

# Software Engineering und Projektmanagement 2.0 VO



## SEPM

### Fallbeispiele 2: HISS

**WS2016/17 – 03.11.2016**

[www.inso.tuwien.ac.at](http://www.inso.tuwien.ac.at)



**INSO - Industrial Software**

Institut für Rechnergestützte Automation | Fakultät für Informatik | Technische Universität Wien

- 1 HISS – ein Fallbeispiel eines mehrjährigen Softwareprojekts
- 2 Zielsetzung und Projektscope
- 3 Projektstrukturen
- 4 Identifikation von Anforderungen
- 5 Modularisierung – wie bleibt das große Ganze überschaubar
- 6 Erkenntnisse

# Ausgangssituation

- 40 Jahre Universitäts- und Gesetzesgeschichte, abgebildet in bis zu 40 Jahre alten, inhomogenen und stark interdependenten Systemen
- Altes „Flickwerk“
- „Schnittstellen-Chaos“
- Mangelnde Systemische Trennung von Datenhaltung, Anwendungslogik und Präsentationsebene
- „Zu Tode gewartet“
- Nicht zukunftsfähig



| Kennzahl                          | Wert                                       |
|-----------------------------------|--|
| Anzahl Mitarbeiter                | > 4.600, davon ca. 3.300 wissenschaftliche |
| Anzahl Studierende                | > 29.000                                   |
| Anzahl der Fakultäten             | 8  |
| Anzahl der Institute              | 52   |
| Anzahl Dienstleister              | 20   |
| Anzahl der eingerichteten Studien | > 50                                       |
| Anzahl der Studienabschlüsse      | > 2.600 jährlich                           |

## Zielsetzung

- Einheitliche technische Architektur
- Homogene und wachstumsfähige Systembasis
- Schrittweise Ablöse der Altsysteme
- Langfristige Sicherstellung der adaptiven und perfektiven Wartbarkeit
- Flexible Plattform für neue Services
- Qualifizierter Docking-Partner für externe und interne Nachbarsysteme

## Ausgangssituation



## Zielbild



# Featureliste (Initialplanung)

## Identifizierte Legacy Features

(wenig und meist veraltete Dokumentation)

|                      |     |
|----------------------|-----|
| Lehre und Verwaltung | 202 |
| Adressbuch           | 11  |
| Forschungsprojekte   | 61  |
| Publikationen        | 31  |

