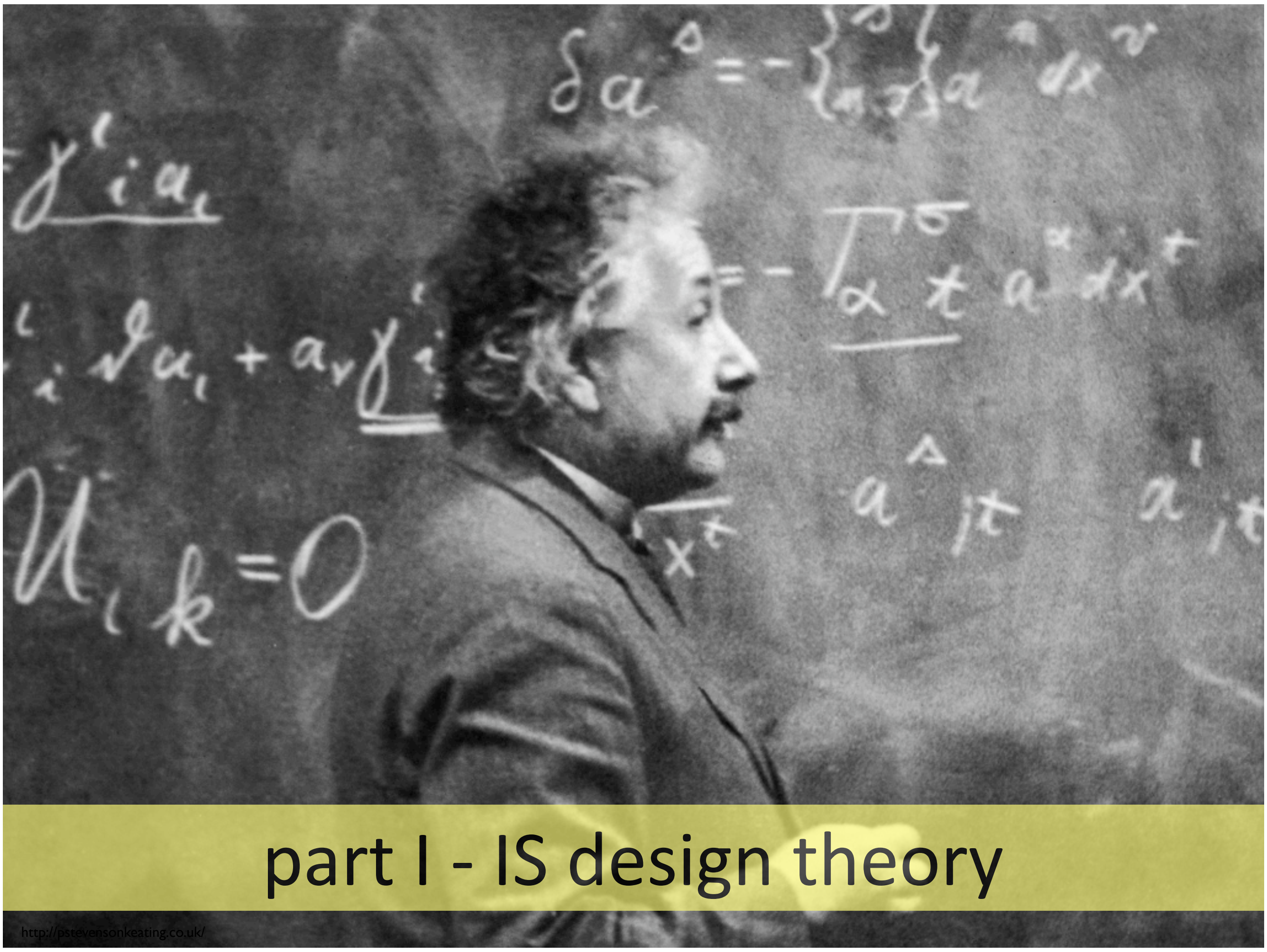
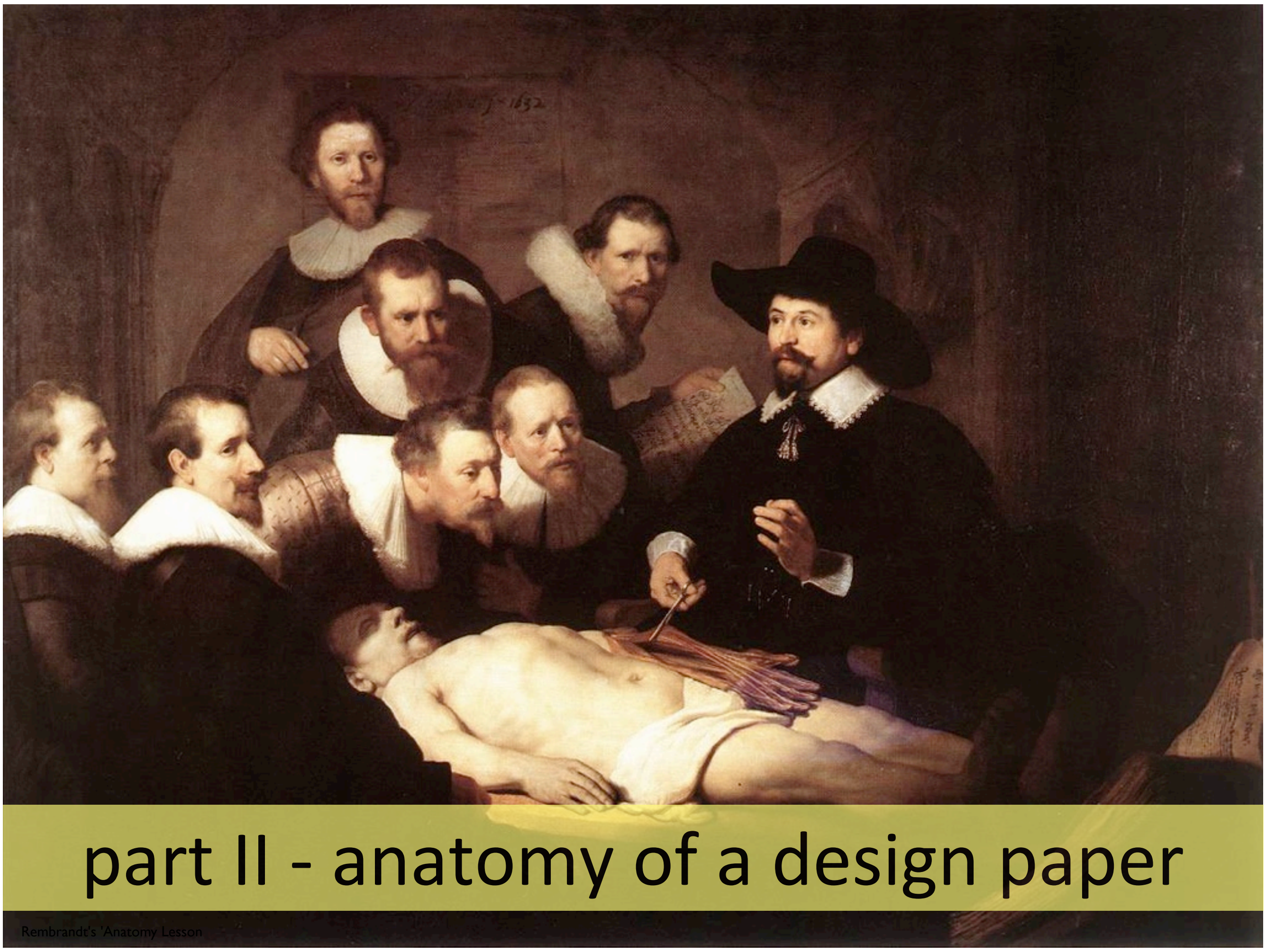
# design science
# research in IS

prepared by yves pigneur
DRIS March 2012

2

1. framework & publication
2. **theory & anatomy**
3. methodology & action design
4. patterns & evaluation
5. design rationale (C-K)


