Do Multi-Fidelity Levels improve Mockup-Driven Development?

René Milzarek, 19.12.2016, Garching-Forschungszentrum

Software Engineering for Business Information Systems (sebis)
Department of Informatics
Technische Universität München, Germany

wwwmatthes.in.tum.de
1. Introduction

2. Problem Identification
   - Requirements Elicitation
   - Research Gap Identification
   - Research Questions

3. Solution Design
   - Definition of Term
   - Prototyping Process
   - Implementation

4. Evaluation

5. Outlook
Siemens GS IT HR
(Information Technology – Human Ressources)

Human Resources and Supply Chain Management services and solutions for all divisions worldwide.

Headquarter
Siemens AG
Global Services – Information Technology
Otto-Hahn-Ring 6
81739 München
Overview

1. Introduction

2. Problem Identification
   - Requirements Elicitation
   - Research Gap Identification
   - Research Questions

3. Solution Design
   - Definition of Term
   - Prototyping Process
   - Implementation

4. Evaluation

5. Outlook
Core Idea

Systematic reuse of user interface prototypes for the generation or scaffolding of user interface code.

→ Non-disposable UI prototypes
Informal Interviews

Conducted with the head of the department and the head of development (N=2).

1. Collaboration
2. Custom component catalog
3. Export UI code
4. Integration with ALM solution
5. On-premise solution
6. Test on target platform
7. Platform support of the prototypes
Semi-structured Interviews

Conducted with N=11 experts of different professional and organisational backgrounds.

- Majority of interviewees worked for a large company (72.7%).
- 6 different roles were covered.
- Mean of professional experience in years was 11.82 years (σ = 7.93).
- 81.8% have used UCD design methods in their professional lifes.
- Remaining 18.2% received high-fidelity mockups as a specification document.
- Application focused on the requirements elicitation phase.
- 81.8% (9 persons) worked with high-fidelity mockups exclusively.
Semi-structured Interviews

- 9 persons (81.8%) did not have an established UCD process.
- 81.8% (9 persons) rated the application predominantly positive.

Negative aspects:

- Create wrong expectation of a production ready user interface
- Disposable character of UCD artefacts
- Collaboration and communication between stakeholders is challenging
- Lack of a standard UI component catalog

Application of UCD methods

- Yes: 82% (9)
- No: 18% (2)
## Problem Statement

### Research Gap Identification

<table>
<thead>
<tr>
<th>Collaboration</th>
<th>Justinmind</th>
<th>iRise Studio</th>
<th>Balsamiq</th>
<th>Pixate Studio Beta</th>
<th>Visual Paradigm</th>
</tr>
</thead>
<tbody>
<tr>
<td>(deliver to endusers, collect feedback)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom Component Catalog</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>?</td>
</tr>
<tr>
<td>Export Code</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Integration with ALM</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Link to requirements, single source for reporting)</td>
<td></td>
<td></td>
<td>✓</td>
<td>x</td>
<td>?</td>
</tr>
<tr>
<td>On-Premise Solution</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>(Host collaboration platform internally)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test on the Target Platform</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Platform-Support</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Multi-Fidelity Mockups</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Most mature Enterprise Solutions

- Justinmind
- iRise Studio

### Academic Research Gap

- Applies
- Partially applies
- Does not apply

✓ = applies, ○ = partially applies, ✗ = does not apply
<table>
<thead>
<tr>
<th>RQ1</th>
<th>What is the definition of Mockup-Driven Development and the different fidelity levels?</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ2</td>
<td>What are the requirements for a Multi-Fidelity Mockup-Driven Development system and how could an implementation look like?</td>
</tr>
<tr>
<td>RQ3</td>
<td>How to evaluate if a Multi-Fidelity Mockup-Driven Development system improves the software development process?</td>
</tr>
</tbody>
</table>
Overview

1. **Introduction**

2. **Problem Identification**
   - Requirements Elicitation
   - Research Gap Identification
   - Research Questions

