# A Design Quality Learning Unit in OO Modeling Bridging Engineer and Artist

Les Waguespack, Ph.D.
Professor & Chairperson of Computer Information Systems

Bentley University
Waltham Massachusetts

ISECON - Wilmington, NC - 2011-11-3

“How is it that one system is more effective, appealing, satisfying and/or more beautiful than another to its stakeholder community?”

Waguespack, L. J. (2010). Thriving Systems Theory and Metaphor-Driven Modeling. London: Springer-Verlag

“How is it that one system is more effective, appealing, satisfying and/or more beautiful than another to its stakeholder community?”

Waguespack, L. J. (2010). Thriving Systems Theory and Metaphor-Driven Modeling. London: Springer-Verlag

# How is it that one object model is better than another?

"How is it that one system is more effective, appealing, satisfying and/or more beautiful than another to its stakeholder community?"

Waguespack, L. J. (2010). Thriving Systems Theory and Metaphor-Driven Modeling. London: Springer-Verlag

How is it that one object model is better than another?

What determines design quality in an object model?

# defining design quality

## quality |ˈkwälətē| noun

1 the standard of something as measured against other things of a similar kind; the degree of excellence of something: *an improvement in product quality* | *people today enjoy a better quality of life.*

• general excellence of standard or level: *a masterpiece for connoisseurs of quality* | *[ as modifier ] : a wide choice of quality beers.*

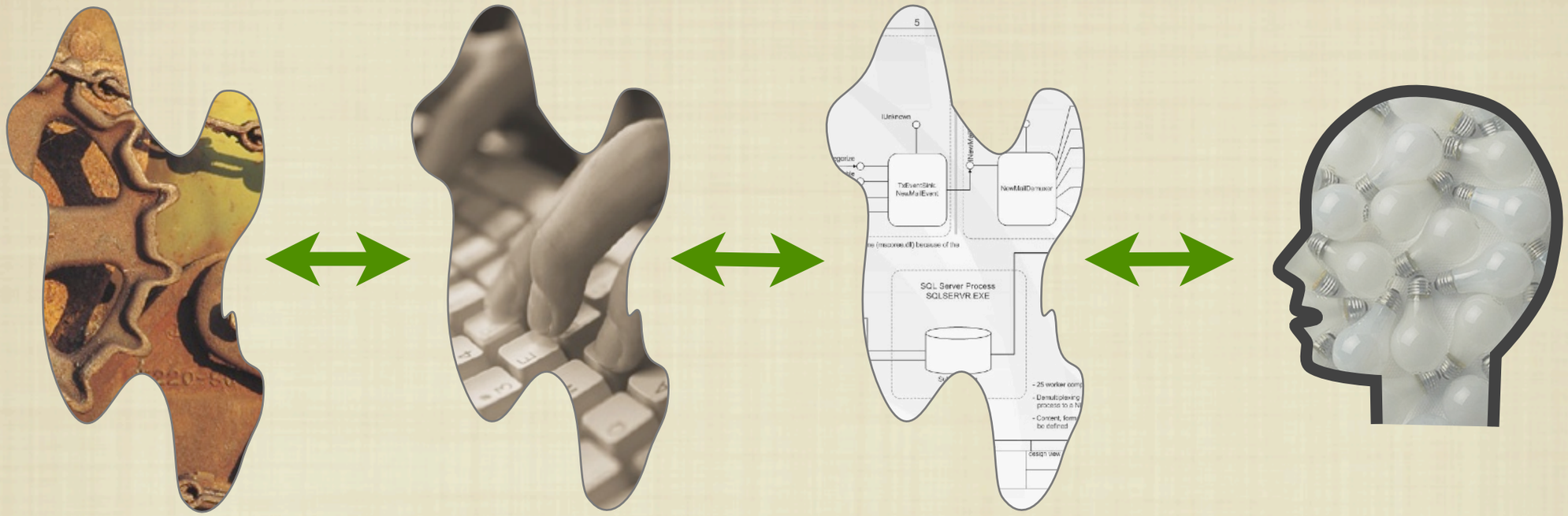
2 a distinctive attribute or characteristic possessed by someone or something: *he shows strong leadership qualities* | *the plant's aphrodisiac qualities.*

## design |dəˈzīn| verb [ with obj. ]

decide upon the look and functioning of (a building, garment, or other object), typically by making a detailed drawing of it: *a number of architectural students were designing a factory* | [ as adj. with submodifier ] (designed) : *specially designed buildings*.