Summe: 305

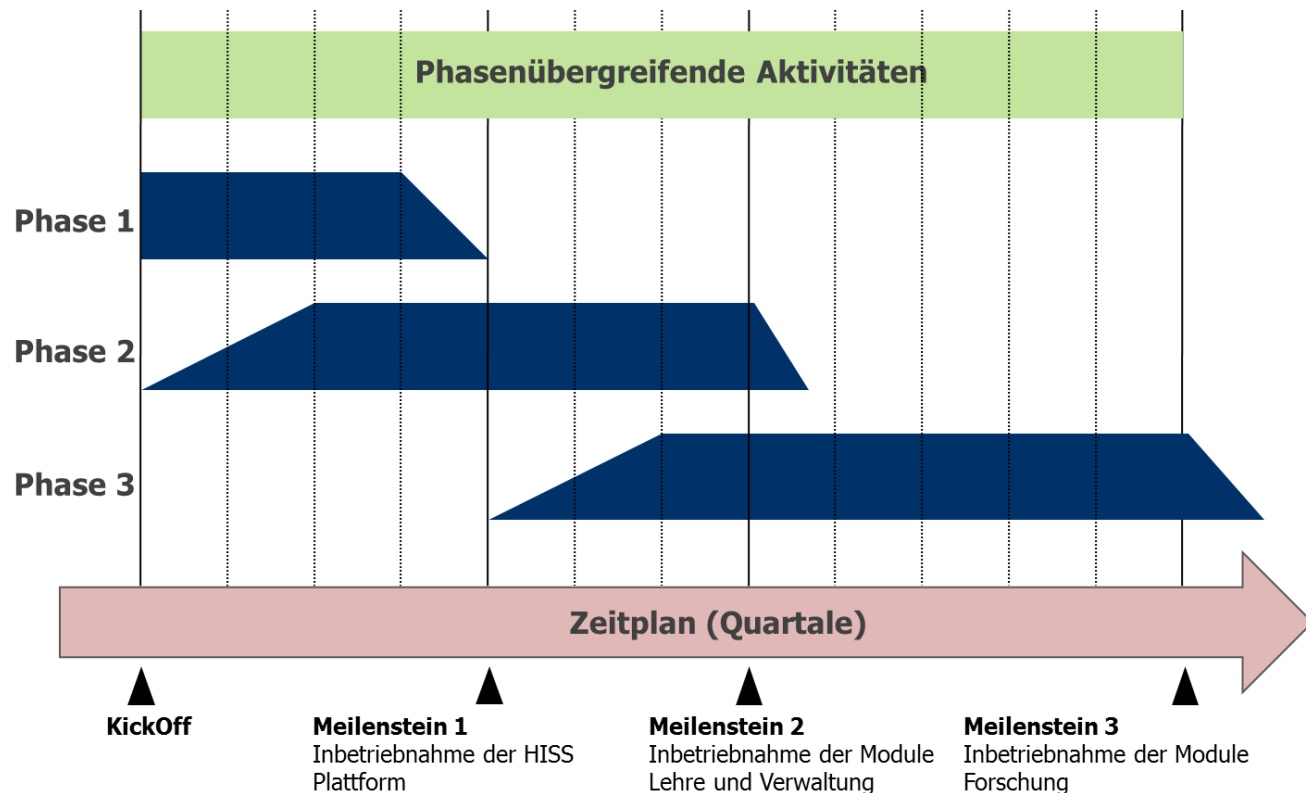
## Geplante HISS Features

|   |    |
|---|----|
| Organisation, Personal, Funktionen/Rechte   | 35 |
| Personalservices                            | 49 |
| Liegenschaften (Gebäude, Räume)             | 3  |
| Studierendenverwaltung, Studienpläne        | 25 |
| Lehrveranstaltungen                         | 58 |
| Prüfungen, Anerkennungen, Zeugnisse         | 17 |
| Integration/Schnittstelle e-Learning        | 1  |
| ZID Services (ID-Management, Mailing, etc.) | 20 |
| Projektdatenbank, Vertragsdatenbank         | 45 |
| Publikationsdatenbank, Aleph Integration    | 23 |
| Reporting und Statistiken                   | 30 |
| Technische Features                         | 23 |

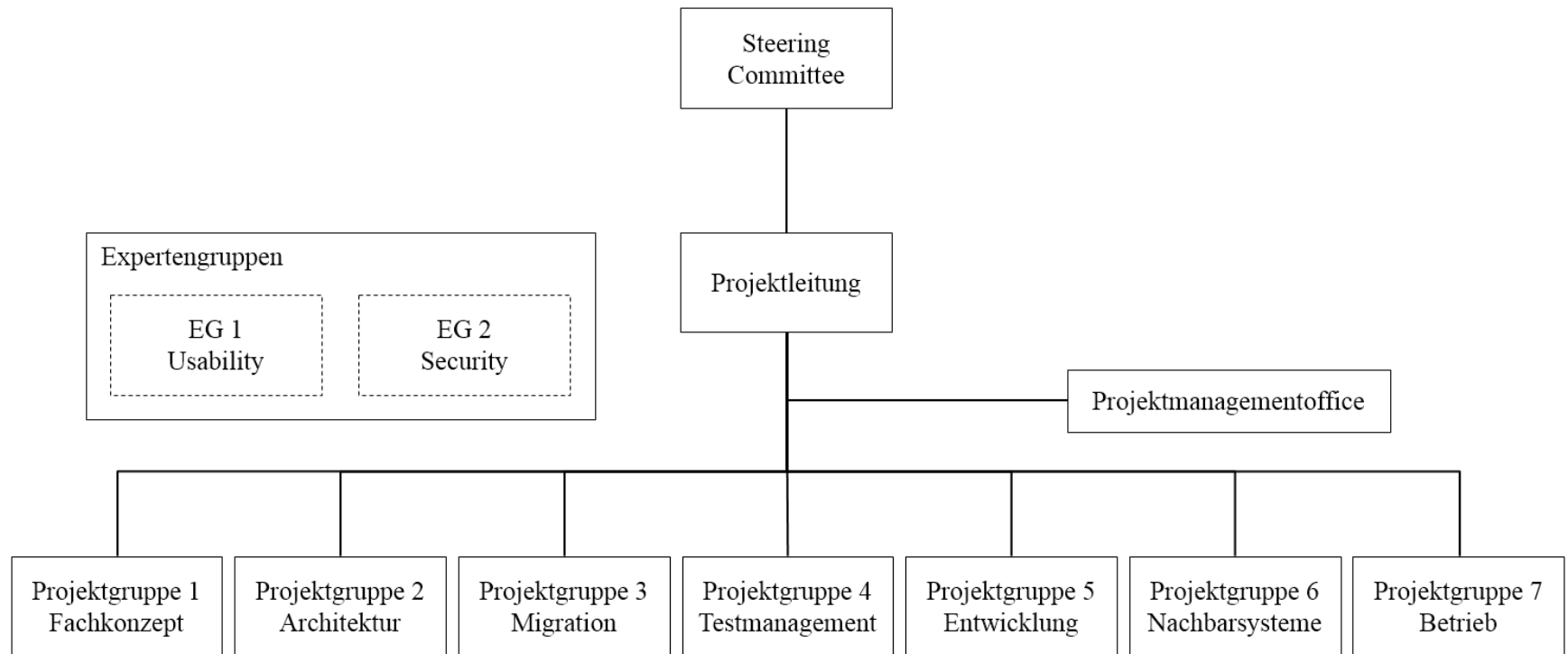
329

## Geplante Projektphasen zur Umsetzung des initial erhobenen Feature-Sets

- Geplante Gesamtdauer von 3 Jahren,
- Erster GoLive geplant nach 1 Jahr



# Projektstruktur





## Steering Committee

- Steuerungsgremium
- Vertreter aus Rektorat, Betriebsrat, Professorenschaft und Hochschülerschaft
- tagt vierteljährlich

