

2009

VON EISBERGEN UND SUPERTANKERN: TOPOLOGIE EINES CAMPUS- MANAGEMENTEINFÜHRUNGSPROZESSES

Monique Janneck
Universität Hamburg

Cyrill Adelberger
Universität Hamburg

Sabine Fiammingo
Universität Hamburg

Ruth Luka
Universität Hamburg

Follow this and additional works at: <http://aisel.aisnet.org/wi2009>

Recommended Citation

Janneck, Monique; Adelberger, Cyrill; Fiammingo, Sabine; and Luka, Ruth, "VON EISBERGEN UND SUPERTANKERN: TOPOLOGIE EINES CAMPUS-MANAGEMENTEINFÜHRUNGSPROZESSES" (2009). *Wirtschaftsinformatik Proceedings 2009*. 126.
<http://aisel.aisnet.org/wi2009/126>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2009 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

VON EISBERGEN UND SUPERTANKERN: TOPOLOGIE EINES CAMPUS-MANAGEMENT- EINFÜHRUNGSPROZESSES

Monique Janneck, Cyrill Adelberger, Sabine Fiammingo,
Ruth Luka¹

Kurzfassung

Der Beitrag schildert Charakteristika und Probleme der Einführung eines Campus-Management-Systems an einer deutschen Hochschule. Die Ergebnisse untermauern die Bedeutung von Organisationsentwicklungsmaßnahmen – ein erheblicher Teil der Probleme lässt sich auf einen Mangel in diesem Bereich zurückführen, deutlich mehr als auf technische Probleme. Zudem hängen Erfolg und Misserfolg – bislang in Forschung und Praxis wenig beachtet – stark von personellen Faktoren wie vor allem der Eigeninitiative der Nutzer ab. Entsprechende Implikationen werden abgeleitet.

1. Einleitung

Auch traditionelle Organisationen sind einer zunehmenden Virtualisierung ausgesetzt: Neben allgegenwärtigen Kommunikationstechnologien durchdringen und verändern große betriebliche Softwaresysteme organisationale Prozesse. In diesem Beitrag beleuchten wir einen Technologieeinführungsprozess, der Veränderungen in nahezu allen Teilbereichen und -prozessen der Organisation mit sich brachte: die Einführung eines Campus-Management- (CM) Systems an einer großen deutschen Hochschule. Diese Software dient der Verwaltung nahezu aller studienbezogenen Vorgänge, von der Studienplatzvergabe und Immatrikulation über die Lehrveranstaltungsplanung und Seminarvergabe bis zur Kontrolle und Bescheinigung von Studien- und Prüfungsleistungen. Sie löste, auch mit dem Ziel einer stärkeren Vereinheitlichung, verschiedenste sowohl EDV- als auch papierbasierte Verwaltungsvorgänge ab und etablierte im Zuge der Einführung von Bachelor- und Masterstudiengängen neue Studien-, Personal- und Organisationsstrukturen. Nahezu alle Universitätsangehörige – Verwaltungsmitarbeiter, Lehrende und Studierende – waren und sind davon betroffen. Die Einführung der Software war von erheblichen technischen und organisatorischen Schwierigkeiten begleitet und wurde von vielen Beteiligten als große Last empfunden. Um Faktoren zu identifizieren, die eine erfolgreiche Implementierung behindern bzw. begünstigen, wurde der Einführungsprozess anhand einer qualitativen Interviewstudie untersucht.

¹ Universität Hamburg, Fachbereich Psychologie, D-20146 Hamburg, Von-Melle-Park 11

2. Softwareeinführung als Rekontextualisierung

Softwareeinführungsprozesse lassen sich als *Rekontextualisierung* formalisierter Praxis verstehen, der eine *Dekontextualisierung* vorausgegangen ist, nämlich die „Übersetzung“ menschlicher Handlungen und organisationaler Praxis in formalisierbare Strukturen und schließlich Algorithmen [5]. Sesink [11] und Crutzen [2] verwenden gar die Begriffe „Dekonstruktion“ und „Rekonstruktion“, um zu verdeutlichen, dass Technologie menschliche Handlungen nicht nur „unterstützt“, wie dies häufig formuliert wird, sondern unter Umständen zerstört und durch etwas Neues ersetzt, was sowohl positive als auch negative Konsequenzen haben kann – die häufig nur schwer vorhersagbar sind. Ob ein Softwareentwicklungsprozess erfolgreich war und zu einem hochwertigen Produkt geführt hat, zeigt sich also häufig erst während der Einführungsphase, der Rekontextualisierung. Diese jedoch liegt typischerweise nicht im Hauptfokus der Softwareentwickler (vgl. [4]).

