



Enterprise Science using the Zachman Framework

Adriaan Vorster

Outline

- Historical perspective
- The essence of information systems
- Lehman's laws
- Complex systems
- The Zachman Framework
- Alchemy versus science
- Conclusion

Perspective



- Those who do not know history are doomed to relive it.
- The only lesson that man does not learn from history, is that man does not learn from history.

The history

- The Mythical Man Month, published in 1975 by Frederick P Brooks, Jr. Professor of Computer Science at University of North Carolina.
- Father of the IBM System 360.
- No Silver Bullet – Essence and Accident published in 1986, predicted that there is no single development, in either technology or management technique, which by itself promises even one order of magnitude improvement within a decade in software productivity, reliability and simplicity.
- 20th Anniversary of Mythical Man Month in 1995, No Silver Bullet still holds

The essence of software



- Complexity
 - Modern software systems are the most complex constructs yet created by mankind
- Conformity
 - Software has to conform to the whims of users, no absolute standards
- Changeability
 - Software is embedded in a socio cultural matrix of applications, users, laws, processing platforms etc. constant pressure for change due to innovative use of applications
- Invisibility
 - Software is invisible and unvisualisable. The human mind can not grasp all aspects of software at once.

A detour into thermodynamics



- There is a measure of disorder, variability or chaos, known as entropy.
- Thermodynamics distinguishes between open, or anentropic, and closed, or entropic, systems.
- Open systems accept energy from the environment and establish order, consider the growth of a tree
- The only natural processes that occur in closed systems are those that lead to a decline in energy and an increase in disorder, consider the decay of a felled tree.
- Closed systems require a constant energy input to maintain system state.

Lehman's laws

- Any piece of software that is used will be changed.
- Unless very stringent measures are enforced, any change to a piece of functioning software will lead to an increase in complexity.
- This increase in complexity is proportional to the initial complexity.
- There is a level of complexity, entropy or chaos, beyond which it will be more cost effective to replace than repair a piece of software - this is known as the limit of maintainability.

Analysis of Lehman's law



Start with a change in entropy that is proportional to the initial entropy

$$\Delta E \propto E$$

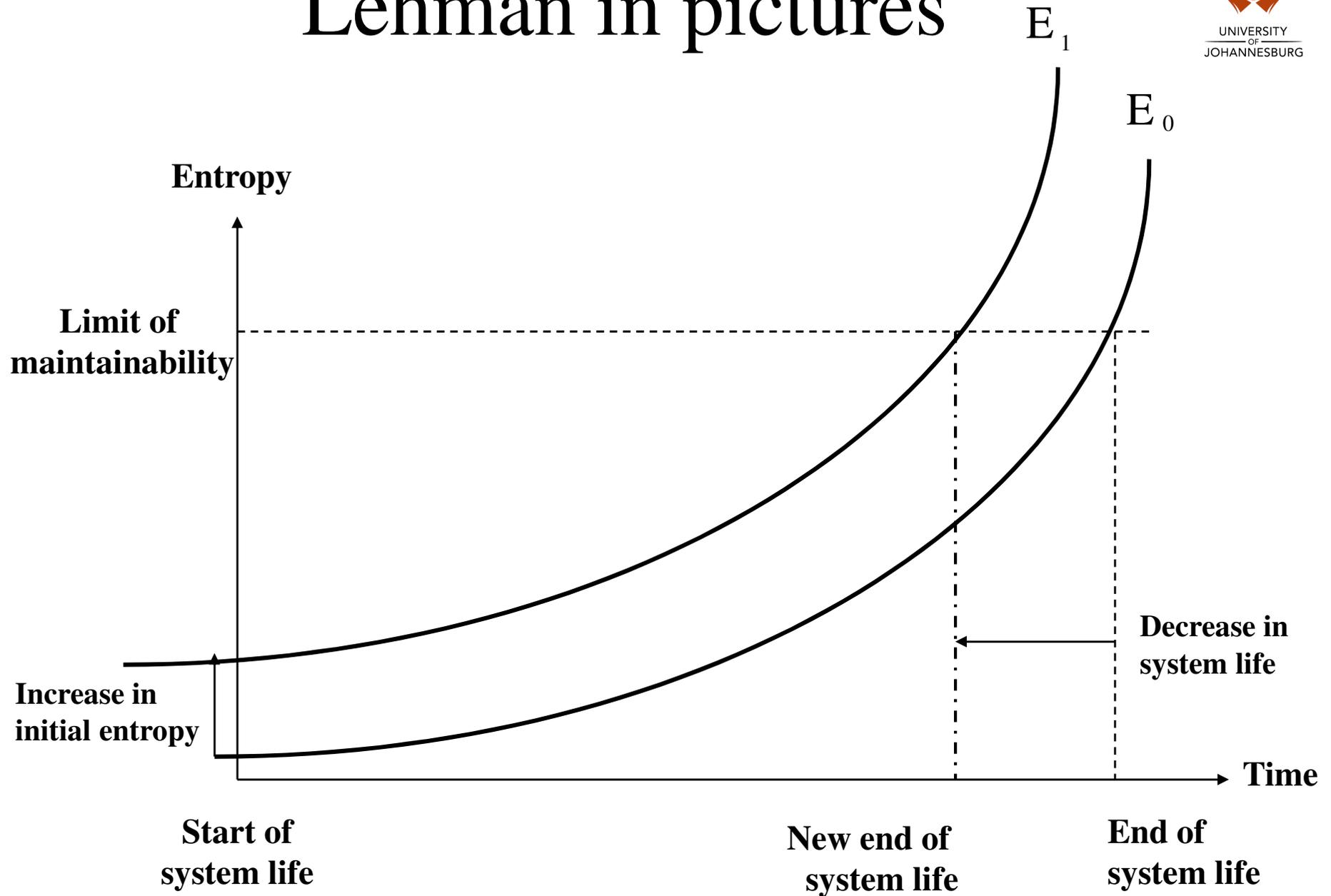
The change over time is:

$$\frac{dE}{dt} = kE$$

The system entropy then becomes:

$$E = E_0 e^{kt}$$

Lehman in pictures



Implications of Lehman's Laws



- All information systems will continually evolve to suit the user requirements.
- There is a level of entropy beyond which it is better to rewrite (replace) than to try and maintain software.
- System life is critically dependent on the initial entropy.
- It requires energy (time, money, people) to maintain the state of a software system.
- System life has a non linear relationship to initial entropy – difficult to estimate.
- Software applications have a finite life span and should be life cycle managed.

