

# Architectural Analysis

**Shawn Bohner**  
Office: Moench Room F212  
Phone: (812) 877-8685  
Email: [bohner@rose-hulman.edu](mailto:bohner@rose-hulman.edu)



**ROSE-HULMAN**  
INSTITUTE OF TECHNOLOGY

# Today's Agenda

- ❖ **Architectural Analysis**

- ❖ **Design Studio:  
Team 11 – Excel Graphing Solution**

# Recall Architectural Building Blocks

**Component** – a unit of computation or a data store (either atomic or composite)

**Connector** – an architectural element that models interactions among components and rules that govern those interactions

**Configuration** (or topology) – a connected graph (composite) of components and connectors which describe architectural structure

# Architectural Analysis

The identification and resolution of the system's non-functional requirements (e.g., security, maintainability) in the context of functional requirements (e.g., calculate trajectory, generate report)

# Goals of Architectural Analysis

- ❖ **Identify and resolve non-functional requirements**
- ❖ **Identify variation points**
- ❖ **Identify most probable evolution points**
- ❖ **Hierarchy of Decision Goals**
  - **Inflexible constraints (e.g., safety or legal)**
  - **Business goals (they pay the money...)**
  - **All others**

# Why does Architectural Analysis Matter?

- ❖ **Reduce risk of missing something key to the design of the system**
- ❖ **Avoid applying excessive effort to low priority issues**
- ❖ **Help align the software product with the business (or system) goals**

# When do we Analyze the Architecture?

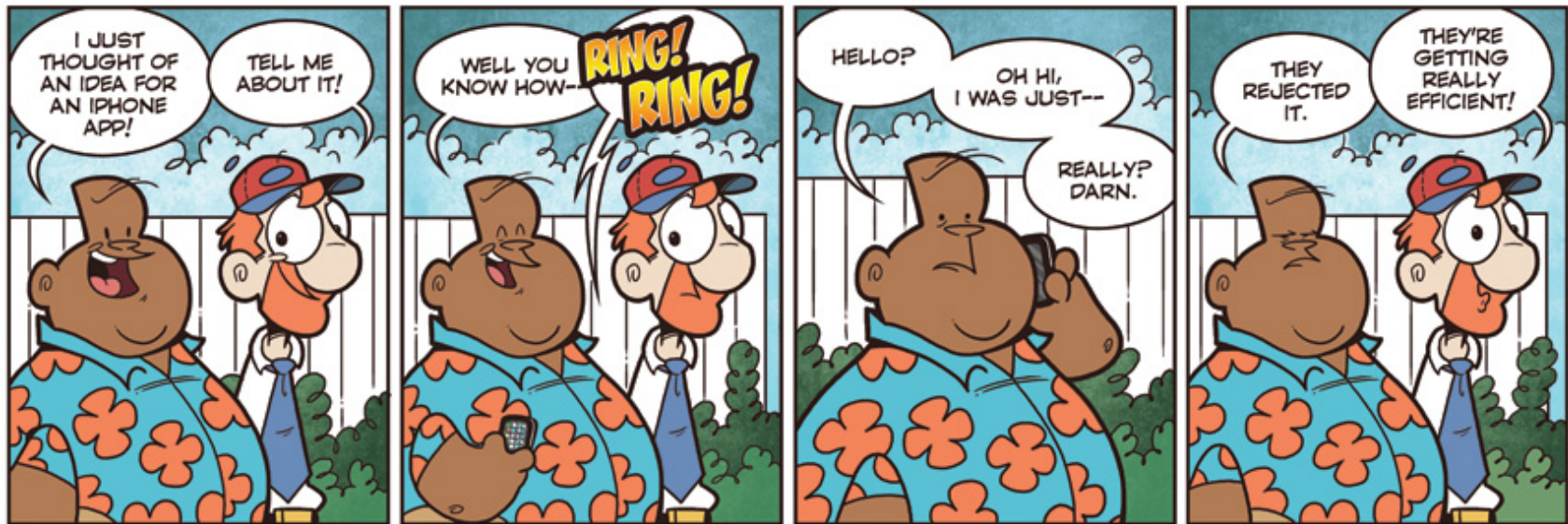
- ❖ **Before first iteration, to manage risk**
- ❖ **In elaboration and after each design iteration**
  - **Act as 'toll-gate' before starting next phase**

