Legacy Student Information System: Replacement or Enhancement?

Leah Burstein
Sr. PL/SQL Programmer, Loyola Law School
Advisor: Dr. Bohdan Oppenheim
SELP Capstone Project, 2015
Agenda

- Background
- Problem Definition
- Alternatives
- SE Aspects (Architecture, Requirements, Risk)
- PM Aspects (Cost, Schedule)
- Req. Allocation Example; Lean/Agile thoughts
- Other Institutions Choices
- Ethical Aspects
- Final Choice
- Conclusion; Reflections & Limitations
Background

- LMU and LLS: different business practices and different SIS
- Both Banner at LMU and Exeter at LLS – since 1998/99
- Exeter de-supported by its vendor in 2002
- LLS IT support (3-4 people, just 1 fulltime):
  - DB backups;
  - user accounts administration;
  - troubleshooting of a base-line product;
  - adding new features (front-end and back-end customizations)
- After 10 years: “outdated & not fully meeting users needs”
- Users worried about not being supported by a major vendor
- 2011/2012 – RFI for COTS
- Ellucian’s Banner is chosen out of 4; merging DB with LMU
- Project is underway at LLS/LMU

Attachment:
Pros and Cons of COTS
Problem Definition

- "Current system outdated & not meeting users needs"
- Law school graduates cannot find legal jobs => lowest number of applicants in 4 decades (all law schools in the US)
- Keep high admission standard => reduce incoming class size by 10%
- Affects long-term budget strategy

It is expected that the SIS chosen by LLS will not only be based on the most modern technology, satisfying different departments' user requirements (in particular, by addressing the current system's gap analysis) but will also minimize the to-be system's implementation and maintenance cost.
Pros & Cons with respect to project goals (eliminating gaps & minimizing cost) to be discussed.
SA: SIS Internal Customers

What people need?

**Students**
- Register for Classes
- Check Schedule
- View Exam Grades
- See Transcript
- Check Degree Audit
- Apply for Loans
- Pay Tuition

**Applicants**
- Apply for Admission
- Check Application Status
- Pay Seat Deposit

**Faculty**
- Print Rosters
- Post Grades

**Staff**
- Admissions
  - Review applications, post decisions, “move” admitted applicants to Registrar...
- Registrar
  - Handle student registration records, publish grades, run transcripts, run DA...
- Financial Aid
  - Handle student loans and scholarships, work-study...
- Student Accounts
  - Handle student payments and charges

Oracle Forms Interface

Web Interface: SWS/FWS (PROWL/SSB)
SA: Project Scope Interrogatives
(Who, What, Where, When, Why, How)

Who?
- Internal Customers (previous slide)
- External Organizations (LMU, DoE, NSC...)

What?
- See previous slide (grades, loans, payments)

Where?
- Loyola Law School (Downtown LA) (Alt. A)
- LLS and LMU campuses (Alt. B)
- Alt. C: possible entirely at LLS or shared between LLS and LMU for database administration part

Why?
- To Address Gap Analysis
- To Modernize Technology
- To Minimize Maintenance Cost
Alt. A: To minimize transition cost, schedule & effort; Preserve independence of campuses
Alt. B: To join operations between 2 campuses and achieve LLS cost savings (costs shifted to LMU)
Alt. C: To avoid conflicts related to objective (e.g. academic schedules) and subjective (business culture) differences between campuses. No add. cost

How?
Alt. A:
- Upgrade database to the latest release (backend); Fix broken front-end; Get “As-Is” architecture stabilized
- Address business gaps by gradually adding new pieces of functionality, leveraging newer technology capabilities
Alt. B: Migrate LLS data to LMU database; make sure nothing breaks for LMU; utilize native Banner functionality and added by LMU functionality; add missing pieces
Alt. C: Adopt Banner for LLS but make it a separate database

When?
Alt. A: initial upgrade < 1 year, followed by ongoing development
Alt. B: initial plan: 1.5 years for everything.
Reality: 2.5 years for initial rollout, still a long way for other steps (1 more year?).
Alt. C: Plan’s B initial calculation of 1.5 years seems feasible.
SA: Exeter Physical and Logical Structure

Student Marketing System (Admissions)

"Common" module (Person entity details, Addresses, Phones, ...)

Career Management System (never implemented)

Student Services System (Registrar)

Student Aid System (Financial Aid)

Student Billing System (Student Accounts)

Student Housing System (not purchased)

"Make Aware"
Exeter System Decomposition (BDD)

Alignment between departmental functions & system functions!

Loyola Law School

- **System** LMU SIS "Banner"
  - exchanges data with
- **System** Other On-Campus Systems
  - gives donations to
  - 1:* exchanges data with
- **System** Other External Systems

- **Actor** Applicant
  - 1:* has interest in 1
- **Actor** Professor
  - 1:* submits grades to 1
- **Actor** Alumni
- **Actor** IT Dept

- **System** LSAC (Law School Admissions Council)
  - flow
- **System** COD (Common Origination and Disbursement) website

- **Actor** Other Law School
  - competes with 1
- **Actor** Department of Education
  - regulates 1
- **Actor** Exeter SIS
  - data resides in 1
  - exchanges data with
- **Actor** Student
- **Actor** Professor
- **Actor** Administrative Staff
- **Actor** Financial Aid Dept
  - 1 works with 1
- **Actor** Office of the Registrar
- **Actor** Student Accounts Dept
- **Actor** Student Billing System
- **Actor** Student Aid System
- **Actor** Student Marketing System
  - 1 works with 1
- **Actor** Office of the Registrar
- **Actor** FINANCIAL AID DEPT
  - 1 works with 1

**Color Legend**
- Our System
- Actors
- External Systems
Appropriate aggregation of functions?

- Alignment between the structures of the SIS and the organization – satisfies Rechtin’s heuristic!
- "In partitioning a system into subsystems, choose a configuration with minimal communications between the subsystems.” (E. Rechtin)
- => Easy to manage security access to different Exeter modules – by department.
Appropriate aggregation of functions?

