hinterlegt wird zusätzlich zu dem Attribut des Layouts die Anmerkungen als Ebene zum Projekt hinzugefügt. Dabei steht jedem Prüfer eine Ebene zur Verfügung, welche bei einer neuerlichen Prüfung wieder angezeigt wird und damit die letzten Anmerkungen anzeigt. Sind bei einer neuerlichen Prüfung keine Anmerkungen mehr nötig wird der Inhalt der Ebene gelöscht, die Ebene selbst bleibt jedoch für die nächste Prüfung erhalten. Sobald eine Prüfung erfolgte, wird die aktuellste Freigabe eines Prüfers bei einem erneuten Druck des Layouts auf dem Plan schriftlich vermerkt.

Da eine Freigabe durch den Auftraggeber meist immer analog erfolgt, muss diese durch einen berechtigten Mitarbeiter in gleicher Weise wie eine interne Freigabe eingegeben werden. Jedoch wird der Namens des Prüfers und des Auftraggebers, welcher die Freigabe mittels Unterschrift erteilt, angegeben. Zusätzlich muss der handschriftlich Freigegebene Plan als Datei unter Versionskontrolle gestellt werden und ist damit mit der Freigabe durch den Auftraggeber im Versionsarchiv direkt in einer Version verbunden. Werden bei der Freigabe durch einen Teilprojektleiter oder den Projektleiter Anmerkungen erstellt, werden die Bearbeiter während des Öffnen eines Projektes darauf aufmerksam gemacht, dass ein Layout durch einen Teilprojektleiter begutachtet wurde und Anmerkungen dazu erstellt wurden. Wenn ein Mitarbeiter auf die Anmerkungen reagiert wird ihm zuerst das Dokument mit den Anmerkungen angezeigt, sowie die zugehörigen Ebenen aufgelistet. Danach kann der Mitarbeiter, durch Auswahl der jeweiligen Ebenen, mit der Einarbeitung der Anmerkungen beginnen. Ist die Einarbeitung aller Anmerkungen abgeschlossen, wird dem Teilprojektleiter, welcher die Anmerkungen hinterlassen hat, eine Nachricht zugesandt und eine neuerliche Freigabe erbeten.

8.6 erweiterte Synchronisation während der Bearbeitung

Die Synchronisation während der Bearbeitung mit den neuesten Versionen der zu betrachtenden Ebenen ist ein Grundbestandteil des Plug-Ins. Diese ist vor allem wichtig, da durch die Verwendung von aktuellen Daten auch die Arbeit passend, zeitnah und qualitativ erledigt werden kann. Das Wissen über die aktuellsten Entwicklungen in einem Projekt sind sozusagen immanent für die erfolgreiche und schnelle Bearbeitung von Aufgaben. Die aktuellen Ebenen, welche einen abgeschlossenen Bearbeitungsschritt darstellen, können dadurch bei jedem Nutzer, in ihrem aktuellen Zustand, angezeigt werden.

Wenn ein Bearbeiter seine Arbeit an einer Ebene abgeschlossen hat und eine neue Aufgabe übernimmt, schließt er die Bearbeitung ab und übergibt somit eine neue Version der Ebene, welche er bearbeitete, an das Versionsarchiv. Nun kann jedoch der Fall eintreten, dass ein Teilaspekt der gesamten Aufgabe schon vor Beendigung der gesamten Bearbeitung für andere Nutzer von Interesse ist. Dazu braucht der Bearbeiter die Möglichkeit einen Zwischenstand seiner Arbeit in das Versionsarchiv zu laden. Der Befehl um dies durchführen zu können, muss zwei Dinge durchführen, einerseits muss er den Zwischenstand der Arbeit an die Versionskontrolle übergeben, andererseits muss er sicherstellen, dass die Bearbeitungssperre für die anderen Bearbeiter weiter aufrecht bleibt. Da diese bei einer Rückgabe an die Versionskontrolle erlischt, muss die Sperre bei Übergabe sofort wieder eingerichtet werden, um zu verhindern, dass ein anderer Benutzer diese versehentlich für sich beanspruchen kann. Der übergebene Zwischenstand wird dabei mit der Anmerkung Zwischenstand an das Versionsarchiv übergeben. Erst bei der wirklichen Beendigung der Arbeit an dieser Ebene wird vom Bearbeiter eine genaue Beschreibung seiner Änderung verlangt. Zwar stellt auch der Zwischenstand einer Zeichnung eine eigenständige Version dar, diese wird aber in der Darstellung der Historie nur angezeigt wenn sie tatsächlich den aktuellsten Stand im Versionsarchiv darstellt. Diese Vorgehensweise erleichtert dann auch die Überschaubarkeit der Darstellung, da dort dann nur abgeschlossene Bearbeitungen dargestellt werden können.

Durch die Funktion einzelne Elemente in eine neue Ebene zu exportieren und dadurch aus der Ursprungsebene zu entfernen, müssen auch die anzuzeigenden Ebenen aktualisiert werden können. Dabei wird bei einer Aktualisierung der angezeigten Ebenen überprüft, ob dem entsprechenden Auswahlset neue Ebenen hinzugefügt wurden. Sollte dies der Fall sein, wird der Nutzer darüber informiert und kann die neu hinzugefügten Ebenen hinzuladen, wenn er dies wünscht.

8.7 Verteilung und Management von Aufgaben

Um Aufgaben und offene Punkte während des Projekts bestmöglich zu verteilen und die Übersicht über deren Bearbeitungsstand zu behalten, ist es einerseits möglich diese direkt zu verteilen oder man nutzt die Fähigkeiten welche eine Versionskontrolle und das Plug-In bieten. Indem Aufgaben in einer Datenbank aufgelistet werden, ähnliches wurde schon bei der Verbindung zweier Varianten und der Abarbeitung von Anmerkungen zu einer Freigabe beschrieben, kann ein Mitarbeiter diese Annehmen und

deren Bearbeitung starten. Dabei wird automatisch die in der Aufgabe zu bearbeitende Ebene geöffnet und das für die Aufgabe entsprechend Auswahlset, für die Auswahl der Ebenen zur Betrachtung, angezeigt. Nach erfolgreicher Beendigung der Arbeiten wird die Aufgabe als erledigt markiert. Wird eine Aufgabe erstellt, welche mehrere Ebenen umfasst, wie zum Beispiel die Abarbeitung von Anmerkungen zu einer Freigabe, dann werden in der Datenbank dementsprechend viele Aufgaben wie Ebenen angegeben wurden erstellt. Für die einzelnen Mitarbeiter erleichtert sich dadurch die Abarbeitung der Aufgaben, da dadurch direkt im Plug-In eine To-Do Liste integriert wird und jeder Teilnehmer einen Überblick über die ausstehenden Arbeiten in einem Projekt hat.

