Heidelberg Eye Explorer
Die technologische Neuausrichtung

Erfahrungsbericht aus einem Reengineering-Projekt

Martin Moro
Lehrstuhl für Wirtschaftsinformatik III

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Universität Regensburg

6. Workshop Software Reengineering
Bad Honnef, 3.-5. Mai 2004

Vorausblick

Agenda

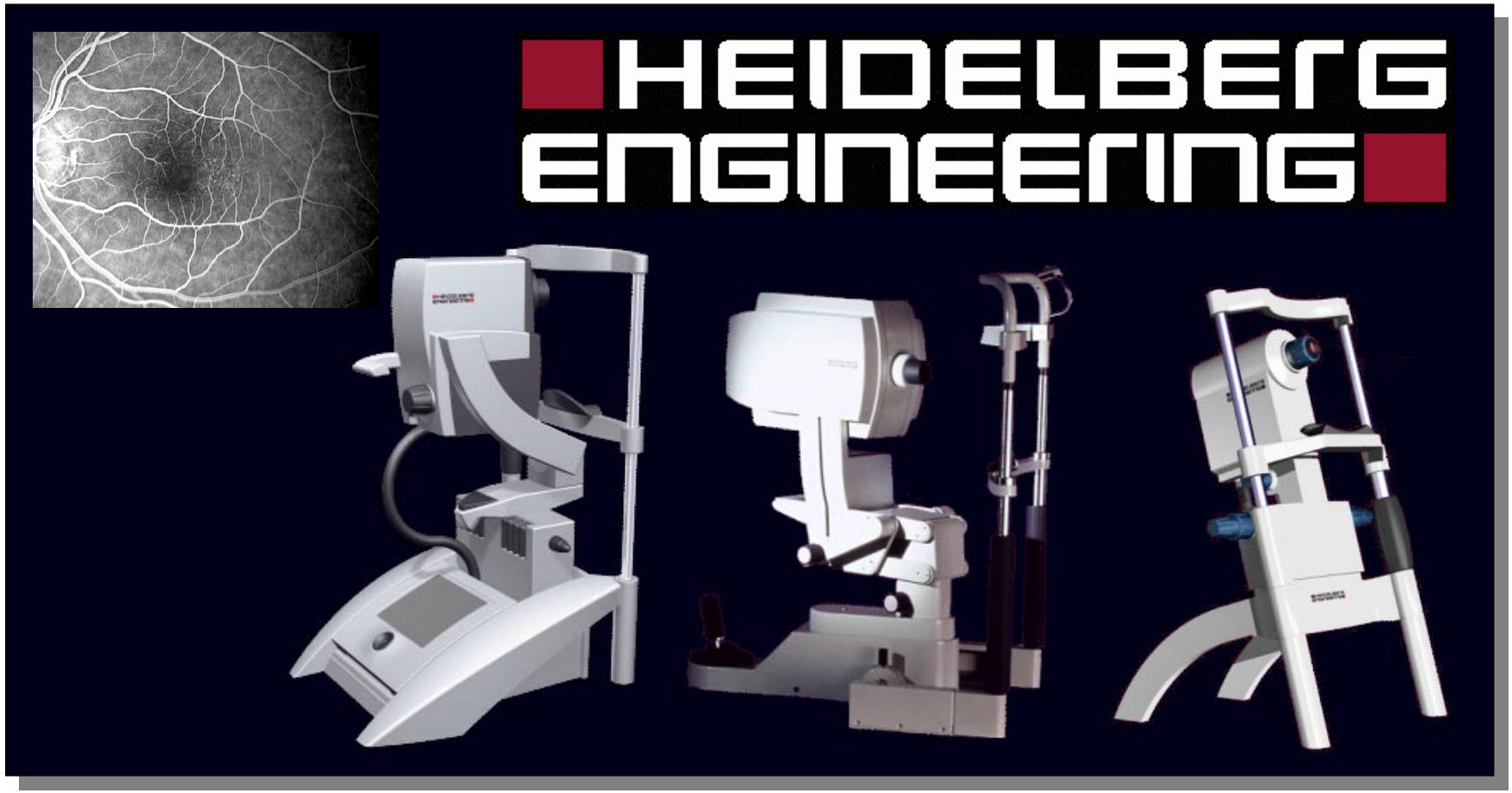
- Einleitung und Problembereich
- Der Weg zur neuen Softwarearchitektur
- Die neue Softwarearchitektur

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

1. Einleitung und Problembereich

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

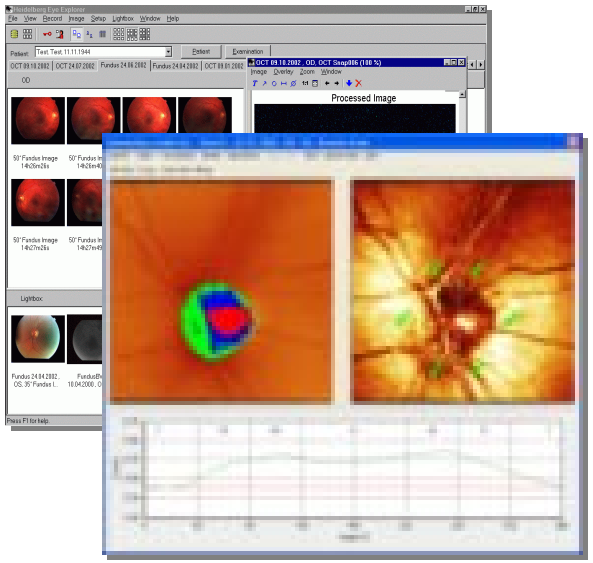
Die Software Heidelberg Eye Explorer



Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Das Einsatzszenario in Augenarztpraxen

- **Augenarztpraxen** unterhalten i.d.R. **nur ein Aufnahmegerät** sowie einen dazu gehörigen PC-Arbeitsplatz mit der Software Eye Explorer.