- Security access in Banner: not flexible, hard to tailor the level of access according to departmental needs.
- Living in the same database (alternative B) requires LLS users to change their business practices + "waste of communication".
Banner by Ellucian – is it worth it?

- Banner 8 – still Oracle Forms & Reports
- 1400 forms (Exeter – 400), almost 100K DB objects (Exeter – 6K) ("-" for manageability)
- Banner 9 – Groovy/Grails (+/-)
- Developed in early 80's; on the back-end over the years - Cobol, Pro*C, Java ("-" for supportability)
- Tons of documentation, live support (+)
- A lot of trainings (+)
- Prospect: Degree Works (+) (Degree Audit - more intuitive for students & advisors)
To-Be System Requirements

Top-level

- SIS shall eliminate identified current system gaps
  - SIS shall be integrated (fewer disparate systems)
- SIS shall be modernized
- Cost of implementation & maintenance shall not exceed allocated budget
- Time of implementation shall not exceed 2 years

Some specifics

- All existing functionality shall be preserved in a to-be state
  - All functions and interfaces should be tested/fixed after upgrade (A)
  - All functions and interfaces should be rewritten in Banner (B,C)
- Financial Aid Upgrade shall take “minimal amount of effort”
  - Development and testing shall not exceed 1 month (A)
  - Applying delivered patches shall not exceed $X$ hours (B,C)

Backup Slide 17: Requirements Diagram
Use Cases

A

- LLS IT
  - Upgrade Database layer to Oracle 11g
  - PITSS (contractors)
    - Upgrade Front-end (Forms) component
  - LLS IT
    - Develop New Functionality
    - Ongoing System Maintenance

B

- LLS IT
  - LMU IT
    - Eucian Consultants
    - Migrate Data to LMU's Banner Database
    - LMU IT
    - Verify Existing LMU Functionality
  - LLS IT
    - LMU IT
      - Evisions Consultants
      - Restore LLS functionality under new system
      - LMU IT
        - Restore LLS functionality
        - Develop new LLS functionality
        - Ongoing System Maintenance

C

- LLS IT
  - LMU IT
    - Eucian Consultants
    - Migrate LLS Data to a separate Banner database hosted by LMU
    - LMU IT
      - Evisions Consultants
      - Restore LLS functionality under new system
      - LLS IT
        - Develop new LLS functionality
        - Ongoing System Maintenance
## Risk Summary Log

(Prepared for SELP-594 PM Course)

### Attachment: Risk Portfolio

**Risk Weight = Probability \times Impact**

(Impact as % of project cost)

(Non-recurring Project Cost ~ $450K – see Slide 18)

<table>
<thead>
<tr>
<th>STATUS</th>
<th>ID #</th>
<th>TITLE</th>
<th>TYPE</th>
<th>ASSMT.</th>
<th>MITIGATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>A</td>
<td>Loss of key players (contractors)</td>
<td>External Predictable</td>
<td>$0.45K (1%x10%)</td>
<td>Keep contacts for possible replacement</td>
</tr>
<tr>
<td>Red</td>
<td>B</td>
<td>Heavy workload for project team members</td>
<td>Internal Non-Technical</td>
<td>$11.25K (50%x5%)</td>
<td>Hire temps to handle routine tasks</td>
</tr>
<tr>
<td>Green</td>
<td>C</td>
<td>Loss of key players (internal team)</td>
<td>Internal Non-Technical</td>
<td>$3.6K (1%x80%)</td>
<td>Raise morale; Bonuses; Hire more people with redundant skills</td>
</tr>
<tr>
<td>Green</td>
<td>D</td>
<td>Cost overruns</td>
<td>Internal Non-Technical</td>
<td>$1.125K (5%x5%)</td>
<td>Make customizations gradually</td>
</tr>
<tr>
<td>Green</td>
<td>E</td>
<td>Resistance to change</td>
<td>Internal Non-Technical</td>
<td>$3.375K (15%x5%)</td>
<td>Raise morale (+bonuses); underscore benefits</td>
</tr>
<tr>
<td>Red</td>
<td>F</td>
<td>Technical complexity</td>
<td>Internal Technical</td>
<td>$22.5K (50%x10%)</td>
<td>Proper design, testing and documentation</td>
</tr>
<tr>
<td>Red</td>
<td>G</td>
<td>Scope creep</td>
<td>Internal Technical</td>
<td>$40.5K (30%x30%)</td>
<td>Proper change management process</td>
</tr>
<tr>
<td>Yellow</td>
<td>H</td>
<td>Schedule slip for new features implementation</td>
<td>Internal Technical</td>
<td>$18K (20%x20%)</td>
<td>Increase number of technical staff</td>
</tr>
<tr>
<td>Green</td>
<td>I</td>
<td>Poor technical documentation</td>
<td>Internal Technical</td>
<td>$2.25K (10%x5%)</td>
<td>Don’t start new mini-project before wrapping up a previous one properly.</td>
</tr>
<tr>
<td>Red</td>
<td>J</td>
<td>Regulatory changes impact existing systems</td>
<td>Legal</td>
<td>$18K (80%x2%)</td>
<td>Allocate time/people resources to implement mandatory changes</td>
</tr>
<tr>
<td>Green</td>
<td>K</td>
<td>Security breach</td>
<td>Legal</td>
<td>$4.5K (0.01%x100%)</td>
<td>Set data transfer standards; implement secure protocols</td>
</tr>
</tbody>
</table>