Zusätzlich zu Aufgaben, welche durch Projektleiter, Prüfer und zusammenführen von Varianten erstellt werden, wird jede Reaktion auf Änderungen, welche durch die Benutzer selbst stattfinden, ebenfalls als neue Aufgabe deklariert. Somit wird nicht mehr der letzte Bearbeiter einer Ebene über Fehler, Unstimmigkeiten oder einen generellen Änderungsbedarf in einer bestimmten Ebene informiert, sondern das gesamte Team. Dadurch lässt sich die Effizienz des Teams noch zusätzlich steigern, da bei der Information an nur ein Teammitglied, die Abarbeitung der Aufgabe verzögert werden kann oder gar ganz vergessen wird, weil dieser erst eine bestehende Bearbeitung abschließen müsste, um die Korrekturen an einer anderen Ebene durchführen zu können. Wenn jedoch Aufgaben bei der Auswahl der nächsten Bearbeitung zur Verfügung stehen, kann der nächste freie Bearbeiter die Aufgabe auswählen und abarbeiten.

Da für die Verteilung der Aufgaben eine Datenbank verwendet wird, welche ebenfalls unter Versionskontrolle steht, ist es möglich auch die Übernahme einer Aufgabe zu protokollieren und daraus eine Übersicht für einen Projektleiter über die Auslastung seiner Teammitglieder sowie über den Fortschritt des Projektes im allgemeinen zu generieren. Dadurch wird es dem Projektleiter möglich, dass er Aufgaben effizienter planen und verteilen kann, sowie Aufgaben speziell an bestimmte Bearbeiter verteilen kann. Dies ist dadurch möglich, da die graphische Darstellung der Aufgaben vom Plug-In übernommen wird und dieses den Benutzer, welcher dieses benutzt identifizieren kann.

8.8 Versionierung für das gesamte Projekt

Das Potential der Versionskontrolle ist jedoch, durch eine reine Verwendung für die Zeichenarbeit und ihre Abläufe, nicht annähernd ausgeschöpft. Großes Potential hat

Versionskontrolle speziell auch in der Verwaltung von allen Daten welche einem Projekt direkt zuordenbar sind. Die Verwendung für die Verbesserung und Beschleunigung von Arbeitsabläufen während der Zeichenarbeiten eines Projektes wurden in den vorangegangenen Kapiteln ausführlich diskutiert. Doch ist diese Verwendungsmethode nicht die ursprüngliche Domäne der Versionskontrolle. Wie schon in Kapitel 4 „Versionskontrolle“ beschrieben wurden Versionskontrollsysteme grundsätzlich für die Verwaltung und Aufzeichnung von Änderungen an textbasierten Dateien entwickelt. In diesem Bereich haben die Systeme auch die stärkste Verbreitung. Auch in der täglichen Architekturarbeit gibt es viele Bereiche in welchen keine Zeichnungen sondern andere Daten produziert werden, zum Beispiel: Korrespondenz mit verschiedenen externen Beteiligten, die Erstellung und Bearbeitung von Verträgen, Abrechnungen, technische und architektonische Beschreibungen, Unterlagen für Einreichungen, Tür- und Fensterlisten, sowie Raumbücher oder Ausstattungsbeschreibungen und Produktentscheidungsblätter. All diese Daten können mit dem selben Versionsarchiv verarbeitet werden. Dadurch wird das Versionsarchiv erst zum vollständigen Gedächtnis des Projektes.

Es ergeben sich jedoch einige Schwierigkeiten wenn das gesamte Projekt in einem Versionsarchiv bearbeitet wird. So ist es nötig, dass die Daten, welche für die Planungsarbeiten relevant sind, in irgendeiner Form von den restlichen das Projekt betreffenden Daten separiert werden. Hierfür eignet sich am besten ein Präfix bei den Versionsanmerkungen welche bei der Rückgabe von Dateien an die Versionskontrolle mitgesendet wird. Dadurch können sämtliche Aktionen in einem Projektarchiv voneinander separiert und ihrer Funktion und Verwendung im Projekt zugeordnet werden. Besonders Wichtig ist diese Vorgangsweise, wenn eine Darstellung der Geschichte des Projekts angezeigt werden soll, oder ein Bearbeiter eine bestimmte Version einer Datei wieder aufrollen möchte.

Durch die Ausweitung der Versionskontrolle auf diese Daten, werden die Daten nicht nur in einem geschichtlichen Kontext abgelegt, sondern können auch mit Versionen der eigentlichen Planung in Relation gesetzt werden. Dadurch wird eine Rückverfolgung der Geschehnisse und die eigentliche Aufgabenabwicklung positiv beeinflusst, da Dokumente, welche direkte Auswirkungen und Nutzen für eine Version der Planung darstellen, direkt mit allen Informationen in einer Version verbunden werden können. Eine definierte Verbindung zwischen Dokumenten und Aufgaben beziehungsweise der Bearbeitungen einzelner Ebenen muss dazu etabliert werden. Ein einfaches Präfix wie

sie bei der Trennung der Versionen für die Bearbeitung in CAAD und sonstigen Dokumenten verwendet wird, kann diese Verbindung jedoch nicht herstellen. In diesem Fall muss erneut auf eine Datenbank zurückgegriffen werden, welche die einzelnen Dokumente eines Projektes auflistet und eine, von einem Bearbeiter einzutragende, Verkettung mit bestimmten Aufgaben und oder Versionen des Projektes beinhaltet. Auf diese Weise können sämtliche Daten, allen beteiligten Bearbeitern, für jede Aufgabe sortiert beziehungsweise aktuell dargeboten werden. Da dadurch jede mögliche Information für jeden Schritt des Projektes direkt mit diesem Verbunden ist und diese auch von Nutzern bei der Abarbeitung von Aufgaben angezeigt werden, wird die Informationsdichte erhöht. Jeder Bearbeiter kann somit auf alle für einen Arbeitsschritt relevanten und vorhandenen Informationen zugreifen und aufgrund dieser Informationen seine Aufgabe qualitativ hochwertiger planen und abschließen. Denn es ist einem Bearbeiter dadurch Möglich sich über alle Sachverhalte bezüglich seiner Aufgabe vorab zu informieren und erst bei Fragen, welche durch die mit dem Arbeitsschritt verbundenen Daten nicht abgedeckt werden, ist er gezwungen Rücksprache mit Kollegen oder Projektleitern zu halten.

