Life Cycle Objective (LCO)

Feasibility Rationale Guidelines (FRD)

Dr. Barry Boehm

CS577 Fall 2002

Outline

- Objective and Motivation
- Content
- Audience and Participants
- Document Outline
  - Section Guidelines

Objective

- Ensure feasibility and consistency of other LCO, LCA package components
  - OCD, SSRD, SSAD, LCP, Prototype
- Demonstrate viable business case for the system
- Identify shortfalls in ensuring feasibility, consistency, and business case as project risk items for LCP

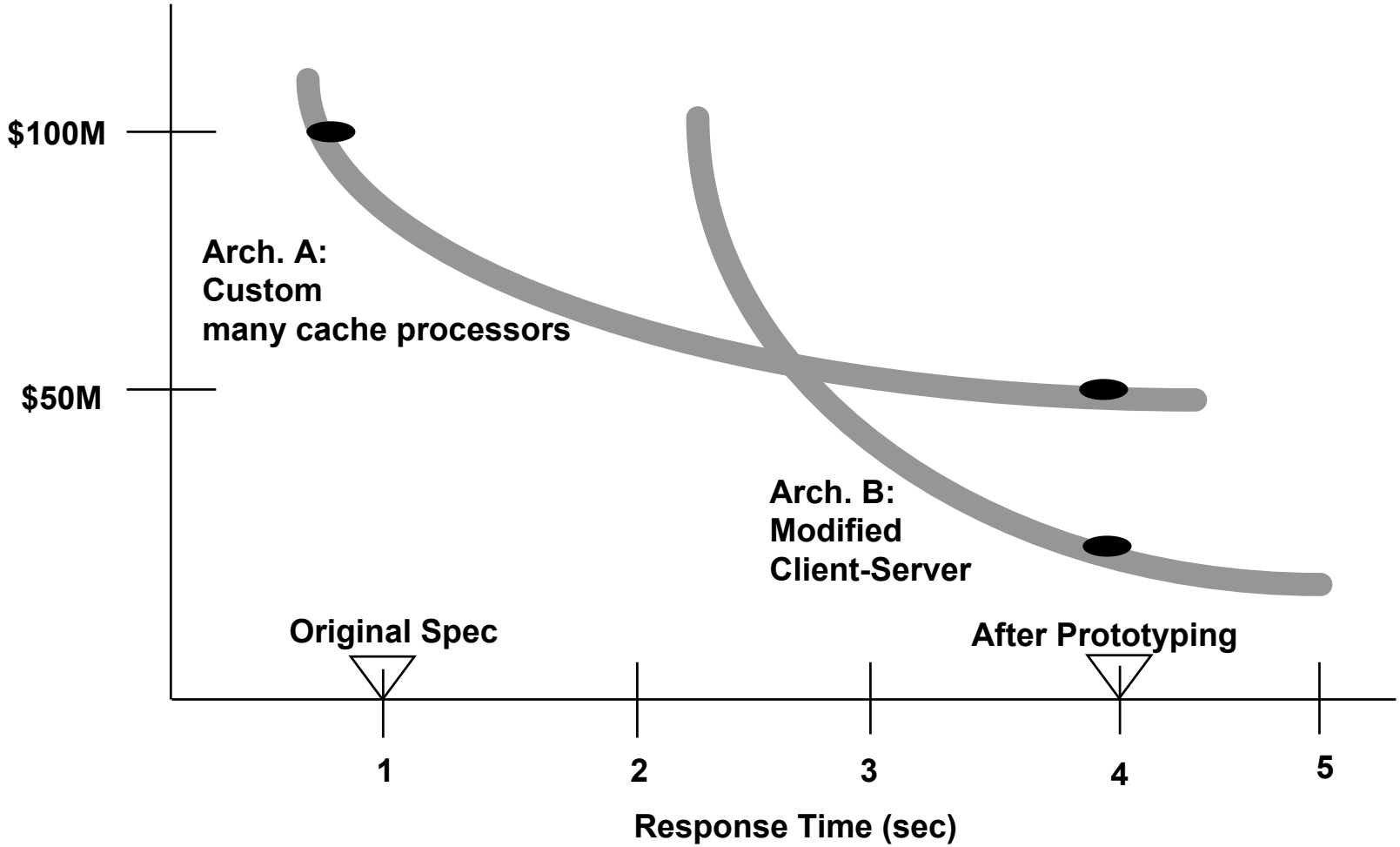
Pass/Fail Condition

- The feasibility rationale covers the key pass/fail question:
  - "If I build this product using the specified architecture and processes, will it support the operational concept, realize the prototyping results, satisfy the requirements, and finish within the budgets and schedules?"

