Feasibility Rationale Guidelines

CS577a Course Notes
Barry Boehm

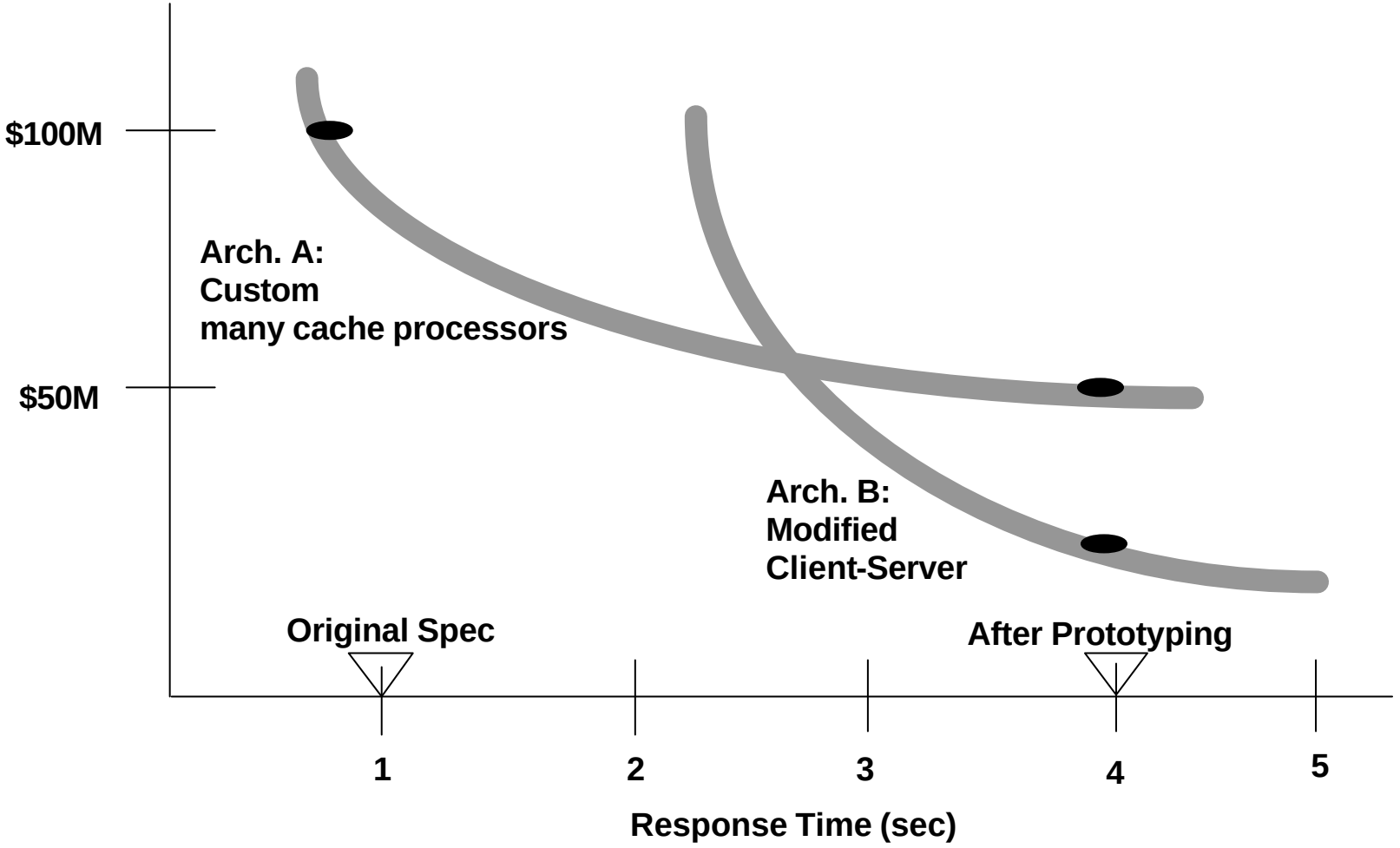
Outline

- **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

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**
- **Risk Assessment**

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 Implementation of Cost Estimate
2.1.3 Operational Cost Estimate
2.1.4 Maintenance Cost Estimate
2.1.5 Estimate of Value Added and Relation to Cost

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 Quality 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
5.2 Off-the-shelf Solutions
6. Appendices
   A. Cash Flow Statement

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 educational support system, the rationale would be expressed in terms of improvement 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
  - Satisfy a viable business case and realize benefits

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 citation 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:
  - Results Chain (OCD 2.1)
  - 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 cost.
- Proof that schedule is enough to deliver required capability
- Should be consistent with Budgets in LCP

2.1.2 Implementation 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
- Site preparation
  - Facilities, equipment, supplies

2.1.3 Operational Cost Estimate

- Provide a summary of the operational costs, i.e., costs to keep the system up
- Common pitfall: No extra cost because the people already work for ISD

2.1.4 Maintenance Cost Estimation

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

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
  - Relate to Results Chain (OCD 2.1)
- Include a Return-On-Investment (ROI) analysis as appropriate
  - Proxy: cost to do new functions manually

2.2 Requirements Satisfaction

- **This section summarizes how well a system developed to the product architecture will satisfy the system requirements.**

2.2.3 Capability Requirements

- Show evidence that the system developed to the product architecture will satisfy the capability requirements, e.g., “capability described/demonstrated/exercised as part of included COTS component,” with a pointer to the results.
- No need to restate obvious mappings from the requirements to the architecture.
  - Example: keep a count of transactions
- Common pitfall: doing this anyway