Diese Kombination aller Projektdaten zeigt seine Stärken besonders dann, wenn ein Projektteam nicht auf einen Standort beschränkt ist. Bei Bearbeitern, welche durch größere Distanzen als einem Raum voneinander getrennt am selben Projekt arbeiten, kann es zu Problemen bei der ständigen Kommunikation kommen, da ein persönliches Gespräch aufgrund der Distanz nicht möglich ist. Kann jedoch ein jeder Mitarbeiter vorab alle Informationen zu einer Aufgabe einsehen und zusätzlich noch auf die Kommentare und Änderungsanmerkungen seiner Kollegen zugreifen, können nötige Aufklärungsgespräche mit Kollegen auf eine Kernfrage minimiert werden. Dabei wird einerseits der Angesprochene Kollege nicht komplett aus seiner derzeitigen Aufgabe gerissen, andererseits kann sich der Bearbeiter auf schriftliche Aussagen verlassen.

9 Fazit

Die in dieser Arbeit entworfene Arbeitsweise unter Verwendung einer Versionskontrolle zeigt, dass, entgegen der momentanen Situation, die Möglichkeiten der Bearbeitungen im Projektverlauf durch die Bearbeiter und dadurch das Ergebnis eines Projektes erweitert werden könnten. Nicht nur die automatisierte Verwaltung und die historische Sicherung des gesamten Projektes, sondern auch die Möglichkeiten des Rücksprungs zu einem früheren Zeitpunkt, ohne die danach entwickelten Elemente zu verlieren, um die Arbeiten in eine alternative Richtung zu lenken, stellen ein Novum in der Architekturarbeit dar. Speziell in der Phase des Entwurfes, wo viele Möglichkeiten, Alternativen und Lösungsvorschläge entwickelt werden, helfen die Funktion eines Versionskontrollsystems parallele Entwicklungen zu forcieren und erfolgreich durchzuführen.

Das mitunter stärkste Argument ein Versionskontrollsystem in die Architekturarbeit zu integrieren, liegt mit großer Sicherheit in der durch das Programm selbstständigen Verwaltung der enthaltenen Elemente, da hierbei automatisch immer die aktuellsten Informationen für alle Beteiligten zur Verfügung gestellt werden können. Dabei kann sich jeder Benutzer zu jedem Zeitpunkt sicher sein, dass die Informationen welche ihm vom System zur Verfügung gestellt werden aktuell und allen Beteiligten ebenfalls bekannt sind. Gepaart mit dem Abgleich der aktuellen Version mit der Version der Informationen zum Zeitpunkt der letzten Arbeitsphase, wodurch dem Nutzer die Neuerungen in einer Ebene angezeigt werden können, wird einem Nutzer ein Arbeitsgerät in die Hand gegeben, mit welchem er seine Aufgaben effizient gestalten und abarbeiten kann.

9.1 Eine neue Art zu entwerfen?

Ob die Verwendung von Versionskontrolle und der Einsatz des Plug-Ins die Entwurfsarbeit verändern kann, wird nur die Zeit zeigen können. Dennoch kann schon zu diesem Zeitpunkt gesagt werden, dass die Versionskontrolle das Potential besitzt die Architekturarbeit zu verändern. Denn schon alleine durch den Wegfall der manuellen Verwaltung von Zeichnungen, welche durch die Versionskontrolle übernommen wird, sowie die Zurverfügungstellung der immer aktuellsten Daten, bietet den Nutzern die Möglichkeit sich stattdessen auf die eigentliche Entwurfs- und Planungsarbeit zu

konzentrieren. Ein Projektteam gewinnt dadurch Zeit, welche im Projektverlauf immer rar und kostbar ist, zusätzlich spart sich das Team noch Nerven, da nicht mehr daran gedacht werden muss, den aktuellsten Stand der Planung zu versenden und oder ihn in die Planung zu integrieren.

Schon aus diesen zwei Gründen ist die Versionskontrolle ein großer Vorteil. Nicht nur für einen Mitarbeiter oder ein Team bietet diese Zeit- und Nervensparnis einen großen Vorteil, wirklich tragend wird dieser Vorteil wenn mehrere Teams aus unterschiedlichen Gewerken zusammenarbeiten. Dabei kumuliert sich nicht nur die Zeitersparnis der einzelnen Bearbeiter, sondern die Vorlaufzeiten bis ein Team die aktuellen Daten eines anderen Teams erhält, werden drastisch reduziert, da jede Änderung für alle Beteiligten sofort ersichtlich ist. Dies sind jedoch nicht die einzigen Fähigkeiten eines Versionskontrollsystems, sondern es werden auch alle Arbeitsschritte und jede noch so kleine Änderung während der Projektlaufzeit und sogar darüber hinaus im Versionsarchiv mitgespeichert und gegebenenfalls sogar mit benutzerspezifischen Anmerkungen hinterlegt. Dadurch kann jeder Arbeitsschritt nachverfolgt werden und zu jeden beliebigen Versionspunkt in der Projektgeschichte gesprungen werden, um von dort aus eine neue Bearbeitung zu starten, oder Elemente daraus in die neueste Version zu integrieren. Dabei muss aber nicht zwingend in der Geschichte des gesamten Projekts nach hinten gesprungen werden, sondern dies kann für unterschiedlichste Inhalte und zu unterschiedlichsten Versionen passieren. Dabei bleiben alle anderen Inhalte in der aktuellen Fassung erhalten, lediglich die Historischen werden durch eine Bearbeitung wieder aktuell. Die so wiederhergestellten Daten ersetzen jedoch nicht ihr zukünftigen Ich, diese bleiben in ihrer letzten Version erhalten und können ebenfalls wiederhergestellt werden. Ebenso bieten Versionskontrollsysteme die Möglichkeit verschiedene Alternativen zu erschaffen und diese gleichzeitig zu entwickeln, diese untereinander zu verbinden oder bei Bedarf wieder in die Hauptentwicklungslinie zurückzuführen.

Durch diese Funktionen ermöglicht die Versionskontrolle neue Herangehensweisen bei der Lösungsfindung und hilft etablierte Arbeitsweisen zu verbessern. Durch die Zeitersparnis bei der Verwaltung von Versionen, kann ein Projektteam bei gleichbleibendem Aufwand wesentlich mehr Zeit in die eigentliche Architekturarbeit investieren und ist dadurch fähig architektonische Ideen in einer höheren Qualität auszuarbeiten. Sozusagen wird einem Projekt mehr Zeit gegeben um sich optimal entwickeln zu können und dadurch zusätzlich an Qualität zu gewinnen.

