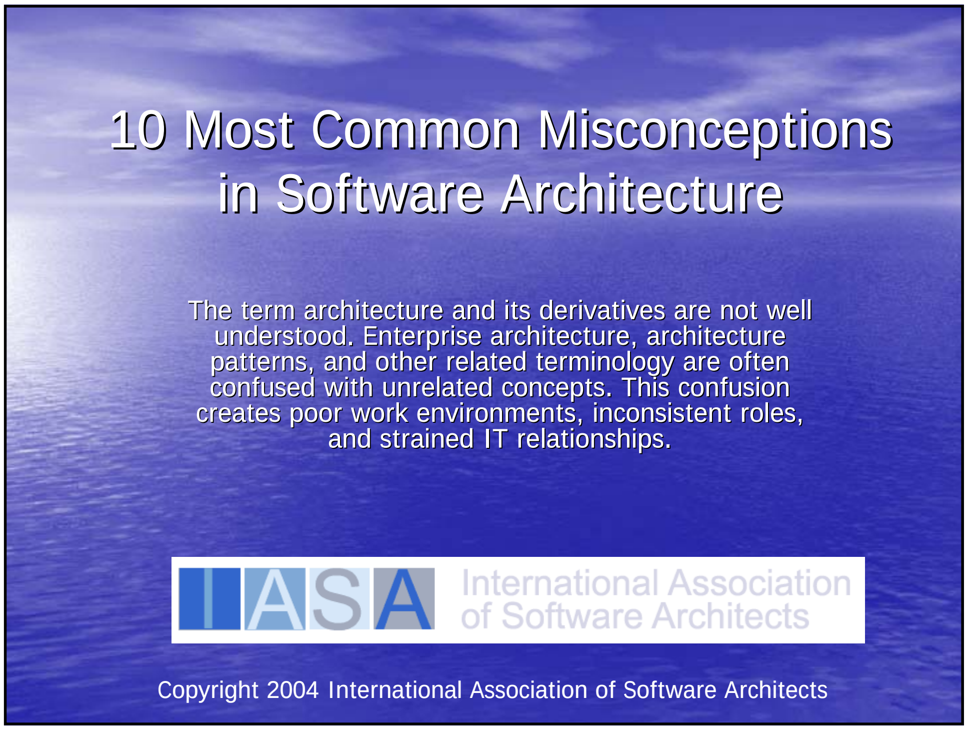
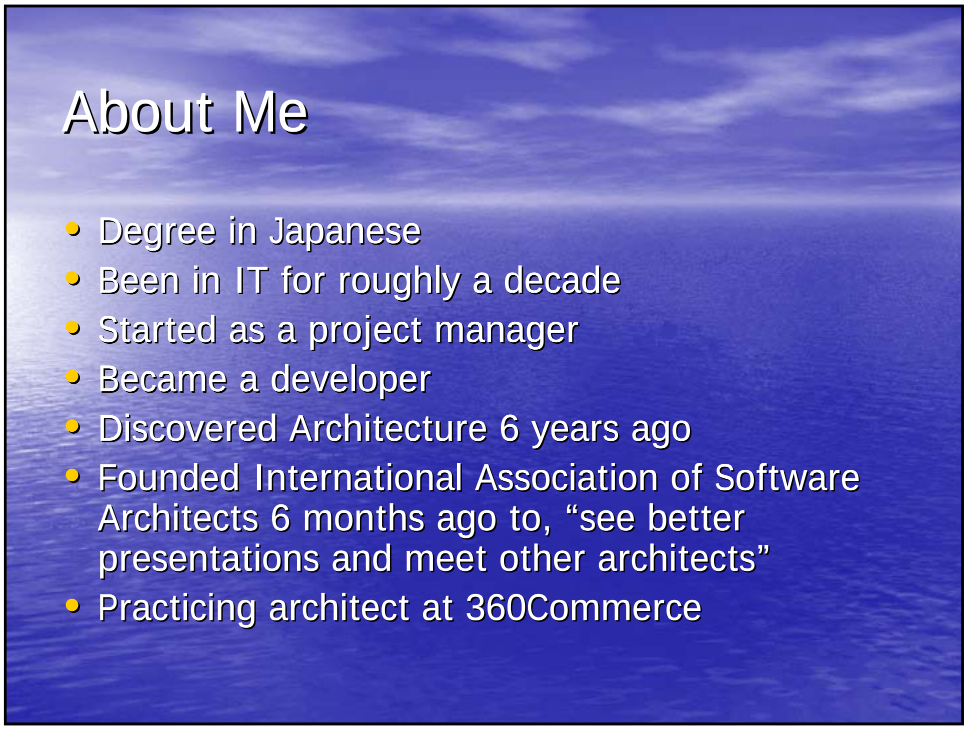


10 Most Common Misconceptions in Software Architecture

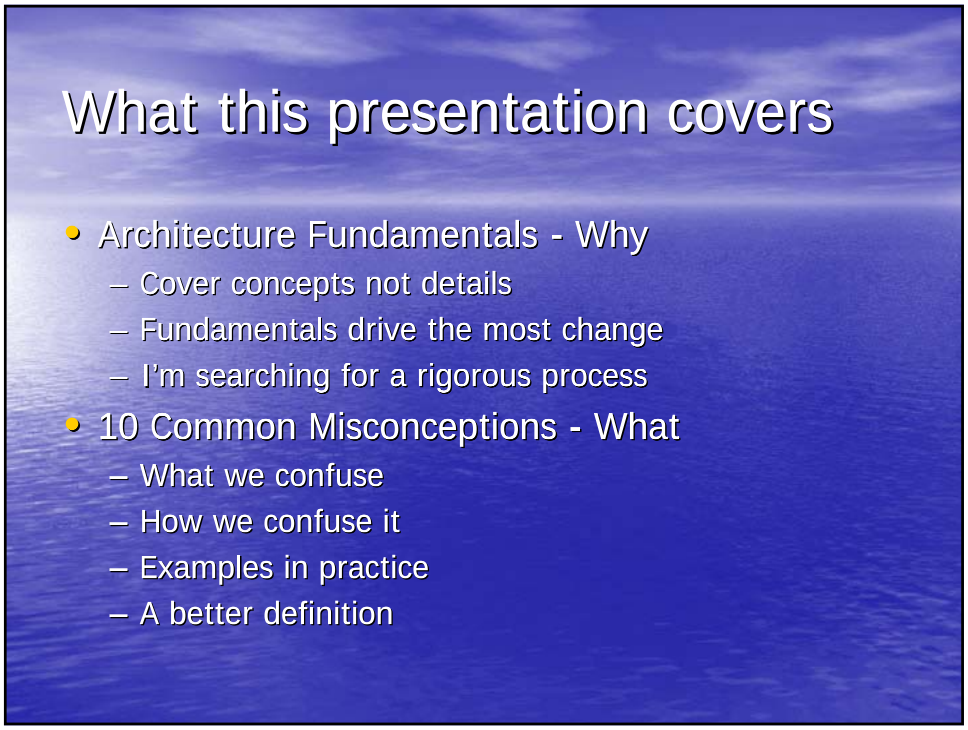
The term architecture and its derivatives are not well understood. Enterprise architecture, architecture patterns, and other related terminology are often confused with unrelated concepts. This confusion creates poor work environments, inconsistent roles, and strained IT relationships.

IASA International Association of Software Architects



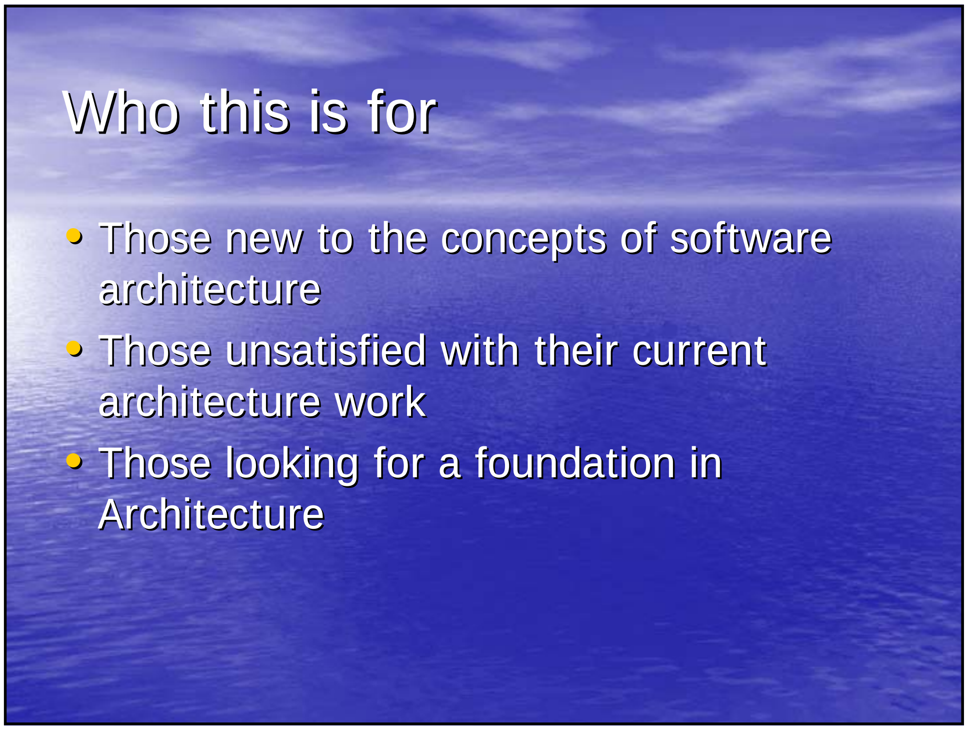
About Me

- Degree in Japanese
- Been in IT for roughly a decade
- Started as a project manager
- Became a developer
- Discovered Architecture 6 years ago
- Founded International Association of Software Architects 6 months ago to, “see better presentations and meet other architects”
- Practicing architect at 360Commerce



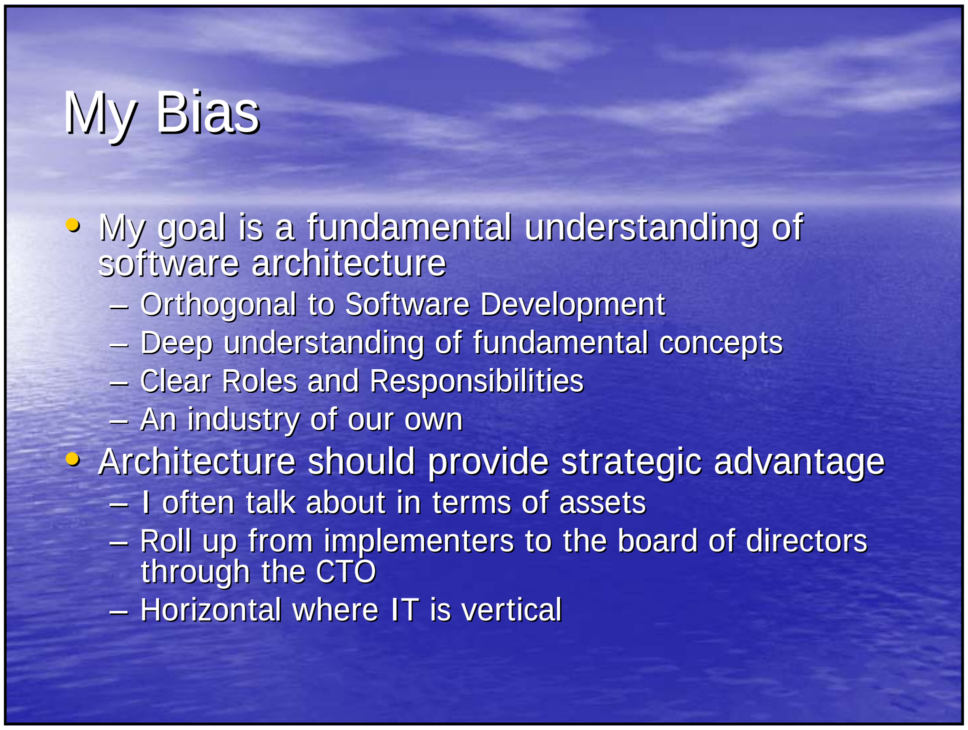
What this presentation covers

- Architecture Fundamentals - Why
  - Cover concepts not details
  - Fundamentals drive the most change
  - I'm searching for a rigorous process
- 10 Common Misconceptions - What
  - What we confuse
  - How we confuse it
  - Examples in practice
  - A better definition