The Need For Feasibility

1. A commercial customer specified a natural language interface for an otherwise simple query system. The project was cancelled after the natural language interface caused a factor-of-5 overrun in project budget and schedule.
2. **A government customer specified a 1-second response time for an extremely large transaction processing system. Meeting this requirement involved a custom architecture and a $100M project. The customer authorized a prototyping activity, which determined that 90% of the transactions needed only 4-second response time. With this performance requirement, a commercial-technology-based solution was achieved for $30M.**
3. A commercial customer specified a project to fully digitize a set of corporate records via scanning and optical character recognition. The resulting cost escalated by a factor of 10 after if was discovered that the records included many hard-to-capture tables, charts, and graphs.

Need for Feasibility Rationale



Content

- Business Case Analysis
  - Satisfactory (Win-Win) return on investment
- Consistency and Feasibility Rationale
  - If we build to this architecture,
  - We will support the operational concept,
  - Be consistent with the prototypes,
  - Satisfy the requirements,
  - And stay within the budgets and schedules in the plan

Audience and Participants

- Primary audience: ARB members
  - Key system stakeholders
  - Experienced peers
  - Technical Specialists in critical areas
- Also valuable to stakeholders outside ARB
- Project manager responsible for content
  - OCD author should prepare business case
  - All stakeholders responsible for consistency and feasibility via Win-Win negotiations
  - Agreements can be contingent on demonstration of feasibility

Outline

1. Introduction

1.1 Purpose of the Feasibility Rationale Document

1.2 References

2. Product Rationale

2.1 Business Case Analysis

2.1.1 Development Cost Analysis

2.1.2 Transition Cost Estimate

2.1.3 Operational Cost Estimate

2.1.4 Maintenance Cost Estimate

2.1.5 Estimate of Value Added and Return on Investment

Outline (Continued)

2.2 Requirements Satisfaction
2.2.1 Operational Concept
2.2.2 Project Requirements
2.2.3 Capability Requirements
2.2.4 Interface Requirements
2.2.5 Level of Service Requirements
2.2.6 Evolution Requirements
2.3 Stakeholder Concurrence

Outline (Continued)

3. Process Rationale

3.1 System Priorities

3.2 Process Match to System Priorities

3.3 Consistency of Priorities, Process and Resources

4. Project Risk Assessment

5. Analysis Results

5.1 Product Features

4.2 Commercial-Off-the-Shelf Solutions

6. Appendix

1. Introduction

- Describe the purpose of the document, and the intended audience
  - For a commercial system, this would involve a business case analysis demonstrating an acceptable financial Return-On-Investment (ROI).
  - For a research and education support system, the rationale would be expressed in terms of improvements in research and educational effectiveness as expressed by the users

1.1 Purpose of the Feasibility Rationale document

- To demonstrate that a system built using the specified architecture and life cycle process will
  - Satisfy the requirements
  - Support the operational concept
  - Remain faithful to the key features determined by the prototype
  - Be achievable within the budgets and schedules in the life cycle plan

1.1 Purpose of the Feasibility Rationale Document (continued)

- To rationalize development decisions in a way the prime audience (the customer and users) can understand
- To enable the customers to participate in the decision process and to express their satisfaction with the product

1.2 References

- Provide complete citations to all documents, meetings and external tools referenced or used in the preparation of this document
- Useful for consistency checking and traceability

2. Product Rationale

- Rationale for the product being able to satisfy the system specifications and stakeholders (e.g. customer, user).
- Should be consistent with:
  - Development costs (from LCP)
  - SSRD and SSAD for Requirements satisfaction

2.1 Business Case Analysis

- Describe the impact of the product in mainly monetary terms.
  - *How much does it cost to develop and to operate?*
  - *How much added value does it generate?*
  - *How high is its return on investment?*
- Non-monetary factors may be also decisive. For instance, "added value" can include the improved quality of the service provided by the product.