9.2 Einsetzbarkeit in der Realwirtschaft

Verschiedene CAAD Hersteller experimentieren bereits, wie im Kapitel 3 „Engineering Document Management Systems“ erwähnt, mit der Implementierung von Versionskontrolltools und versuchen diese Systeme in der Architektur salonfähig zu machen. Trotz der Probleme und Schwächen welche diese Produkte immer noch besitzen, befindet sich das in dieser Arbeit vorgestellte Plug-In durchaus in guter Gesellschaft. Es verwendet nicht nur Ideen, welche schon in EDMS Einzug gehalten haben, sondern es erweitert diese noch und versucht das gesamte Potential von Versionskontrollsystemen zu nutzen. Schon EDMS bringen, durch die Zurverfügungstellung einer automatischen Verwaltung von Dokumenten, den Nutzern wertvolle Zeitvorteile, können aber nur eine begrenzte Geschichte darstellen. Ähnlich einem normalen Dateisystem mit Sicherungskopien wobei die letzten Änderungen gespeichert werden, jedoch verlorengehen wenn diese Wiederhergestellt werden. Da das Plug-In nicht nur die Verwaltung übernimmt, sondern auch die gesamte Geschichte für alle Nutzer bereitstellen kann, sowie die Möglichkeit von gleichzeitiger Entwicklung von Alternativen bietet, kann es ein wesentlich stärkeres Arbeitsgerät werden als die vorhandenen Systeme bereits sind.

Aktuelle Systeme welche die Möglichkeit bieten gleichzeitig an Projekten zu arbeiten oder mit Versionskontrolle ausgestattet sind, können meist nur einen Teilbereich der Aufgaben abdecken und brauchen ein weiteres Programm, welches die restlichen Bereiche abdeckt. So gibt es zwar CAAD Software welche es den Benutzern erlaubt Zeichnungen durch mehrere Nutzer gleichzeitig zu bearbeiten, oder Systeme die den Benutzern die Versionierung und Verwaltung von Zeichnungen abnehmen können. Jedoch kann kein Programm alle Funktionen, welche gleichzeitig benötigt würden, um dem Potential eines Versionskontrollsystems, welches durch ein Plug-In mit der CAAD Software kommuniziert, effektiv etwas entgegenzustellen. Die Möglichkeiten die ein, wie in dieser Arbeit beschriebenes, Plug-In bieten kann, erweitern nicht nur die Möglichkeiten wie an die Erledigung von Aufgaben herangegangen werden kann, sondern bieten einen auch neue Möglichkeiten der Zusammenarbeit unter Kollegen.

Dabei müssen sich Benutzer lediglich auf die neuen Funktionen des Plug-Ins einstellen und sind nicht gezwungen, wie bei anderen Lösungen ein neues Programm erlernen zu müssen. Die Bedienung des eigenen Werkzeuges wird dabei nicht verändert, lediglich der Ablauf wie mit der Arbeit begonnen wird und wie diese Beendet wird wird durch

das Plug-In verändert. Schon im Test, Kapitel 7 „Bearbeitung während eines Beispielprojektes“, wurde deutlich dass die Umstellung auf die Funktionen des Plug-Ins für die Nutzer sehr einfach ist. Die einzige Änderung welche ein Nutzer dabei erfährt ist, dass er immer aktuell alle Informationen, welche durch das Versionskontrollsystem geteilt werden, erhält und auch seine aktuellen Fortschritte aktuell mit allen Kollegen teilt. Das Plug-In ist so aufgebaut, dass ein Nutzer dieses, zumindest in den Hauptfunktionen, intuitiv nutzen kann. Für die Verwendung der Geschichtssprünge und der Erzeugung und Bearbeitung von Alternativen wird zumindest ein Grundverständnis von deren Anwendung verlangt. Dabei muss ein Benutzer nicht die einzelnen Schritte welche im Hintergrund ausgeführt werden verstehen. Es reicht wenn dem Benutzer klar ist, was passiert wenn er in der Geschichte zurückspringt oder welche Funktion Alternativen übernehmen können. Dabei reicht das Repertoire von einer simplen alternativen Planung für eine einzelne Ebene, bis hin zum Rückschritt auf eine ältere Version des gesamten Projektes oder einer Sammlung von Ebenen, um von dort aus ein Projekt in eine neue Richtung zu lenken.

9.3 Vorteile bei der Arbeit

Durch eine Einführung des Plug-Ins in die Architekturarbeit wird den einzelnen Akteuren ein Arbeitsgerät in die Hand gegeben, mit welchem sie eine gänzlich abgewandelte Arbeitsweise verfolgen können. Zwar schließt die Verwendung des Plug-Ins, die etablierten Arbeitsweisen nicht aus, durch die Verwendung werden sich jedoch auch diese mit Sicherheit verändern, da schon die direkte Zusammenarbeit von verschiedenen Gewerken, mittels des unter Versionskontrolle stehenden Projektes, erleichtert wird. So können alle Beteiligten weiterhin mit ihrer klassischen Arbeitsweise weiterarbeiten, doch durch die Verbindung der einzelnen Benutzer und die Tatsache, dass immer alle die selben Grundlagen besitzen, wachsen die unterschiedlichen Planungsbereiche näher aneinander und erreichen dadurch eine bessere Integration aller Beteiligten. Die klassische Arbeitsweise, welche auf gesamten Zeichnungen aufbaut, wird sich dabei sehr schnell auf eine Arbeitsweise umstellen, bei welcher einzelne Ebenen zusammen ein Ganzes ergeben. Der Vorteil, dass Elemente je nach Bedarf in der Arbeitsoberfläche ein und ausgeblendet werden können und dass verschiedene Ebenen mehrere Verwendungsmöglichkeiten besitzen, welcher dadurch in den Arbeitsalltag Einzug hält, verändert automatisch eine klassische Arbeitsweise. Die Arbeit mit unterschiedlichen Ebenen, welche eine gesamte Zeichnung darstellen können, wird

durch die Einführung von Auswahlsets begünstigt und erlaubt es diese Sammlungen auch unterschiedlichen Layouts zuordnen zu können. Dadurch wird es nicht nur ermöglicht, dass mehrere Benutzer an einem Plan arbeiten können, sondern auch dass Ebenen ebenfalls in anderen Auswahlsets verwendet werden können. Durch die Möglichkeit verschieden Ebenen in einer solchen Sammlung zu diktieren, ob sie druckbar sind oder nicht, wird den Nutzern erlaubt auch Ebenen in ihrer Bearbeitung einblenden zu können, welche für die Erledigung einer Aufgabe als Grundlage dienen, jedoch nicht in einem Ausdruck gezeigt werden sollen. Da dabei nicht immer alle Ebenen einer Sammlung angezeigt werden müssen, sondern für jede Aufgabe nur die nötigen Ebenen angezeigt werden können, wird die Übersichtlichkeit der Planungsunterlagen bei der Erledigung von Aufgaben stark erhöht und fördert damit zusätzlich die Konzentration eines Bearbeiters auf eine bestimmte Aufgabe.