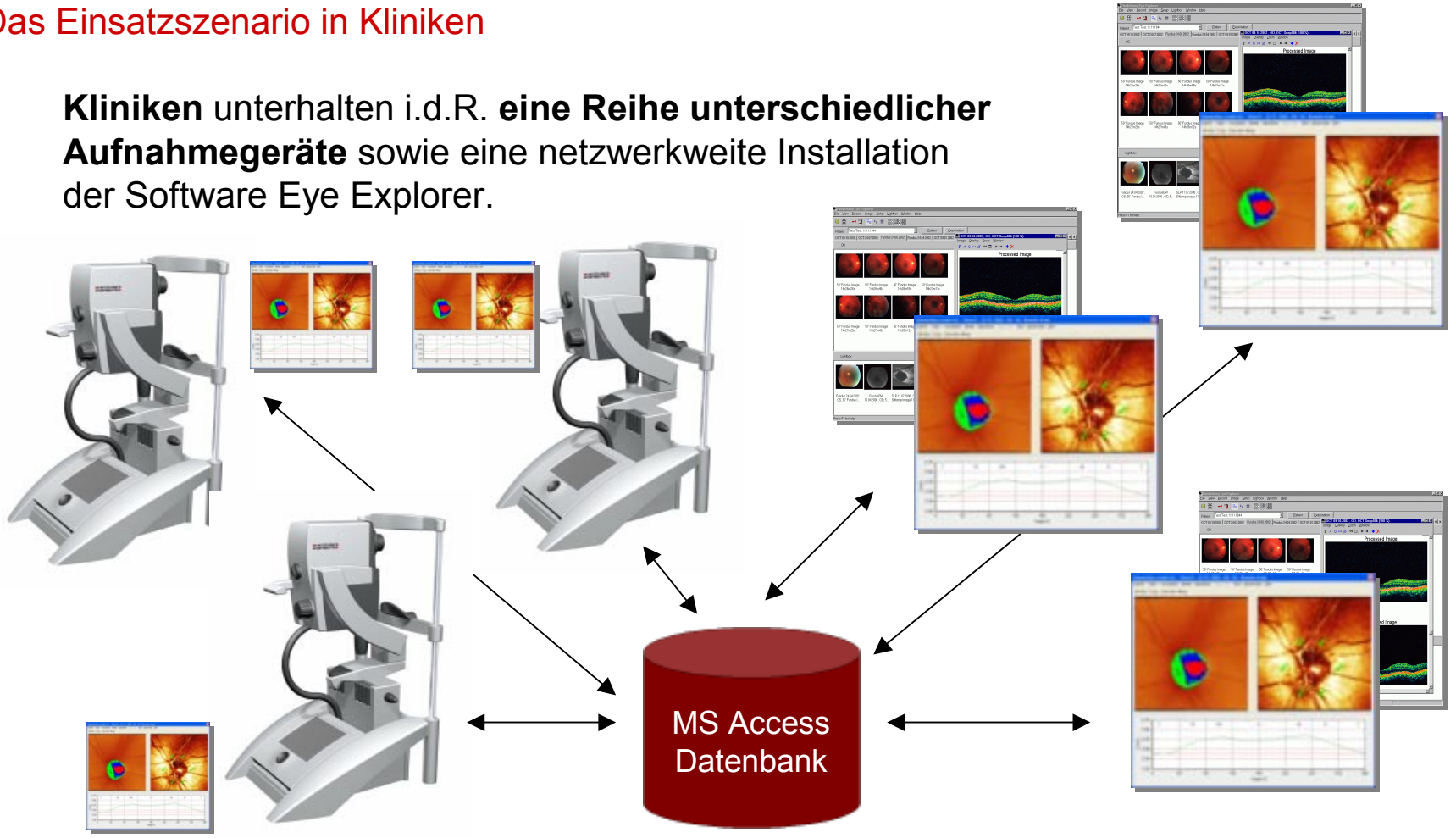
Heidelberg Eye Explorer

Heidelberg Retina Angiograph 2
(beispielhaft für Produktpalette)

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Das Einsatzszenario in Kliniken

- **Kliniken** unterhalten i.d.R. **eine Reihe unterschiedlicher Aufnahmegeräte** sowie eine netzwerkweite Installation der Software Eye Explorer.



Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Einleitung und Problembereich

Der daraus entstehende Problembereich

- Die **historisch gewachsene Software** Heidelberg Eye Explorer ist **technologisch nicht für den Einsatz in einem Kliniknetzwerk vorbereitet**.
- Die MS Access **Datenbank** ist mit dem entstehenden **Datenvolumen überfordert**.
- **Wichtige Informationen** liegen **außerhalb der Datenbank** in separaten Dateien und sind somit **nicht durchsuchbar und für den Arzt nicht erschließbar**.
- **Gleichzeitige Zugriffe** auf die Daten **konkurrieren** zueinander.
- Die Software sieht **kein Berechtigungskonzept** für einen Mehrbenutzerbetrieb vor.
- **Änderungen an den Daten** werden **nicht unmittelbar** an die anderen Arbeitsstationen **gemeldet**.