2.2.3 Capability Requirements

- For each **critical** requirement, indicate:
  - **Criticality:** Describe how essential this requirement is to the overall system
  - **Technical issues:** Describe any design or implementation issues involved in satisfying this requirement
  - **Cost and schedule:** Describe the relative or absolute costs associated with the technical issues related to satisfying that particular requirement
  - **Risks:** Describe the circumstances under which this requirement might not be satisfied, and what actions can be taken to reduce the probability of this occurrence
  - **Dependencies with other requirements:** Describe interactions with the other requirements
  - **Example:** oversized object compression, navigation

2.2.4 Interface Requirements

- Show evidence that the system developed to the product architecture will satisfy the critical interface requirements
- Should be consistent with:
  - System Interface Requirements (SSRD 3.0)
  - Environment and Data Requirements (SSRD 4.0)
- Example: IDA-LA Database and Ingest/Cataloguing (Teams 15, 16)

2.2.5 Quality Requirements

- Show evidence that the system developed to the product architecture will satisfy the critical quality requirements
- Should be consistent with Quality Requirements (SSRD 3)
- Example: multiple-platform portability

A Field Guide to Software Architecture Attributes Analysis Methods

Method	Examples	Strengths	Potential Concerns
Interface Checking	StP, RDD-100	• Static integrity • Traceability	• Dynamic integrity • Performance, cost, schedule analysis • Subjective attributes
Formalized Models	Rapids, Wright, HDM, AAA	• Static, dynamic integrity • Security • Interoperability	• Model granularity and scalability • Performance, cost, schedule analysis • Subjective attributes
Scenario Analysis	SAAM	• Subjective attributes -usability, modifiability •Human-machine system attributes: -safety, security, survivability	• Largely manual, expertise-dependent • Scenario representativeness; method scalability • Ver/Val/Accreditation • Integrity, performance, cost, schedule analysis
Simulation; Execution	Network VOA; UNAS	• Performance analysis • Dynamic integrity • Reliability, survivability, accuracy	• Model granularity and scalability • Input scenario representativeness • Ver/Val/Accreditation • Cost, schedule, subjective attributes
Parametric Modeling	A4, COCOMO, Queuing Models	• Cost schedule analysis • Reliability, availability analysis • Performance analysis	• Subjective attributes • Static, dynamic integrity • Ver/Val/Accreditation • Input validation

2.2.6 Evolution Requirements

- Show evidence that the system developed to the product architecture will satisfy the evolution requirements
- Should be consistent with Evolution Requirements in SSRD 5.0
- Example: compatibility with downstream COTS product integration

2.3 Operational Concept Satisfaction

- Summarize product's ability to satisfy the key operational concept elements and critical scenarios
- Should be consistent with Operational Scenarios (OCD 5.0)
- Example: Restricted-access portal (Team 21)

2.4 Stakeholder Concurrence

- Summarize stakeholder concurrence by reference to:
  - Win Win negotiation results
  - memoranda of agreements, etc.
- This section serves as a protocol of decisions made during the development process and further summarizes the findings so that green light can be given to proceed with the development

2.4 Stakeholder Concurrence (continued)

- 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)

3. Process Rationale

- This sections describes the rationale of the development process being able to satisfy the stakeholders (e.g. customer).
- This section is highly dependent on other documents:
  - Life Cycle Plan (LCP) for milestones
  - System and Software Requirements Description (SSRD) for system capabilities
- 577b: IOC in 10 weeks; transition in 2 weeks

3.1 System Priorities

- **Summarize priorities of desired capabilities and constraints. Priorities may express time and date but also quality and others. (e.g. performance).**

3.2 Process Match to System Priorities

- Provide rationale for
  - Ability to meet milestones
  - Rationale for choice of process model (decision table)
  - Spiral Cycles, Anchor points
  - Increments; Design-to-Schedule options

3.3 Consistency of Priorities, Process and Resources

- **Provide evidence that priorities, process and resources match.**
  - **Budgeted cost and schedule are achievable**
  - **No single person is involved on two or more full-time tasks at any given time.**

4. Project Risk Assessment

- Major sources of risk
  - Description
  - Risk Exposure RE = Prob(Loss) * Size (Loss)
  - Risk Reduction Leverage = $(RE_{before} - RE_{after})/cost$
  - Actions to Mitigate Risk
  - Contingency Plan
- Schedule risk: low-priority deferrable requirements

Software Risk Management Techniques

Source of Risk	Risk Management Techniques
1. Personnel shortfalls	• Staffing with top talent; key personnel agreements; team-building; 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; design 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; incremental 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

Software Risk Management Techniques

Growth Envelope	Understanding of Requirements	Robustness	Available Technology	Architecture Understanding		
Limited			COTS		Buy COTS	Simple Inventory Control
Limited			4GL, Transform		Transform or Evolutionary Development	Small Business-DP Application
Limited	Low	Low		Low	Evolutionary Prototype	Advanced Pattern Recognition
Limited to Large	High	High		High	Waterfall	Rebuild of old system
	Low	High			Risk Reduction Followed by Waterfall	Complex Situation Assessment
		High		Low		High-Performance Avionics
Limited to Medium	Low	Low-Medium		High	Evolutionary Development	Data Exploitation
Limited to Large			Large Reusable Components	Medium to High	Capabilities-to-Requirements	Electronic Publishing
Very Large		High			Risk Reduction & Waterfall	Air Traffic Control
Medium to Large	Low	Medium	Partial COTS	Low to Medium	Spiral	Software Support Environment

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**

5.1 Product Features

- **Advantages**
- **Limitations**
- **Tradeoffs Considered**
- **Changes Considered**

5.2 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

6. Appendices

- List or provide any references to supporting documentation