Durch den Umstand, dass jeder Benutzer bei seiner Arbeit mit einem bestimmten Auswahlset oder einer Variante immer die dafür aktuellsten Ebenen zur Verfügung gestellt bekommt, wird es für alle Beteiligten leichter sich gegenseitig abzustimmen. So kann, wie auch schon im Test gezeigt, eine Besprechung erleichtert werden, da jeder Teilnehmer die selben Informationen besitzt und nicht vorab geklärt werden muss ob alle Beteiligten auch den selben Stand der Informationen besitzen. Zusätzlich wird die Kommunikation auf kurzem Wege gefördert, da die Zeitintervalle, bei welchen die neuesten Daten ausgetauscht werden, auf die Erledigung einzelner Aufgaben in einem Projekt beschränkt sind und somit jedem Mitarbeiter die Möglichkeit gegeben wird, sofort nach Übergabe der erledigten Arbeit diese mit seiner Arbeit abzustimmen und zu kommentieren.

Neben der Notwendigkeit die Arbeitsweise nicht von vornherein umstellen zu müssen, bietet das Plug-In auch die Chance mit Alternativen und Historischen Projekthinhalten zu arbeiten. Durch die Art und Weise wie ein Versionskontrollsystem Daten und deren Geschichte verwaltet, wird es Möglich, dass jegliche Version des Projektes in einer Alternative weiterbearbeitet werden kann. Mithilfe dieser Funktionalität kann ein Projekt nicht nur gleichzeitig durch verschiedene Bearbeiter bearbeitet werden, sondern sie können zu jeder Version des Projektes zurückkehren und von dort aus eine neue Alternative starten oder einen Teil der Geschichte des Projektes wieder in die Gegenwart holen um den Gegenwärtigen zu ersetzen. Dabei gehen aber die ersetzten Informationen nicht verloren, vielmehr wird einfach nur eine neue Version erstellt. Wenn die Nutzer sich entschließen eine neue Alternative zu kreieren, sind sie in der

Lage dabei nur einzelne Elemente einer Version als Grundlage für eine Alternative auszuwählen, wodurch sie weiterhin in der Lage bleiben alle anderen Elemente und Ebenen in der Version der Hauptentwicklung aktuell verwenden zu können. Oder das Team entschließt sich die komplette Planung zur Version X als Basis für eine neue Alternative zu nehmen, wodurch diese Alternative völlig losgelöst von der Hauptentwicklungslinie bearbeitet werden kann. Bei der Planung von Ausbauten in einem Projekt wird es dadurch Möglich jede Mietereinheit als eigenständiges Projekt in einem Projekt zu führen und diese schon frühzeitig zu beginnen, da die Rohbauplanung immer mit dem Unterprojekt synchron gehalten werden kann. Dies ist deshalb möglich, da ein Versionskontrollsystem wie Subversion alle Elemente, welche durch das System verwaltet werden, immer zu einer globalen Version zusammenfasst und dadurch die aktuelle Rohbauplanung immer dem aktuellen Versionsstand des Unterprojektes entspricht.

Die automatisierte Verwaltung aller Ebenen, Alternativen und Auswahlsets erlaubt es den Benutzern sich, statt die Verwaltung manuell zu erledigen, auf die Planungsarbeit zu konzentrieren. Dabei gewinnt jeder Mitarbeiter nicht nur die Zeit die er mit der Verwaltung verbringen würde, sondern die gesamte Projektarbeit kann dadurch beschleunigt werden, da auch die unterschiedlichen Fachplaner, durch die ständig aktuellen Unterlagen, sehr viel effizienter planen können.

Bei jedem Beginn einer Aufgabe werden vom Plug-In die geöffneten Ebenen auf Unterschiede zur letzten Bearbeitung durch diesen Nutzer überprüft. Dies hat nicht nur den Vorteil, dass ein Benutzer vor Beginn seiner Arbeit sehen kann welche Dinge sich seit seinem letzten Zugriff auf eine Ebene verändert haben, sondern bietet auch die Möglichkeit für alle Beteiligten auf kurzem Wege, dass ihre eigene Arbeit durch die Kollegen kontrolliert werden kann. Durch diese Kontrollmöglichkeit können Fehler oder Unachtsamkeiten in der Planung, welche bei Nichtbeachtung ungewollte Auswirkungen auf den Projektverlauf haben könnten, oder Folgeprobleme produzieren würden, auf ein Minimum reduziert werden.

Für einen Projektverantwortlichen ergeben sich durch die Funktionen zur Benutzerverwaltung und Aufgabenplanung ungeahnte Möglichkeiten. Er kann die durch das System von den Benutzern automatisch generierten Aufgaben zur Berichtigung von Problemen oder zur Bereinigung von Schnittstellenproblemen, sowie die durch die Prüfer erzeugten Aufgaben, welche zu einer Freigabe führen könnten, verwalten und

auch an bestimmte Mitarbeiter verteilen. Zusätzlich ist es ihm auch Möglich neue Aufgaben zu erstellen und diese durch die Bearbeiter abarbeiten zu lassen. Dabei kann er auch steuern welcher Mitarbeiter welche Aufgaben zugeteilt bekommt und deren Wichtigkeit für den Projektfortschritt festlegen. Da das Plug-In mitprotokolliert welcher Mitarbeiter welche Aufgabe gerade bearbeitet, kann auch eine Ressourcenplanung durchgeführt werden und so sicher gestellt werden, dass kein Mitarbeiter überlastet wird und andere keine Aufgaben haben.

9.4 Einschränkungen bei der Arbeit

Eine Arbeitsweise mit der Unterstützung durch Versionskontrollsysteme funktioniert nur, wenn das Verwaltungssystem von einem Server aus operiert. Dabei braucht es spezieller Sicherheitsmaßnahmen. Bei einer Verwendung in einem Firmennetzwerk müssen die selben Sicherheitsmaßnahmen getroffen werden, wie sie auch heute schon durch die einzelnen Büros angewendet werden. Wird das System verwendet um verschiedene Standorte miteinander zu verbinden, braucht es wesentlich höhere Sicherheitsvorkehrungen. Denn dafür muss ein Zugang auch außerhalb des Intranets eines Büros möglich sein und dieser muss gegen Zugriffe von unberechtigten Dritten mittels Firewalls und anderen Abwehrmechanismen gesichert werden. Dabei geht es nicht ursächlich darum Daten zu schützen, wobei dies auch eine Priorität sein sollte, sondern vielmehr darum dass durch die Interaktion von Dritten die Archive zerstört werden könnten. Die Möglichkeit das Versionsarchiv von überall auf der Welt ansprechen zu können, stellt einen großen Vorteil für ein Unternehmen dar, da es dadurch den Mitarbeitern möglich wird von überall zu arbeiten und hilft bei der Integration von anderen Fachplanern welchen damit ebenfalls Zugang zu aktuellem Planmaterial gegeben werden kann, wenn diese vorab in die Benutzung des Tools eingeschult wurden beziehungsweise ein Plug-In zur Verfügung gestellt bekommen welches in eingeschränkter Form deren Mitarbeit an einem Projekt ermöglicht.