part I - IS design theory
http://psstevensonkeating.co.uk/


part II - anatomy of a design paper
Rembrandt's 'Anatomy Lesson

0

reminder

# questions

## what is the analogy with research?

- ✓ cynicism
- ✓ narcissism
- ✓ empathy

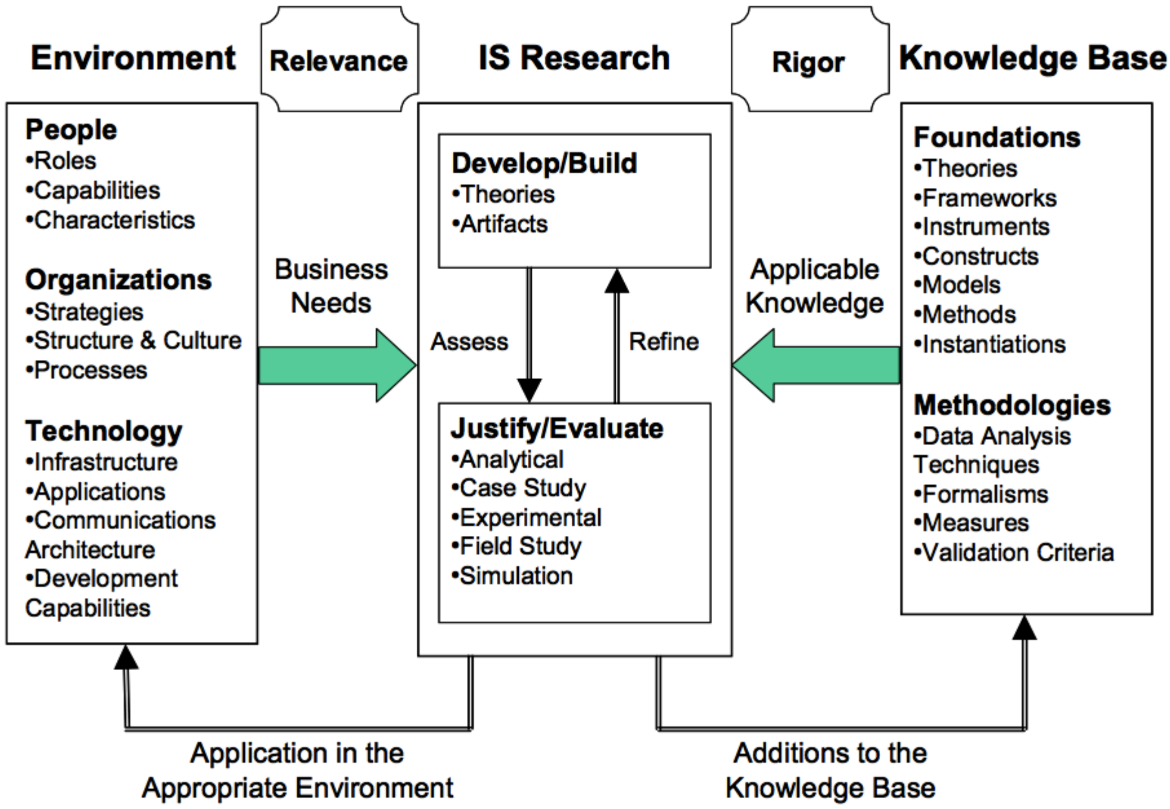
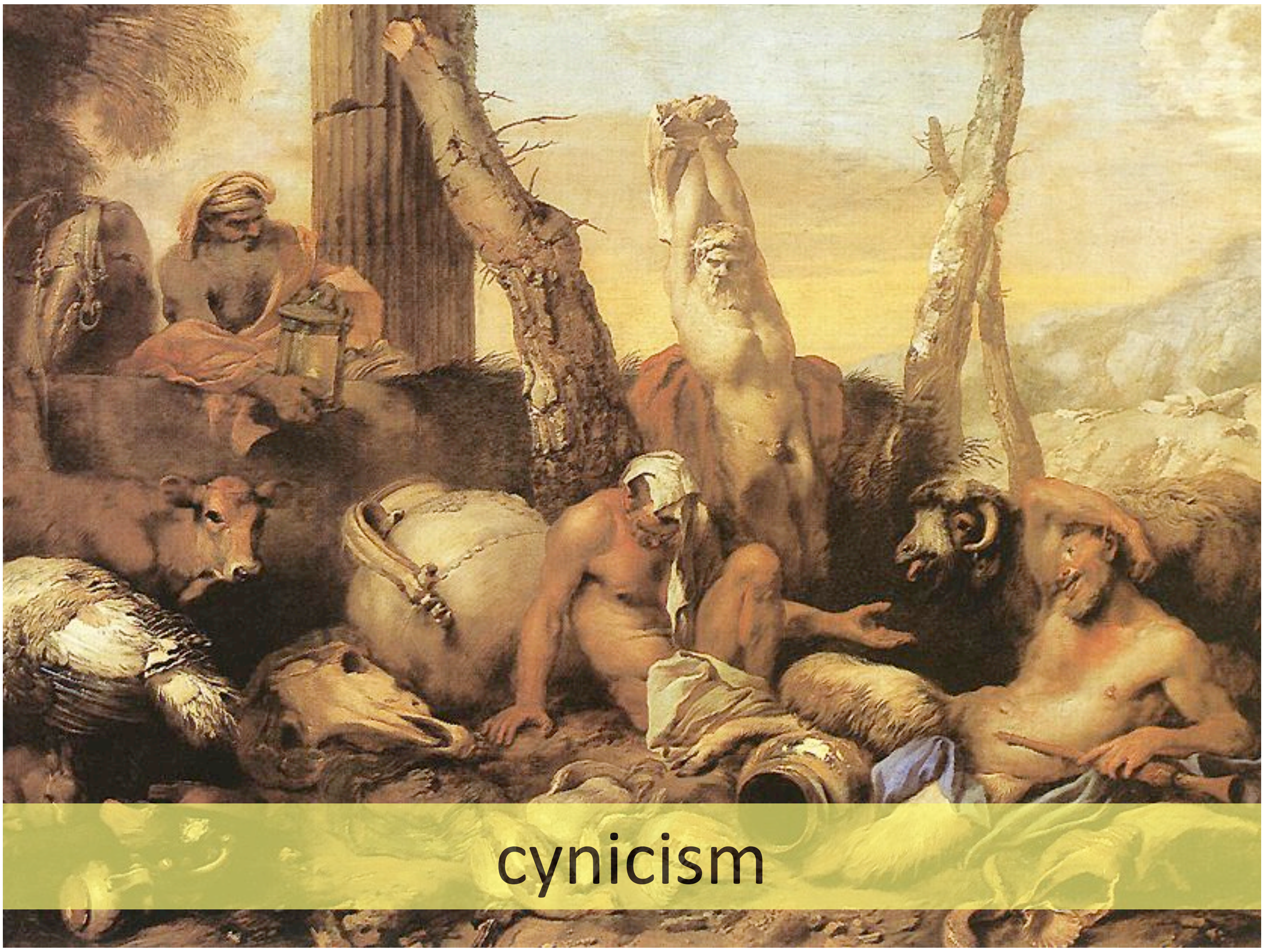



Figure 2. Information systems research framework



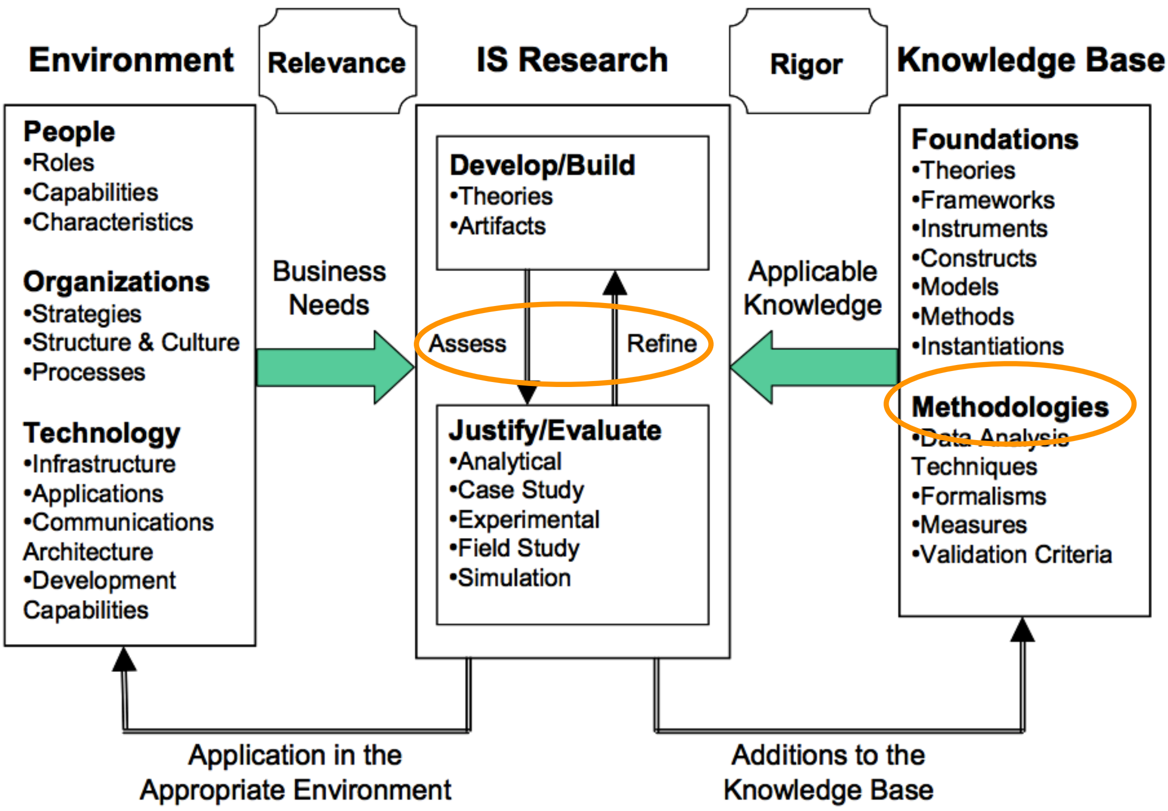
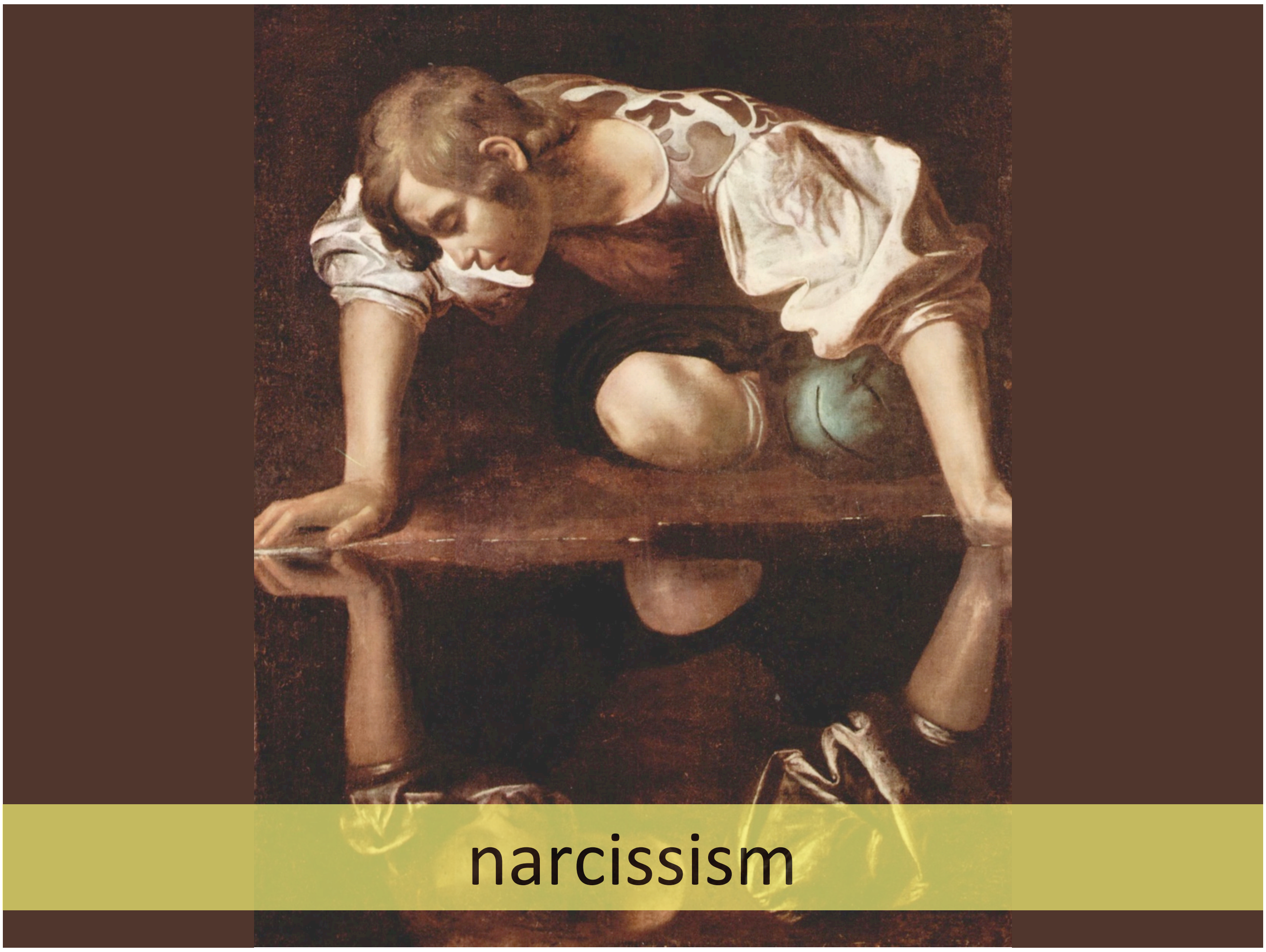