**Status Legend:** Red = Critical, Yellow = Watch, Green = Managed
Student Information System Upgrade

Risk Matrix

(Prepared for SELP-594 PM Course)

Attachment: Risk Portfolio

Red = Critical
Yellow = Watch
Green = Managed
Additional Risks for Alternative B
(or rather, some known facts about Alternative B)

- Each decision must be reviewed by both campuses
- Access issues dictated by LMU policies
- Longer implementation path for customizations
- No real Risk Mitigation for the current path. 
  *Everything identified as risk – happened!*

(Loss of key players, Cost overruns, Schedule slips)
- Can be avoided with Alternative C
### PM Aspects

<table>
<thead>
<tr>
<th>Conceptual Phase</th>
<th>Feasibility Phase</th>
<th>Detail Planning Phase</th>
<th>Implementation Phase</th>
<th>Termination Phase</th>
<th>Program Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current System Gap Analysis</td>
<td>LLS IT Response</td>
<td>Check Availability of Involved Consultants</td>
<td>Upgrade Hardware</td>
<td>Write User Guides for New Functionality</td>
<td>Project Conceptual Description</td>
</tr>
<tr>
<td>Preliminary selection of COTS vendors</td>
<td>LLS IT Demos</td>
<td>Detailed Project Implementation Plan</td>
<td>Install Operating System</td>
<td>Organize User Training</td>
<td>Project Scope Document</td>
</tr>
<tr>
<td>RFI</td>
<td>Preliminary Cost Estimate for LLS In-House Upgrade Solution</td>
<td>Refined Cost and Schedule Estimates</td>
<td>Upgrade Database from 8i to 11g</td>
<td>Complete Technical Documentation</td>
<td>Project Requirements Document</td>
</tr>
<tr>
<td>Vendors' Demos</td>
<td>Preliminary Time Line for LLS In-House Upgrade Solution</td>
<td>Sign Official Contracts with all the consultants</td>
<td>Mass Forms Conversion</td>
<td>Develop System Backup Schedule</td>
<td>Work Breakdown Structure</td>
</tr>
<tr>
<td></td>
<td>PITSS Proof of Concept Workshop</td>
<td></td>
<td>Forms Testing</td>
<td>Develop Policy for Handling New Requests</td>
<td>Statement(s) of Work</td>
</tr>
<tr>
<td></td>
<td>Solution Selection</td>
<td></td>
<td>Fix &quot;broken&quot; forms</td>
<td>Establish Procedures for Production Deployment</td>
<td>Master Schedule</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Overall System Testing</td>
<td>Celebrate Project Completion</td>
<td>Risk Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fix identified bugs/issues</td>
<td></td>
<td>Issue Log</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sign-Off for Production</td>
<td></td>
<td>Cost Estimate/Actual Cost Reports</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Database Upgrade in Production</td>
<td></td>
<td>Communications Plan</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Milestone: Upgraded System Launch</td>
<td></td>
<td>Change Management Plan</td>
</tr>
</tbody>
</table>

### Alt. A – WBS & Stage I Cost

(Prepared for SELP-594 PM Course)

<table>
<thead>
<tr>
<th>Conceptual Phase</th>
<th>Feasibility Phase</th>
<th>Detail Planning Phase</th>
<th>Implementation Phase</th>
<th>Termination Phase</th>
<th>Program Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>$63,000</td>
<td>$36,000</td>
<td>$13,700</td>
<td>$253,800</td>
<td>$19,500</td>
<td>$66,000</td>
</tr>
</tbody>
</table>

### Backup Slides 1-2:

**MoE**

\[ F \text{ times} \]

### Attachment: SOW

### Attachment: Non-Recurring Cost Estimate

**Total Cost: \$452,000**
### Alternative A – Schedule (Stage I)

(Prepared for SELP-594 PM Course)

<table>
<thead>
<tr>
<th>Task</th>
<th>Scope: Master Schedule Phases</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Feasibility and Preliminary Planning Phase</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Detail Planning Phase</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Implementation Phase</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Conversion / Sustaining / Termination Phase</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Milestone:** Preferred COTS Vendor Selection

**Milestone:** ICT Solution Selection

**Milestone:** Critical Design Review

**Milestone:** End of Customizations

**Milestone:** Closeout Review

---

### Task: Implementation Phase

<table>
<thead>
<tr>
<th>Task</th>
<th>Scope: Implementation Phase</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2 Install Operating System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3 Upgrade Database from 8i to 11g</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.4 Perform Mass Forms Conversion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5 Perform Forms Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.6 Fix “broken” forms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.7 Perform Overall System Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.8 Fix identified bugs/issues</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9 Prepare to wrap-up the first stage of the system upgrade</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Milestone:** Production Readiness Review for Upgrade Stage

| | | |
| | | |

**Milestone:** Upgraded System Launch

| | | |
| | | |

**Milestone:** End of Identified Customizations

---

**Phase** | **Duration**
---|---
Conceptual | 7 m
Feasibility | 3 m
Execution | 14 m (overlaps w/det. planning)
Just Upgrade | 4 months!

