Why are you doing this study?

Theory in software engineering research

Klaas-Jan Stol
University College Cork, Ireland
Lero, the Research Ireland Centre for Software
SINTEF Digital, Norway

Or:

On limão and limão siciliano in Software Engineering Research

Klaas-Jan Stol
University College Cork, Ireland
Lero, the Research Ireland Centre for Software
SINTEF Digital, Norway

Or:

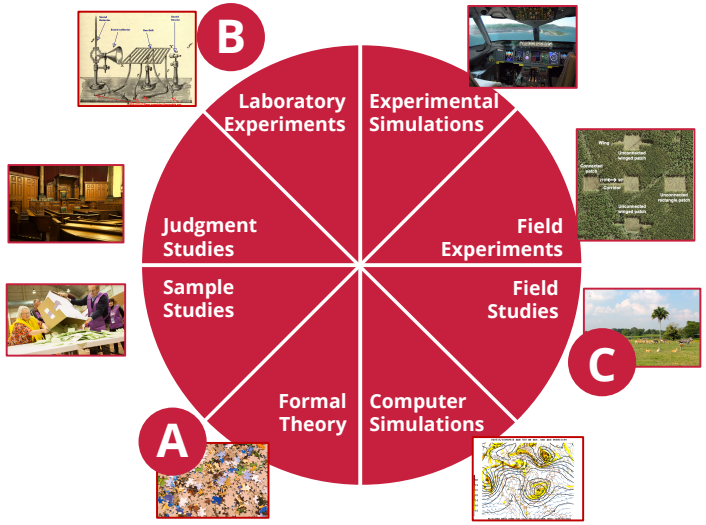
A RISC Instruction Set for Theorizing in Software Engineering Research

Klaas-Jan Stol
University College Cork, Ireland
Lero, the Research Ireland Centre for Software
SINTEF Digital, Norway

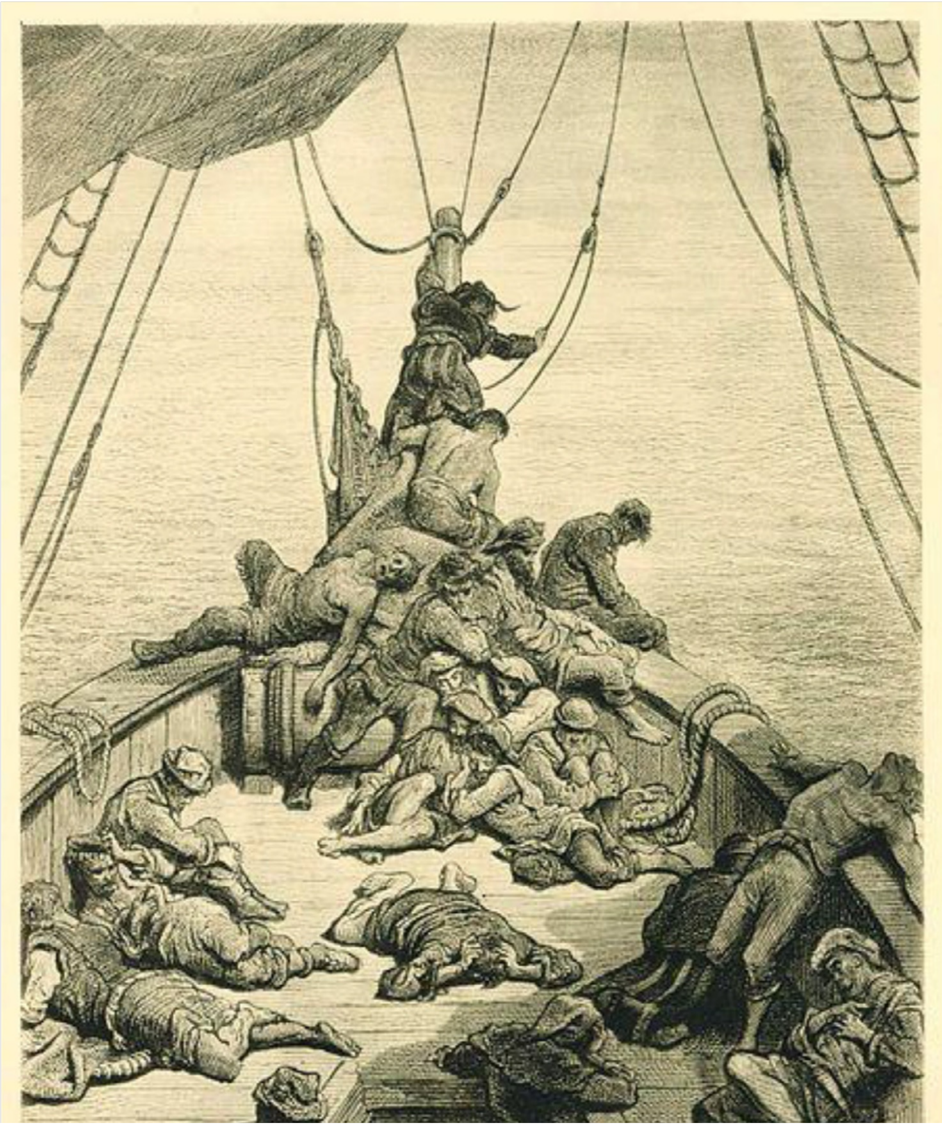
THESIS

We don’t recognize the theoretical trees for our empirical forest.

PROMISE

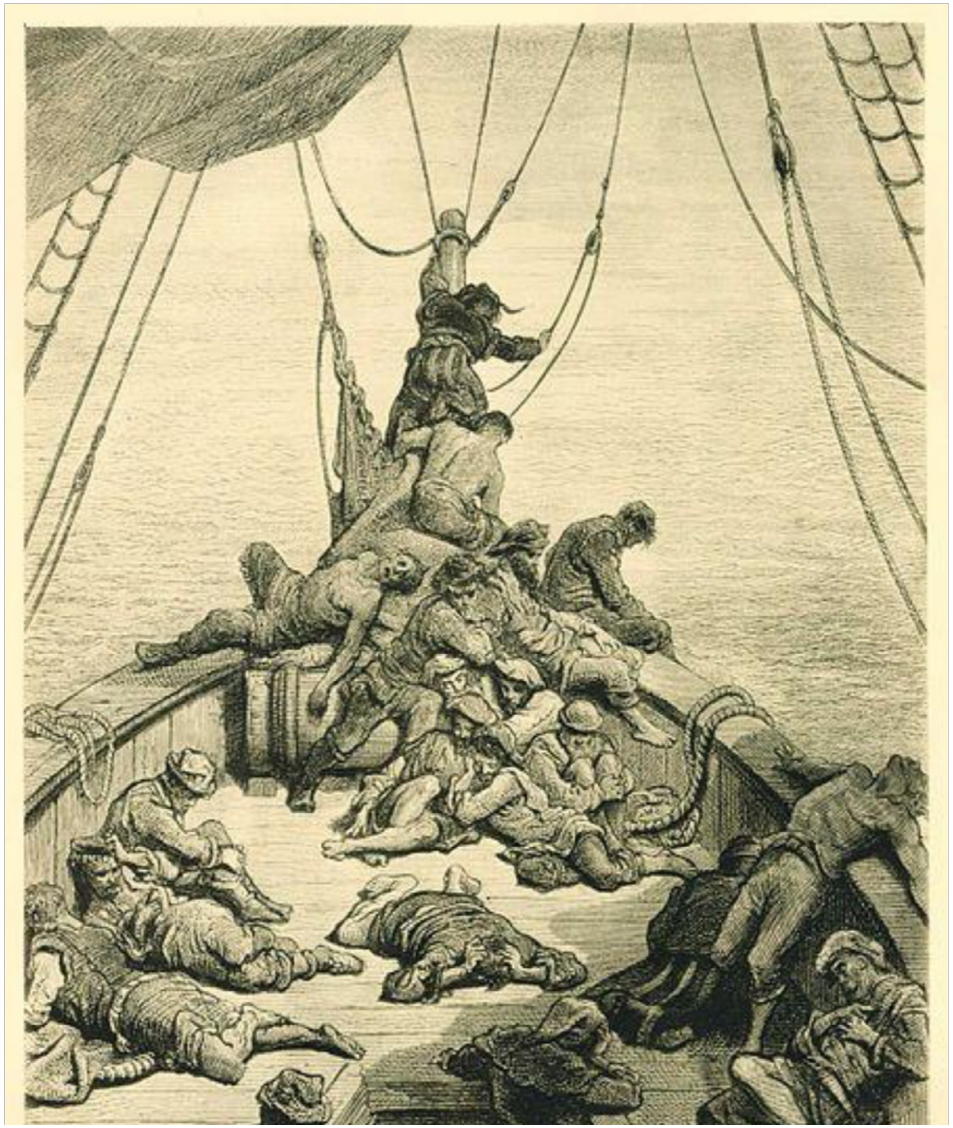


Name the things we do to theorize, so we can recognize them.