## Expertengruppen

- Security und Usability Engineering als permanente Begleitprozesse

## Projektteam

- Zwischen 10 und 32 Vollzeitmitarbeiter
- Wöchentliche Jour fixes

## Iteratives Vorgehen

## Stakeholder

- Alle am Projekt beteiligten Personen, die ein Interesse an dem jeweiligen Softwareprojekt haben
- Wichtige Anforderungsquelle
- Beeinflussen ein Softwareprojekt und werden selbst wiederum vom organisatorischen und politischen Umfeld beeinflusst

| Stakeholder (Auszug)                             | Studierende | Vortragende | Studien-<br>dekane | Studien-<br>abteilung |
|--|-------------|-------------|--------------------|-----------------------|
| Teilbereiche (Auszug)                            |             |             |                    |                       |
| Lehrveranstaltungsdaten,<br>Teilnehmerverwaltung | x           | x           | x                  |                       |
| Prüfungen und Zeugnisse                          | x           | x           | x                  | x                     |
| Termin-, Raumverwaltung, Kalender                | x           |             |                    |                       |
| Lehrunterlagen, Abgaben                          | x           | x           |                    |                       |
| Lehrveranstaltungsevaluierung                    | x           | x           | x                  |                       |
| Studienplan, -fortschritt                        | x           |             | x                  | x                     |
| Nachbarsysteme                                   | x           | x           |                    | x                     |

## Ein kleines Rechenbeispiel:

Änderungen am System erfolgen auf Basis der „Vorgehensweise Zuruf“. Nutzer sprechen direkt mit den Entwicklern und vereinbaren (meist mündlich) Änderungen.

Der Entwickler implementiert den Wunsch des Nutzers.

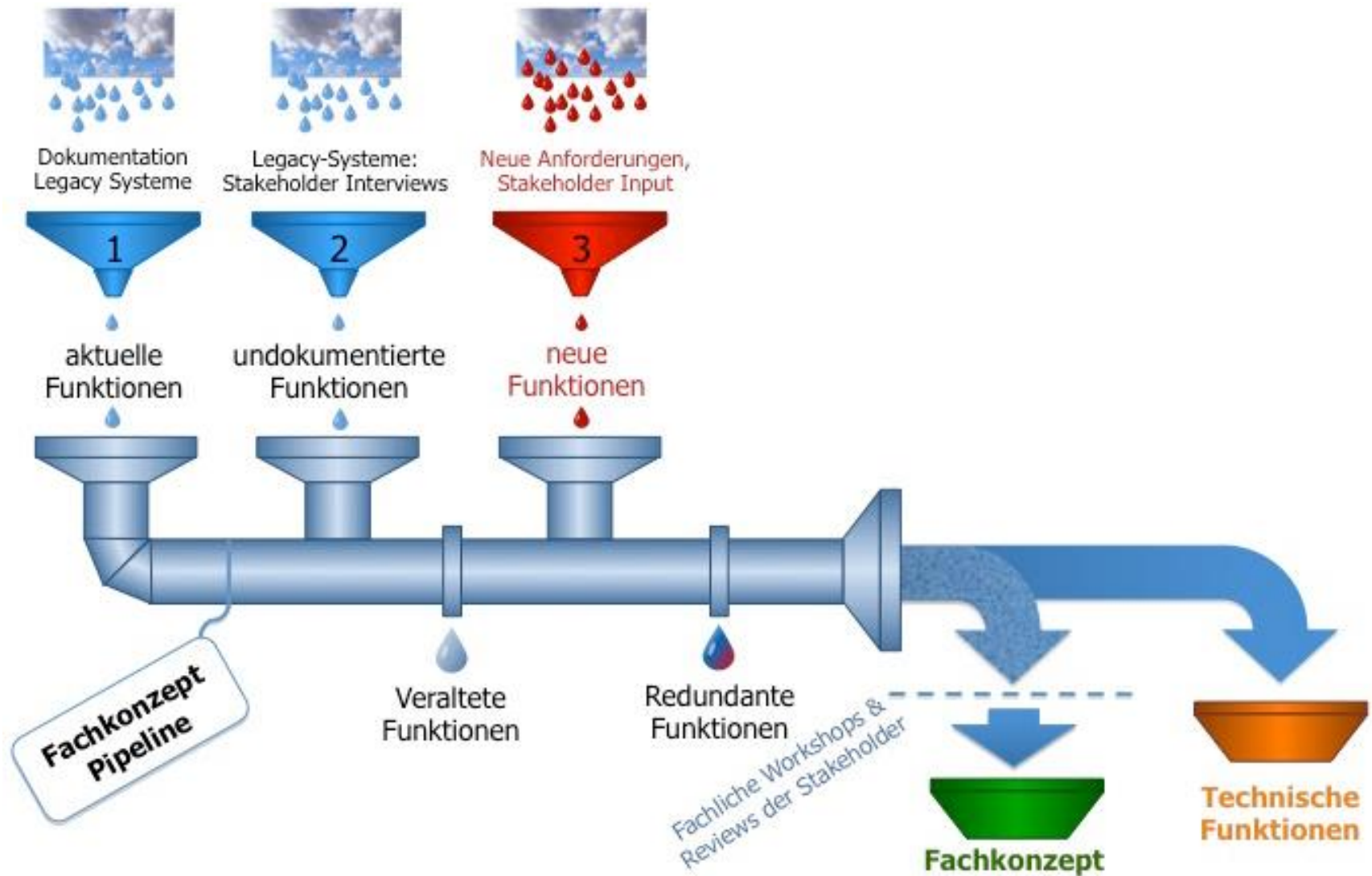
Über ein Jahr verteilt implementiert der Entwickler 25 bis 30 Wünsche von drei Anwendern. Vier weitere Entwickler tun Ähnliches.