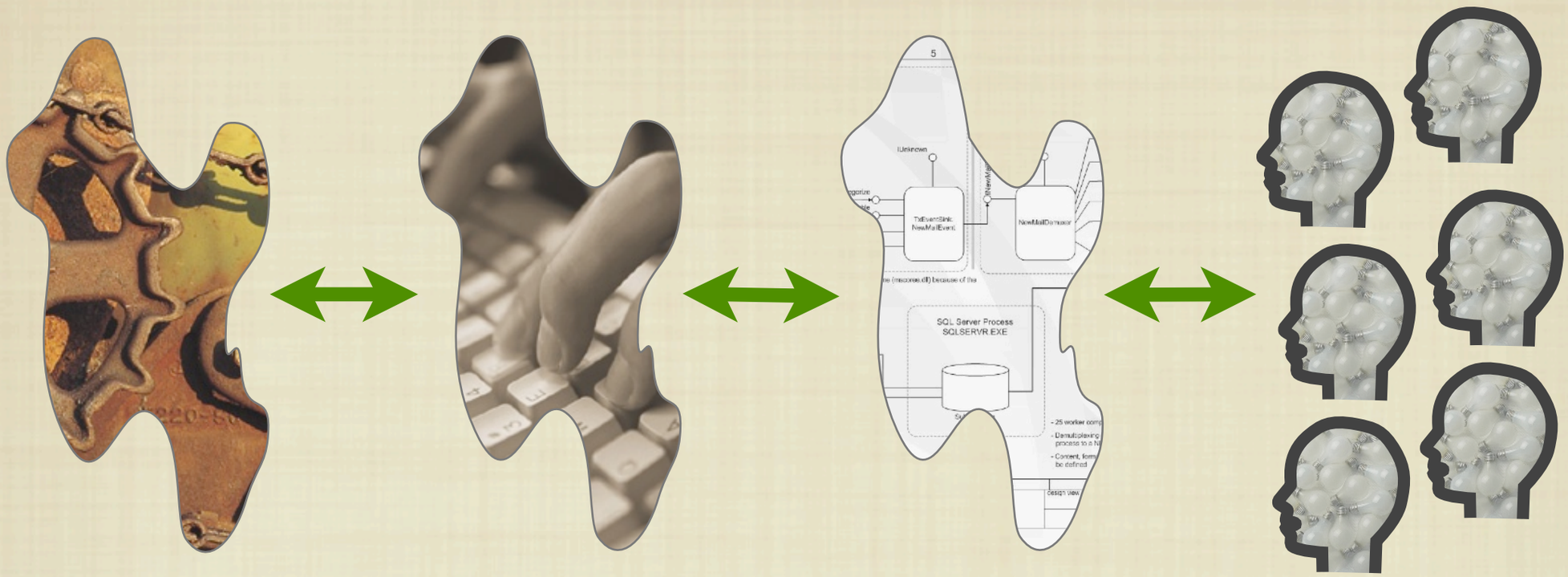
• do or plan (something) with a specific purpose or intention in mind: [ with obj. and infinitive ] : *the tax changes were designed to stimulate economic growth*.

# the individual's experience of design quality




| implementation | threshold | expectation | mindset |
|---|---|---|---|
| the assembled artifact's realization that creates the opportunity for observation | the point of encounter between the expectation and the system's features | the subset of the observer's mindset (conscious or unconscious) that is specifically relevant to the event | the "mental picture" the observer brings to the experience within which they will "understand" the experience |

# the community's experience of design quality




| implementation | threshold | expectation | mindset |
|---|---|---|---|
| the assembled artifact's realization that creates the opportunity for observation | the point of encounter between the expectation and the system's features | the subset of the observer's mindset (conscious or unconscious) that is specifically relevant to the event | the "mental picture" the observer brings to the experience within which they will "understand" the experience |