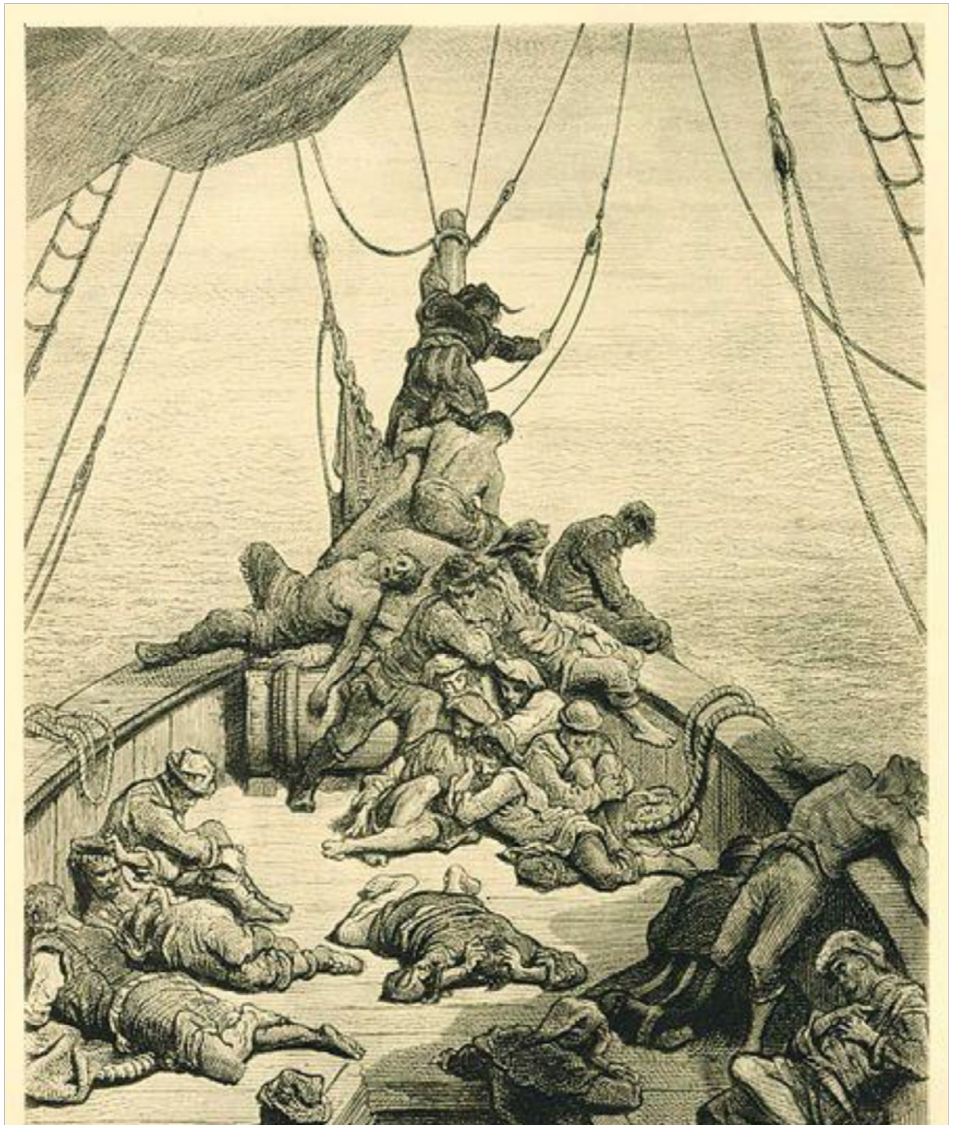
❑ Drunk Sailors

Image source: Nashawaty, BS Motassem, et al. "Nutrition and art: 19th century images of scurvy, chlorosis, and pellagra." *Clinics in Dermatology* 39.5 (2021): 858-863.



- ❑ Drunk Sailors
- ❑ ICSE PC members after reviewing 2,500 papers

Image source: Nashawaty, BS Motassem, et al. "Nutrition and art: 19th century images of scurvy, chlorosis, and pellagra." *Clinics in Dermatology* 39.5 (2021): 858-863.



- ❑ Drunk Sailors
- ❑ ICSE PC members after reviewing 2,500 papers
- ❑ Sailors with scurvy

Image source: Nashawaty, BS Motassem, et al. "Nutrition and art: 19th century images of scurvy, chlorosis, and pellagra." *Clinics in Dermatology* 39.5 (2021): 858-863.

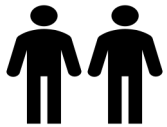
Scurvy Research

(following the traditions of Software Engineering Research...)

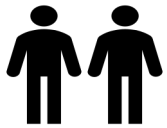
How Do Sailors Cope with Scurvy? A Qualitative Study

Mining Ship Logs to Understand Scurvy Risk Factors

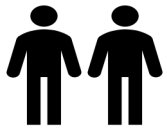
A Systematic Literature Review of Scurvy Prevention Techniques



