A CSS Refactoring Story

Optimize Printing

Marc Schuler – Alcor Academy – Running – May 2022

Last year during autumn season

- Spent a Lot of Time for maintaining the Printing-Software
- Complex analyses
- Identified a handful of bugs

Analysis of the bugs

- Deep in the code
- Not possible to fix it without structure change
- Possible to solve with balcony over balcony
- The uncertainty of adding new bugs is great

- → Decision: reimplement a small part of the printing where the bugs are in. In this case:
  - Print request management
  - Exceptionhandling

The plan for the improvement

- Reimplementing the print request management
- The effektiv Printing-Software stays as it is
- Use modern Softwareachitecture (Onion Architectur)
- Reduce deprecated Frameworks (OpenMDX)
- Improv of maintailability with a clear data structure
- By not changing the functionality

This Spring we started with this little project

We had to realize that what seemed so clear and simple at the beginning now seemed completely unclear and complex.

Facts

- Software is 15 years old
- The range of functions was not clear
- Lot of functionality added in the last 15 years
- Many balconies were built to fix bugs

We start with

- Analyzing the code
- Talking to the users
- Analyzing the data and logs

Present Day

- Overview of the functionality
- Desinging the new software
- Starting implementation

My Conclusion

- Not possible to do only e technical refactoring
- Is needed to understand was is the business – need of the software
- Ask the question: What is still needed? (Business-Refactoring)

Questions

Thank you!

References

Images:

- Google search

Story:

- CSS Project ‘Optimize Printing’

Marc Schuler

- marc.schuler@css.ch