A Systematic Analysis of the Effect of Task Clarity on Software Development Design

Werner Mellis
### How to describe software development design?

<table>
<thead>
<tr>
<th>Dimensions of Software Development Design</th>
<th>Dimensions of Software Development Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational Design</td>
<td>Process Design</td>
</tr>
<tr>
<td>Division of Labor</td>
<td>Kind of Horizontal Division of Labor</td>
</tr>
<tr>
<td>Coordination</td>
<td>Kind of Coordination</td>
</tr>
<tr>
<td>Process Design</td>
<td>Important Coordination Mechanism</td>
</tr>
<tr>
<td>Formalization</td>
<td>Interface to the Customer</td>
</tr>
<tr>
<td>Motivation and Leadership</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td><strong>...</strong></td>
<td><strong>...</strong></td>
</tr>
<tr>
<td>Engineering Design</td>
<td><strong>...</strong></td>
</tr>
<tr>
<td>Requirements Engineering</td>
<td><strong>...</strong></td>
</tr>
<tr>
<td>Quality Assurance</td>
<td><strong>...</strong></td>
</tr>
<tr>
<td>Architectural Design</td>
<td><strong>...</strong></td>
</tr>
<tr>
<td>Design Methods</td>
<td><strong>...</strong></td>
</tr>
<tr>
<td><strong>...</strong></td>
<td><strong>...</strong></td>
</tr>
</tbody>
</table>

**Software Development Design**

**Contingency Approach**

**Organizational Analysis**

**Deriving Recommendations**

**Conclusion**

© W. Mellis
## Software development design standard
- explicit prescriptions -

<table>
<thead>
<tr>
<th>dimensions of software development design</th>
<th>PSQM</th>
</tr>
</thead>
<tbody>
<tr>
<td>process design</td>
<td></td>
</tr>
<tr>
<td>kind of horizontal division of labor</td>
<td></td>
</tr>
<tr>
<td>kind of coordination</td>
<td>in advance</td>
</tr>
<tr>
<td>important coordination mechanism</td>
<td>planning, organizational programming</td>
</tr>
<tr>
<td>interface to the customer</td>
<td>batch, unidirectional, narrow band, formal ...</td>
</tr>
<tr>
<td>quality assurance</td>
<td>verification</td>
</tr>
</tbody>
</table>
## Software development design standard
- implicit assumptions -

<table>
<thead>
<tr>
<th>Dimensions of software development design</th>
<th>PSQM</th>
</tr>
</thead>
<tbody>
<tr>
<td>process design</td>
<td>incremental, sequential</td>
</tr>
<tr>
<td>kind of horizontal division of labor</td>
<td>process oriented</td>
</tr>
<tr>
<td>kind of coordination</td>
<td>in advance</td>
</tr>
<tr>
<td>important coordination mechanism</td>
<td>planning, organizational</td>
</tr>
<tr>
<td>interface to the customer</td>
<td>batch, unidirectional,</td>
</tr>
<tr>
<td></td>
<td>narrow band, formal ...</td>
</tr>
<tr>
<td>quality assurance</td>
<td>verification</td>
</tr>
</tbody>
</table>
### Are there alternatives?
- not caused by ignorance and inability -

<table>
<thead>
<tr>
<th>??</th>
<th>dimensions</th>
<th>PSQM</th>
</tr>
</thead>
<tbody>
<tr>
<td>process design</td>
<td>incremental, sequential</td>
<td></td>
</tr>
<tr>
<td>kind of horizontal division of labor</td>
<td>process oriented</td>
<td></td>
</tr>
<tr>
<td>ad hoc</td>
<td>kind of coordination</td>
<td>in advance</td>
</tr>
<tr>
<td>mutual adjustment</td>
<td>important coordination mechanism</td>
<td>planning, organizational programming</td>
</tr>
<tr>
<td>interface to the customer</td>
<td>batch, unidirectional, narrow band</td>
<td></td>
</tr>
<tr>
<td>quality assurance</td>
<td>verification</td>
<td></td>
</tr>
</tbody>
</table>
Contingency approach