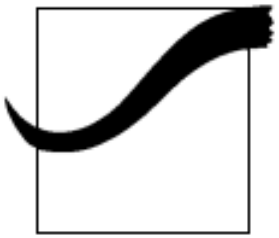
= Motivation für ein Reengineering.

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Führt zu Kooperation mit entsprechender Rollenverteilung

UNIVERSITÄT REGENSBURG

KLINIKUM



Lieferant für Ideen zukünftiger Anforderungen und Referenzinstallation im Rahmen einer Klinik.

HEIDELBERG ENGINEERING

Umsetzung der neuen Architektur in die bestehende Software Heidelberg Eye Explorer.

Universität Regensburg
Lehrstuhl für Wirtschaftsinformatik III

Zulieferung von softwaretechnischen Know How für das Architekturkonzept und der Entwicklung der kritischen Komponenten.

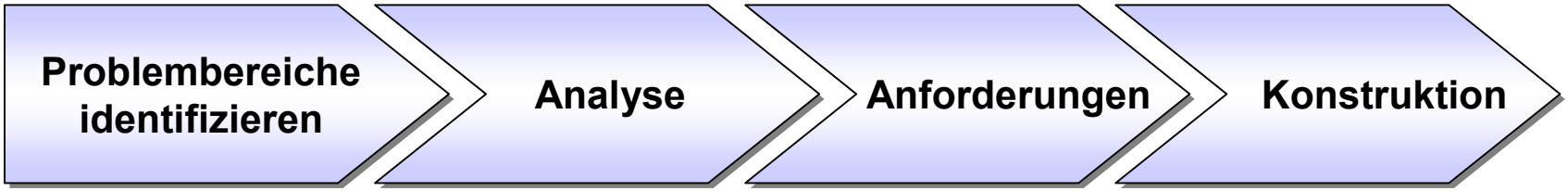
Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

2. Der Weg zur neuen Softwarearchitektur

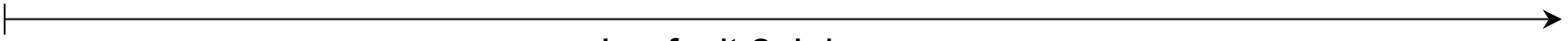
Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Die Vorgehensweise des Reengineering

Vor der Inangriffnahme des Software-Reengineering wurde folgende grundsätzliche Vorgehensweise überlegt:



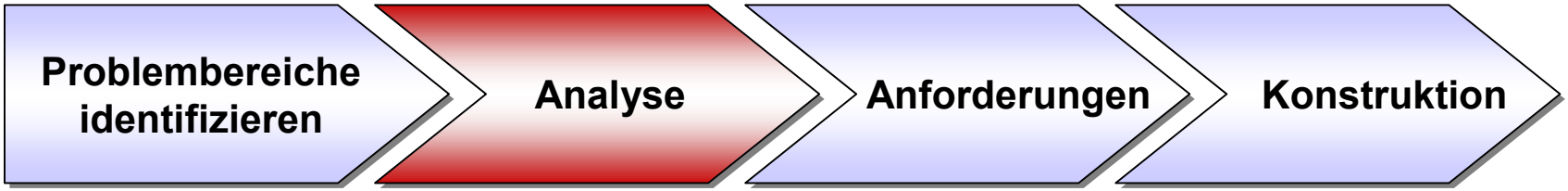
Problembereiche identifizieren	Analyse	Anforderungen	Konstruktion
Motivation, anzustrebende Verbesserungen aus Benutzersicht definieren (bereits gezeigt)	der bisherigen SW, Verstehen der Architektur, Probleme in der Software und betroffene Module erkennen	an neue Software-architektur ableiten und formulieren	der neuen Software-architektur, Vergleich von Designalternativen, Kriterienmatrix, Mischen von Ideen



Laufzeit 2 Jahre

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Der Weg zur neuen Softwarearchitektur