# Quality Design: The application of quality Principles in the process of creating artifacts

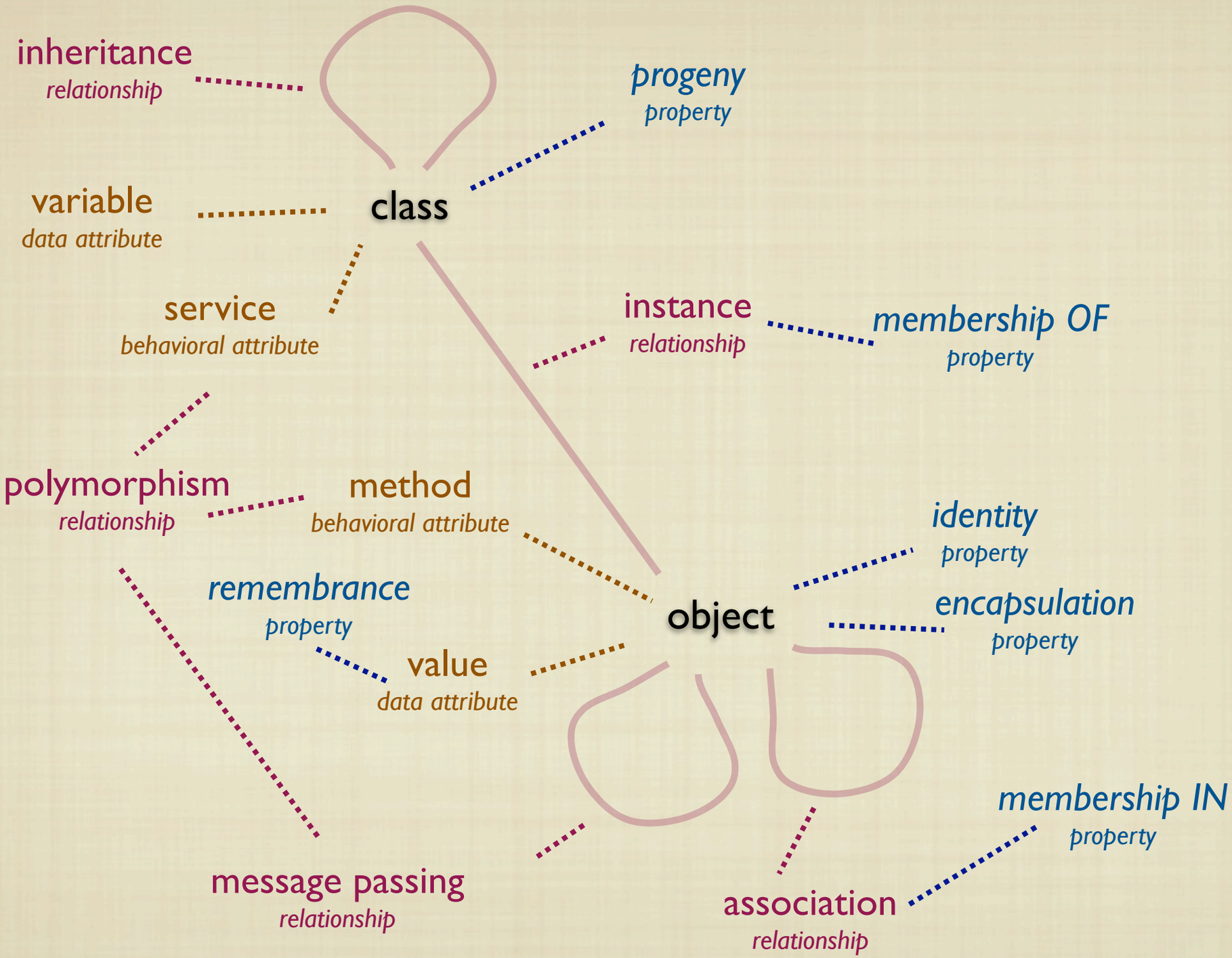
"How is it that one system is more effective, appealing, satisfying and/or more beautiful than another to its stakeholder community?"