Das ergibt pro Jahr ca. 100 Änderungen. Über mehrere Jahre.

Außer den Änderungen im Code gibt es kaum Dokumentation über diese Vorgänge.

***Sind diese undokumentierten Änderungen relevant für eine Ablöse?***

# Fachliche Analyse



## > 50 Workshops 2008-2009 zu einer sehr breiten Themenpalette

- Einbindung von Vertretern aller Nutzergruppen
- Erhaltung und Wiedergewinnung der formalen und informellen Ablaufinformationen

## Die faktischen Ergebnisse der Workshops

- Unterschiedlich gelebte Prozesse, selbst innerhalb einer Fachabteilung
  - Vielfach sehr unterschiedliche Vorstellungen bzgl. des SOLL Zustands
- ➔ Abwarten aller (mehrerer tausend) Entscheidungen im Abstrakten wäre enorm zeit- und kostenintensiv gewesen
- ➔ Bereitstellung der Systeminfrastruktur für eine iterative Entwicklung zum Realziel

# Herausforderungen Altdatenbestand

- Veraltete oder fehlende Dokumentation
  - 77% (346) der Tabellen und 84% (5616) der Spalten des abzulösenden Systems nicht dokumentiert
- Fehlende Normalisierung
  - Datenmodell spiegelte dateibasierte Datenhaltung (aus den 60er und 70er Jahren) wider
  - Keinerlei Fremdschlüssel-Definitionen
- Intransparentes Datenmodell
  - Legacy Tabellen im Produktivsystem, nicht erkennbar auf Basis der Bezeichnung
  - Tabellen mit unterschiedlichstem und nicht augenscheinlich erkennbarem Verwendungszweck (z.B. temporäre Tabellen für Batch-Prozesse)
- Unklare Datenhoheit
  - „Zerstreuung“ der Daten über mehrere Systeme und Datenbanken
  - Diffuse organisatorische Datenverantwortlichkeiten

**Systeme, die auf korrupten Daten aufsetzen, erzeugen „Information“ aus falschen Daten und sind damit unbrauchbar**

- Ausführliche Analyse des Altdatenbestands als Voraussetzung zur Identifikation der zu migrierenden Daten
  - 1) Analyse des IST Zustands
  - 2) Kategorisierung der Datenobjekte
  - 3) Zugriffs- und Traffic Analyse
- Abgleich aller Attribute aller zu migrierenden Datenobjekte gegen das Soll-Datenmodell
- Kompensation fehlender und fehlerhafter Attribute durch Entwicklung geeigneter Algorithmen zur Vervollständigung der inkonsistenten Datensätze
- Integritätsprüfungen zur Sicherstellung der Datenintegrität

} Zentrale und umfassende (Re-)Dokumentation des Altdatenbestands

# Re-Dokumentation von Datenbanktabellen und -spalten

- Verwendungszweck der Tabelle
- Spaltenname
- Datentyp
- Nullwerte
- Bedeutung der Spalte
- Indexe

Tabelle PR

alle Prüfungen seit 1968, Verrechnungssätze sind auf ORACLE.PR

Tabellendefinition

| Name    | Typ          | nullable | Bemerkung                                      |
|---------|--------------|----------|--|
| UNIV    | varchar2(1)  | N        | Universitätscode                               |
| MNR     | number(22)   | N        | Matrikelnummer                                 |
| LFD     | number(22)   | Y        | Laufnummer der Datensätze (s. 1.)              |
| PRART   | varchar2(1)  | Y        | Art der Prüfung (s. 2.)                        |
| LV      | varchar2(8)  | N        | Lehrveranstaltungsnummer oder Studienplanpunkt |
| DATUM   | number(22)   | N        | Prüfungsdatum (JJJJMMTT)                       |
| ART     | varchar2(1)  | Y        | Prüfungsart (s. 3.)                            |
| KNR1    | varchar2(3)  | N        | Studienkennzahl 1                              |
| KNR2    | varchar2(3)  | Y        | Studienkennzahl 1                              |
| MODUS   | varchar2(1)  | Y        | Prüfungsmodus (s. 4.)                          |
| VERR    | varchar2(1)  | Y        | Verrechnungskennzeichen                        |
| NOTE    | varchar2(1)  | Y        | Note (s. 5.)                                   |
| FRIST   | number(22)   | Y        | Reprobationsfrist (s. 6.)                      |
| SEM     | varchar2(5)  | Y        | Prüfungsemester (JJJS)                         |
| PRUEFER | varchar2(10) | Y        | Personalnummer des Prüfers                     |
| ZEUGNIS | number(22)   | Y        | Datum Zeugnis gedruckt (JJJJMMTT - geplant!)   |