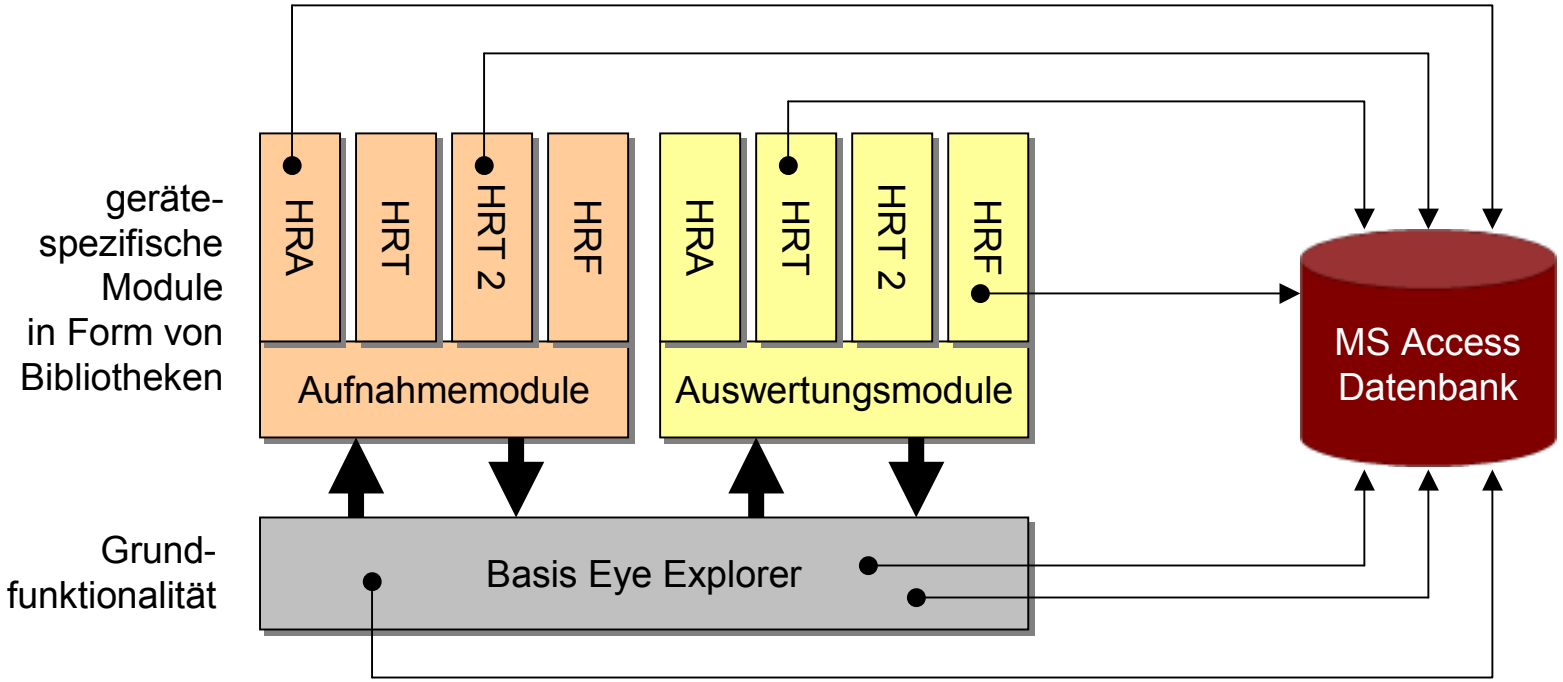


- Durchsicht der bisherigen Software
- Verstehen der Architektur
- Probleme in der Software erkennen
- Probleme auf Softwaremodule zurückführen

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Der Weg zur neuen Softwarearchitektur

Schritt 1: Analyse
a) Durchsicht der bisherigen Software, Verstehen der Architektur, Probleme erkennen



- **hochgradige Verflechtungen** innerhalb der Module und in Richtung der Basis
- **Code** zum Zugriff auf die Datenbank über die ganze Software **verstreut**
- Jedes Modul benutzt **eigene Datenstrukturen** und **speichert Ergebnisse zwischen**

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Schritt 1: Analyse
b) Probleme auf Softwaremodule zurückführen

Die erkannten Probleme lassen auf folgende betroffene Module schließen:

1 Basis Eye Explorer	**68.000 LOC**
5 Aufnahmemodule	**100.000 LOC**
5 Auswertungsmodule	**280.000 LOC**
3 Importmodule	20.000 LOC
6 Low-Level-Bibliotheken für HW-Zugriff	95.000 LOC

betroffen
von Reengineering

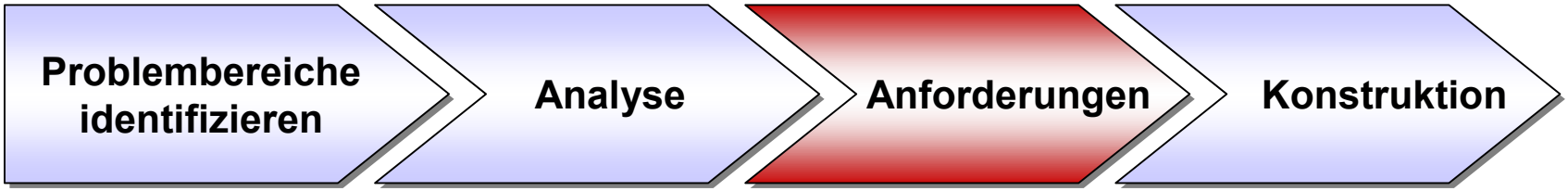
Learned:

Im Projekt basierte die Analyse auf Quellcode und geringer techn. Dokumentation.
Bei der **Analyse durch Externe** wäre ein **Softwaremodell äußerst hilfreich**.

Kompensiert wurde dies **durch eine Vielzahl an Gesprächen und Reviews**.