Waguespack, L. J. (2010). Thriving Systems Theory and Metaphor-Driven Modeling. London: Springer-Verlag

# How is it that one object model is better than another?

# What determines design quality in an object model?

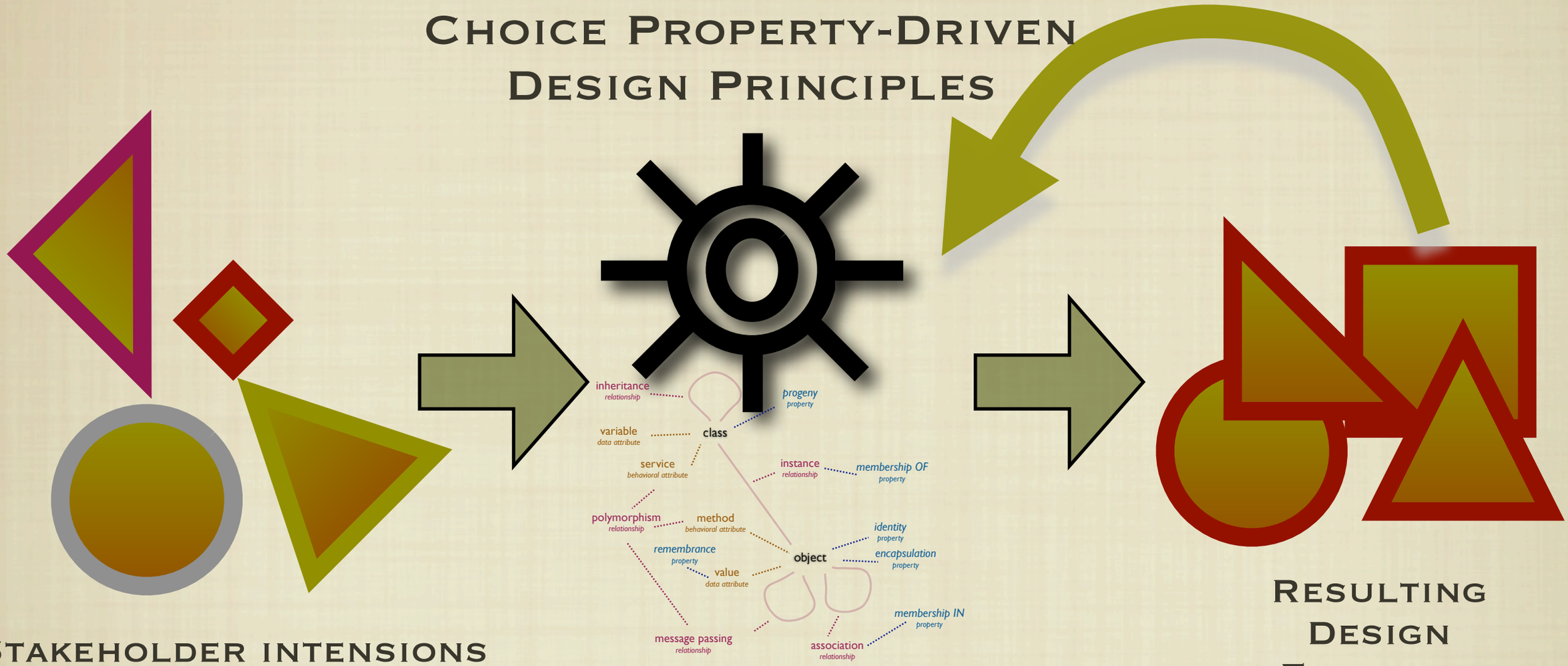
# Object-Oriented Ontology




Waguespack (2009). A Two-Page "OO Green Card" for Students and Teachers. Information Systems Education Journal, 7 (61).

