# Implementation

Benjamin Herzog (s68311) 04.07.2014

#### Inhalt

- 1. Aufgabe
- 2. Einordnung
- 3. Lösungsweg
- 4. Ergebnisse
- 5. Reflexion

#### 1. Aufgabe

- geeignete Tools finden
- Sybase-API verwendbar anlegen
- Interface bauen
- Klassen schreiben

#### 2. Einordnung

- Realisierung des Entwurfs
- Bauen von Schnittstellen
- Einhaltung der Rahmenbedingungen
- Wechsel von abstrakter in konkrete Ebene

#### 3. Lösungsweg

- Entwicklungsumgebung
  - NetBeans (Java)
  - Qt Project (C++)
  - Microsoft Visual Studio (C#)
  - (Entwicklung mit HTML, JS, PHP)

#### 3. Entwicklungsumgebungen - NetBeans

- Java als 'entwicklerfreundliche' Sprache
- betriebssystemunabhängig
- relativ langsam
- große Anzahl nutzbarer Bibliotheken (Sybase)
- aufwendiges Erstellen des GUI

### 3. Entwicklungsumgebungen - Qt Project

- C++ als bekannte Sprache aus anderen Modulen
- betriebsystemunabhängig
- Erstellung des GUI über Interface-Builder
- sehr komplizierte Sybase-API
- objektorientiert

#### 3. Entwicklungsumgebungen - MS VS

- sehr moderne Sprache C#
- leistungsfähige IDE
- einfache, starke GUI-Erstellung
- reine Windows-Entwicklung
- gute Sybase-API
- sehr gute Projekt-Organisation

# 3. Lösungsweg

- BitBucket
- Team-Foundation-Server
- Github

# 3. Lösungsweg - BitBucket

- private Repositories
- bis zu 5 Benutzer
- Versionskontrolle

#### 3. Lösungsweg - Team-Foundation-Server

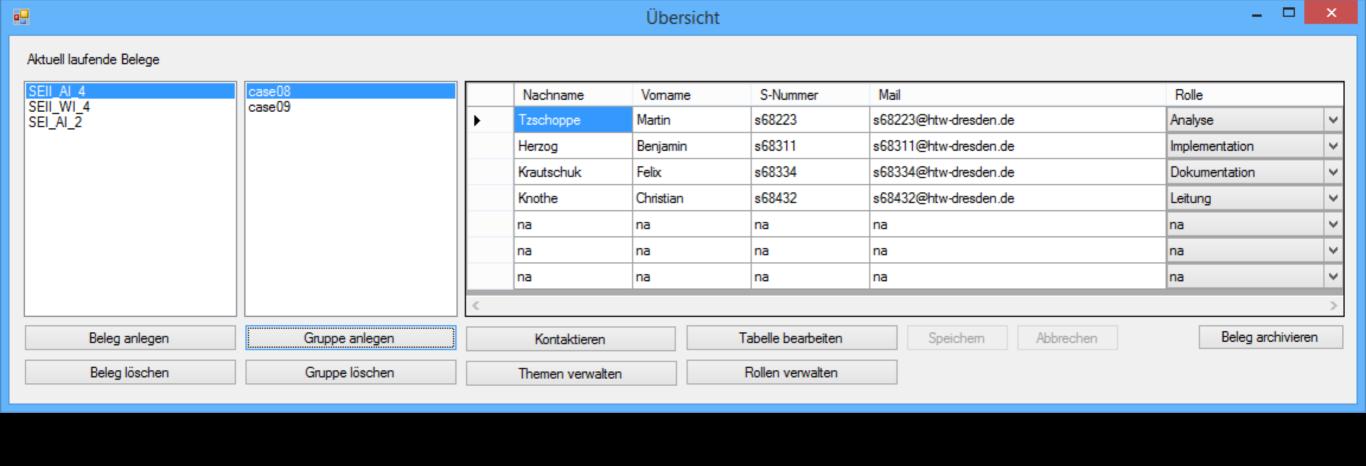
- kostenlose Version f
  ür bis zu 5 Benutzer
- kollaboratives Arbeiten
- Versionskontrolle
- Dienst von Microsoft
- Integration in Visual Studio