2.1.1 Development Cost Analysis

- Provide a summary of the full development cost, including hardware, software, people, and facilities costs.
- Proof that schedule is enough to deliver required capability
- Should be consistent with Budgets in LCP

2.1.2 Transition Cost Estimate

- Rough estimate of costs which accumulate during transition of the product into production use (e.g. training)
- Operational and support software
  - Data preparation, COTS licenses
  - Operational readiness testing
- Site preparation
  - Facilities, equipment, supplies

Transition Costs Estimate Example

From Hispanic Digital Archive's LCA

The costs of implementing the Hispanic Digital Archive will consist mainly of hardware currently not provided by ISD. The tools provided by ISD are free for USC student use and COTS software licenses would be obtained by ISD. The HDA will require server hardware to hold the DBMS and disk storage space and this is expected to be provided by ISD as a part of the IBM Digital Library (IBM DL). The only hardware costs will include two PCs for administrative use to manage the system….

Roughly 50,000 items are to be digitized and stored in the Hispanic Digital Archive. This will require scanning of archive materials and loading the digitized information in the IBM Digital Library. Several Operators (2-6 Operators) would be required to be trained to capture images. The cost of digitization as detailed in [LCP 5.2] would be $140,000. This price figure falls within the amount in which the sponsor is expected to donate yearly, which is about $100,000 for three years for the design, implementation, deployment, and maintenance of the Hispanic Digital Archive….

Using 10 minutes per item as the item entry rate, 24,960 items can be entered per year so the 50,000 items can be entered in just over 2 years. Because of unforeseen circumstances, we have allotted 3 years for the complete entry of data in to the Hispanic Digital Archive. Therefore, the projected completion of the HDA will occur during June 2002, the target date set forth by the Customer. The Customer will be required to hire a qualified Project Manager to assist in approving data entered by Operators and the cost for hiring this person would be $40,000 per year for 3 years.

2.1.3 Operational Cost Estimate

- Provide a summary of the operational costs, i.e., costs to keep the system up

The primary operational cost that will be incurred from the Streamline Unicorn Report Generator is the use of extra disk space in the storage location of the database reports. SURG will operate on an existing Unicorn report or a previously created SURG report and will subsequently create another filtered report, which will also consume disk space, so disk capacity may eventually be an issue. However, it is anticipated that this will be minimal as well, since the output of the SURG reports shall be less than or equal to the original report

From Data Mining the Library Catalogue's LCA

2.1.4 Maintenance Cost Estimate

- Provide a summary of maintenance costs if applicable
- Should be consistent with Budgets in LCP

Maintenance Estimate Example

From Data Mining the Library Catalogue's LCA

The maintenance of the SURG project, using the Source Lines of Code (SLOC) from the Constructive Cost Model (COCOMO) II estimate developed in Section 5.2.3 of the Life Cycle Plan, is developed in the steps below:

1. *Determine the Annual Change Traffic (ACT) fraction to be used in the maintenance estimate:*
   Since it is anticipated that SOME changes will be required in order to include the features that were removed from the development list for the CS577b semester due to schedule constraints, an ACT rate of 25% will be used in this maintenance equation.
2. *Obtain the estimated development in Man Months (MM)D, as well as the SLOC count for the project:*
   From the COCOMO II SLOC Estimate from Section 5.2.3 of the Life Cycle Plan:
   $(MM)_D$ EST "Most Likely" = 10.7
   SLOC = 4914
3. *Estimate the Annual Maintenance Effort (MM)AM using the COCOMO estimate:*
   $(MM)_{AM} = 1.0\ (ACT)\ (MM)_D$
   $(MM)_{AM} = 1.0\ (.25)\ (10.7) = 2.675$
4. *Determine the number of Fulltime Software Person(s) that will be required to maintain this project per year, using the following equation:*
   $(FSP)_M = (MM)_{AM}/12$
   $(FSP)M = (2.675)_{AM}/12 = .2$ or slightly more than 1/5 time from a Full Time Software Person will be required to make 25% maintenance changes per year

2.1.5 Estimate of Value Added and Relation to Cost

- Provide a summary of cost with and without the product and how much value is added by it. The value added may also describe non-monetary improvements (e.g. quality, response time, etc.) which can be critical in customer support and satisfaction.
- Include a Return-On-Investment (ROI) analysis as appropriate.