Indexes

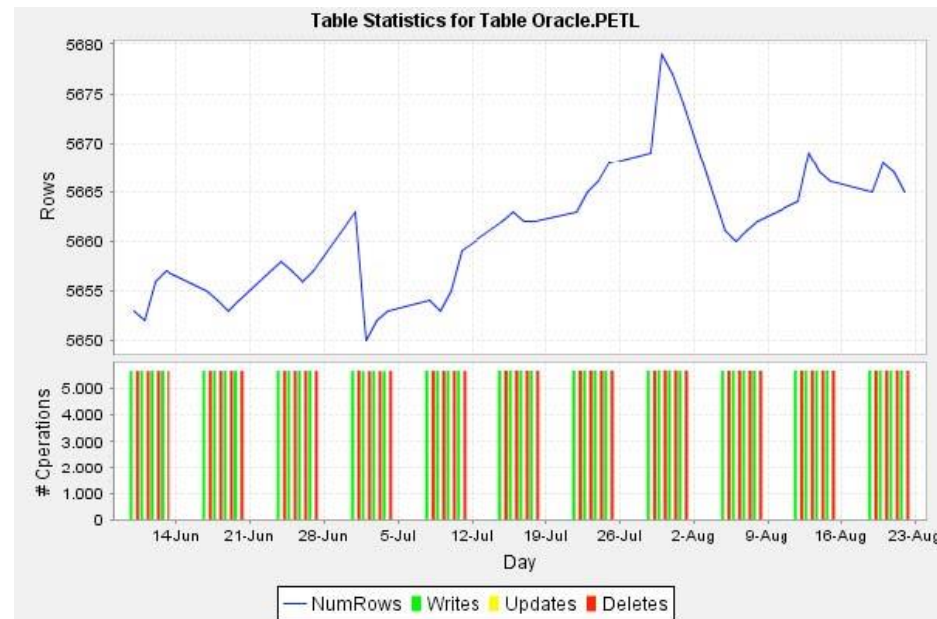
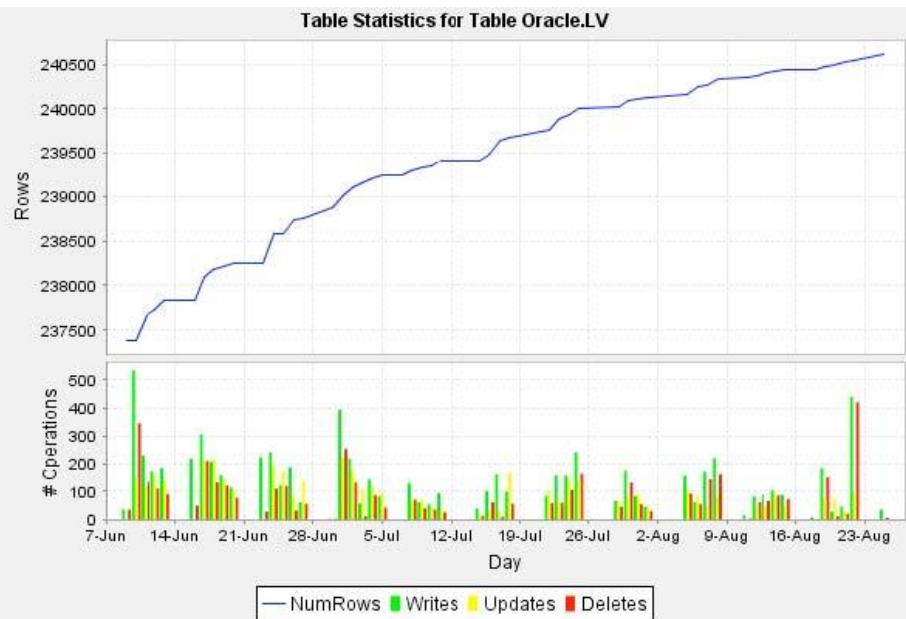
PRIND [UNIQUE]