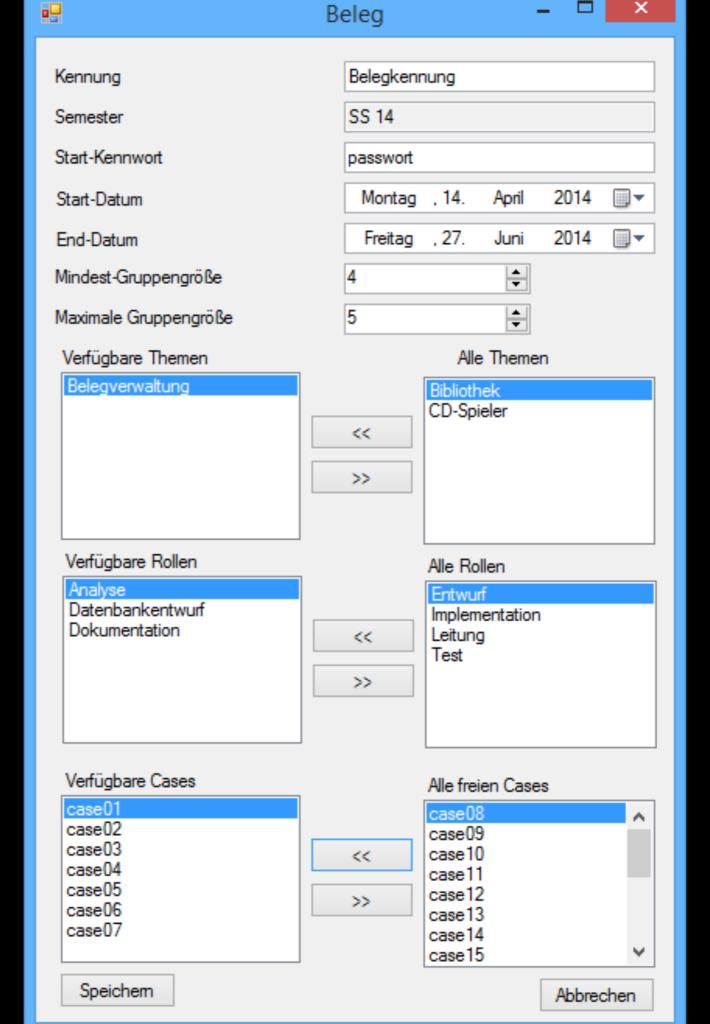
#### 3. Lösungsweg - Github

- Versionskontrolle über Git
- Anlegen von Organisationen
  - einfache Zusammenarbeit
- mehrere Gits für mehrere Projekte
  - Programme für Dozent, Studenten und Planung des gesamten Projektes
- public oder private
- plattformunabhängig

## 4. Ergebnisse - GUI

- Interface Designer von Visual Studio
- automatische Erstellung \*.Designer.cs
- automatische Erkennung von Properties im Code
- Usability-Verbesserung auf Knopfdruck







#### case11

	Nachname	Vomame	S-Nummer	Mail	Rolle	
<b>&gt;</b>	Mustermann	Max	s68444	s68444@htw-dresden.de	Leitung	٧
	na	na	na	na	na	٧
	na	na	na	na	na	٧
	na	na	na	na	na	٧
	na	na	na	na	na	٧
	na	na	na	na	na	٧
	na	na	na	na	na	٧

Thema:

Belegverwaltung

Speichem

Änderungsmöglichkeiten zwischen 14.04.2014 und 27.06.2014

#### 4. Ergebnisse - allgemein

- Klassen als C#-Klassen (.cs)
- Nutzung der Sybase-API über dafür erstellte Database.cs Klasse
- GUI über Windows Forms mit zugehörigen Klassen
- Unterteilung in Programm für Dozent und Studenten
- Nutzung von ReSharper für Code-Verbesserung

### 4. Ergebnisse - Beispiel Datenbank

```
List<string> getFreieCases(Gruppe gruppe)

{
    Database database = new Database();
    List<string[]> ergDB = database.ExecuteQuery(
        "select Casekennung from Zuordnung_BelegCases where Casekennung not in
        (select Gruppenkennung from Zuordnung_GruppeBeleg where Belegkennung=\"" +
        gruppe.Belegkennung + "\") and Belegkennung=\"" + gruppe.Belegkennung + "\"");
    if (ergDB.Count == 0) return null;
    List<string> erg = new List<string>();
    foreach (string[] strings in ergDB)
    {
        erg.Add(strings[0]);
    }
    return erg;
}
```

# 4. Ergebnisse - Beispiel Datenbank 2

```
db.ExecuteQuery("delete from Beleg where Belegkennung=\"" + temp.BelegKennung + "\"");
db.ExecuteQuery("delete from Zuordnung_BelegCases where Belegkennung=\"" + temp.BelegKennung + "\"");
db.ExecuteQuery("delete from Zuordnung_BelegThema where Belegkennung=\"" + temp.BelegKennung + "\"");
db.ExecuteQuery("delete from Zuordnung_BelegRolle where Belegkennung=\"" + temp.BelegKennung + "\"");
db.ExecuteQuery("delete from Zuordnung_GruppeBeleg where Belegkennung=\"" + temp.BelegKennung + "\"");
```

#### 4. Ergebnisse

- schnelles, natives Programm
- übersichtliches Interface
- sicher dank strikter Trennung von Student und Dozent
- nur im Intranet verwendbar

#### 5. Reflexion

- Implementation dank Planung viel übersichtlicher
- Arbeitsaufteilung
- Autoritäten respektiert
- Werkzeuge gut gewählt

#### 5. Reflexion

- längerer Zeitraum —> engere Zusammenarbeit
- stärker am Entwurf beteiligen