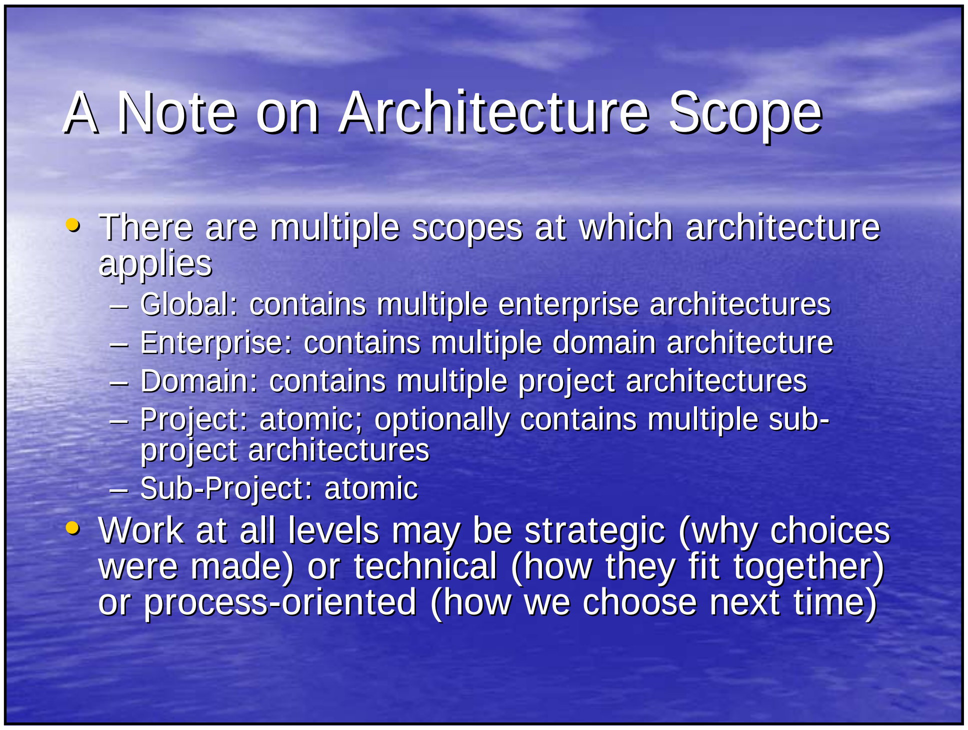
Who this is for

- Those new to the concepts of software architecture
- Those unsatisfied with their current architecture work
- Those looking for a foundation in Architecture



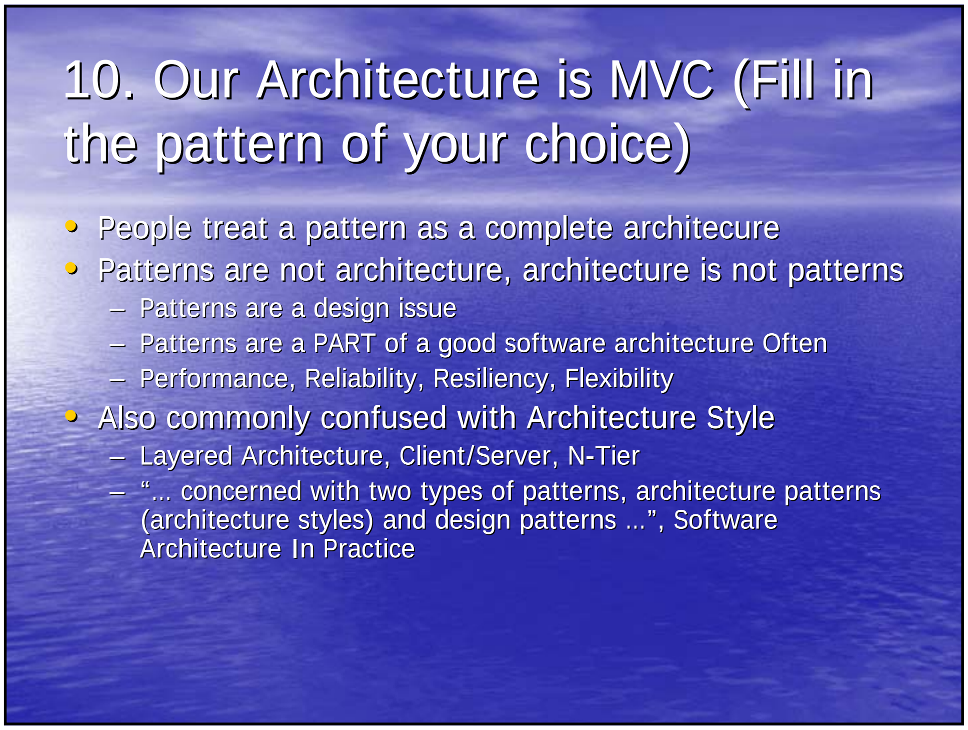
My Bias

- My goal is a fundamental understanding of software architecture
  - Orthogonal to Software Development
  - Deep understanding of fundamental concepts
  - Clear Roles and Responsibilities
  - An industry of our own
- Architecture should provide strategic advantage
  - I often talk about in terms of assets
  - Roll up from implementers to the board of directors through the CTO
  - Horizontal where IT is vertical



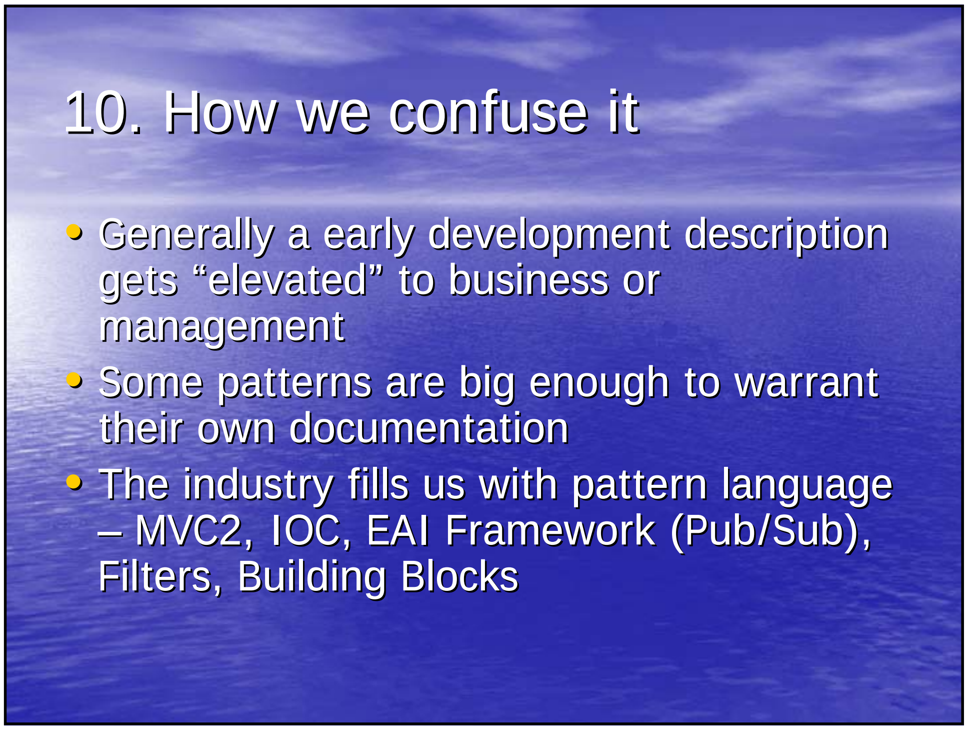
A Note on Architecture Scope

- There are multiple scopes at which architecture applies
  - Global: contains multiple enterprise architectures
  - Enterprise: contains multiple domain architecture
  - Domain: contains multiple project architectures
  - Project: atomic; optionally contains multiple sub-project architectures
  - Sub-Project: atomic
- Work at all levels may be strategic (why choices were made) or technical (how they fit together) or process-oriented (how we choose next time)



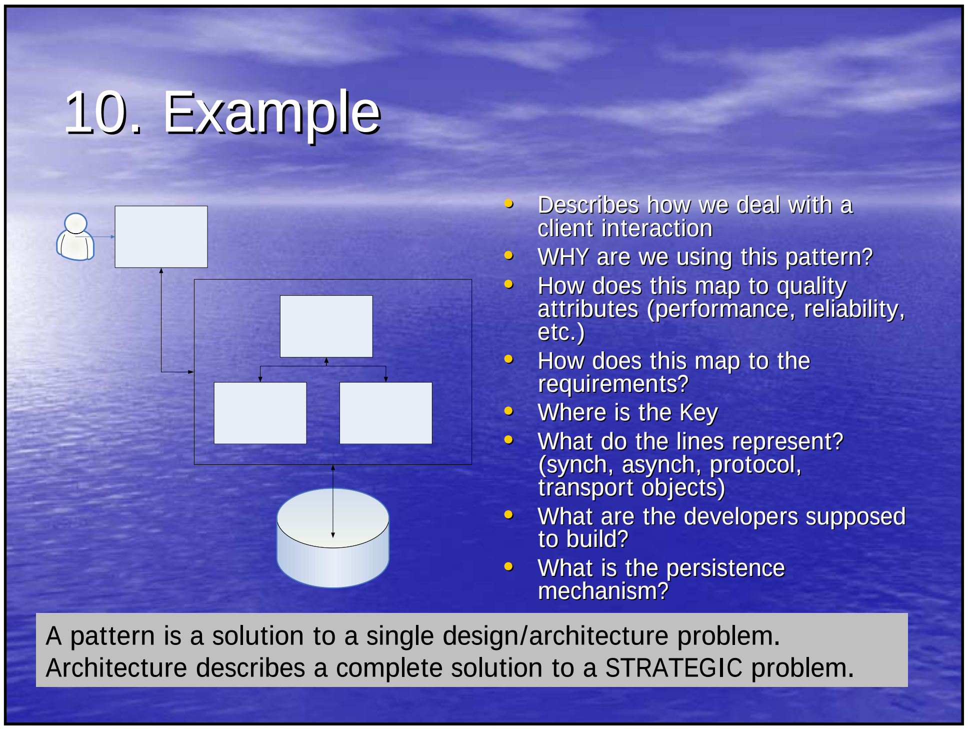
10. Our Architecture is MVC (Fill in the pattern of your choice)

- People treat a pattern as a complete architecure
- Patterns are not architecture, architecture is not patterns
  - Patterns are a design issue
  - Patterns are a PART of a good software architecture Often
  - Performance, Reliability, Resiliency, Flexibility
- Also commonly confused with Architecture Style
  - Layered Architecture, Client/Server, N-Tier
  - "... concerned with two types of patterns, architecture patterns (architecture styles) and design patterns ...", Software Architecture In Practice

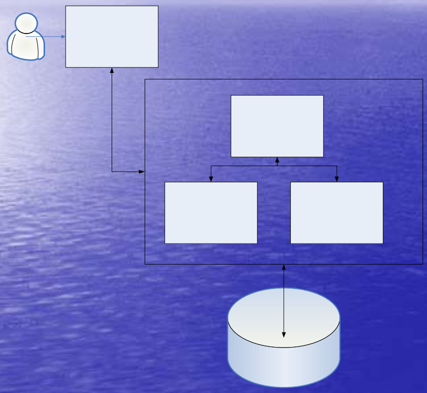


10. How we confuse it

- Generally a early development description gets "elevated" to business or management
- Some patterns are big enough to warrant their own documentation
- The industry fills us with pattern language – MVC2, IOC, EAI Framework (Pub/Sub), Filters, Building Blocks



10. Example



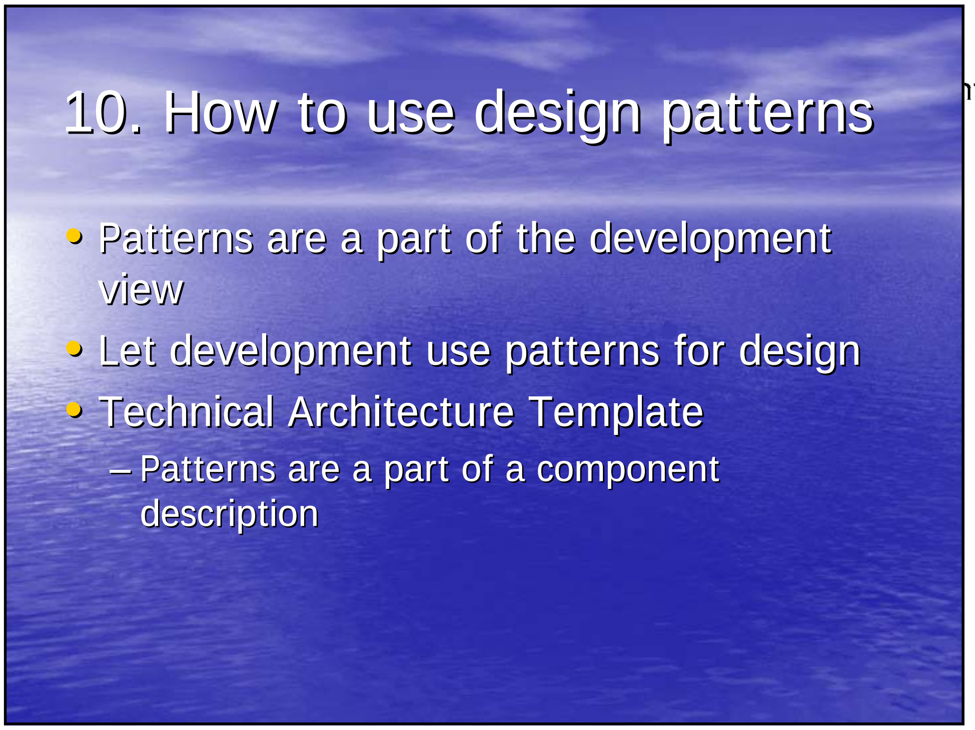
- Describes how we deal with a client interaction
- WHY are we using this pattern?
- How does this map to quality attributes (performance, reliability, etc.)
- How does this map to the requirements?
- Where is the Key
- What do the lines represent? (synch, asynch, protocol, transport objects)
- What are the developers supposed to build?
- What is the persistence mechanism?

