Value Model Clash Analysis

Mohammed Al-Said

Annual Research Review
USC-CSE
March 2003
Research Overview

• **Problem**
  – Currently, many software models exist
  – Each model is based on certain assumptions
  – Many software projects embrace models with conflicting assumptions and get into trouble
  – Little information exist about the problem of model clashes

• **Research:**
  – Develop technique(s) to identify model clashes
  – Determine the relationship between model clashes and risk

• **Value:**
  – Reduce software project risk and rework
### Common Models Used in SW Development

#### Value/Success Models
- Win-Win
- Business Case Analysis
- Software Warranties
- QFD
- 10X
- Six Sigma
- Award Fees
- IKIWISI
- JAD

#### Process Models
- Spiral
- Waterfall
- Open Source
- Business Process Reengineering
- CMM’s
- Peopleware
- IPT’s
- Agile
- Evolutionary

#### Product Models
- UML
- CORBA
- COM
- Requirements
- Architectures
- Product Lines
- OO Analysis & Design
- Domain Ontologies
- COTS
- GOTS

#### Property Models
- System Dynamics
- Metrics
- Capabilities
- Simulation and Modeling
- Environment

- Some model assumptions conflict
Model Clashes SpiderWeb

Users Value Propositions

- Many features
- Changeable Requirements
- Applications Compatibility
- High levels of Service
- Voice in acquisition
- Flexible Contract
- Early Availability
- Ease of Transition
- Ease of Maintenance
- Applications Compatibility
- Voice in acquisition

Maintainers Value Propositions

- High levels of Service
- Voice in acquisition
- Flexible Contract
- Ease of Transition
- Ease of Maintenance
- Applications Compatibility
- Voice in acquisition

Acquirers Value Propositions

- Mission: cost/effectiveness
- Limited budget, schedule
- Government standards Compliance
- Political correctness
- Development visibility & control
- Rigorous contract
- Flexible contract
- Ease of meeting budget & schedule
- Stable requirements
- Freedom of choice: Process
- Freedom of choice: Team
- Freedom of choice: COTS/reuse

Developers Value Propositions

- Early Availability
Examples of Conflicting Assumptions

<table>
<thead>
<tr>
<th>Model</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterfall</td>
<td>Complete requirements are specified in advance of system design and implementation: Frozen requirements</td>
</tr>
<tr>
<td>COTS</td>
<td>COTS product(s) capabilities and structures constrain system requirements and architectures</td>
</tr>
</tbody>
</table>

Consequence:

- Unnecessary custom software development
- Change of requirements
Examples of Conflicting Assumptions

<table>
<thead>
<tr>
<th>Model</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Assurance</td>
<td>System testing ensures against failure presentation under different operational circumstances</td>
</tr>
<tr>
<td>COTS</td>
<td>COTS maintenance and evolution are out of the control of the customer, developer, and user.</td>
</tr>
</tbody>
</table>

**Consequence:**

- Degradation to system’s reliability and availability
- System’s long term viability
Examples of Conflicting Assumptions

<table>
<thead>
<tr>
<th>Model</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAD</td>
<td>System development is tightly scheduled (i.e. timeboxed)</td>
</tr>
<tr>
<td>COTS</td>
<td>Development schedule allows enough time to assess, evaluate, and integrate COTS product(s)</td>
</tr>
</tbody>
</table>

**Consequence:**

- Select the wrong COTS product
- Schedule impact
- Functionality impact
- Aesop Example: 6 Months --> 2 Years
Model Assumptions Identification
Approach
<table>
<thead>
<tr>
<th>Assume Yes</th>
<th>Requirements Aspect</th>
<th>Assume No</th>
</tr>
</thead>
<tbody>
<tr>
<td>★ Waterfall ★ Fixed-reqts/cost/schedule contract ★ Formal derivation and verification</td>
<td>Completely specified in advance of design and development</td>
<td>★ Evolutionary development ★ Agile methods ★ COTS-assessment-intensive system ★ Stakeholder win-win</td>
</tr>
<tr>
<td>★ Waterfall ★ Fixed rqts./cost/schedule contract</td>
<td>Low risk of infeasibility</td>
<td>★ Spiral ★ Cost/schedule as independent process ★ Heterogeneous COTS elements</td>
</tr>
<tr>
<td>★ Waterfall ★ Formal derivation and verification ★ Milestone planning</td>
<td>Highly Stable</td>
<td>★ Evolutionary development ★ Agile methods ★ Cost/schedule as independent process</td>
</tr>
</tbody>
</table>

–Models in left column will clash with models in right column
Relationship between Model Clashes and Risk