Software development design

Contingency approach

Organizational analysis

Deriving recommendations

Conclusion

© W. Mellis
### Contingency approach

<table>
<thead>
<tr>
<th>unclear task</th>
<th>clear task</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>adaptive model</strong></td>
<td><strong>dimensions</strong></td>
</tr>
<tr>
<td>process design</td>
<td>kind of horizontal division of labor</td>
</tr>
<tr>
<td>kind of coordination</td>
<td>important coordination mechanism</td>
</tr>
<tr>
<td>interface to the customer</td>
<td>quality assurance</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Clear and Unclear Development Tasks

• **Uncleanness of Requirements**
  – Customer and technological requirements are unclear.

• **Dynamic Nature of Requirements**
  – Customer and technological requirements change during development time.

• **Technology Dynamics**
  – Knowledge about software technology and software process technology change during development time.
Informations Flows and Interdependencies

Development activities

Modules 1 - x

Analyse Module 1  
Design Module 1  
Implem. Module 1  
Test Module 1  

Analyse Module 2  
Design Module 2  
Implem. Module 2  
Test Module 2  

Analyse Module x  
Design Module x  
Implem. Module x  
Test Module x  

Information flow in case of clear development task.

Information flow in case of unclear development task.
A Systematic Analysis of the Effect of Task Clarity

Process Design

- Project Start
- Concept Freeze
- Market Introduction

Concept Development
Implementation

Concept Time
Response Time

Total Lead Time

Software development design
Contingency approach
Organizational analysis
Deriving recommendations
Conclusion

© W. Mellis
Design of Positions, Organizational Structure

Software development design

Contingency approach

Organizational analysis

Deriving recommendations

Conclusion

© W. Mellis
Quality Assurance

- Verification vs. validation

- prove correctness vs. help to understand the problem

- formal vs. informal
Interface to the Customer

Development process

Deriving recommendations

Conclusion

Software development design

Contingency approach

Organizational analysis

© W. Mellis
Coordination

- coordination in advance
  - by plan
  - by program

- ad hoc coordination
  - direct advice
  - mutual adjustment
## Conclusion

<table>
<thead>
<tr>
<th>unclear task</th>
<th>clear task</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaptive model</td>
<td>transformational model</td>
</tr>
<tr>
<td>parallel</td>
<td>incremental, sequential</td>
</tr>
<tr>
<td>product oriented</td>
<td>process oriented</td>
</tr>
<tr>
<td>extensively ad hoc</td>
<td>extensively in advance</td>
</tr>
<tr>
<td>mutual adjustment</td>
<td>planning, organizational programming</td>
</tr>
<tr>
<td>frequent, bidirectional, broad band</td>
<td>interface to the customer</td>
</tr>
<tr>
<td>validation, informal</td>
<td>batch, unidirectional, narrow band</td>
</tr>
<tr>
<td>quality assurance</td>
<td>verification, formal</td>
</tr>
</tbody>
</table>

### Dimensions
- process design
- kind of horizontal division of labor
- kind of coordination
- important coordination mechanism
- interface to the customer
- quality assurance

### Model
- incremental, sequential
- process oriented
- extensively in advance
- planning, organizational programming
- batch, unidirectional, narrow band
- verification, formal
Transformational model

- Analysis
- Architecture review
- Design review
- Module test
- Integration
- System test
- Interface test
- Module design
- Coding
- Function and behavior
- Prescriptions for implementation
- Program code
- Prescriptions for integration

• Requirements
• Modules
• Interfaces
• Function and behavior
• Internal structure of modules
• Prescriptions for implementation
A Systematic Analysis of the Effect of Task Clarity

Adaptive model

- Environment (application, competition, technology)
- Organization
  - Team 1
  - Team 2
- Software
  - Module 1
  - Module 2
- Architect / Allocator

Software development design
Contingency approach
Organizational analysis
Deriving recommendations
Conclusion
Thank you for listening!