A pattern is a solution to a single design/architecture problem.
Architecture describes a complete solution to a STRATEGIC problem.

Client

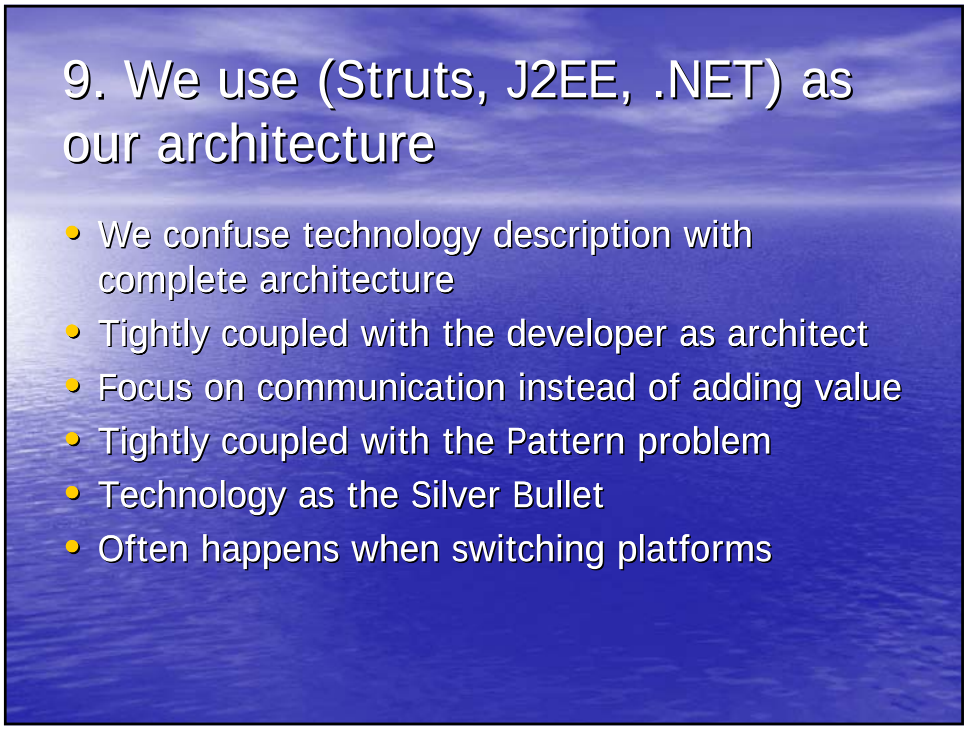
troller

Model



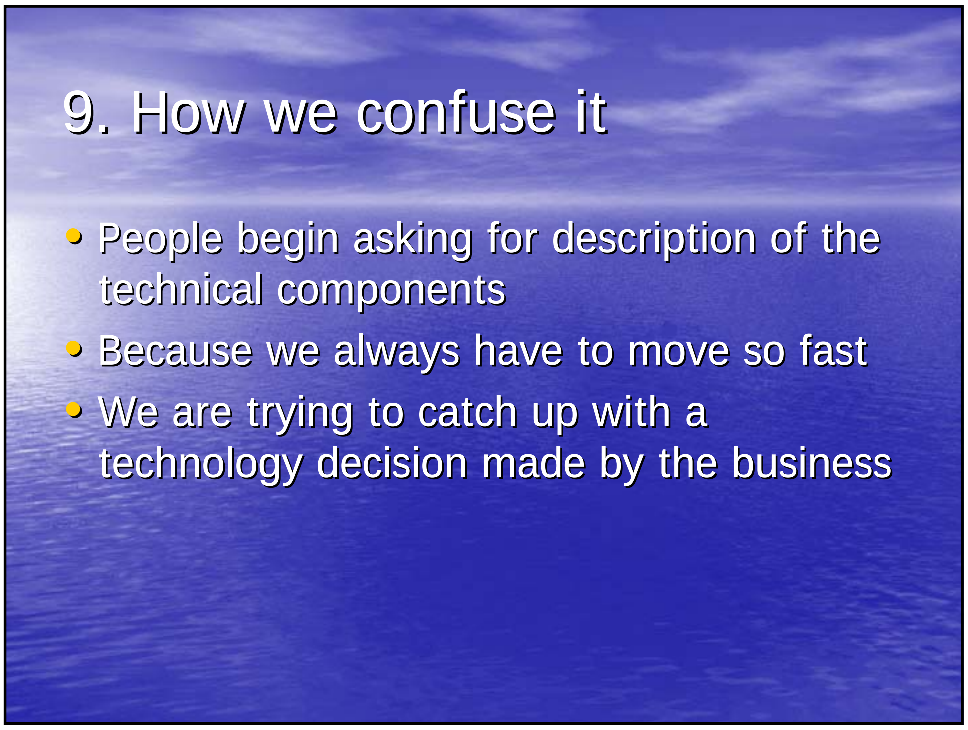
10. How to use design patterns

- Patterns are a part of the development view
- Let development use patterns for design
- Technical Architecture Template
  - Patterns are a part of a component description



9. We use (Struts, J2EE, .NET) as our architecture

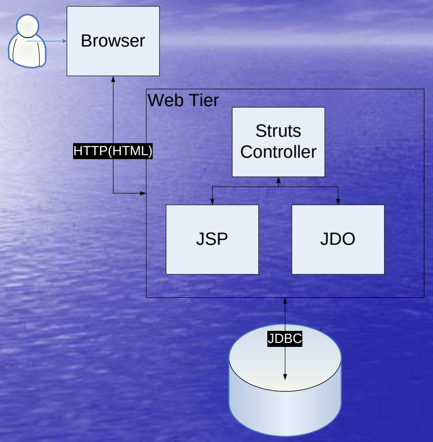
- We confuse technology description with complete architecture
- Tightly coupled with the developer as architect
- Focus on communication instead of adding value
- Tightly coupled with the Pattern problem
- Technology as the Silver Bullet
- Often happens when switching platforms



9. How we confuse it

- People begin asking for description of the technical components
- Because we always have to move so fast
- We are trying to catch up with a technology decision made by the business

9. Example

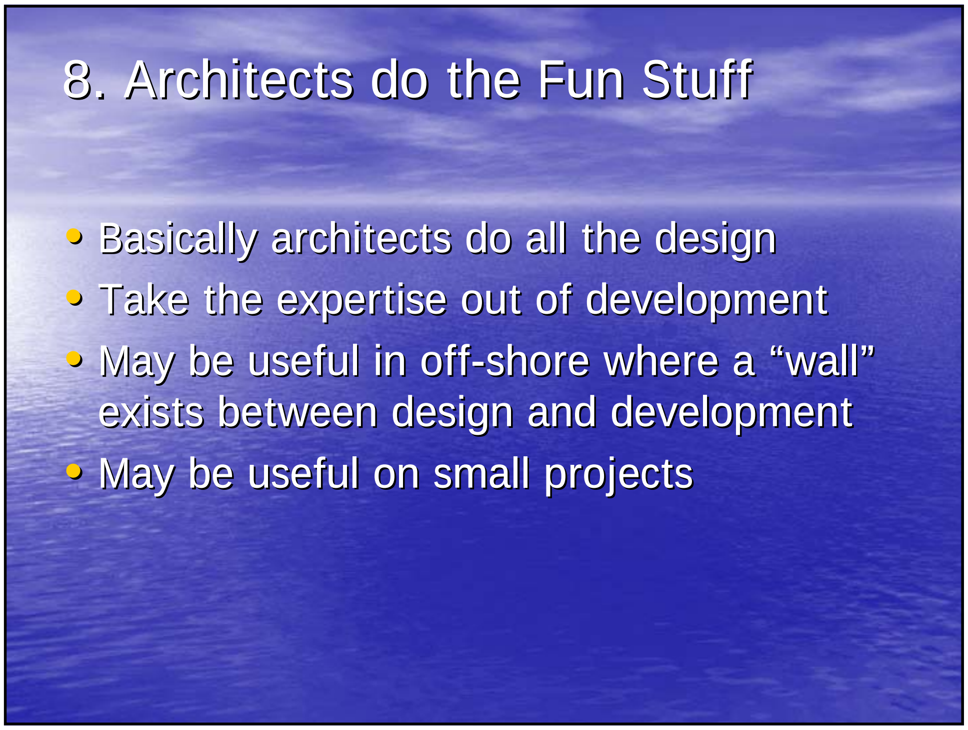


- Definitely more informative than the last view
- Why did we choose Struts?
- What is the deployment/infrastructure view?
- What does this mean to the business?
- Will Struts need to be customized?
- How does this map to use cases?
- Where will business logic be implemented?

A technology stack defines the generic limits of the design. Architecture describes how and why the stack will be used.

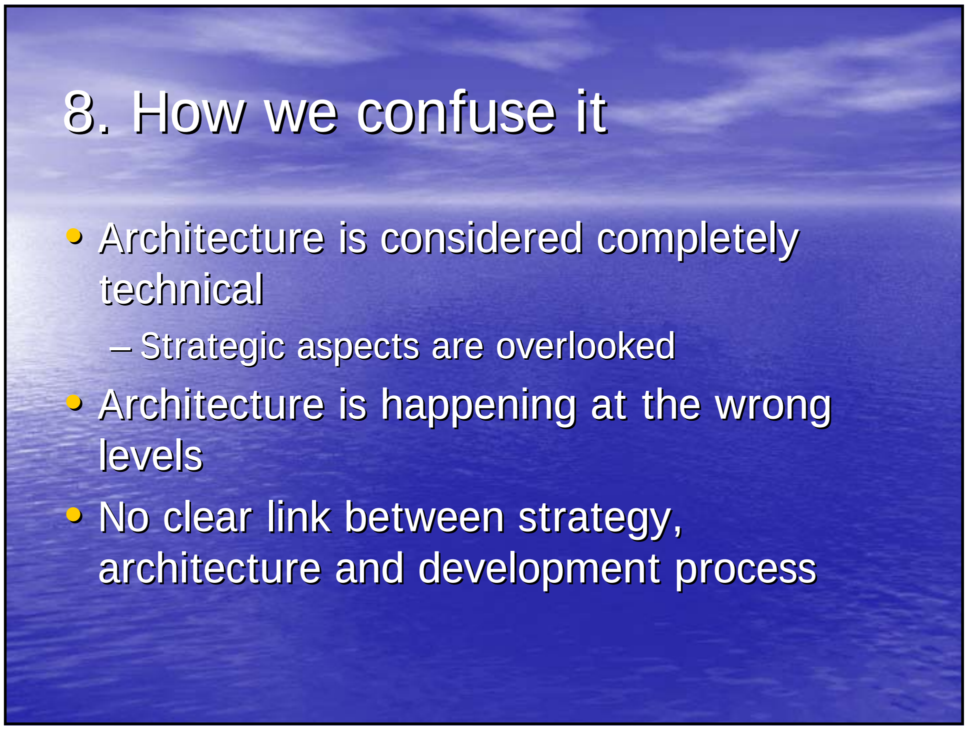
9. Technical Architectures

- Make technical architecture an actual artifact
- Create or find a template
  - Always link decisions with requirements
  - Connect coarse-grained components
  - Prototype
- Technology architectures can be generated at higher levels in the organization in contains-a relationships
- Technical evaluation should happen before the project is funded (by a technologist please)
  - Especially in cases where new technology is being considered