Ein Zugang zu den einzelnen Versionsarchiven muss dabei mindestens durch die Verwendung eines eindeutigen Benutzernamens und Kennworts erfolgen. Dies ist nicht nur nötig um Unbefugten den Zugang zu Informationen zu Verwehren, sondern durch die Verwendung von Benutzernamen und Passwort können einzelne Aktionen und Versionskommentare direkt einem Mitarbeiter zugeordnet werden. Diese Informationen braucht das Plug-In um überhaupt die richtigen Informationen vom Versionskontrollsystem zu beantragen und in weiterer Folge dem Benutzer zur

Verfügung stellen zu können. Zusätzlich wird mit Benutzernamen und Passwort auch der Zugang zu gewissen Funktionen des Plug-Ins gesteuert, so kann nur ein berechtigter Benutzer Veränderungen an den Benutzerlisten ändern, Freigaben durchführen, die Masterdateien pflegen, oder Änderungen an den Projektarchiven durchführen.

Bei heutigen Arbeitsweisen arbeitet ein Büro meist mit einem lokalen Server, welcher die Daten für die Benutzer zur Verfügung stellt, dadurch ist es den Benutzern möglich trotz des Ausfalls der Kommunikation mit den Kollegen in gewohnter Weise weiter zu arbeiten, solange eine Verbindung mit dem lokalen Server hergestellt werden kann. Wird das Projekt jedoch auf einem externen Server verwaltet um allen Projektbeteiligten die Mitarbeit zu ermöglichen, kann es zu einem Ausfall dieser Verbindung kommen und ein weiterarbeiten wird unmöglich, bis die Verbindung wiederhergestellt wurde. Sollte es also zu einem Ausfall des Servers oder der Kommunikation zu diesem kommen, können keine neuen Aufgaben mehr übernommen werden und auch keine beendeten Aufgaben an die Versionskontrolle übergeben werden. Hierbei hilft das Verhalten von Subversion den Benutzern insofern, da es nur atomare, also ganzheitliche Übertragungen, durchführt. Dadurch kann zumindest verhindert werden, dass unvollständige Übertragungen eine Bearbeitung beenden und dadurch Informationen verloren gehen. Nichts desto trotz ist das System bei einem Ausfall nur noch in einer geöffneten Bearbeitung verwendbar und kann auch keine Übertragungen mehr durchführen.

Nicht nur durch die Möglichkeit der gegenseitigen Kontrolle, sondern auch durch das Verhalten von Versionskontrollsystemen, welche jede Änderung mit Benutzer und Zeitstempel versehen, kann bei den Mitarbeitern das Gefühl der ständigen Überwachung entstehen. Denn durch die gesamten Vorteile, welche das Plug-In zur Aufgabenverteilung, Bearbeitung, Erstellung von Alternativen und alle anderen dem Benutzer zur Verfügung stellt, wird dieser auch zu einem, bis zu einem gewissen Grad, gläsernen Mitarbeiter. Bei der Ressourcenplanung wird einem Teamleiter angezeigt welche Mitarbeiter in welchem Projekt aktiv sind und welche Aufgaben sie dort gerade übernommen haben, zusätzlich werden auch die letzten abgeschlossenen Aufgaben angezeigt, dadurch wird es einfacher den Mitarbeiter spezielle Aufgaben zuzuteilen, da diese auf die vorhergehenden Aufgaben abgestimmt werden können. Ein Verantwortlicher hätte theoretisch dadurch die Möglichkeit der Überwachung, durch die Daten welche das Plug-In und die Versionskontrolle zur Verfügung stellt, seiner Mitarbeiter. Denn das System zeichnet zur eigenen Verwendung die gesamten

Tätigkeiten eines Mitarbeiters und über Umwege die Geschwindigkeit mit welcher dieser Aufgaben abwickelt, sowie wie oft auf diese eine Reaktion aufgrund von Fehlern geschieht, auf. Einerseits sollten diese Informationen genutzt werden um die Ressourcenverteilung besser zu planen, andererseits besteht dabei immer die Gefahr eines gewissen Missbrauchs, speziell da diese Daten nur Zahlen darstellen und keine Aussage über die Größe einer Aufgabe oder die Gründe einer Reaktion auf die beendete Bearbeitung bereitstellen.

Die Versionskontrolle behält zwar die gesamte Historie eines Projektes bereit, kann aber dabei nicht auf ein funktionierendes Backupsystem verzichten. Lediglich die Anzahl der Backups zu unterschiedlichen Zeiten kann verringert werden, da jedes Backup zum Zeitpunkt X alle Versionen eines Projektes zu diesem Zeitpunkt enthält und diese dadurch erhalten sind. So ist es nicht mehr nötig unterschiedlichste Backups zu unterschiedlichsten Zeiten von Projekten zu behalten. Im Versionskontrollsystem kann es andererseits nie zu einem Datenverlust kommen, solange das Archiv intakt ist, denn sollte in einer Version ein Fehler auftreten kann einfach auf eine der vorhergehenden Versionen zurückgesprungen werden und von diesem Punkt aus die Arbeit wieder fortgesetzt werden. Auch ist es möglich nur die defekten Dateien durch ihre Vorgängerversionen zu ersetzen. So reicht schon ein zuverlässiges Backup genügen um das gesamte Projekt und seine Geschichte zum Zeitpunkt X wieder herstellen zu können, sollte das gesamte Archiv nicht mehr verwendbar sein. Aus Sicherheitsgründen ist jedoch zumindest ein Backup des Backups und oder die Verwendung von parallelen Sicherungssystemen zu empfehlen um im Falle eines Fehlers in einem Backup auf einem Ersatz zurückgreifen zu können.

