**Class announcements**
- Database project due today
- Folio #3 due today
- **Team business paper due Wednesday**
- From Ch. 13, read 13.1, 13.2, and 13.4 for next time
- Wed. speakers: Samir Chaudry, Robert Culpi
- Final Exam Monday, 6/8, 7:30 – 10:30 PM
  - contact the teaching staff if you need special accommodations or have a conflict

**Today's Class**

- Announcements
- Complete comments on Chapter 8
- **Guest Speaker: Mark Cianca**
  - Chapter 9: Applications and the Organization
  - Project Management
- **Tracie Kemmerle**
  - Akamai Case Presentation
- Lecture on Akamai Case

**Chapter 8 continued....**

- Last lecture we covered:
  - Externalities
  - Network effects
  - Critical mass
  - Positive (and negative) feedback
  - Lock-in, switching, and standards
- Today we finish with:
  - Software economies
  - Intellectual property
  - Privacy policy and regulation

**Concepts from Chapter 8**

- Externality
- Network effects
- Critical mass
- Positive feedback
- Standardization
- Lock-in
- Switching costs
- Phases in supplying content (create, replicate, distribute)
- Software economies
- Sunk costs
- Intellectual property
- Copyright
- Patent
- Privacy policy
- Privacy regulation

**Economic Properties of Information**

- Phases in supply of content
  - Creation
    - Author (book, song, music)
    - Produce (movie, performance)
    - Collect (almanac, weather)
    - Some combination (news reporting)
  - Replication (for each user)
  - Distribution (network? Physical means?)
Software economies

- Creation is expensive
  - Sunk costs
  - Development is risky
- Replication is inexpensive
- Costs of maintenance and upgrade are **SIGNIFICANT!**
- Related to lock-in

Intellectual Property

- "Information or ideas with commercial value for which the government has granted exclusive property rights, including, copyright, patent, trademark, and trade secret"
- We’ll discuss:
  - Copyrights and Patents
  - Privacy Policy
  - Regulations

Copyrights and Patents

- Copyright grants an author exclusive control over original works of information content or software
- A patent is granted by the federal government for the right to make, use or sell an invention for a fixed period of time.
- More on each of these.....

Copyrights

- Prevents others from appropriating, replicating or displaying a work without permission of the copyright holder.
- Limitations: fair use - we can copy materials for use in our class
- Major implications for digital materials
  - What is digital fair use?
  - Digital Millenium Copyright Act (DMCA) used by RIAA and others to make charges of copyright violation called infringement.

Copyrighting Software

- Software license stipulates the terms and conditions under which copyrighted software is used
  - Freeware - copyright, but is given away
  - Shareware - copyright, but seeks voluntary payment
  - Copyleft - limits derivative works to those distributed free of charge
- Infringement - violation of copyright
- Piracy - selling large volumes of copyrighted material

Patents

- Encourage research and development due to exclusive rights (for a limited time)
- Patents in computing and communications may be cross-licensed for mutual benefit (examples include Microsoft and Lexmark, Microsoft and HP)
- Software patent controversy - when is software an innovation? When not?
Privacy Policy

- The issue of personally identifiable information (PII)
  - What is PII?
    - Name, data of birth, SSN, credit card info, drivers license, medical or health information
  - Issues:
    - Identity theft
    - Fraud
    - Discrimination
  - Secure transmission (encryption)
  - Concerns when dealing with 3rd party outsourcing
    - Security of data
    - Persistence of data

Regulations

- A selection of privacy laws:
  - FERPA Family Educational Rights and Privacy Act - 1974 (updated many times through 2009)
  - HIPAA - Health Insurance Portability and Accountability Act - 1996
  - Fair Credit Reporting Act (FCRA) - 1997
  - Gramm-Leach-Bliley Act - 1999

Chapter 9

Applications and the Organization

Build vs. Buy?