# The Design Process

## Choice Property-Driven Design Principles




Stakeholder intensions
requirement elements
model elements
design elements

Representational
paradigm or Ontology

Resulting
Design
Elements

| | Choice Property |
|---|---|
| 1 | Stepwise Refinement |
| 2 | Cohesion |
| 3 | Encapsulation |
| 4 | Extensibility |
| 5 | Modularization |
| 6 | Correctness |
| 7 | Transparency |
| 8 | Composition of Function |
| 9 | Identity |
| 10 | Scale |
| 11 | User Friendliness |
| 12 | Patterns |
| 13 | Programmability |
| 14 | Reliability |
| 15 | Elegance |

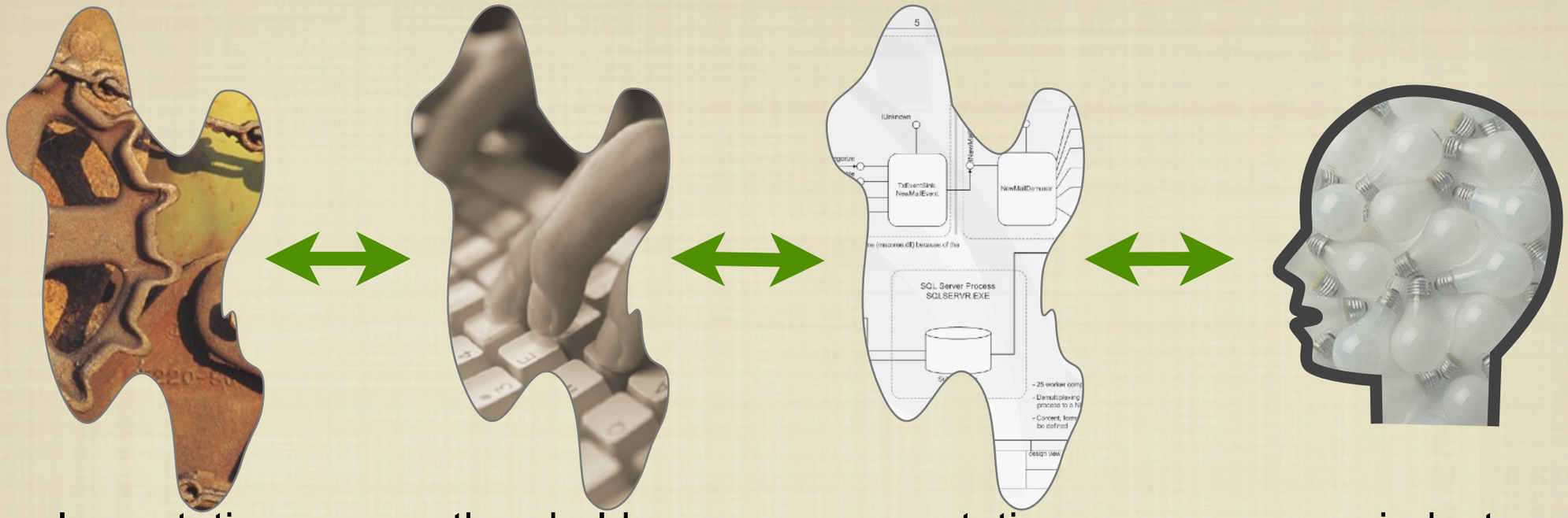
These choice properties propose a coherent, descriptive language including:

- a vocabulary for describing and comparing aspects of system components and structures, and
- design actions to guide design choices leading to desirable system characteristics.

| | Choice Property | Modeling Action | Action Rendition |
|---|---|---|---|
| 1 | Stepwise Refinement | elaborate | develop or present (a theory, policy, or system) in detail |
| 2 | Cohesion | factor | express as a product of factors |
| 3 | Encapsulation | encapsulate | enclose the essential features of something succinctly by a protective coating or membrane |
| 4 | Extensibility | extend | render something capable of expansion in scope, effect, or meaning |
| 5 | Modularization | modularize | employing or involving a module or modules as the basis of design or construction |
| 6 | Correctness | align | put (things) into correct or appropriate relative positions |
| 7 | Transparency | expose | reveal the presence of (a quality or feeling) |
| 8 | Composition of Function | assemble | fit together the separate component parts of (a machine or other object) |
| 9 | Identity | identify | establish or indicate who or what (someone or something) is |
| 10 | Scale | focus | (of a person or their eyes) adapt to the prevailing level of light [abstraction] and become able to see clearly |
| 11 | User Friendliness | accommodate | fit in with the wishes or needs of |
| 12 | Patterns | pattern | give a regular or intelligible form to |
| 13 | Programmability | generalize | make or become more widely or generally applicable |
| 14 | Reliability | normalize | make something more normal, which typically means conforming to some regularity or rule |
| 15 | Elegance | coordinate | bring the different elements of (a complex activity or organization) into a relationship that will ensure efficiency or harmony |