---

Attachment: Schedule

---

20
Support Phase of SDLC: Financial Aid Regulatory Requirements Example
(The biggest difference between life-cycle support in Exeter and Banner)

**Exeter**
- Every year, 1 month spent analyzing, implementing, testing and deploying changes required by external organizations:
  - back-end changes in the static Exeter tables (e.g. state and other tax allowances, income and asset protections allowances (IPA/APA), used in calculation of the Expected Family Contribution (EFC)), changes of formatting in Institutional Student Information Record (ISIR) data file generated from Free Application for Federal Student Aid (FAFSA)
  - changes in Forms to display the new ISIR fields
  - changes in XML export/import interfaces that exchange data with the COD website

**Banner**
- No additional programming is required
- DBA applies some patches supplied by Ellucian (approximately 5 times a year)
- Financial Aid department users still should test the changes – because history shows that Ellucian patches cannot be trusted blindly.
- Less work, but sometimes more stress (when Ellucian is running late fixing bugs reported by clients)

Backup Slide 3: BDD
## Alternative A – Recurring Cost (Year 1 Post-Upgrade)

<table>
<thead>
<tr>
<th>Recurring Cost Elements</th>
<th>Cost</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yearly Upgrade of Financial Aid Module</td>
<td>$27,500</td>
<td></td>
</tr>
<tr>
<td>Functional users: Reading regulations, identifying areas of change</td>
<td>$2,000</td>
<td>20 hrs every November</td>
</tr>
<tr>
<td>Functional users: Preparing tailored requirements for IT</td>
<td>$1,000</td>
<td>10 hrs every early December</td>
</tr>
<tr>
<td>IT: Initial programming according to the request</td>
<td>$10,000</td>
<td>100 hrs distributed over 2 months</td>
</tr>
<tr>
<td>Functional users: Testing</td>
<td>$10,000</td>
<td>100 hrs in January-February</td>
</tr>
<tr>
<td>IT: Fixes</td>
<td>$4,000</td>
<td>40 hrs in January-February</td>
</tr>
<tr>
<td>IT: Putting Changes to Production</td>
<td>$500</td>
<td>5 hrs in early March</td>
</tr>
<tr>
<td>Performing Scheduled System Backups &amp; Maintenance</td>
<td>$255,800</td>
<td></td>
</tr>
<tr>
<td>Backing up database servers</td>
<td>$24,000</td>
<td>5 hrs every weekend=&gt;20 hrs every month</td>
</tr>
<tr>
<td>Backing up other servers</td>
<td>$24,000</td>
<td>5 hrs every weekend=&gt;20 hrs every month</td>
</tr>
<tr>
<td>Analyzing System Logs</td>
<td>$6,000</td>
<td>5 hrs every month</td>
</tr>
<tr>
<td>Troubleshooting Emergency Situations</td>
<td>$1,800</td>
<td>18 hrs distributed over a year</td>
</tr>
<tr>
<td>Other</td>
<td>$200,000</td>
<td>Various licenses $ 200K / yr (paid in advance)</td>
</tr>
<tr>
<td>Regular User Support</td>
<td>$21,400</td>
<td></td>
</tr>
<tr>
<td>Setting up new user accounts/termination</td>
<td>$6,000</td>
<td>5 hrs every month</td>
</tr>
<tr>
<td>Identity Management-related maintenance</td>
<td>$9,600</td>
<td>8 hrs every month</td>
</tr>
<tr>
<td>Releasing database locks</td>
<td>$600</td>
<td>6 hrs distributed over a year</td>
</tr>
<tr>
<td>Extracting on demand data reports for various departments</td>
<td>$4,000</td>
<td>40 hrs distributed over a year</td>
</tr>
<tr>
<td>Providing on demand user training</td>
<td>$1,200</td>
<td>12 hrs distributed over a year</td>
</tr>
</tbody>
</table>

Total Cost Y1: $304,700
Y1-Y10: $3,047,000
Stage 1: $452,000
Grand Total: $3,499,000

Attachment: Recurring Cost Estimate

Grand Total Alt. A: $3.5 M
Cost & Schedule for Alternative B
(currently underway at LLS/LMU)

- Schedule: 1.5 years planned, 2.5 years happened
- Cost: Allocated budget for transition stage = $1 M
  
  "In less than 1/3 time planned spent close to 1/2 of the budget"

  1/3 time = 0.5 years => $0.5 M; then in 2.5 years - $2.5 M

  - Just first step, including data migration
  - Lost LLS functionality is not fully restored yet (+1 year?)