# **Recall Points of Change from Protected Variation Discussion...**

- ❖ **Variation points: points of change *in the existing system or requirements***
  - e.g., multiple tax calculators
- ❖ **Evolution points: points of change that *may arise in the future* but not currently present**
  - e.g., hand-held POS devices



# Cartoon of the Day



Not Invented Here™ © Bill Barnes & Paul Southworth

NotInventedHere.com

Used by permission. <http://notinventedhe.re/on/2009-11-23>

# Common Steps in Architectural Analysis

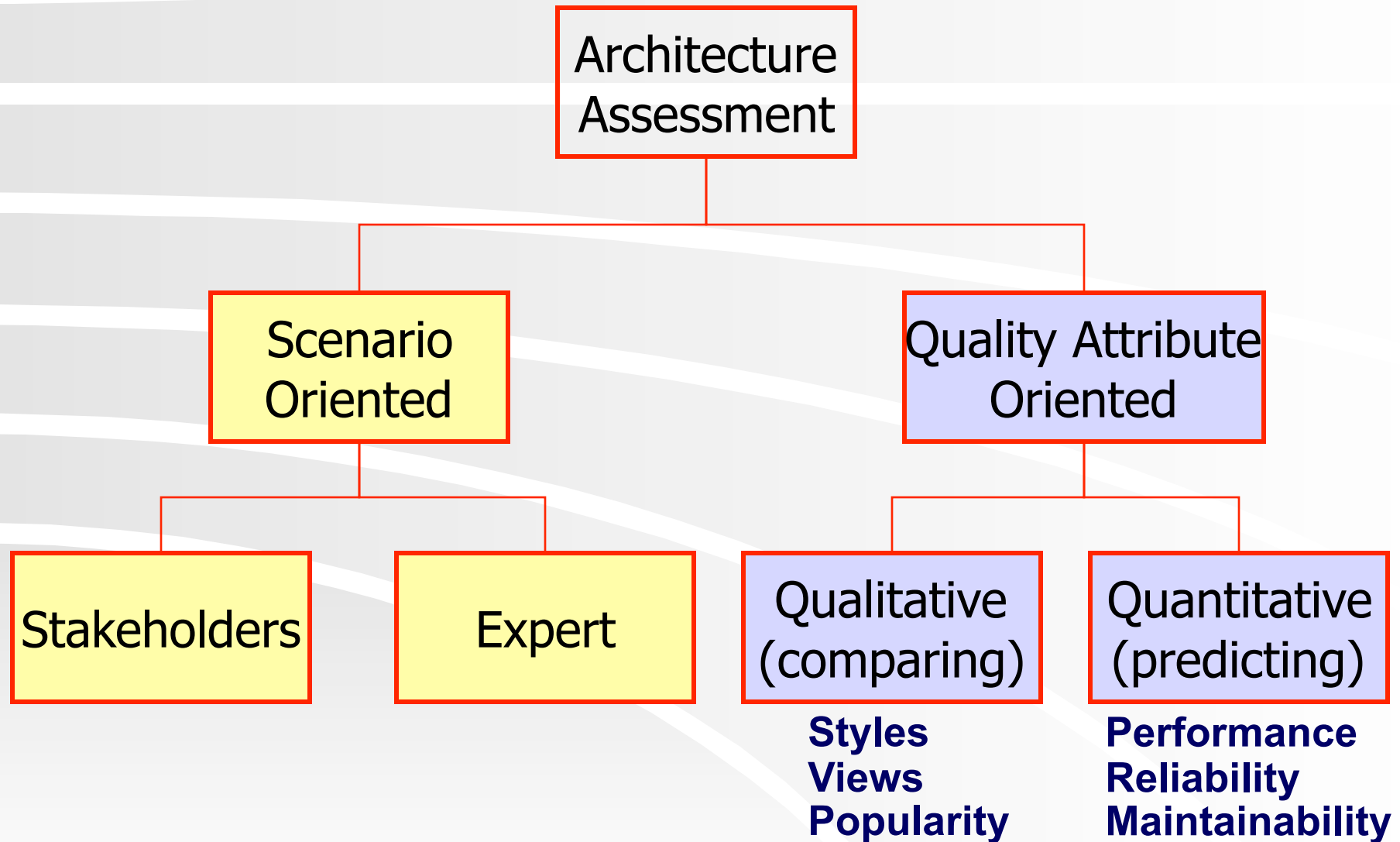
- ❖ **Identify and analyze non-functional requirements (AKA architectural drivers or factors) that impact architecture**
- ❖ **Evaluate alternative designs and create solutions to resolve impacts (AKA architectural decisions)**
  - **Formulate “quality scenarios” that define measurable/observable metrics**