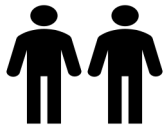
Quart cider



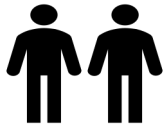
25 drops of elixir of vitriol



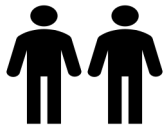
2 spoonfuls of vinegar



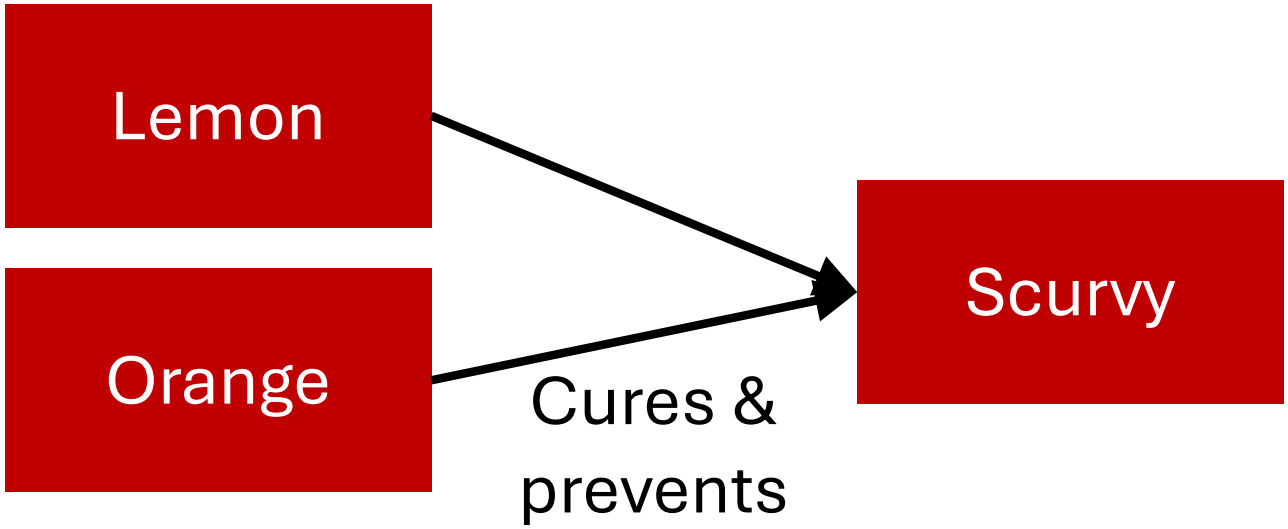
½ pint of sea water



Nutmeg-size purgative electuary

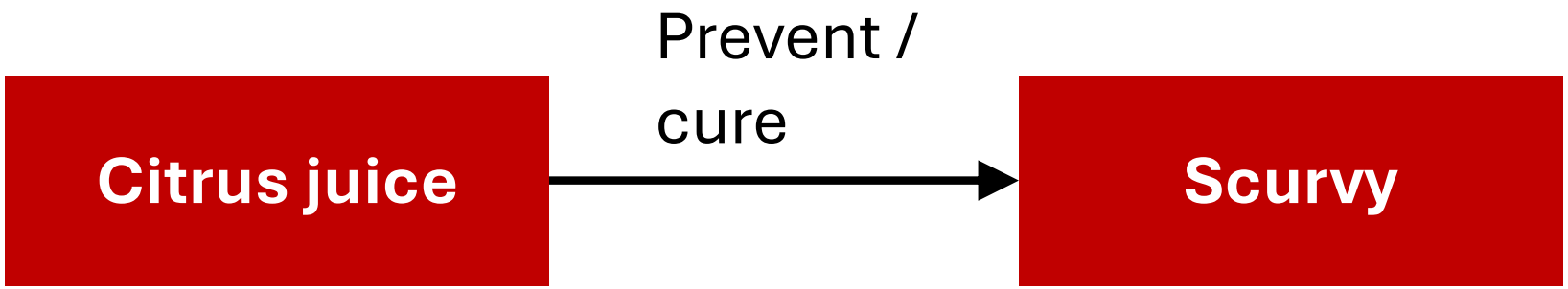


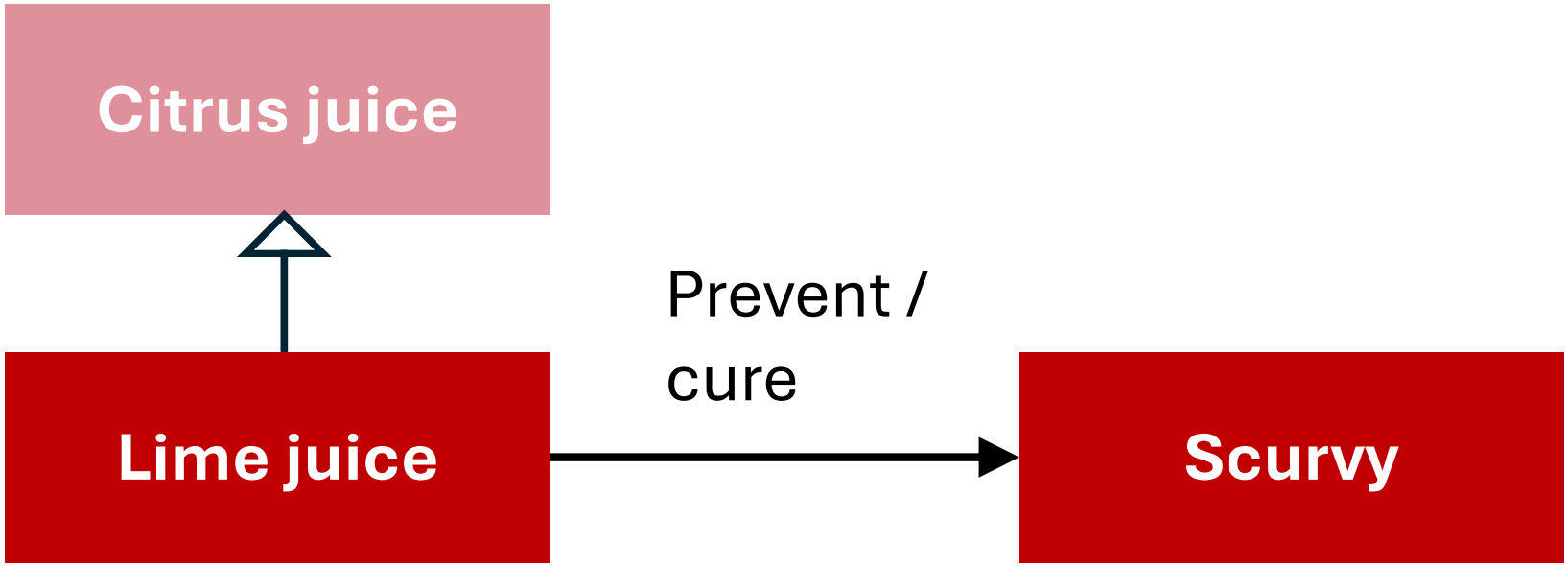
2 oranges + 1 lemon

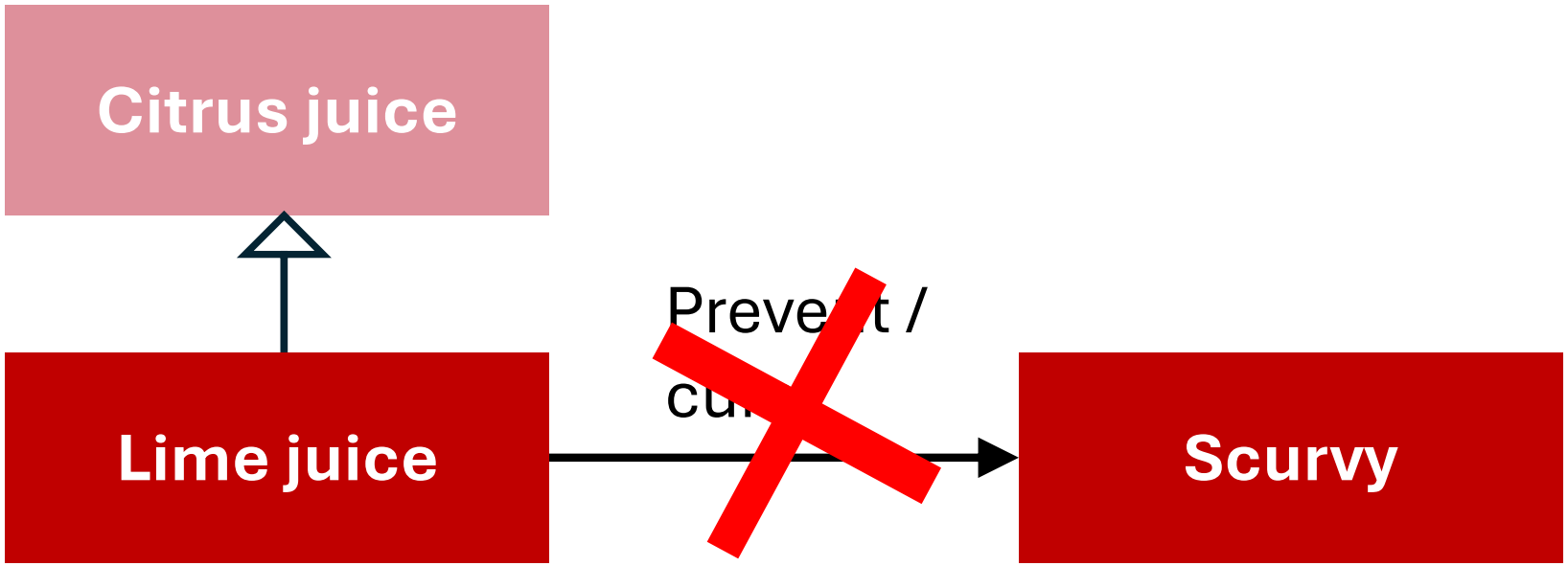


Aha!

It must be the citrus!







Oops!

Not the juice, but Vit C

Hands up:

Who is actively theorizing in their research?

Common findings in SE studies:

“Management support is important.”

“Communication matters.”

“Stakeholder involvement improves outcomes.”

“Resistance to change can hinder implementation.”

“Tools must fit existing workflows.”

(AI anybody …?)

“Context matters.”

Mature research fields:

- **Cumulative**
- **Mechanisms**
- **Explanations**

Software Engineering:
limited cumulative tradition +
phenomenon-driven

SE lacks focus on theory building.

But:

We DO theorize – we just don’t know it very well.

How does one

Theorize

Nik Hassan et al.: useful products of theorizing

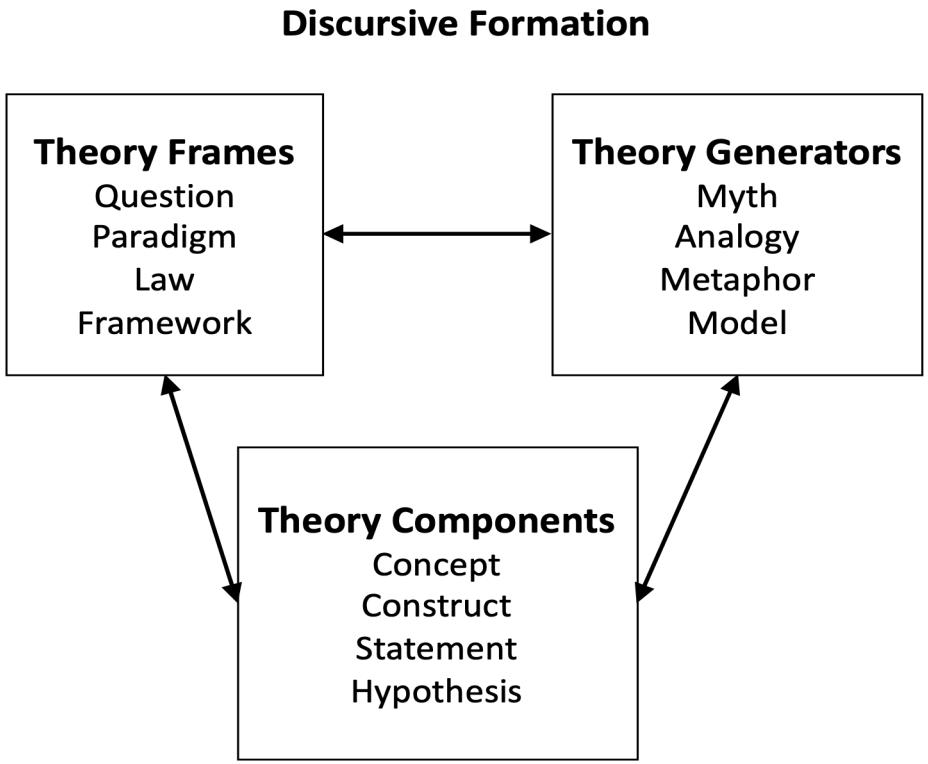


Image source: Hassan et al. "Products of theorizing—Towards native theories of emerging information technologies." *Journal of Information Technology* 38.4 (2023): 372-381.

These are the nouns.

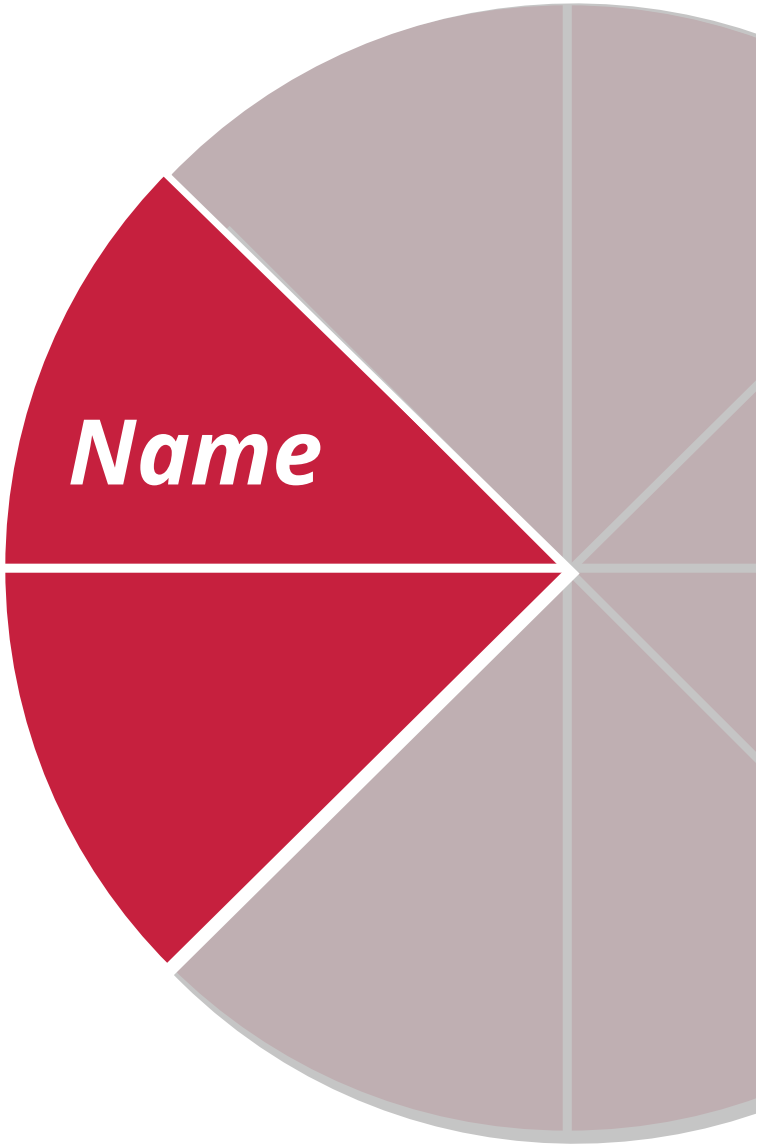
What we need are the verbs: intellectual operations

Name

Johannsen 1909

Introduced the word ‘gene’

Created a stable object of inquiry in the emerging field of genetics



Source:
Nils Roll-Hansen: The holist tradition in twentieth century genetics. Wilhelm Johannsen’s genotype concept. J Physiol 592. 11 (2014)