- Recurring Cost:
  
  Banner support = 848 K/Year (?) => $8.5 M in 10 years

- Grand Total Alt. B: $11 M
- Shifting Costs to LMU...
Cost: Alternative C
(originally promised)

- Schedule: 1.5 years planned – feasible (see next slide)
- Non-Recurring Cost: Allocated budget for transition stage – feasible (see next slide)
- Recurring Cost:
  - Banner support - still can use LMU license (0 cost)
  - Additional 1 DBA position: ~ 100K a year – enough for 15 years post-migration, paid by $1.5M saved during original implementation period. (Previous slide: $1M planned but $2.5M happened)
- No need to coordinate each step => less frustration
- Preserves LLS approach to business practices
  (if Banner allows it...)
Alt. B, C Stage I: How to do it better?

- Functional Training (not me)
- Technical Training (me)
- Data Mapping (func. cons. absent)
- Data Conversion (5-7 rounds???)
- What I would do for Alt. C?
  - Enough training first, while mapping
  - Conversion – 2 rounds and without Ellucian!
Example of Requirements Allocation for Alt. A + Lean/Agile Aspects

• Use Case: **Note Takers for Accommodated Students**

Backup Slides 4-13: **Note Takers**

• Incorporated a "shadow system" into Exeter: got rid of 2 Access DB, connected 3 different environments via APEX

• Satisfies 2 top-level requirements: "SIS shall be integrated (fewer disparate systems)" and "System modernization"

• Prototype (2012) written in 2 weeks => Agile!

• Streamlines flow, cuts waste, imperfections visible => Lean!

• Thoughts about Agile vs. Waterfall and Agile vs. Lean

Backup Slides 14-16: **Agile/Lean**
Other Institutions’ Choices

- CalTech – Exeter
- MIT – Customized Exeter

The approach for the Next Gen Student System is to:
- evolve our student system by incrementally adding critical business functionality
- assure long-term sustainability rather than replace the entire system
- enhance the user experience for faculty, students and staff
- stabilize the technical infrastructure.

- UChicago, Georgetown – Banner
- Smaller colleges: Empower, Campus Management’s CampusVue, Three Rivers’ CAMS (Microsoft-based), some of them not customizable; Jenzabar (MSSQL too)
- Kuali Student – open source (Boston College, U. of WA)
Ethical Aspects

- Banner born in early 80’s; “old” back-end architecture; only front-end is redesigned because “nobody sees the back-end.”
- No motivation? 1,400 customers, 50 law schools – no happy reference.
- Ellucian consultants: presented as experts, most efficient way of handling initial data migration; but in reality ...
- LLS: obscurity (C->B); merging departments; jobs terminated.
- Internal customers suffered (students, faculty and - most of all - staff).
Let’s stay positive: What else about Banner is good?

• Regulatory Fin. Aid Upgrades Supplied by Vendor
• Ellucian Support for Daily Troubleshooting
• Degree Works
• TouchNet Integration for Billing and Payments
  – But even here - difficulties with terms separation...
  – Bonus but not “identified gap”
Final Choice

- Depends on the strategic goal!
- From Engineering and Cost standpoint – Alt. A.
- If the main goal is to join intercampus operations then Alt. B.
- If you want to preserve org. identity you would pick A (like MIT).
- If you are in financial trouble – don’t start big projects. Survive with the known system until better times. Other alternatives will emerge.

Score Card:
Grades from 1 (lowest) to 3 (highest), broken to Transition/Maintenance
Risk for A-maint currently High (and “grade” is low), but can be mitigated

<table>
<thead>
<tr>
<th>Aspect</th>
<th>A (trans/maint)</th>
<th>B (trans/maint)</th>
<th>C (trans/maint)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>3/3</td>
<td>1/2</td>
<td>2/1</td>
</tr>
<tr>
<td>Schedule</td>
<td>3/3</td>
<td>1/2</td>
<td>2/1</td>
</tr>
<tr>
<td>Risk</td>
<td>3/1</td>
<td>1/3</td>
<td>2/2</td>
</tr>
<tr>
<td>Effort</td>
<td>3/1</td>
<td>1/3</td>
<td>2/2</td>
</tr>
<tr>
<td>User Satisfaction</td>
<td>3/3</td>
<td>1/1</td>
<td>2/2</td>
</tr>
<tr>
<td>Mgmt Satisfaction</td>
<td>1/1</td>
<td>3/3</td>
<td>2/2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>28</strong></td>
<td><strong>22</strong></td>
<td><strong>22</strong></td>
</tr>
</tbody>
</table>

A is an overall winner!
Reflections and Limitations

• Worked on this all 3+ years of SELP program, learned a lot
• Why didn’t I convince them? “Demand-side benefits of scale” (“network effects”) – buyers trust larger companies
• Banner is a trap: Customer switching costs are high
• Started from pure frustration, came to understanding
• Discovered that I am Lean and Agile! 😊
• Wanted to add Risk function considerations – but not sure how to do it correctly.
• Hard to get real data about costs => inaccuracy of estimates
Latest Reflection:

Banner implementation was called “building the plane mid-flight”
Summary

- Question: which To-Be system would be the most cost-effective and sustainable, Post-Upgrade or Post-Change?
- Answer: A > C > B
- Why? More players involved in the alternative B implementation and support than alternative A, and these players are more “expensive.” Even though the maintenance of the supported system is sometimes “easier” than unsupported – it is not necessarily cheaper.
- Time and cost gain from selecting A approach over B approach is more than enough to finance additional enhancements for system A and eliminate all system gaps.
- But... the final decision cannot be purely SE-based.
- What about the “other” type of the architecture?
Thank you for listening!