Purchase off the Shelf
- less time and cost
- benefits of using a "standard" solution
- support available
- must mold org to app
- no potential for competitive advantage

Outsource
- developers not as familiar with org as you
- more opportunity for customizing than off the shelf
- contractor may share knowledge with competitors
- contractor may have too much bargaining power

Make
- most customizable of 3
- easier iteration between conceptualization and development needed
- most risky
- org may lack competency to do it

Application Lifecycle

- It is important to think beyond acquiring an application
  - How do we come with the idea?
  - How do we architect it.
  - How do we implement?
  - How do we extend and maintain it?
  - For this reason, the software engineering community came up with:
    - Application Lifecycle Model

Application Lifecycle

Stages:
1. Conceptualization
2. Analysis
3. Architecture Design
4. Development Evolution
5. Testing and Evaluation
6. Deployment
7. Operations, Maintenance, and Upgrade
1) Conceptualization

What is the vision?
- What are the objectives?
- What is the business case? What problem or opportunity does it address?
- EXAMPLE: HHC to inform flight attendants which passengers are low and high value.

- Business Case:
  - Increase repeat business from high value customers.

2) Analysis

- Describe what the application will do.
- Enough info to allow “stakeholders” to review idea
- Don’t make highly detailed specifications
- Describe scenarios in which it is used
  - (Use Cases)

2) Analysis -- Example

- Example: Reporting Function
  - FA wants to report that passenger in 13F is bad.
  - FA clicks “report pass.” button followed by 13F
  - HHC finds from its data that Joe Schmoe is in 13F
  - When HHC is in radio range of WiFi AP, HHC tells server that Joe Schmoe is bad.

Application Lifecycle

Stages:
1. Conceptualization
2. Analysis
3. Architecture Design
4. Development Evolution
5. Testing and Evaluation
6. Deployment
7. Operations, Maintenance, and Upgrade
3) Architecture Design
- Decompose the application into subsystems
  - Hardware, software
  - Try use commercial off the shelf subsystems
  - Try to use standard infrastructure layers
    - Operating system, network, middleware, etc.
    - More current example would be Identity Management infrastructure

3) Architecture Continued
- Define the functionality, interaction and interfaces of subsystems
- While doing this, consider
  - Scalability
    - How easily can we increase the number of users and maintain performance?
  - Extensibility
    - How easily can we add new features in the future?
  - Administration
    - How much work will it take by humans to keep this running properly?
    - (Remember Sun thin vs fat client discussion)

Application Lifecycle
Stages:
1. Conceptualization
2. Analysis
3. Architecture Design
4. Development Evolution
5. Testing and Evaluation
6. Deployment
7. Operations, Maintenance, and Upgrade

4) Development Evolution
- Develop the details
  - Develop/program custom subsystems
  - Have contractor build outsourced pieces
  - Put together with off-the-shelf components
- Incremental
  - Start with simplest implementation and get it working
  - Later add more features.

Application Lifecycle
Stages:
1. Conceptualization
2. Analysis
3. Architecture Design
4. Development Evolution
5. Testing and Evaluation
6. Deployment
7. Operations, Maintenance, and Upgrade

5) Testing
- A must!
  - Functional Specs form basis for test specs
- If architected well, we can test subsystems independently.
- Alpha test - offline test of prototype
- Beta test - test in intended environment with cooperative users
  - Example - give HHC to initial group of FA's
Application Lifecycle

Stages:
1. Conceptualization
2. Analysis
3. Architecture Design
4. Development Evolution
5. Testing and Evaluation
6. Deployment
7. Operations, Maintenance, and Upgrade

6) Deployment
- Convert from previous processes if necessary
  - Example: CISCO ERP (all at once)
  - Or, you could do incrementally (how?)
- Train users
  - Example: Frito-Lay HHC
- Data importation
  - (if necessary)

Application Lifecycle

Stages:
1. Conceptualization
2. Analysis
3. Architecture Design
4. Development Evolution
5. Testing and Evaluation
6. Deployment
7. Operations, Maintenance, and Upgrade

7) Operations, Maintenance, Upgrade
- Maintain Security
- Repair Problems, either in code directly or to address changes to other subsystems (OS upgrades, DB upgrades, etc.)
- Correct performance short comings (Cisco ERP)
- Add features

Application Lifecycle Model (ALM)

concluding remarks
- ALM rarely followed precisely
- Many times projects loop between stages
- ALM followed more closely in larger companies
- Alternative:
  - Rapid Iterative Prototyping
    - (Cisco did some of this in the ERP case.)