Die Rekontextualisierung von Software ist häufig von Konflikten begleitet [5], die einigen zentralen Faktoren zugeschrieben werden können:

- *Restriktionen durch Formalisierung.* Formalisierung menschlicher Handlungen durch Softwareeinsatz geht häufig mit (durchaus gewollter) erhöhter Standardisierung einher, die einerseits (Arbeits-) Prozesse effizienter und transparenter machen, andererseits aber auch jene Flexibilität verhindern kann, die notwendig ist, um auf Unerwartetes und Ausnahmefälle reagieren zu können (vgl. [5]). Um die Grenzen der Formalisierung zu beschreiben, sprechen Krause et al. [5] von *vorläufigen* und *notwendigen Formalisierungslücken*: Während vorläufige Lücken Prozesse beschreiben, die einer Automatisierung prinzipiell zugänglich sind, bezeichnen notwendige Formalisierungslücken Aktivitäten und Prozesse, die einer hohen Flexibilität bedürfen und nur sehr vorsichtig oder gar nicht automatisiert werden sollten.
- *Vergegenständlichung (verborgener Strukturen).* Die Formalisierung von Aufgaben im Rahmen von Softwareentwicklung kann Strukturen, Prozesse und Beziehungen ans Licht bringen, die vorher im Dunkeln gehalten wurden. Resultierende Konflikte werden zwar nicht von der Technologie ausgelöst, aber diese trägt dazu bei, sie aufzudecken [3].
- *Schaffung neuer Strukturen.* Neue Technologie deckt nicht nur existierende Strukturen auf, sie bringt auch fast unvermeidlich neue Strukturen und Routinen mit sich [3]. Folglich müssen sich die betroffenen Akteure anpassen und ihre Gewohnheiten und Handlungen verändern. Womöglich erleben sie Veränderungen hinsichtlich ihres Status oder ihrer Position – wovon einige Akteure profitieren, andere jedoch nicht.
- *Rekontextualisierung in einem neuen Kontext.* Speziell beim Einsatz von Standardsoftware finden Dekontextualisierung und Rekontextualisierung in unterschiedlichen Kontexten statt: Die Software wird für einen abstrakten oder idealisierten Nutzungszweck entwickelt, der von der tatsächlichen Nutzung stark abweichen kann. Entsprechend schwierig kann es für die Nutzer sein, zugrunde liegende Designprinzipien zu verstehen und sie in Beziehung zu ihren eigenen Aufgaben und Interessen zu setzen.
- *Angst vor Veränderung.* Jenseits konkreter Änderungen durch die Einführung neuer Technologien kann diese – wie andere Veränderungsprozesse auch – bei den Beteiligten Ängste auslösen, die – begründet oder nicht – die Technologieaneignung und damit verbundene Organisationsentwicklung behindern oder sogar zum Boykott führen können.

All diese Schwierigkeiten und Phänomene ließen sich auch in der vorliegenden Untersuchung beobachten. Sie werden in Abschnitt 4 dargestellt.

3. Forschungskontext und Methodik

Die hier dargestellte Untersuchung hat die Einführung einer universitätsweiten Campus-Management- (CM) Software zur Verwaltung von Studierendendaten, Studien- und Prüfungsleistungen bis hin zum Raum- und Lehrveranstaltungsmanagement zum Gegenstand. Dieser umfangreiche und anspruchsvolle Softwareeinführungsprozess war und ist von erheblichen technischen und organisatorischen Schwierigkeiten begleitet. Ziel der Untersuchung war es, Faktoren zu identifizieren, die eine erfolgreiche Einführung behinderten bzw. begünstigten. Darauf aufbauend sollen Maßnahmen entwickelt werden, um die organisatorische Einbettung der Software zu verbessern.

Zur Datenerhebung wurden 35 halbstrukturierte qualitative Interviews (vgl. [6]) durchgeführt. Um eine große Bandbreite unterschiedlicher Sichten zu erreichen, wurde auf eine starke Heterogenität der Interviewpartner geachtet, die in unterschiedlichen Rollen und unterschiedlicher Intensität an der Softwareeinführung beteiligt waren (Verwaltungskräfte, Dekane, Planer, Modellierer, Sekretariate, Softwareentwickler, Projektmanager, Support-Mitarbeiter, studentische Vertreter). Die Interviews wurden aufgezeichnet und anschließend im Wortlaut transkribiert. Anhand einer Teilmenge der Interviews wurde ein Kategoriensystem induktiv generiert. Die verbleibenden Interviews wurden deduktiv weitercodiert, wobei z.T. noch neue Subcodes entstanden bzw. bestehende Unterkategorien zusammengefasst wurden. Jedes Interview wurde von 2-3 unabhängigen Personen codiert, wobei sich eine hohe Inter-Rater-Übereinstimmung fand. Insgesamt wurden ca. 4200 Codierungen vergeben (Doppelcodierungen waren möglich), die anschließend pro Kategorie weiter zusammengefasst und paraphrasiert und z.T. auch quantitativ ausgewertet wurden.

4. Ergebnisse

Die Kategorien, die aus den Interviewdaten herausgearbeitet wurden, lassen sich anhand der bekannten *Leavitt-Raute* (vgl. [7]) mit den Einflussfaktoren der Mensch-Computer-Interaktion in Organisationen veranschaulichen (Abb. 1):