Buy versus build

- Very few enterprises build their own software applications.
- Most enterprises buy packaged applications.
- What is the difference?
- The vendor has:
 - Identified a data model
 - Chosen a technology model
 - Written some partially debugged code based on generic business rules
 - Most probably not given you the details of any of these
- There is no significant difference in the analysis required to implement a packaged solution

Financial implications

- Software systems represent irreversible investments.
- It is difficult to escape from the investment decision without sacrificing large value.
- Software systems have little or no intrinsic or resale value.
- Irreversible investments are handled through good up front analysis and phased implementation.

What is the success rate?



- Only 30% of implemented ERP systems could be considered a success when measured against the original system goals of lower costs, better functionality and greater enterprise support.
- Systems implementations that were preceded by rigorous pre-implementation planning and design produced better results in 56% of cases.

Complex systems



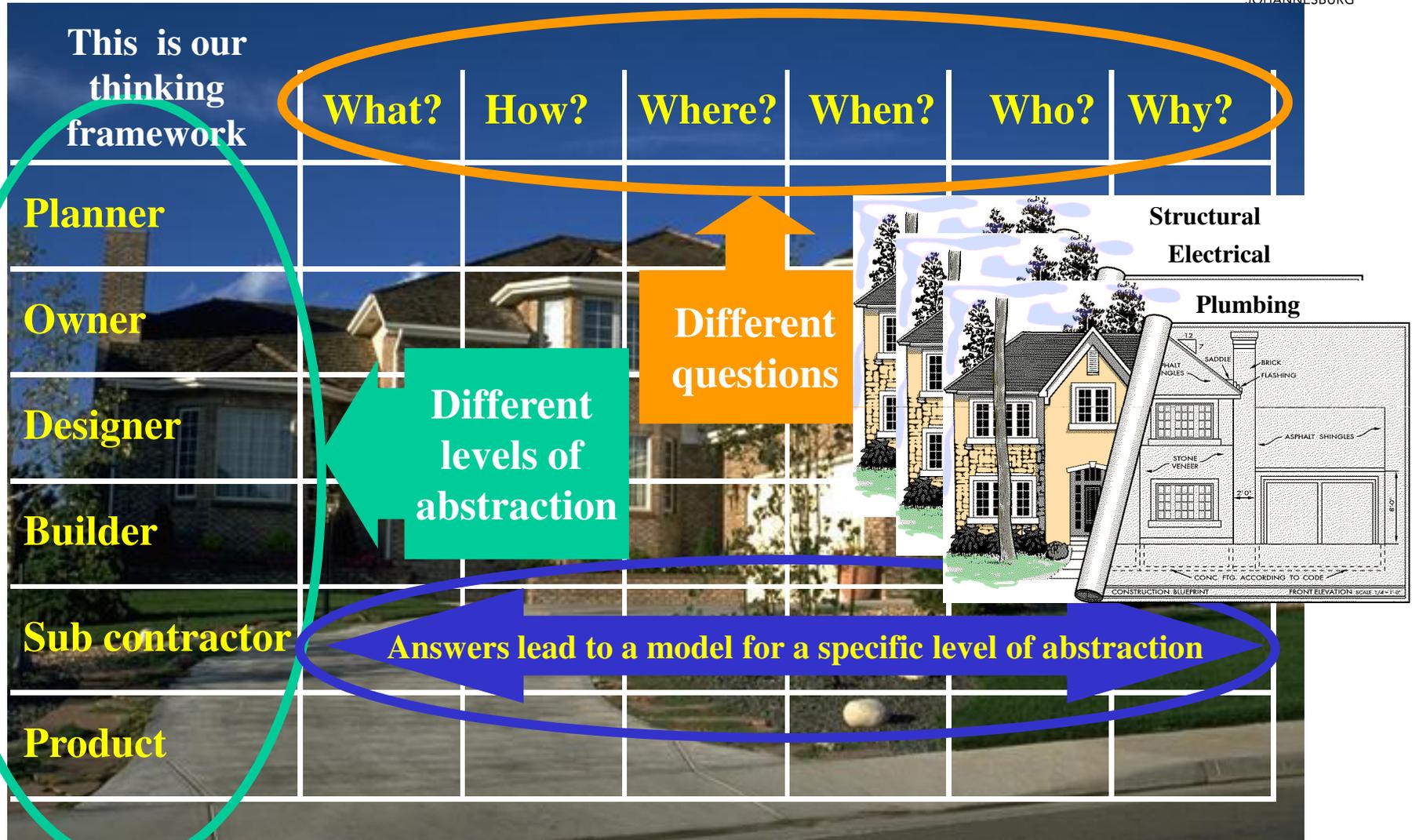
- Complex systems inherently exhibit non linear behavior.
- These responses are the result of the product of the interactions of the parts, not the sum of the interactions.
- Complex systems can not be analysed by taking them apart and treating each component in isolation.
- Locally optimising parts of complex systems invariably leads to the sub-optimisation of the whole system.
- The only way to handle complex systems is to concentrate on the components whilst maintaining a sense of the contextual alignment of the components.

Let us build something complex like a house



UNIVERSITY OF JOHANNESBURG

We begin by establishing a representative model.



By combining all the models we have established an architecture!

Let us Talk



	What?	How?	Where?	When?	Who?	Why?
Planner						
Owner						
Designer						
Builder						
Sub contractor						
Product						

The Zachman Framework

This is where you want to go

Systems Architecture	What	How	Where	Who	When	Why	
Planner model	List of things important to business.	List of processes the business performs	List of locations where business operates	List of users / groups / units of the business	List of business events / cycles	List of business goals / strategies	Strategic scope model
Owner model	Entity relationship diagram.	Business processes.	Logistics network.	Organization chart.	Business event / trigger char	Business plan.	Business model
Designer model	Data architecture	Application architecture	Distributed systems architecture	User interface architecture	State transition diagram	Business rules	Conceptual systems model
Builder model	Data design	Application design	Hardware systems design	User interface design	State transition design	Knowledge design	Technology model
Sub-contractor model	Data definition	Program	Network architecture	Access architecture	Timing / interrupts	Rule definition	Detailed representations
Product model	Data	Application	Communication	Users	Schedules	Rules / strategy	Functional systems
	Data	Function	Network	Users	Time	Motivation	IS Architecture

This is the domain you have to address

This is where you are

How to build legacy or “throw away” systems



Systems Architecture	What	How	Where	Who	When	Why	
Planner model	List of the business processes	List of business processes					Strategic
Owner model	Business requirements	Business requirements					Business model
Designer model	Business requirements	Business requirements					Conceptual systems model
Builder model	Business requirements	Business requirements				Knowledge design	Technology model
Sub-contractor model	Data definition	Program					Implementation
Product model	Data	Application	Communication	Users	Schedules	Rules / strategy	Functional systems
	Data	Function	Network	Users	Time	Values	IT Architecture

Do not build models

Fix situations by fire fighting

Concentrate on this tier



Why is the previous approach wrong?