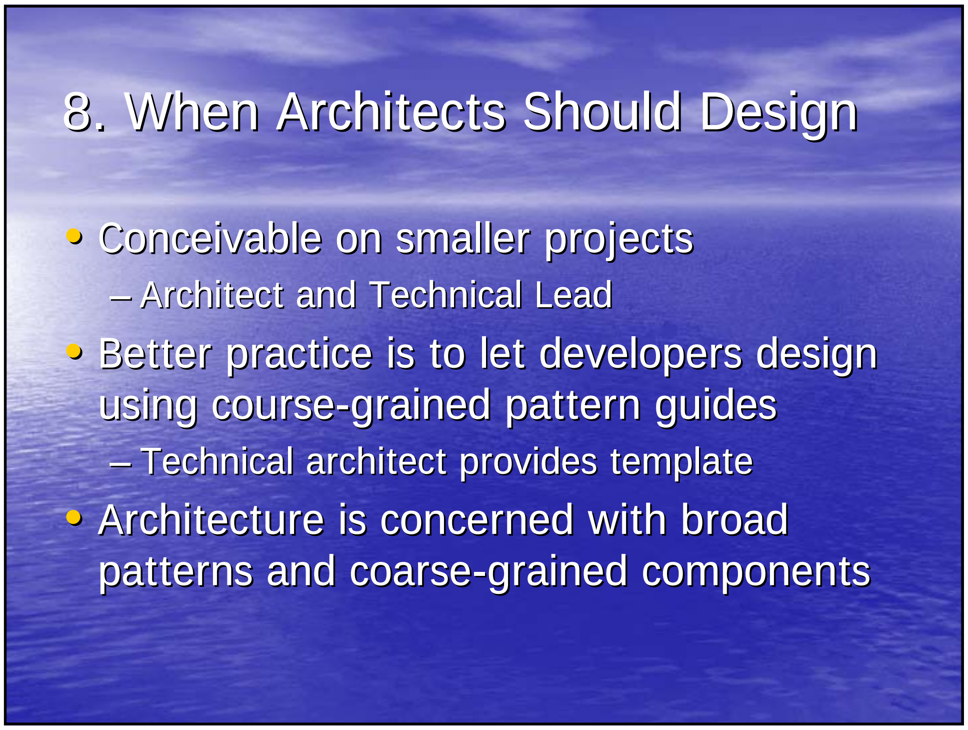
8. Architects do the Fun Stuff

- Basically architects do all the design
- Take the expertise out of development
- May be useful in off-shore where a “wall” exists between design and development
- May be useful on small projects



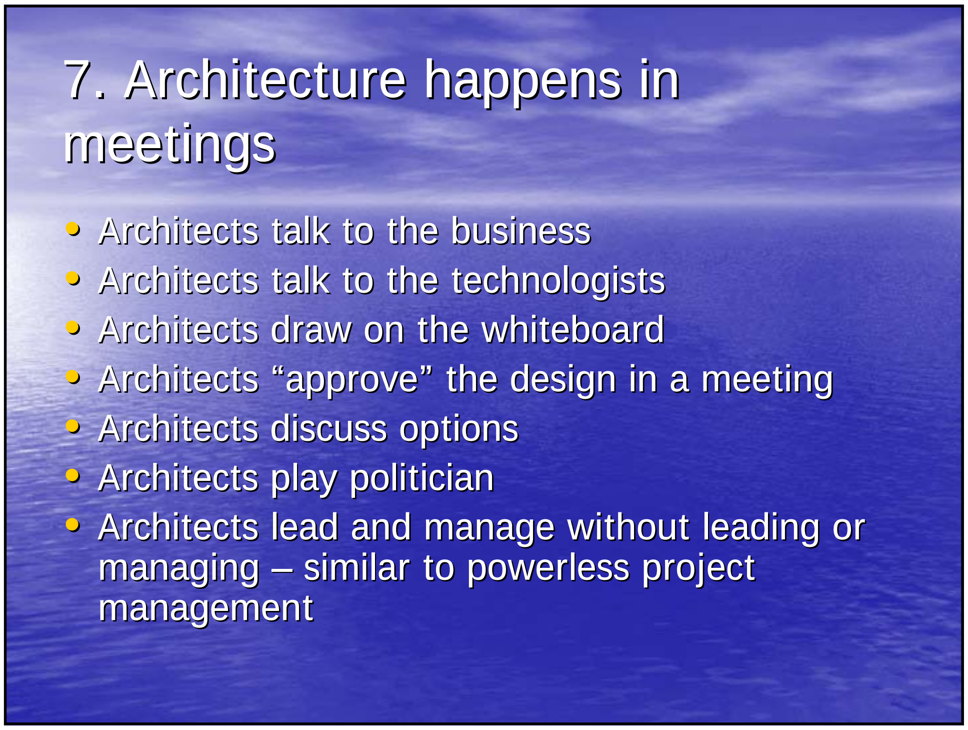
8. How we confuse it

- Architecture is considered completely technical
  - Strategic aspects are overlooked
- Architecture is happening at the wrong levels
- No clear link between strategy, architecture and development process



8. When Architects Should Design

- Conceivable on smaller projects
  - Architect and Technical Lead
- Better practice is to let developers design using course-grained pattern guides
  - Technical architect provides template
- Architecture is concerned with broad patterns and coarse-grained components



7. Architecture happens in meetings

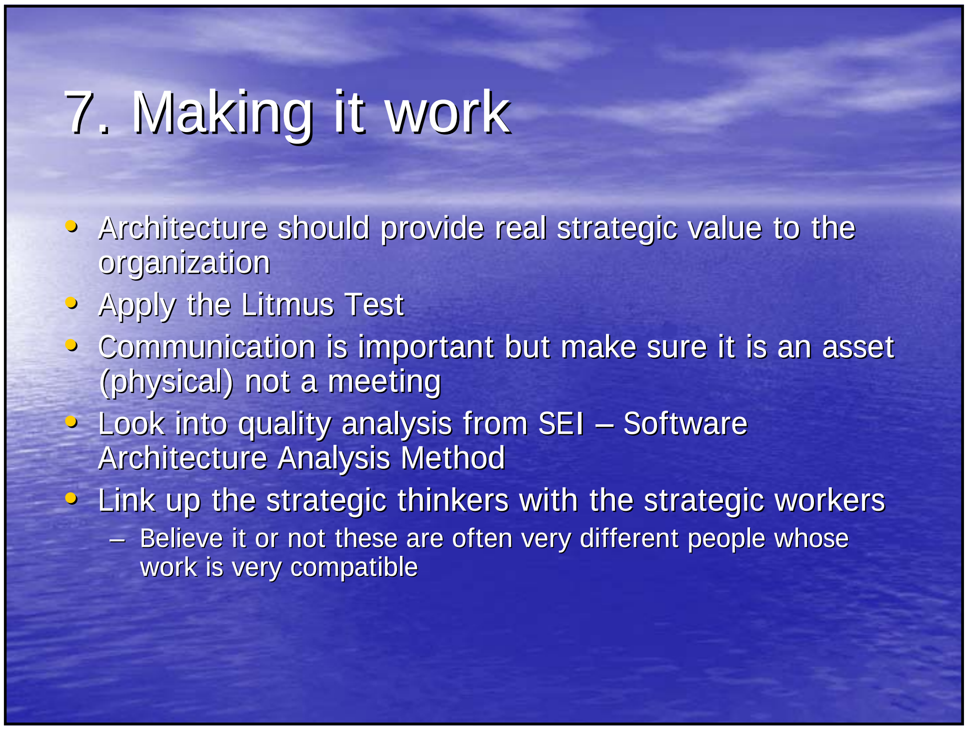
- Architects talk to the business
- Architects talk to the technologists
- Architects draw on the whiteboard
- Architects “approve” the design in a meeting
- Architects discuss options
- Architects play politician
- Architects lead and manage without leading or managing – similar to powerless project management

7. How we confuse it

- Architects should be middle-men between the business, the developers and management
- No set roles for architecture
- Friend of the CTO
- CxO understands the value of architecture but IT is pushing back
  - Architects are left with communication as their only value add

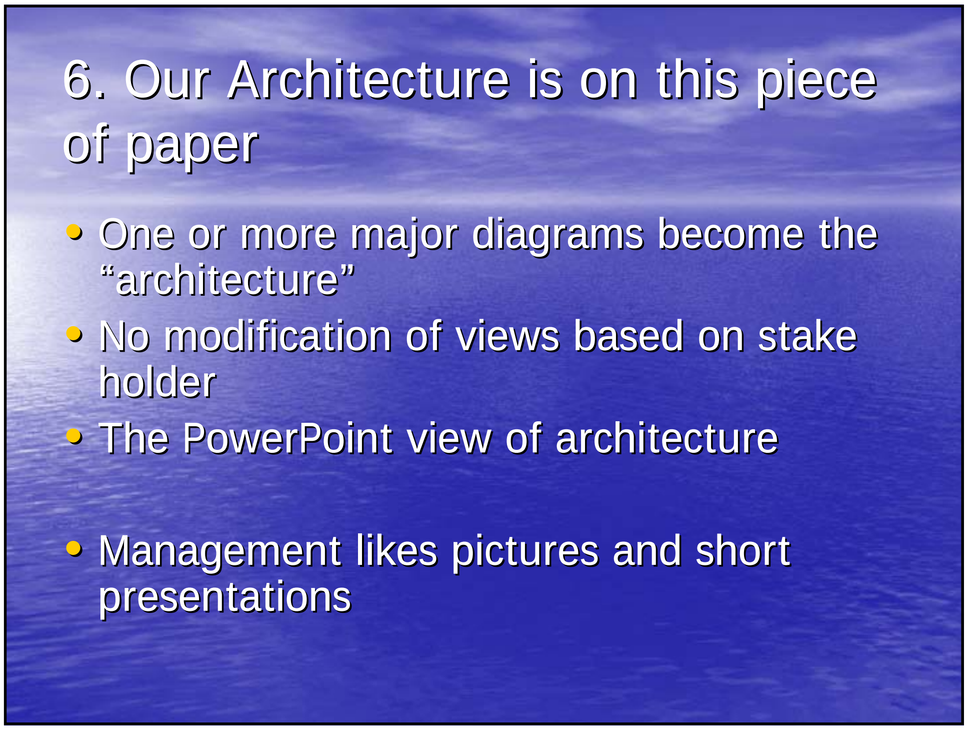
Architecture Litmus Test

- Any organization, which claims to have architecture, must have a sufficient set of valuable architectural assets that are understood and used by a large percent of IT.
- Any organization, which claims to be implementing architecture, must have a process and/or methodology which regularly generate valuable architectural assets.



7. Making it work

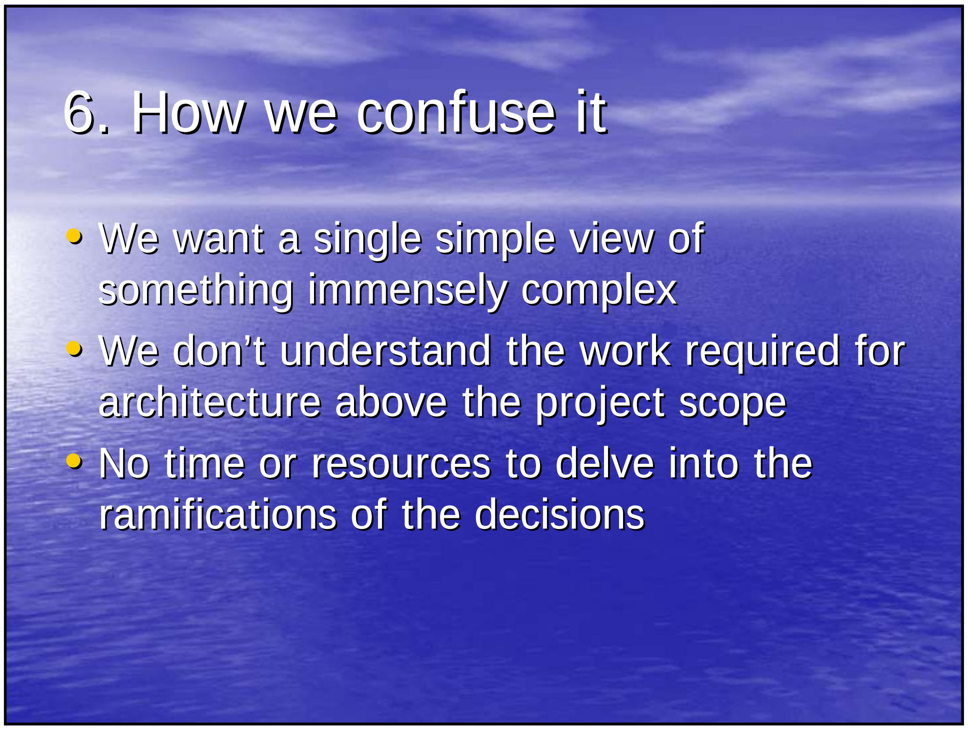
- Architecture should provide real strategic value to the organization
- Apply the Litmus Test
- Communication is important but make sure it is an asset (physical) not a meeting
- Look into quality analysis from SEI – Software Architecture Analysis Method
- Link up the strategic thinkers with the strategic workers
  - Believe it or not these are often very different people whose work is very compatible



6. Our Architecture is on this piece of paper

- One or more major diagrams become the "architecture"
- No modification of views based on stake holder
- The PowerPoint view of architecture