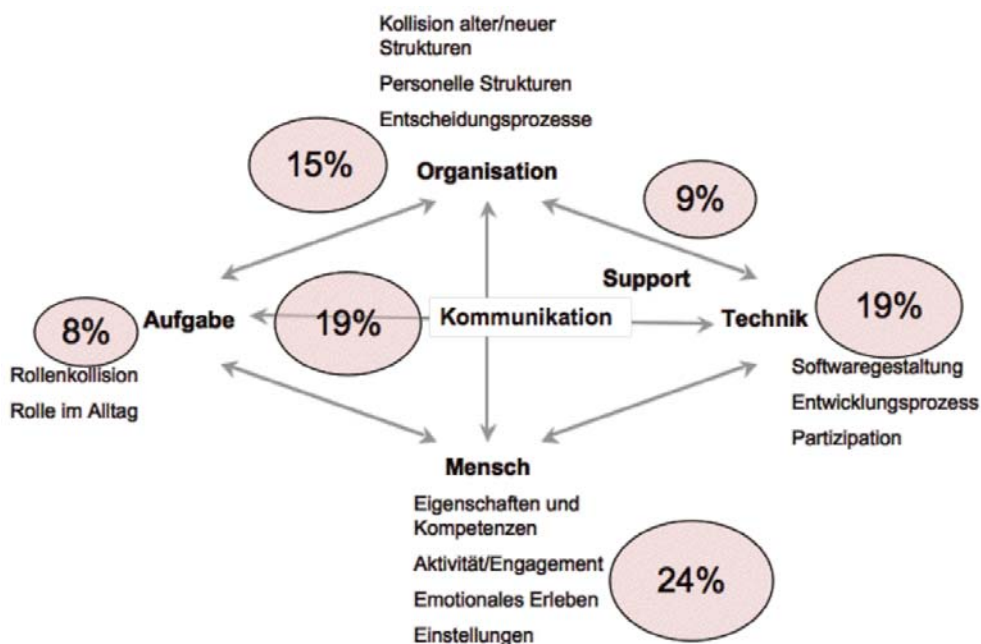


Abb. 1: Einordnung der Kategorien (mit Unterkategorien) in die Leavitt-Raute, Prozentangaben: Anteil an allen Codierungen (fehlende %: Sonstiges)

Auf der Seite der *Technik* lassen sich die Aussagen der Befragten zum Entwicklungsprozess sowie zur Gestaltung der Software verorten, auf der Seite der *Organisation* diejenigen Kategorien, die organisatorische Strukturen und Prozesse und deren Veränderung betreffen, unter *Aufgabe* die Aussagen zu arbeitsorganisatorischen Aspekten und Veränderungen und unter *Mensch* alle Kategorien, die personelle Faktoren wie Eigenschaften und Verhaltensweisen der Befragten betreffen. Die Kategorien „Kommunikation“ sowie „Support“ wurden als übergreifend eingestuft und können entsprechend im Netz der Raute eingeordnet werden. Betrachtet man die Anzahl der Codierungen für die jeweiligen Faktoren (Abb. 1 & 2), sticht die Kategorie „Mensch“ mit 24% der Codierungen als größte Einzelkategorie heraus, gefolgt von „Kommunikation“ mit 19%. Der Faktor „Technik“ vereint ebenfalls 19% der Codierungen auf sich. Auf die Kategorie „Organisation“ entfielen 15% der Codierungen, auf „Aufgabe“ 8%, auf „Support“ 9%. Unter „Sonstiges“ wurden Einzelkategorien mit jeweils nur vergleichsweise wenigen Nennungen subsumiert (6%).

Knapp 20% der Textstellen wurden *zusätzlich* zur inhaltlichen Kategorisierung als „Probleme“ codiert, um zu analysieren, in welchen Oberkategorien besonders häufig Schwierigkeiten beschrieben werden. Dies ist bei der Kategorie „Organisation“ der Fall: Diese schlägt insgesamt nur mit ca. 15% der Codierungen zu Buche, macht jedoch 26% der Problemnennungen aus – überdurchschnittlich viele Probleme treten offenbar in diesem Bereich auf. In den übrigen Kategorien entspricht die Anzahl der Problemnennungen dagegen fast genau dem jeweiligen Gesamtanteil (Abb. 2).

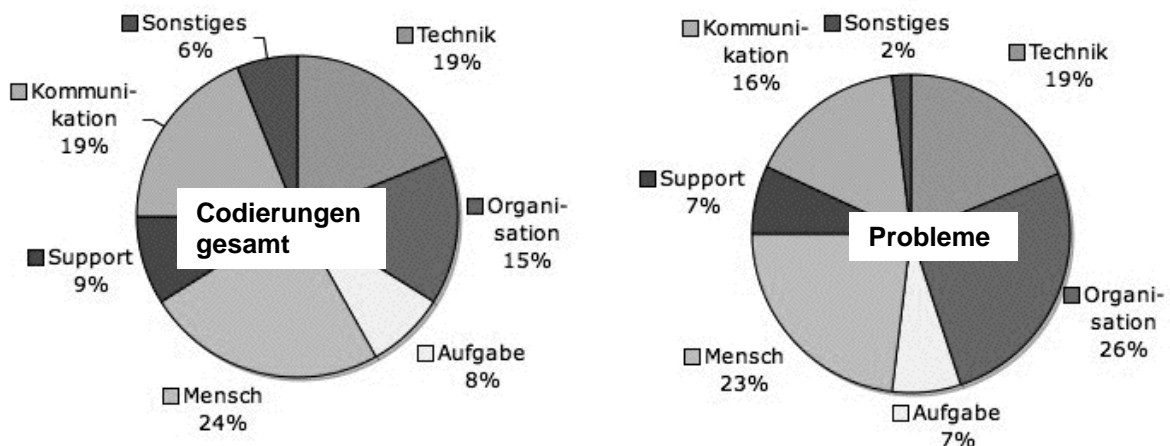


Abb. 2: Verteilung der Codierungen insgesamt (links) sowie der Probleme (rechts) im Vergleich

Die vier Oberkategorien „Technik“, „Organisation“, „Aufgabe“ und „Mensch“ wurden jeweils von nahezu allen Befragten angesprochen, ebenso „Kommunikation“. Allerdings variiert die Zahl der Interviewpartner teilweise erheblich innerhalb der Unterkategorien.