Systems Architecture	What	How	Where	Who			
Planner model							
Owner model							
Designer model							
Builder model							
Sub-contractor model							
Product model	Data	Application	Communication	Users	Schedules	Rules / strategy	
	Data	Function	Network	Users	Time	Values	IT Architecture

Management constantly changes strategy as business constantly changes to meet new requirements

This causes more fragmentation of functionality and data

We quickly contrive localised (undocumented spreadsheet) solutions

We are stuck with these systems

What is a framework used for?



This is our thinking framework	What?	How?	Where?	When?	Who?	Why?
Planner	<p>A framework is a classification scheme that enables focused concentration on selected aspects of a subject or object while retaining a sense of the contextual, or holistic, perspective.</p> <p>J A Zachman - 1987</p>					
Owner						
Designer						
Builder						
Sub contractor						
Product						



Features of the Zachman Framework

- The columns are all primitive, unique and non overlapping.
- There is no sense of order in the ranking of the columns, the Zachman Framework could be seen as a tube where column 6 meets up with column 1.
- The rows are ordered in levels of abstraction.
- Each downward transition presents a decision opportunity.
- You do not have to make the decision, somebody else will make it at their level of abstraction.
- This may not align with the business drivers, and could violate the design parameters, but will in probability not bother you until disaster strikes.

Some observations



- Your enterprise has all of the Zachman models.
- You have a business rule model.
- You have a process model.
- You have a data model.
- You have a geographic dispersion model.
- You have a temporal model.
- You have a people model. (Form follows function)
- All these models are always present at every level of abstraction.
- Even if you chose not to make them explicit.
- Even if you ignore the models they do not go away, you simply wind up not knowing where to look when disaster strikes.

More observations

- All the models are orthogonal, ie there is NO natural projections between them.
- You could make business rules in isolation that are unenforceable.
- You could design business processes in isolation that are unsupported.
- You could design a data model that does not relate to the enterprise.
- Only by considering all six columns at a specific level of abstraction could you design a complete model for that level of abstraction, understanding the constraint set, compromises and trade-offs that are required.

Is this important in the real world?

- Are there any real world examples of all the preceding technospeak?
- Are there real examples that relate to the six primitive interrogatives?
- Why are the interrelations said to be important?

Some Causal Relationships in Data Quality



- The Business Rule provides the containing structure for the Business Process
- The Business process gets the work done and sets the Competency requirements
- The Competency requirements allows for the deployment of a System
- The System will generate DATA
- The above is a set of absolutely causal links in the DATA QUALITY CHAIN.
- Violate any of these, get unskilled users to make up their own rules, deploy lots of spreadsheets, and use any business process, and you **WILL GENERATE DUFF DATA!**

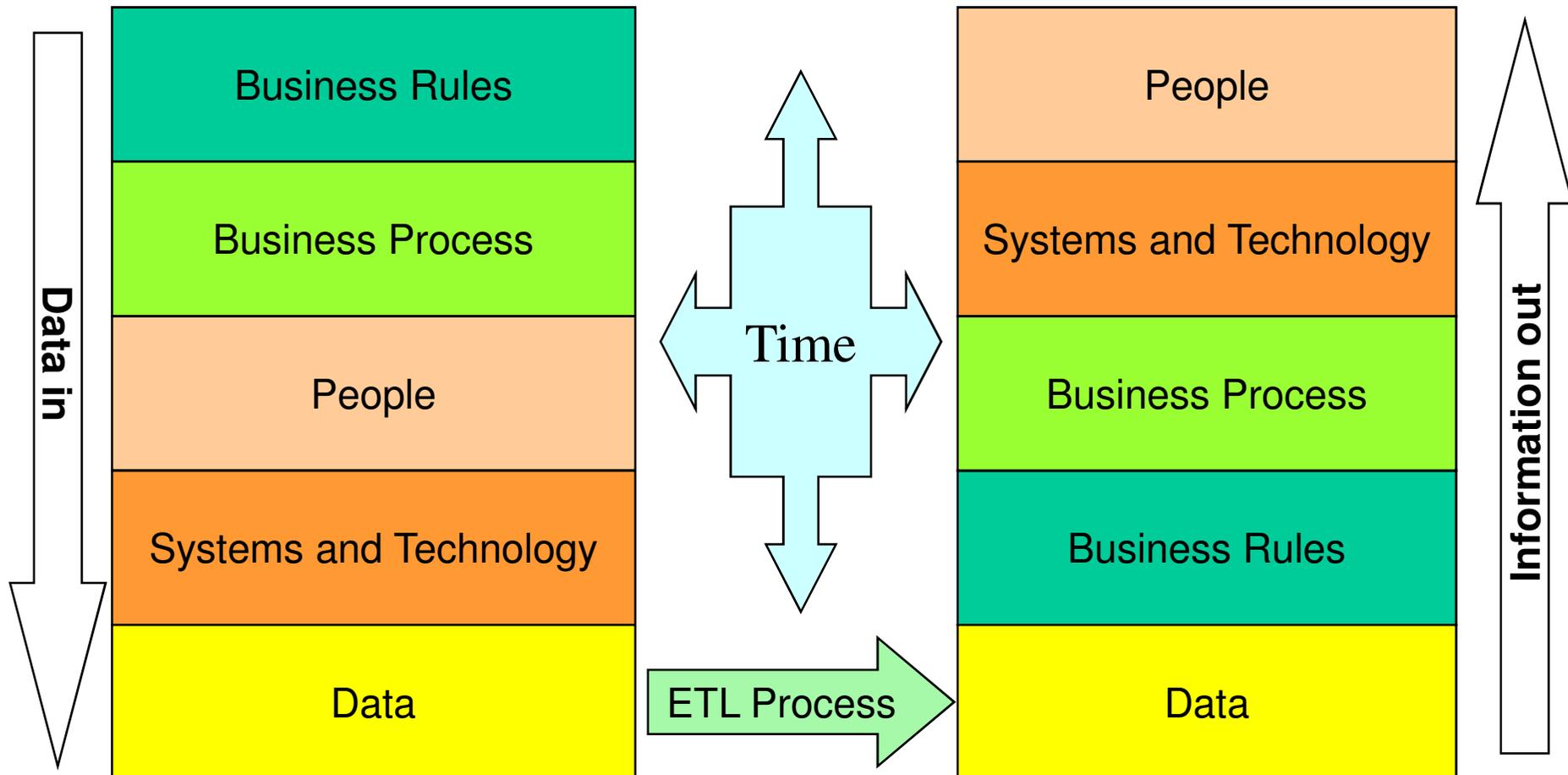
Causality principles in Information Systems