Thank You!

Any questions?
Legacy Student Information System: Replacement or Enhancement?

Backup Slides

Leah Burstein
Sr. PL/SQL Programmer, Loyola Law School
Advisor: Dr. Bohdan Oppenheim
SELP Capstone Project, 2015
Trade Space Analysis and Measures of Effectiveness (MoE)

**Objective functions**

\[ C = \sum_{i=1}^{n} t_i r_i \]

- \( i = 1 ... n \) – actors
- \( r_i \) – hourly rate for actor \( i \)
- \( t_i \) – time spent by actor \( i \)

**Objective function**

\[ f(I, F, L) \]

- \( I \) – number of new required interfaces
- \( F \) – number of new functions needed to address all functionality gaps
- \( L \) – average number of lines of code per program unit

---

**SIS A (Exeter In-House Upgrade)**

- Cost of Initial Upgrade
- Cost of Writing New Functionality
- Yearly Cost of System Maintenance
- Cost of New Hardware
- Cost of OS Upgrade (Consultant)
- Cost of DB Upgrade (Consultant)
- Cost of Forms Upgrade (PITSS)
- Testing: Cost of User’s Work
- Fixes: Cost of Programmer’s Work

**Values**

- Hardware: Dollars
- OS: Dollars
- DB: Dollars
- Forms: Dollars
- Test: Dollars
- Fix: Dollars
- Budget Allocated for Hardware: Dollars
- Budget Allocated for OS Upgrade: Dollars
- Budget Allocated for DB Upgrade: Dollars
- Budget Allocated for Forms Upgrade: Dollars
- Budget Allocated for Testing: Dollars
- Budget Allocated for Fixes: Dollars
- Hourly Pay Rate of a Programmer: Dollars per Hour
- Time of Programmer Work: Hours
- Hourly Pay Rate of a Tester: Dollars per Hour
- Time of testing Work: Hours
- Hourly Pay Rate of a DBA Consultant: Dollars per Hour
- Time of DBA Consultant Work: Hours
- Hourly Pay Rate of a Unix OS Specialist: Dollars per Hour
- Time of a Unix specialist Work: Hours
- # of People involved in Coding: Integer
- # of New Code to Write: Integer
- # of People involved in Maintenance: Integer

---

**SIS B (Banner, Common Database)**

- Total Cost of System Change
- Cost of LMU Fixe
- Cost of Writing New Functionality
- Yearly Cost of System Maintenance
- Time spent on restoring missing LLS functionality under new system
- Time spent on verification of LMU processes “unbroken” by joining LLS data
- Cost of restoration
- Cost of Verification

**Properties**

- Total Cost of System Change
- Cost of Data Migration: Dollars
- Time of verification for existing LMU functionality: Hours
- Time to restore lost LLS functionality: Hours
- Number of Missing Functions
- Average Missing Function Complexity
- Upper Time Limit To Spend on Functionality Restoration
- Hourly Rate of LMU Functions Restorer
- Budget Allocated for Migration
- Time Spent by Party: Hours
- Pay Rate of party: Dollars per Hour
- Number of parties Involved
- Number of LMU Processes to Verify
- Hourly rate of LMU Processes verificator
- Average Process Complexity
- Upper time limit to spend on verification
- # of Lines of New Code to Write: Integer
- # of People involved in Coding: Integer
- # of People involved in Maintenance: Integer
- Licences Cost: Dollars

---

**SIS C (Banner, Separate Database)**

- Cost of Data Migration: Dollars
- Time to restore lost LLS functionality: Hours
- # of Lines of New Code to Write: Integer
- # of People involved in Coding: Integer
- # of People involved in Maintenance: Integer
Support Phase of SDLC: Financial Aid Regulatory Requirements Example

B, C

Applying upgrade patches for regulatory changes
Id = "4B"
Text = "DBA shall not spend more than 2 hours"

ISIR Import

COD Communication Module (Method A)

TC: Time Spent on implementation of each regulatory change

XML Import

XML Export

A

Government Regulations
Id = "1"
Text = "Specific XML format, can change several times a year"

External

COD (Common Origination and Disbursement) website

requirement

Custom development for regulatory changes
Id = "4A"
Text = "Time of implementation (development and testing) shall not exceed 1 month"

The biggest difference between life-cycle support in Exeter and Banner
Example of Requirements Allocation for Alt. A + Lean Aspects

- High-Level Requirement: “SIS shall be integrated (fewer disparate systems)” (Req. ID=3)
- Use Case: Note Takers for Accommodated Students
- Disabilities Act of 1990; Confidentiality (Req. !)
- As-Is:
Note Takers: Value Stream Mapping

(Current State)

<table>
<thead>
<tr>
<th>No.</th>
<th>Actions</th>
<th>Resp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Print report from Accommodation Database to identify previous note takers</td>
<td>J</td>
</tr>
<tr>
<td>2</td>
<td>Print Roster report from Exam Database to identify all students in the class</td>
<td>J</td>
</tr>
<tr>
<td>3</td>
<td>Send email to solicit help, finding matches</td>
<td>R</td>
</tr>
<tr>
<td>4</td>
<td>Note Taker Agreement sign off</td>
<td>S-NT</td>
</tr>
<tr>
<td>5</td>
<td>Notes are submitted by students to R via email (word doc attachment, pdf). Notes are due 1 week after lecture. File naming convention: Name of class, section, professor, date of lecture, NT1 or NT2.</td>
<td>NT</td>
</tr>
<tr>
<td>6</td>
<td>Checking for correctness/reminders to late senders; resubmits</td>
<td>R</td>
</tr>
<tr>
<td>7</td>
<td>Sending emails with notes to note receivers</td>
<td>R</td>
</tr>
</tbody>
</table>

Frustrations...