In den folgenden Abschnitten wird die Einführung des CM-Systems, verstanden als Rekontextualisierungsprozess auf den vier Ebenen der Leavitt-Raute, beschrieben und analysiert. Die Zwischenkategorien „Kommunikation“ und „Support“ werden aus Platzgründen nur gestreift.

4.1. Technische Rekontextualisierung

Der Entwicklungsprozess ist gekennzeichnet durch eine vergleichsweise lange Produktauswahlphase und eine im Vergleich dazu kurze Entwicklungsphase bis zum Produktivstart des Systems (Abb. 3). (Die Dauer dieser Vorbereitungsphase wurde allerdings von den Befragten sehr unterschiedlich wahrgenommen, was einerseits aus der stark variierenden direkten Involviertheit zu erklären und verständlich ist, andererseits aber als Hinweis auf mangelnde Transparenz und Prozesskommunikation gesehen werden kann). Der Zeitplan wurde ausnahmslos als gravierend zu knapp und die bereitgestellten Ressourcen – sowohl für die technische als auch die organisatorische Einführung – als

viel zu gering eingeschätzt. Während jedoch von Entwicklerseite geäußert wurde, die Zeit- und Ressourcenproblematik sei bekannt gewesen und letztlich in Kauf genommen worden, herrscht auf Nutzerseite der Eindruck vor, Aufwand und Komplexität des Projekts seien unterschätzt worden – was entsprechend mit einer negativen Bewertung der Projektverantwortlichen einhergeht.

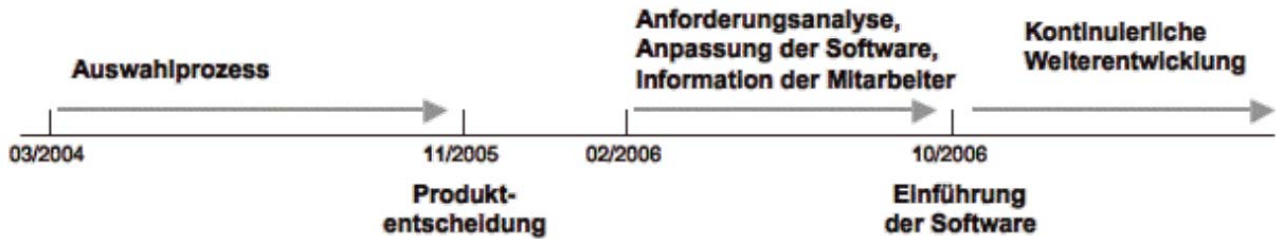


Abb. 3: Zeitstrahl

Zur Produktauswahl standen mehrere Systeme, von denen keines als Ideallösung angesehen wurde. Mit ausschlaggebend für die Entscheidung war letztlich das Angebot einer Entwicklungspartnerschaft, in deren Rahmen die Software schrittweise über mehrere Jahre hinweg den Bedürfnissen der Universität angepasst werden sollte.

Die Chancen zu einer *partizipativen* Softwaregestaltung unter Einbezug möglichst vieler Beschäftigter, die dieses Modell der Entwicklungspartnerschaft bot, wurden jedoch nur sehr eingeschränkt wahrgenommen: Zwar wurden Vertreter der Fachbereiche/Fakultäten durch Mitarbeiter der Universitätsverwaltung zu den Prozessen und Strukturen an ihren Einrichtungen befragt, jedoch ist die beinahe einhellige Einschätzung der Befragten, eine ausreichend gründliche Anforderungsanalyse habe nicht stattgefunden. Dies zeigt sich auch in der quantitativen Auswertung der Ergebnisse: Nur 4% der Codierungen sind der Kategorie „Partizipation“ zuzuordnen, nur etwas mehr als die Hälfte der Befragten (n=20) äußerten sich überhaupt zur Anforderungsanalyse.

Die Maßnahmen zur Nutzerbeteiligung wurden bei der Codierung unterschieden in *Pull-Maßnahmen*, d.h. Aktivitäten, die Nutzer selbst angestrengt haben, um Einfluss auf den Entwicklungsprozess zu nehmen, und *Push-Maßnahmen*, also Aktivitäten von Entwicklerseite, um die Nutzer zu involvieren. Letztere wurden sehr selten – von gerade einmal fünf Befragten – geschildert, während immerhin doppelt so viele (n=11) von Pull-Aktivitäten berichten².

Insgesamt entsteht der Eindruck, dass die Beteiligung am Entwicklungsprozess bzw. an der Anforderungsanalyse sehr stark von der persönlichen Initiative, Bereitschaft und auch Hartnäckigkeit der jeweiligen Personen abhängig war, sich zu Wort zu melden und gerade auch Probleme und Bedenken zu thematisieren. Nach Einschätzung vieler Befragter wäre eine breitere Beteiligung für die Akzeptanz des Systems sowie die Durchsetzung von Veränderungen sehr nützlich gewesen. Aufgrund der mangelnden Akzeptanz entwickelte sich die Software zunehmend als Sündenbock für negativ bewertete Veränderungen, die jedoch in anderen organisatorischen Prozessen begründet liegen (s.u.). Zudem berichten mehrere Befragte von „gefühlten“ Veränderungen durch die augenscheinliche Übernahme von Verwaltungsabläufen durch ein technisches System. Das CM-System wird als quasi eigenständiger Akteur wahrgenommen, dem man sich, da die Funktionsweise der Software im Dunkeln bleibt, ausgeliefert fühlt.