|      |
|------|
| Name |
| MNR  |
| LFD  |



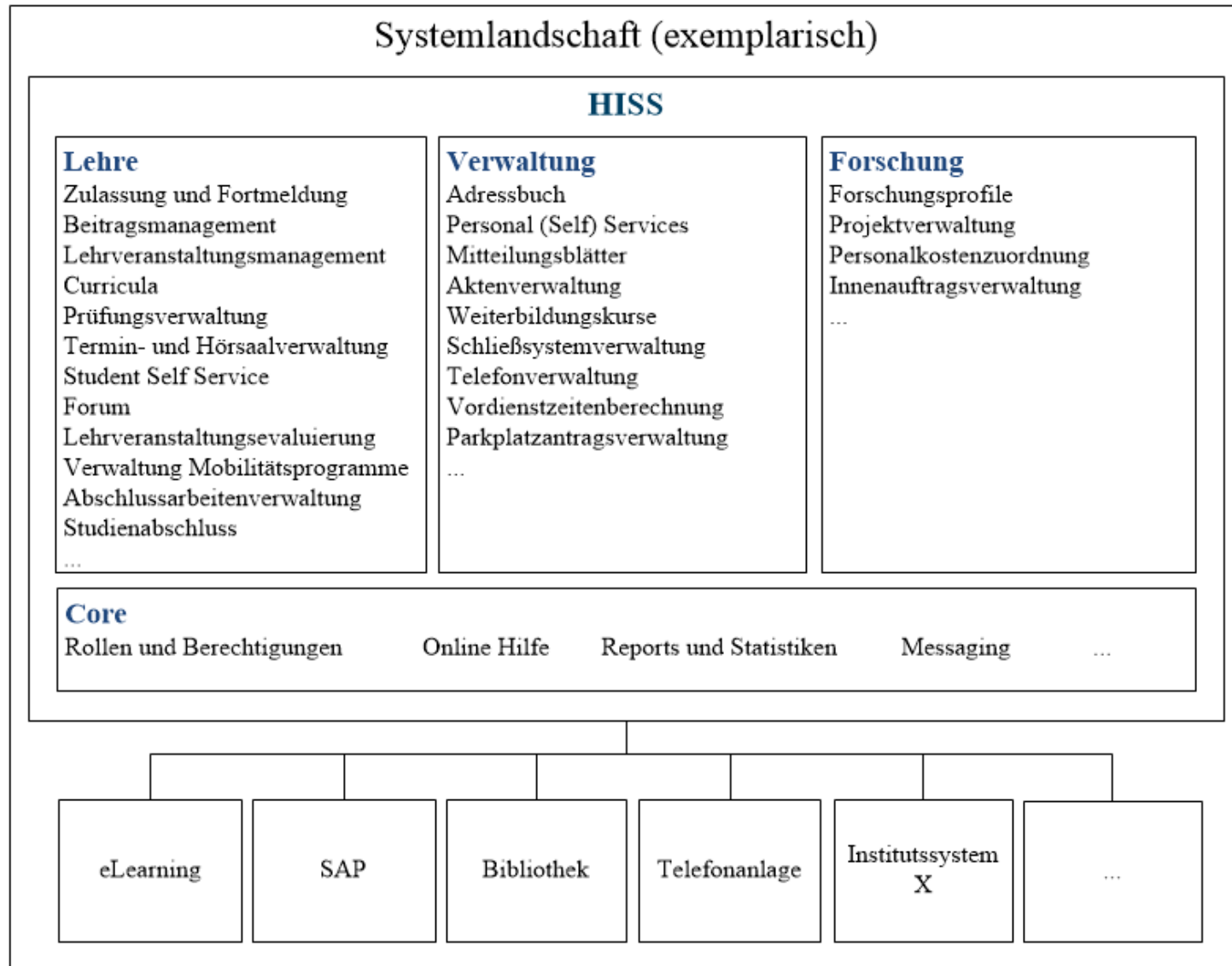
# Zugriffs- und Traffic Analyse

- für sämtliche Tabellen wurden statistische Zugriffsdaten erhoben
- Tabellarische und grafische Auswertungen



| numrows | writes | updates | deletes | lastwrite        |
|---------|--------|---------|---------|------------------|
| 4037991 | 32653  | 0       | 31678   | 15.07.2008 09:55 |
| 4037991 | 58170  | 0       | 56495   | 15.07.2008 09:55 |
| 4037991 | 75962  | 0       | 73901   | 15.07.2008 09:55 |

# Modularisierung



- Fachliche Bereiche bleiben strikt getrennt
- Klare Datenhoheit bei überschaubaren Modulgrößen
  - Jedes Modul hält seine Daten in einer eigenen Datenbank
  - Fachlich zusammenhängende, aus wenigen Tabellen bestehende Datenbanken
- Unabhängige (Weiter-)Entwicklung einzelner Module
  - Unterstützt eine schrittweise Einführung
- Module sind einzeln lauffähig und compilierbar
  - Keine Beeinträchtigung des Gesamtsystems während Entwicklungs- und Testphasen
  - Erlaubt das Austauschen einzelner Module ohne das Gesamtsystem austauschen zu müssen

## Ablöse eines zeit- und kostenintensiven, manuellen Prozesses

- Selbstdruck von Zeugnissen und Bestätigungen für Studierende, z.B.
  - Einzelzeugnisse
  - Sammelzeugnisse
  - Studienblätter
  - Studienbestätigungen
  - Erfolgsbestätigungen
  - Bestätigung gem. Familienlastenausgleichsgesetz
- Entlastung der Studien- und Prüfungsabteilung um ~300.000 Dokumente pro Jahr
- Postversand entfällt
- Online Validierung mittels Barcode und URL durch Dritte möglich
- Zugriffsmöglichkeiten für Dritte individuell und pro Dokument konfigurierbar

