Low Ceremony Architecture

or

Tastes Great! Less Filling!

Jeromy Carriere
eBay, Inc.
jcarriere@eBay.com

with thanks to many great folks at Vistaprint
Vistaprint

- Business domain: marketing services for small business
- Attitude: data-driven, energetic, passionate, execution-focused
- Tension: we’re not a startup any more
A Cultural Immune Response

• A perception of big-“A” architecture
• A lack of observed value in “setting standards” and “drawing diagrams”
• Disconnect between “architects” and development practice
The Problem

- This is typical of companies in the internet space
- Ironically, they need it most
- Need to invest in an uncertain future - how?
HELLO
my name is

Opportunity
A Definition of Architecture

“A framework for informed and visible decision-making that explicitly trades off among traditionally competing quality properties such as time to market, performance, flexibility, maintainability and testability.”
The Shift

• Make every engineer responsible for architectural reasoning
• Make architectural investment visible
• Be accountable
Organizational Changes

- Centralized platform architecture
- Decentralized solution architecture
Architectural Guiding Principles

• The “guardrails” for consistent, scalable decision-making

• There are five
The Architectural Guiding Principles

1. Articulate distinct choices and select the one that maximizes full economic value

2. Choose designs that will result in the smallest, least coupled software

3. Build software that is robust, intentionally designed, with understood limitations

4. Utilize standardization to increase predictability and scalability

5. Plan for post development phases, including deployment, testing, operations and maintenance
Themes and a Roadmap

• Simple, visible themes to address architectural challenges

• A goal-based roadmap for execution
## Roadmap

<table>
<thead>
<tr>
<th>Theme</th>
<th>Short Term (now-3 months)</th>
<th>Mid Term (3-6 months)</th>
<th>Long Term (6 months+)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural Guiding Principles</td>
<td>Define</td>
<td>Apply</td>
<td>Refine</td>
</tr>
<tr>
<td>Improve visibility</td>
<td>- Current state architecture models&lt;br&gt;- Define methodology for “Early Warning System”</td>
<td>- Track architecture quality contributions&lt;br&gt;- Establish Early Warning System steering committee to identify quality “cliffs”</td>
<td>- Target state architecture models&lt;br&gt;- Enhance load environment to accurately reflect customer behavior</td>
</tr>
<tr>
<td>Create architecture processes</td>
<td>- Project architect accountability&lt;br&gt;- Consulting architect oversight&lt;br&gt;- Code review approach</td>
<td>- Architecture and design reviews&lt;br&gt;- Routine code reviews&lt;br&gt;- Adherence to quality gates</td>
<td>- Consistent major project postmortems</td>
</tr>
<tr>
<td>Manage impact</td>
<td>- Create isolation through components&lt;br&gt;- Document cross-business unit interactions</td>
<td>- Deploy tools to validate and monitor design rules&lt;br&gt;- Track problems that cross business unit boundaries</td>
<td>- Map business processes to interaction model</td>
</tr>
<tr>
<td>Regularize architecture decision-making</td>
<td>- Pilot LAAAM for quality analysis</td>
<td>- Apply economic models for valuing sustaining/enablement projects</td>
<td>- Consistently apply build/buy decision process&lt;br&gt;- Track design debt</td>
</tr>
<tr>
<td>Consistency of practice</td>
<td>- Knowledge gap analysis</td>
<td>- Establish approach to continuous training</td>
<td>- Syllabus designed&lt;br&gt;- Knowledge rollout strategies piloted</td>
</tr>
</tbody>
</table>
Roles, not Titles

• “Project” and “consulting” architects
• Clearly-defined accountability
• Well-specified decision rights
Autonomy

- The key to managing complexity
- In organizations, processes and systems
Practices

• Valuation of transformative projects and architecture practices
• LAAAM for rational and transparent decision-making
• Managing technical debt as a portfolio
• Design and code reviews
• Explicit current and target state
Aside: measuring productivity

Martin: Just ran into a request to measure productivity of software projects using Line of Code per hour. *sobs* #zombieswontdie


Martin: @sjcarriere I disagree:
http://www.martinfowler.com/bliki/CannotMeasureProductivity.html

Me: @martinfowler I expected. 😊 I believe there are meaningful measures that we can use to improve productivity within the software process.

Martin: @sjcarriere true, but that’s very different to saying you can measure productivity itself.

Me: @martinfowler You have to measure something to improve it. For software productivity, it starts with the long-term value of the output.

Ref: How to Measure Anything by Hubbard
Outcome

• Success!
• Well, mostly
Final Words

• Architects, go write code!*

*Or, do whatever it takes to understand what your architecture practice should be!
Some Lessons

• Architecture is the bearer of quality, but reasoning about architecture is reasoning about potential.

• You’ll rarely (never?) know in advance if a decision is right, but make sure you know afterward.

• Figure out who defines goodness of your work and make them happy.

• Enabling autonomy of organizations and systems is the way you scale.

• Don’t let “pragmatism” become a disguise for shortsightedness.

• Plan for technology retirement, not just adoption.

• Don’t boil the frog with standards.

• Be dumb. Ask smart questions.

• Don’t undervalue slack.