Darüber hinaus zeigten sich insbesondere in der Einführungsphase noch erhebliche technische Mängel und Probleme der Gebrauchstauglichkeit der Software, die zu Frust und Ablehnung bei den Nutzern führten.

² Ein ähnliches Bild zeigt sich auch im Hinblick auf Supportmaßnahmen, die hier nicht gesondert behandelt werden.

4.2. Organisatorische Rekontextualisierung

Eine grundsätzliche Schwierigkeit des Einführungsprozesses lag in der strukturellen Ausgangssituation. Die Universität bestand ursprünglich aus einer jahrzehntelang gewachsenen, dezentralen Struktur vieler Fachbereiche, die relativ unabhängig operieren konnten. Im Zuge des Bolognaprozesses und der damit verbundenen Einführung von Bachelor/Master-Studiengängen wurden jedoch erhebliche Umstrukturierungen notwendig. Auch wurde von Seiten der Universitätsleitung eine stärkere Vereinheitlichung und Zentralisierung gewünscht, die mit der Einführung der CM-Software erleichtert bzw. ermöglicht werden sollte. Diese Umstrukturierungen bargen erhebliches Konfliktpotential: Die „Kollision alter und neuer Strukturen“ bildet die größte Unterkategorie im Bereich Organisation, zwei Drittel der Befragten sprachen diese Thematik an. Die resultierenden Schwierigkeiten sind z.T. unabhängig von der Softwareeinführung: Die schiere Größe und die gewachsene, heterogene Struktur der Universität machen Veränderungsprozesse entsprechend energieaufwändig und träge – ein Befragter nutzte hier das Bild eines „Supertankers“, der nur langsam und schwerfällig den eingeschlagenen Kurs wechselt. Eine weitere Befragte nutzte für den Softwareentwicklungsprozess die ganz ähnliche Metaphorik eines „Eisberges“, der die „Riesentitanic“ Universität rammt und ihr, wenn er sie auch nicht ganz zum Untergang bringt, doch erheblichen Schaden zufügt.

Auf der anderen Seite entstand jedoch der Eindruck, die Prozesse sollten an die Software angepasst werden und nicht umgekehrt, gerade auch mit dem Ziel einer stärkeren fakultätsübergreifenden Vereinheitlichung. Während eine solche Vereinheitlichung prinzipiell z.T. durchaus auf Verständnis oder Zustimmung seitens der Befragten stieß, wurde wiederholt kritisiert, konkrete Abläufe seien nicht inhaltlich, sondern durch die Möglichkeiten der Software begründet worden. Problematisch war dabei, dass eben diese in der Software bereits vorgesehenen Strukturen – insbesondere das Rollen- und Rechtekonzept – in der Vorbereitungsphase, d.h. ohne reale Nutzungserfahrung, in ihren Auswirkungen kaum einschätzbar und vorhersehbar waren. Die Befragten sahen sich mit der Aufgabe konfrontiert, ins Ungewisse planen und Strukturen und Prozesse festlegen zu müssen, ohne diese erproben zu können – und mit der Erwartung, dass sie schwer zu verändern sein würden, wenn sie erst implementiert wären. Diese Aussicht, folgenschwere Entscheidungen ohne tragfähige Grundlage treffen zu müssen, resultierte vielmals in Zögerlichkeit, durch die sich der ohnehin enge Zeitplan für die Einführung noch verschärfte.

Entgegen der Wahrnehmung vieler Befragter wurde im Laufe der Entwicklung durchaus eine große Anzahl von Änderungswünschen aufgenommen und umgesetzt. Problematischerweise stellte der beschrittene Zwischenweg, der weit weniger Standardisierung mit sich brachte als ursprünglich angestrebt, offenbar keine Seite wirklich zufrieden. Der Softwareentwicklungsprozess sowie die Software selber litten unter der erheblichen Zunahme an Komplexität eines ohnehin schon vielschichtigen Systems. Die Organisationseinheiten fühlten sich dennoch oft in ein Korsett gepresst oder gedrängt, ihre funktionierenden Abläufe ohne inhaltliche Begründung der Software anpassen zu müssen.

Das Spannungsfeld zwischen Zentralität und Dezentralität zeigte sich auch im Bereich der *Kommunikation*, auf die an dieser Stelle als wesentliche Problemkategorie kurz eingegangen werden soll: Um Informationsflüsse in einer so großen Organisation zu gewährleisten, müssen „Kommunikationsknoten“ definiert werden, die Informationen nach bestimmten Regeln weitergeben und den Informationsfluss kontrollieren. Dies wird umso wichtiger, je weiter – sowohl organisatorisch als auch räumlich – Sender und Empfänger voneinander entfernt sind. Je weniger Kommunikationswege vorgeplant wurden, desto häufiger rissen Kommunikationsketten ab: Bei den Empfängern kamen wichtige Informationen häufig nicht oder zu spät an, den Sendern wiederum wurde oft gar nicht transparent, dass Informationen ihr Ziel nicht erreicht hatten. Darüber hinaus ist anzumerken, dass

Informationen häufig *informell* weitergegeben wurden: Der Grad der Informiertheit hing somit stark von der individuellen Involviertheit und Vernetzung (vgl. Abschnitt 4.4.) ab.

