Conflicts Among Architecture Evaluation Criteria

Barry Boehm, Hoh In, USC

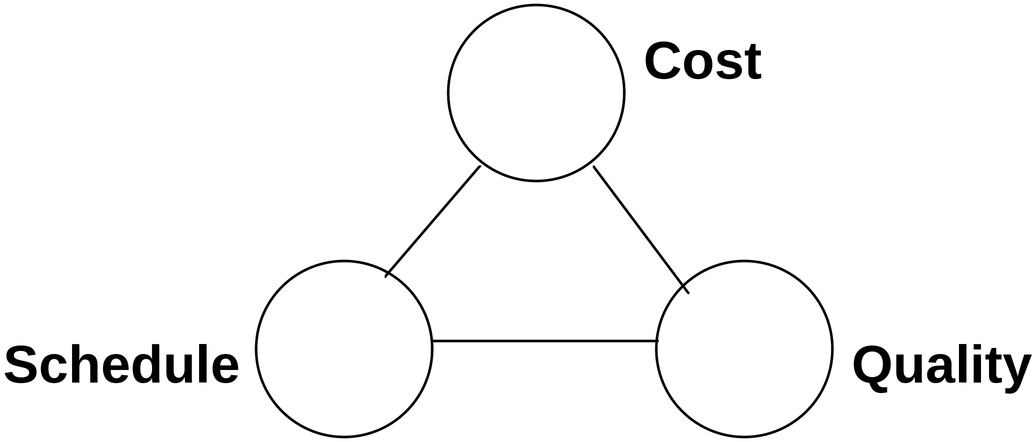
GSAW 98

February 25, 1998

Outline

- **Taxonomy of Evaluation Criteria**
  - **Generic Sources of Conflict**
- **Ground System Architecture Criteria Conflicts**
- **Characterizing Architecture Criteria Conflicts**
  - **Emerging Tools and Techniques**
- **Using Domain Criteria to Evaluate Architectural Choices**
- **Summary and References**

A Familiar Example



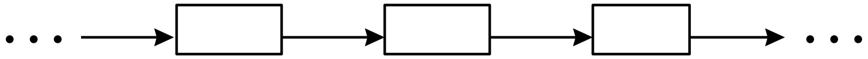
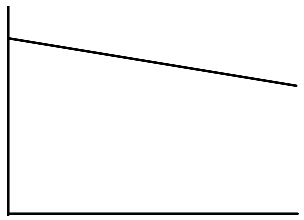
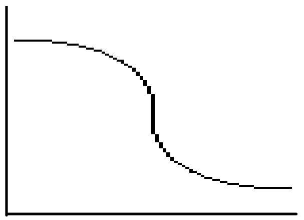
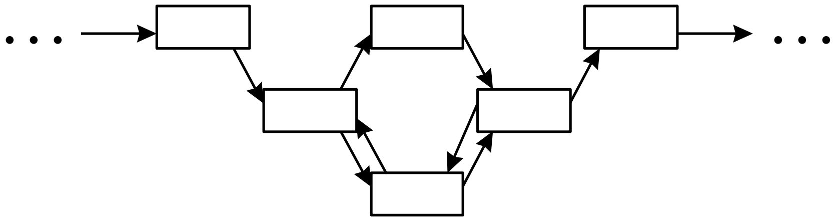
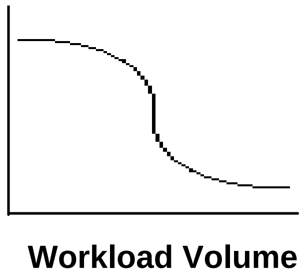
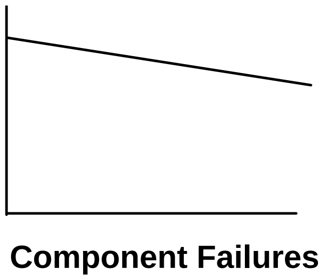
- **Can’t simultaneously optimize all three**
- **Criteria usually oversimplified**
  - –Development vs. life-cycle vs. product line
  - –Software vs. sub-system vs. system
  - –Dimensions of desired quality attributes
  - –Risk

Taxonomy of Evaluation Criteria

–All combinations are candidate sources of conflict
–These imply other criteria (e.g., reusability)

- **Technical Scope**
  - Software, computer resources, ground system, satellite mission system
- **Life Cycle Scope**
  - Development, life cycle, product line
- **Dimensions of Desired Attributes**
  - Cost, schedule, performance, adaptability, interoperability, usability, dependability
- **Risk**

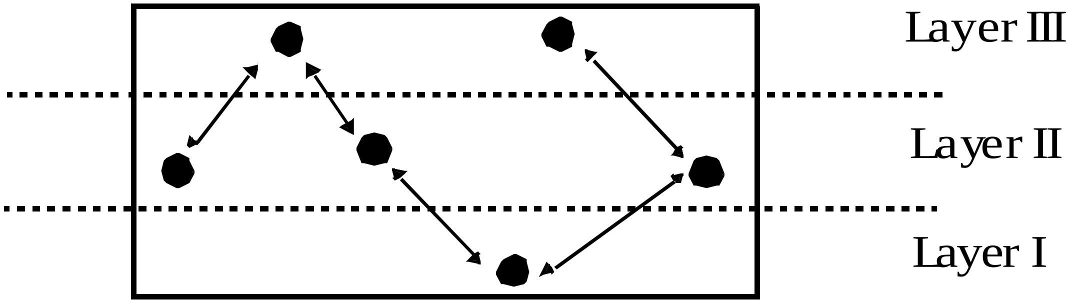
Ground Station Architecture Choices Have Differing Criteria Conflicts