- Management likes pictures and short presentations

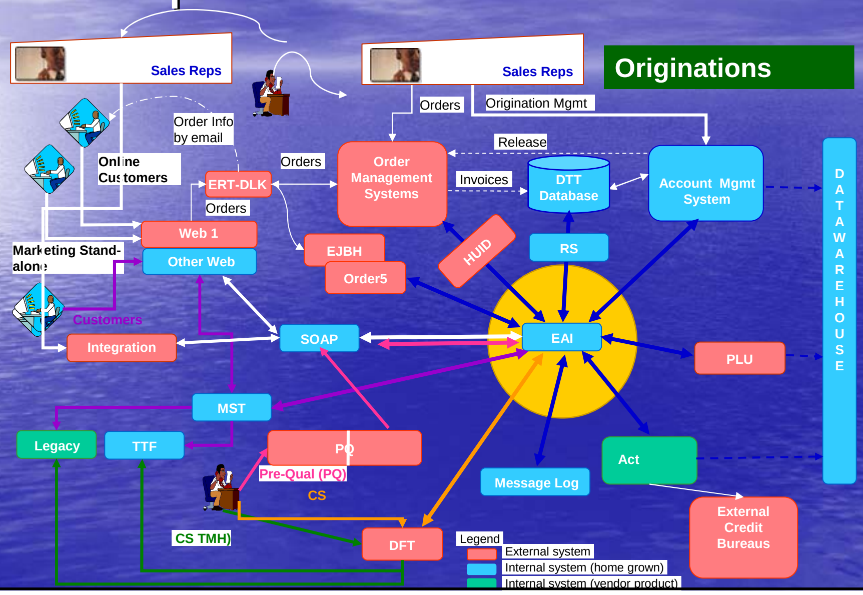


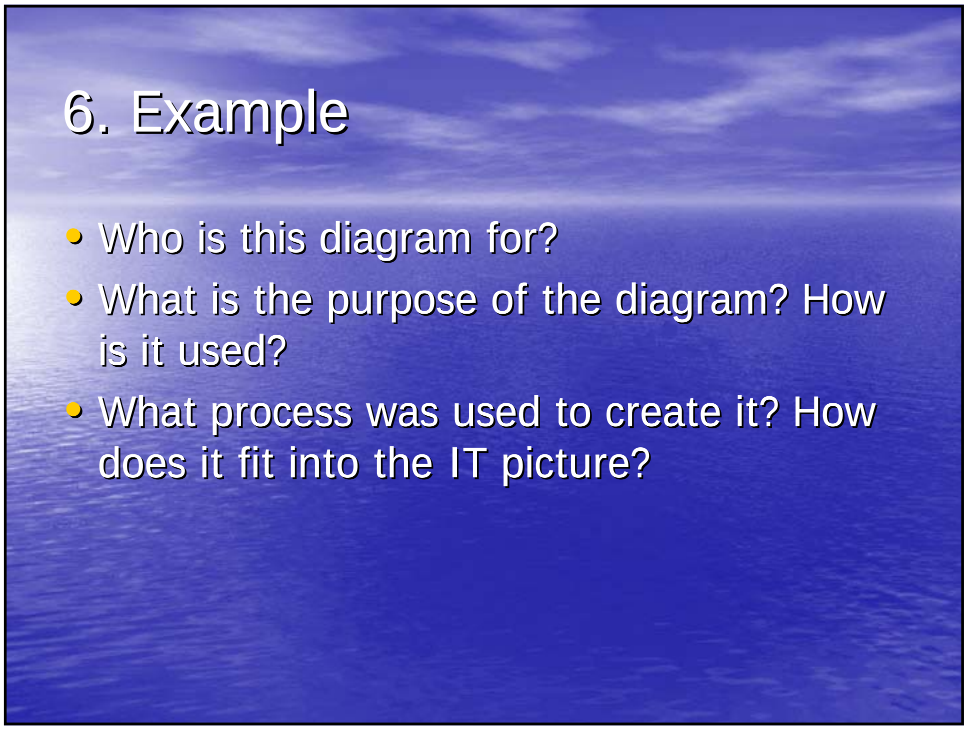
6. How we confuse it

- We want a single simple view of something immensely complex
- We don't understand the work required for architecture above the project scope
- No time or resources to delve into the ramifications of the decisions

6. Example

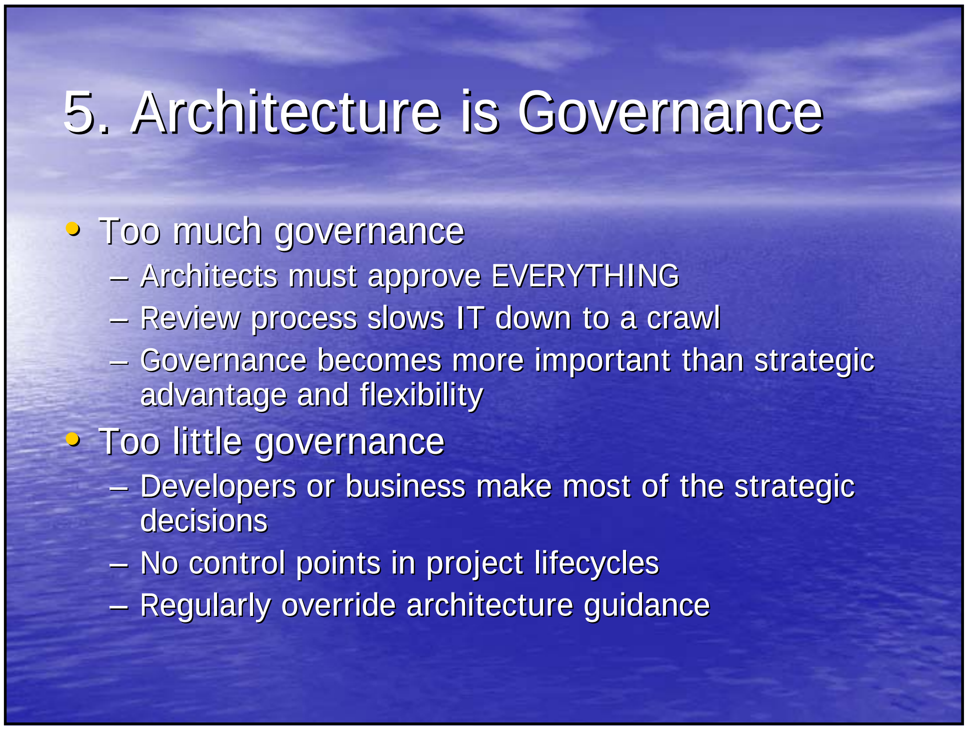
Architecture in 2002 – Names Changed to Protect the Innocent





6. Example

- Who is this diagram for?
- What is the purpose of the diagram? How is it used?
- What process was used to create it? How does it fit into the IT picture?



5. Architecture is Governance

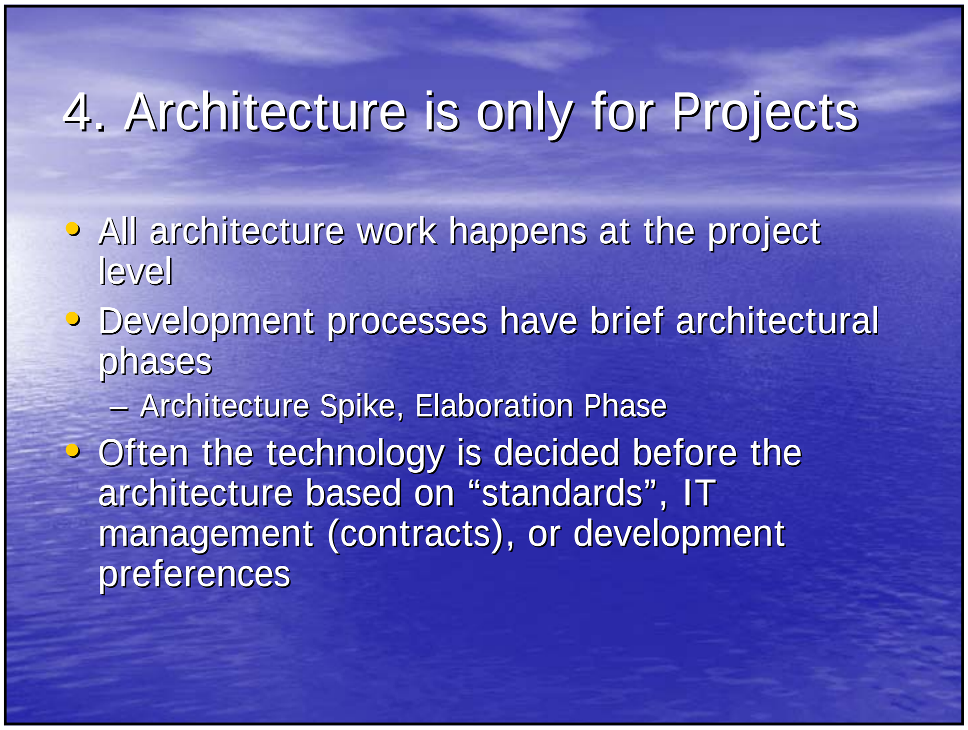
- Too much governance
  - Architects must approve EVERYTHING
  - Review process slows IT down to a crawl
  - Governance becomes more important than strategic advantage and flexibility
- Too little governance
  - Developers or business make most of the strategic decisions
  - No control points in project lifecycles
  - Regularly override architecture guidance

5. How we confuse it

- We don't realize architecture has a complete methodology
  - Best example is TOGAF
- Comes from a view that architects "bless" a project instead of work on it

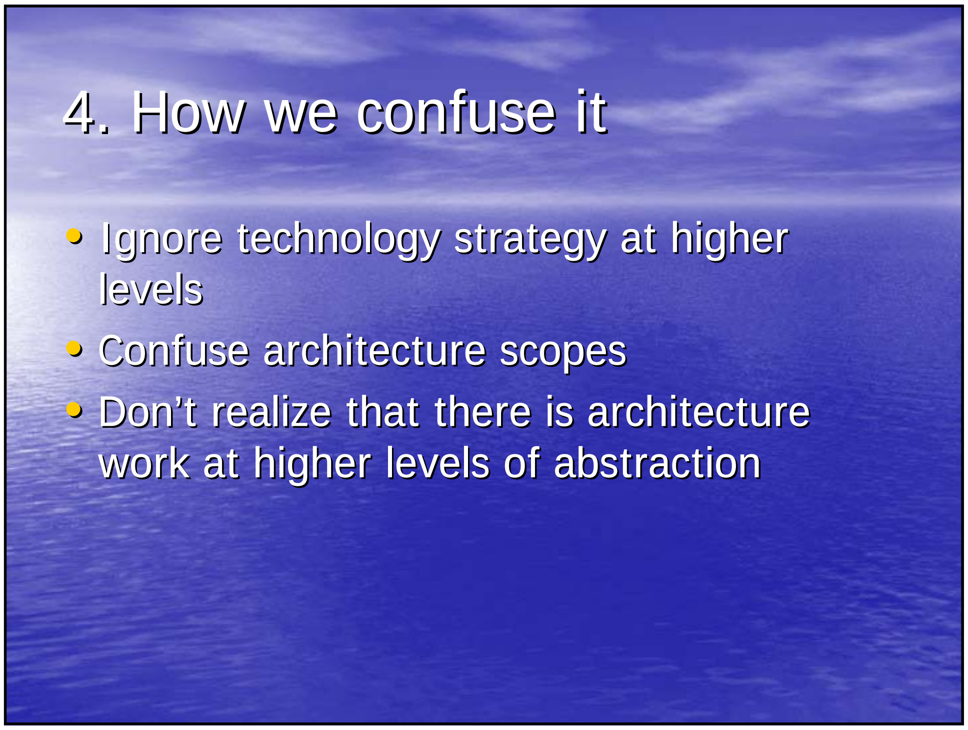
5. Making Governance Work

- Architects should WORK on the project
  - Optimally each project >= a given scope has an assigned architect
  - Technical architect should be consultant and generate documents for architecture review
  - Creates architecture buy-off without heavy weight review processes
- Architects on project should roll-up into a architecture organization
  - Review should happen one step above
  - No global review for all projects (relate to other organizational units)