Figure 2. Information systems research framework

cynicism



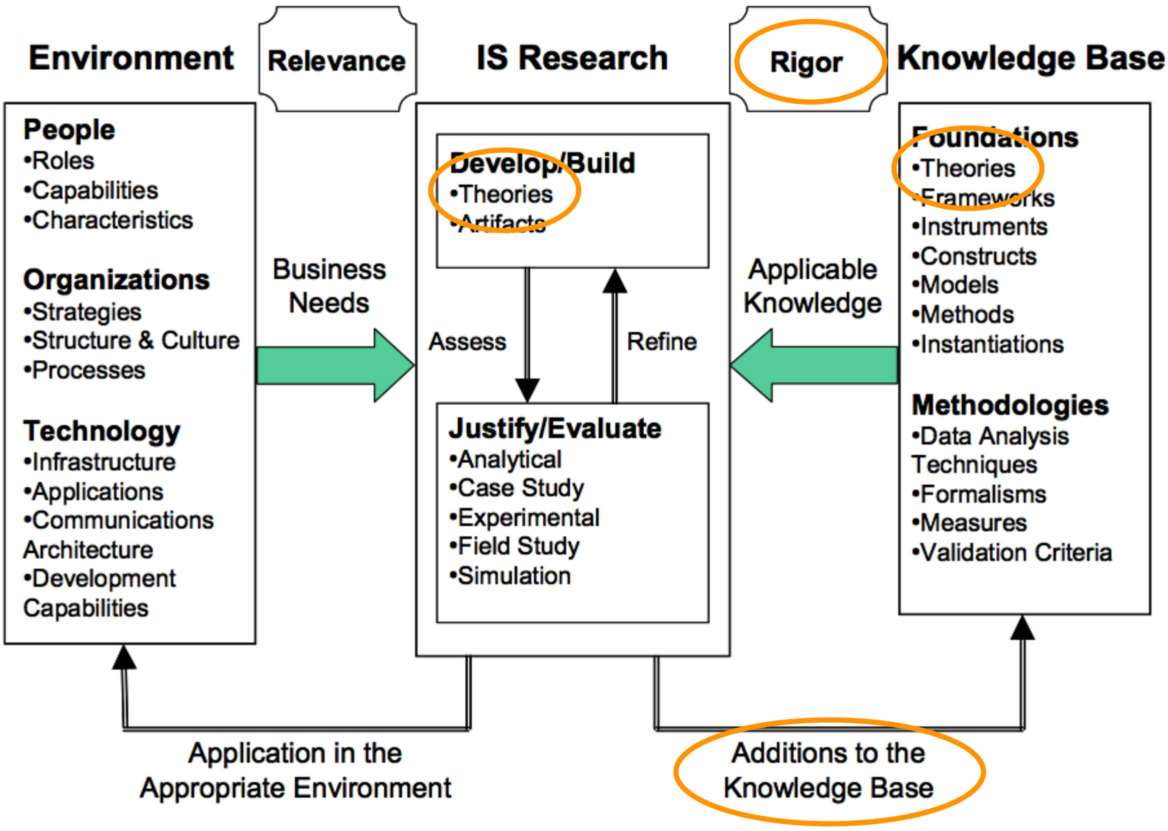
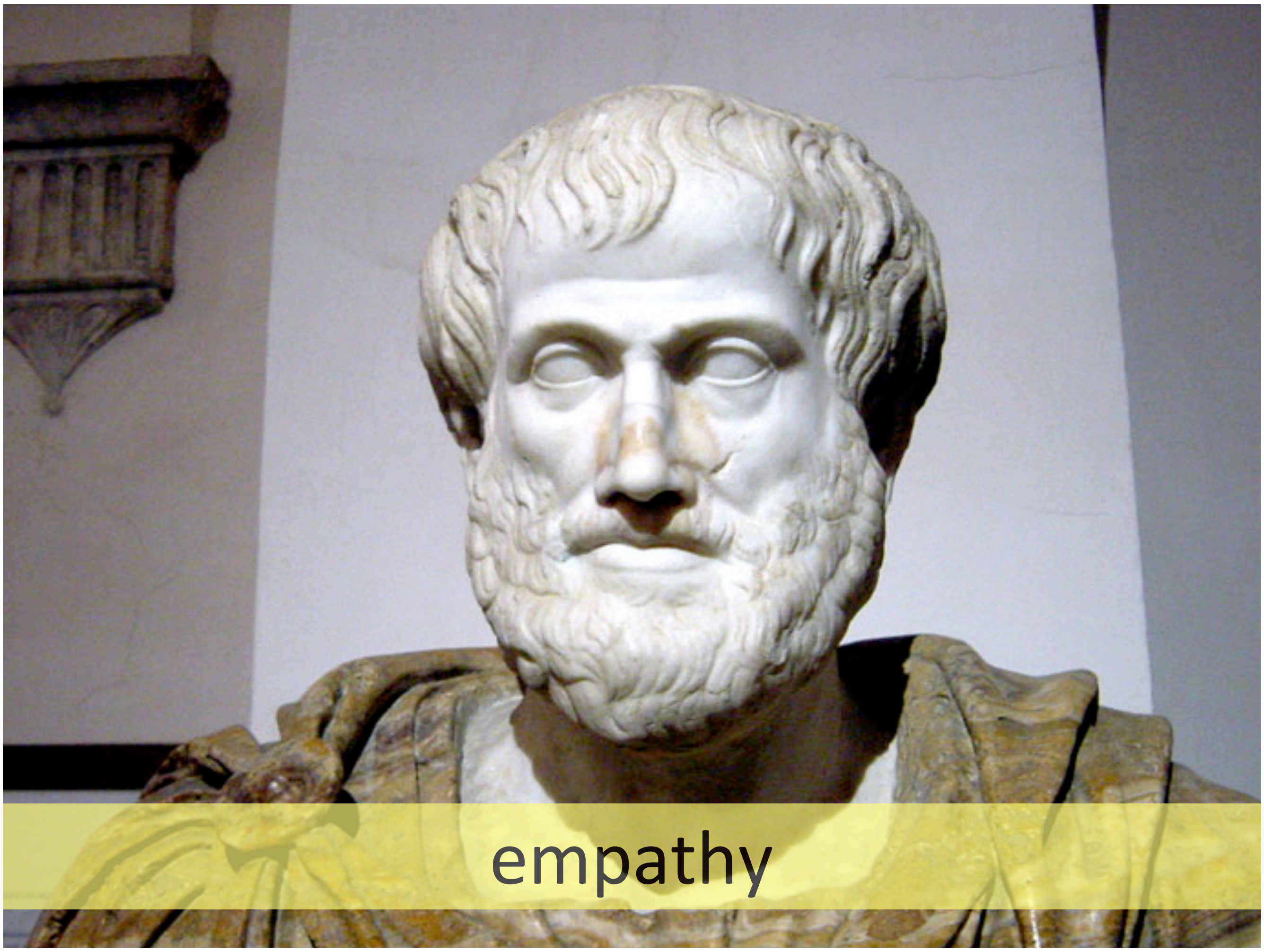




Figure 2. Information systems research framework

narcissism



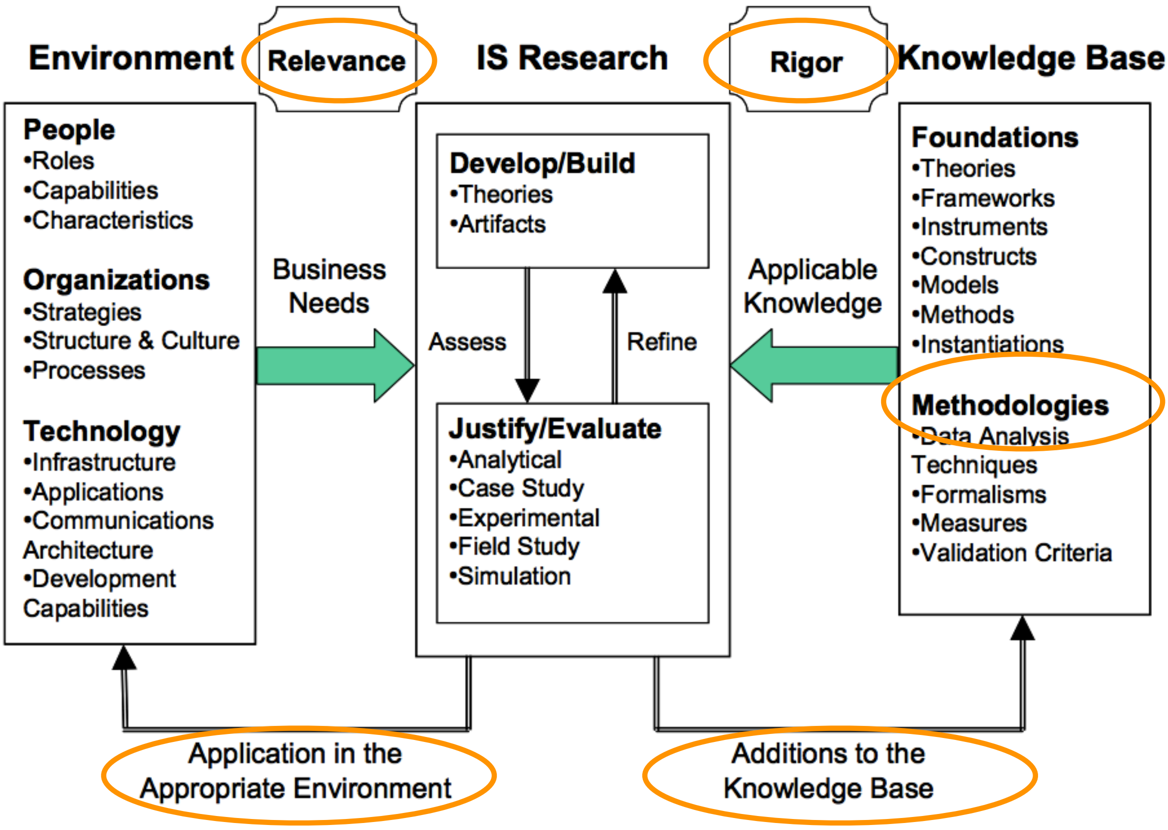
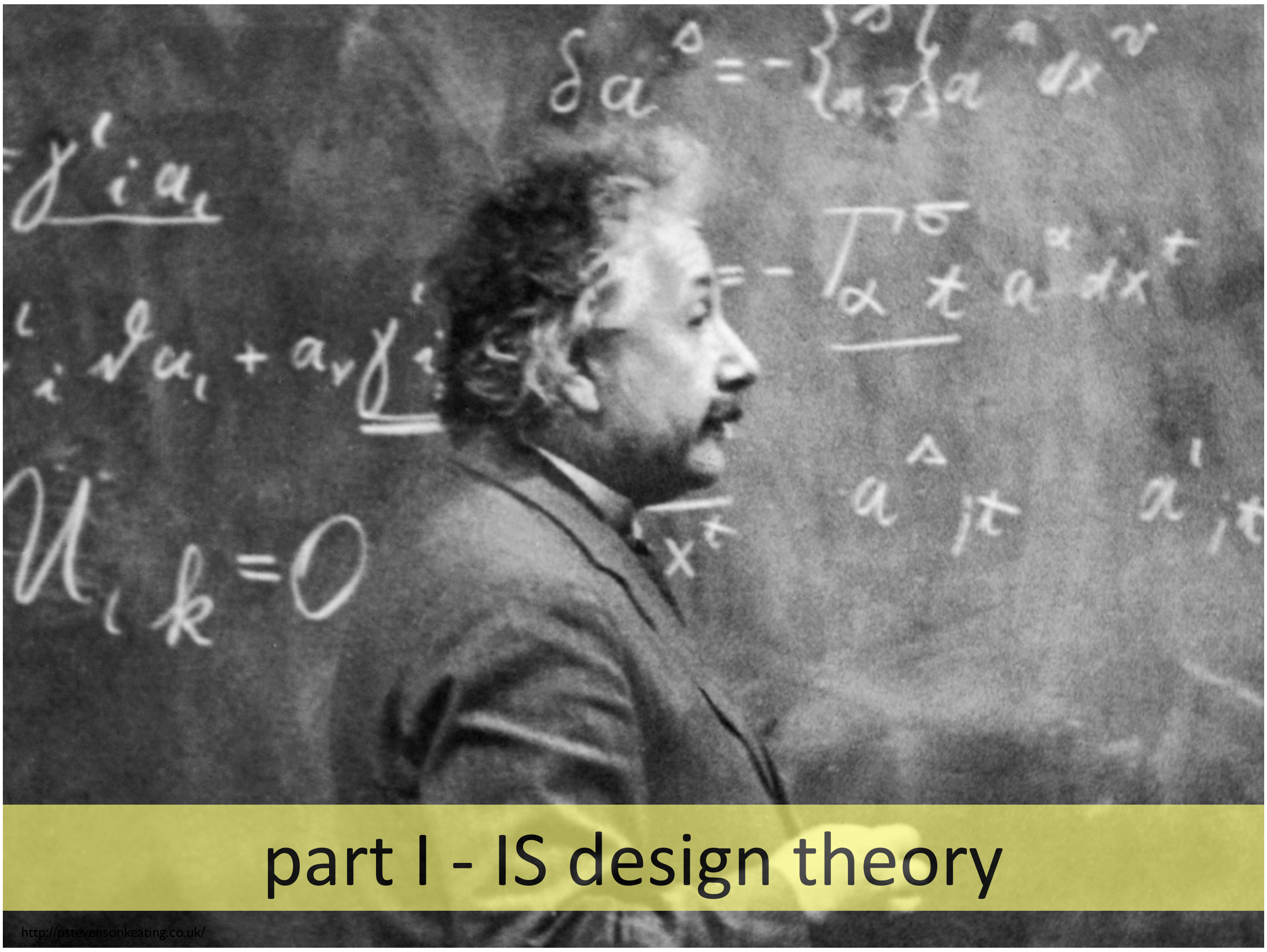