UNIVERSITY
OF
JOHANNESBURG

Operational systems
(getting quality data in)

Information provision
(getting meaningful information out)



GIGO principle: If you allow Garbage In you will get Garbage Out

Alchemy versus science



- Alchemy relies on the casting of spells and incantations to turn base objects into valuable artifacts.
- Alchemy is risky with a high failure rate.
- Alchemy depends on a lack of knowledge
- Science relies on an understanding of the primitives to build value through analysis and design.
- Science is precise with high repeatability and predictable outcomes.
- Science requires vast knowledge and experience.

Alchemy ceased when the periodic table of the elements, an ordered classification scheme, based on the knowledge of the primitives of the elements, the electron structure, was established.

The end of Alchemy



UNIVERSITY
OF
JOHANNESBURG

Periodic Table of the Elements

	1A																0	
1	1 H																2 He	
2	3 Li	4 Be										5 B	6 C	7 N	8 O	9 F	10 Ne	
3	11 Na	12 Mg										13 Al	14 Si	15 P	16 S	17 Cl	18 Ar	
4	19 K	20 Ca	21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr
5	37 Rb	38 Sr	39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe
6	55 Cs	56 Ba	57 * La	72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Tl	82 Pb	83 Bi	84 Po	85 At	86 Rn
7	87 Fr	88 Ra	89 + Ac	104 Rf	105 Ha	106 106	107 107	108 108	109 109	110 110								

• Lanthanide Series

+ Actinide Series

58 Ce	59 Pr	60 Nd	61 Pm	62 Sm	63 Eu	64 Gd	65 Tb	66 Dy	67 Ho	68 Er	69 Tm	70 Yb	71 Lu
90 Th	91 Pa	92 U	93 Np	94 Pu	95 Am	96 Cm	97 Bk	98 Cf	99 Es	100 Fm	101 Md	102 No	103 Lr

What is the Periodic Table?

- In the words of John Zachman we could probably describe the Periodic Table as a classification scheme, based on a profound knowledge of the primitives, that enables focused concentration on selected aspects of a subject or object while retaining a sense of the contextual, or holistic, perspective.

Enterprise alchemy



- Enterprise alchemists rely on the casting of spells to ward off evil, increase fertility and effect change.
- For a fee they will use tool focused spells like BPR, ERP, SCM, CRM, balanced scorecard, six sigma....
- These spells and incantations have a short term positive effect whereafter the enterprise normally declines until the next application of alchemy.
- Many second wave implementations of ERP systems are examples of the re-application of a powerful potion that did not take on the first attempt.
- Enterprises that do not build explicit models, enterprises that do not use primitives to depict reality, have no defense against enterprise alchemists.

Enterprise science



- Build the Zachman models, to measure is to know.
- Row 1 model
 - Who are we, what are we doing and where are we going, RFI
- Row 2 model
 - What do the enterprise functions require, RFP
- Row 3 model
 - Building plan, compromise between our models and application models
- Row 4 model
 - Implementation plan
- Row 5 model
 - Integration plan
- Row 6
 - Functioning system

Results of enterprise science



- Information systems that are inherently aligned with the enterprise strategy.
- Enterprise knowledge that is explicit and captured in an architectural repository.
- Competitive advantage through process owners having taken ownership of their applications.
- High quality communication through the use of a unique and precise vocabulary.
- Maintainable systems due to the availability of detailed blueprints and enterprise knowledge base.
- A quality data resource that supports the enterprise.