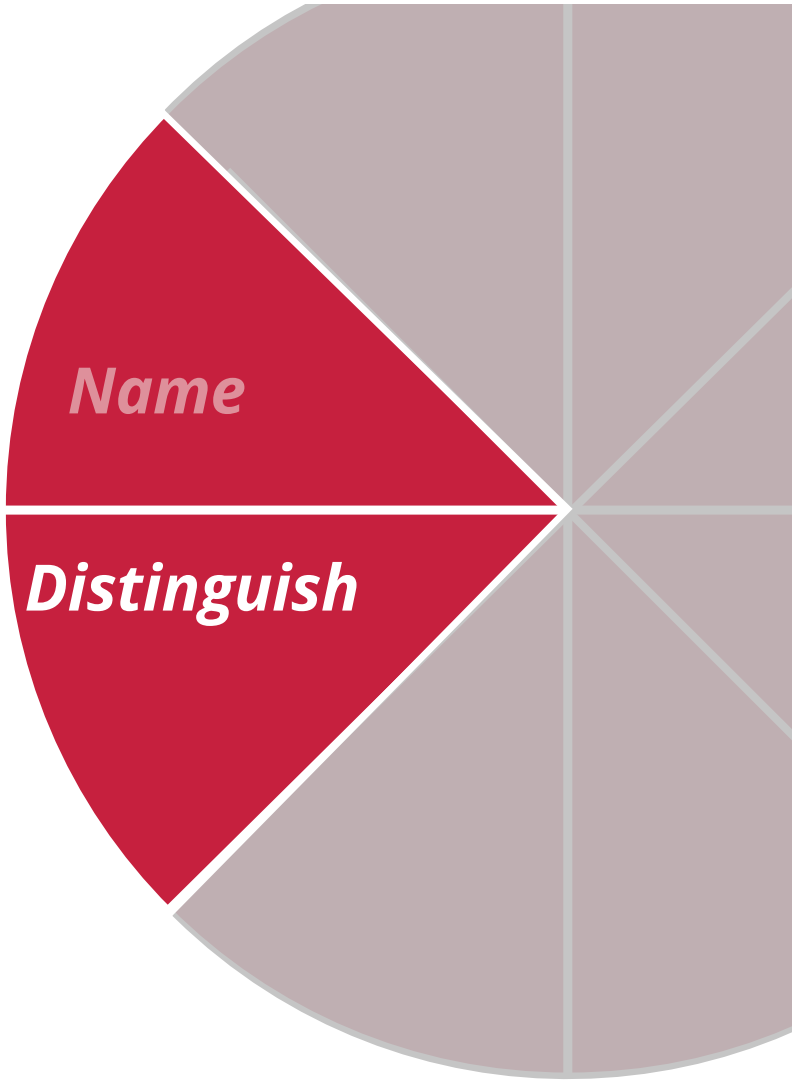
Distinguish

Robert Koch ca. 1880

Infection ≠ disease

Infection means presence of micro-organism

Does not imply **disease**

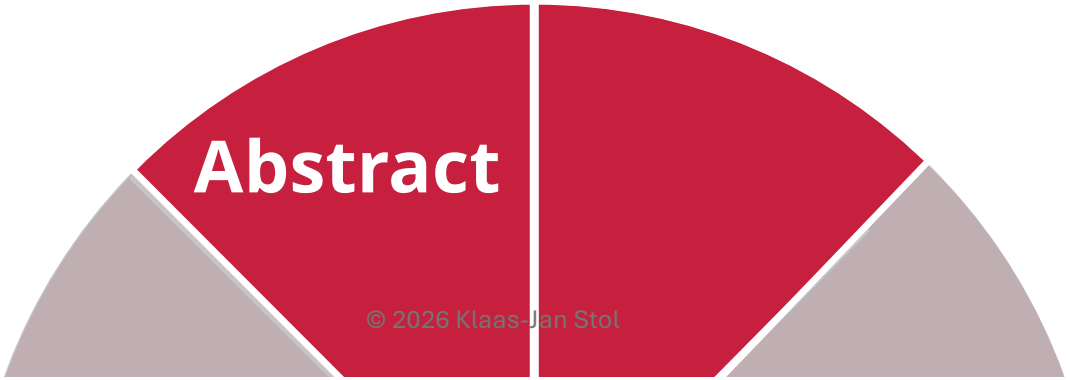


Abstract

Isaac Newton 1687

- **Apple falling to the earth**
- **moon orbiting earthy**

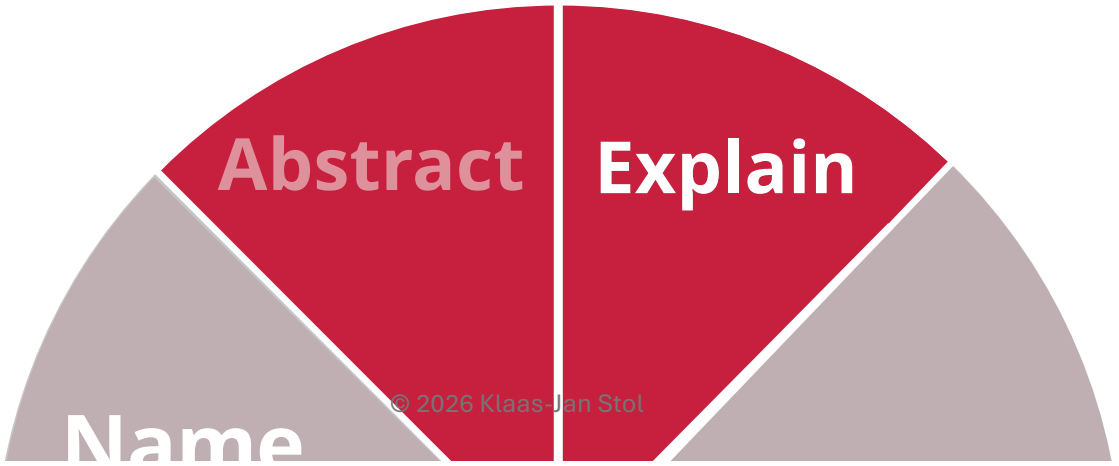
… are instances of the same phenomenon.

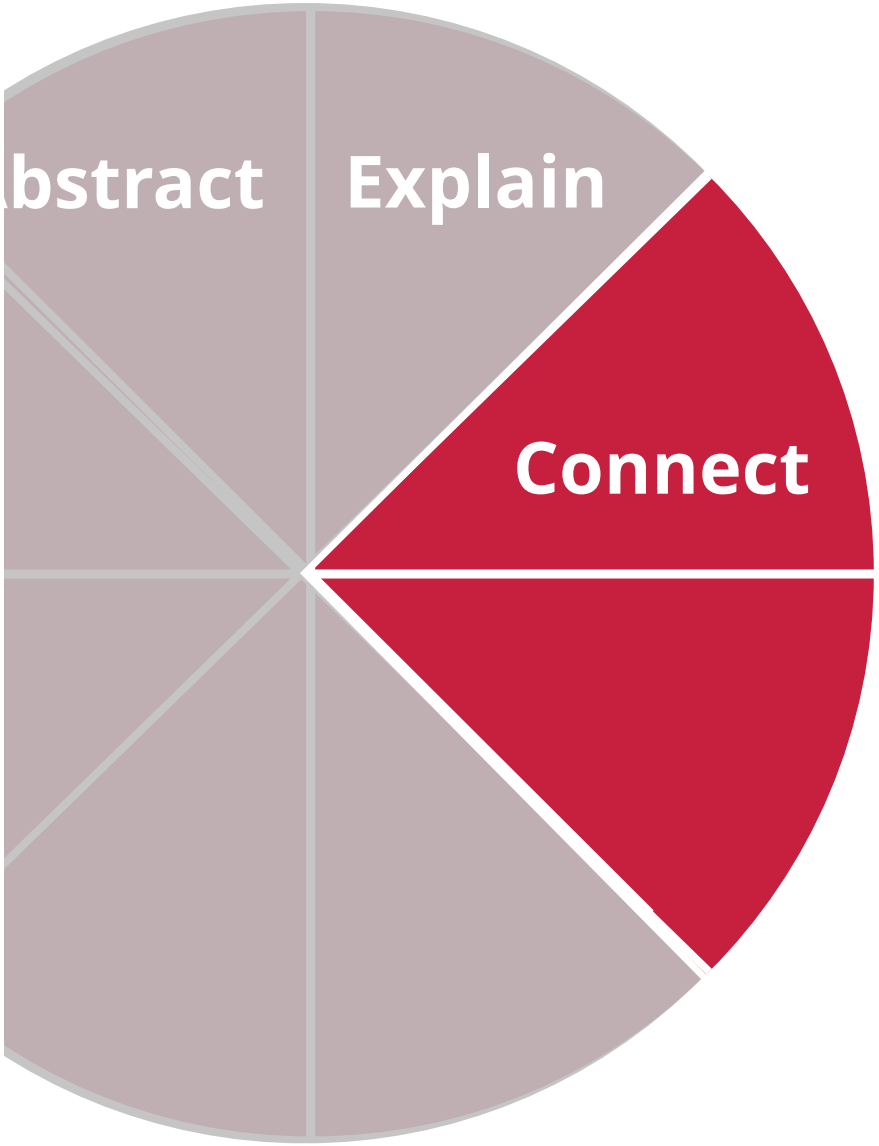


Explain: Scurvy 1747 - 1932

Vitamin C as explanation proposed in 1932

Until then: citrus **only associated** with curing scurvy, **without knowing how/why**



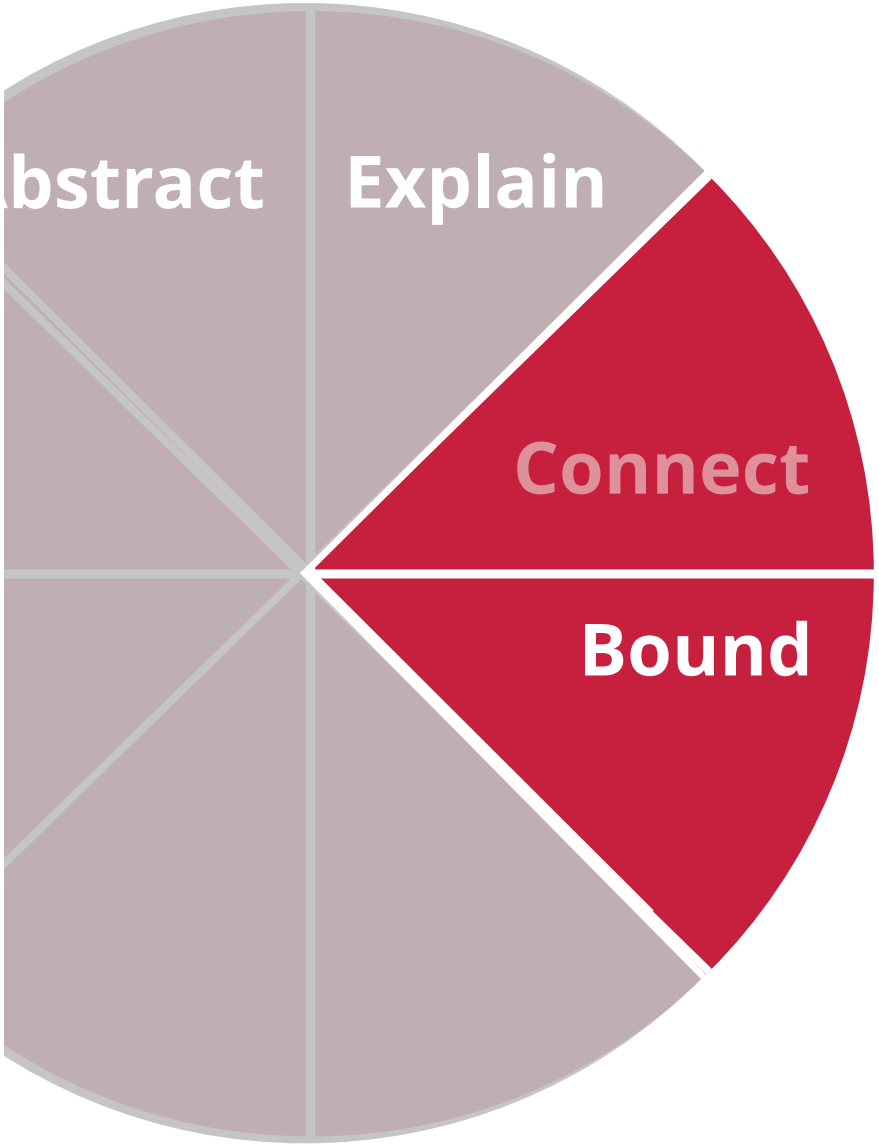


Connect

George Boole 1847:
Boole Algebra

Claude Shannon 1937:

Boole Algebra to represent electrical circuits



Bound:

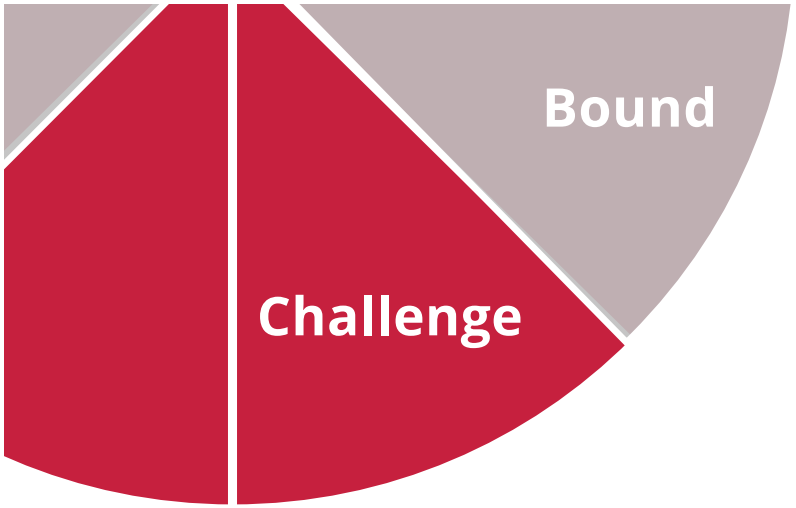
Isaac Newton proposed same mechanics explain terrestrial and celestial motion

But:

Breaks down under conditions:

- **Very high speed**
- **Very strong gravitational fields**

Newton wasn't entirely wrong: valid within boundaries



Challenge

Rutherford 1911

Assumption: Atom is a soft blob

Experiment:
Most particles went through, some bounced

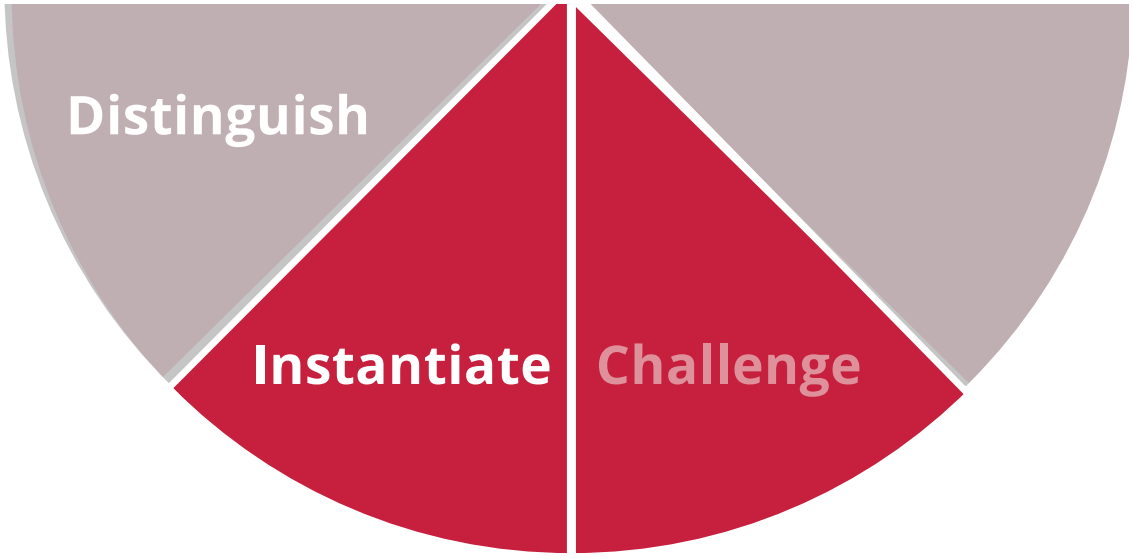
Problem:
A soft blob should not make particles bounce

New idea:
Atom has a tiny, hard nucleus

Instantiate

Mendel 1850-60s:

- **Proposed general inheritance rules based on pea plants**



Garrod 1902:

- **What does Mendel's theory look like for human disease?**

“But Software Engineering is different…it’s ENGINEERING!”

to Name: code-anomaly agglomeration

Before:

Smells studied separately

After:

New unit of analysis: clusters

Oizumi, Garcia, Colanzi, Ferreira, von Staa. When Code-Anomaly Agglomerations Represent Architectural Problems. SBES 2014.

to Name: code-anomaly agglomeration

Before:

Smells studied separately

After:

New unit of analysis: clusters

New questions:

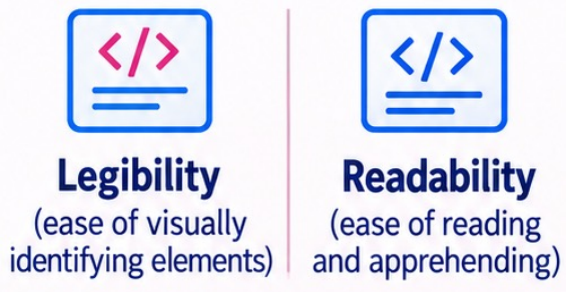
- Which clusters matter?
- Which indicate architectural problems?
- Which should be prioritized?

Oizumi, Garcia, Colanzi, Ferreira, von Staa. When Code-Anomaly Agglomerations Represent Architectural Problems. SBES 2014.

to Distinguish: legible vs. readable code

Before:

After:

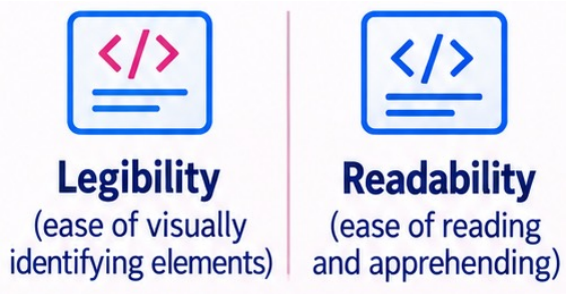


Oliveira, Bruno, Madeiral, Castor. *Evaluating Code Readability and Legibility: An Examination of Human-centric Studies*. ICSME 2020

to Distinguish: legible vs. readable code

Before:

After:



New questions:

- Are we measuring the same thing?
- What affects legibility?
- What affects readability?
- Predict same or different outcomes?

Oliveira, Bruno, Madeiral, Castor. *Evaluating Code Readability and Legibility: An Examination of Human-centric Studies*. ICSME 2020

to Abstract: software ecosystems

Before:

Open Source
How do OSS communities work?

Product Lines & Platforms
How do we manage variability?

Software supply networks
How do firms manage dependencies?

Software business
How do firms create and capture value?

Apache case

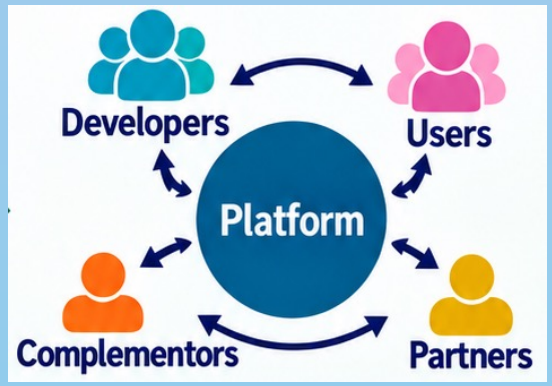
Software product lines

Supply Network modelling

Software business

After:

Software Ecosystems



to Explain: *a theory of org structures*

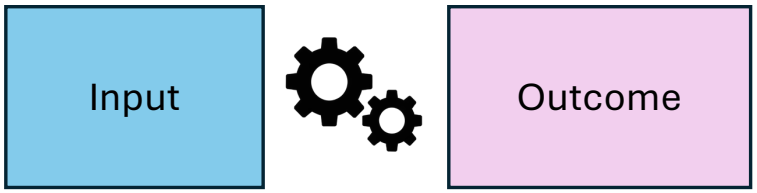
Before:

Orgs adopt different development & infrastructure structures.

Why?

After:

… a theory to explain organizational structures for development and infrastructure professionals



Leite, Lago, Melo, Kon, Meirelles. A Theory of Organizational Structures for Development and Infrastructure Professionals. IEEE Transactions on Software Engineering 2023

to Explain: *a theory of org structures*

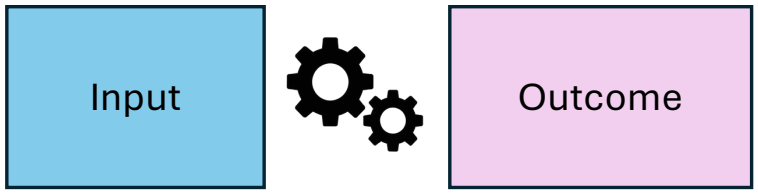
Before:

Orgs adopt different development & infrastructure structures.

Why?

After:

… a theory to explain organizational structures for development and infrastructure professionals



New questions:

- Do proposed relationships hold elsewhere?
- How do different org structures perform?
- How do different org structures affect productivity, experience, …

Leite, Lago, Melo, Kon, Meirelles. A Theory of Organizational Structures for Development and Infrastructure Professionals. IEEE Transactions on Software Engineering 2023

to Connect: control theory & self-adapting systems

Before:

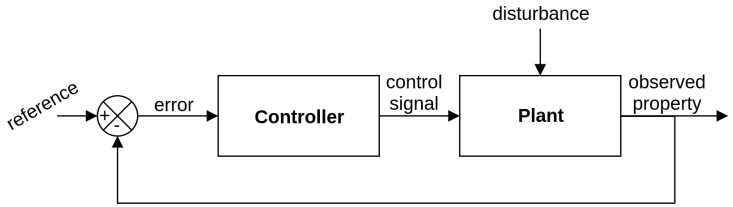
Self-adaptation:
- Monitor
- React

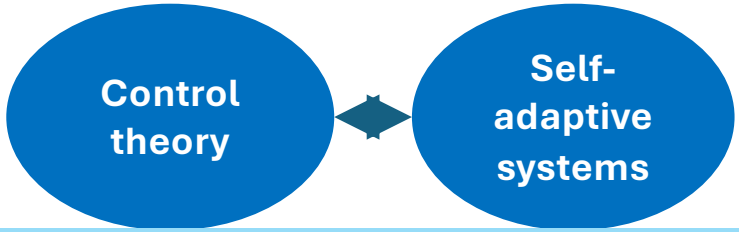
Concerns:
- Predictability
- Robustness
- Evaluation

After:

Use control-theoretic reasoning:

- Setpoint / reference
- Error
- Control action





Caldas, Rodrigues, Gil, Rodrigues, Vogel, Pelliccione. A Hybrid approach combining control theory and AI for engineering self-adaptive systems. SEAMS 2020

to Connect: control theory & self-adapting systems

Before:

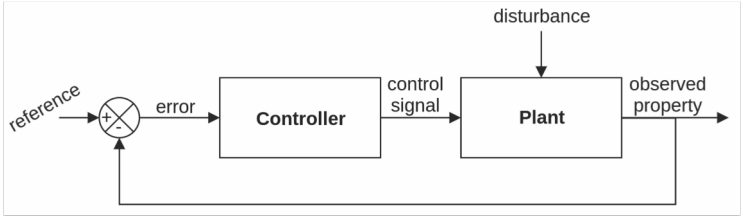
Self-adaptation:
- Monitor
- React

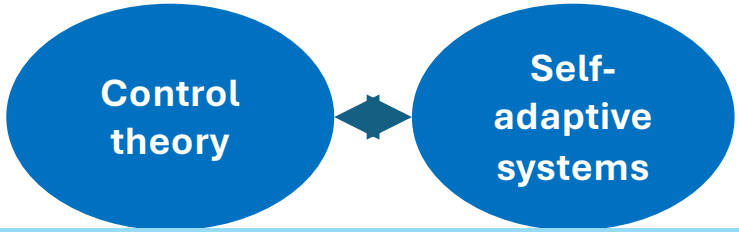
Concerns:
- Predictability
- Robustness
- Evaluation

After:

Use control-theoretic reasoning:

- Setpoint / reference
- Error
- Control action

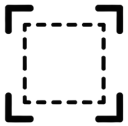




New questions:

- What software goals can be setpoint?
- Multiple competing adaptation goals?
- Stability under uncertainty?
- Learn/change at runtime?