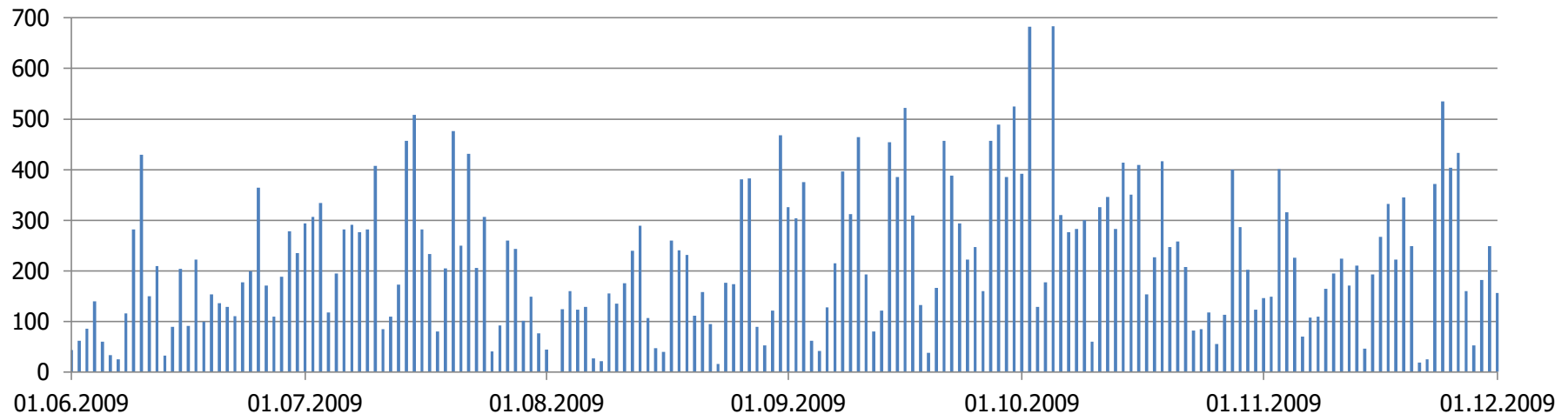
## Juni – November 2008:

- 60.000 Poststücke
- 10.000 Studierende in der Studien- und Prüfungsabteilung

## Juni – November 2009:

- 0 Poststücke
- 62.000 Downloads

### Download Einzelzeugnisse Juni – November 2009



## Releases

- Inbetriebnahme erster Funktionalitäten initial geplant 1 Jahr nach Projektstart, tatsächlich 4 Monate früher erfolgt (Ruf nach Quick Wins und schnelle Sichtbarkeit von ersten Projektergebnissen)
- Iterative Ablöse bzw. Inbetriebnahme von (teils gänzlich neuen) Teilfunktionalitäten über 4 Jahre
  - ➔ initial identifiziertes Feature-Set von 329 Features wuchs innerhalb der ersten 2 Jahre Projektlaufzeit auf > 800!
- Wöchentlich (Zwischen-)Releases ab Inbetriebnahme eines Moduls

- „Entwicklung auf Zuruf“ muss unterbunden werden
- Derartige Migrationsprojekte sind auch Organisationsprojekte
- Der Mut, wichtige Entscheidungen rechtzeitig zu treffen, verhindert vorhersehbare Projektverzögerungen und Budgetüberschreitungen
- Bei langlaufenden Projekten ändern sich (meist)
  - Prioritäten
  - Personen
  - Entscheidungsträger
  - Vorgehen
  - Technologien
  - Anforderungen

## Die Einführung eines Campus Management Systems ist begleitet von erheblichen Herausforderungen

- **Planerisch**

„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.“ [1]

- **Technisch**

Herausforderungen bei der Migration von Daten, wenn sich die Systematik und Umfang der Datenhaltung des Altsystems stark von der des neuen Systems unterscheiden oder die Qualität der historischen Daten unzureichend ist [2]

- **Faktor Organisation**

Überforderung der Akteure durch die gleichzeitige Durchführung von Organisationsreformen, Studienreformen und der Einführung eines neuen Campus Management Systems → System wird voreilig und unrechtmäßig zum „bequemen Sündenbock“ für Mängel in der Organisation [3]