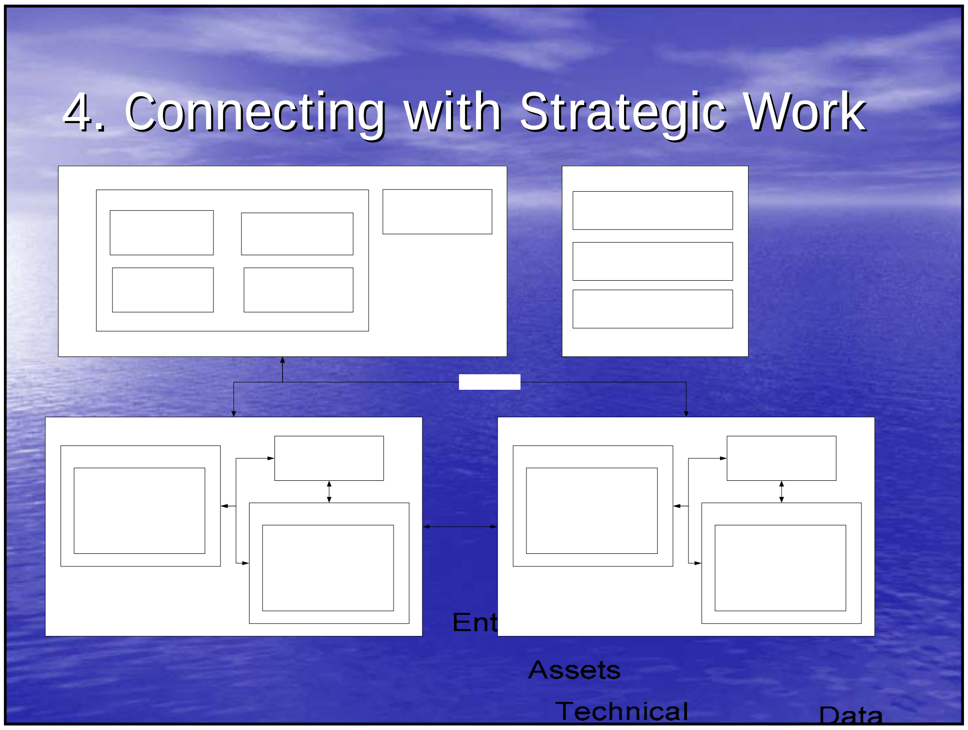
4. Architecture is only for Projects

- All architecture work happens at the project level
- Development processes have brief architectural phases
  – Architecture Spike, Elaboration Phase
- Often the technology is decided before the architecture based on "standards", IT management (contracts), or development preferences

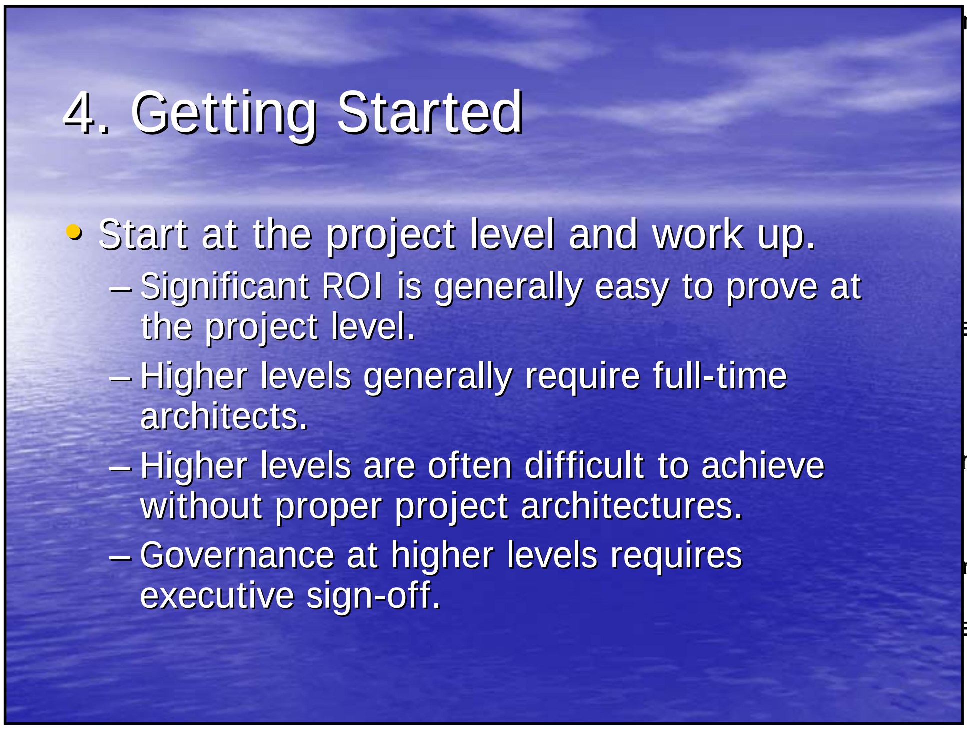


4. How we confuse it

- Ignore technology strategy at higher levels
- Confuse architecture scopes
- Don't realize that there is architecture work at higher levels of abstraction

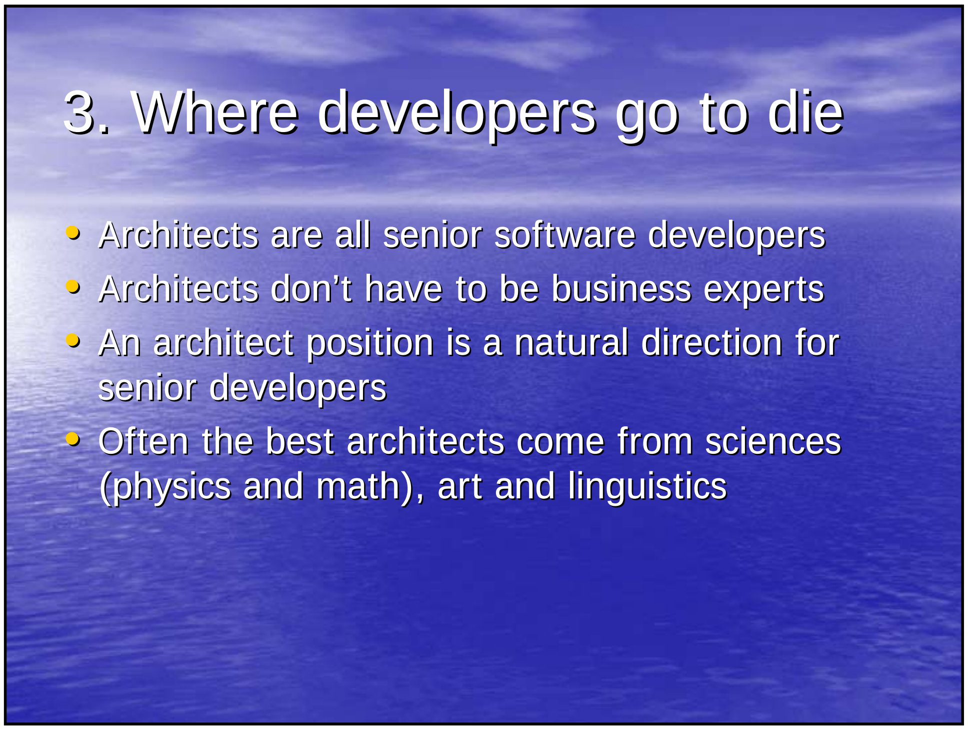


4. Connecting with Strategic Work



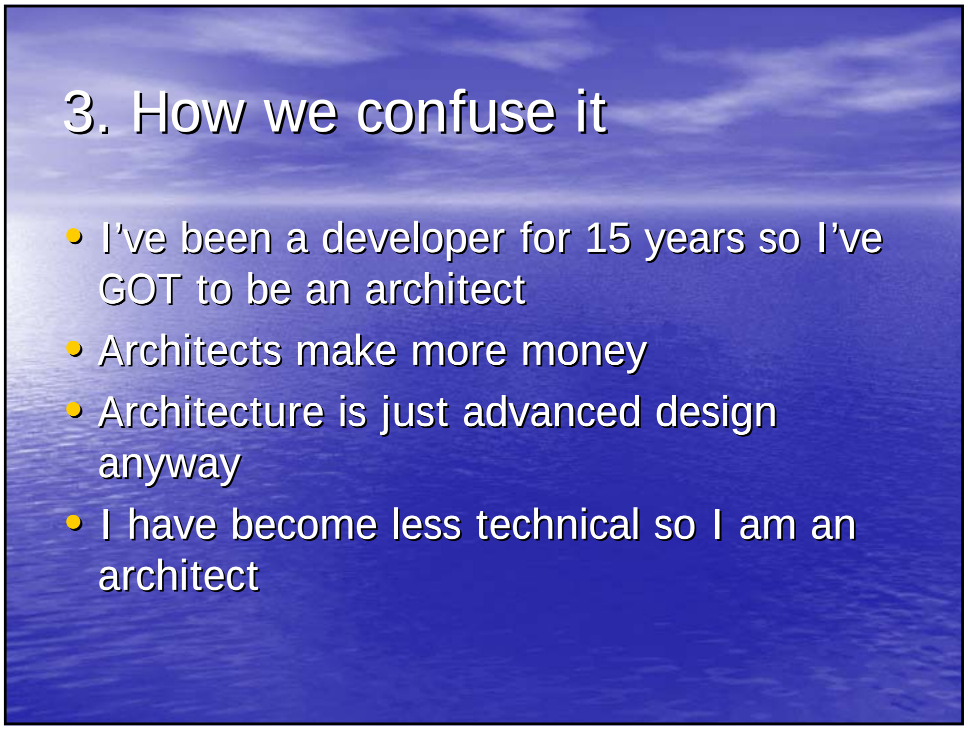
4. Getting Started

- Start at the project level and work up.
  - Significant ROI is generally easy to prove at the project level.
  - Higher levels generally require full-time architects.
  - Higher levels are often difficult to achieve without proper project architectures.
  - Governance at higher levels requires executive sign-off.



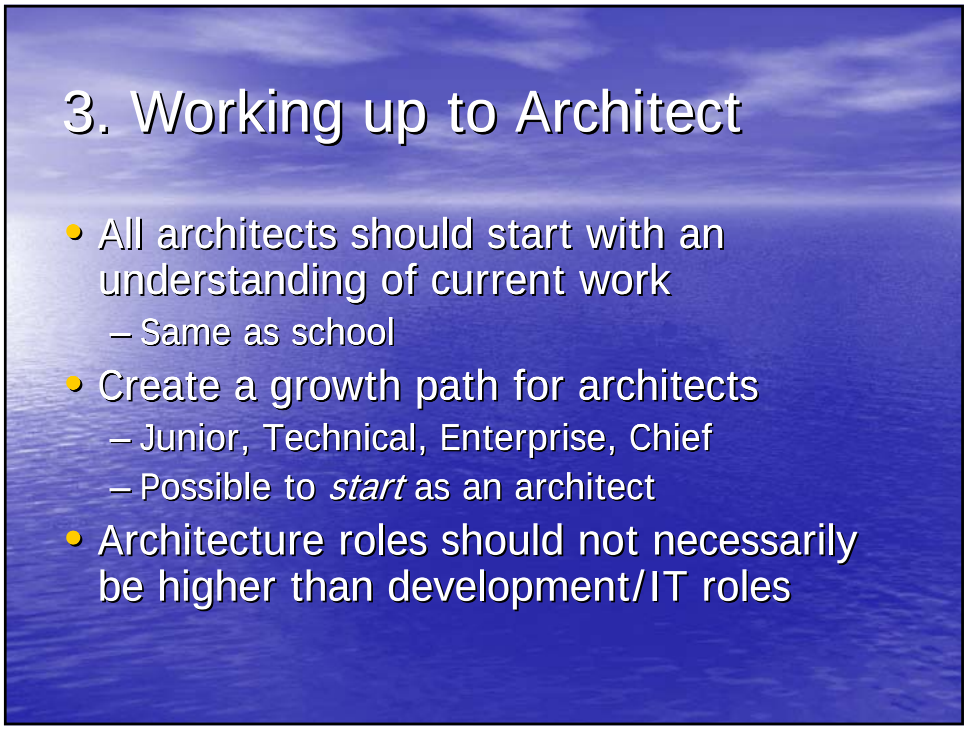
3. Where developers go to die

- Architects are all senior software developers
- Architects don’t have to be business experts
- An architect position is a natural direction for senior developers
- Often the best architects come from sciences (physics and math), art and linguistics



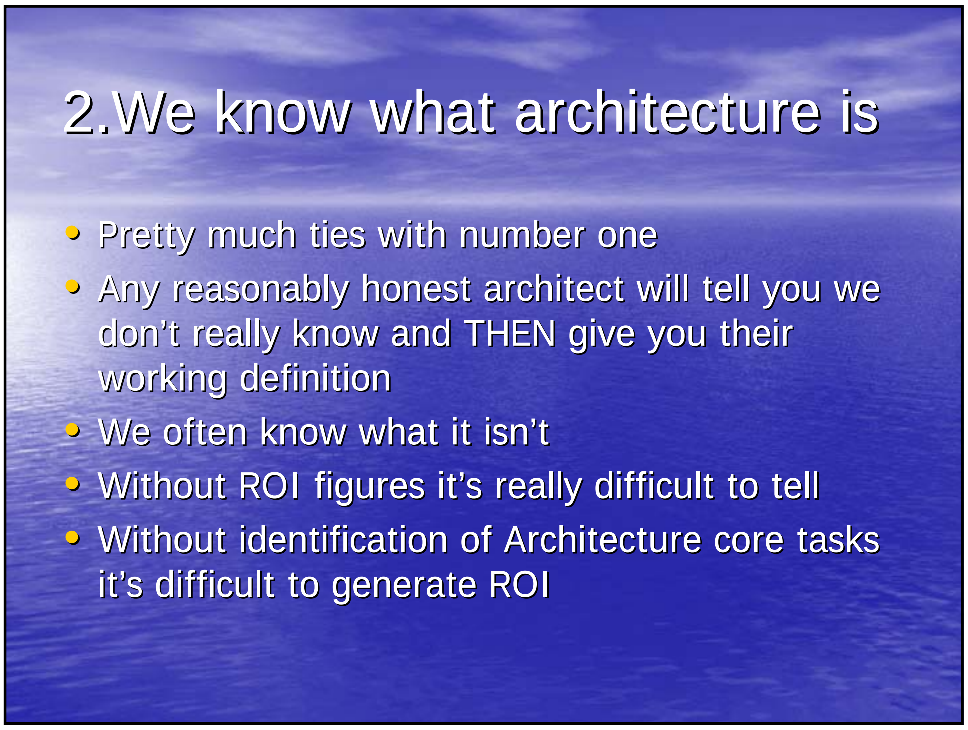
3. How we confuse it

- I’ve been a developer for 15 years so I’ve GOT to be an architect
- Architects make more money
- Architecture is just advanced design anyway
- I have become less technical so I am an architect