Caldas, Rodrigues, Gil, Rodrigues, Vogel, Pelliccione. A Hybrid approach combining control theory and AI for engineering self-adaptive systems. SEAMS 2020



to Bound: AOP for exception handling

Before:

Aspect Oriented Programming

Exception handling is cross-cutting

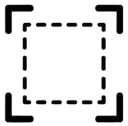
AOP is good for exception handling?

After:

IT DEPENDS

- Type of handler
- Context
- Reuse opportunities
- …

Castor et al. On the Modularization and Reuse of Exception Handling with Aspects," *Software: Practice and Experience. 2009*



to Bound: AOP for exception handling

Before:

Aspect Oriented Programming

Exception handling is cross-cutting

AOP is good for exception handling?

After:

IT DEPENDS

- Type of handler
- Context
- Reuse opportunities
- …

New questions:

- When does aspectization help?
- When harmful or ineffective?
- Trade-offs with other concerns?

Castor et al. On the Modularization and Reuse of Exception Handling with Aspects," *Software: Practice and Experience. 2009*

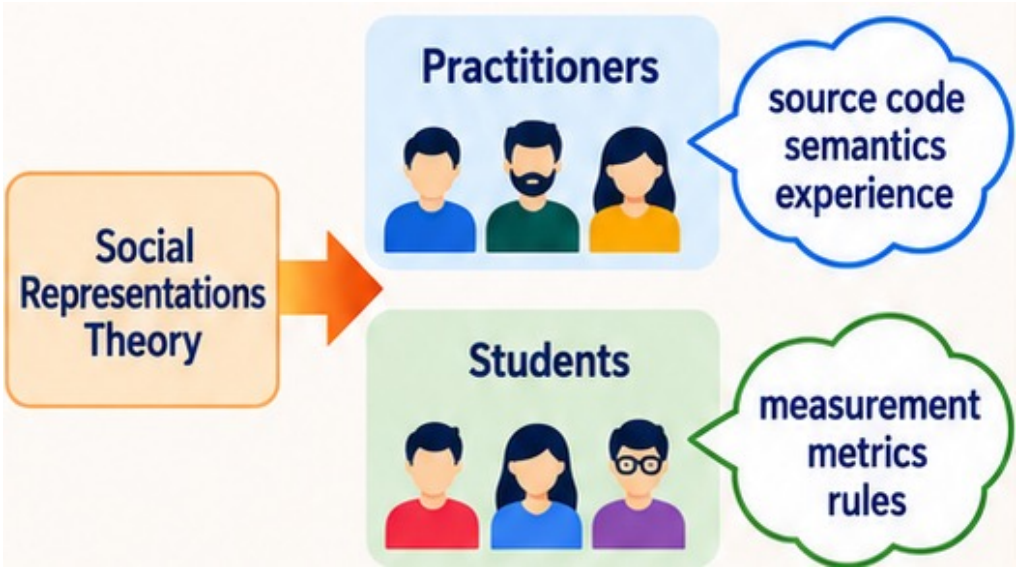
to Instantiate: Social representations theory

Before:

Artifact-focused

Individual judgment

After:



De Mello, Uchoa, Oliveira, Oliveira, Fonseca, Garcia, De Mello. Investigating the Social Representations of Code Smell Identification: A Preliminary Study. CHASE 2019

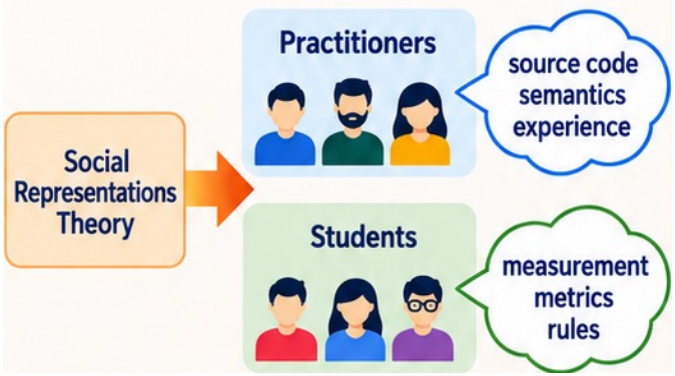
to Instantiate: Social representations theory

Before:

Artifact-focused

Individual judgment

After:



New questions:

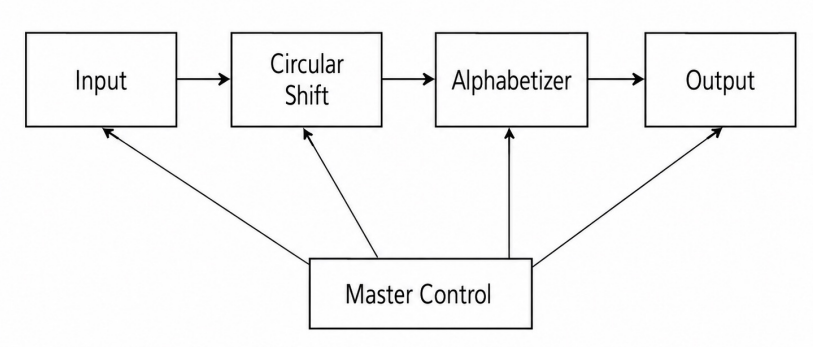
Research focused on measurable or structural smells.

Revisit research to focus on semantic smells

De Mello, Uchoa, Oliveira, Oliveira, Fonseca, Garcia, De Mello. Investigating the Social Representations of Code Smell Identification: A Preliminary Study. CHASE 2019

to Challenge: system decomposition

Before:



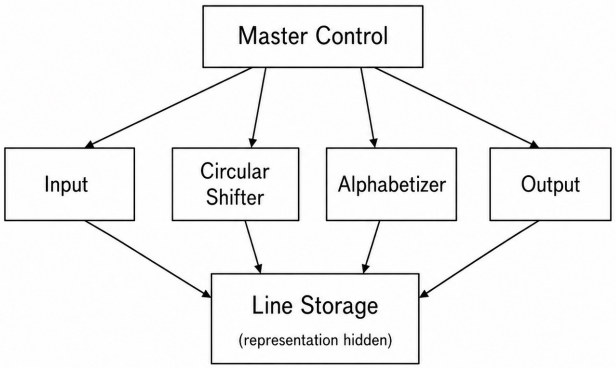
Decomposition based on flow of processing steps

Flowchart → decomposition

After:

Information hiding

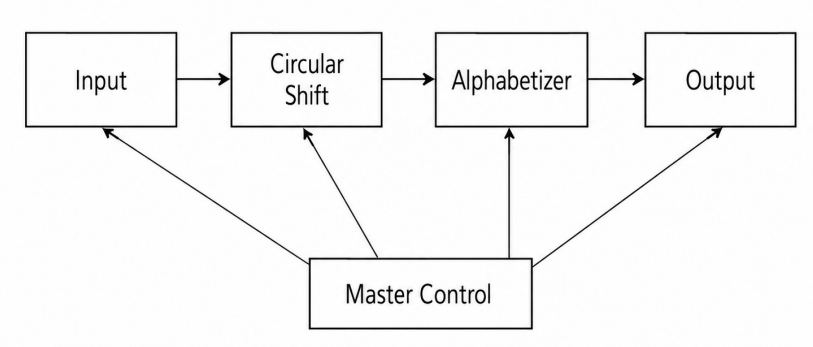
What design decisions should a module hide?



Parnas. On the Criteria To Be Used in Decomposing Systems into Modules. CACM 1972

to Challenge: system decomposition

Before:



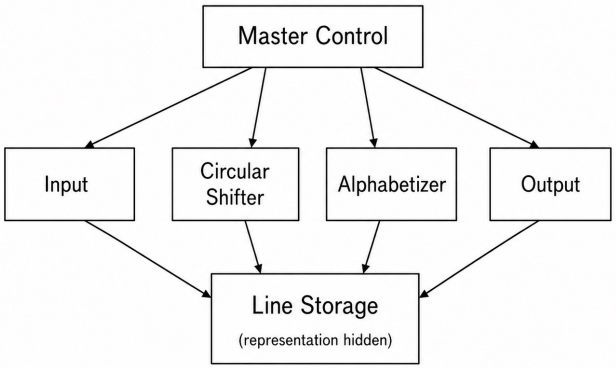
Decomposition based on flow of processing steps

Flowchart → decomposition

After:

Information hiding

What design decisions should a module hide?



New research:

Abstract data types, encapsulation

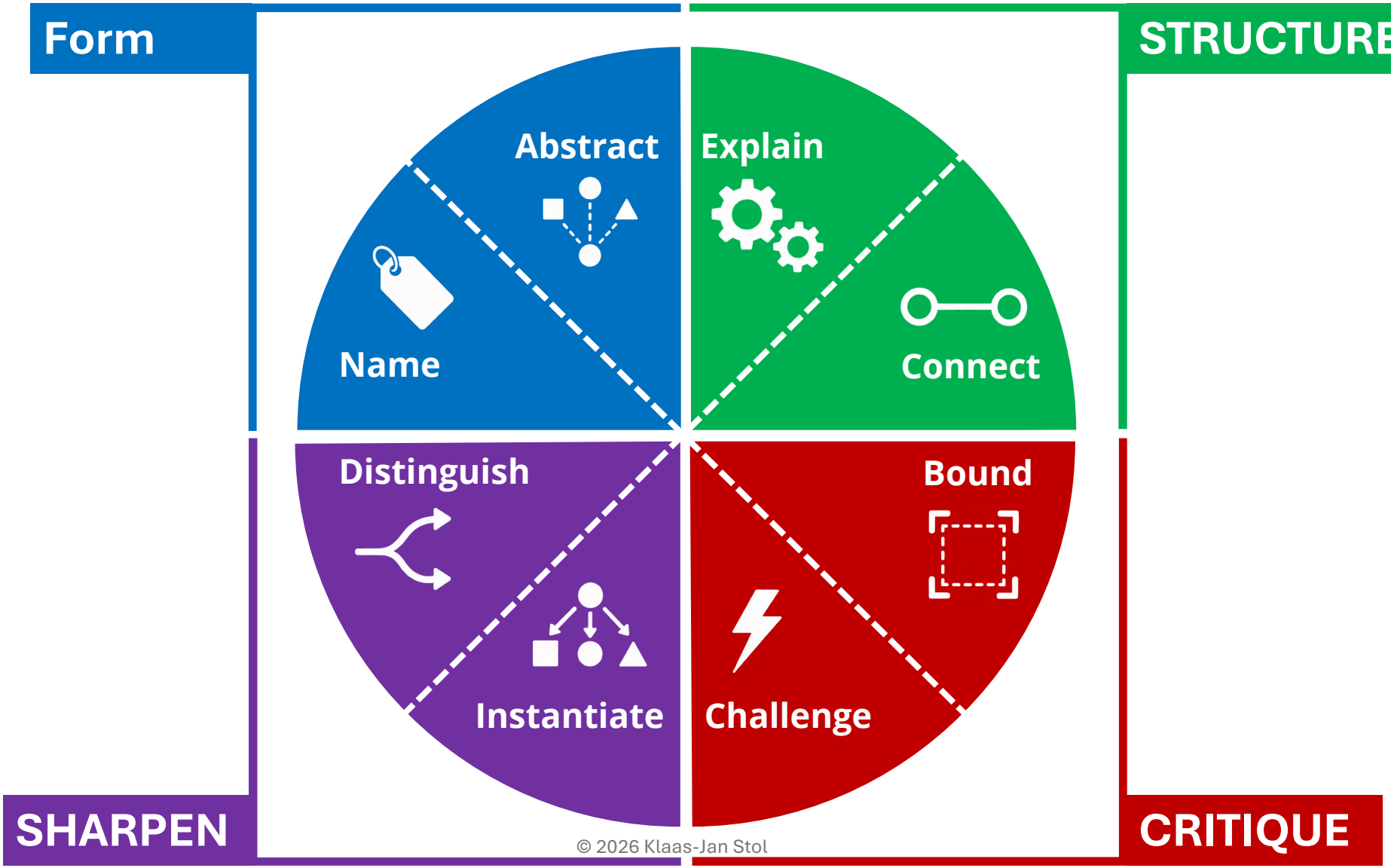
Object Orientation

Design

Software architecture

Product families (lines)

Parnas. On the Criteria To Be Used in Decomposing Systems into Modules. CACM 1972

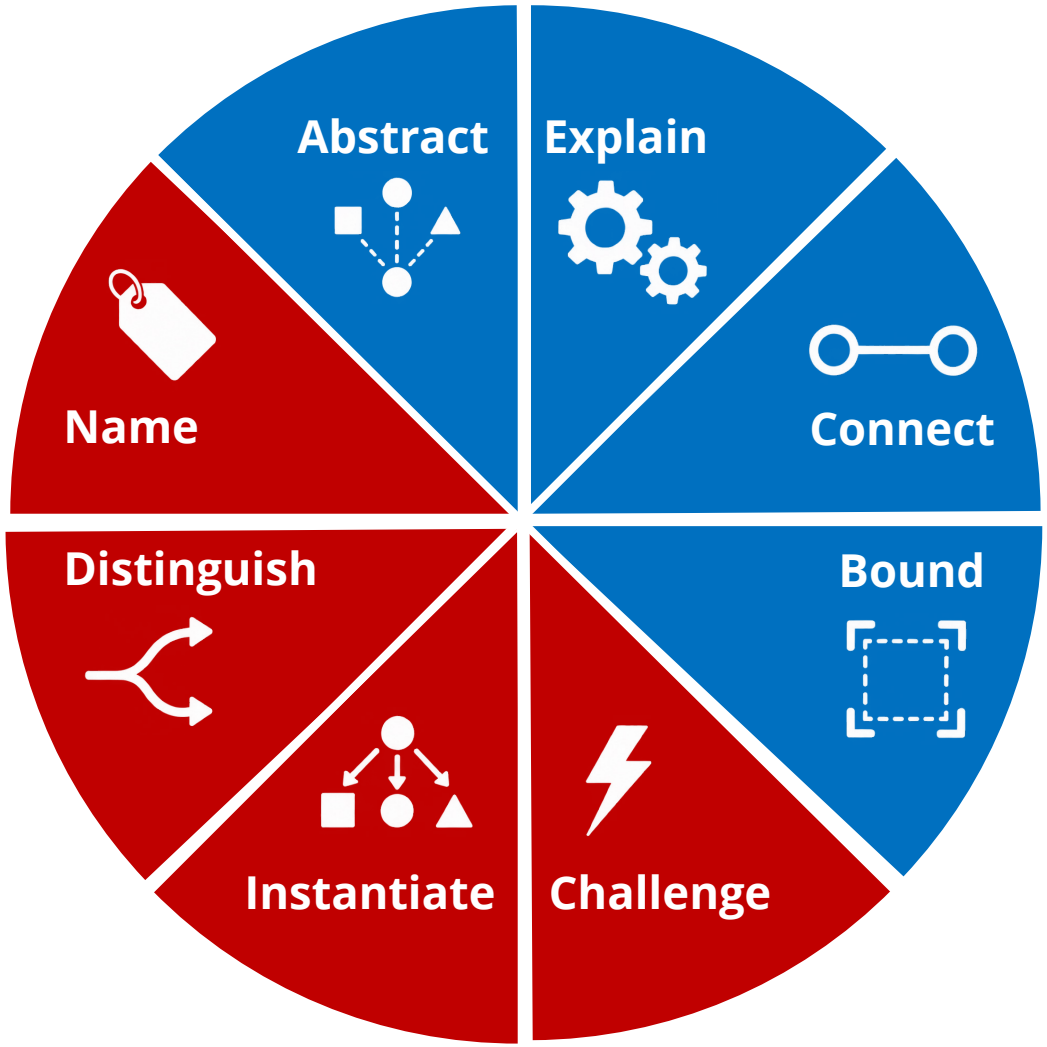


DIFFERENTIATE

Naming, distinguishing it differentiates the thing

Instantiating turns generality into particularity

Challenge casts shadow on assumptions of the phenomenon



UNIFY

Abstracting unifies different things in the same category

Explain makes observations coherent

Connect unifies two intellectual traditions

Bounding unifies contradictory findings by specifying conditions

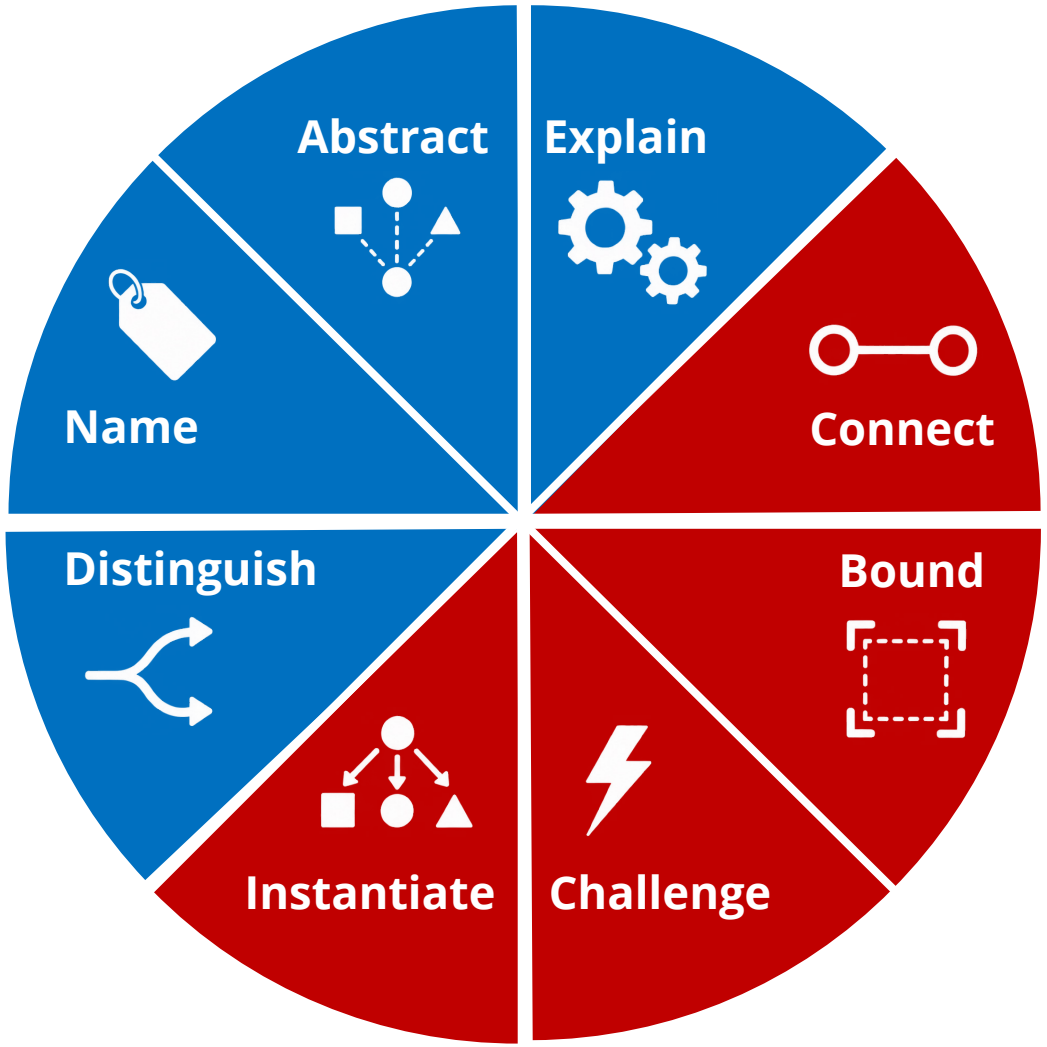
CONSTITUTE

Give the phenomenon an identity

Clarify what a phenomenon is and is not

Identify higher order category

Constitute an intelligible account of a phenomenon



SITUATE

Situate one body of knowledge relative to another.

Situate a claim relative to bounding conditions

Situate an established view against an alternative.

Situate a general theory in a new domain.