Time Spent With Current vs Proposed

In the current environment a Unicorn report can be as large as 2 Gigabytes, but are, on average, no larger than the size of memory and as many as 2000 reports are run in any given month (per the client estimates). Therefore, if an average size of a Unicorn report is 16 MB, and 5,135 pages of information are generated for the client to decipher, and it takes our client 1/2 seconds to scan each page for the information, the time spent deciphering reports takes:

½ Seconds/Page * 5,135 Pages/Report * 2000 Reports / 3600 Seconds/Hour = 1,426 Hours

It is hoped that SURG will reduce the size of the reports by at least 1/4, 1/3 or 1/2, which would reduce these numbers to:

Reduced by ¼: ½ Seconds/Page * 3,851 Pages/Report * 2000 Reports / 3600 Secs/Hr = 1,070 Hours

Reduced by 1/3: ½ Seconds/Page * 3,389 Pages/Report * 2000 Reports / 3600 Secs/Hr = 941 Hours

Reduced by 1/3: ½ Seconds/Page * 2,568 Pages/Report * 2000 Reports / 3600 Secs/Hr = 713 Hours

From Data Mining the Library Catalogue's LCA

ROI Analysis Example (Part I)

From Data Mining the Library Catalogue's LCA

Inception and Elaboration Time Invested (CS577a)	
Meetings with Full Team & Individual Members (10% time for 12 weeks)	48 Hours
Email time (1.5% time for 12 weeks)	7 Hours
Architecture Review Board(s)	6 Hours
Total (Inception and Elaboration Time)	61 Hours

Construction and Transition Time Invested (CS577b)	
Meetings with Full Team & Individual Members (7% time for 12 weeks)	34 Hours
Email time (1% time for 12 weeks)	5 Hours
Architecture Review Board(s)	6 Hours
Transition Setup (rough estimate)	10 Hours
Total (Construction/Transition Time)	54 Hours

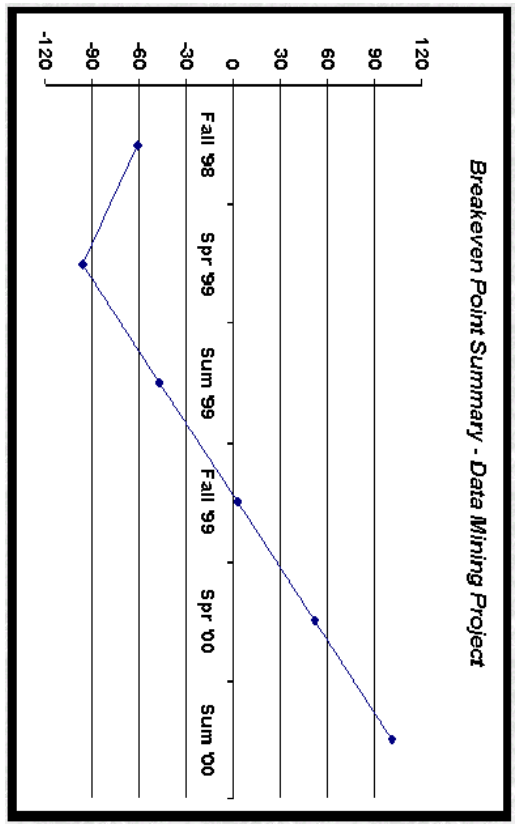
Semester Maintenance	
Maintenance Time (disk cleanup @ 2.5% time for 16 week semester)	16 Hours
Work w/maintenance team personnel on updates (1/5 Inception/Elaboration time)	12 Hours
Total (Semester Maintenance Time)	28 Hours

ROI Example (Part II)

From Data Mining the Library Catalogue's LCA

Using the previous numbers as the Investment Costs, and calculating hours saved for one person as the time it takes to review an original sized report compared to a SURG filtered report of 1/3 the original Unicorn size (See Section 2.1.5.1), the Return On Investment for this project is shown in the table and chart below:

