The Role of Architecture in Outsourced Software Development

Michel R.V. Chaudron
chaudron@chalmers.se

Chalmers/Gothenburg University, Sweden
Leiden University, The Netherlands

Outline

• Based on research of my PhD student Werner Heijstek

• Intro Software Architecture
• Intro Global Software Development
• Modeling and Architecture in GSD
  • Anti-patterns and pitfalls
  • Impact on communication, process, product
• Concluding Remarks
Intro Software Architecture

Software Architecture =def= 
The decomposition into subsystems
+ The relations between the subsystems
+ The principles that guide the evolution
+ The design decisions that determine the quality properties of the system

The Architecture Modeling Zoo

Loose → Sketchy

Agile development favours personalisation style for disseminating knowledge

Strict → Very strict

Traditional ‘waterfall’ development was heavy on the codification of knowledge

⇒ Many different styles in the zoo
Value of Architecture Modelling

- Understanding
- Communicating
- Precise Specification
  "contract"
- Guiding / Monitoring
- Analysing / Predicting

Option for benefits are manifold, but scattered
More codification → more automation

Relating Quality of Sequence Diagrams
and Defect Density in Classes

Quality of SD is significantly correlated with defect density.
More detailed SD => fewer defects
Global Software Development

Software work undertaken at geographically separated locations across national boundaries in a coordinated fashion.

Involves:
- Communication – information exchange
- Coordination – groups, activities and artifacts
- Control – adhering to goals and policies

Arguments for doing global software development

- Reducing development cost
- Compensate shortage of IT skilled staff
- Reducing development time (“follow the sun”)
- Outsource non-core business
Chaudron, Lund, Lucas Days, 24 Oct 2012

Challenges in GSD

**Temporal**
- When can I reach them?
- Waiting delays:
  - next telco
  - response to e-mail

**Geographical**
- Who should I contact?
- Lack of informal communication

**Socio-cultural**
- Cultural misunderstandings
- Different domain knowledge
- Different values (quality, hierarchy)

Local architect: “developers think their code is good enough when it resembles the architecture.”

Extra Costs of Offshoring (in %)

<table>
<thead>
<tr>
<th></th>
<th>Meta-Group</th>
<th>CIO magazine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search and Contract</td>
<td>1 – 2</td>
<td>0.02 – 2</td>
</tr>
<tr>
<td>Communication</td>
<td>1 – 3</td>
<td></td>
</tr>
<tr>
<td>Process Changes</td>
<td>0 - 10</td>
<td>1 – 10</td>
</tr>
<tr>
<td>Cultural Differences</td>
<td>2 – 5</td>
<td></td>
</tr>
<tr>
<td>Lost Productivity</td>
<td></td>
<td>3 – 27</td>
</tr>
<tr>
<td>Efficiency</td>
<td>0 – 20</td>
<td></td>
</tr>
<tr>
<td>Governance</td>
<td>5 - 7</td>
<td>6 - 10</td>
</tr>
<tr>
<td>Turnover at Offshore site</td>
<td>1 - 2</td>
<td></td>
</tr>
<tr>
<td>Travel</td>
<td>2 - 3</td>
<td></td>
</tr>
<tr>
<td><strong>Total extra %</strong></td>
<td><strong>13 – 52 %</strong></td>
<td><strong>10 – 50%</strong></td>
</tr>
</tbody>
</table>

Source: Offshoring information technology, E. Carmel and P. Tja, 2005
**Development Process & Handover**

*Waterfall Process*
- Requir. Eng.
- Architecture
- Design
- Implementation
- Test
- Maintenance

*Incremental Process*
- continuous change
- no handover point
- Maintenance

Incremental processes need continuous access to architecture

---

**Long Chain / Hierarchy**

- Information moves slowly
- Information gets corrupted → low architecture compliance
- Remote architect ‘protects’ prestigious information advantage.

- Remote developers have no access to client
- Authority for deciding on architecture is often at local site only
- Remote developers are unknown → low team-feel/trust
Move Architecture Responsibility?

Promises a cheaper solution ...

But, the remote architect has
- less domain knowledge
- no client contact
→ slow down & reduce quality

Staff Turn over

Staff turnover → knowledge evaporation

Codified knowledge can survive
Impact of Architecture in GSD

Communication

• Standardized notation

• Less ambiguity

“developers asked for clarification of non-UML diagrams”

Better understanding of requirements

Impact of Architecture in GSD

Improved knowledge sharing

- Involvement in creating architecture increases understanding and commitment

- All developers have access to design information at any time & place (assuming tooling is in place)

- Codification reduces knowledge evaporation
Impact of Architecture in GSD

Process

• Dissemination of architecture to all developers needs explicit attention

• Verification of understanding of and of conformance to the architecture seems a good practice

• More strict modeling of architecture requires → more strict process of change management

---

Summary

Understandable software architecture representation

+ → Developers knowledge of software architecture

+ → Compliance of software architecture

+ → Software Quality

+ → Project Success

+ → positive influence

The solution is neither to add documentation, or to abandon documentation – it is to get better documentation. – D.L. Parnas, 2006

Low Hanging Fruit

Use
- Architecting Case Tool
- Modeling Guidelines
- Shared Versioning Repository
Open for Collaboration
chaudron@chalmers.se

- You are our ‘users’ – we need your feedback
  Your project? Let us hear your ideas!

- (EU) projects in the making:
  - Mixed Criticality Systems
  - Automated Quality Driven Design / Design Space Exploration
  - Assessing severity of defects/flaws in UML/system designs
  - Automated updating of UML based on progressing src code

• Complementary work

The Impact of Model Driven Development on the Software Architecture Process

Werner Heijstek  Michiel R. V. Chaudron

Leiden Institute of Advanced Computer Science
Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
http://www.liacs.nl/~heijstek,chaudron
{heijstek,chaudron}@liacs.nl
Arguments for doing global software development

- Reducing development cost
- Reducing development Lme (“follow the sun”)
- Compensate shortage of IT skilled staff