9.5 Ausblick in eine Zukunft mit Versionskontrolle

Durch die stärkere Vernetzung der einzelnen Planer im Architekturgeschehen wird es immer wichtiger die Informationen sicher und aktuell unter allen Beteiligten zu teilen. Da die Schnittstellen zwischen den Beteiligten immer näher zusammen rücken werden, müssen den Beteiligten Werkzeuge mit auf den Weg gegeben werden mit deren Hilfe sie der immer schneller werdende Entwicklung der Projekte effizient entgegenreten können. Durch die daraus resultierende Gleichzeitigkeit von Planungsschritten wird es umso wichtiger, dass den Bearbeitern die Möglichkeit der gegenseitigen Kontrolle gegeben wird. Dadurch kann gerade bei einer immer schnelleren Abarbeitung von Aufgaben sichergestellt werden, dass Schnittstellenprobleme und Flüchtigkeitsfehler,

welche schwerwiegende Auswirkungen haben würden, sobald als Möglich entdeckt und behoben werden können. Gleichzeitig wird die Erhaltung von Informationen über den Projektablauf und -verlauf immer wichtiger. Da dadurch viele Einzelschritte für zukünftige Projekte gesichert und in diesen wiederverwendet werden können. Diese durch viele Projekte aufgebaute Wissens und Detaildatenbank erleichtert den einzelnen Planern bei neuen Projekten die Arbeit, da auf schon verifizierte Elemente und Lösungsansätze schnell für ein neues Projekt zurückgegriffen werden kann.

Ein effizientes Aufgabenverteilsystem entlastet nicht nur die einzelnen Verantwortlichen sondern ermöglicht es auch den Planern einen Überblick über die bevorstehenden Arbeiten zu behalten und bei Fragen zu einer Aufgabenstellung sofort mit den beteiligten Planern und Fachplanern Rücksprache halten zu können. Um diese Rücksprachen so effizient wie möglich zu gestalten, ist es eine Grundvoraussetzung dass alle Planer über den selben Wissensstand und die selben aktuellen Unterlagen verfügen. Dadurch können interne Besprechungen im generellen kürzer gehalten werden, da eine Abstimmung über gewisse Entwicklungsschritte schon vorab über kurze Kommunikationswege möglich wird.

Versionskontrollsysteme und Programmiererweiterungen, wie das in dieser Arbeit erarbeitete Plug-In, stellen dabei einen sehr guten Weg dar, die Anforderungen an den Architektur Arbeitsplatz der Zukunft abzudecken und für Weiterentwicklungen noch genug Raum offen zu lassen. Deshalb kann davon ausgegangen werden, dass Systeme, wie das in dieser Arbeit gezeigte, die Möglichkeiten der Versionskontrollsysteme in der Architekturarbeit etablieren werden.

Literaturverzeichnis

An Engineering Document Management System, Hannu Peltonen, Tomi Männistö, Kari Alho, Reijo Sulonen, 1993, Helsinki University of Technology , ,, http://www.researchgate.net/publication/2727482_An_Engineering_Document_Management_System, Zugriff am 30.04.2015

Beyond the Shared Whiteboard: Issues in Computer Supported Collaborative Design, Wassim M. Jabi, Theodore W. Hall, 1995, in Proceedings of CAAD Futures '95: The Sixth International Conference on Computer-Aided Architectural Design Futures, Singapore , ISBN 9971-62-423-0, S. 719-725, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.4506&rep=rep1&type=pdf>, Zugriff am 12.03.2015

Design Versioning - Problems and Possible Solutions for the Automatic Management of Distributed Design Processes, Sven Schneider, Jörg Braunes, Torsten Thurow, Reinhard König, 2011, Computer Aided Architectural Design Futures 2011, Liege, Belgien , ISBN 9782874561429, S. 669-681, http://cumincad.scix.net/data/works/att/cf2011_p042.content.pdf, Zugriff am 30.04.2015

Einführung eines Product-Lifecycle-Management-gestützten Änderungsprozesses in einem internationalen Großprojekt, Anneke Lühr, 2011, Hochschule für angewandte Wissenschaften Hamburg , ,, http://edoc.sub.uni-hamburg.de/haw/volltexte/2012/1556/pdf/Luehr_Anneke_110831.pdf, Zugriff am 30.04.2015

Entwicklung eines verteilbaren und kooperativ nutzbaren objektorientierten CAAD-Produktmodellierkerns, T. Hauschild, R. Hübler, 1997, in Internationales Kolloquium über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen , IKM , 14 , 1997 , Weimar , Bauhaus-Universität , ,, <http://e-pub.uni-weimar.de/opus4/frontdoor/index/index/docId/447>, Zugriff am 16.03.2015

Open Source Development with CVS, 3rd Edition, Karl Fogel, Moshe Bar, 2003, , 1-932111-81-6,, http://cvsbook.red-bean.com/OSDevWithCVS_3E.pdf, Zugriff am 15.05.2015

Past, present and future issues with co-operating, seen from the engineering practice, Gernot Pittioni, 2001, in ACCOLADE - Architecture, Collaboration, Design, Delft University Press (DUP Science) , ISBN 90-407-2216-1, S. 83-94, <http://cumincad.architexturez.net/doc/oai-cumincadworks.id-8c32>, Zugriff am 17.03.2015

Pro Git second edition, Scott Chacon & Ben Straub, 2014, , 978-1484200773, The Expert's Voice, <https://progit2.s3.amazonaws.com/en/2015-05-02-7ddee/progit-en.478.pdf>, Zugriff am 07.05.2015

RCS—A System for Version Control, Walter F. Tichy, 1985, Purdue University Department of Computer Sciences , ,, <http://users.ninthfloor.org/~ashawley/rcs/tichy1985rcs/rcs.html>, Zugriff am 07.05.2015

Synchronous Support and Emergence in Collaborative CAAD, Mary Lou Maher, John S Gero, Milad Saad, 1993, in U. Flemming and S. Van Wyk (eds), CAAD Futures 1993, North-Holland, Amsterdam , ISBN 0-444-89922-7, S. 455-470, <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=2C8021D2616A45A479F7893FA46FFF9B?doi=10.1.1.86.5404&rep=rep1&type=pdf>, Zugriff am 26.03.2015

Towards a Process-Centric, Asynchronous Collaborative Design Environment, Jeng Tay-Sheng, 2001, in Proceedings of the Fifth Conference on Computer-Aided Architectural Design Research in Asia, 15-24. CAADRIA, National University of Singapore , ISBN 981-04-2491-4, S. 15-24, <http://cumincad.architexturez.net/doc/oai-cumincadworks.id-1a59>, Zugriff am 19.03.2015

Understanding Version-Control Systems (DRAFT), Eric Raymond, 2008, , ,, <http://www.catb.org/~esr/writings/version-control/version-control.html>, Zugriff am 05.05.2015