# Priorities and Future Proofing

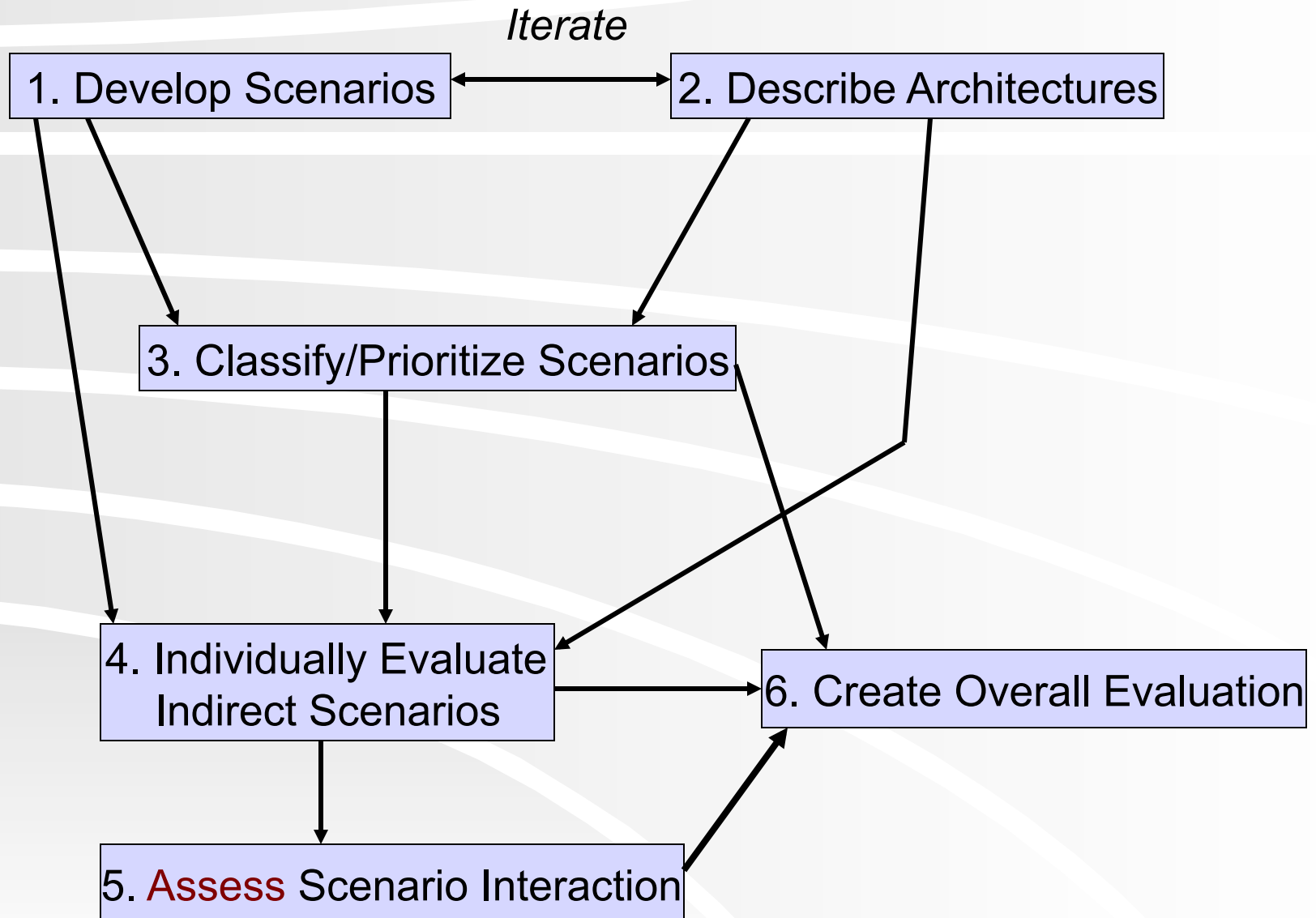
- ❖ ***Future Proofing*** can lead to over-engineering for changes that are unlikely to occur
  - Exception: Prudent Future Proofing like Year 2000
- ❖ **Priorities drive under-engineering**
  - Getting it done over getting it done right