Figure 2. Information systems research framework

empathy




# 1

# IS design theory

# The anatomy of a design theory

Shirley Gregor & David Jones
*Journal of the Association for Information Systems, 2007, 8(5): 312–335*

▸ see also [Gregor, 2006] *The nature of theory in IS*

## questions

what is the focus of the paper?

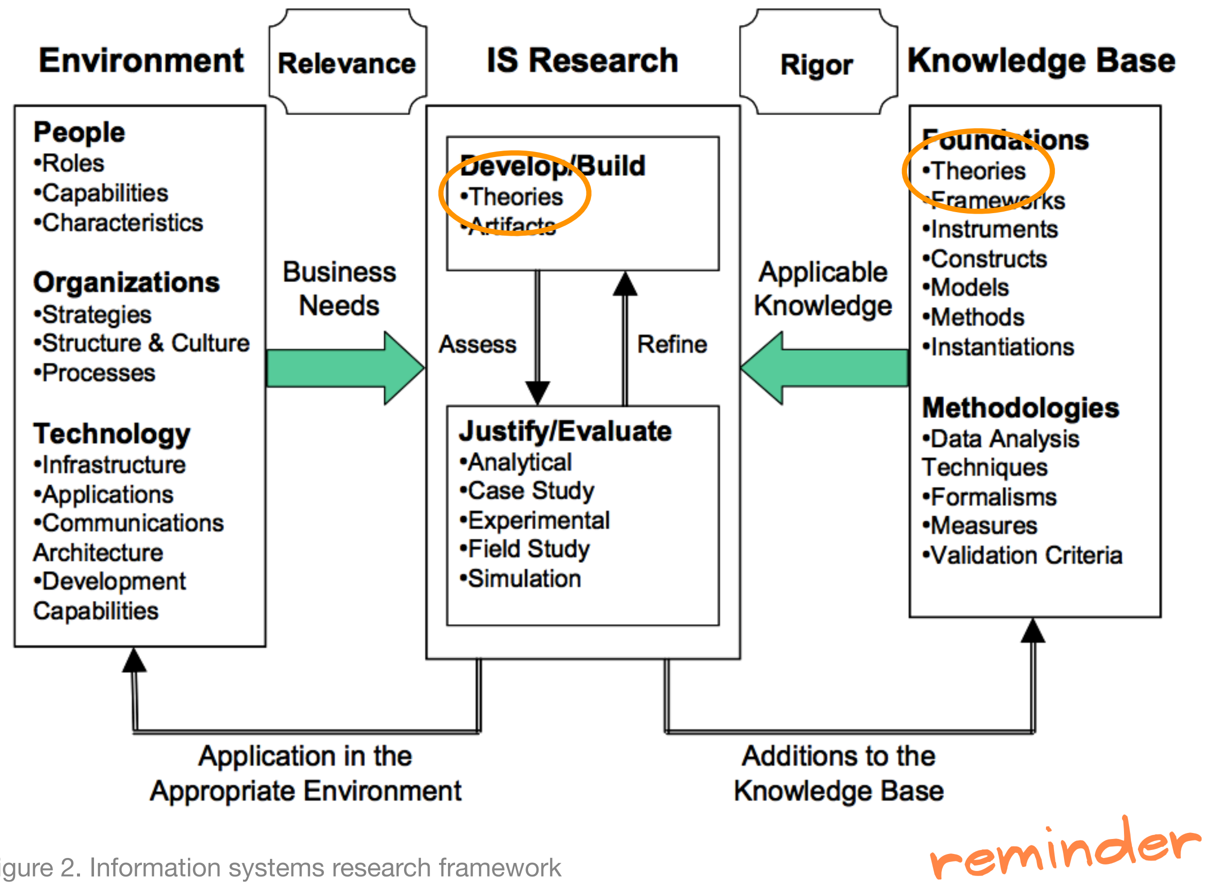



Figure 2. Information systems research framework

# focus on ...

- anatomy of design theories in the discipline of IS
- **how design knowledge can be expressed as theory**
- rigor and legitimacy of IS as an applied discipline
- an avenue for the more systematic specification of design knowledge
- understanding the nature of design theories
  (for better supporting cumulative building of knowledge)

## design science ...

“ ... always needs to contribute to theory ”

- [Gregor, 2007]

## design theory

“ Because design is a practice, a theory of design is not possible in the same sense as a theory for chemistry is.”

- [Hooker, 2003]

taxonomy of theory types in IS research
approaches to design theorizing
specification for IS design theory
components of an IS design theory

questions

---

how to classify theories in IS?

## the nature of theory in information systems

> “ classified with respect to the manner in which central goals are addressed:
> analysis, explanation, prediction, and prescription ”
>
> - [Gregor, 2006]

## components

I. theory for analyzing
II. theory for explaining
III. theory for predicting
IV. theory for explaining and predicting
V. theory for design and action

| Theory Type | Distinguishing Attributes |
|---|---|
| I. Analysis | Says what is.<br>The theory does not extend beyond analysis and description. No causal relationships among phenomena are specified and no predictions are made. |
| II. Explanation | Says what is, how, why, when, and where.<br>The theory provides explanations but does not aim to predict with any precision. There are no testable propositions. |
| III. Prediction | Says what is and what will be.<br>The theory provides predictions and has testable propositions but does not have well-developed justificatory causal explanations. |
| IV. Explanation and prediction (EP) | Says what is, how, why, when, where, and what will be.<br>Provides predictions and has both testable propositions and causal explanations. |
| V. Design and action | Says how to do something.<br>The theory gives explicit prescriptions (e.g., methods, techniques, principles of form and function) for constructing an artifact. |

Table 2: A Taxonomy of Theory Types in Information Systems Research

from [Gregor, 2006]

| Theory Component (Components Common to All Theory) | Definition |
|---|---|
| Means of representation | The theory must be represented physically in some way: in words, mathematical terms, symbolic logic, diagrams, tables or graphically. Additional aids for representation could include pictures, models, or prototype systems. |
| Constructs | These refer to the phenomena of interest in the theory (Dubin's "units"). All of the primary constructs in the theory should be well defined. Many different types of constructs are possible: for example, observational (real) terms, theoretical (nominal) terms and collective terms.* |
| Statements of relationship | These show relationships among the constructs. Again, these may be of many types: associative, compositional, unidirectional, bidirectional, conditional, or causal. The nature of the relationship specified depends on the purpose of the theory. Very simple relationships can be specified: for example, "x is a member of class A." |
| Scope | The scope is specified by the degree of generality of the statements of relationships (signified by modal qualifiers such as "some," "many," "all," and "never") and statements of boundaries showing the limits of generalizations. |
| **Theory Component (Components Contingent on Theory Purpose)** | **Definition** |
| Causal explanations | The theory gives statements of relationships among phenomena that show causal reasoning (not covering law or probabilistic reasoning alone). |
| Testable propositions (hypotheses) | Statements of relationships between constructs are stated in such a form that they can be tested empirically. |
| Prescriptive statements | Statements in the theory specify how people can accomplish something in practice (e.g., construct an artifact or develop a strategy). |

Table 3: Structural components of theory

from [Gregor, 2006]

taxonomy of theory types in IS research
approaches to design theorizing
specification for IS design theory
components of an IS design theory

