



IBM Software Sales

# Architectural Thinking

The Art of Architecting



IBM Software Sales

# The soul of architecture

Photos by Pete Bouchard  
Music by Samuel Barber

# Agenda

- Who we are  
*The systems architect in the industry*
- What we do  
*The product of the systems architect is models*
- How we do that thing we do  
*The process breaks down large intractable problems into manageable chunks*

## A word on resources

- *The Art of Systems Architecting* 2<sup>nd</sup> Edition, Mark W. Maier and Eberhardt Rechtin. ISBN 0-8493-0440-7

This was a major source for this presentation. Many of the words on these charts come directly from this book. There was just no better way to say what needed to be said.

- *The Timeless Way of Building*, Christopher Alexander. ISBN 0-19-502402-8

This was the source for the spirit and soul of architecture work that I wanted to pass on. I believe I have quoted and attributed properly Alexander's words.

- *TOGAF – The Continuing Story*, Chris Greenslade, Chair of the Architecture Forum, The Open Group; founder and principal of Architecting-the-Enterprise, LLC

Charts used by permission for the purpose of this presentation only.



IBM Software Sales

# Who we are

Definitions



# Table of contents

## **Who we are**

What we do

How we do that thing we do

## Some definitions

- Architect
- Architecture
- Architecting

# Architect

- One who is engaged in the practice of creating systems architectures, primarily in a client centric role, through use of a methodical process employing a combination of artistic and engineering approaches.
- *“...an individual engaged in the process of architecting, regardless of domain, job title, or employer; by definition and practice both. From time to time an architect may perform engineering and an engineer may perform architecting – whatever it takes to get the job done.”*

*Maier and Rechtin*



# Architect or Engineer

- Architecting deals largely with unmeasurables using nonquantitative tools and guidelines based on practical lessons learned; that is, architecting is an *inductive* process.
- Engineering deals almost entirely with measurables using analytic tools derived from mathematics and the hard sciences; that is, engineering is a *deductive* process.
- ... every individual act of building is a process in which space is differentiated. It is not a process of addition, in which preformed parts are combined to create a whole, but a process of unfolding, ... in which the whole precedes the parts, and actually gives birth to them, by splitting.
- Today's systems architecting is driven by the same purposes as civil architecture – to create and build systems too complex to be treated by engineering analysis alone.

# Architecture

- A set of emerging attributes which connote the high level set of functions or attributes of a system, or system of systems.

A set of descriptions comprising a model that represents in abstract sets of systems components to be assembled as a system or system of systems.

- A structure, a process, or a profession

# Architecting

- *“The word 'architecting' refers only to the process. Architecting is an invented word to describe how architectures are created much as engineering describes how 'engines' and other artifacts are created.”*
- *“...architecting is characterized by dealing with ill-structured situations, situations where neither goals or means are known with much certainty. ...The architect seeks satisfactory and feasible problem-solution pairs ... “*
- *“... the weight will fall on heuristics and “art” during architecting”*
- *“... architecting is a continuing, day-to-day learning process.”*

**Maier and Rechtin (Preface)**

# The position of IT Architects

We know solutions to every problem? What's your problem?



How do I know what I want, when I don't know what you can do for me



## On the architect's role

*“The architect acts to translate between the problem domain concepts of the client and the solution domain concepts of the builder. Great architects go beyond the role of intermediary to make a visionary combination of technology and purpose that exceeds the expectation of the builder or client.”*

Maier & Rechtin

## Different problem solving at different levels of complexity

- At higher levels, architecting methods, experience based heuristics, abstraction, and integrated modeling must be called into play.
- Purely analytical techniques, powerful for the lower levels, can be overwhelmed at the higher ones.
- The basic idea is to simplify problem solving by concentrating on its essentials.
- Consolidate and simplify the objectives.
- Abstract the system at as high a level as possible, then progressively reduce the level of abstraction.

## A systems approach

- Focus on the system as a whole
  - What is required
  - What is feasible
- Systems are collections of different things which together produce results unachievable by the elements alone.
- “(The) ... “system functions” derive almost solely from the interrelationships among the elements, a fact that largely determines the technical role and principal responsibilities of the systems architect.”
- No one buys a system to own the system, they buy it to own what it does.
- The architecture must be grounded in the client's purpose.

## A purpose orientation

- Systems architecting is driven by a client's purpose
  - President Kennedy didn't say build me an Apollo 3 stage rocket and a Lunar excursion module.
- Useful purpose, affordable cost, acceptable period of time
- Useful purpose is predominant
- The architect works with the client and the builder on problems and solutions.



## A modeling methodology

- Modeling is the centerpiece of systems architecting – a mechanism of communication to clients and builders...
- The architect's deliverable is an architecture representation or model.
- An initial architecture is a vision. An architectural description is a set of specific models.

## Insights and heuristics

- A chess master does not think many moves ahead, they see a pattern on the board and have the insight and experience to know the outcome.
- Heuristics are codified succinct expressions from lessons learned through your own or other's experience. Heuristics are a key tool of the systems architect.