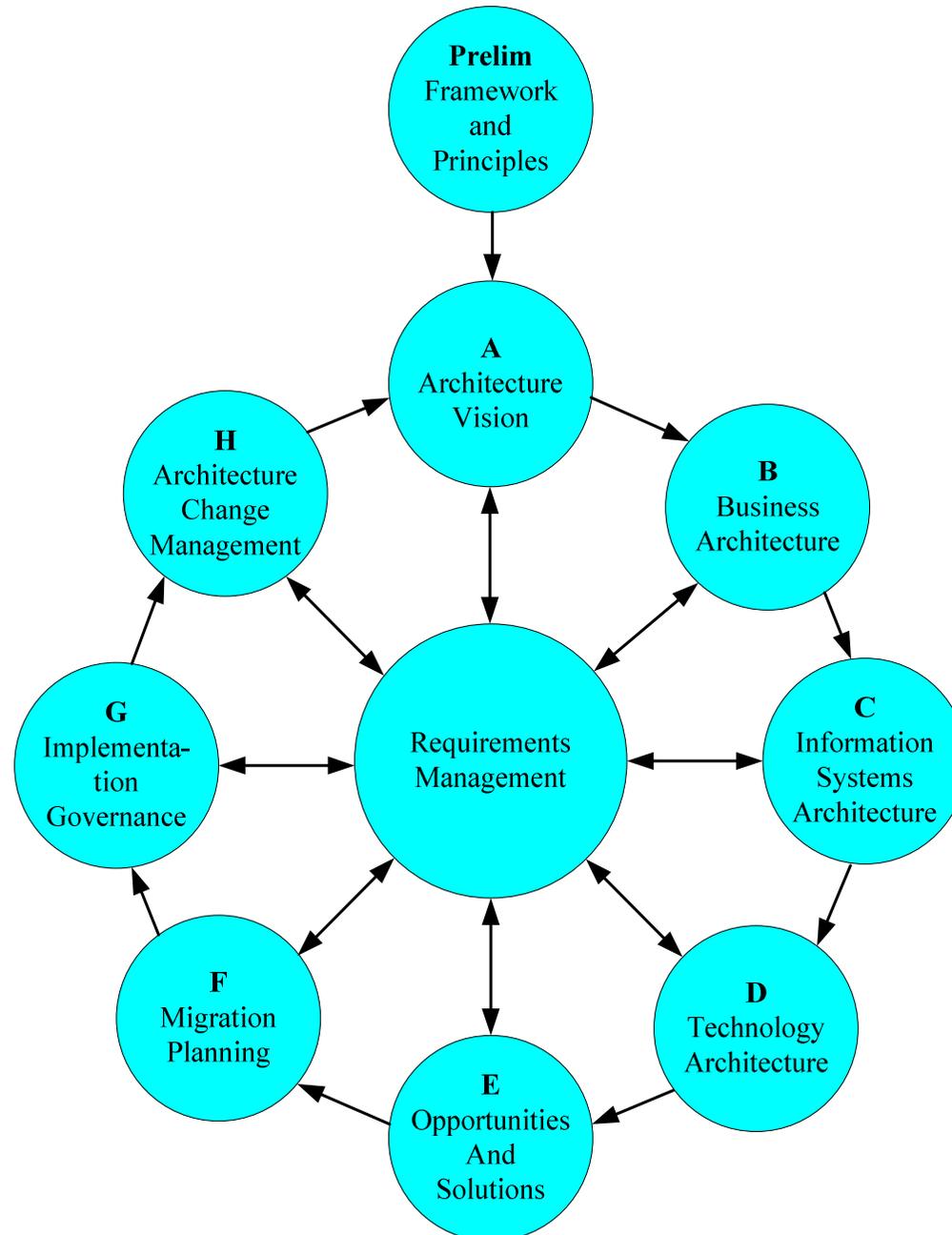
Criticisms of the Zachman Framework

- Nobody has ever built all of the models of the Framework in detail.
- As a corollary nobody has ever used all of the elements of the periodic table.
 - This does not devalue the utility, principles or logic of the Periodic Table, as a matter of fact, knowledgeable use of this classification scheme continues to spawn new products, business opportunities and creates wealth.

More criticisms of the Zachman Framework

- Zachman has presented a thinking tool with no set implementation methodology
- Whilst this statement used to have some credibility, the development of the TOGAF based Architecture Development Methodology provides a consistent and repeatable methodology for implementing Enterprise Architecture.

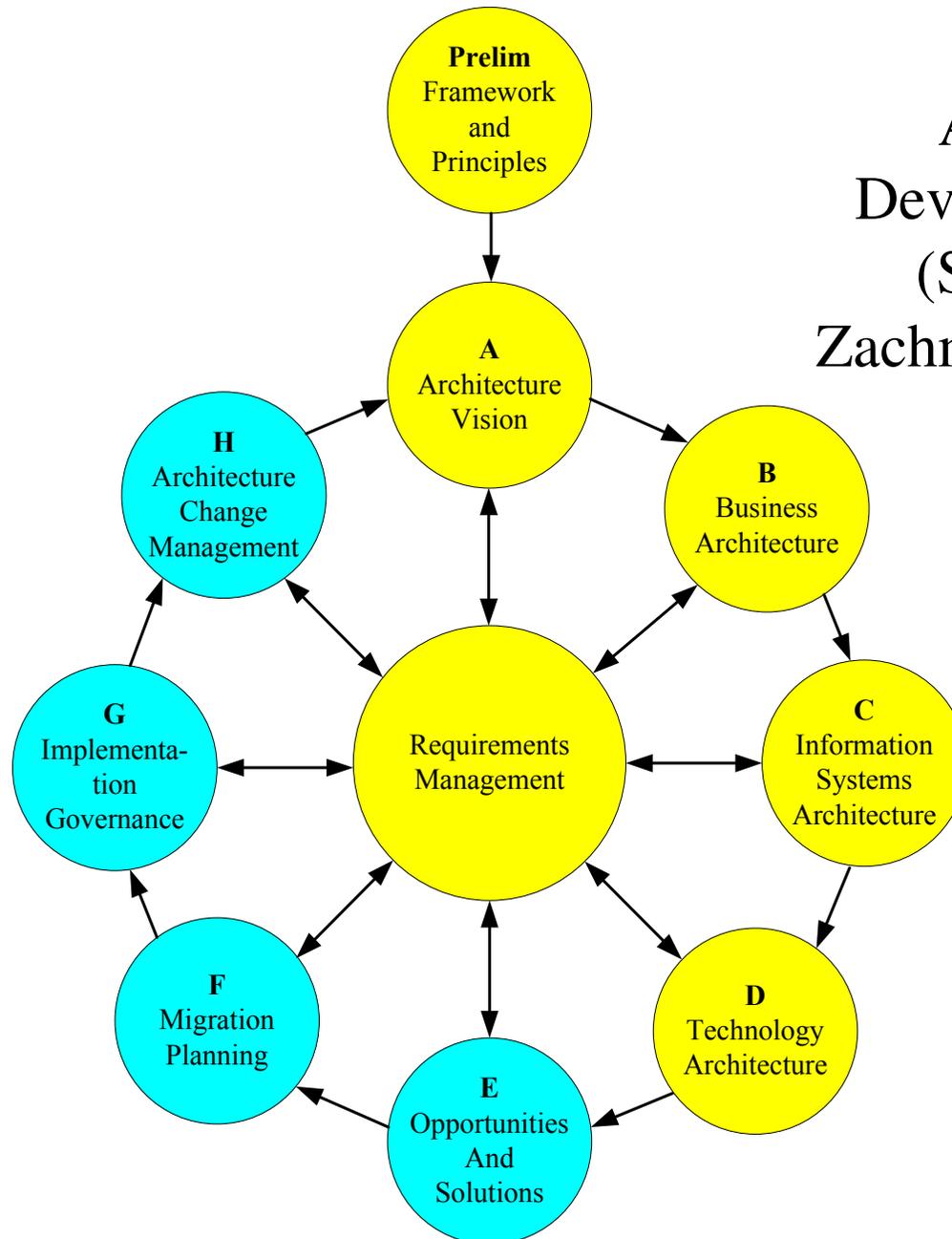
TOGAF Architecture Development Methodology