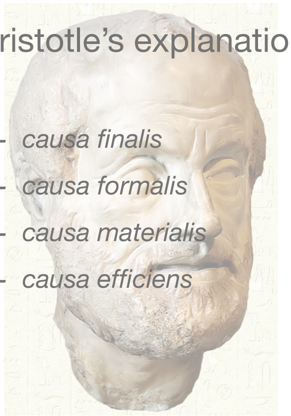
questions

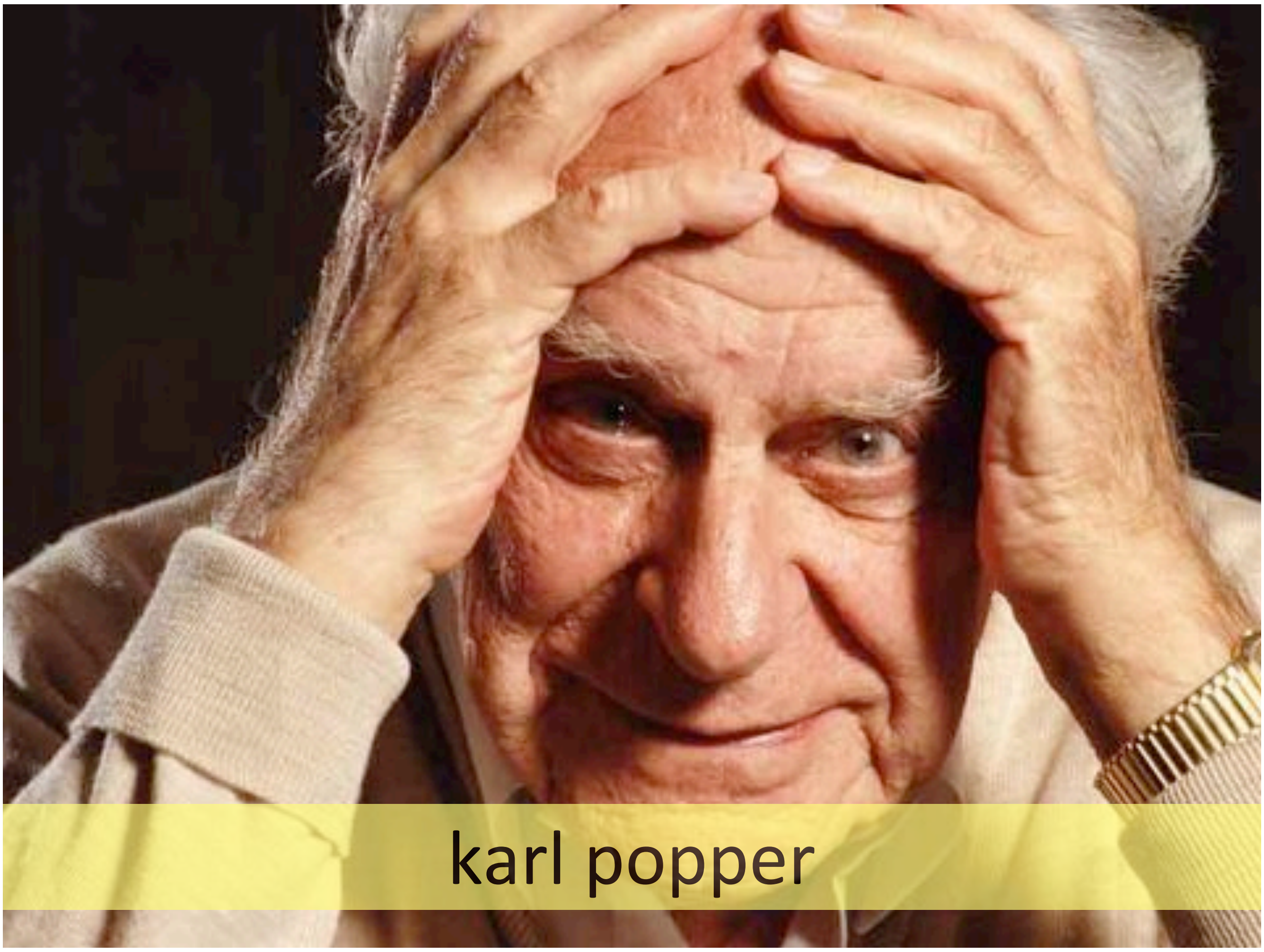
---

what are the perspectives the authors consider?

# philosophy of science and technology

- definition of theory (Popper, 1980)
- structural nature of theory and theory building (Dubin, 1978)
- theory might relate to technology (Bunge, 1979)
- Aristotle’s explanation of “thing” (Hooker, 1993)
  - *causa finalis*
  - *causa formalis*
  - *causa materialis*
  - *causa efficiens*

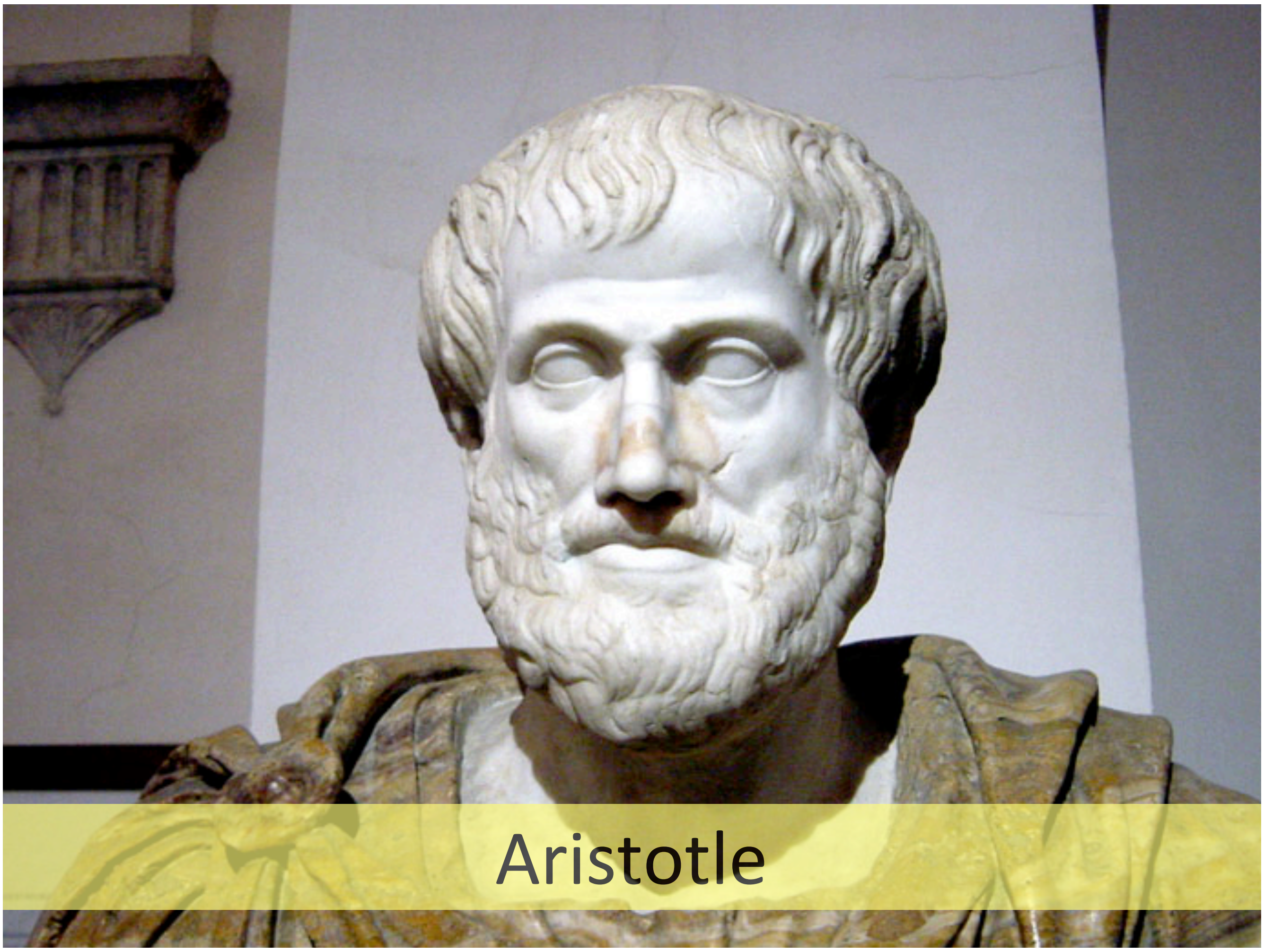




# theory

“ Scientific theories are universal statements. Like all linguistic representations they are systems of signs or symbols. Theories are nets cast to catch what we call ‘the world’; to rationalize, to explain and to master it. We endeavor to make the mesh even finer and finer. ”

- [Popper, 1980]



## things

*causa finalis* (purpose)

*causa formalis* (construct)

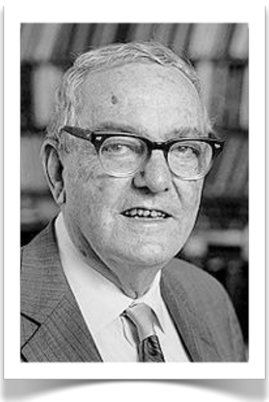
*causa materialis* (relationships)

*causa efficiens* (implementation)

- [Aristotle, -300]

## the science of artificial

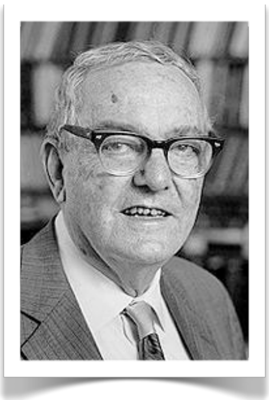
- description of an artifact (Simon, 1986)
- as a hierarchy of semi-independent components
- without a full understanding of the workings of its parts
- a theory of design only partly formalizable



from Hevner's talk in 2007 *"Design Science Research: Rigorous and Relevant "*

## theory of a design

> “ a theory of a system design does not depend on having an adequate micro-theory of the natural laws that govern the system components. Such a micro-theory might indeed be simply irrelevant. ”
>
> - [Simon, 1996]



## constructive research and design science

- design research as a knowledge-building activity
- system development approach to research & systemeering (Iivari, 1983)
- software engineering research with stages (Gregg, 2001)
- and design science (March and Smith, 1995) (Hevner *et al.*, 2004)

## role of the artifact as ...

- the main contribution ? or
- the “test” of a design theory ?

## work in other disciplines

- history of design-science concerns (Cross, 2001)
- design patterns in architecture (Alexander et al., 1977)
- prescriptive management theory and “technological rules” (van Aken, 2004, 2005)

## information systems design theory (ISDT)

- main inspiration: (Walls et al., 1992)
- components
  - *meta-requirements*
  - *meta-design*
  - *design method*
  - *kernel design (product & process) theories*
  - *testable design (product & process) hypotheses*

taxonomy of theory types in IS research
approaches to design theorizing
specification for IS design theory
components of an IS design theory

# questions

what are the phenomena of interest for design research?

what is the compatibility with *"constructs, models, methods and instantiations"* artifacts as proposed by (March and Smith, 1995) (Hevner et al,, 2004) ?

what is the definition of IS design theory?

what are the "anatomy of an IS design theory"?