***The art of the architect is knowing what battles are worth fighting – where it's worth investing in designs that provide protection against evolutionary change.***

# Architecture Analysis/Assessment



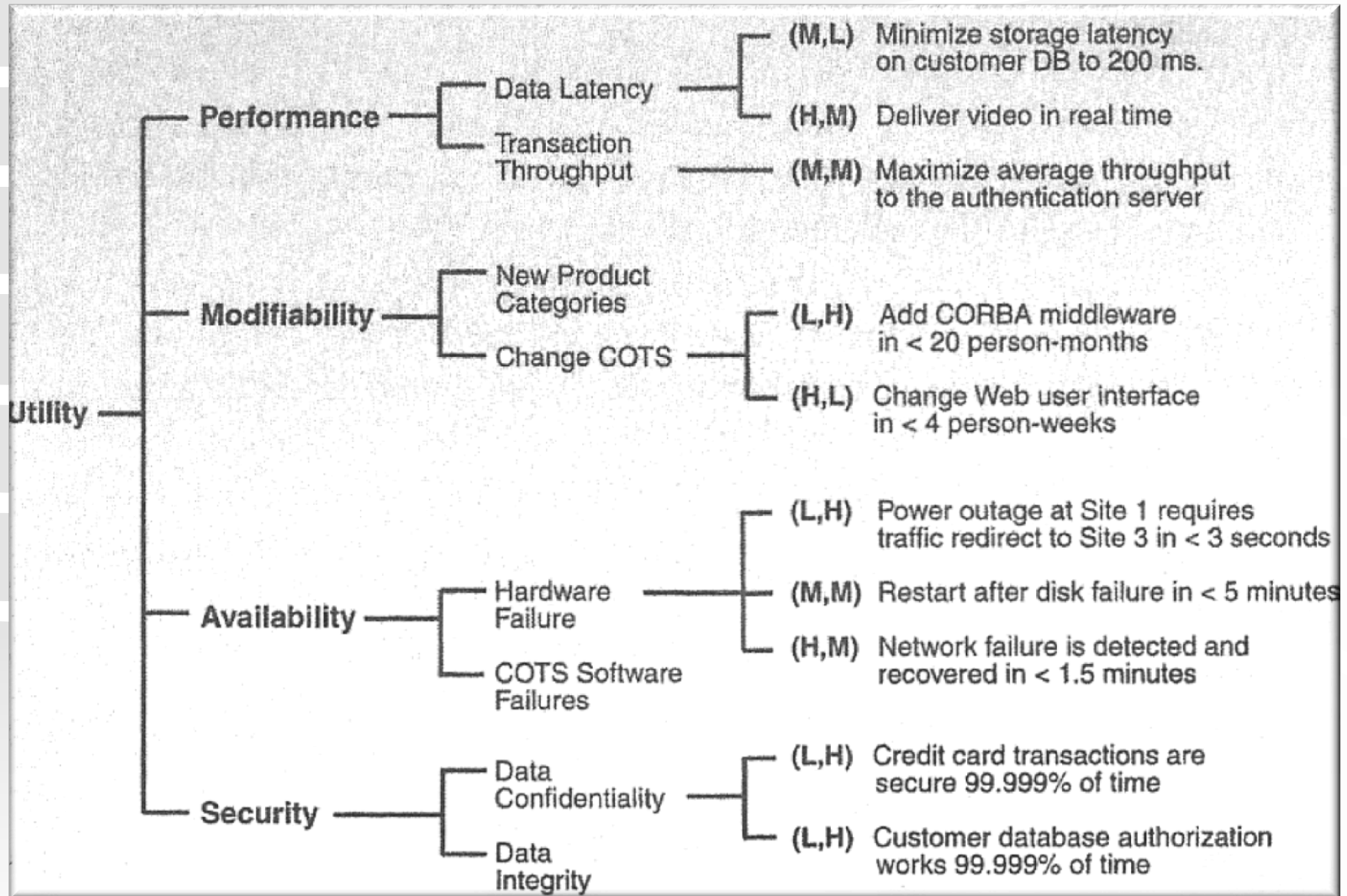
# SAAM – SW Architecture Analysis Method