**Success comes from wisdom.**

**Wisdom comes from experience.**

**Experience comes from mistakes.**

- Start a notebook or a place to write down your heuristics. Share them with others.

## Builder-architected systems

- Systems architecting occurs in the context of an acquisition process
- This is a “form-first” architectural approach, with technology-driven systems rather than purpose driven systems.
- Begins with a builder-conceived architecture in mind rather than with a set of client-accepted purposes.
- The customer judges the value of the product after it has been developed and produced.
- This calls into practice the use of a prototype demonstration.
- Uncertainty of end purpose is a major risk.
- Form-first can often produce a solution looking for a problem

## What type of people are we ?

- A study at USC confirms David Keirse's assessment
- The NT personality type is ideal for architecture team membership.
  - Abstract in word usage – speak of what can only be seen conceptually
  - Utilitarian in tool usage – usefulness more important and acceptability
  - Strategic intellect – want to increase the efficiency of their systems
- The INTP turns out to be more suited to the role
  - Introverted – think where you get your energy from
  - Probing – tendency to look around for a variety of options
- MBTI and KTS (<http://keirse.com/>) show tendencies registered on continua – we all function in areas different from our type.

# The Introvert Advantage

- Keep energy inside, making it difficult for others to know them.
- Be absorbed in thought
- Hesitate before speaking
  - Come up with the perfect thing to say 30 minutes later
  - If it's an area of interest they will talk all day
  - Clutch under pressure – your mind goes blank
- Avoid crowds and seek quiet
- Lose sight of what others are doing
- Proceed cautiously in meeting people and participate only in selected activities
- Not offer ideas freely; may need to be asked their opinion
- Get agitated without enough time alone or undisturbed
- Reflect and act in a careful way
- Not show much facial expression or reaction

## Some facts of life

- Introverts are about 1/3 of the population (INTP is about 1% of the population)
- In building a team, finding complementary personalities is better than finding similarity.
- When friction occurs because of other's strengths remember why you wanted them on the team in the first place.



IBM Software Sales

# What we do

Activities and tools

## TTWOB Architect

- Architects are observers of behavior.
- “...a place is given its character by certain patterns of events that keep on happening there...”
- Architects create space where these patterns of behavior can happen, flourish, and be generative – or as Alexander puts it, be alive.
- From this activity a language of patterns emerges permitting endless possibilities of creation.
- Our role is to observe the behavior and pattern language in our customers, and to expand their vocabulary.



# Defining an IT Architecture

- ANSI/IEEE Standard 1471-2000 Recommended Practice for Architectural Description

Conceptually an IT Architecture is

The fundamental organization of a system,  
embodied in its components,  
their relationships  
to each other  
and the environment,  
and the principles governing its design and evolution.

Practically it is represented in Architectural Descriptions from the viewpoints of the Stakeholders

## Some more ANSI/IEEE definitions

- **Architect:**  
the person, team, or organisation responsible for systems architecture
  
- **Architecting:**  
the activities of defining, documenting, maintaining, improving and certifying proper implementation of an architecture.
  
- **Architectural description**  
a collection of products to document an architecture.

## Architecture views - definitions

- **System Stakeholder:**
  - an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system
- **View:**
  - a representation of a whole system from the perspective of a related set of concerns
- **Viewpoint: (a schema of the information in a view)**
  - acts as a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis

## Architecture view

- Description of the architecture from the viewpoint of a specific stakeholder
- The main mechanism of communication between the architect and the stakeholder
- Used to ensure accuracy of understanding of the current system
- Used to ensure the architecture meets the need of each stakeholder
- The collection of views comprises the description of the architecture

# What is an Architectural Framework?

- An architectural framework is a tool for:
  - Designing a broad range of architectures
  - Assisting the evaluation of different architectures
  - Selecting and building the right architecture for an organization
- It embodies best practice and acknowledged wisdom
- It presents a set of services, standards, design concepts, components and configurations
- It guides the development of specific architectures

# What is an Architectural Framework?

- Use of a framework leads to:
  - The use of common principles, assumptions and terminology
  - The development of information systems with better integration and interoperability, especially with respect to issues that affect the whole enterprise
- **WARNING!**
  - A framework does not make architectural design an automatic process
  - It is a valuable aid to experienced and knowledgeable IT Architects

# The Open Group Architectural Framework

- 2001: TOGAF - version 7
  - New sections on Architecture Patterns, Architecture Principles, Architecture Compliance Reviews
  - Significant additional material on Business Scenarios
  - Comparisons of TOGAF with other frameworks
  - Further integration of IEEE Std 1471-2000 into TOGAF
  - Metis model of the TOGAF ADM
  - Positioning of TOGAF relative to enterprise architecture
  