Versioning in early-stage hospital simulation, Gabriel Wurzer, 2013, in Proceedings of the International Workshop on Innovative Simulation for Health Care , ISBN 978-88-97999-26-3, S. 65-71, ,

Versionskontrolle mit Subversion: Für Subversion 1.7: (Übersetzt aus der Revision 5039), Ben Collins-Sussman, Brian W. Fitzpatrick und C. Michael Pilato, 2011, , ,, <http://svnbook.red-bean.com/de/1.7/svn-book.pdf>, Zugriff am 07.05.2015

Abbildungsverzeichnis

Abb. 1: Verfahren bei der Bearbeiten entsperren Funktion (eigene Abbildung).....	18
Abb. 2: Verfahren bei der merge before commit Funktion (eigene Abbildung).....	19
Abb. 3: Verfahren bei der commit before merge Funktion (eigene Abbildung).....	20
Abb. 4: Versionskontrolle mit Momentaufnahmen aus Pro Git second edition, Scott Chacon & Ben Straub, 2014, Seite 32.....	20
Abb. 5: Versionskontrolle mit Änderungsverfolgung aus Pro Git second edition, Scott Chacon & Ben Straub, 2014, Seite 32.....	21
Abb. 6: Versionskontrolle in RCS – einzelne Dateien haben eigene Version (eigene Abbildung).....	23
Abb. 7: Versionskontrolle in Subversion – das gesamte Archiv bildet eine Version (eigene Abbildung).....	26
Abb. 8: Versionskontrolle mit GIT – Arbeitsablauf während der Bearbeitung (eigene Abbildung).....	28
Abb. 9: Verästelungen während eines Projektablaufes (eigene Abbildung).....	50
Abb. 10: Maske zur Projektinitialisierung (eigene Abbildung).....	51
Abb. 11: Projektauswahlmaske des Plug-Ins (eigene Abbildung).....	57
Abb. 12: Bearbeitung mit dem Plug-In starten (eigene Abbildung).....	57
Abb. 13: „Ebenen zur Ansicht wählen“ Maske des Plug-Ins (eigene Abbildung).....	58
Abb. 14: Ablauf der Überprüfung auf Änderungen (eigene Abbildung).....	60
Abb. 15: Maske zur Auswahl des zu bearbeitenden Layouts (eigene Abbildung).....	66
Abb. 16: Lageplan des Testprojektes in 1100 Wien, Ausschnitt aus dem Flächenwidmungsplan für Wien, http://www.wien.gv.at/flaechenwidmung/public/start.aspx , Zugriff am 30.06.2015...71	71
Abb. 17: der Arbeitsablauf bei der Arbeit mit unterschiedlichen Ebenen (eigene Abbildung)....	77
Abb. 18: Der Entwicklungsstand vor Beginn der Phase III mit eingefärbten Ebenen (eigene Abbildung).....	78

Abb. 19: Planausschnitt Endstand nach Abschluss der Phase III mit eingefärbten Ebenen (eigene Abbildung).....	79
Abb. 20: Planausschnitt Wohnung 1 Endstand der Phase III mit eingefärbten Ebenen (eigene Abbildung).....	80
Abb. 21: Planausschnitt Detail Lärmschutz zu Bahnlinie (eigene Abbildung).....	80

Lebenslauf

Wien, August 2015

Josef Öhreneder

Herndlgasse 4/18
1100 Wien

Tel.: +43 (0) 699 / 11765239

E-Mail: josef.oehreneder@gmail.com

geb.: 19.08.1982 Gmunden, OÖ



Ausbildung:

- 1993-2001 HIB Schloss Traunsee, Gmunden
- 2005 Ausbildung zum Rettungssanitäter und Einsatzlenker
- 2001-2015 Technische Universität Wien - Architektur
- Okt. 2015 Diplomarbeit an der Technischen Universität Wien
Thema:
Versionierung im Zuge der Konzept- &
Entwicklungsphase anhand eines AutoCad Plug-Ins

Berufserfahrung:

- 2008 Mitarbeit im Büro Deinhammer (Eferding OÖ)
sowie Mitarbeit an diversen Wettbewerben
- 2009-2014 Planer / Junior Projektleiter Ingenos Gobiet ZT GmbH
Projekterfahrung in allen Planungsphasen, ÖBA für
Innenausbau,
Wartung und Weiterentwicklung der CAD Standards und
Automatisation von Autocad, Keyuser für AutoCad &
MS Office

Sprachliche Kenntnisse:

- Deutsch (Muttersprache)
- Englisch (fließend)
- Französisch (Schulkenntnisse)
- Spanisch (Grundkenntnisse)

Sonstige Qualifikationen:

- Programmkenntnisse: Autocad 2000 – 2015, Rinceros, Microsoft Office
(inkl. Access), VBA, VB.NET für Autocad,
Adobe Creative Suit 4-6, Archi Physik (Grundkenntnisse)
- 2005 – 2006 Zivildienst beim Roten Kreuz Oberösterreich
- 2006 – 2015 Freiwilliger Mitarbeiter Rotes Kreuz OÖ Bad Ischl (derz.
Ruhend)
- Führerschein der Klassen A & B inkl. Einsatzlenker Rotes
Kreuz OÖ

Hobbys:

Apnoe, Klettern, Wandern, Radfahren, Baumschnitt und Veredelung,
Gerätetauchen, Reisen

Projektreferenzen:

Junior Projektleitung:

- diverse Mieterausbauten Marximum,
- Umbau Sandoz Novartis Viertel Zwei
- Thermische Sanierung Clerfaytgasse (Architektur)
- Brandschutzplanerstellung für diverse Projekte

ÖBA Innenausbau Umbauten Sandoz Novartis Viertel Zwei

Mitarbeit bei:

- Hoher Markt 10 „Palais Principe“,
- Unternehmenszentrale Bestattung Wien (+Vertretung der
Projektleitung),
- Nordbahnstraße 3 „Smart Office“,
- Lokalausbau JOMA Hoher Markt (+Vertretung der
Projektleitung),
- Neurologisches Rehazentrum Kittsee

CAD Koordinator für das Projekt Nordbahnstraße 3 „Smart Office“
Unterstützung der Fachplaner im In- und Ausland

Aufmaß und Erstellung von Bestandsunterlagen im Neu- und Altbau

Bestandserhebung und Aufmaß der Autobahnmeistereien in
Wien und Umgebung

diverse Machbarkeitsstudien für HochTief Development in Wien

Fertigstellungen verschiedenster Projekte

Mietbereichsplanung nach GIF für Mietverträge IZD Tower

Automatisation von Planungsprozessen

(Plandokumentation, Flächenkalkulation, ...)