Akamai Case

Tracie Kemmerle
Main Ideas

- Background of Akamai
- Problems of the www "The World Wide Wait"
- Version 1.0: FreeFlow
- Version 2.0: EdgeSuite
- Version 3.0: EdgeComputing

Background of Akamai

- On Sept. 11, 2001, Akamai cofounder and chief tech officer Danny Lewin died aboard one of the planes that stuck the World Trade Center.
- The U.S. economy was stuck in a deep recession that had hit tech and advertising markets particularly hard.
- Akamai's managers were rolling out a new service that could fundamentally reposition the company.
- They launched its EdgeSuite service, which moved the company beyond its traditional role of content delivery into the assembly, presentation, and delivery of data from the Internet's edges.

The World Wide Wait

- As the Internet expanded it grew congested.
- Topology: The Internet's reliability and performance problems were related to its "network of networks" architecture.
- Bottlenecks: Users sometimes experienced slow transaction processing at e-commerce sites and were unable to access Web sites that had been inundated with user "hits".
- Content Delivery Networks- A solution to Bottlenecks

FreeFlow

- This was Akamai's first service, introduced commercially in April 1999.
- To use FreeFlow, a content provider first identified and tagged the objects that it wanted to serve over the Akamai network.
- Targeted FreeFlow at the 500 most heavily trafficked Web sites and sites run by the 2,000 largest global corporations.
- By the 3rd quarter of 2001, Akamai had over 13,000 servers in 954 networks across 63 countries.
EdgeSuite

- Akamai’s competitive landscape changed considerably during 2001.
- A new service called EdgeSuite was created, which employed Edge Side Includes (ESI), a markup language used to accelerate the dynamic assembly and delivery of Web-based applications at the Internet’s edges.
- Offered a powerful value proposition, both for Akamai’s traditional customers with content-rich Web sites and for enterprise customers that increasingly relied on the Internet to distribute information and provide Web-based applications for customers, channel partners, suppliers, and remote employees.

EdgeComputing

- In May 2003, Akamai and IBM announced that EdgeComputing customers could run Java applications created using IBM’s WebSphere software development tools from Akamai’s edge servers.
- By early 2004, Akamai had about a dozen customers running Java applications.
- The computer peripheral manufacturer Logitech, for ex, held a contest on its Web site that gave away thousands of prizes in a matter of hours.
- Battle between Java and Microsoft’s .NET
- We’ll become the “Intel Inside” Web services.

Summary of How Article Relates to ISM 50 Concepts

- Networks Effects
  When the value of a product or service to an individual consumer depends on the number of other consumers adopting it.
  Everyone wants the fastest serving Internet
- Value of Locked-in Consumer
  The supplier has an opportunity to sell upgrades, aftermarket products, etc. at higher-than-competitive prices to a locked-in customer.
  When creating new applications they were able to get already existing customers to sign up for the new application.

OSI Layers

- Application
  Internet Explorer, Outlook Email, Real Player, ...
- Presentation
- Session
- Transport
- Network
- Link
- Physical

Modulation Schemes: QAM, OFDM, etc…

Internet Protocol (IP), …
TCP, UDP

Akamai Case–Problem: Internet Bottlenecks

- **First Mile** (Server Capacity) - 70% of website performance problems according to one study
- **Backbone** - Plentiful, but some shortage within metropolitan areas
- **Peering** - Exchange of traffic between NSPs
- **Last Mile to home**
  - 56 K modems are slow
  - Shared LAN limitations

Web Caching at a proxy server at an ISP

![Diagram of web caching at a proxy server at an ISP](image)
Solutions

- **Expand Bandwidth**
  - Being done.
- **Mirroring web sites**
  - Put exact copy of same web page to multiple servers
  - Tricky to duplicate content
- **Caching**
  - Problem: Stale Content
  - Problem: Hard to count "click throughs"
- **Content Distribution Networks (CDN)...**