- 2002: TOGAF - Version 8

**Technical Edition**

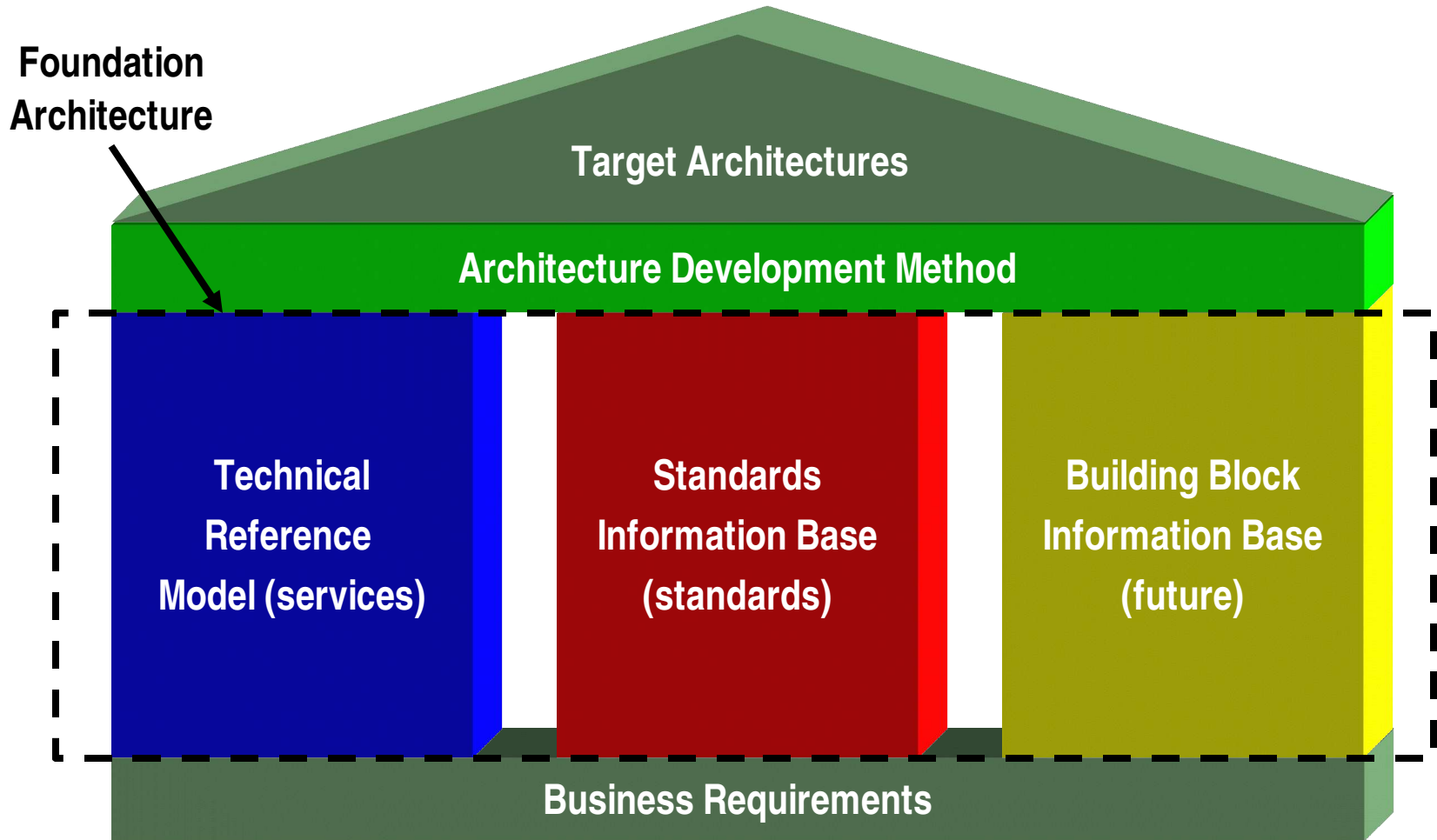
**Enterprise Edition**

# TOGAF consists of

- An Architecture Development Method (ADM)
  
- Foundation Architecture
  - A Technical Reference Model (TRM)
  - A Standards Information Base (SIB)
  - Building Blocks Information Base (BBIB)
  
- Resource Base contains advice on:
  - Architecture views                      Business scenarios
  - IT Governance                            Architecture patterns
  - ADL    Case studies
  - TABB                                        Architecture principles
  - Architecture contracts                  ...



