The enclosed forthcoming ISR commentary is simply the context of this study, not the working paper for the study to be presented.
1. Competition increasingly among platform ecosystems
   - Compete for developers & users
   - Innovation outside firm boundaries
   - New functionality never envisioned by platform designers

2. Platform success hinges on attracting & retaining developers
   … iPhone vs. Blackberry
   … Modules often non-portable, idiosyncratic

**Platform abandonment**: Post-adoption discontinuance by a module developer
Research Question

Software platform modularity ? Software platform abandonment?

**RQ1.1:** How does platform modularity influence systems integration costs?
1. **Technical** modularity
2. **Organizational** modularity (decision rights configuration)

**RQ1.2:** How do systems integration costs influence platform abandonment?
- Changes in *any* subsystem can trigger integration problems
  - *Ongoing* (re)-integration costs

Two Overarching Ideas Developed

![Diagram showing the relationship between technical modularity, systems integration costs, and platform abandonment.]

- Idea #1: Technical Modularity \(\rightarrow\) Systems Integration Costs \(\rightarrow\) Platform Abandonment
- Idea #2: Organizational Modularity
Summary of Key Findings

Modular Systems Theory

Design principle for complex systems

- Interacting subsystems ~ interdependent and independent
- Intentionally increase independence among subsystems
  - Allows subsystems to independently evolve yet interoperate
- A relative construct (Baldwin-Clark 2006)
  - Two modules of same platform can differ in their modularity
  - UoA = module
- Both technical and organizational property (Langlois 2002)
Technical & Organizational Modularity

Technical modularity: Degree to which a module is loosely-coupled to the platform codebase via standardized interfaces.

1. Loose-coupling... internal changes in a module don't ripple out
2. Interface standardization... stable, predefined module interfaces

Organizational modularity: Division of module-specific decision rights between platform owner and module developer.

- where does authority for technical decisions reside
  ... key element of org structure (Nault 1998)
- shared between module developer & platform owner

- This encompasses...
  - features
  - functionality
  - design
  - implementation
  - user interface

Systems Integration Costs

<table>
<thead>
<tr>
<th>IS</th>
<th>&quot;Integration&quot; difficulty</th>
<th>Goodhue et. al 2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org theory</td>
<td>&quot;Coordination costs&quot;</td>
<td>Gulati-Singh 1998</td>
</tr>
<tr>
<td>Industrial economics</td>
<td>&quot;Systems integration&quot;</td>
<td>Brusoni 2001</td>
</tr>
<tr>
<td>Computer science</td>
<td>&quot;Software composability&quot;</td>
<td>Messerschnitt-Szyperski</td>
</tr>
<tr>
<td>Product development</td>
<td>&quot;Complementary integration&quot;</td>
<td>Nambisan 2002</td>
</tr>
</tbody>
</table>

Two Types of Integration Costs

- Module-Platform Integration
- Cross Module Integration

Effort required by module developer to manage dependencies between module and
  - platform codebase (module-platform integration)
  - other modules (cross-module integration)
Research Model

[Diagram of research model with various factors and hypotheses]

Research Methodology

Survey of Mozilla Firefox extension developers
... UoA = module (extension)
... 6,500 modules (extensions)

Why the Firefox platform?
1. **No principal-agent** relationship (removes agency confounds)
2. **No money** involved (removes pricing confounds (Tirole 2003))
3. **Open source** code (removes IP confounds)
4. Prevents confounding effects from **cross-platform differences**

342 usable responses (~34%)
... Secondary data on several variables (from source code)
... All 13 categories; no “experimental” add-ons

Stable psychometrics (EFA; \( \alpha > .8 \))
... New scales for systems integration cost, abandonment

http://www.pragmatictheory.com/
Hypothesis Testing Approach

**H3, H4:** Mediated-moderation (Muller 2005 Psyc Bulletin procedure)

**H1, H2:** Sobel mediation test

Organizational Modularity

Platform Abandonment Intention

Rival Explanations (13 controls)

Controls for Rival Explanations

<table>
<thead>
<tr>
<th>1. Module technical characteristics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Module complexity</td>
<td>-.25*** (4.5)</td>
</tr>
<tr>
<td>Cross-module dependencies</td>
<td>-.03 (63)</td>
</tr>
<tr>
<td>Open source/proprietary dummy</td>
<td>.04 (1.1)</td>
</tr>
<tr>
<td>Extension type (12 dummies; 1 kept)</td>
<td>.1* (2.03)</td>
</tr>
<tr>
<td>Module evolution rate (ver / months)</td>
<td>-.19*** (4.3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Module market characteristics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Module subsumption risk</td>
<td>.02 (59)</td>
</tr>
<tr>
<td>User base (# downloads)</td>
<td>.02 (23)</td>
</tr>
<tr>
<td>Perceived extension portability</td>
<td>-.01 (-32)</td>
</tr>
<tr>
<td>User interest (# reviews)</td>
<td>-.1 (-77)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. Developer characteristics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer tenure (# years)</td>
<td>.19*** (4.2)</td>
</tr>
<tr>
<td>Developer experience (# extensions)</td>
<td>-.03 (-.82)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. Process characteristics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Outcome control</td>
<td>-.1* (2.1)</td>
</tr>
<tr>
<td>Platform decision rights centralization</td>
<td>-.02 (-.35)</td>
</tr>
</tbody>
</table>

§ = secondary data
### Results

Organizational Modularity...

- ...weakens loose-coupling’s benefits
- ...strengthens standardization’s benefits

- LC decreases SI costs more when DRs lean towards platform owners...
- ...than when they lean towards module developers

- Standardization decreases SI costs more when DRs lean towards module developers...
- ...than when they lean towards platform owners

Bootstrap of 1,000, 2,000, and 5,000 in PLS model
Theoretical Contributions

Software platform modularity \( \xrightarrow{\text{How?}} \) Software platform abandonment?

\( \xrightarrow{\text{Why?}} \)

1. **Why**: Systems integration costs
2. **Fit**: Technical modularity benefits contingent on org modularity
3. **Unexplored tensions** within tech modularity dimensions

Future Work

1. **How much modularity is just right?**
   - When do modularization costs overwhelm benefits?

2. **Designing IT portfolios for evolvability**
   - SOA and ongoing re-integration costs

3. **Software platforms as “ecosystems”** *forthcoming ISR*
   - Platform exclusivity, evolutionary dynamics, multi-homing
   - iPhone, Android, Ubuntu