TOGAF Architecture Development Methodology



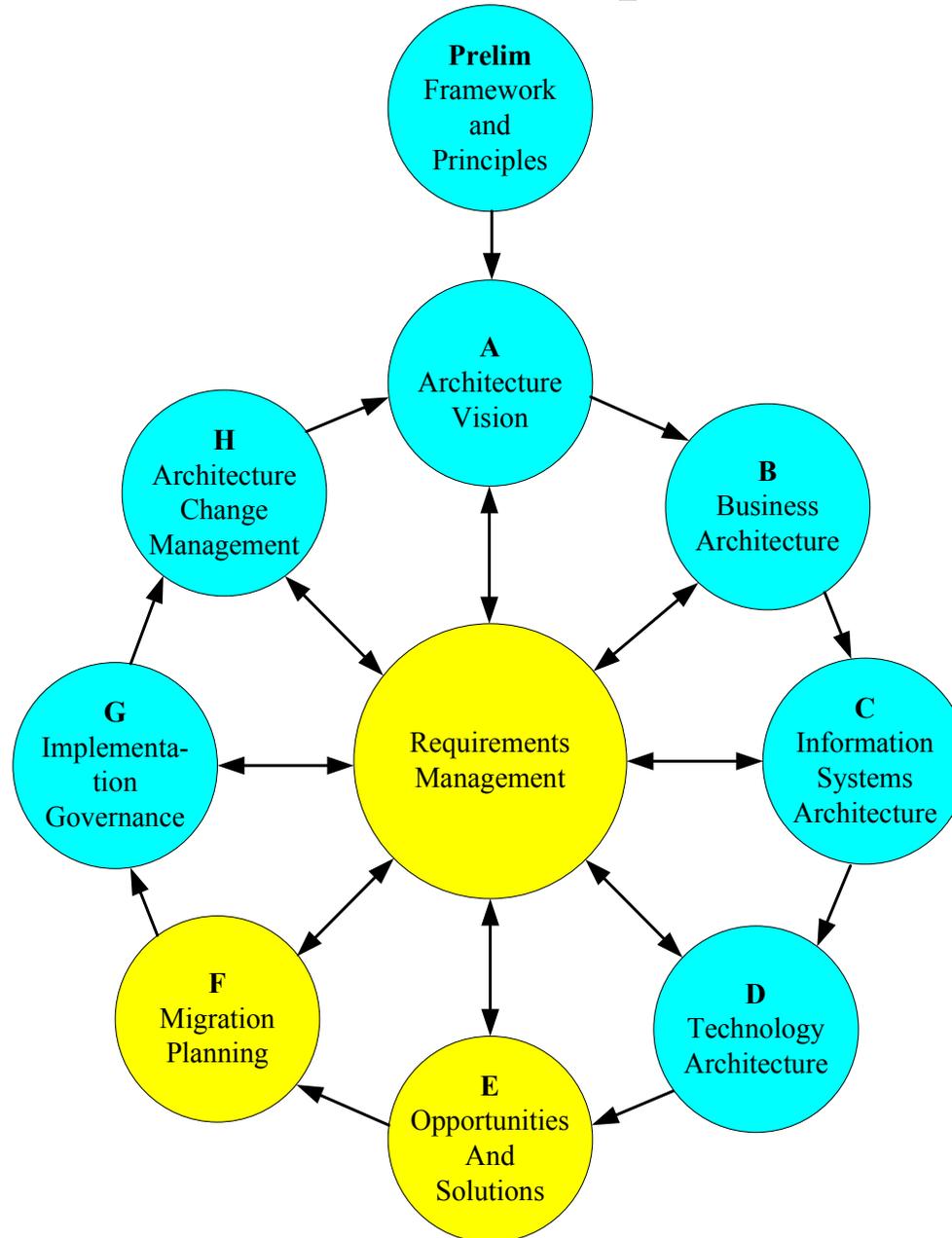
Architecture
Development Phase
(Start with the
Zachman Framework!)