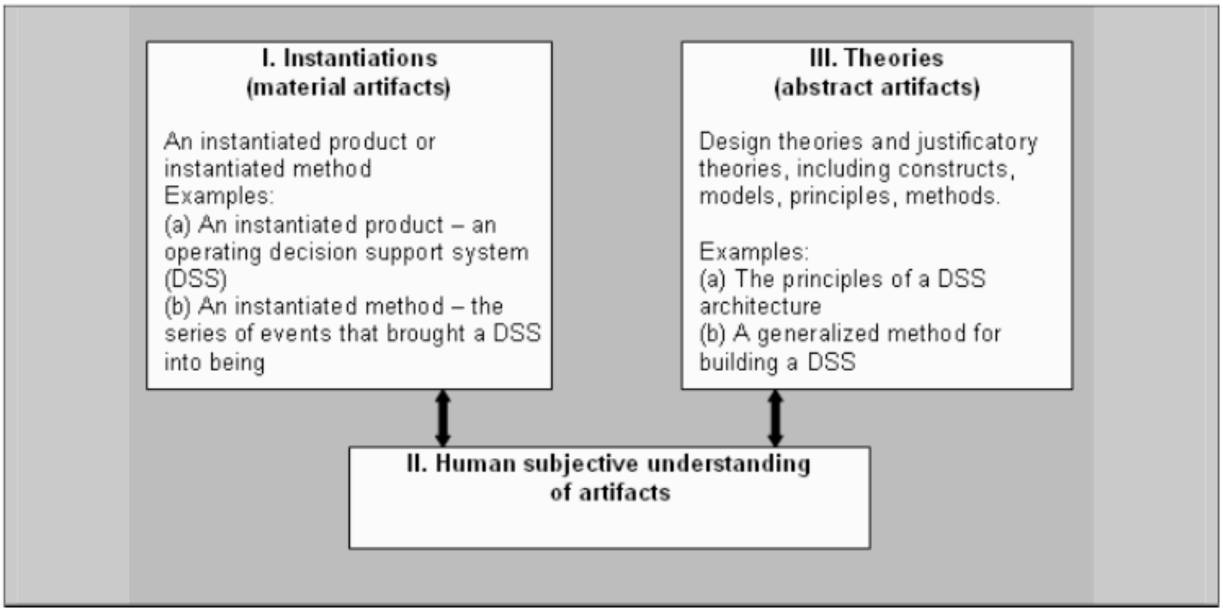



Figure 1: Relationships among IS/IT artefacts

# IS design theory ...

> “ ... the principles inherent in the design of an IS artifact that accomplishes some end, based on knowledge of both IT and human behavior. The ISDT allows the prescription of guidelines for further artifacts of the same type. ”

- [Gregor and Jones, 2007]

# components of an IS design theory

| | |
|---|---|
| **1. Purpose & scope** | core components |
| **2. Constructs** | |
| **3. Principles of form and function** | |
| **4. Artifact mutability** | |
| **5. Testable propositions** | |
| **6. Justificatory knowledge** | |
| **7. Principles of implementation** | additional components |
| **8. Expository instantiation** | |

| Article details | The design theory anatomy |
|---|---|
| The introduction says better database technology is needed to increase human productivity. (Motivation is also provided: This need is significant because current approaches are failing.) | The **purpose and scope** of the theory are stated. |
| The relational database model has principles such as "the order of rows in the tables is arbitrary and irrelevant." | **Principles of form and function** incorporating underlying **constructs** (such as "table") are given. |
| The argument is made that the relational model allows for relatively simple adaptation and change to base tables, while user views appear unchanged. | **Artifact mutability** is addressed. |
| Statements are made such as "A relational database can perform as well as a non-relational database." | These statements are **testable propositions**. |
| It is shown how the relational model works, by reference to underlying set theory and also human cognitive processes. | **Justificatory knowledge** (kernel theory) is provided. |
| Guidelines are given on how to produce a relational database through normalization procedures. | **Principles of implementation** are given. |
| An illustration of working relational databases is provided. | An **expository instantiation** is given. |

Table 1: example of skeleton of a design theory (from Codd, 1970, 1982)

| Component | Description |
|---|---|
| Core components | |
| 1) Purpose and scope (the *causa finalis*) | "What the system is for," the set of meta-requirements or goals that specifies the type of artifact to which the theory applies and in conjunction also defines the scope, or boundaries, of the theory. |
| 2) Constructs (the *causa materialis*) | Representations of the entities of interest in the theory. |
| 3) Principle of form and function (the *causa formalis*) | The abstract "blueprint" or architecture that describes an IS artifact, either product or method/intervention. |
| 4) Artifact mutability | The changes in state of the artifact anticipated in the theory, that is, what degree of artifact change is encompassed by the theory. |
| 5) Testable propositions | Truth statements about the design theory. |
| 6) Justificatory knowledge | The underlying knowledge or theory from the natural or social or design sciences that gives a basis and explanation for the design (kernel theories). |
| Additional components | |
| 7) Principles of implementation (the *causa efficiens*) | A description of processes for implementing the theory (either product or method) in specific contexts. |
| 8) Expository instantiation | A physical implementation of the artifact that can assist in representing the theory both as an expository device and for purposes of testing. |

Table 2: Eight components of an IS design theory

taxonomy of theory types in IS research
approaches to design theorizing
specification for IS design theory
components of an IS design theory

| Component | Description |
|---|---|
| Core components | |
| 1) Purpose and scope (the *causa finalis*) | "What the system is for," the set of meta-requirements or goals that specifies the type of artifact to which the theory applies and in conjunction also defines the scope, or boundaries, of the theory. |
| 2) Constructs (the *causa materialis*) | Representations of the entities of interest in the theory. |
| 3) Principle of form and function (the *causa formalis*) | The abstract "blueprint" or architecture that describes an IS artifact, either product or method/intervention. |
| 4) Artifact mutability | The changes in state of the artifact anticipated in the theory, that is, what degree of artifact change is encompassed by the theory. |
| 5) Testable propositions | Truth statements about the design theory. |
| 6) Justificatory knowledge | The underlying knowledge or theory from the natural or social or design sciences that gives a basis and explanation for the design (kernel theories). |
| Additional components | |
| 7) Principles of implementation (the *causa efficiens*) | A description of processes for implementing the theory (either product or method) in specific contexts. |
| 8) Expository instantiation | A physical implementation of the artifact that can assist in representing the theory both as an expository device and for purposes of testing. |

Table 2: Eight components of an IS design theory

questions

how to define the components?

how to illustrate them by references to examples?

# 1. purpose and scope

- “what the system is for”
- the set of meta-requirements or goals that specifies the type of artifact to which the theory applies and in conjunction also defines the scope, or boundaries, of the theory
- context-dependent
- provides guidance for evaluation and contribution

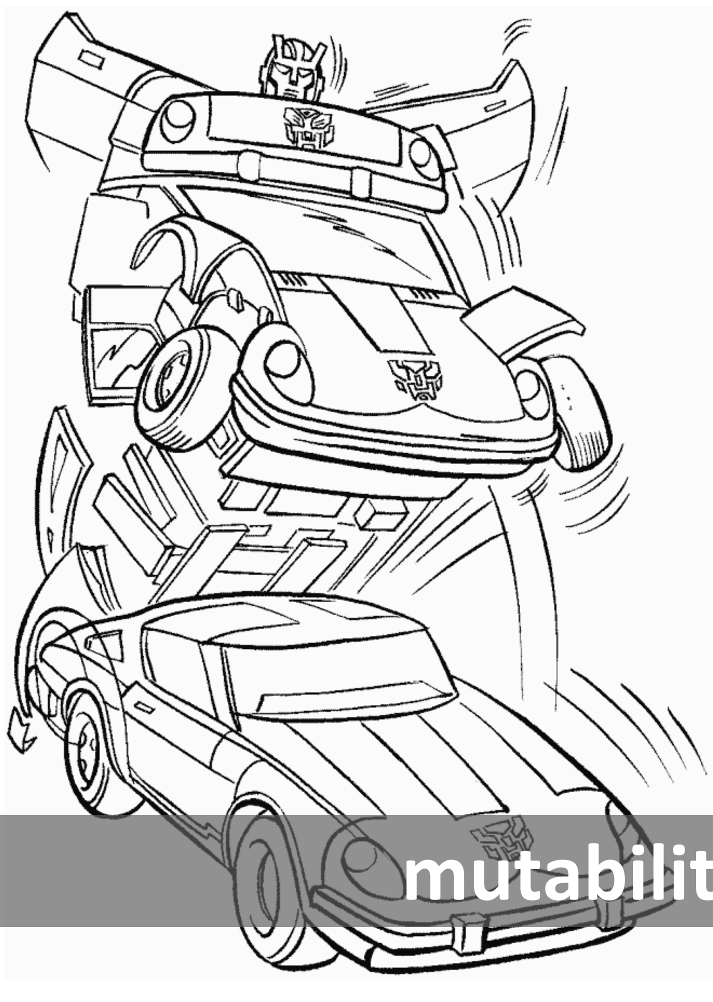
# 2. constructs

- representations of the entities of interest in the theory
- words, symbols, diagrams ...
- clear definition
- decomposing problems in semi-independent parts

**forms and functions**

# 3. principles of form and function

- the abstract “blueprint” or architecture that describes an IS artifact, either product or method/intervention
- structural and functional properties



**mutability**

# 4. artifact mutability

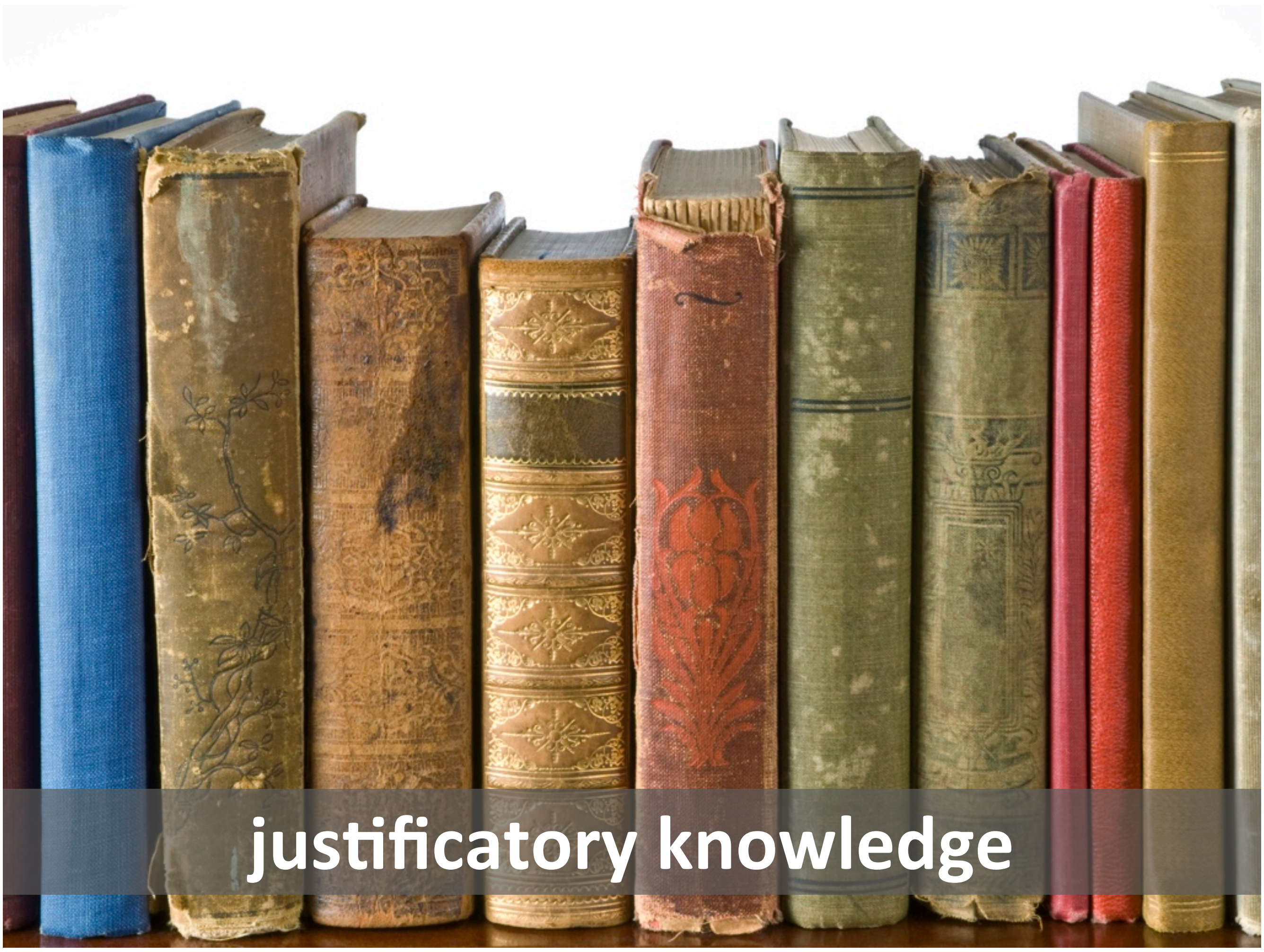
- the changes in state of the artifact anticipated in the theory, that is, what degree of artifact change is encompassed by the theory

