New approaches to software applications meeting organizational and process needs

by

David G Messerschmitt
University of California at Berkeley

Copyright notice

© Copyright David G. Messerschmitt, 2006. This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License.

To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/2.0/ or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
Software Ecosystem

Vendor issues

Upstream value: requirements

User-centric issues

Downstream value

Marketplace: Industrial-organization issues

History: typical evolution

User-internal development (differentiation)

Buy (one of several incompatible) vendor solutions

Examples:
- Design tools for semiconductors
- ERP

Evolution toward standardization
History: typical evolution

- Make
- Buy

Reasons for abandoning “make”

- Increasing development/maintenance costs, less and less directed toward differentiation
- Maturing consensus among organizations on needs and practices
- Growing intra- and inter-organizational interoperability and compatibility issues
History: typical evolution

Problems with "buy"

- Forced to choose among competing options
  - Heavy switching costs to change vendors
  - Incompatibility issues among vendors
- Agency costs
- Organization and processes become software-defined (exactly backwards)
  - Disruption
  - Deployment costs
Biggest challenge

• Designing software to meet organizational and process needs is arguably the least mature and weakest link in the software industry

Better way?

• Undoubtedly!
• Conceptual understanding developing
• Experiments and role models under way
Some characteristics of a “better way”

- End-user organizations heavily engaged in defining needs and requirements
- Ability to modify, extend, evolve as business changes
- Interoperability and compatibility

Interoperability and compatibility across vendors and boundaries
Summary

• Disruptive innovation most often originates with end-users, not their vendors
• Collaboration among vendors and end-users throughout the design-evolution cycle benefits innovation, helps match organization and process, and achieves needed commonality
• New industrial-organization and collaborative models needed to support this

End-user innovation
Democratizing Innovation: a few salient points

• Innovation by end users is frequent, pervasive, and important
  – Information asymmetry (uses vs. technology)
  – Customized solutions (development costs vs. agency costs)

• Technology (e.g. collaboration tools) and platforms (e.g. CAD) empower democratized innovation

Con’t

• End users usually place innovations in the public domain, and this magnifies value
• User innovation communities are common
• Open source software not as new as it appears
**Democratizing Innovation:**
vendor/producer/provider roles

- Systematizing search for and designs based on user innovations has proven to be highly successful for manufacturers (e.g. 3M)
- Platforms for innovation (e.g. custom semiconductors)
- Services (e.g. open source software)

**Benefits of diversity**

- Major disruptive innovations come from a variety of unexpected sources
- Widely varying approaches and perspectives, and freedom of thought are critical
- Large laboratories or companies are not effective at this
- This is another reason most such innovation originates in the end-user community
Collaborative entrepreneurship

• Advanced economies focus on creating and exploiting innovation
• Only a small fraction of innovation within a single firm can be captured and appropriated by that firm
• “Networked collaboration” among firms is the coming major advance in industrial organization

Collaborative entrepreneurship addresses several needs

• Diversity of ideas
• Sharing end-user resources for common solutions
• Addressing user-vendor information asymmetry through collaboration
• Achieving vendor interoperability and compatibility
Information asymmetry

Vendors
- Re-engineer rather than automate
- Generalize
- Technology innovation

End-users
- Own the problem
- Domain expertise
- Know objectives and constraints

Community software development

- Standardization via common code base (rather than documentation), especially one that focuses on a platform
- Flexibility and evolution through:
  - local modifications/extensions
  - plug-ins
Example: e-Learning

- Unsatisfactory past approaches:
  - Internally developed handcrafted solutions
  - Incompatible monolithic commercial solutions
- Now trying community development

e-Learning application example
Open staged standards development

Spec Consortia → Programs, Testbeds, Markets → Standards Bodies → Approved Standards

R&D Concepts → User Needs → Technical Trends

Specifications, Best Practice → New products, Pilot Programs, Application Profiles, Testbeds → Consensus, Consolidation, Conformance

Source: Fred Beshears, UC Berkeley

Software applications across organization boundaries

by
David G. Messerschmitt
University of California at Berkeley
High payoff for collaborative entrepreneurship in inter-organizational applications

• Commonality (not exclusivity) the core goal
• Achieves diversity
• Much higher value in standardized (but extendible) solutions
• Hooks into internal processes/systems

Conclusions

• Entrepreneurship is not research, but commercialization of existing technologies for new pre-qualified purposes
• Great opportunities for software applications meeting organizational and process needs, but special approaches are needed
Actionable ideas

- Embrace and encourage end-users as a major source of innovation, usually free for the taking
- Focus on new possibilities empowered by technology
- Value speed more than exclusivity
- Initiate and join collaborative entrepreneurship communities that include major end-user organizations