2011 Fall Accommodated Students

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>John D.</td>
</tr>
<tr>
<td>5</td>
<td>Jack B.</td>
</tr>
<tr>
<td>6</td>
<td>Mary N.</td>
</tr>
<tr>
<td>18</td>
<td>Tina F.</td>
</tr>
</tbody>
</table>

2011 Fall Class List for Notes

<table>
<thead>
<tr>
<th>Professor</th>
<th>Units</th>
<th>Notetaker #1</th>
<th>Notetaker #2</th>
<th>Alternate(s)</th>
<th>Distribution of Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative Dispute Resolution E1</td>
<td>Horn</td>
<td>K.B.</td>
<td>A.B.</td>
<td>A.C., D.R., F.T.</td>
<td>6</td>
</tr>
<tr>
<td>Appellate Advocacy E3</td>
<td>Kennedy</td>
<td>M.W.</td>
<td>R.A.</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>Bankruptcy D1</td>
<td>Schechter</td>
<td></td>
<td>J.T.</td>
<td></td>
<td>4, 5</td>
</tr>
</tbody>
</table>
Note Takers: To-Be

- Elimination of Access databases
- Elimination of Email client
- 3 Levels of Customization:
  - Interface for Notes files upload to database (Exeter Student Web Services (SWS) + Oracle Application Express (APEX))
    - Actor: Note Taker
  - Interface for Notes files download (SWS+APEX)
    - Actor: Accommodated Student
  - Interface for assigning Note Takers, access to uploaded files to check correctness, and emailing
    - Actor: Student Affairs Department Employee
- Satisfies 2 top-level requirements: “fewer disparate systems” and “system modernization” (APEX not available in early Oracle versions)
Prototype

1. Assigning Note Takers (by Student Affairs Dept.)

<table>
<thead>
<tr>
<th>Course</th>
<th>Section</th>
<th>Title</th>
<th>Faculty</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>4149</td>
<td>E1</td>
<td>Animal Law</td>
<td>Sande Buhal</td>
<td>2</td>
</tr>
<tr>
<td>4D07</td>
<td>E1</td>
<td>Bankruptcy</td>
<td>Anne Wells</td>
<td>3</td>
</tr>
<tr>
<td>4D07</td>
<td>E1</td>
<td>Criminal Procedure</td>
<td>Marcy Strauss</td>
<td>4</td>
</tr>
<tr>
<td>4J77</td>
<td>E1</td>
<td>Cross Examination Intensive Workshop</td>
<td>Susan Poebls &amp; Sean Kennedy</td>
<td>2</td>
</tr>
<tr>
<td>513</td>
<td>E1</td>
<td>Disability Rights Legal Center</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4J65</td>
<td>E1</td>
<td>Fundamentals of Bar Exam Writing</td>
<td>Susan Bakhshian &amp; Jean Boylan</td>
<td>2</td>
</tr>
<tr>
<td>4C06</td>
<td>E1</td>
<td>Hate Speech Seminar</td>
<td>Marcy Strauss</td>
<td>2</td>
</tr>
<tr>
<td>4N01</td>
<td>E1</td>
<td>Income Taxation I</td>
<td>Benjamin Duncan</td>
<td>3</td>
</tr>
<tr>
<td>4J35</td>
<td>E1</td>
<td>Introduction to Intellectual Property</td>
<td>Lee Petherbridge</td>
<td>3</td>
</tr>
<tr>
<td>4J43</td>
<td>E1</td>
<td>Law Practice Management Seminar</td>
<td>Sande Buhal</td>
<td>2</td>
</tr>
<tr>
<td>4D21</td>
<td>E1</td>
<td>Police Practices Seminar</td>
<td>Steven Luie</td>
<td>2</td>
</tr>
<tr>
<td>4130</td>
<td>E1</td>
<td>Public Interest Law Practice Seminar</td>
<td>Danielle Colon &amp; James Gilliam</td>
<td>2</td>
</tr>
</tbody>
</table>

Exeter Custom Forms
2. Emailing Students (Student Affairs)

Select Email Template

Student Affairs - Gen. Mail 1

Subject

Message to LLS Notetakers

Body

<html><meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<body bgcolor="#CCFFFF">
<p>
Dear <FIRST_NAME>:</p>
<p></p>
You are receiving this message because you signed up to be a notetaker for <b>2010 SP</b> semester. To upload your files, please login to SWS and read a message on "Welcome" page to find a link to the upload page. Make sure to follow naming conventions for your files.
<p></p>
<p>With any questions please contact <a href="mailto:studentaffairs@lls.edu">Student Affairs office</a>.</p>
<p>Thank you!</p>
</body></html>
3. Uploading Files (by Note Takers)