- evolution is a key unresolved issue

# 5. testable propositions

- truth statements about the design theory
    1. whether the meta-design satisfies the meta-requirements
    2. whether the artifact is consistent with the meta-design
- generality is a key issue

falsifiability

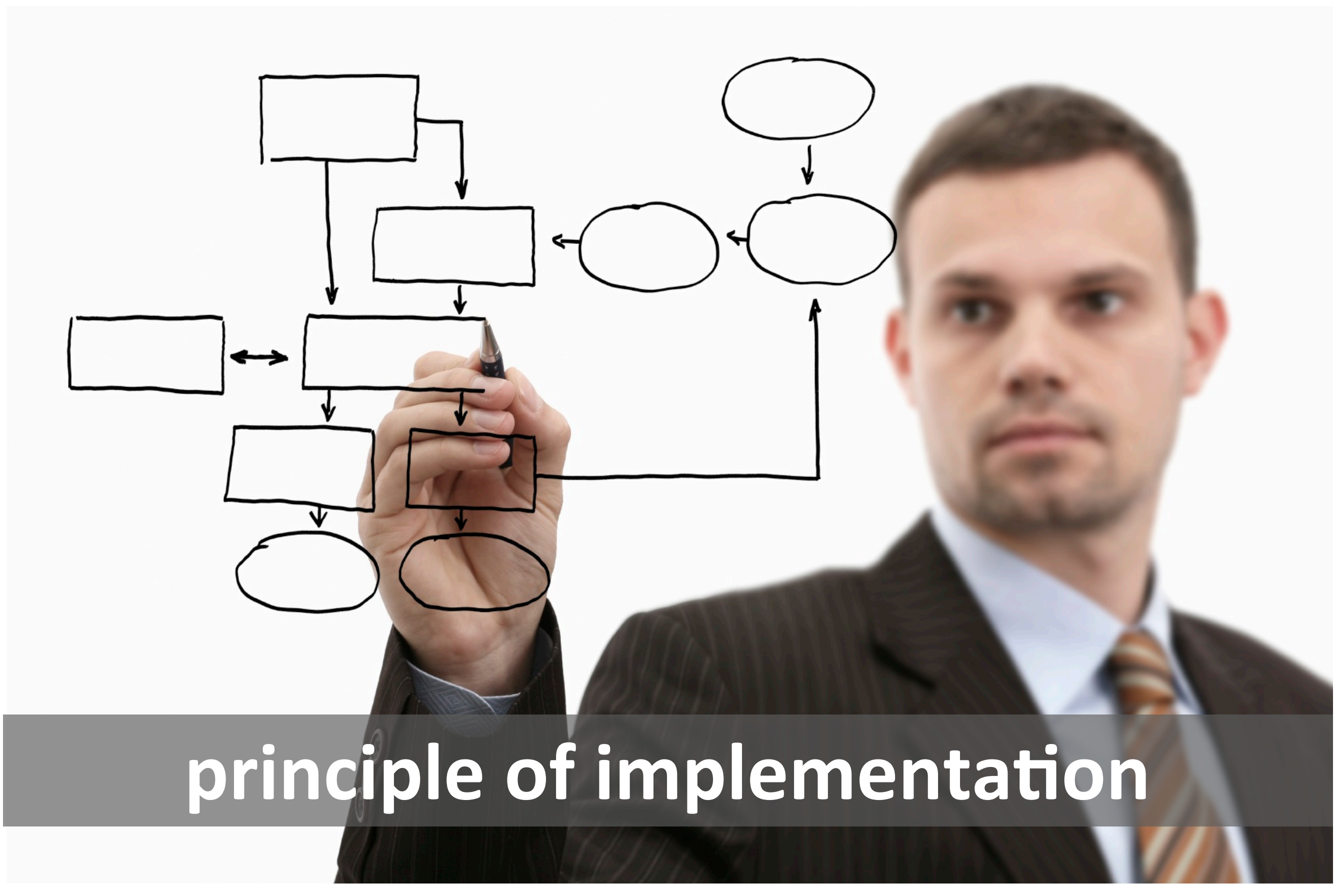


# 6. justificatory knowledge

- the underlying knowledge or theory from the natural or social or design sciences that gives a basis and explanation for the design (kernel theories)

- theory or not theory? this is the question ...

- from behavioral sciences or other design sciences?



# 7. principles of implementation

- a description of processes for implementing the theory (either product or method) in specific contexts

- generic or customized process

# 8. expository instantiation

- a physical implementation of the artifact that can assist in representing the theory both as an expository device and for purposes of testing
- prototype, mock-ups, and scenarios of real systems to help explain the design

| | *Type* | *Component examples* |
|---|---|---|
| (1) | Purpose and scope | The aim is to develop a fault threshold policy to determine when system integration occurs during a process of incremental systems development. The policy is developed for homogeneous systems, where modules are similar in size and complexity and all faults take roughly the same effort to fix. The policy is appropriate for systems that can be tested frequently and at relatively low cost. The policy is designed to consider a number of project parameters (such as complexity). |
| (2) | Constructs | Examples are: incremental development, system integration, fault threshold, testing, faults detected. |
| (3) | Principles of form and function | The policy uses a derived expression to give dynamic guidelines for when system integration should occur, with (1) a region of no integration, (2) a region where integration occurs depending on a fault count, and (3) a region in which systems integration should always take place. |
| (4) | Artifact mutability | The designers consider the effects of team learning that occur over multiple construction cycles and show how the policy will vary over a number of cycles. |
| (5) | Testable propositions | Predictions about outcomes are provided that are tested in simulation experiments. |
| (6) | Justificatory knowledge | Theory is offered relating to group coordination processes, team cognition, software development productivity, and fault growth models. |
| (7) | Principles of implementation | Not a great deal of detail is given on how to build a concrete version of this abstract policy in specific projects. An example is given where the formulae in the policy are applied to an imaginary scenario. It is stated that it might be necessary to build some randomness into the model in a real-life project and this is left for further work. |
| (8) | Expository instantiation | Examples of the policy in action are provided through simulations. |

Table 4: Components of a design theory for a software threshold fault theory

▸ exercice

from [Chiang and Mookerjee, 2004]

taxonomy of theory types in IS research
approaches to design theorizing
specification for IS design theory
components of an IS design theory
concluding remarks

## concluding remarks

- role of the artifact for expository purpose
- mutability of the artifact is an issue
- guidelines to write papers and thesis ?
- lessons
  - *importance of specifying goals*
  - *"search" nature of theory building for designs*
  - *role of creativity and imagination: (not) compatible with science?*

article anatomy

# IS design theory ...

> “ ... helps define what is unique about the IS discipline, namely, the construction of mutable artifacts where complexity arises from the interaction of humans with information technology. ”

- [Gregor and Jones, 2007]

| Component | Description |
| --- | --- |
| Core components | |
| 1) Purpose and scope (the *causa finalis*) | “What the system is for,” the set of meta-requirements or goals that specifies the type of artifact to which the theory applies and in conjunction also defines the scope, or boundaries, of the theory. |
| 2) Constructs (the *causa materialis*) | Representations of the entities of interest in the theory. |
| 3) Principle of form and function (the *causa formalis*) | The abstract “blueprint” or architecture that describes an IS artifact, either product or method/intervention. |
| 4) Artifact mutability | The changes in state of the artifact anticipated in the theory, that is, what degree of artifact change is encompassed by the theory. |
| 5) Testable propositions | Truth statements about the design theory. |
| 6) Justificatory knowledge | The underlying knowledge or theory from the natural or social or design sciences that gives a basis and explanation for the design (kernel theories). |
| Additional components | |
| 7) Principles of implementation (the *causa efficiens*) | A description of processes for implementing the theory (either product or method) in specific contexts. |
| 8) Expository instantiation | A physical implementation of the artifact that can assist in representing the theory both as an expository device and for purposes of testing. |

Table 2: Eight components of an IS design theory

# part II - anatomy of a design paper

Rembrandt's 'Anatomy Lesson

# 2

# anatomy of a paper

# Anatomy of a design science paper

Shirley Gregor & David Jones

*a research note*

- see also [Vaishnavi and Kuechler, 2007] *Design science research methods and patterns*

idealized structure

example

## questions

# what is the ideal structure of a design paper?

| | | |
|---|---|---|
| 1. introduction | the purpose or goal and scope of the theory | ISDT component (1)  |
| | definition of constructs | ISDT component (2)  |
| | motivation, significance, outline of article | similar to conventional articles |

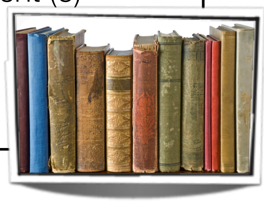
from Gregor & Jones, 2008 *The anatomy of a design science paper: a research note*

| | | |
|---|---|---|
| 2. literature review background | what was known about these systems before work in this article was begun | similar to conventional articles |
| | problems, gaps in knowledge and reasons for a new theory being needed | similar to conventional articles |



| | | |
|---|---|---|
| 3. research methodology | description of design science/constructive research approach. | see<br>[March and Smith, 1995]<br>[Hevner, 2003]<br>[Gregor and Jones, 2007] |

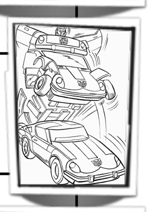
| | | |
|---|---|---|
| 4. Justificatory knowledge | theory from IS and natural and social science that informs the design theory. | ISDT component (6) |



the placing of this component is debatable. It could be in Section 2 or 5.