# More about TOGAF

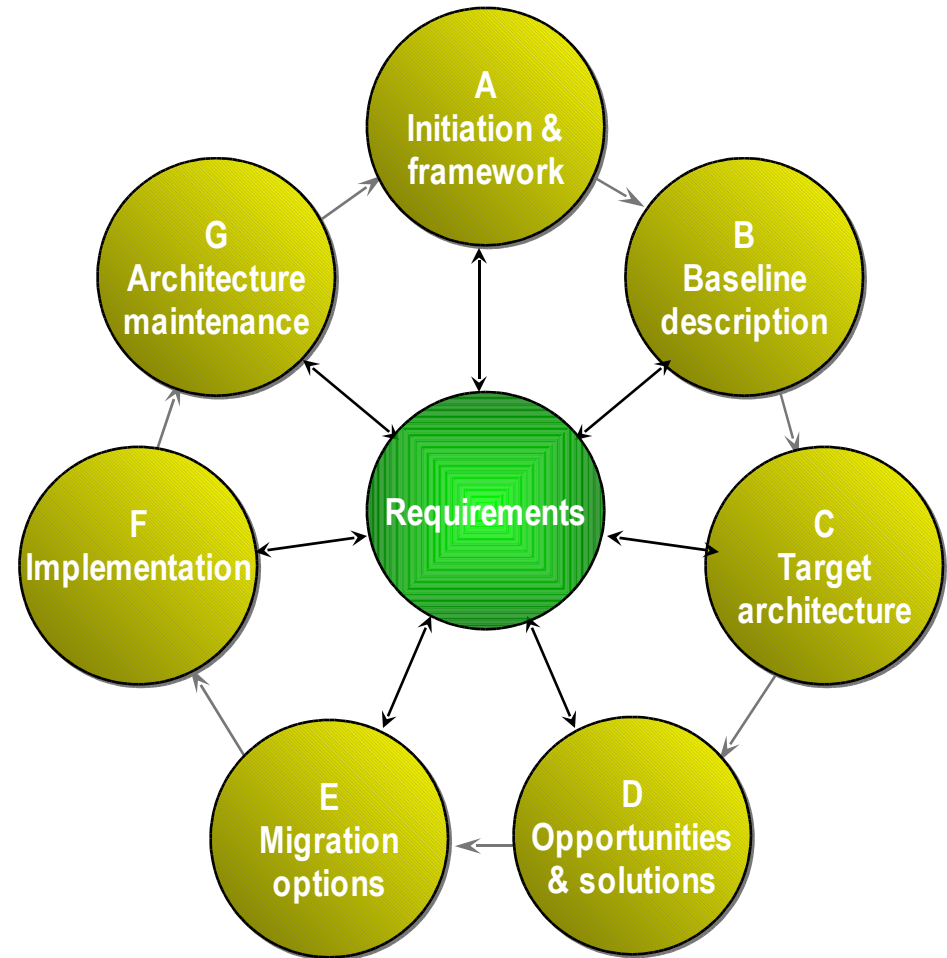


# Architecture Development Method

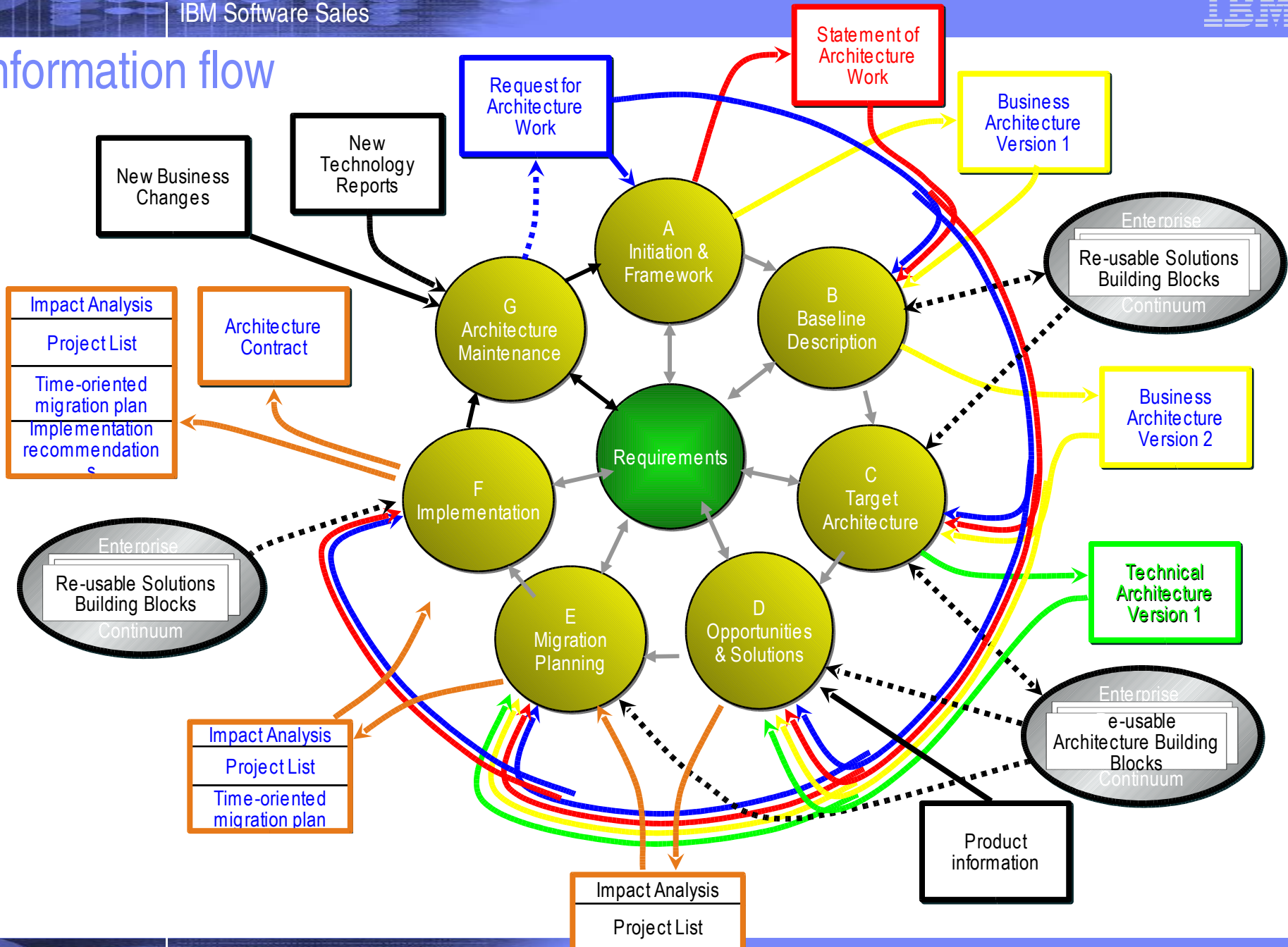
- Start with a foundation architecture
- Follow the phases of the ADM