4.3. Aufgaben-Rekontextualisierung

Wie oben erwähnt, stellte sich als zentrales Problem die Rollen- und Rechtevergabe dar, die ursprünglich – auch aufgrund datenschutzrechtlicher Überlegungen – recht starr gestaltet wurde, was sich jedoch als nicht haltbar erwies, so dass nach und nach eine größere Flexibilität ermöglicht wurde. In der Praxis überschneiden sich Rollen häufig: Die implementierten Rollen erwiesen sich entsprechend als zu rigide und bildeten die Realität nicht ab. Z.T. waren für bestimmte Bereiche – etwa Sekretariate/Geschäftszimmer – ursprünglich gar keine Rollen bzw. Zugriffsrechte vorgesehen, wenngleich die entsprechenden Aufgaben dort anfielen.³ Die veränderte Rollenstruktur erforderte oftmals eine erhebliche Neuorganisation der Arbeitsabläufe – oder führte zu großen Problemen, wenn diese unterblieb. In der Wirkung nicht zu unterschätzen ist auch der Verlust an Aufgaben, Kompetenzen und/oder KnowHow, der zum Teil mit einer Neuorganisation der Aufgabenbereiche einherging und -geht. Dies hatte negative Konsequenzen sowohl für das Erleben der betroffenen Personen als auch für die Organisation, der so Wissen verloren gehen kann.

Die teilweise fehlende Funktionalität, mangelnde Gebrauchstauglichkeit, aber auch der große (zu erwartende) Funktionsumfang sorgte für beträchtlichen Zusatzaufwand in den Arbeitsprozessen. Die Benutzer hatten das Gefühl, sie müssten die technischen Möglichkeiten vollständig nutzen, auch wenn es, mit Abstand betrachtet, manchmal wenig zweckdienlich erschien. Für den Großteil der Befragten spielt die CM-Software mittlerweile eine immense Rolle in ihrem Arbeitsalltag, viele nutzen sie zumindest zeitweilig täglich mehrere Stunden lang. Dies unterstreicht die Bedeutung einer ergonomischeren Gestaltung der Software sowie einer verbesserten organisatorischen Einbettung: Das CM-System ist keine Randerscheinung, über deren Tücken man hinwegsehen kann, es hat den Arbeitsalltag vieler Beschäftigter stark durchdrungen.

4.4. Menschliche Rekontextualisierung

Wie oben beschrieben, nahmen personelle Faktoren den größten Raum in den Interviews ein; alle Befragten thematisierten diese. Dies ist zum einen dadurch zu erklären, dass die Einführung der Software von vielen Emotionen begleitet war, zum anderen dadurch, dass die individuellen Herangehensweisen und das persönliche Engagement entscheidenden Einfluss auf die Involviertheit im Softwareentwicklungsprozess und damit auch auf dessen Qualität hatten.

Der knappe Zeitplan der Softwareeinführung und der damit einhergehende (teilweise noch unfertige bzw. unzureichende) Zustand der Software sowie der zugrunde liegenden technischen Infrastruktur führten fast durchgängig zu einer erheblichen Mehrbelastung der Beschäftigten, die in den meisten Bereichen nach wie vor anhält. Der Zusatzbelastung wurde durch teilweise massiv anfallende Überstunden begegnet. Andere Aufgaben wurden häufig vernachlässigt, was wiederum Probleme in den betreffenden Arbeitsfeldern nach sich zog. Entsprechend breiten Raum nahmen die diesbezüglichen Schilderungen der Befragten ein. Die Art der Einführung, insbesondere der enge Zeitplan, stieß auf breite Ablehnung. Dabei wird das Campus-Management-System prinzipiell durchaus akzeptiert: Keiner der Befragten verneinte die grundsätzliche Notwendigkeit einer solchen Software. Die Art und Weise der Einführung jedoch erzeugte Frust, worunter die Akzeptanz der Software stark litt

³ Interessanterweise spielt die Rollenzugehörigkeit (oder auch Hierarchieebene bzw. Bereichszugehörigkeit) in den übrigen Kategorien keine Rolle: Die Befragten unterschieden sich dort kaum in ihren Schilderungen.

(vgl. die in 4.2. beschriebene Sündenbockproblematik). Die Befragten fühlten sich ausgenutzt und „verheizt“ und vermissten eine angemessene Würdigung dessen, was sie geleistet haben.

Wie oben bereits angesprochen, war das *persönliche Engagement* – das Anwenden von Pull-Strategien, s. 4.1. – der Befragten offenbar ausschlaggebend dafür, wer bzw. welche Einheiten an der Anforderungsanalyse sowie am Entwicklungsprozess wie stark beteiligt wurden. Solche Strategien waren vor allem hartnäckiges Nachfragen bzw. generell die selbst initiierte Kontaktaufnahme mit dem Entwicklerteam, das Nutzen bestehender informeller Kontakte zur Informationssuche, das (ungefragte) Weiterleiten von Fehlermeldungen oder Weiterentwicklungsideen und -wünschen sowie generell eine sehr intensive Befassung mit der Software und den zugrunde liegenden Prozessen, die die entsprechenden Nutzer zu versierten und gefragten Ansprechpartnern seitens der Entwickler machten. Eine Befragte bezeichnete dies als „Mitentwicklerbonus“, den sie durch diese intensive Beschäftigung schließlich erhalten habe und der es ihr auch ermöglichte, Sonderwünsche durchzusetzen.

