Patchwork Prototyping for Web Applications

Medieninformatik Hauptseminar Wintersemester 2009 / 2010 „Prototyping“
Contents

• History

• Definitions
  • Mashup
  • Rapid prototyping
  • Patchwork prototyping

• Patchwork prototyping
  • Comparison
    • Low fidelity prototyping
    • High fidelity prototyping
    • Horizontal and vertical prototyping
  • Project examples

• Conclusion
History

**Participatory design** (users involved in design process)
- At the beginning: Computerization of work
- Later: Widespread use of computers
- Today: Internet access is usual

**Free /Libre Open Source Software (FLOSS)**
- No licensing costs
- Lower risk of rights violations
- Over the time: Growing amount of products
- Today: Great innovative potential for web applications
**Definition: Mashup**

Data sources (XML files, ...)

Application Programming Interfaces

---

Popular example: **HousingMaps.com** (invented in 2005)
- Collects data from Craigslist.org (real-estate listings)
- Allows filtering of information
- Shows the processed data with Google Maps API
Definition: Mashup
Definition: Rapid prototyping

Time is an important and precious resource!

- Quickly develop and iterate new application
- Improve communication between developers and users
- Insights of strengths / weaknesses during development phase
- Explore new features and alternatives

Advantages
- Work more efficiently
- Spend less money for wrong developments
- Application has not to be finished
Definition: Patchwork prototyping

Mashup \rightarrow \text{Patchwork prototyping} \rightarrow \text{Rapid prototyping}

- Approach to rapid prototyping
- Participatory design concept
- Produces high fidelity prototypes
Definition: Patchwork prototyping

- Using mashups, FLOSS and web services
- „Glueing“ different components together
- Collecting feedback of users (user-driven method)
- Continuously developing prototypes
- Using the prototypes in daily work activities

Example: Hiking community
- Display hiking paths on a map: Google Maps
- Bulletin board for discussion and questions: phpBB
- Charts for height levels of hiking routes: ExtJS
- Collection of excellent routes: MediaWiki
**Patchwork prototyping**
Comparison to low fidelity prototypes

- Produced with office materials (pen, paper, scissors, ...)
  - Computer is not necessary
- Facilitator needed (for demonstration purposes)
- Useful in early project phase

<table>
<thead>
<tr>
<th><strong>Similarities</strong></th>
<th><strong>Differences</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Can be produced cheap</td>
<td>Knowledge of programming languages not necessary</td>
</tr>
<tr>
<td>Can be produced fast</td>
<td>Help to decide about fundamental design issues</td>
</tr>
<tr>
<td></td>
<td>Limited functionality</td>
</tr>
<tr>
<td></td>
<td>Limited interactivity</td>
</tr>
</tbody>
</table>
Patchwork prototyping
Comparison to low fidelity prototypes
Patchwork prototyping
Comparison to high fidelity prototypes

Similarities
• Simulate real functionality
• Offer complete interactive UI
• Feedback already available after a short period of time
• Nearly same behaviour and functionality as the final product → Prototypes sometimes seen as final product

Differences
• More time needed for developing and modifying
• Higher development costs
• Skilled programmers are always required
Patchwork prototyping
Comparison to high fidelity prototypes
**Patchwork prototyping**  
Comparison to horizontal and vertical prototypes

**Horizontal prototypes**  
- Include a lot of features  
- Realized features are only superficially complete

**Vertical prototypes**  
- Include only selected features  
- Realized features are almost like in the final product
Patchwork prototyping
Comparison to horizontal and vertical prototypes

Patchwork prototypes
- Include a lot of features
- Realized features are only superficially complete
- Share similarities of both kinds
- Original categorization of the two kinds does not match anymore

Patchwork prototypes
- Include only selected features
- Realized features are almost like in the final product
Project example
Patching together community tools

- 1997: Start of the project platform
- 2003: Implementation of LAMP (Linux, Apache, MySQL, PHP)

- First: Development of community platform from scratch
- Later: Use of phpBB (bulletin board) and TinyMCE (WYSIWYG editor)
- Problem: Security vulnerability in phBB $\rightarrow$ successful attack

- First: Components ,,glued“ together with simple hyperlinks
- Later: More complex adaption (single sign-on for all components)

- At the end: phpBB software was seen as overscaled
- Result: ,,Homemade“ bulletin board that offered a better integration with the rest of the platform
Project example „Wasabe“
Combining powerful search engines

Wikipedia / Amazon Search and Browse Environment
- Parallel search for term in Amazon and Wikipedia
- No own Wikipedia API → use of Google SOAP API
- Maximum of multiple hundred lines of source code

Further improvements
- Use of Amazon API to get related items and ISBNs
- Collected ISBN numbers used to get information about the availability of items in the local library
- Use of AJAX (Asynchronous JavaScript and XML) to speed up the application
Project example „Wasabe“
Combining powerful search engines
Conclusion

Patchwork prototyping is a promising new technology

• It combines positive attributes of ...
  • Low and high fidelity prototypes
  • Horizontal and vertical prototypes
  • Large number of APIs, mashups and qualitative FLOSS
    → Large number of patchwork prototypes in the future?

But...
• ...not yet relevant in academic literature
• ...not possible in every scenario
• ...main „needs“ for patchwork prototyping have to be fulfilled
  (e.g. collection of feedback, availability of third-party software)