an organization-specific architecture

more reusable building block assets in  
the Enterprise Continuum



# Information flow



# The Zachman Framework

	<i>What?</i> Data	<i>How?</i> Function	<i>Where?</i> Network	<i>Who?</i> People	<i>When?</i> Time	<i>Why?</i> Motivation	
<i>Planner's Viewpoint</i> Contextual	Validated principles	Validated principles	Validated principles		Business goals	Refined Business Business	Scope
<i>Owner's Viewpoint</i> Conceptual	Data model managem't view	Business processes	Gap analysis results		Data model managem't view	Business goals & drivers	Enterprise Models
<i>Designer's Viewpoint</i> Logical	Common application services		Technology architecture version 0.2		Application information Interop. view	Constraints on technology architecture	Systems Models
<i>Builder's Viewpoint</i> Physical	Technology architecture version 0.1		Technology architecture version 0.3		Technology architecture version 0.4		Technology Models
<i>Sub-contractor's Viewpoint</i> Out-of-context							Detailed Representations
Functioning Enterprise							Actual Systems

## What is the Enterprise Edition?

- An Enterprise Architecture is the technical foundation of an effective IT strategy

- It consists of four types of architecture:

Business architecture

Information system architectures

    Data or information architecture

    Application architecture

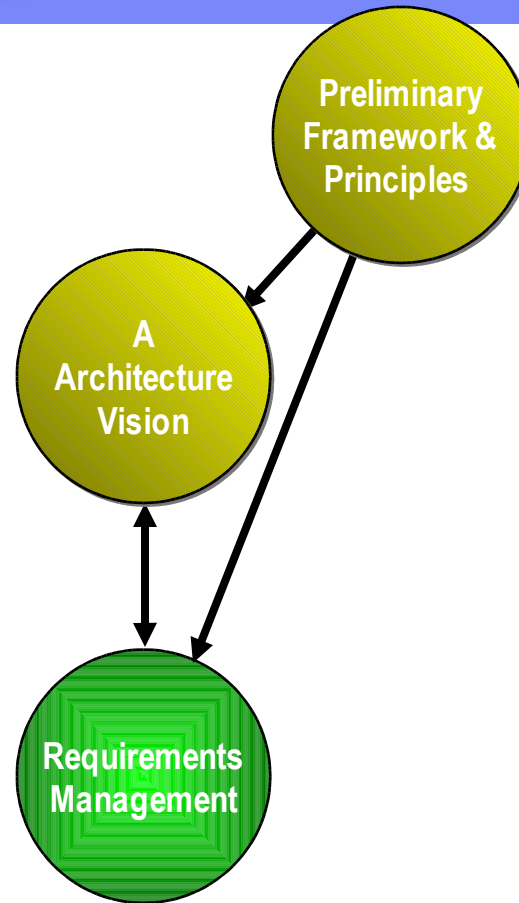
Technology architecture

**TOGAF 8  
Enterprise Edition**

**TOGAF 7 Technical Edition**

- All these are related

# The Enhanced ADM



## Preliminary steps (1)

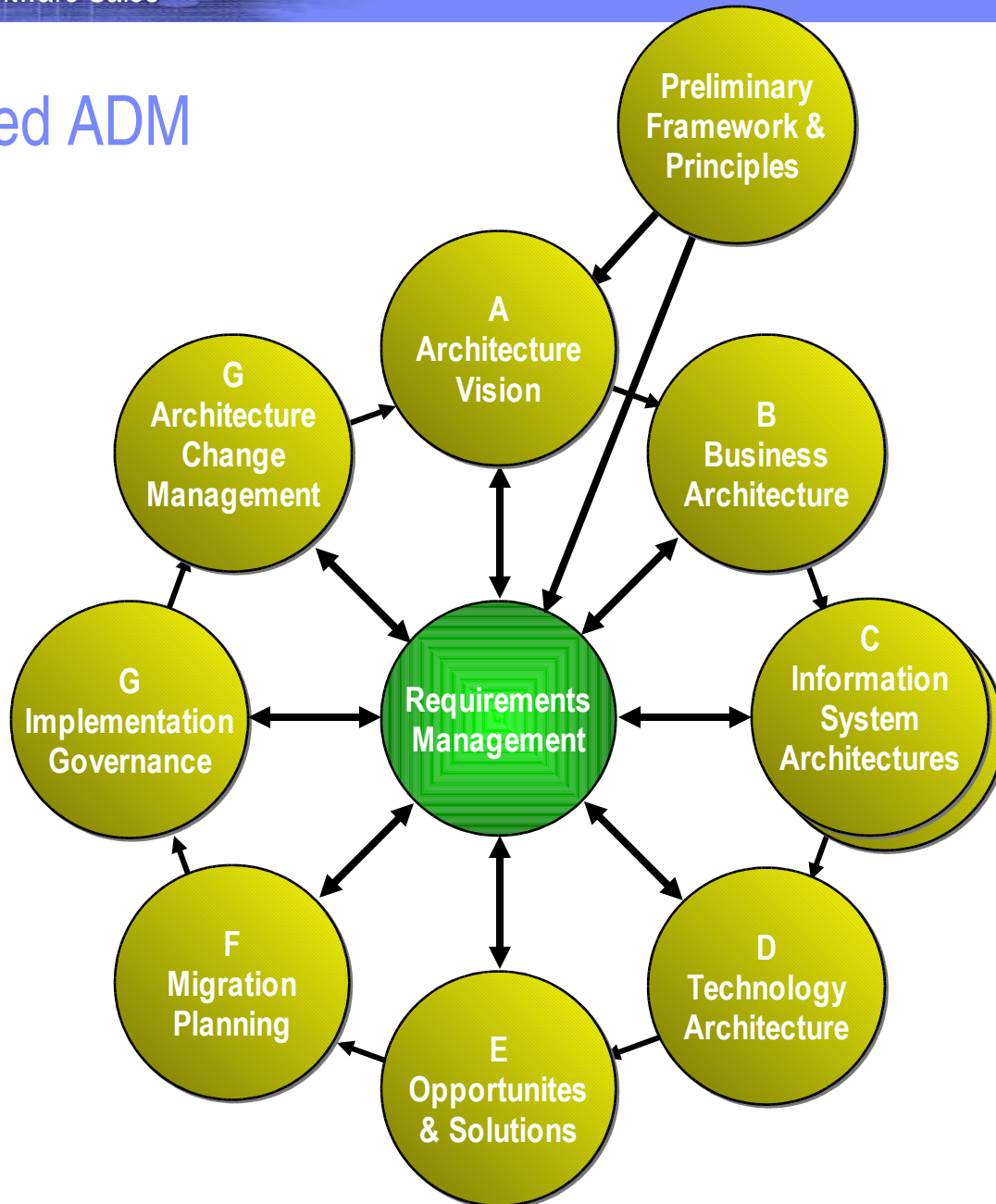
- Getting the buy-in
  - The most difficult stage
  - The most important stage
  
- Establishing the Architectural Framework
  - Customizing, configuring and selecting options suitable for the organization
  
- Providing a foundation for the framework by establishing:
  - Architecture principles – to guide all future work on all future architectures
  - IT Governance
  - Architecture compliance procedures

## Preliminary steps (2)

- Integrating the framework with existing procedures
  - Preserving tried, trusted, or mandated procedures
- Choosing the tools
- Training the staff - architects and others
- Creating a repository for Building Blocks (BBIB?)
- Monitored pilot project
  - Built-in leeway to allow for familiarization and fine-tuning

