

Demo Day Preparation

Dirk Riehle, Univ. Erlangen

AMOS C01

Licensed under [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/)

Agenda

1. Demo day

- a. Demo day slide
- b. Demo day posters
- c. Demo day video

2. Final release

- a. Final documentation
- b. Final planning documents
- c. Final release tag

3. After-work

- a. Project report
- b. Project retrospective

1. Demo Day

The AMOS Demo Day

The demo day is the final day of the course

- The demo day is organized as a fair (“Messe”)
- Student teams show the results of their project
- Audience are industry partners and fellow students

Demo Day Process

Opening (20 min.)

- General introduction
- 1 slide, 1 min. introduction by each team (one speaker)

Demos (duration depends on number of projects)

- After the introduction, we split up into the different projects
- One demo room (online) for each project
- At least one person from each project needs to be ready to demo
- Demo day participants (including students without booth duty) roam around

Closing (5 min.)

- We come together in the main room to say goodbye

Typical Demo Day Schedule

Time	Duration	Responsible	Title	Room
10:15	10 min	Riehle	Introduction	Main room
10:25	10 min	Teams	One slide summary	Main room
10:35	20 min	Teams	Demo	Demo rooms
10:55	20 min	Teams	Demo	Demo rooms
11:15	20 min	Teams	Demo	Demo rooms
11:35	20 min	Teams	Demo	Demo rooms
11:55	20 min	Teams	Demo	Demo rooms
12:15	5 min	Riehle	Conclusions	Main room

One-Time Deliverable: Demo Day Slide

Please create **one** 16x9 slide to show during the demo day opening

Online Demo Room

We will create Zoom breakout rooms for each project

You will demo your work in your breakout room

- **Demo your project using your laptop**
- Support your demo using a slide deck; should contain
 - One slide on product management
 - One slide on software development
 - A team photo slide (can be screenshot)
- Have the demo day video ready as a backup

Use of Corporate Identities

Please use your university logo

Please use your team logo

Please use your industry partner logo, but ask first

One-Time Deliverable: Demo Day Posters

Please create two posters for your demo day booth

- One product management / product owner poster
- One software development poster

Twenty Minutes of Fame

At	What happens / to do
0min.	Demo room opens, participants stream in
1min.	Demo starts
10min.	Demo finishes, discussion starts
19min.	60 second countdown to room closing starts
20min.	Room closes, everyone is pulled back into the main room

Demo Preparation

Dos

- Have a clean user interface
- Have a story to tell, for example,
 - A day in the life of ...
 - A workflow example
- Use domain terms and examples
 - In the user interface (labels, titles)
 - In the stories you tell
- Make the demo data reentrant
 - You will have to start over repeatedly
 - You want to start at the same point

Don'ts

- Use “test1” or “user2” as labels
- Not follow the advice on the left

Demo First! Slides... Fifth? Nineteenth?

Demo your software! That's what it is about

A good approach is to demo a main feature or use case

- And only then explain it using slides

And then demo another feature or use case that adds to the previous one

- And then explain it using slides

And finally demo a third feature

- After which you will open the discussion

Don't be afraid of open time for questions

One-Time Deliverable: Demo Day Video

Please create a 3 min. video demoing your work

- **You should show running software, not just talk about it!**

The demo video will be your demo backup

- Will also be posted on our blog and on LinkedIn

Demo Execution

Have two people ready to demo

- One explains what is going on (talks to people)
- One demos the software in line with story

2. Final Release

One-Time Deliverable: Final Documentation

Please clean-up and finalize your documentation

One-Time Deliverable: Final Planning Documents

Please clean up your planning documents, in particular the final release plan

One-Time Deliverable: Final Release Tag

Please clean up your code base, in particular set the final-release tag

3. After-work

One-Time Deliverable: Project Report

Please create a project report using our template

- The report will be posted on our blog and on LinkedIn (together with your video)

One-Time Deliverable: Project Retrospective

Please perform a project retrospective (guided by your Scrum Master)

Summary

1. Demo day

- a. Demo day slide
- b. Demo day posters
- c. Demo day video

2. Final release

- a. Final documentation
- b. Final planning documents
- c. Final release tag

3. After-work

- a. Project report
- b. Project retrospective

Thank you! Any questions?

dirk.riehle@fau.de – <https://oss.cs.fau.de>

dirk@riehle.org – <https://dirkriehle.com> – [@dirkriehle](#)

Legal Notices

License

- Licensed under the [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/) license

Copyright

- © Copyright 2023 Dirk Riehle, some rights reserved

Archive

Example Posters 1 / 2

**FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG**

**JOWBANA**



Personalfragebogen 2.0

Personnel Questionnaire Automation



Personalfragebogen 2.0

Benutzername
[Benutzername eingeben]

Passwort
[Passwort eingeben]

Remember me! ☐ (Passwort speichern)

© 2015 ARSO GROUP 4



Employees - Overview

Benutzername: [Benutzername eingeben]

Passwort: [Passwort eingeben]

Remember me! ☐ (Passwort speichern)

© 2015 ARSO GROUP 4





Personalfragebogen 2.0* is a personal data management software solution, supporting companies of any size in hiring new employees more efficiently.