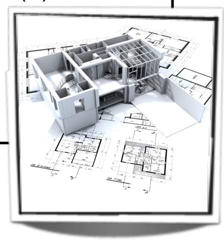
from Gregor & Jones, 2008 *The anatomy of a design science paper: a research note*

| | | |
|---|---|---|
| 5. Specification of the designed artifact | meta-requirements - the purpose of the class of artifacts addressed by the theory | ISDT component (1) |
| | process by which the designer arrived at their solution | optional; might assist in demonstrating credibility |
| | principles of form and function | ISDT component (3) |
| | consideration of artifact mutability | ISDT component (4) |
| | principles of implementation | ISDT component (7) |
| | testable design propositions | ISDT component (5) |




from Gregor & Jones, 2008 *The anatomy of a design science paper: a research note*

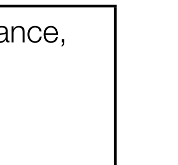
| | | |
|---|---|---|
| 6. instantiation | description of any working system or method in use | |
| | ISDT component (8) | ISDT component (8) |



from Gregor & Jones, 2008 *The anatomy of a design science paper: a research note*

| | | |
|---|---|---|
| 7. evaluation | describe tests of systems/ method in use | could show performance, functionality, user acceptance ... |
| | evaluate against criteria for design science/constructive research | |

| | | |
|---|---|---|
| | testable design propositions | ISDT component (5) |




from Gregor & Jones, 2008 *The anatomy of a design science paper: a research note*

| | | |
|---|---|---|
| 8. discussion and conclusions | summarize work and findings, discuss limitations, establish significance | similar to conventional articles |



| | | |
|---|---|---|
| 9. references | | |

from Gregor & Jones, 2008 *The anatomy of a design science paper: a research note*

idealized structure
## example


*Information Systems Research*
Vol. 15, No. 1, March 2004, pp. 3–21
ISSN 1047-7047 | EISSN 1526-5536 | 04 | 1501 | 0003
DOI 10.1287/isre.1040.0012


# A Fault Threshold Policy to Manage Software Development Projects

I. Robert Chiang
School of Business, University of Connecticut, Storrs, Connecticut 06084, robert.chiang@business.uconn.edu
Vijay S. Mookerjee
School of Management, University of Texas at Dallas, Richardson, Texas 75083, vijaym@utdallas.edu

This paper presents a project management policy in which the appearance of software faults during system construction is used to determine the timing of system integration activities (e.g., team meetings, analyzing modules for interface inconsistencies, system fault correction, and so on). System integration is performed only if a threshold fault count has been exceeded; otherwise, module development is allowed to continue. We derive an expression for calculating fault thresholds and analyze the policy to reveal the presence of three operating regions: (1) a region in which development should continue with no system integration, (2) a region in which system integration occurs if a threshold fault count has been exceeded, and (3) a region in which system integration should always take place. Analytical and numerical results demonstrate how the fault thresholds change with system complexity, team skill, development environment, and project schedule. We also show how learning that occurs during each round of system integration leads to less frequent integration in the future, and lower total construction effort. Simulation experiments reveal that the fault threshold policy can be applied even if several homogeneity assumptions in the model are relaxed, allowing for differences in the propensity among modules to accumulate faults and the effort needed to correct these faults. Finally, the fault threshold policy outperforms a fixed-release policy in which system integration occurs whenever a fixed number of modules has been released.

*Key words*: software project management; quality-driven integration policy; incremental development; team coordination
*History*: Sandra Slaughter, Associate Editor. This paper was received on November 12, 2001, and was with the authors 8 months for 2 revisions.

Example design article structure (Chang and Mookeerjee, 2004)

| | Type | Component examples |
|---|---|---|
| (1) | Purpose and scope | The aim is to develop a fault threshold policy to determine when system integration occurs during a process of incremental systems development. The policy is developed for homogeneous systems, where modules are similar in size and complexity and all faults take roughly the same effort to fix. The policy is appropriate for systems that can be tested frequently and at relatively low cost. The policy is designed to consider a number of project parameters (such as complexity). |
| (2) | Constructs | Examples are: incremental development, system integration, fault threshold, testing, faults detected. |
| (3) | Principles of form and function | The policy uses a derived expression to give dynamic guidelines for when system integration should occur, with (1) a region of no integration, (2) a region where integration occurs depending on a fault count, and (3) a region in which systems integration should always take place. |
| (4) | Artifact mutability | The designers consider the effects of team learning that occur over multiple construction cycles and show how the policy will vary over a number of cycles. |
| (5) | Testable propositions | Predictions about outcomes are provided that are tested in simulation experiments. |
| (6) | Justificatory knowledge | Theory is offered relating to group coordination processes, team cognition, software development productivity, and fault growth models. |
| (7) | Principles of implementation | Not a great deal of detail is given on how to build a concrete version of this abstract policy in specific projects. An example is given where the formulae in the policy are applied to an imaginary scenario. It is stated that it might be necessary to build some randomness into the model in a real-life project and this is left for further work. |
| (8) | Expository instantiation | Examples of the policy in action are provided through simulations. |

Table 4: Components of a design theory for a software threshold fault theory

from [Chiang and Mookerjee, 2004]

# 1. introduction (3 pages)

- significance of incremental development
- claim existing research does not give much guidance on how to manage incremental development (relevance > gap identified)
- claim the paper advances a “fault threshold policy for incremental improvement” *(purpose and scope)*
- explanation of terms *(constructs),* and
  overview of the policy problem, with diagram

from [Chiang and Mookerjee, 2004]

## 2. literature review (2 pages)

- on work on coordination and team integration, and on improving software team productivity
- concluding that methods to manage software faults in an evolving system have not yet been prescribed (gap identified)
- MISSING > methods section

from [Chiang and Mookerjee, 2004]

## 3. fault threshold policy (5 pages)

- an analytic description of the software fault threshold policy, and a detailed description of the way the policy operates *(principles of form and function)*
- an example is provided *(an hypothetical instantiation)*
- MISSING > principles of implementation

from [Chiang and Mookerjee, 2004]

# 4. learning effects (3 pages)

- investigation of the behavior of the policy in the context of multiple cycles, as learning benefits occur *(artifact mutability)*

- a comparison of the new policy with another method ...

- question: right place?

    ✓ whether this sub-section might have been placed differently, in the evaluation section

from [Chiang and Mookerjee, 2004]

# 5. simulation experiments (3 pages)

- an evaluation of the new method, with three purposes:

    ✓ that the new method/model makes accurate predictions

    ✓ that it is robust with respect to underlying assumptions

    ✓ that it is applicable to heterogeneous systems (generalizability)

- an overall assessment of the policy

- future directions for research

from [Chiang and Mookerjee, 2004]

## 6. summary and conclusion (1 page)

- summary of the paper
- speculation that most organizations would benefit from frequent system testing using the model, as it has economic benefits *(a testable hypothesis)*

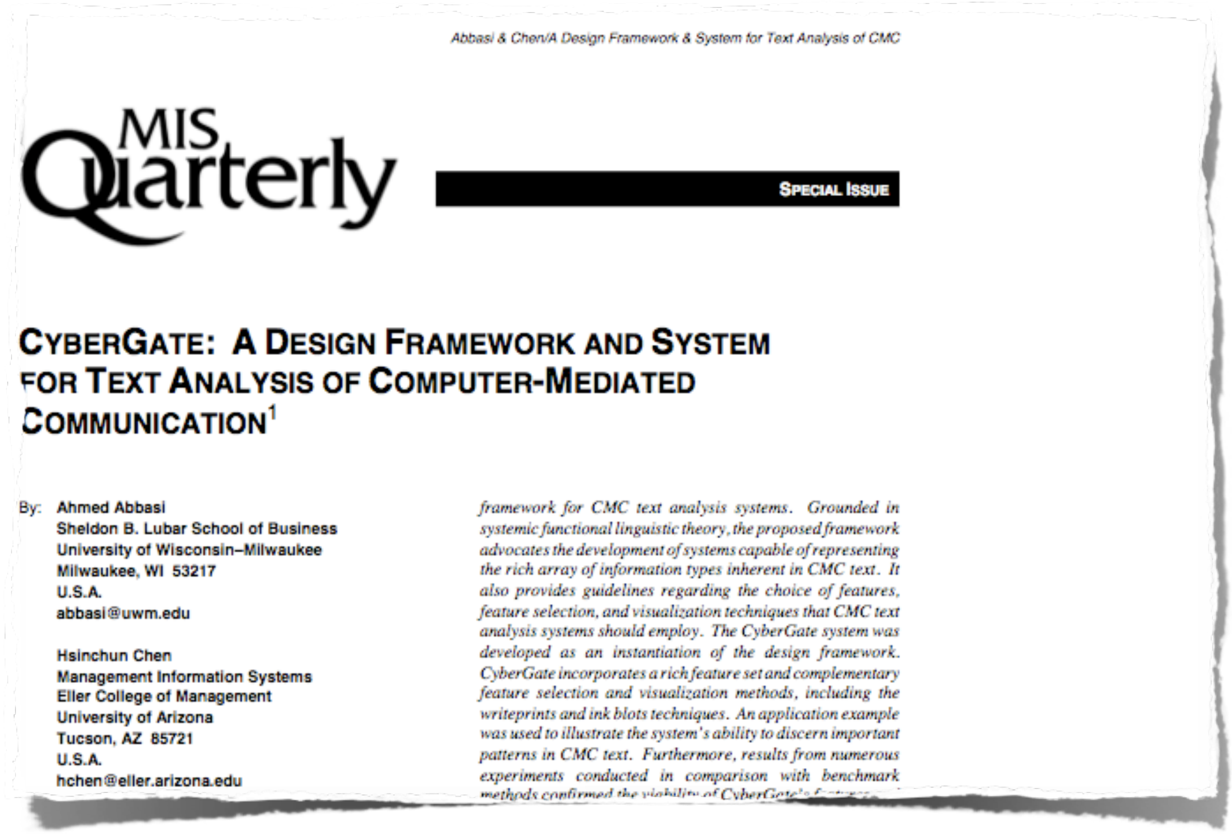
from [Chiang and Mookerjee, 2004]

idealized structure

example

example of a “IS design theory” paper

recent



Abbasi & Chen/A Design Framework & System for Text Analysis of CMC

MIS Quarterly

SPECIAL ISSUE

CYBERGATE: A DESIGN FRAMEWORK AND SYSTEM FOR TEXT ANALYSIS OF COMPUTER-MEDIATED COMMUNICATION[1]

By: Ahmed Abbasi
Sheldon B. Lubar School of Business
University of Wisconsin–Milwaukee
Milwaukee, WI 53217
U.S.A.
abbasi@uwm.edu

Hsinchun Chen
Management Information Systems
Eller College of Management
University of Arizona
Tucson, AZ 85721
U.S.A.
hchen@eller.arizona.edu

*framework for CMC text analysis systems. Grounded in systemic functional linguistic theory, the proposed framework advocates the development of systems capable of representing the rich array of information types inherent in CMC text. It also provides guidelines regarding the choice of features, feature selection, and visualization techniques that CMC text analysis systems should employ. The CyberGate system was developed as an instantiation of the design framework. CyberGate incorporates a rich feature set and complementary feature selection and visualization methods, including the writeprints and ink blots techniques. An application example was used to illustrate the system's ability to discern important patterns in CMC text. Furthermore, results from numerous experiments conducted in comparison with benchmark methods confirmed the viability of CyberGate's ...*

Example of an "IS design theory" article (Abbasi and Chen, 2009))

# anatomy of a paper

| |
|---|
| 1. introduction |
| 2. literature review background |
| 3. research methodology |
| 4. justificatory knowledge |
| 5. specification of the designed artifact |
| 6. instantiation |
| 7. evaluation |
| 8. discussion and conclusions |
| 9. references |

next

# A design science research methodology for information systems research

Ken Peffers, Tuture Tuunanen, Marcus Rothenberger and Samir Chatterjee
*Journal of the Management Information Systems, 2008, 24(3): 45-77*

- see also [Vaishnavi and Kuechler, 2007] *Design science research methods and patterns* ch. 2-4

next

# Action Design Research

Maug Sein, Ola Henfridsson, Sandeep Purao, Matti Rossi & Rikard Lindgren
MIS Quarterly, 2010, 35(1): 37–56.