| | Choice Property | Modeling Action | Action Rendition Through Object Modeling |
|---|---|---|---|
| 1 | Stepwise Refinement | elaborate | employing class inheritance to expose and elaborate responsibilities and information management through levels of abstraction |
| 2 | Cohesion | factor | locating both service and data attributes for independent sufficiency |
| 3 | Encapsulation | encapsulate | delineate the responsibilities, knowledge, and interface of objects |
| 4 | Extensibility | extend | service abstractions that enable and control polymorphic extension |
| 5 | Modularization | modularize | individual and successive applications of encapsulation to compartmentalize design decisions and abstract system structure |
| 6 | Correctness | align | self-validating object interfaces that implement verification behavior |
| 7 | Transparency | expose | structural & behavioral relationships the show “fit” and “cooperation” |
| 8 | Composition of Function | assemble | design favoring simple parts combined for sophisticated function |
| 9 | Identity | identify | stakeholder visible constructs reflected in classes and relationships |
| 10 | Scale | focus | grouping objects & relationships in simplifying wrappers and facades |
| 11 | User Friendliness | accommodate | using user’s terminology and visible topology to maintain a familiarity that invites users into validation and verification |
| 12 | Patterns | pattern | nurturing familiarity and empowering evolution through polymorphism |
| 13 | Programmability | generalize | predicting and enabling adaptation of behavior without construction |
| 14 | Reliability | normalize | clearly distinguishing essential elements derived from business rules from artifacts necessary for technological compatibility or platform |
| 15 | Elegance | coordinate | satisfaction from an intuitively obvious design based not on having nothing else to add, but rather having nothing else that can be left out |

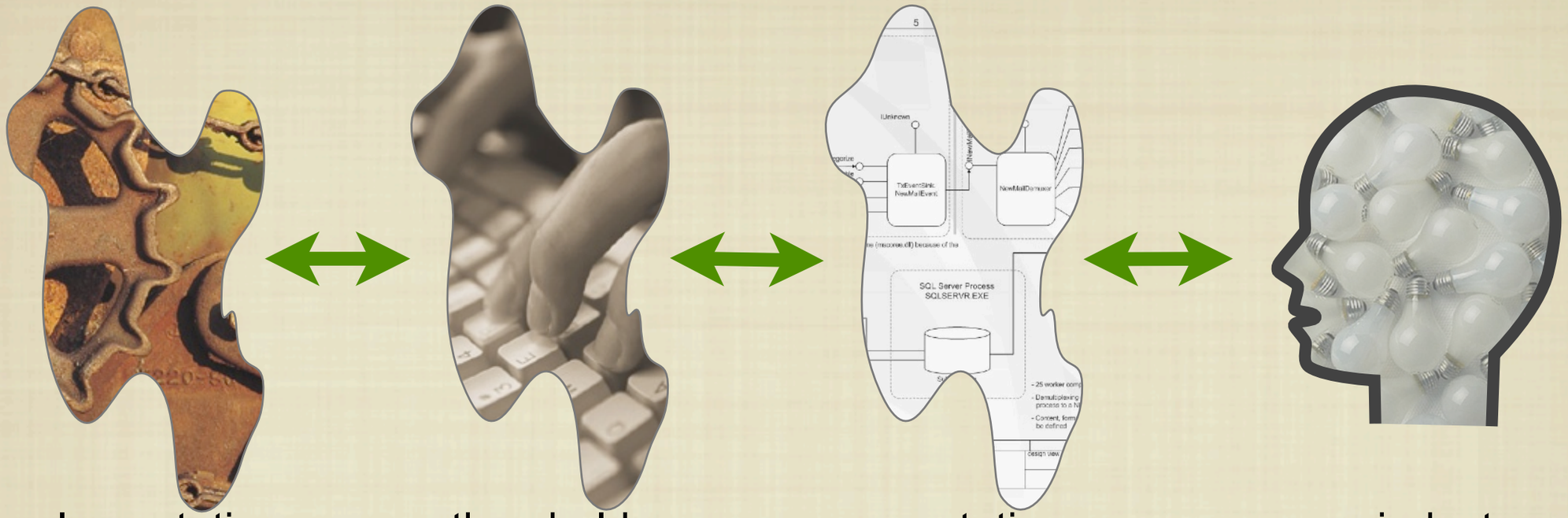
# design quality across the system models