CDN: Akamai FreeFlow

- Thousands of servers at NAPs or IAP/POPs
  - Each server holds data for many customers. Akamai CDN software ensures that distributed data elements are synchronized with customers’ origin servers.
  - Which Akamai server does an end user see?
    - Akamai algorithms route end user requests to the best server based on network congestion analysis.
    - Secondary benefit: usage analysis
      - Akamai software pushes end user usage data back to Akamai’s customers for analysis (like Google Analytics)
    - Review: who are Akamai’s customers?
      - Ans: large corporations who wanted to boost performance (thus their end user customers’ experience) on their corporate web sites. Examples: Apple, Yahoo!, ...

Freeflow

- Deployed in 1999
- **Akamai Infrastructure**
  - 13000 servers in 954 networks by 2001
- **Customers**
  - Large Commercial Websites
- **Revenue model** - $2000 per mbps served
  - (For comparison, normal Internet access cost 500 mbps at time)

2000 Financials

- $196 Million Loss (before special charges)
- $90 million revenue
- 320 gross margin, after deducting
  - server depreciation
  - payments to network partners
  - Data center space
  - But, most expenses of shouldn’t grow at same rate as number of customers, so margin should improve.
- $201.5 million SG&A
  - Selling general and administrative
  - (largely sales force cost)
  - Again, this might not grow at same rate as the number of customers.
- $40 million R&D

Competition

- Hosting firms (substitute)
  - Exodus
- Other CDNs
  - Sandpiper, Adero, Mirror Image
- **Content Alliances**
  - Akamai’s competitors banded together to share networks
2001 Market Changes

Bad
- Dot-coms bust
- Customers leave
  - "churn rate goes to 22% per quarter"

Good
- Hosting firms go bust (Exodus)
- Some CDN competitors go bust
- Competing CDN alliances mired in problems

EdgeSuite

- Assemble dynamic pages at edges rather than just serve heavy objects
- Value proposition
  - Performance improvement
  - Cost and complexity reduction
  - Scalability
  - Security
- Pricing - higher than old service
- Soon edge suite dominated revenue

Technology

Dynamic CDN technology: ESI (edge side includes)

Develop as open standard why?

Akamai not big and credible enough to force a de-facto standard on market

Marketing

- Difference in selling old vs new products:
  - Old product
    - Geared toward speeding up websites
    - Revenues of their clients depended on speed
    - Easier to get sale
  - New Product
    - Simplify company IT function
    - Cost vs. revenue center
    - Harder sell. More data driven.
    - Consequently new product needs more professional sales force
- Channels?
  - Distribution Partners (IBM) credibility
  - Direct Sales Force too

Recent Performance

<table>
<thead>
<tr>
<th></th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>$210,981</td>
<td>$241,133</td>
<td>$244,976</td>
<td>$263,334</td>
<td>$286,741</td>
</tr>
<tr>
<td>Total cost and operating expenses</td>
<td>$101,348</td>
<td>$123,175</td>
<td>$137,690</td>
<td>$217,088</td>
<td>$249,103</td>
</tr>
<tr>
<td>Net income (loss)</td>
<td>$109,633</td>
<td>$117,958</td>
<td>$107,286</td>
<td>$46,246</td>
<td>$47,638</td>
</tr>
<tr>
<td>Net income from discontinued operations</td>
<td>$109,633</td>
<td>$117,958</td>
<td>$107,286</td>
<td>$46,246</td>
<td>$47,638</td>
</tr>
<tr>
<td>(in thousands, except share data)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Akamai’s Recent Products

Some current products, from Akamai’s web site:

**Advertising decision solutions** — better ad targeting from click-through analysis that leverages customer behavior patterns from Akamai’s large network of servers

**Application performance solutions** — the ADN networks we talked about. Now called Web Application Accelerator, IP Application Accelerator, etc.

**Digital asset solutions** — the CDN networks we talked about. Now called Akamai Media Delivery, Dynamic Site Accelerator, etc.

2009 content / application delivery network competitors

Web article by Mircea Goia, Sept. 10, 2008