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Der Weg zur neuen Softwarearchitektur



- Nicht-funktionale Anforderungen an die SW
- Anforderungen an das Reengineering
- Iterative Annäherung an eine vollständige Auflistung der Anforderungen

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Schritt 2: Anforderungen
a) Nicht-funktionale Anforderungen an die Software

Neben den funktionalen Anforderungen waren im Rahmen der **Entscheidung für eine Softwarearchitektur** die **nicht-funktionalen Anforderungen ausschlaggebend**:

- **Sowohl für Einzelplätze als auch für Netzwerke** in Kliniken **geeignet**, wobei die Einzelplätze deutlich überwiegen (ca. 80% der Installationen).
- Weltweit verbreitete bisherige Software Heidelberg Eye Explorer **möglichst einfach auf die neue Technologie** (durch den Anwender) **migrierbar**.
- **Datenbank flexibel austauschbar**, so dass die Vorgaben der Kliniken (Oracle, DB2, MS SQL Server) berücksichtigt werden können (Einzelplatz mit Datenbank MySQL).
- Auf wachsende Benutzer- und Leistungsanforderungen **skalierbar**.
- **Offen für die Ankopplung angrenzender Softwaresysteme** (möglichst keine properitäre Technik bzw. Datenformate)**.**

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Schritt 2: Anforderungen
b) Anforderungen an das Reengineering

Darüber hinaus sind aber auch **Anforderungen speziell des Reengineerings** zu beachten:

- Die bereits entwickelten Module sollen **soweit wie möglich in ihrer fachlichen Logik beibehalten** werden (Schutz des geleisteten Entwicklungsaufwandes).
- Als Entwicklungssprache soll **weiterhin C++** benutzt werden (Entwicklungsteam auf C++ ausgerichtet).

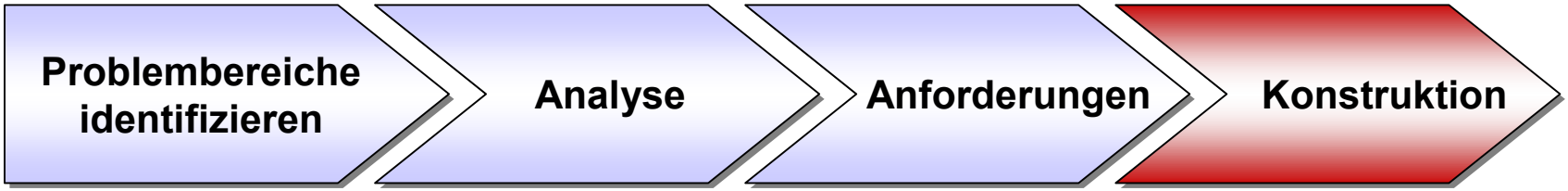
Learned:

Anforderungen konnten **nicht von Anfang an vollständig klar formuliert** werden.

Erst die **iterative Überarbeitung** in einer Reihe von Gesprächen führte zu einer **ausreichend vollständigen Auflistung (für die Entscheidungsfindung wichtig)**.

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Der Weg zur neuen Softwarearchitektur

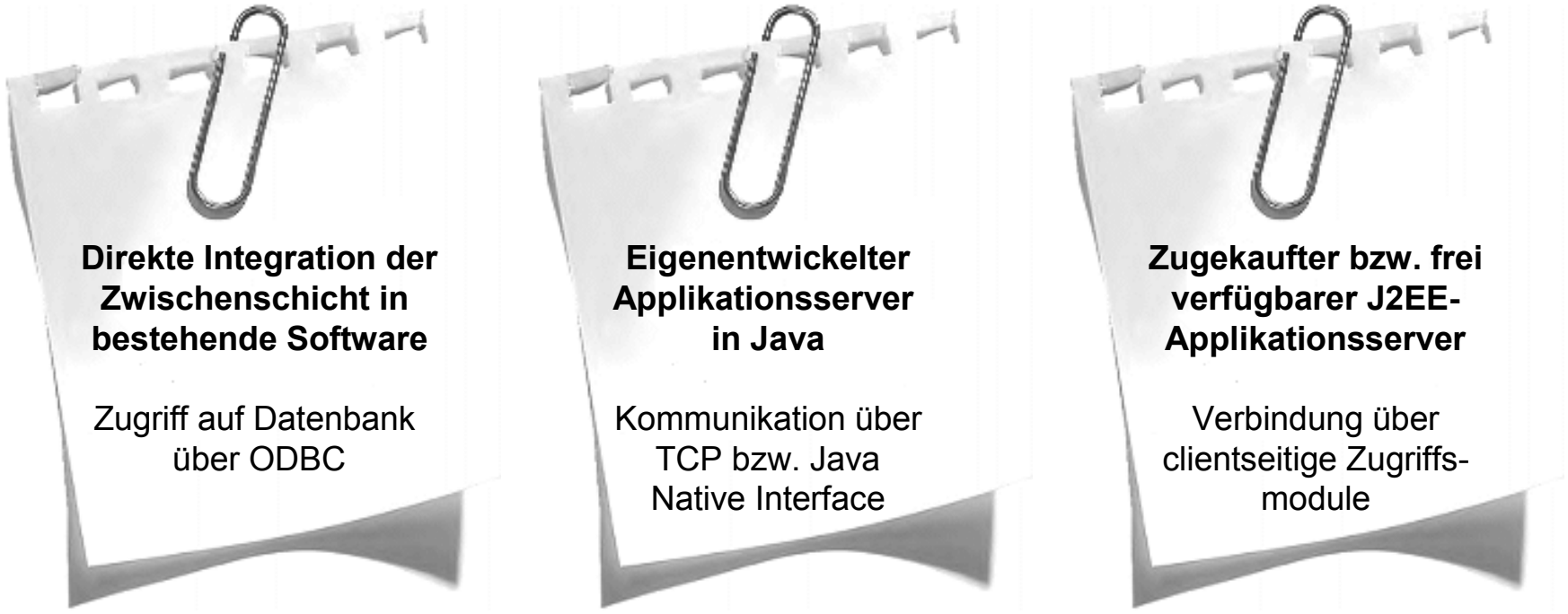


- Suche von potentiellen Lösungsansätzen
- Vergleich v. Alternativen
- Kriterienmatrix
- Mischen von Ideen

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Schritt 3: Konstruktion
a) Suche von potentiellen Lösungsansätzen

Begonnen wurde mit einer informellen **Sammlung von Lösungsideen (Auszug):**



- Tatsächlich als **Story-Cards** notiert, um eine **Diskussion** am Tisch zu ermöglichen.