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>4I49</td>
<td>E1 - Details</td>
</tr>
<tr>
<td>4130</td>
<td></td>
</tr>
</tbody>
</table>

Closed Sections List

Exeter SWS (Oracle 8i)

APEX (Oracle 11g)

Uploaded file should be named "CourseName_SectionName_Date_NT1(2)"

Available immediately!
4. Downloading Files (by Accommodated Students)

2011 Summer

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>4I49</td>
<td>E1 - Details</td>
</tr>
<tr>
<td>4N01</td>
<td>E1 - Details</td>
</tr>
</tbody>
</table>

Exeter SWS (Oracle 8i)

For 4I49 the notes are there:

![Download Course Notes](image)

- Download: 4I49_E1_060112_NT1.docx
- Last Updated: 01-JUN-2012 08:55AM

APEX (Oracle 11g)

For 4N01 they are not there yet:

![Download Course Notes](image)

No data found.
5. Control (by Student Affairs Dept.)

Oracle Forms 4.5

Making imperfections visible

APEX (Oracle 11g)
Note Takers: Flow Streamlining

Before

<table>
<thead>
<tr>
<th></th>
<th>Current</th>
<th>Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivery Time</td>
<td>&gt;1 week</td>
<td>&lt; days</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
<td>80%</td>
</tr>
<tr>
<td>Frustrations</td>
<td></td>
<td>80%</td>
</tr>
<tr>
<td>Customer Satisfaction</td>
<td></td>
<td>80%</td>
</tr>
<tr>
<td>Quality</td>
<td></td>
<td>80%</td>
</tr>
</tbody>
</table>

Email Communication ↓ 90%

As: Receiving Notes

NT: Submitting Notes

SA: Assigning note takers

After

SA: Assigning note takers

NT: Submitting Notes

AS: Receiving Notes

Pull

SA: Checking Notes

Activates the database:

db1

db2
Note Takers: Conclusions

- POC was written with current Exeter DB, so this is technically possible even without Exeter upgrade.
- But Alt. A would upgrade Exeter to 11g after which APEX can be installed in the same database as Exeter!
- POC written in < 2 weeks (June 2012) and demoed to Student Affairs but... SIS search was already in progress. Instead of going Lean, we bought a “monument.”
- APEX can be added to Banner too, but as a separate database (Alt. B), although with Alt. C we could do it in the same database (Apex is just an additional database schema).
Wasn’t it Agile?

From [http://www.agilemanifesto.org](http://www.agilemanifesto.org):

**Manifesto for Agile Software Development**

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

*This is what we do at LLS IT – although driven not by methodology but by common sense, tough schedules and limited workforce...*
Agile vs. Waterfall

"relating to or denoting a method of project management, used especially for software development, that is characterized by the division of tasks into short phases of work and frequent reassessment and adaptation of plans." (Google)

"agile methods replace high-level design with frequent redesign"

- flexible design,
- adaptive planning,
- evolutionary development.

- No contradiction in general – still can do SE V&V – just smaller portions
- Requirements are not effective when the users are not sure what they want
- Discovering bugs at the end is costly
- Risk of Agile: losing key players is more painful (Waterfall more plan-based than person-based)
Agile vs. Lean

Similarities
- Lean Startup methodology (2011, Eric Ries)
  - Minimum Viable Product (MVP) - the smallest thing you can build (and deploy) that delivers customer value
  - Sprint ~ Cycle Time
  - Continuous Deployment ~ Continuous Improvement

Differences
- Terminology (Sprint, Increment, Scrum (incremental technique), Product Backlog, Sprint Backlog, Scrum Master vs. Cycle Time, Kanban, Kaizen, VSM, Task Sheet, Sensei)
- Not so much focused on Waste (although eliminates it!)
To-Be System Requirements

- **Functionality**
  - Id = 1
  - Text = "All existing functionality shall be preserved in the To-Be state"

- **Ecosystem**
  - Id = 3
  - Text = "SIS shall be "integrated" (fewer disparate systems)"

- **Interfaces**
  - Id = 2
  - Text = "All old interfaces shall remain functional"

- **Interfaces for Plan A**
  - Id = 2A
  - Text = "Old interfaces shall be tested/fix after upgrade"

- **Interfaces for Plan B**
  - Id = 2B
  - Text = "Old interfaces shall be rewritten for the new system or replaced by existing Banner functionality"

- **Custom development for regulatory changes**
  - Id = 4A
  - Text = "Time of implementation (development and testing) shall not exceed 1 month"

- **Applying upgrade patches for regulatory changes**
  - Id = 4B
  - Text = "DBA shall not spend more than 2 hours"

- **Cost of Implementing New System**
  - Id = 5
  - Text = "Total Cost Shall Not Exceed 1 Min. Dollars"

- **Time of Implementation for the New System**
  - Id = 6
  - Text = "Total Time Shall Not Exceed 2 Years"
Internal Block Diagram: Grades Submission and Display

- **Professor**
  - Hand
  - Give Excel sheet with grades

- **OTR Employee**
  - Hand
  - Hand + FTP Client

- **File System on Database Server**
  - Hand
  - Procedures call
  - read the file

- **Database procedure to enter faculty input into a table**
  - Enter Grades
  - realization

- **SSS Module**
  - Web Page for Faculty
  - Web Page for Students
  - Database Table
  - Insert grades

- **Student**
  - Eye
  - Display Grades