Architecture	Performance	Reliability
Pipe and Filter 		
Layered 		

Architecture Choices Have Conflicts ⇔ No Universal Architecture Solution

- **Can characterize most common architecture conflicts**
- **Can embed these into conflict-advisor tools and techniques**
  - **Unit Operations, Software Architecture Analysis Method (SAAM) -- SEI**
  - **Attribute Strategies, Quality Attribute Risk and Conflict Consultant (QARCC) -- USC**

Example Architecture Attribute Strategy: Layering



- Definition:
  - **A hierarchical architectural composition in which each layer can communicate only with the adjacent upwards or downwards layer**
- Effects on quality attributes:
  - **Evolvability, Interoperability, Portability, Reusability: (+, hide sources of variation inside interface layers)**
  - **Performance (-, need more interfaces, and data and/or control transfers, via protocol)**
  - **Development Cost, Schedule: (-, more to specify, develop, verify)**

Primary Architecture Attribute Strategies

Quality Attributes	Architecture Strategies
Dependability	Assurance Monitoring & Control, Diagnostics, Fault-tolerance functions, Input acceptability checking, Instrumentation, Intrusion detection & handling, Redundancy
Interoperability	API-driven, Layering
Usability	Error-reducing user input/output, GUI-driven
Performance	Architecture balance, Parallelism, Performance Monitoring & Control, Pipelining
Adaptability	Change-source hiding, Input assertion/type checking, Layering
Cost & Schedule	4GL-driven, Architecture Balance, COTS/ Reuse-driven

Architecture Criteria Conflict Summary

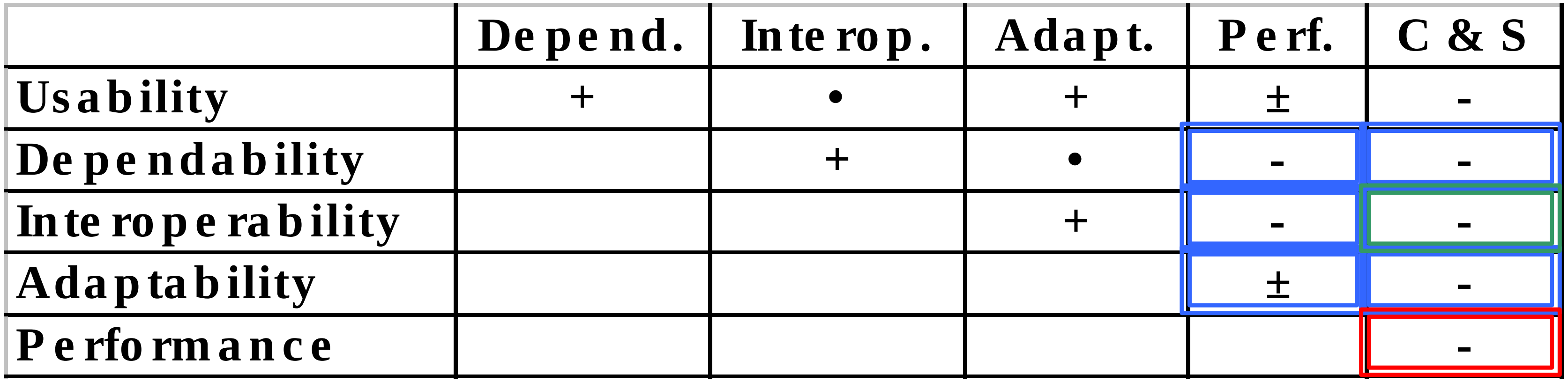
	Depend.	Interop.	Adapt.	Perf.	C & S
Usability	+	•	+	±	-
Dependability		+	•	-	-
Interoperability			+	-	-
Adaptability				±	-
Performance					-

+ : Criteria support each other

• : Criteria relatively independent

- : Criteria conflict with each other

Relative Criticality of Criteria Conflicts:
USC Workshop Survey



	Depend.	Interop.	Adapt.	Perf.	C & S
Usability	+	•	+	±	-
Dependability		+	•	-	-
Interoperability			+	-	-
Adaptability				±	-
Performance					-

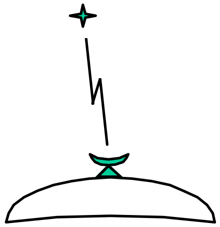
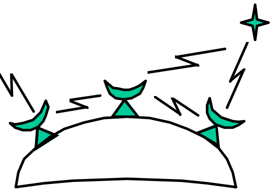
🟥 : Average rating 9 on scale of 10

🟦 : Average rating 7 - 8

🟩 : Average rating 6 - 7

Future Opportunity:

Using Domain Criteria to Evaluate Architecture Choices

Routing Rqts. Data Volume	Low 	High 
Low	Growth-Driven Choice of Layered or Pipe & Filter	Layered
High	Pipe & Filter	Pre-routed Pipe & Filter

Summary

- Critical criteria are domain-dependent, situation-dependent
  - No universal architecture solution
- Architecture criteria conflict analysis techniques becoming available
- Opportunity to develop Ground System domain guidelines for addressing architecture criteria conflicts
  - Will discuss in Thursday breakout session

References

L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, Addison Wesley, 1997.

B. Boehm and H. In, “Identifying Quality-Requirement Conflicts,” IEEE Software, March 1996, pp. 25-36.

M Shaw and D Garlan, Software Architecture, Prentice Hall, 1996.

Relative Criticality of Criteria Conflicts:
USC Workshop Survey

	Depend.	Interop.	Adapt.	Perf.	C & S
Usability	+	•	+	±	-
Dependability		+	•	- 🟦	- 🟦
Interoperability			+	- 🟦	- 🟩
Adaptability				± 🟦	- 🟦
Performance					- 🟥

🟥 : Average rating 9 on scale of 10

🟦 : Average rating 7 - 8

🟩 : Average rating 6 - 7