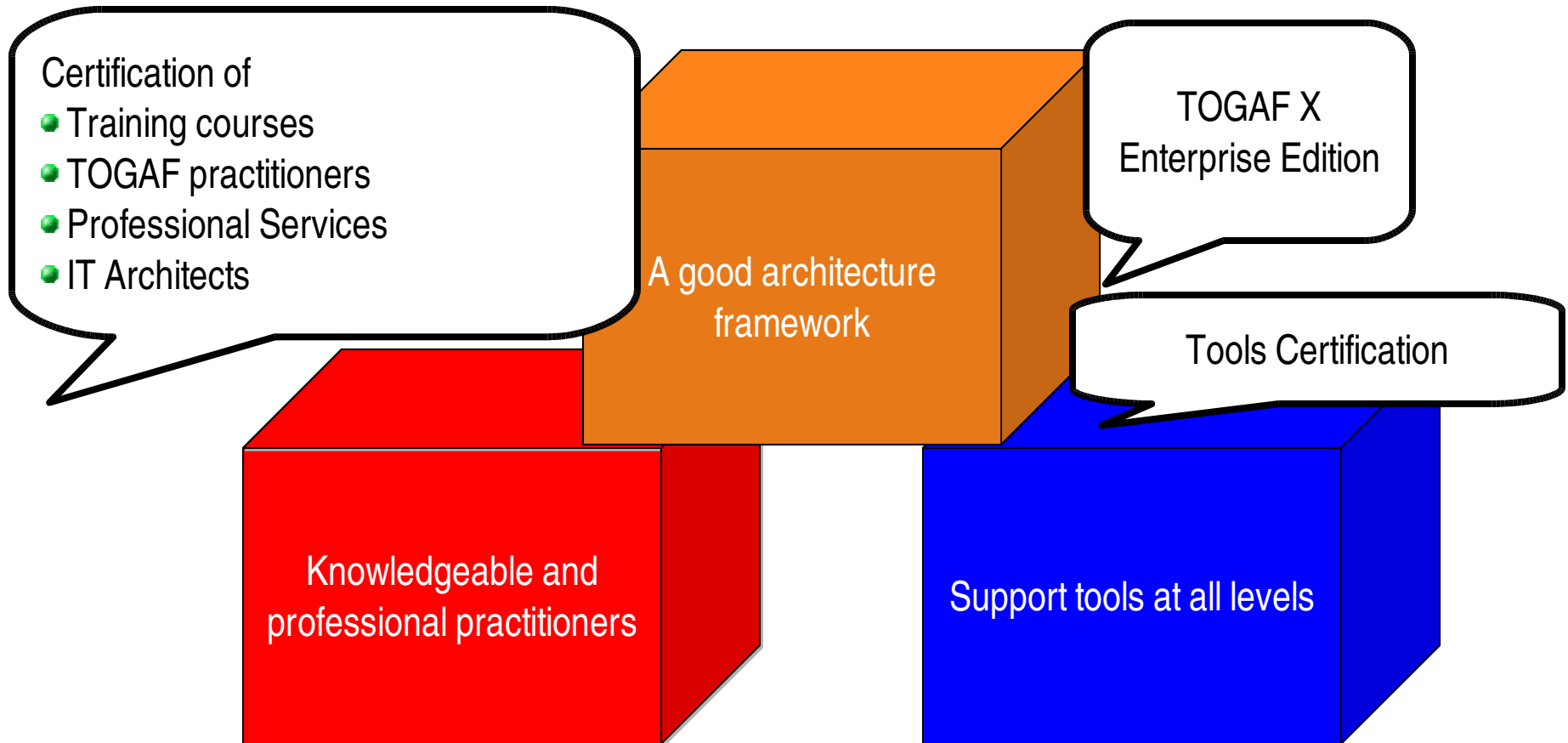


# The Enhanced ADM



# The Architecture Forum

- Striving to achieve a total, practical, architectural solution



# C4ISR Architecture Framework

- TOGAF began its life as DoDAF, which the DoD contributed to The Open Group.
- DoD went on to develop C4ISR in response to Congressional requirements.
  - Version 1 published 1996
  - Version 2 published in 1997
- Consists of three views with required and optional work products
  - Operational Architecture View
  - System Architecture View
  - Technical Architecture View

# ISO RM-ODP

- Reference Model for Open Distributed Processing description framework
- Goes beyond description specifications to functions systems should provide
- Espouses five normative viewpoints
  - Enterprise specification
  - Information specification
  - Computational specification
  - Engineering specification
  - Technology specification

# The classical architecting methods

- Normative (solution-based)
  - Examples: building codes and communication standards
- Rational (rule-based)
  - Examples: systems analysis and engineering
- Participative (stakeholder-based)
  - Examples: concurrent engineering and brainstorming
- Heuristic (lessons learned)
  - Examples: **Simplify. Simplify. Simplify.** And **SCOPE.**
- Normative and Rational are analytic, deductive, experiment-based and well understood. Both are science based.
- Participative and Heuristic are nonanalytic, inductive, less understood – a process of vision, insights, intuitions ... artistically based.

## Art complements Science

- The art of architecting is key to unprecedented and new applications.
- Art complements science where it is weakest: in dealing with immeasurables, in reducing past experience and wisdom to practice, in conceptualization, in providing sanity checks ...
- The nature of architecting changes as the project moves from phase to phase.
  - Earliest stages: structuring of hopes, needs, and dreams into a synthesizing of feasible technical possibilities.
  - Later stages: integration of and mediating among competing subsystems and interests ... a time for normative and rational methods



IBM Software Sales

# How we do that thing we do

Taking up  
Letting go

## Heuristics as tools

- Heuristics are guides or rules of thumb; abstractions of experience.
- Heuristics are distilled wisdom, carefully chosen, narrow in scope.  
*Look before you leap and He who hesitates is lost (not valid)*
- You know one when you hear one because it fits your model of the world and it seems to be a self-evident truth.

**Don't assume that the original statement of the problem is necessarily the best, or even the right one.**

**In partitioning, choose the elements so that they are as independent as possible; that is, elements with low external complexity and high internal complexity.**

**Build in and maintain options as long as possible in the design and implementation of complex systems. You will need them.**



## Criteria for selection

- The heuristic must make sense in its original domain or context.  
Strong correlation between the heuristic and the successes or failures of specific systems, products, or processes.
- The general sense, if not the specific words, should apply beyond the original context.  
**Before the flight it's opinion; after the flight it's obvious**
- The heuristic should be easily rationalized in a few minutes or on less than a page.  
**If you can't explain it in five minutes, either you don't understand it or it doesn't work.**
- The opposite statement of the heuristic should be foolish.
- The heuristic's lesson should have stood the test of time and gained broad consensus.