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Schritt 3: Konstruktion
b) Vergleich von Designalternativen

- Der **Vergleich der verschiedenen Lösungsansätze** stellte sich als **äußerst schwieriger und langwieriger Prozess** heraus.

- **Schwierigkeiten** ergaben sich insbesondere **bei der Prognose**, welchen **Grad der Erfüllung an nicht-funktionalen Anforderungen** die Designalternativen erreichen.
  Wie sollte z.B. die geforderte Flexibilität, Performance sichergestellt werden ?
  Und wie sollte dies den Lösungsansätzen angesehen werden ?

- **Gelöst** wurde dies durch
  - **Rücksprache** und Beratung **mit Experten**,
  - der **Recherche in Erfahrungsberichten** zu den Technologien und
  - der teilweisen **Überprüfung anhand von** einfachen **Prototypen**.

Learned:

Nützlich sind **Szenarios zur Quantifizierung nicht-funktionaler Anforderungen**.

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Schritt 3: Konstruktion
c) Kriterienmatrix

- Ergebnis dieser Arbeiten war eine **Kriterienmatrix der Lösungsansätze mit ihren Eigenschaften** (Vorteile und Nachteile im Bezug auf das Reengineering-Projekt).

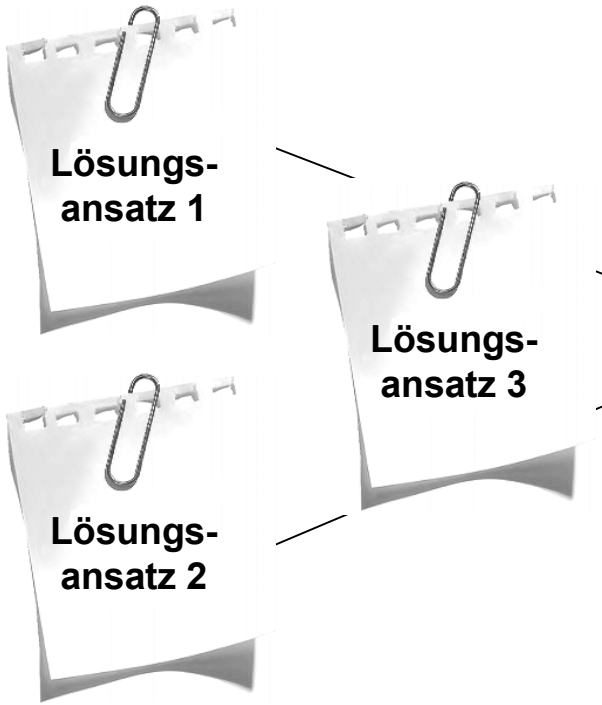
- **Auszug aus der iterativ erstellten Kriterienmatrix:**

Lösungsansatz	*Vorteile*	*Nachteile*
Direkte Integration der Zwischenschicht in bestehende Anwendung	leicht zu integrieren, geringer Entwicklungsaufwand, gute Performance für Einzelplätze	keine Offenheit gegenüber angrenzenden Software-systemen
Eigenentwickelter Applikationsserver in Java	Plattformunabhängigkeit durch die Sprache Java (auch für Unix-Server),	Schwierige Technologie JNI und nicht gewünschte Entwicklungssprache Java
Zugekaufter bzw. frei verfügbarer J2EE-Applikationsserver	Standardtechnologie, die auch von anderen Systemen genutzt werden könnte	Für den großen Teil der Einzelplatzinstallationen schlicht zu komplex

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Schritt 3: Konstruktion
d) Die Mischung an Ideen als optimale Lösung

- Der Vergleich der Designalternativen führte bald zu der Erkenntnis, dass nur eine **Mischung aus den angedachten Ideen** zum Erfolg führen kann.



Für dieses Projekt optimale Lösung:
Form einer nachrichtenbasierten Middleware

Flexibel auf die Bedürfnisse der Arbeitsumgebung anpassbare Lösung:

- für **Einzelplätze** direkt in die Software linkbar
- für **Netzwerke** XML-Kommunikation über TCP
- durch die Sprache C++ kann bereits entwickelter **Code wiederverwendet** werden

XML ist offen für angrenzende Systeme, die direkte Integration sichert den Einzelplätzen die geforderte Performance.