3. **Solution Design**
   - Definition of Term
   - Prototype-Driven Development Process
   - Implementation

4. **Evaluation**

5. **Outlook**
Definition

„Degree of exactness with which something is copied or reproduced“, Oxford Dictionary

- Performed literature review to identify fidelity-levels and their artefacts
  - Low-Fidelity Prototypes (Sketch & Wireframe)
  - High-Fidelity Prototypes (Mockup & Software Prototype)
- Analysed the number of style properties of a button across different fidelity levels (Sketch: 7 → Mockup: 37 → Product: 71)
Solution Design
Definition of Terms – Multi-Fidelity

Prototypes

Low-Fidelity Prototype

High-Fidelity Prototype

Sketch

Wireframe

Mockup

Software Prototype

Product
Definition of Terms – Multi-Fidelity

Prototypes

Low-Fidelity Prototype

High-Fidelity Prototype

Product

Sketch

Wireframe

Mockup

Software Prototype
Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering, Rivero 2014

- Coined the term: MockupDD (Mockup-Driven Development)

- Create User Stories and Mockups → Mapping through a SUI (Structural User Interface) Model

- Use the SUI Model to generate Code and MDWE Models

→ Focusing on the transition between high-fidelity mockup and the product

→ No benefits from a multi-fidelity approach
Solution Design
Prototype-Driven Development Process

Prototype-Driven Development Process

- **Create and Improve Mockups**
  - Customer
  - User

- **Share Prototypes**
  - Collect Feedback

- **Include**
  - Specify new Component
  - Generate Code
  - UI Component Catalog

- **Prototype Tool**
  - Prototyping Tool

- **Create and Improve**
  - Mockups

- **Include**
  - Usability Engineer
  - Requirements Engineer

- **Specify new Component**
  - Software Engineer
  - Software Engineer
  - Requirements Engineer

- **Generate Code**
  - UI Code
Component Catalog

Prototype for the creation and maintenance of UI components

- Utilizing the "backend-as-a-service" Parse in a Docker Compose setup
- Definition of custom view model based on UIML
- Development of an AngularJS web application
Solution Design
Implementation - Demo

Proteon - Collaborative Platform

Structure
EXPAND ALL   COLLAPSE ALL

General Information

Name: Table
Tag: sebis-table
Category: Tables
Description: Customised table with optional pagination.

Created By: Unspecified
Created At: 11/15/2016 10:22
Last Change By: Unspecified
Last Change At: 11/24/2016 00:39

Table
table

Table Row
<tr>

Table Cell
<td>

Table Cell
<td>

Table Row
<tr>

Table Cell
<td>

Table Cell
<td>

Pagination
<sebis-pagination>
Overview

1. Introduction

2. Problem Identification
   - Requirements Elicitation
   - Research Gap Identification
   - Research Questions

3. Solution Design
   - Definition of Term
   - Prototyping Process
   - Implementation

4. Evaluation

5. Outlook
Usability Walkthrough

- Custom and System Usability Scale (SUS) questionnaire conducted (N=8)
  - 5 point Likert-scale
- 75% (6 persons, $\sigma = 1.07$) strongly agreed that the collaboration and reuse of existing components is improved
- 4 persons (50%, $\sigma = 0.76$) agreed that the process could accelerate the software development
- SUS score of 67.19 (average of 68 in literature)

⇒ No clear benefits from a multi-fidelity approach
⇒ Process enhances collaboration and enables systematic reuse
1. Introduction

2. Problem Identification
   - Requirements Elicitation
   - Research Gap Identification
   - Research Questions

3. Solution Design
   - Definition of Term
   - Prototyping Process
   - Implementation

4. Evaluation

5. Outlook
Outlook

Future Work

- Source Code Repository
  - Import
    - UI Component Catalog

Create and Improve Mockups

- Include
  - Prototyping Tool

Generate Code

- User
  - Collect Feedback
    - Customer
  - Share Prototypes

- Specify new Component
  - UI Code

- Create and Improve Mockups
  - Import
    - Source Code Repository
Thank you! Questions?
Backup
## Definition of Terms – Fidelity Levels