3. Working up to Architect

- All architects should start with an understanding of current work
  - Same as school
- Create a growth path for architects
  - Junior, Technical, Enterprise, Chief
  - Possible to *start* as an architect
- Architecture roles should not necessarily be higher than development/IT roles



2.We know what architecture is

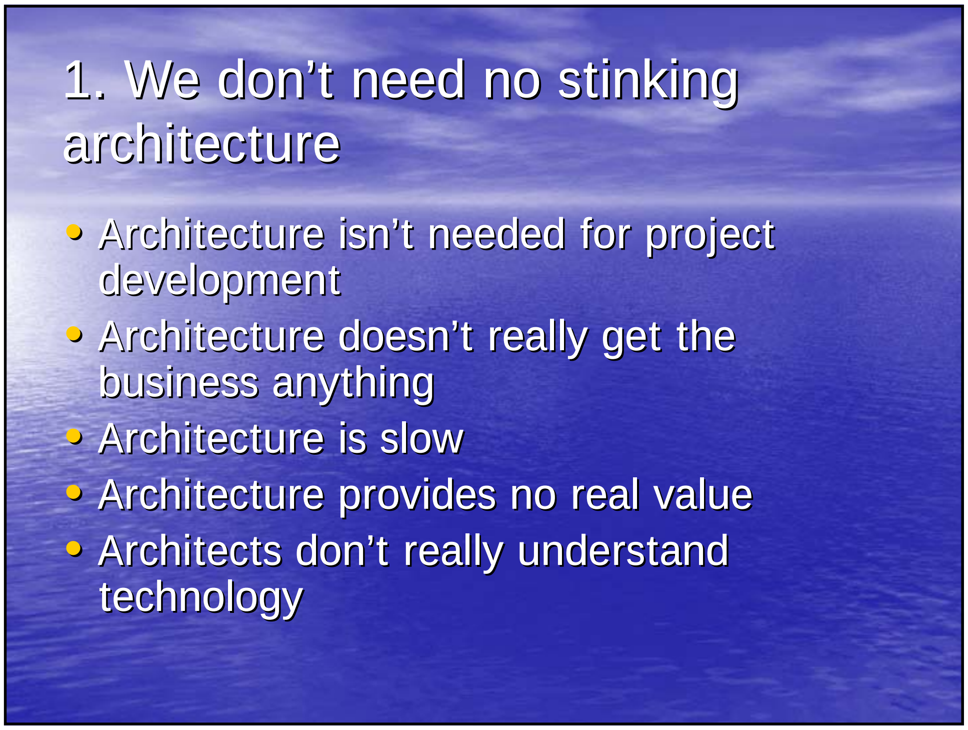
- Pretty much ties with number one
- Any reasonably honest architect will tell you we don’t really know and THEN give you their working definition
- We often know what it isn’t
- Without ROI figures it’s really difficult to tell
- Without identification of Architecture core tasks it’s difficult to generate ROI

2. How we confuse it

- No one knows what architecture is yet
  - SEI website contains 26 printed pages of definitions
  - While some commonality exists there is no single set of definitions
- We confuse strategic, technical, and process-oriented work
- Architecture focuses on abstract concepts and can easily be confused
- IT doesn't have the resources to do architecture research
- IT industry doesn't recognize it as separate
- Timelines, political pressures, resource constraints

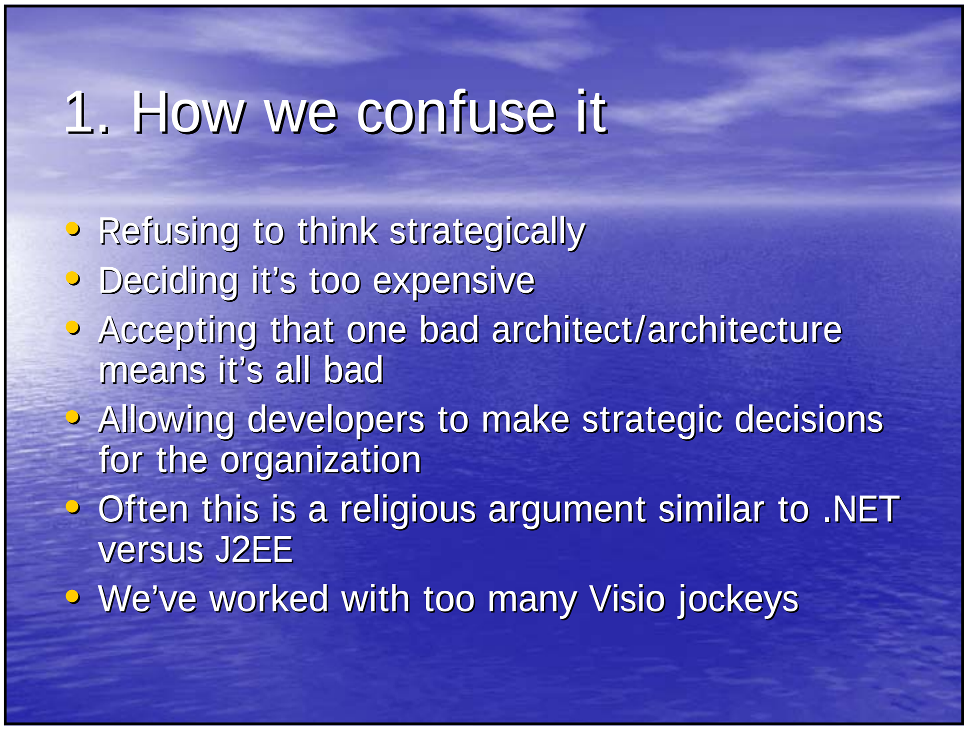
2. Avoiding local minima

- Always treat architecture work as work
  - Architects must deliver tangible value to the organization
- Never assume you know what architecture is
  - This one is for the practicing architects
- Read and compare everything you can get your hands on
  - There's really not that much out there yet
- Look into ways to calculate ROI from the architecture separate from the project
- Tie architects together across the organization
  - Causes churn in architecture



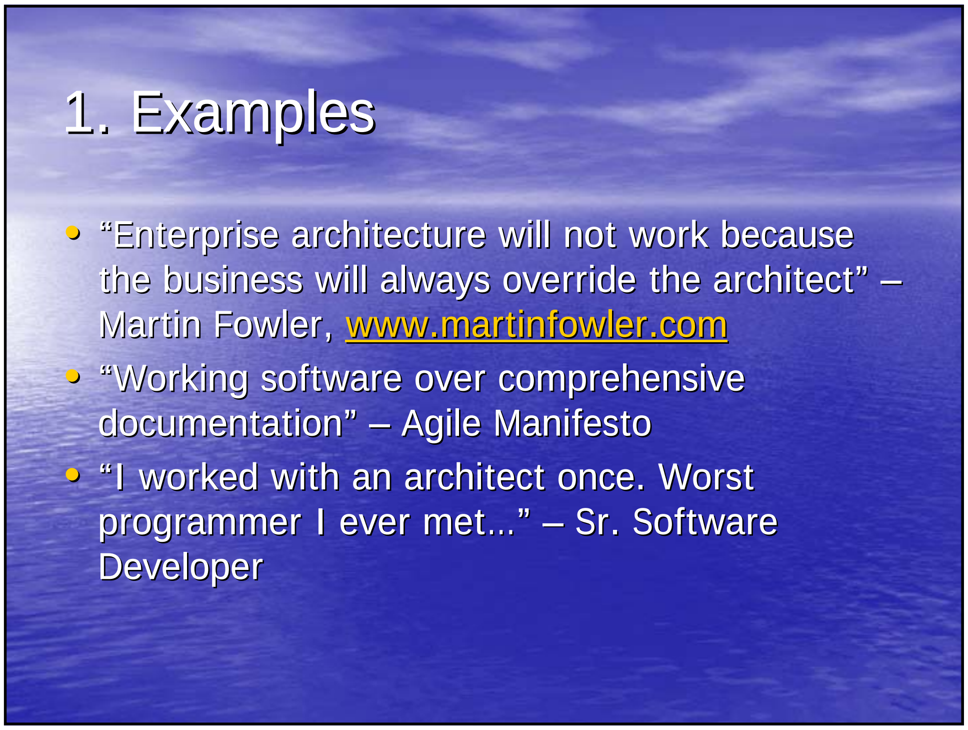
1. We don't need no stinking architecture

- Architecture isn't needed for project development
- Architecture doesn't really get the business anything
- Architecture is slow
- Architecture provides no real value
- Architects don't really understand technology



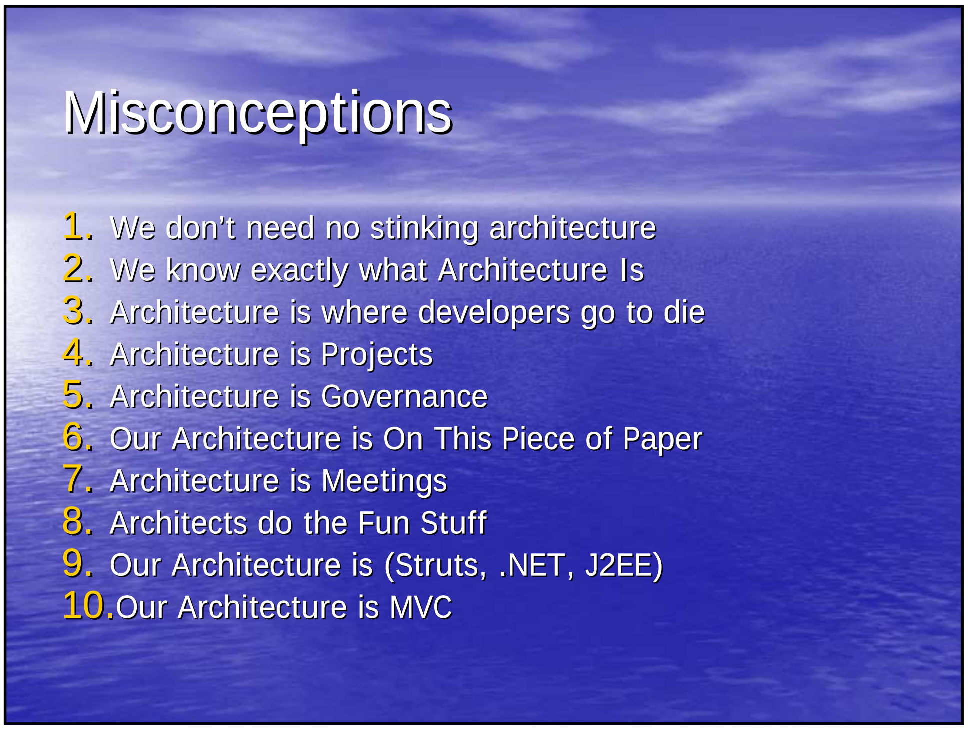
1. How we confuse it

- Refusing to think strategically
- Deciding it's too expensive
- Accepting that one bad architect/architecture means it's all bad
- Allowing developers to make strategic decisions for the organization
- Often this is a religious argument similar to .NET versus J2EE
- We've worked with too many Visio jockeys