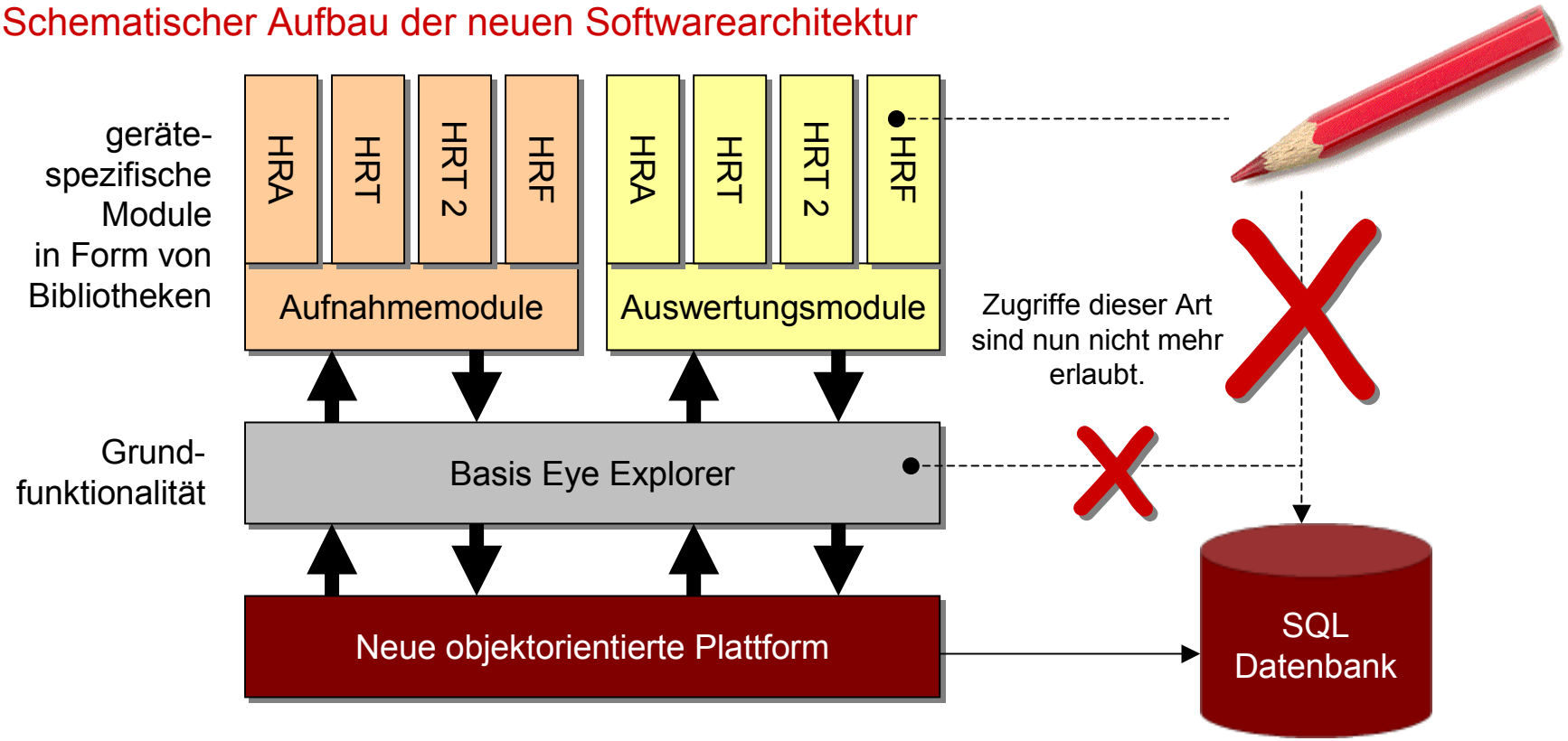
Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

3. Die neue Softwarearchitektur

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Die neue Softwarearchitektur

