# Web Programming Project

## Webshop with Ruby on Rails

Marco Füllemann / Boban Glisovic

# Introduction

# Technical Overview

- Ruby on Rails 4.0
  - devise
  - wkhtmltopdf
  - Paperclip
- jQuery
- foundation
- MySQL

# Ruby on Rails

- object-oriented
- MVC based Web-Framework
- active Record pattern
- principals
  - don't repeat yourself
  - convention over configuration
- scaffolding
- use of ruby package manager „gem"

# features

- T4.2: dynamic navigation based on categories (hiercharchical)
- T4.3: dynamic product list (selected category)
- T5.3: multi language support (de, en)
  - Items, Categories via DB
  - static content via I18n Internalization
  - language_id stored in user session

# features

- T6.3: extended the User model to add some address information
  - Order process checks for complete address
- T6.4: confirmation emails sent after:
  - registration
  - Ordering items
- T7.1:????

# features

- T8.2:  shopping cart implemented via Javascript
  - data stored in a cookie (only id's)
  - add/remove/change items
  - after „order-now" content is passed to an order object

# features

- T8.3: User Accounts implemented with devise
  - authentication framework
  - no role based access control(rbac) implemented yet
- T10.4: Administrator Pages for dynamic content (Categories/Items)
  - CRUD Models

# features

- T10.1/10.2/10.3: MySQL setup
  - DBSchema based on Rails Models
  - DB migrations scripts for changes (rake task)
  - module mysql2 for rails
  - active record as ORM

# features

- T11.1: REST-Webservice for Place (City) Lookup by ZIPCode.
  - Google maps API (geocoder)
  - RESTful
  - Via AJAX Request
- T12.1: non dynamic content stored in yaml files
  - Translations
  - Database information

# features

- T12.2: AJAX
  - shoppingcart
  - Place lookup (Webservice)
  - Store language_id in session
  - Dynamically add and remove form fields in admin site (editing items and categories)
- T12.3: Templates
  - ERB-Template engine

# features

- T12.4: PDF creation
  - Dynamically renders a PDF with order information
  - Use of webkit html2pdf (like browsers do)
  - wickedPDF gem
  - sent by mail to customer

# teamwork

- defined the responsible for each task
- supported each other
- ????

# Lessons learnt

- using a framework != less work to do
  - kind of work changes
  - follow their rules
- realize step-by-step
  - Do not try to make a „jack of all trades device"
- ?