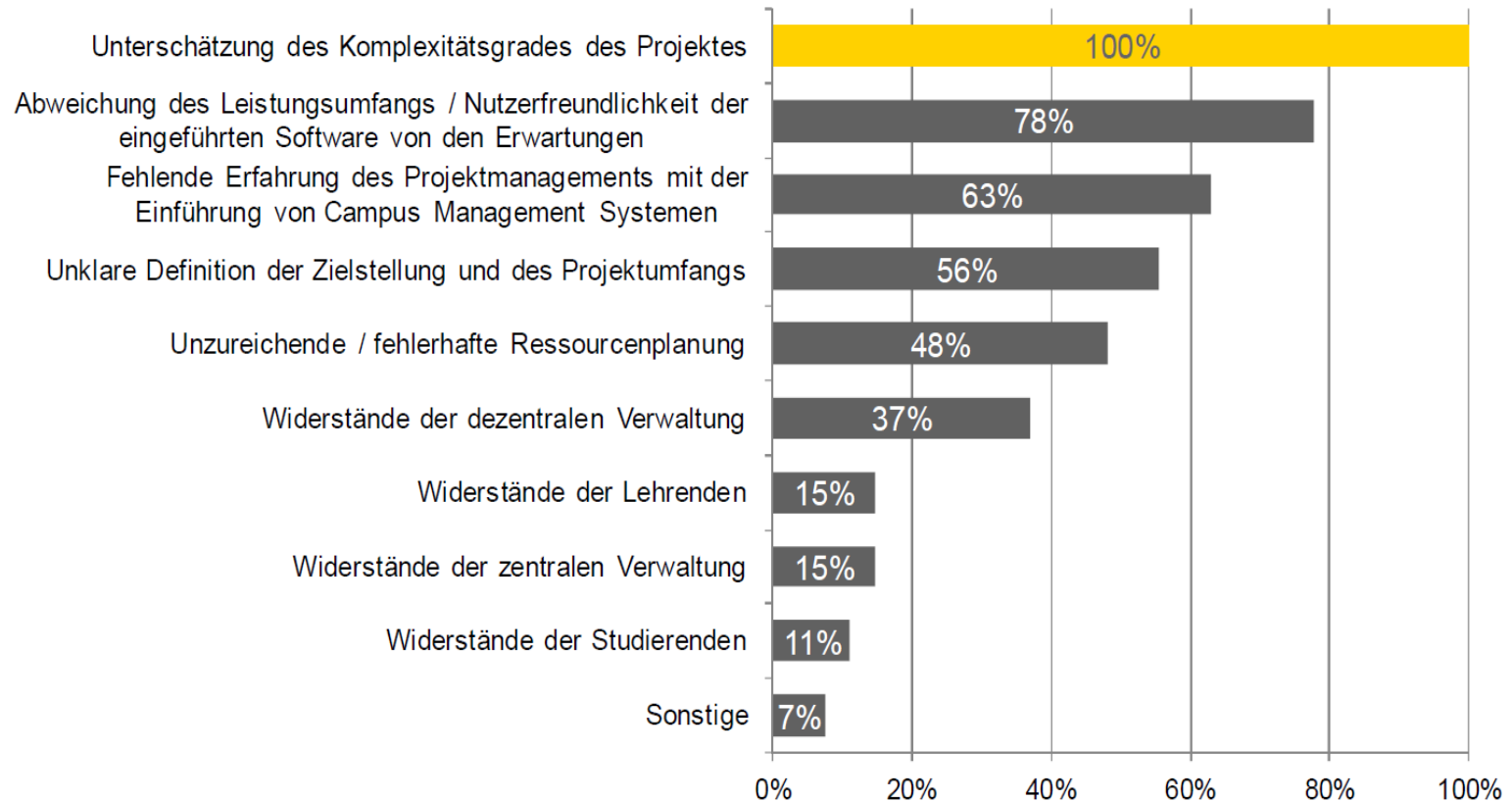
- **Fachlich und Personell**

Die Ergebnisqualität und die Akzeptanz des Systems hängen sehr stark von der Beteiligung am Entwicklungsprozess sowie der Anforderungsanalyse durch persönliche Initiative, Bereitschaft und Hartnäckigkeit der jeweiligen Personen ab [4]

Ein stabiler Kreis von entscheidungsrelevanten Beteiligten ist essentiell [5]



# Fallstricke – Erfahrungen von Universitäten



Ergebnisse einer Befragung von > 100 Hochschulen im deutschen Sprachraum, entnommen aus L. Weigel und B. Saggau, „Campus-Management zwischen Hochschulautonomie und Bologna-Reform“, *Ernst & Young Campus-Management-Studie*, Hamburg, 2012.

- [1] H. Fischer und C. Hartau, „STiNE an der Universität Hamburg zur Einführung eines integrierten Campus Management Systems“, in Wirtschaftsinformatik Proceedings 2009, 2009, S. 533–542.
- [2] G. Oevel und M. Toschläger, „Einführung eines prozessorientierten Campusmanagement an der Universität Paderborn – ein Erfahrungsbericht“, in Prozessorientierte Hochschule. Allgemeine Aspekte und Praxisbeispiele, A. Degkwitz und F. Klapper, Hrsg. Bad Honnef, 2011, S. 128 – 146.
- [3] H. Fischer und C. Hartau, „STiNE an der Universität Hamburg zur Einführung eines integrierten Campus Management Systems“, in Wirtschaftsinformatik Proceedings 2009, 2009, S. 533–542.
- [4] M. Janneck, A. Cyrill, S. Fiammingo, und R. Luka, „Von Eisbergen und Supertankern: Topologie eines Campus-Managementeinführungsprozesses“, in Wirtschaftsinformatik Proceedings 2009, 2009, S. 453 – 462.
- [5] L. Degenhardt, H. Gilch, B. Stender, und K. Wannemacher, „Campus-Management-Systeme erfolgreich einführen“, in Wirtschaftsinformatik Proceedings 2009, 2009, S. 463–472.