Darüber hinaus finden sich weitere Indikatoren, dass persönliche *Eigenschaften, Vorlieben und Kompetenzen* Einfluss auf Teilhabe am und Gestaltung des Entwicklungsprozesses haben, wie etwa das individuelle Zeit- und Ressourcenmanagement und Problemlösestrategien.

5. Diskussion und Fazit

Der hier vorgestellte Rekontextualisierungsprozess war, wie viele Softwareentwicklungs- und Einführungsprozesse, von erheblichen Widerständen und Schwierigkeiten begleitet. Diese sind zum einen auf einen allzu knappen Zeitplan sowie Unzulänglichkeiten im Softwareentwicklungsprozess zurückzuführen, zu einem größeren Teil jedoch auf personelle Faktoren (23%), mangelnde organisatorische Einbettung (26%) sowie Kommunikationsschwierigkeiten (16%), die zusammen genommen etwa zwei Drittel der Problemschilderungen ausmachen. Technische Probleme und mangelnde Gebrauchstauglichkeit machen demgegenüber einen deutlich geringeren Teil aus (10%; dies entspricht etwa der Hälfte der im Faktor „Technik“ verorteten Probleme). In den folgenden Abschnitten werden Implikationen aus der Fallstudie für Forschung und Praxis abgeleitet.

5.1. Implikationen für die Praxis

Das Fallbeispiel hebt erneut die Bedeutung von Organisationsentwicklungsmaßnahmen in Verbindung mit Softwareentwicklungsprozessen hervor – ein erheblicher Teil der aufgetreten Probleme lässt sich auf einen Mangel in diesem Bereich zurückführen, deutlich mehr als auf technische Probleme. Die Verteilung finanzieller und sonstiger Ressourcen in Softwareentwicklungsprojekten ist jedoch typischerweise genau umgekehrt. Zudem wurde deutlich, wie wichtig es für eine erfolgreiche Einführung ist, heterogene Nutzer(gruppen) einzubinden und die Hürden für eine Beteiligung möglichst gering zu halten: Pull-Strategien und -Maßnahmen bei Softwareentwicklung und Support, die eine starke Eigeninitiative der Nutzer erfordern, sollten im Vergleich zu Push-Maßnahmen, bei denen die Entwickler initiativ auf die Nutzer zugehen, im Hintergrund stehen.

Folgende Ansatzpunkte lassen sich demnach anhand der Untersuchungsergebnisse identifizieren, um den fortschreitenden CM-Einbettungsprozess zu verbessern:

- *Stärkung der Partizipation*: Wie im Hinblick auf den gesamten Prozess deutlich wurde, war und ist die Beteiligung am Softwareentwicklungsprozess maßgeblich von der Eigeninitiative der Beteiligten abhängig. Hier sollten verstärkt Maßnahmen ergriffen werden, um eine breitere und weniger selektive Beteiligung zu erreichen – auch in der Hoffnung und Erwartung, dass sich eine solche Partizipation wiederum positiv auf die Beteiligung und das Engage-

- ment auswirkt. Arbeiten zu *verteilter Softwareentwicklung* (Distributed Participatory Design, vgl. [4]) können hier ggf. konkrete Hinweise zur Ausgestaltung des Prozesses geben.
- *Fokus auf Organisationsentwicklung*: Während der *technische* Support, der hier aus Platzgründen nicht näher thematisiert wurde, von den Befragten durchaus positiv beurteilt wird, findet sich kein Pendant im Bereich der *Organisationsentwicklung*. Die hier entstehenden Aufgaben mussten von den jeweiligen Fakultäten/Einrichtungen größtenteils allein bewältigt werden. Die Schaffung unterstützender Strukturen sowie die Aufstockung entsprechender Ressourcen scheinen unumgänglich.
 - *Verbesserung der Kommunikation*. Ebenfalls nur gestreift wurde hier die Rolle der Kommunikation, die, wie aus der Problemverteilung ersichtlich ist, ebenfalls recht problematisch beurteilt wurde, da Informationen häufig nicht bei den Adressaten ankamen und sich der Einführungsprozess dadurch weiter verzögerte bzw. Entscheidungen auf der Basis unzureichender Informationen getroffen wurden. Im Bereich der Kommunikation, so scheint es, könnte mit vergleichsweise wenig Aufwand viel positive Veränderung erreicht werden. Dies betrifft zum einen effizientere Kommunikationswege, zum anderen eine ehrlichere, transparentere Informationspolitik, die zur Akzeptanz beitragen könnte.
 - *Formalisierungslücken*: Im Hinblick auf die Softwareentwicklung selber entsteht der Eindruck – in Verbindung mit einem noch immer immensen Weiterentwicklungspensum – dass eine *Reduzierung der Komplexität* sinnvoll wäre. Gerade angesichts teilweise noch immer sehr unterschiedlicher Strukturen und Prozesse und daraus resultierender Sonderwünsche und Anforderungen stellt sich die Frage, ob tatsächlich jeder Aspekt des Studienbetriebs durch das CM-System abgedeckt werden muss und sollte. Ggf. könnte eine flexiblere Software, die nicht jeden Einzelfall formalisiert und abdeckt, sondern den Akteuren Entscheidungsspielräume lässt, eine angemessenere und vor allem auch effizientere Lösung bieten.

Wenngleich die Ergebnisse der vorliegenden Untersuchung in erster Linie den Prozess und die Situation an der betreffenden Universität widerspiegeln, so kann doch vermutet werden, dass ähnliche Phänomene auch an anderen Hochschulen eine Rolle spielen, insbesondere was die Rolle personeller Faktoren sowie die Bedeutung begleitender Organisationsentwicklungsmaßnahmen angeht. Hochschulen, die erst am Beginn eines ähnlichen Prozesses stehen, können dementsprechend von der Erfahrungen und Ergebnissen profitieren.