Schematischer Aufbau der neuen Softwarearchitektur

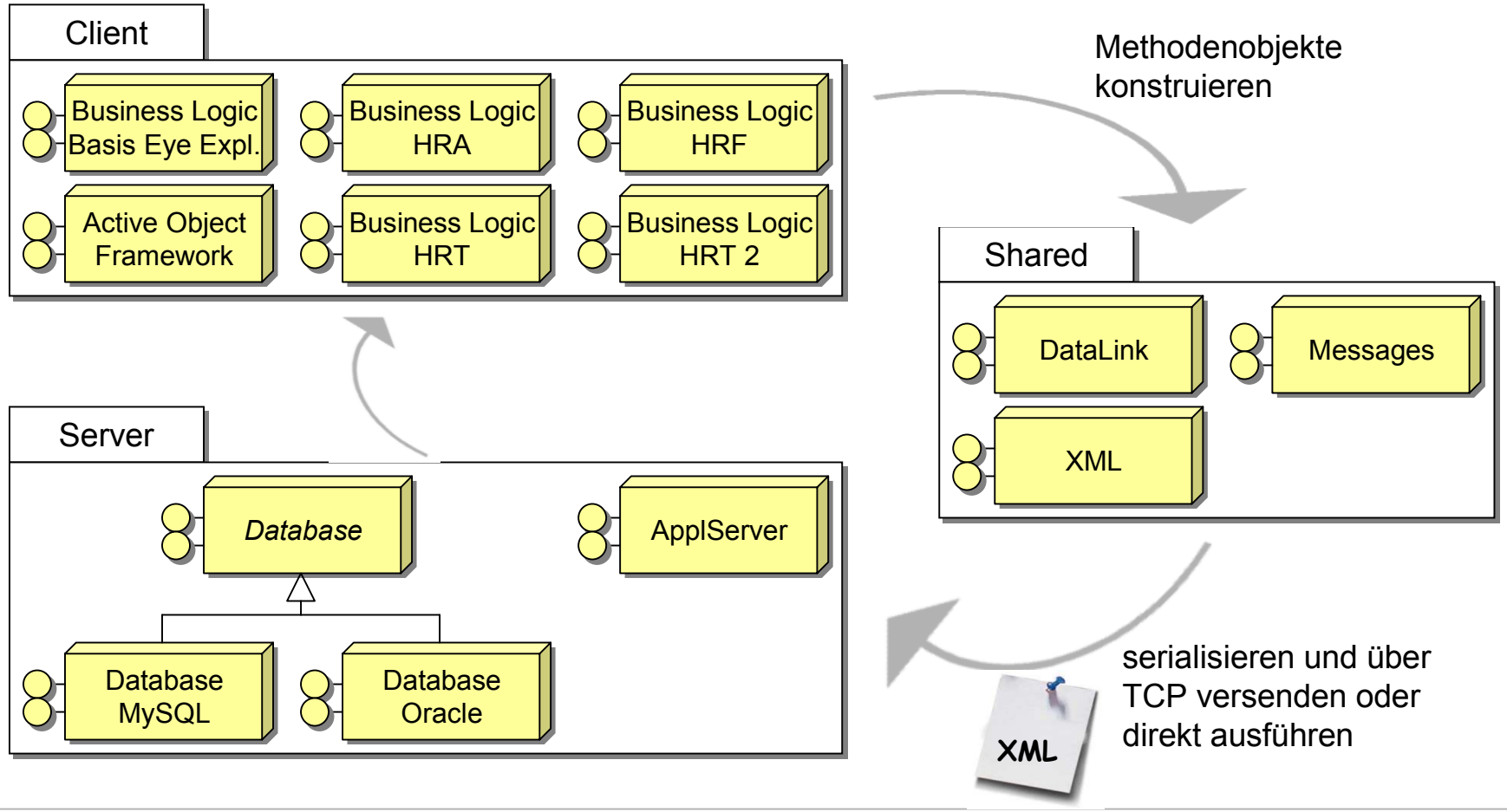


- Eingezogen wurde eine **neue objektorientierte Plattform**, die sowohl die **fachlichen Klassen** als auch die **Verfahren zur Kommunikation mit der Datenbank** kapselt.
- **Vorhandene Logik** in den gerätespezifischen Modulen **wird beibehalten**.

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Die neue Softwarearchitektur

Die neue Softwarearchitektur im Detail (Auszug)



Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Zukünftige Arbeitsschritte

- **Fertiggestellt ist heute**

  a) die neue objektorientierte Plattform und
  b) der Applikationsserver in seiner Grundfunktionalität.
  c) Test der neuen Plattform in ihrer Funktionalität.

  Arbeitsaufwand bis heute: ca. 14 Mann-Monate.

- **Noch durchzuführen ist**

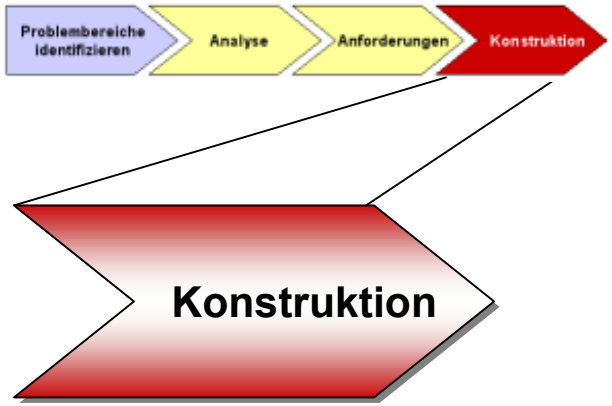
  a) die Integration der Plattform in die bestehende Software Eye Explorer und
  b) in die gerätespezifischen Module sowie
  c) Tests und Performancemessungen.

  Geschätzter noch zu erbringender Aufwand: ca. 12 Mann-Monate.

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Lessons Learned

Standardisierungsmöglichkeiten im Prozess



Darin verborgene Tätigkeiten:

- Suche von potentiellen Lösungsansätzen
- Vergleich von Alternativen
- Kriterienmatrix
- Mischen von Ideen

Vorgehensmodelle wie der Rational Unified Process helfen bei *„Problembereiche identifizieren“, „Analyse“* und *„Anforderungen“* aber wenig bei der *„Konstruktion“.*

Projekt: Schwierigkeit bei der Suche nach der opt. Lösung.

Ein standardisierter Bewertungsprozess hilft bei der Bewertung von Lösungsansätzen auf die Erfüllung nicht-funktionaler Anforderungen (-> Dissertation).

Auszug aus den abgedeckten Punkten:

- Anwendung von **visuellen Bewertungsverfahren** und **Metriken** (z.B. Kopplung)
- **Problembereiche aufspüren**
- systematisch **Lösungsalternativen vergleichen**
- **Erfahrungen** für spätere Projekte **speichern**.

Martin.Moro@wiwi.uni-regensburg.de
0941/943-3215

Fragen

Vielen Dank für die Aufmerksamkeit !

- Haben Sie noch Fragen, Anmerkungen, Tipps oder Wünsche ?

Martin Moro
Martin.Moro@wiwi.uni-regensburg.de
Tel: 0941 / 943-3215