<table>
<thead>
<tr>
<th>Category</th>
<th>Criterion</th>
<th>Sketch</th>
<th>Wireframe</th>
<th>Mockup</th>
<th>Software Prototype</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Technique</td>
<td>paper-based</td>
<td>computer-based</td>
<td>computer-based</td>
<td>software-based</td>
<td>software-based</td>
</tr>
<tr>
<td></td>
<td>Speed</td>
<td>fast</td>
<td>fast</td>
<td>slow</td>
<td>slow</td>
<td>slowest</td>
</tr>
<tr>
<td></td>
<td>Cost</td>
<td>cheap</td>
<td>cheap</td>
<td>expensive</td>
<td>expensive</td>
<td>most expensive</td>
</tr>
<tr>
<td>Fidelity</td>
<td>Low-Fidelity</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Medium-Fidelity</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>High-Fidelity</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Behaviour</td>
<td>Navigation</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Interactive Elements</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Structure</td>
<td>Responsive Design</td>
<td>multiple static screens</td>
<td>multiple static screens</td>
<td>single interactive screen</td>
<td>CSS / Other Technology</td>
<td>CSS / Other Technology</td>
</tr>
<tr>
<td></td>
<td>Placeholders</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>

✓ = applies, ○ = optionally applies, ✗ = not applied
## Definition of Terms – Fidelity Levels

<table>
<thead>
<tr>
<th>Category</th>
<th>Criterion</th>
<th>Sketch</th>
<th>Wireframe</th>
<th>Mockup</th>
<th>Software Prototype</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Information</strong></td>
<td>Label</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Images</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Style</strong></td>
<td>Colors</td>
<td>black &amp; white</td>
<td>black &amp; white</td>
<td>colored</td>
<td>colored</td>
<td>colored</td>
</tr>
<tr>
<td></td>
<td>Icons</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Typography</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

✓ = applies, ○ = optionally applies, ✗ = not applied
Solution Design
Definition of Terms – Multi-Fidelity

<table>
<thead>
<tr>
<th>Sketch</th>
<th>Low Fidelity</th>
<th>„Paper and Pencil“ or „Whiteboard and Post-It“ approach.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>Basic functionality &amp; UI interaction</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wireframe</th>
<th>Medium Fidelity</th>
<th>Skeletal illustration of the UI, which usually has no styling, colors or graphics.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>Content of the UI</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prototype</th>
<th>High Fidelity</th>
<th>Almost undistinguishable from the final UI, could often be executed on the final platform.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>Design, Fully executable UI prototype</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th></th>
<th>Code of the final UI, which often relies on frontend frameworks (e.g. Bootstrap, Foundation)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>➔ Automatically generate!</td>
<td></td>
</tr>
</tbody>
</table>
Solution Design
Definition of Terms – Multi-Fidelity

Multi-fidelity Prototyping of User Interfaces

- Identified the same research gap of missing support for fidelity transitions
- Focus on the transition from “no-fi” (hand drawn) to “lo-fi” → Gesture recognizer
- Low shape detection speed → Problematic when used for complex UIs
- Static templating: “custom element could be drawn in lo-fi and a predefined widget could be added in me-fi or hi-fi”

Definition of Terms – Mockup-Driven Development

Mockup Driven Web Development

- Definition of Cascading Tree Sheets (CTS)
  → Describe relationship between content and structure

- CTS as input for the generation of a web application

Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering

- Coined the term: MockupDD (Mockup-Driven Development)

- Mockup as “requirement elicitation helper”

- Create User Stories and Mockups
  → Mapping through a SUI (Structural User Interface) Model

- Use the SUI Model to generate Code and MDWE Models