TOGAF Architecture Development Methodology



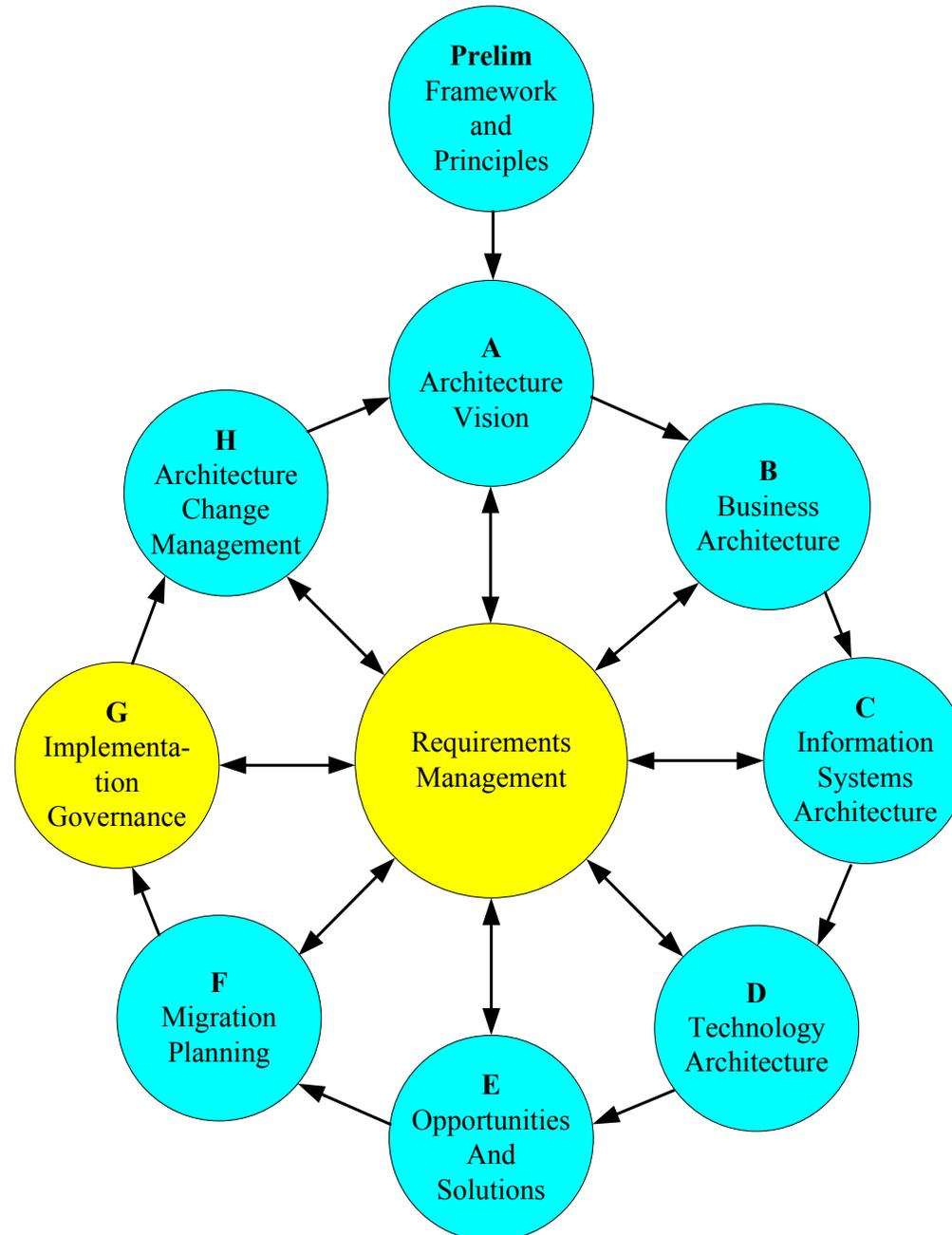
Business Transformation and Planning



TOGAF Architecture Development Methodology



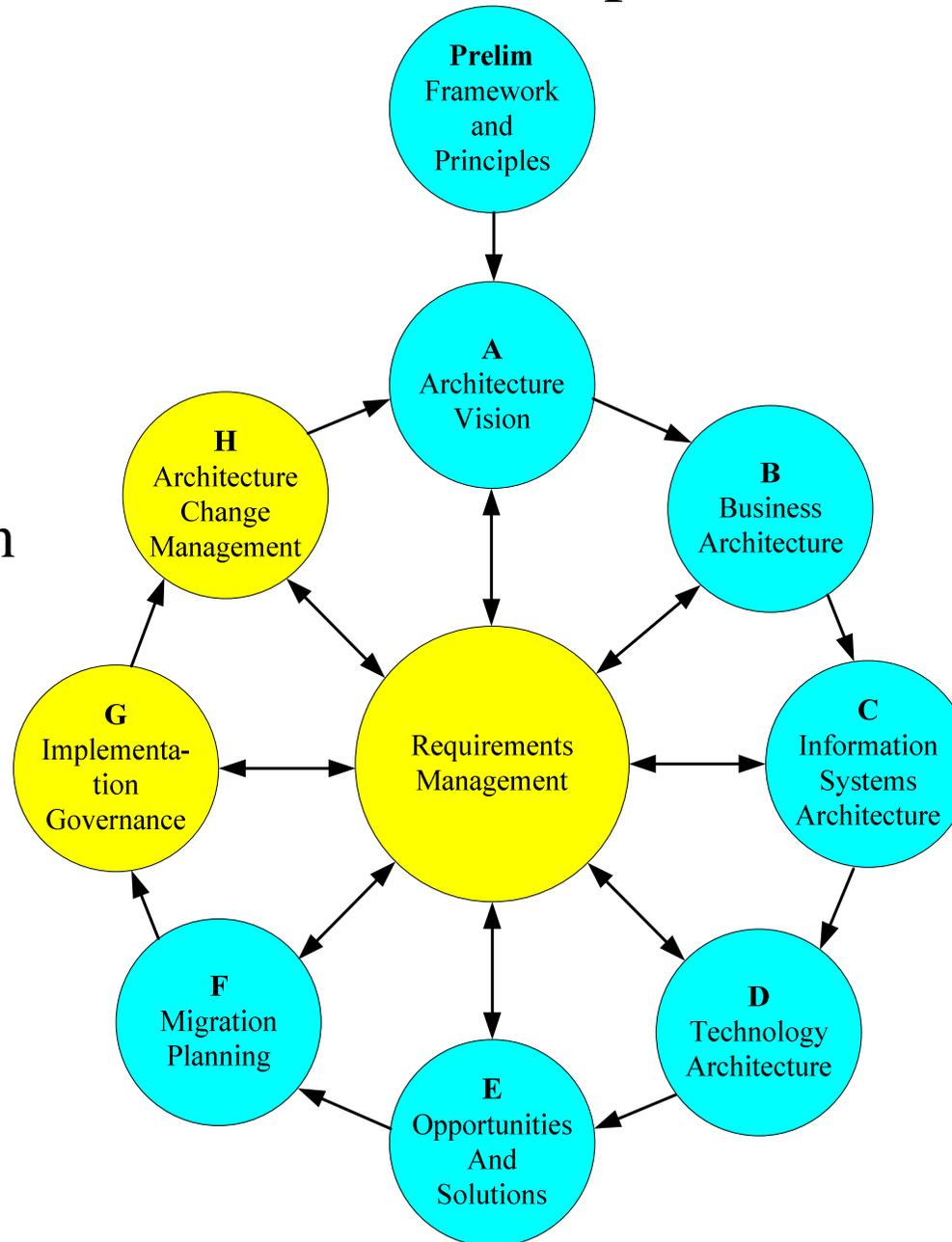
Implementation
and
Governance
of the
Implementation