1/3 Year Semesters	Fall '98	Spr '99	Sum '99	Fall '99	Spr '00	Sum '00
Hours Time Saved Per Month (1 person - Using 1/3 report size reduction)		5	19	19	19	19
Reports per Semester		19	78	78	78	78
Time Saved In Hours		19	78	78	78	78
Cumulative Hours		19	97	175	252	330
Time Invested in Hours	61	54	28	28	28	28
Cumulative Hours	61	116	144	172	200	229
Return On Investment		0.17	0.67	1.01	1.26	1.44



2.3 Stakeholder Concurrence

- Stakeholders may be anybody involved in the development process.
- For instance, a developer may argue that a certain response time cannot be achieved in a crisis mode unless nonessential message traffic is eliminated.
- Customer may argue that the product does not satisfy his/her win conditions (e.g. cost).

Stakeholder Concurrence Example

From Data Mining the Library Catalogue's LCA

Issue	Resolution
Conflict between desire to incorporate functionality of SURG into existing Unicorn system, and the UCS constraint to use existing applications available.	The SURG utility will be a standalone application to be used as a filter to Unicorn files AFTER they have been prepared using standard procedures. This will leave the existing system in tact, but provide the functionality desired.
Potential conflict between using the existing back-end RDBMS for SURG and the required platform support, since the RDBMS currently used is on a UNIX platform, and the requested interfaces should be on Mac and PC.	It has been determined that the development of SURG will be done using Visual Basic, and the back-end RDBMS will be MS Access, which works hand-in-hand with Visual Basic.
Architecture development in short amount of time	A primary set of features has been selected since there is a limited amount of time for both the design and the development of this product (See 3.1 System Responsibilities, High and Normal priorities for a list of the "Core Capabilities" for this project). If time is available, some of the additional features will be explored (See 3.1 Systems Responsibilities, Low priorities).
SURG should flag duplicate records only ONCE – the performance of the system will deteriorate if duplicate records are flagged for every SELECT/FORMAT option on the same report.	Duplicate records will be flagged ONCE. This feature is still listed as a "risk", and therefore has been placed in the Low priority, or second set of features.

4. Project Risk Assessment

- Any combinations of capabilities or objectives whose feasibility is difficult to assure, are major sources of risk.
- Risk Assessment consists of risk identification, risk analysis and risk prioritization.
  - Organize major sources of risk into a Top-10 (or Top-N) risk items list to be monitored via LCP 4.1 *(Risk Management and Monitoring Procedures)*
  - For critical risks indicate:
    - Description
    - Risk Exposure : magnitude and probability of loss
    - Risk Reduction Leverage: in reducing risk exposure
    - Actions to mitigate risk
    - Contingency plan
  - Identify low-priority requirements that can be left out in case of schedule slippage

Software Risk Management Techniques

Table 1 Software Risk Management Techniques

Source of Risk	Risk Management Techniques
1. Personnel shortfalls	• Staffing with top talent; key personnel agreements; team-buildin training ; tailoring process to skill mix; walkthroughs.
2. Schedules, budgets, process	• Detailed, multi-source cost and schedule estimation; design to cost; incremental development; software reuse; requirements descoping; adding more budget and schedule; outside reviews.
3. COTS, external components	• Benchmarking; inspections; reference checking; compatibility prototyping and analysis
4. Requirements mismatch	• Requirements scrubbing; prototyping; cost-benefit analysis; desig to cost; user surveys
5. User interface mismatch	• Prototyping; scenarios; user characterization (functionality; style, workload); identifying the real users
6. Architecture, performance, quality	• Simulation; benchmarking; modeling; prototyping; instrumentation; tuning
7. Requirements changes	• High change threshold: information hiding; incremental development (defer changes to later increments)
8. Legacy software	• Reengineering; code analysis; interviewing; wrappers; increment deconstruction
9. Externally-performed tasks	• Pre-award audits, award-fee contracts, competitive design or prototyping
10. Straining computer science	• Technical analysis; cost-benefit analysis; prototyping; reference checking

Sample risk analysis from Team 16-1 Spring '99
(only the first three risks are shown)