The product improves the hiring process by automating the collection of personal data during the hiring procedure, and provides aid in managing the collected data.

Demo Day
Friedrich-Alexander Universität
Erlangen
Wednesday, 15 July 2015 10.15-11.45



*Personalfragebogen 2.0 is a joint project between FAU's OSR Group and DATEV eG.

**FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG**

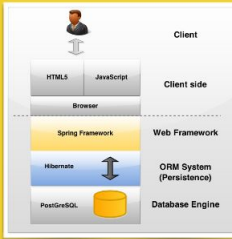
**JOWBANA**

Personalfragebogen 2.0

Personnel Questionnaire Automation



Software Architecture



```
graph TD
    Client[Client] <--> Browser[Browser]
    subgraph Client_side [Client side]
        Browser
    end
    subgraph Web_Framework [Web Framework]
        Spring[Spring Framework]
    end
    subgraph ORM_System [ORM System (Persistence)]
        Hibernate[Hibernate]
    end
    subgraph Database_Engine [Database Engine]
        PostgreSQL[(PostgreSQL)]
    end
    Browser <--> Spring
    Spring <--> Hibernate
    Hibernate <--> PostgreSQL
```

Technology

Name	Function
Spring Framework (4.1.6)	Java based Web Framework
Java SE (7u79)	Fundamental Platform
HTML5	Client-side core technology
Selenium (2.45.0)	UI Testing/Integration Testing
JUnit (4.12)	Java Unit Testing Framework
Hibernate ORM (4.3.9)	ORM System for persistence
PostgreSQL (9.4.1)	Database Management System
Tomcat 7.0.61	For local deployment

*Personalfragebogen 2.0 is a joint project between FAU's OSR Group and DATEV eG.

Example Posters 2 / 2

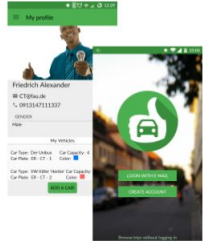


CroudTrip!

sponsored by Elektrotit

The CroudTrip! application wants to revolutionize the car-ride-sharing market with its easy, user-friendly and highly automated way of organizing shared Trips!

The Product



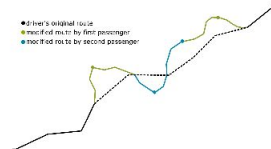
- Offer and join shared Trips at short-notice!
- For drivers: Easily find passengers on the way you are going anyway ... and earn money with it!
- For passengers: Reach your destination comfortably!
- We will automatically match you to the best offer in real-time!
- Simply check-in and check-out of your Trips using NFC on your device!
- No direct Trips? No problem - Join a SuperTrip! with multiple drivers!

The Concept



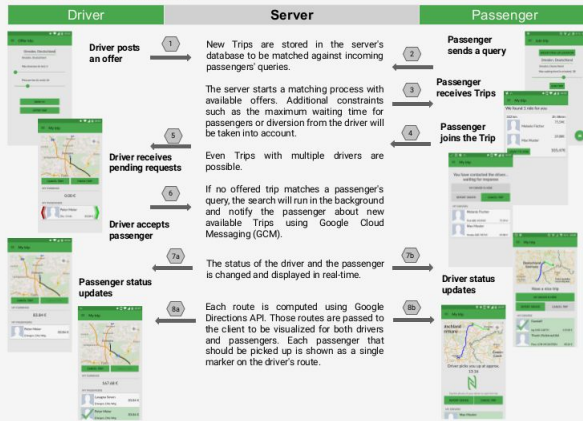
- Combine multiple offered routes to serve passengers even if there is no direct connection available
- Find routes which can pick up a passenger from his start position or drive to his final destination
- Subdivide those routes, compute the closest pair of those waypoints and use it as a "connection point"
- If the distance of the closest pair is too large, start a recursive matching process with these two waypoints

Multiple Passengers

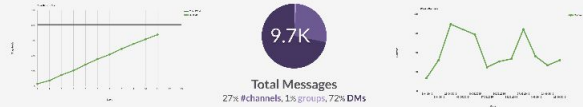


- Match multiple passengers with one driver who will pick them up and bring them to their destinations in an optimal order
- Optimal order is constrained by given internal order of each waypoint pair, because each passenger has to be picked up before the driver reaches his destination location
- Compute optimal order by solving the Travelling Salesman Problem via Brute Force (max. 4 passengers)

The Interactions



The Process



- Total # of story points: 370
- Development Speed: 30.9
- 13.23% of total effort used for bugfixing

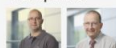
- Slack as main communication tool
- Integrations for Travis CI, Github and Crashlytics

- Total # of Commits: 727
- Lines of Java Code: 15362
- Lines of Comments: 3938

The Team



The Sponsor



The University