TOGAF Architecture Development Methodology



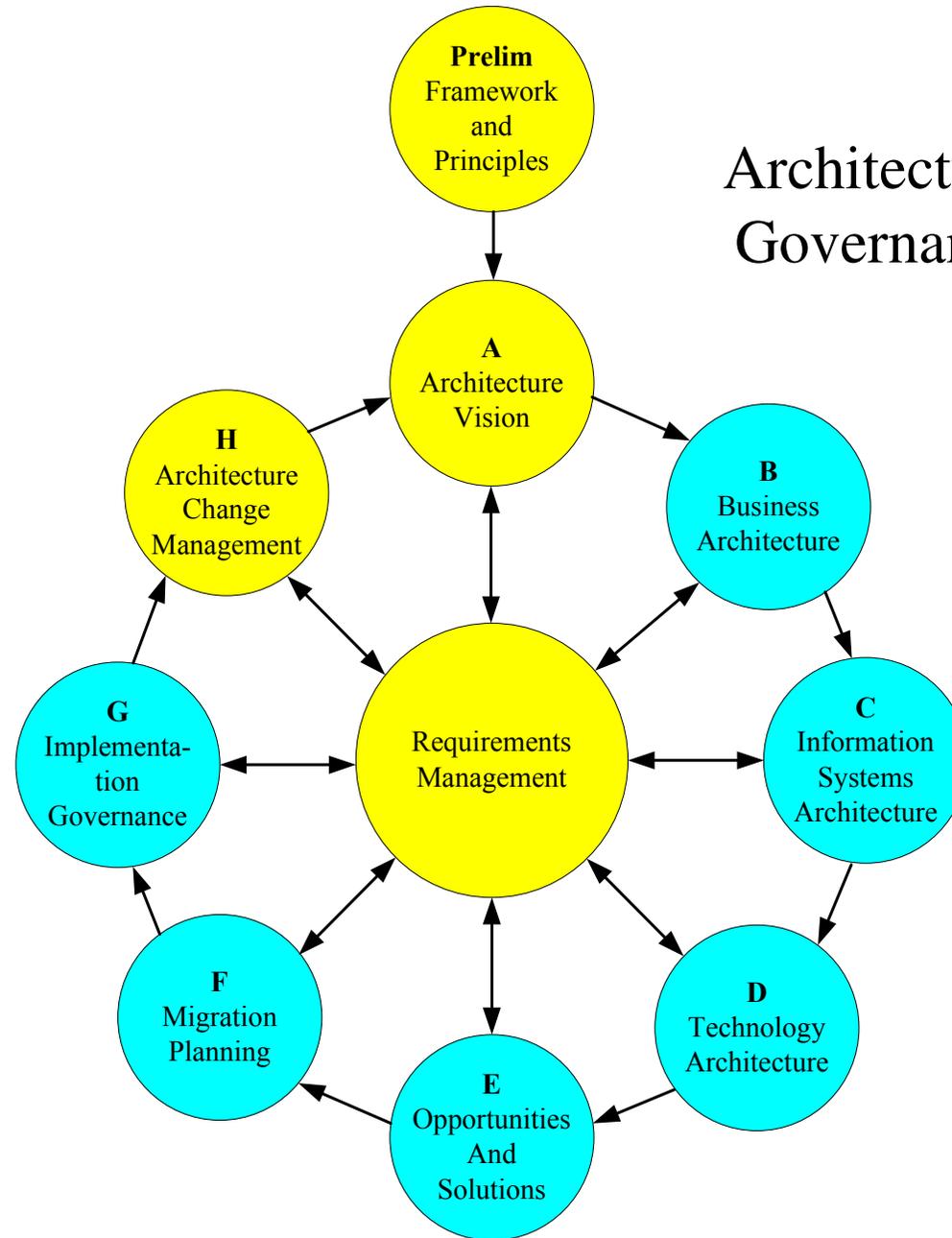
Value
Realisation



TOGAF Architecture Development Methodology



Architecture Governance



More on the TOGAF ADM



- The ADM presents a structured and internationally recognised methodology that may be used to develop architectures.
- The most common of these are the Business, (Information), Data, Application and Technology architectures.
- TOGAF ADM is supported by several vendor's tools
- TOGAF ADM has been mapped to other industry frameworks like FEAF, COBIT and ITIL
- TOGAF ADM certification indicates that the practitioner has achieved competence in the use of the ADM to develop architectural products.

Conclusion

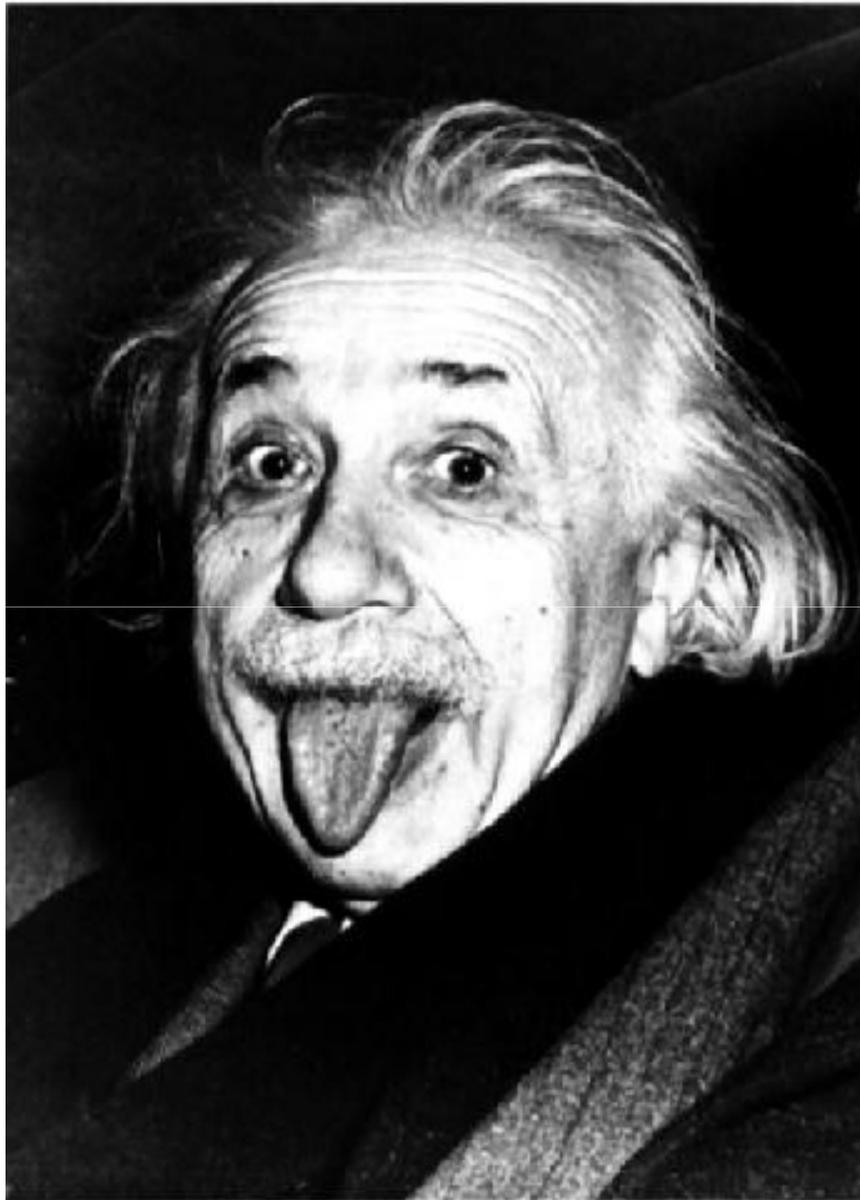


- Enterprise systems are complex systems that are under constant requirement to change and will tend to descend into chaos.
- The rate of the descent into chaos is determined by the initial entropy, both of the software and the intended environment, and the changes required to make the system conform to the enterprise.
- The life cycle management of software applications require the establishment and maintenance of explicit representations at the correct levels of abstraction.
- The Zachman framework provides an inclusive thinking tool that allows you to manage the complexity inherent in modern information systems.
- Combining Zachman with TOGAF provides an industry standard and certified way to create sustainable and value adding enterprise architecture

Zachman Framework Resources



- Training
 - Three day course on the Zachman Framework presented by RealIrm refer to:
www.realirm.com/zachman-framework-course
 - TOGAF certification: four day course presented by RealIrm refer to :
www.realirm.com/811-course-description



The significant
problems we face
will not be solved
by the same level
of thinking that
caused them

A. Einstein



Questions?