# Factor Table in Supplementary Spec.

Factor	Measures and quality scenarios	Variability (current flexibility and future evolution)	Impact of factor (and its variability) on stakeholders, architecture and other factors	Priority for Success	Difficulty or Risk
<b>Reliability—Recoverability</b>					
Recovery from remote service failure	When a remote service fails, reestablish connectivity with it within 1 minute of its detected re-availability, under normal store load in a production environment.	current flexibility - our SME says local client-side simplified services are acceptable (and desirable) until reconnection is possible. evolution - within 2 years, some retailers may be willing to pay for full local replication of remote services (such as the tax calculator). Probability? High.	High impact on the large-scale design. Retailers really dislike it when remote services fail, as it prevents them from using a POS to make sales.	H	M
Recovery from remote product database failure	as above	current flexibility - our SME says local client-side use of cached “most common” product info is acceptable (and desirable) until reconnection is possible. evolution - within 3 years, client-side mass storage and replication solutions will be cheap and effective, allowing permanent complete replication and thus local usage. Probability? High.	as above	H	M

# Factors: Quality Attribute Utility Tree



Note: High-Medium-Low for (Priority, Risk)

# Technical Memos: Documenting Decisions

- ❖ Summarize the *issue*
- ❖ List the relevant *architectural factors*
- ❖ Describe the chosen *solution*
- ❖ Give the *motivation* for choosing the **solution**
- ❖ Note any *unresolved issues*
- ❖ Identify *alternatives considered*

Including rationale for rejecting alternatives





# Separation of Concerns

Architectural factors  
often *cross-cutting*

- ❖ **Some large-scale techniques for SoC:**
  - **Modularize/encapsulate into separate components**
    - E.g., persistence service/façade, layered arch.
  - **Use decorators**
  - **Use post-compilers or aspect-oriented techniques**
  - **Architecture description languages (ADLs)**

# Key Themes in Architectural Analysis

- ❖ Architectural concerns especially related to **non-functional** requirements
- ❖ Architectural concerns involve system-level, **large-scale**, and broad problems that involve **fundamental design decisions**
- ❖ Architectural analysis is about **understanding the interdependencies and tradeoffs** in design decisions
- ❖ Architectural analysis is about the generation and evaluation of **alternative solutions**

# **Design Studio:**

## **Team 11: Excel Graphing Solution**

- ~5 minutes: team describes problem and current solution (if any)**
- ~3 minutes: class thinks about questions, alternative approaches**
- ~12 minutes: on-board design with team modeling and instructor advising/facilitating**

# Homework and Milestone Reminders

- ❖ **Read Chapters 34 and 35**
- ❖ **Milestone 5– Iteration 3 Junior Project System with finalized Design Document**
  - **Due by 11:59pm Friday, February 19<sup>th</sup>, 2010.**