5.2. Implikationen für die Forschung

Während die Bedeutung der organisatorischen Prozesse für eine erfolgreiche Softwareeinführung und die Notwendigkeit einer entsprechenden Begleitung organisationaler Veränderungsprozesse im Sinne einer *integrierten Organisations- und Technikgestaltung* schon lange bekannt sind [z.B. 9, 14], wird der Faktor „Mensch“ – also die individuellen Besonderheiten und Herangehensweisen der beteiligten Personen – bislang nur wenig beachtet. Dies gilt auch für die Forschung zu *Technologieaneignungsprozessen*, die, wenn überhaupt, vor allem so genannte Mediatoren – wie beispielsweise Supportmitarbeiter – in den Blick nimmt [z.B. 1, 10]. In der vorliegenden Untersuchung stellten sich personelle Faktoren jedoch als gewichtige Einflussfaktoren dar, auch im Hinblick auf die Softwareentwicklung: Offenbar waren das persönliche Auftreten, die jeweiligen (informellen) Beziehungen sowie die Eigeninitiative und die Bereitschaft, auch ungefragt Kontakt zum Entwicklungsteam aufzunehmen und Erfahrungen und Wünsche mitzuteilen, ausschlaggebend für den Grad der Partizipation.

In der Forschung zu Technologieaneignungs- und Softwareentwicklungsprozessen sollten die individuellen Eigenschaften der beteiligten Nutzer stärker Beachtung finden. Eine weitere, detaillierte Analyse des vorliegenden Datenmaterials wird hier möglicherweise weitere Ansatzpunkte liefern,

z.B. im Hinblick auf Persönlichkeitsmerkmale sowie individuelle Kompetenzen. Die Perspektive der Softwareeinführung als *Rekontextualisierung* kann dazu beitragen, den Blick von einer zu starken Technikfokussierung hin zu den organisationalen und personellen Faktoren zu leiten, die weniger während der Entwicklungs-, aber dafür stark in der Einführungsphase zum Tragen kommen. Einige Hinweise, wie die Rekontextualisierung bereits während der Softwareentwicklung – also der *Dekontextualisierung* – Beachtung finden kann, werden beispielsweise bei Gumm & Janneck [4] thematisiert und sollten Gegenstand weiterer Forschung sein.

6. Literatur

- [1] BANSLER, J.P. & HAVEN, E., Technology-Use Mediation: Making Sense of Electronic Communication in an Organizational Context, in: M. Pendergast, K. Schmidt, C. Simone & M. Tremaine (eds.), Proceedings of the 2003 International Conference on Supporting Group Work (pp. 135-143). New York 2003.
- [2] CRUTZEN, C., Dekonstruktion, Konstruktion und Inspiration, in: Fiff Kommunikation 3/2001, 47-52.
- [3] FINCK, M. & JANNECK, M., Das Unvorhersehbare steuern? Zum Umgang mit der komplexen Dynamik in Technologieaneignungsprozessen, in: D. Gumm, M. Janneck, R. Langer & E. Simon (Hrsg.), Mensch, Technik, Ärger? Zur Beherrschbarkeit soziotechnischer Dynamik aus transdisziplinärer Sicht (S. 85-102). Hamburg 2008.
- [4] GUMM, D. & JANNECK, M., Requirements Engineering for Software Recontextualization, in: T. Tiainen, H. Isomäki, M. Korpela, A. Mursu, M.-K. Paakki & S. Pekkola (Eds.), Proceedings of 30th Information Systems Research Seminar in Scandinavia, IRIS30, University of Tampere, Finland 2007.
- [5] KRAUSE, D., ROLF, A., CHRIST, M. & SIMON, E., Wissen, wie alles zusammenhängt, in: Informatik-Spektrum, 29 (4), S. 263-273 (2006).
- [6] MAYRING, P., Qualitative Inhaltsanalyse: Grundlagen und Techniken, 8. Auflage. Weinheim 2003.
- [7] OBERQUELLE, H. (Hrsg.), Kooperative Arbeit und Computerunterstützung. Stand und Perspektiven. Göttingen 1991.
- [8] ORLIKOWSKI, W. J., Evolving the notes: organizational change around groupware technology, in: C. U. Ciborra (ed): Groupware and Teamwork: invisible Aid Or Technical Hindrance (pp. 23-59). Wiley Series In Information Systems. New York 1996.
- [9] ORLIKOWSKI, W.J., YATES, J., OKAMURA, K. & FUJIMOTO, M., Shaping Electronic Communication: The Metastructuring of Technology in the Context of Use, in: Organization Science, 6, (4), 423-444 (1995).
- [10] ROLF, A., Mikropolis 2010. Marburg 2008.
- [11] SESINK, W., Wozu Informatik? Ein Antwortversuch aus pädagogischer Sicht. In F. Nake et al. (Hrsg.), Informatik zwischen Konstruktion und Verwertung (S. 59-62). Uni Bremen, FB Mathematik & Informatik, Bericht 1/04 (2003).
- [12] WULF, V. & ROHDE, M., Towards an Integrated Organization and Technology Development, in: Proceedings of the ACM Symposium on Designing Interactive Systems (pp. 55-64). New York 1995.