## Generating useful heuristics

- Use humor and carefully chosen words to produce a “bite”
- Use words that transmit “the thrill of insight”
- Try embedding both descriptive and prescriptive messages
- Don't make it so elegant that it only has meaning to its creator
- Turn conditional statements into separate but related heuristics, focused on dealing with that condition.

# Applying heuristics

- If it works, then it's useful.
- Knowing when and how to use a heuristic is as important as knowing the what and why.
- Heuristics work best when applied early to reduce the solution space.
- Strive for balance – there can be too much of a good thing
- Practice, practice, practice
- Heuristics aren't reality, either.
- A pattern (Alexander) is a specific form of prescriptive heuristic.

## Focused on the software or the problem ?

- Our architectural models and descriptions tend to focus on the structure of the software, not on the problem we are asked to solve.
- Our models tend to be abstractions of programming languages and approaches.
- IEEE 1471-2000 helps by focusing on the stakeholder's concerns.
- Rather than deriving views from the structure of the software, deriving the views from the stakeholders allows us to communicate the models back to the stakeholders in their own language.
- This approach allows the architect to stay connected with the client's requirements and purpose.

## Models and modeling

- The product of the architect is “a vision”; it is intangible.
- The progression of models during the design life cycle can be visualized as a steady reduction of abstraction.
- As architectural decisions are made (and recorded) the range of options narrows and the models become more specific.
- Eventually the models become construction drawings and itemized budgets, and pass into the hands of the builders.
- Part of the architect's role is to determine which views are most critical to system success, build models for those views, and then integrate as necessary to maintain system integrity.

# The purpose of models

- Match the desirability of purposes with the practical feasibility of a system to fulfill those purposes.
- Help the customer clarify abstract objectives through provisional and explanatory models.

- Recall definitions

Model

approximation, representation of selected aspects

View (collection of models)

representation of a system from a related set of concerns

Viewpoint

a template for constructing a view

## Models of form

- Physically identifiable elements of, and interfaces to what will be constructed.
- Closely tied to particular construction technologies
- Scale models – prototypes and proofs of concept
- Block diagrams – system interconnect diagrams, system flow diagrams, structure charts, class and object diagrams

## Behavioral (functional) models

- Describe specific patterns of behavior by the system
- Threads and scenarios – a sequence of system operations, a.k.a. use-cases.
- Data and event flow – allow threads to be collapsed into more compact models



# Performance models

- Describes or predicts how effectively an aspect of the architecture satisfies some function
- Usually quantitative and at a system level
- These are “ilities” or nonfunctional requirements
- Products of science and engineering, particularly the formal methods

# Data Models

- What data does the system retain and what relationships among the data does it develop and maintain ?
- Entity-Relationship diagrams for relational databases
- In data-intensive systems, generating intelligent behavior is a matter of finding relationships and imposing a persistent structure on the records.

## Why we use Methods

- The earlier definition stated in part that a systems architect is someone who uses a methodical approach.
- The use, transformation, and creation of method tools is seen as a sign of maturity in an architect.
- As a craft person becomes more mature in their practice, they grow in their facility to use the tools of their craft in different ways.
- This leads to generativity in one's profession and extends the current boundaries of practice.
- But, one's use of methods should not be “mechanical”

## Free from all method

*“The more we learn to use this method, the more we find that what it does is not teach us processes we did not know before, but rather opens up a process in us, which was part of us already. ...this capacity in us is not accessible, until we first go through the discipline which teaches us to let go of our fears.”*

Alexander – TTWOB pg.13

## Does the Method actually restrict us?

*“ ...we have so far beset ourselves with rules, and concepts, and ideas of what must be done...that we have become affraid of what will happen naturally, and convinced that we must work within a 'system' and with 'methods' since without them our surroundings will come tumbling down in chaos.”*

Alexander – TTWOB pg.14

## Methods provide structure

- Methods provide structure to our thought processes to give us an idea of “the next thing to do”
- Once you let go of the fear of not having “the next thing to do” the structures created by your use of method, in your mental “muscle memory” will take over.
- You will come to understand that the structure of the method which once empowered you feels restrictive. This is the breakthrough point at which your creativity is freed.
- But, you really have not left method behind, it still informs what you do.

## Shedding method

*“ ...we must first learn a discipline which teaches us the true relationship between ourselves and our surroundings.*

*Then, once this discipline has done its work,...we will be ready to give up the discipline, and act as nature does.*

*This is the timeless way of building: learning the discipline-and shedding it”*

Alexander – TTWOB pg.16



IBM Software Sales

Here endeth the lesson

This presentation was created  
with [OpenOffice.org](http://OpenOffice.org)  
on the Fedora project Linux client



## Additional resources

- *Please Understand Me II*, David Keirsey ISBN 1-885705-02-6

This is the source for the information on the Myers-Briggs Type Indicator and the Keirsey Temperment Sorter.

- *The Introvert Advantage*, Marti Olsen Laney ISBN 0-7611-2369-5

For all of us who are introverts and can't figure out why we don't seem to fit in. This is the book on how to survive in an extrovert world.