Risk Pair <Probability of loss, Size of loss> answers:

I. What is the likelihood that unsatisfactory outcome will occur?

II. If unsatisfactory outcome occurs, what is the measure of loss?

Model-Clash Pair <Occurrence Probability, Severity>?
Model Clashes Occurrence Probability & Severity

CeBASE Data Base

Fall2000-Spring2002 Projects

Weekly Progress Reports

Project Top 10 Risks

Assumptions

Models  Model Clashes  Stakeholders

Frequency  Severity

Occurrence Probability
## Mode Clashes Occurrence Probability & Severity: Example

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>S</th>
<th>M</th>
<th>OP</th>
<th>SV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer needs extensive, on demand user/customer interaction</td>
<td>Dev</td>
<td>SS</td>
<td>0.82</td>
<td>M</td>
</tr>
<tr>
<td>User/Customer are available at limited times</td>
<td>User/Cust</td>
<td>SS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User/Customer are free to add/modify system requirements during the system implementation</td>
<td>User/Cust</td>
<td>PD</td>
<td>0.46</td>
<td>H</td>
</tr>
<tr>
<td>Changes/additions to system requirements require extra budget and schedule</td>
<td>Dev</td>
<td>PP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Formulating Risk Statement using Model Clashes*

If Models Clash then there is potential harmful Consequence (Time) (risk)

Formally:
\[ \exists x y z w \ model(x) \land model(y) \land assumption(x,z) \land assumption(y,w) \land clash(z,w) \rightarrow \exists v \ consequence(clash(z,w),v) \text{ where } (x \neq y \neq w \neq z \neq v) \]

* Builds on CMU-SEI Risk Management work
Model Clashes Co-occurring Consequences

**Business Case:** Developer implements the product in 12 months

**COCOMO:** 20 months are needed to implement the product

**Formally:**

\[ \exists x y z w \ model(x) \land model(y) \land assumption(x,z) \land assumption(y,w) \land clash(z,w) \rightarrow \exists v_1 \ldots v_n ((\text{consequence}(\text{clash}(z,w),v_1) \land \text{consequence}(\text{clash}(z,w),v_2) \land \ldots \land \text{consequence}(\text{clash}(z,w),v_n)) \right) \]

where \( (x \neq y \neq z \neq w \neq v_1 \neq v_2 \neq \ldots v_n) \)
Model Clashes Cascade Consequences

assumptions

**Waterfall:** Complete requirements are specified in advance of system design and implementation

**COTS:** Requirements are flexible and negotiable till COTS is selected

cascade consequence set

- Late requirements
- Late development
- Lost product benefits

Formally:

\[ \exists xyzw \ model(x) \land model(y) \land assumption(x,z) \land assumption(y,w) \land clash(z,w) \rightarrow \exists v_1 \ldots v_n ((\text{consequence}(\text{clash}(z,w),v_1) \rightarrow \text{consequence}(\text{clash}(z,w),v_2) \rightarrow \ldots \rightarrow \text{consequence}(\text{clash}(z,w),v_n)) \]

where \( x \neq y \neq z \neq w \neq v_1 \neq v_2 \neq \ldots v_n \)
Value of Model Clashes Identification and Avoidance

Total Risk Exposure = \( \sum_{i=1}^{n} MC_i(PL \times SL) \)

Risk Reduction - LCO and LCA Milestones

Projects

LCO
LCA
Distribution of Clashes between Software Models

Model Clashes Types Distribution

- Product-Product: 30%
- Product-Process: 6%
- Product-Property: 16%
- Product-Success: 4%
- Process-Process: 3%
- Process-Property: 3%
- Property-Property: 13%
- Success-Property: 7%
- Success-Success: 16%
- Success-Process: 4%
Distribution of Clashes between Software Models
Contribution of Model Clashes Types to Software Project Risk

Contribution of Model Clash Types to Project Risk

- Product-Product: 24%
- Product-Process: 6%
- Product-Property: 9%
- Process-Process: 10%
- Process-Property: 2%
- Property-Property: 13%
- Success-Property: 18%
- Success-Product: 10%
- Success-Process: 3%
- Success-Success: 7%
Contribution of Inter and Intra Model Clashes to Project Risk

Contribution of Inter and Intra Model Clashes to Project Risk

% Total Risk

Inter Model Clashes

Intra Model Clashes
Research Contribution

1. Assumptions and model-clashes of the most common software engineering models

2. An insight into model-clashes properties: (causes, patterns, relations)

3. Processes and visualization techniques for rapid model-clash identification

4. The relation between model-clashes and risk in software projects
Questions?