Source of Risk	Description	Potential Exposure
1. Lack of Group Interaction	Lack of group interaction would fall into the category of high communications overhead between the group members and stakeholders. Since 4 of the 5 members of the group lived on the same floor, the communications overhead was not considered to be high. However, this turned out to be the major cause for the delay of much of the documentation. **Action for Mitigation**: Increased # of group meetings	High
2. Personnel Shortfalls	Personnel shortfalls were evident in the beginning. Only our prototype had an idea of Object-oriented programming. This meant that the rest of the group had to get up to speed, especially the architect. This situation really cannot be mitigated except through training. **Action for Mitigation**: Going over each individual document collectively has helped every member understand their respective part.	High
3. Personnel Shortage	A Personnel shortage problem has already been encountered during the pre-LCO phase of the project. The actions taken to mitigate this risk were the assignment of secondary roles for each member. But this resulted in a high workload for the remaining group. **Action for Mitigation**: Secondary Roles for each member.	

5. Analysis Results

- Identify architectural alternatives and impact
- Identify unfeasible architectures or rejected alternatives; document criteria for rejection to avoid having the rejected architectural alternative selected in ignorance at some other point
- Describe feasible architectural alternatives which were rejected due to solution constraints on the way that the problem must be solved, such as a mandated technology. Those architectural alternatives may be reconsidered should the solution constraints be relaxed.

5.1 Product Features

- Discuss advantages to be gained by new (or modified) system
- List any limitations of the new system
- Discuss any alternative ways in which the new system could be realized

Sample of Analysis Results from Team 16-1 Spring '99

Name	Trade-Offs and Criteria for Rejection
Single Binary File Storage	The original database was only going to be located in one binary file. This complicated things between the Range Numbers, Map Level information, and Side note information. We decided to separate the database into three separate files which means the User Subsystem will be required to download all three files to be able to work.
Map Display	The original plan was to have the applet produce a separate window containing the map. Because an applet could not be printed, this did not fulfill one of the critical requirements defined. The alternative solution and the solution we have decided to go with in our architecture is a dynamically generated html page. This would solve the critical requirement of printability.
Servlet vs. Applet improvement	Benchmarks still pending, but since servlets are server side, client does not have to wait for anything to load. Search efficiency is the same. Use of servlets resolves browser compatibility issues encountered with applets. It also solves data synchronization issues & path generation issues.
Servers	Need faster server to handle load, but since Columbia has load balancing web servers with 3 Ultra 2 servers with 600+ MB RAM each, this should not be a problem. The use of servlets centralizes maintenance and overhead costs since client side system requirements have been reduced. Columbia Web Servers are specifically optimized to balance load. ACIS was consulted on this issue.
Database Option	The original plan was to have the administrative application run off Windows. In order for the database to be uploaded to the Cunix server, we would have to know FTP protocol. This complicated things so the resolution was to have the application run on an administrative Cunix account and the display would be exported to the Windows terminal.

[Refer to Alternatives and Trade-offs Considered (**OCD 6.3**)]

Evaluation of architectural alternatives (a possible approach)

Alternative 1

- Description (ID, name, purpose, scope, boundary)
- Context (System, Context, Stakeholder, Mission, Architecture)
- Views (static, dynamic)
- Impact/Constraint (use Component/COTS/Middleware/Database/Platform/HCI/ etc.)

Alternative 2

…

Evaluation of Alternatives

5.2 Commercial-off-the-shelf Solutions

- List of existing products that should be investigated as potential solutions.
- Reference any surveys that have been done on these products.
- Is it possible to buy something that already exists or is about to become available? It may not be possible at this stage to say with a lot of confidence, but any likely products should be listed here.
- Also consider whether there are products that must not be used.

Sample of FRD 5.2 (formerly 5.1) from Team 16-1 Spring '99

5.1 Commercial-Off-The-Shelf Solutions

The following is a list of **COTS**, which we considered using or are already using for the BSL.

DBMS:
Call number and shelf information can be stored in a DBMS. This would allow for better scalability and would allow for the system to evolve. [**OPT-2**]

Visual Cafe (evaluation copy):
The developer Paolo DeDios has opted to use the evaluation copy of Visual Cafe for rapid prototyping. [**WINC-5**]

JDK:
JDK was decided upon because it ensured maximum portability [**WINC-1**]

JSDK:
Servlet Development Kit v.2.0

Apache Jserv v.1.0b3:
Free Apache web server module that facilitates communication between browser, web-server, and BSL system. Apache is the web server used by Cunix.