implementation threshold expectation mindset

## Choice Property-Driven Design Principles

# design quality across the system models




implementation

threshold

expectation

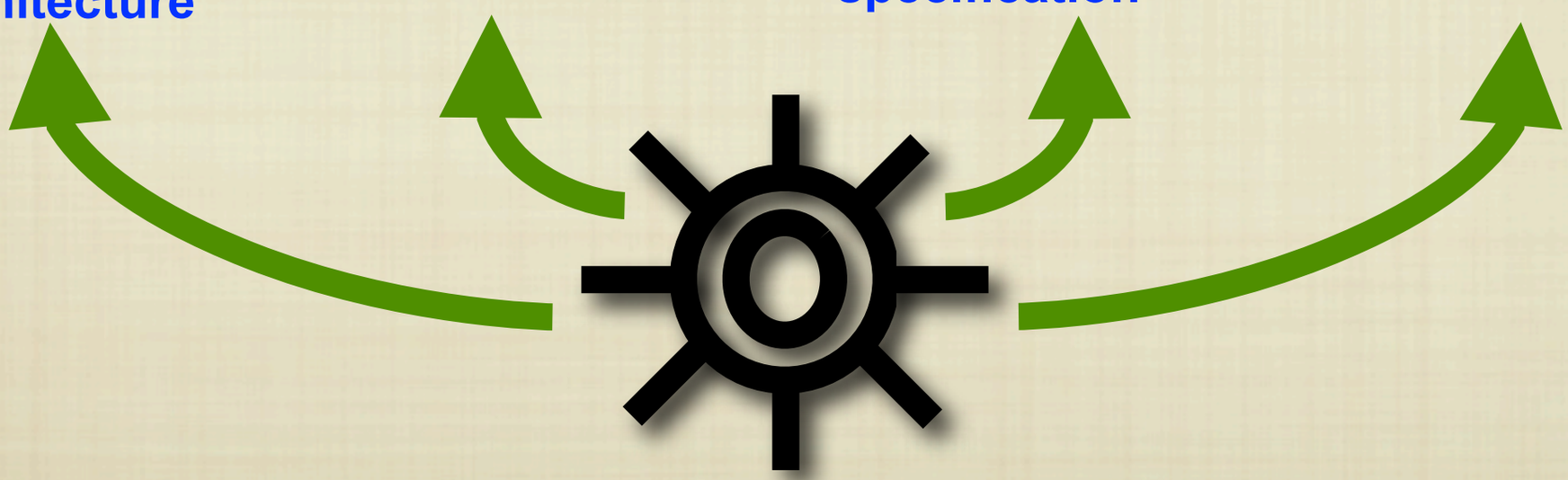
mindset

**hardware & software architecture**

**interfaces**

**requirements engineering & specification**

**object oriented "systems think"**



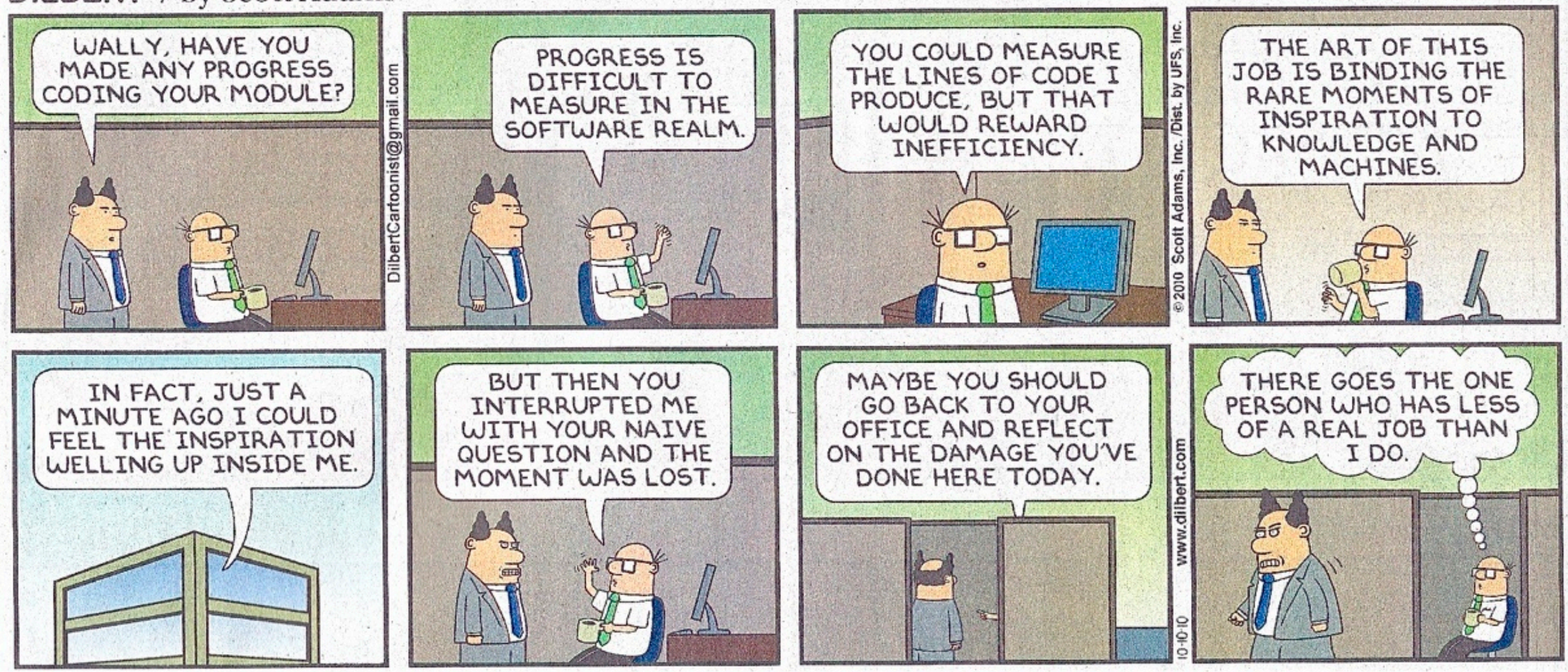
CHOICE PROPERTY-DRIVEN DESIGN PRINCIPLES

OCTOBER 10, 2010

Boston Sunday Globe

NEW ENGLAND'S LARGEST NEWSPAPER

**DILBERT®/ by Scott Adams**



Waguespack

## Thriving Systems Theory and Metaphor-Driven Modeling

How is it that one system is more effective, appealing, satisfying and/or more beautiful than another to its stakeholder community? This question drove Christopher Alexander's fifty-year quest to explain great physical architecture and give birth to pattern-languages for building that underpin much of modern systems engineering.

How is it that so many individual stakeholders consistently recognize the same quality, the same beauty in a system? This question led George Lakoff to research the role of conceptual metaphor in human understanding.

What is essential to stakeholders' satisfaction with systems? Fred Brooks addressed this question in No Silver Bullet: Essence and Accidents of Software Engineering.

This monograph fuses these diverse streams of thought in proposing Thriving Systems Theory by translating Alexander's properties of physical design quality into the abstract domain of information systems and modeling. Metaphor-Driven Modeling incorporates the theory while examining its impact throughout the system life cycle: modeling, design and deployment. The result is holistic and innovative, a perspective on system quality invaluable to students, practitioners and researchers of software and systems engineering.

Les Waguespack is a computer science Ph.D., professor and chairperson of computer information systems at Bentley University, USA. Dr. Waguespack's experience as programmer, software engineer, software architect, database architect, project manager and systems consultant underpins 35 years of teaching and research, the last 20+ years teaching object-oriented modeling and systems engineering to undergraduates, graduate students and practicing professionals.

ISBN 978-1-84996-301-5

9 781849 963015

springer.com

Waguespack

Thriving Systems Theory and Metaphor-Driven Modeling

Leslie J. Waguespack

# Thriving Systems Theory and Metaphor-Driven Modeling

Springer