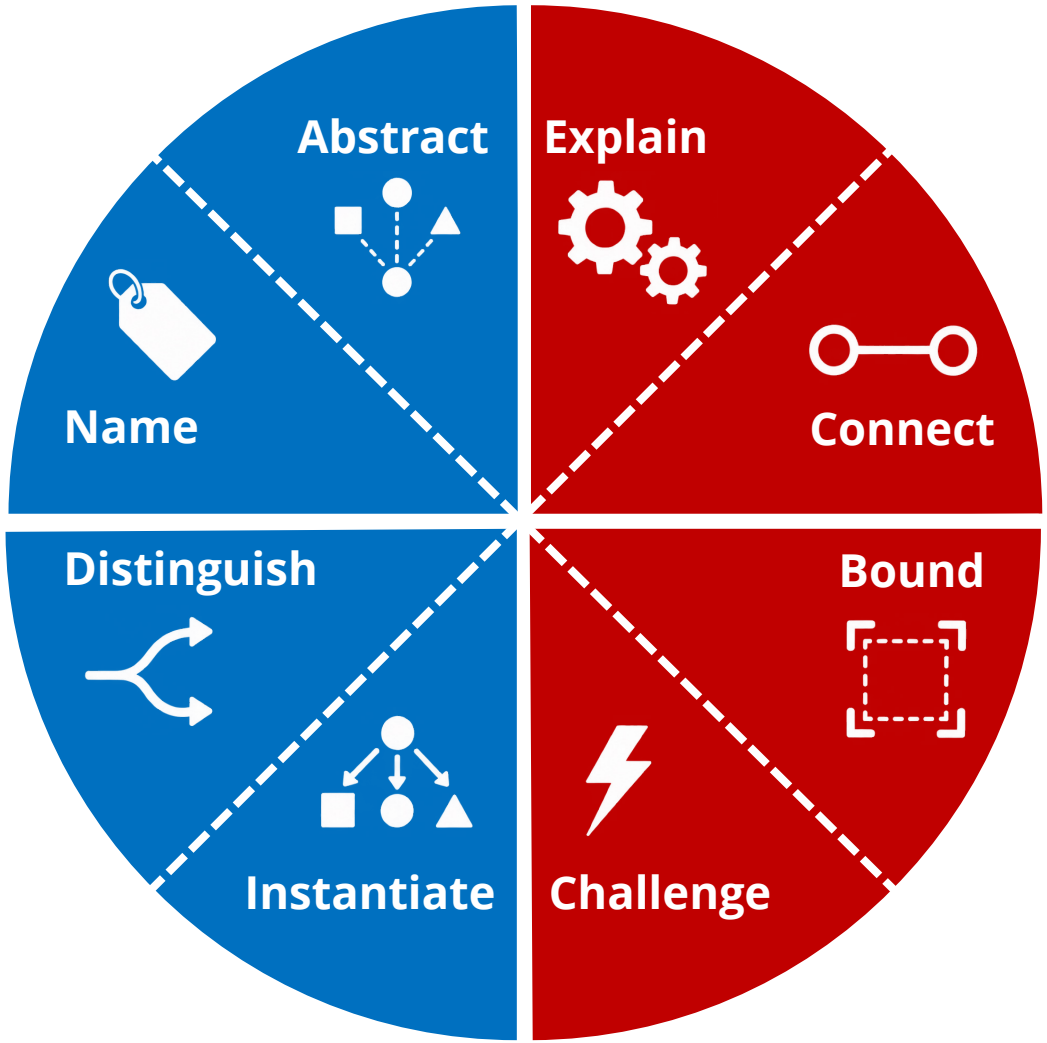
REPRESENT

Naming changes what we see

Abstracting changes the level of representation

Distinguishing changes the way we represent a thing

Instantiate makes a concrete representation of a general theory



PREDICATE

Say how/why

State relationships

Qualify claims

Question assumptions & assertions

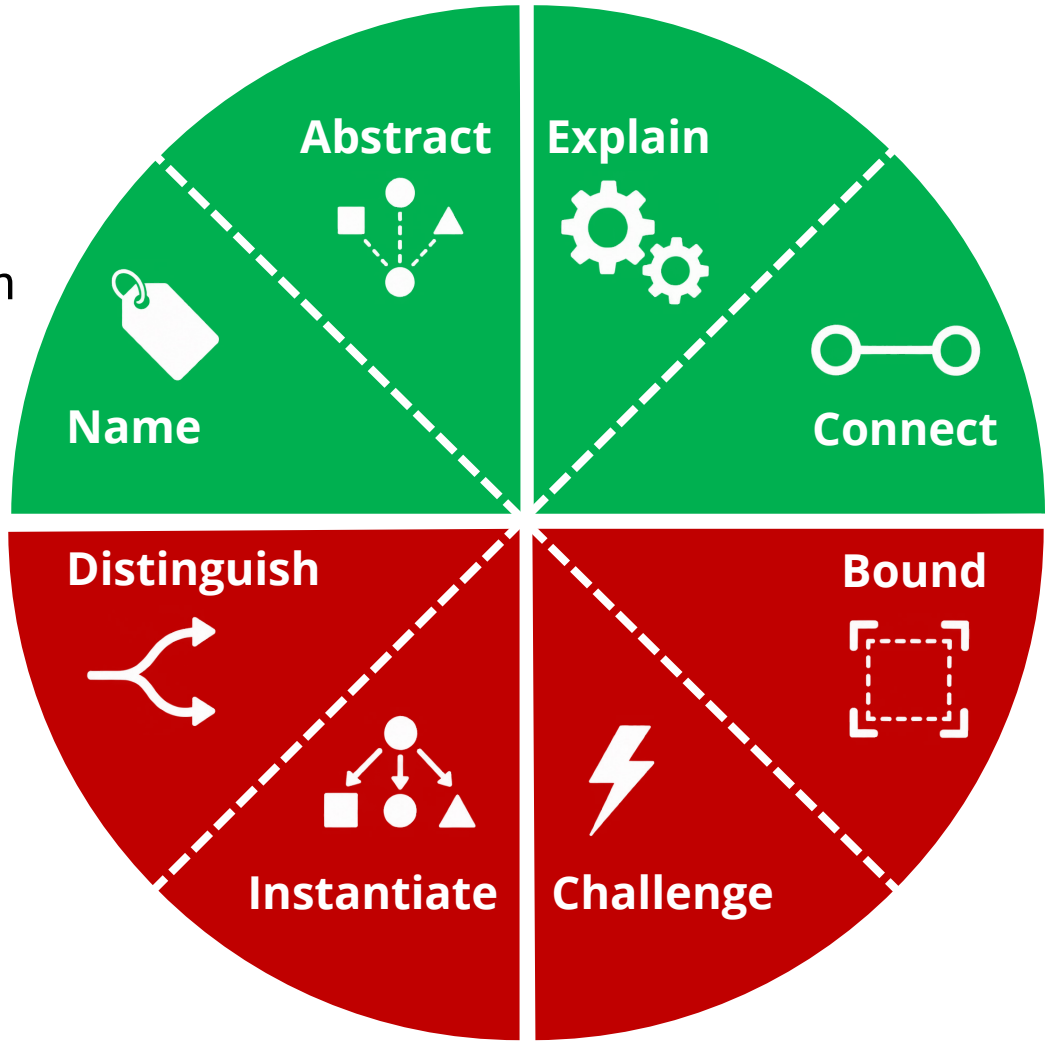
EXPAND

Expand by creating a new object of inquiry

Expand by moving from concept to broader class of concepts

Expand body of knowledge by supplying an explanation

Expand by linking two intellectual ideas



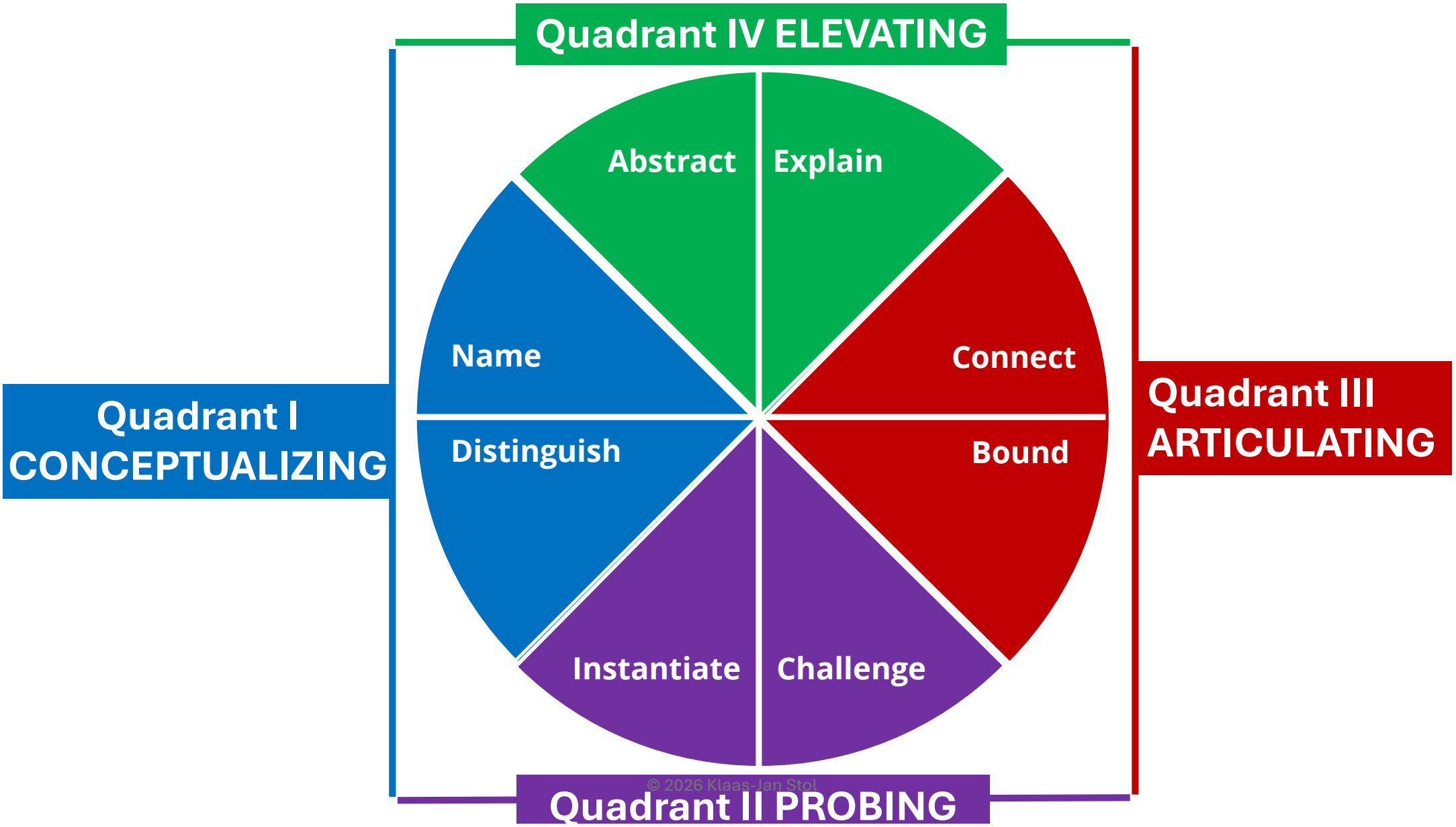
DISCIPLINE

Discipline a broad claim

Challenge an existing claim

Constrain abstraction to a concrete instance

Sharpen an existing concept into multiple different things



How to theorize? Basic operations / primitives for theorizing – a RISC architecture

A vocabulary for theorizing in SE research

Helps address: What is the theoretical contribution (=longer surviving knowledge)?

Does THIS PRESENTATION make a theoretical contribution?