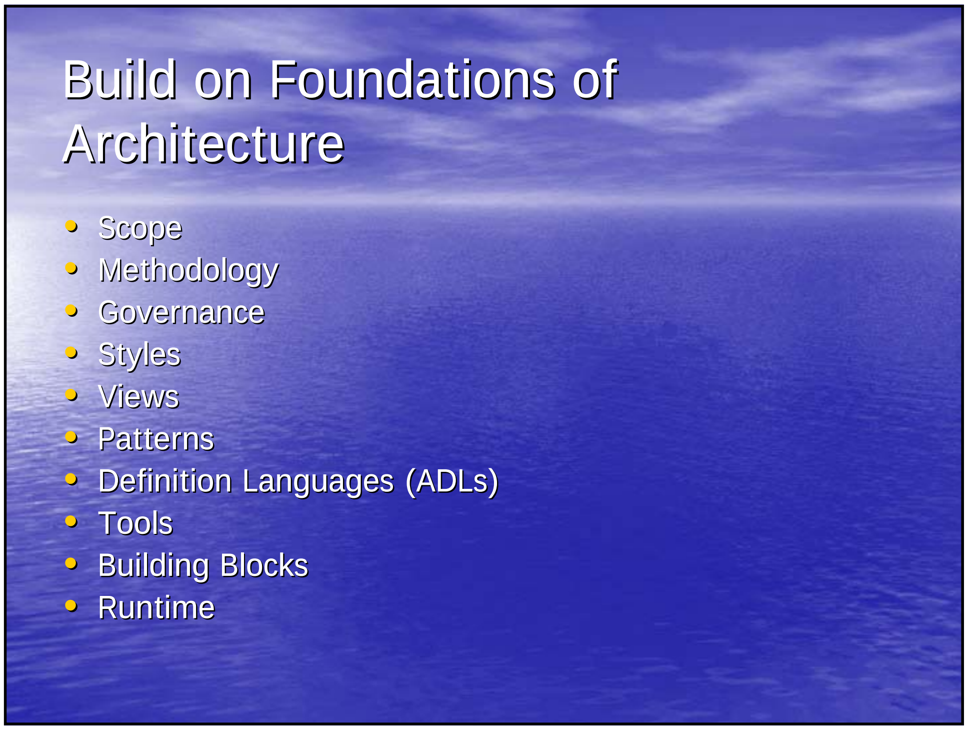
1. Examples

- “Enterprise architecture will not work because the business will always override the architect” – Martin Fowler, www.martinfowler.com
- “Working software over comprehensive documentation” – Agile Manifesto
- “I worked with an architect once. Worst programmer I ever met…” – Sr. Software Developer



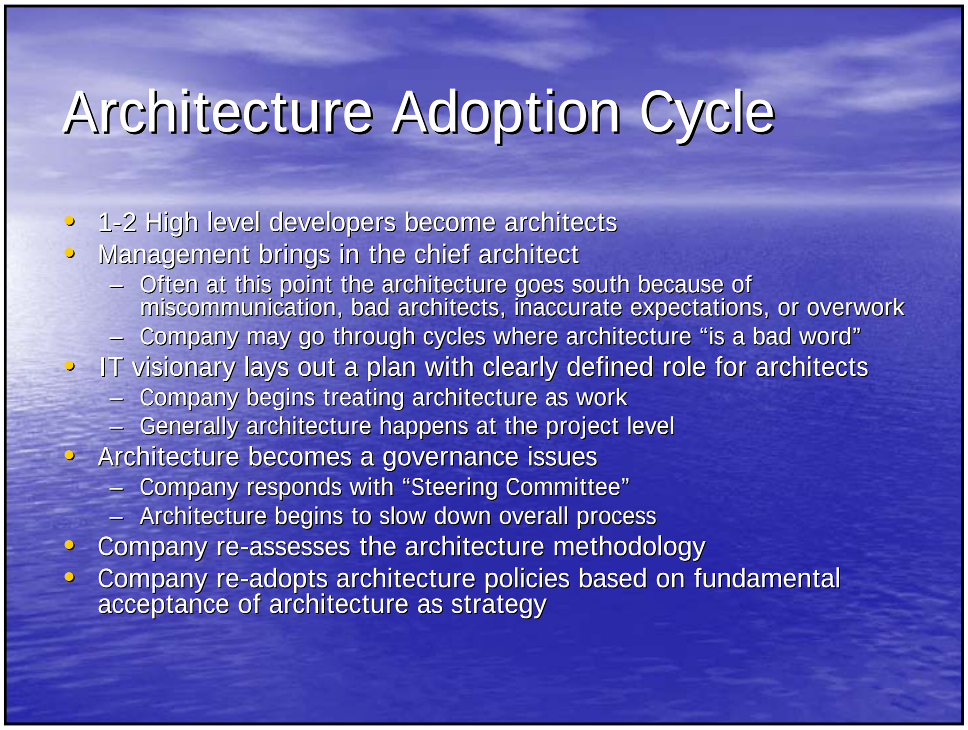
Misconceptions

1. We don’t need no stinking architecture
2. We know exactly what Architecture Is
3. Architecture is where developers go to die
4. Architecture is Projects
5. Architecture is Governance
6. Our Architecture is On This Piece of Paper
7. Architecture is Meetings
8. Architects do the Fun Stuff
9. Our Architecture is (Struts, .NET, J2EE)
10. Our Architecture is MVC



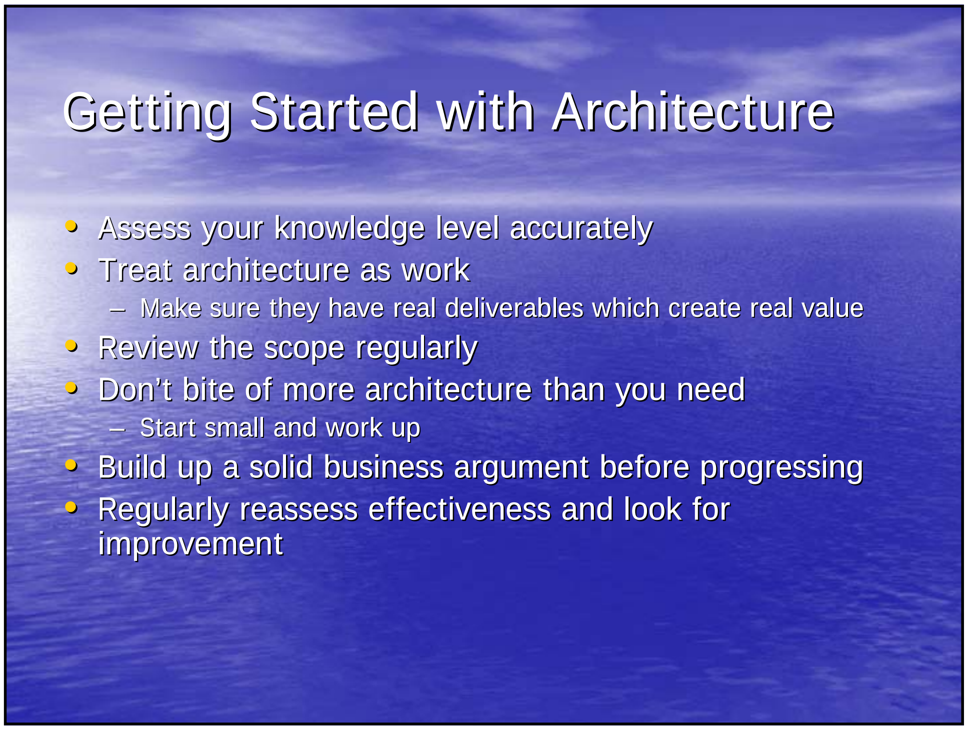
Build on Foundations of Architecture

- Scope
- Methodology
- Governance
- Styles
- Views
- Patterns
- Definition Languages (ADLs)
- Tools
- Building Blocks
- Runtime



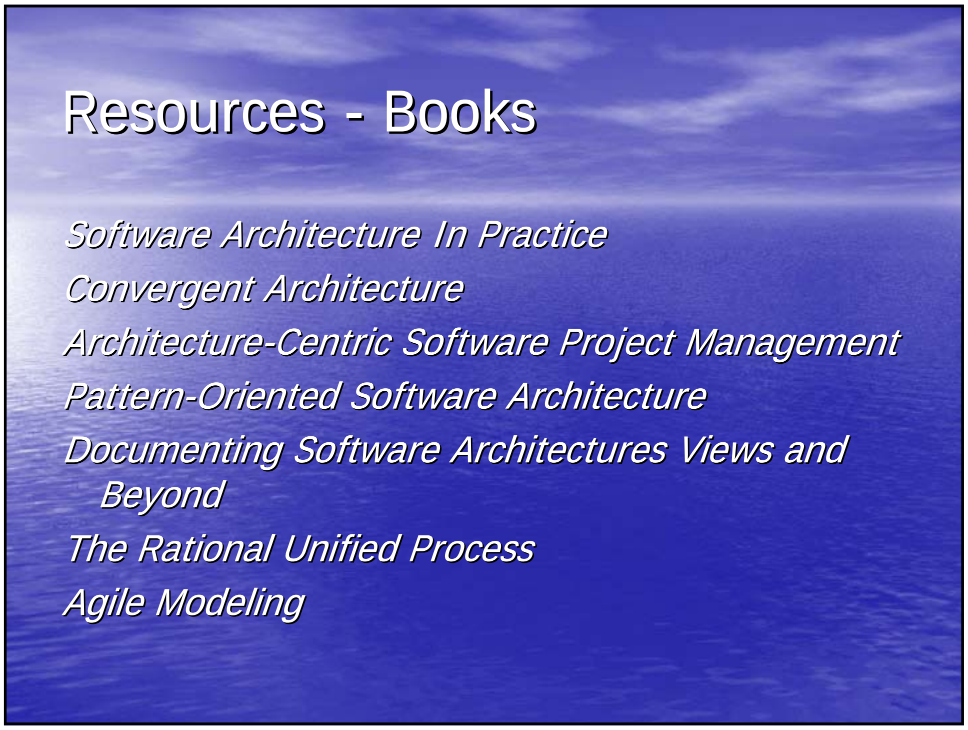
Architecture Adoption Cycle

- 1-2 High level developers become architects
- Management brings in the chief architect
  - Often at this point the architecture goes south because of miscommunication, bad architects, inaccurate expectations, or overwork
  - Company may go through cycles where architecture “is a bad word”
- IT visionary lays out a plan with clearly defined role for architects
  - Company begins treating architecture as work
  - Generally architecture happens at the project level
- Architecture becomes a governance issues
  - Company responds with “Steering Committee”
  - Architecture begins to slow down overall process
- Company re-assesses the architecture methodology
- Company re-adopts architecture policies based on fundamental acceptance of architecture as strategy



Getting Started with Architecture

- Assess your knowledge level accurately
- Treat architecture as work
  - Make sure they have real deliverables which create real value
- Review the scope regularly
- Don't bite of more architecture than you need
  - Start small and work up
- Build up a solid business argument before progressing
- Regularly reassess effectiveness and look for improvement



Resources - Books

Software Architecture In Practice

Convergent Architecture

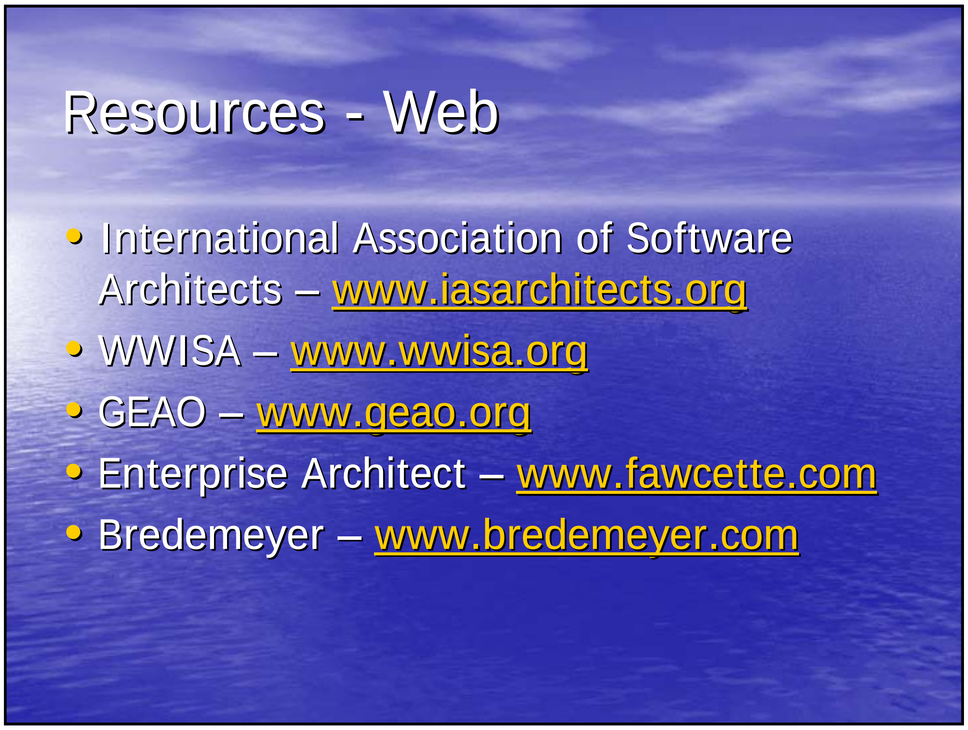
Architecture-Centric Software Project Management

Pattern-Oriented Software Architecture

Documenting Software Architectures Views and Beyond

The Rational Unified Process

Agile Modeling



Resources - Web

- International Association of Software Architects – www.iasarchitects.org
- WWISA – www.wwisa.org
- GEAO – www.geao.org
- Enterprise Architect – www.fawcette.com
- Bredemeyer – www.bredemeyer.com