



Reverse-Engineered Validity: Retrofitting a Legacy Test Through Assessment Engineering

47th Language Testing Research Colloquium
Troy Cox, Ph.D., Brigham Young University
Petia Alexieva, Ph.D. , **DLIELC** (not FLC)
Matthew Wilcox, Ph.D., Brigham Young University
June 4, 2026



The Keyboard That Wasn't

On entrenchment, switching costs, and why “better” doesn’t always win

Freshman year at BYU

I was introduced to the Dvorak keyboard.
Ergonomic. Efficient. Faster. The evidence was
real. The logic was sound.

This would be the future, they said.

And yet — QWERTY is still here. On every keyboard
in this room.

Not because it's better. Because it was first.



Why systems persist

Once established, a system creates its own gravity:

- Infrastructure is built around it
- • People are trained to use it
- • Stakeholders have invested in it
- • Switching costs are real and high
- That was my first thought when I sat down with DLIELC and looked at their ECL test.
- *This test isn't going away. So what do we do with it?*

What Is a Wicked Problem?

Rittel & Webber (1973): problems that resist clean solutions

“A class of social system problems which are ill-formulated, where the information is confusing, where there are many clients and decision makers with conflicting values, and where the ramifications in the whole system are thoroughly confusing.”

No Definitive Formulation

Every attempt to solve it changes your understanding of the problem itself

No Stopping Rule

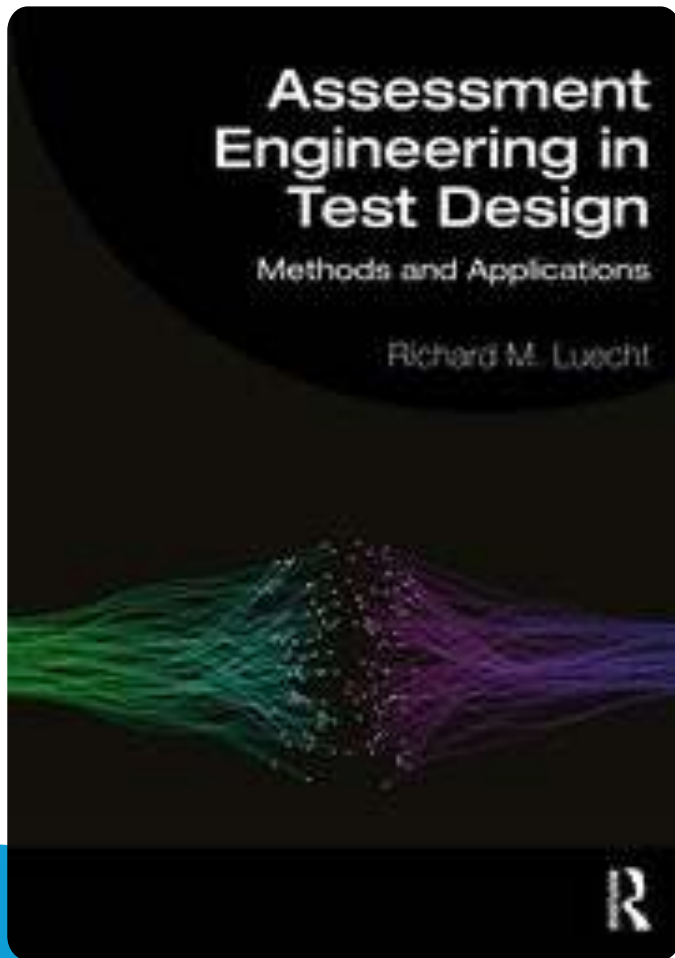
Solutions are not right or wrong — they are better or worse, and you never fully “finish”

Multiple Stakeholders

Each stakeholder has a different read on the problem — and a different definition of success

Entrenched Systems

Solutions must work within constraints that can't easily be removed. Like QWERTY. Like a legacy test.

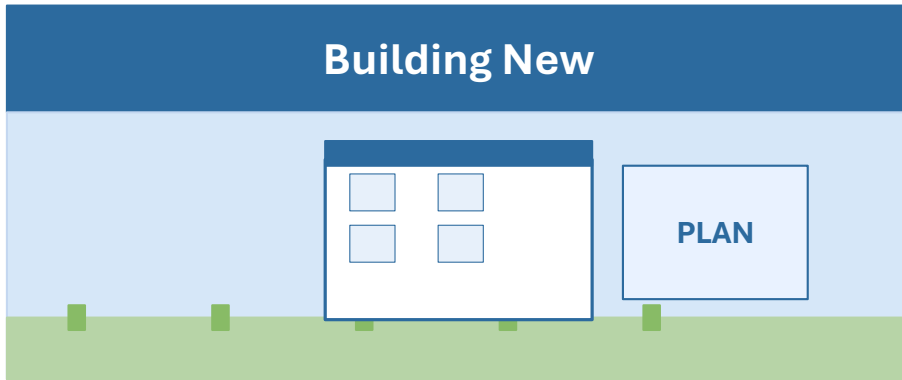


The Framework Behind Our Work

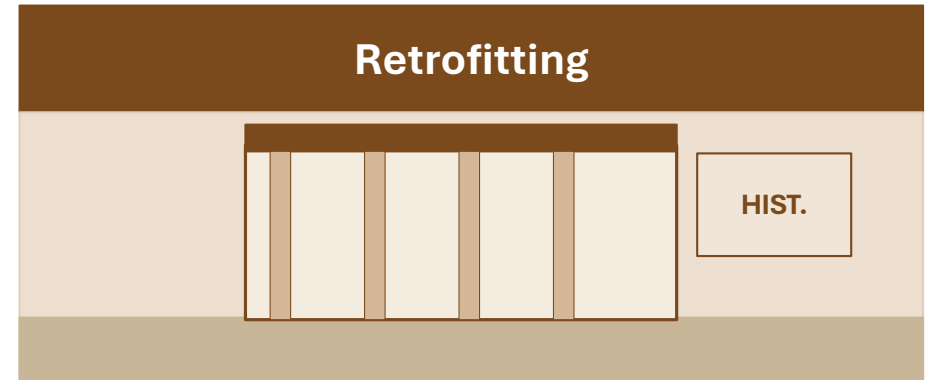
- **Assessment Engineering in Test Design**

<https://www.routledge.com/Assessment-Engineering-in-Test-Design-Methods-and-Applications/Luecht/p/book/9781032586823>

Two Ways to Build



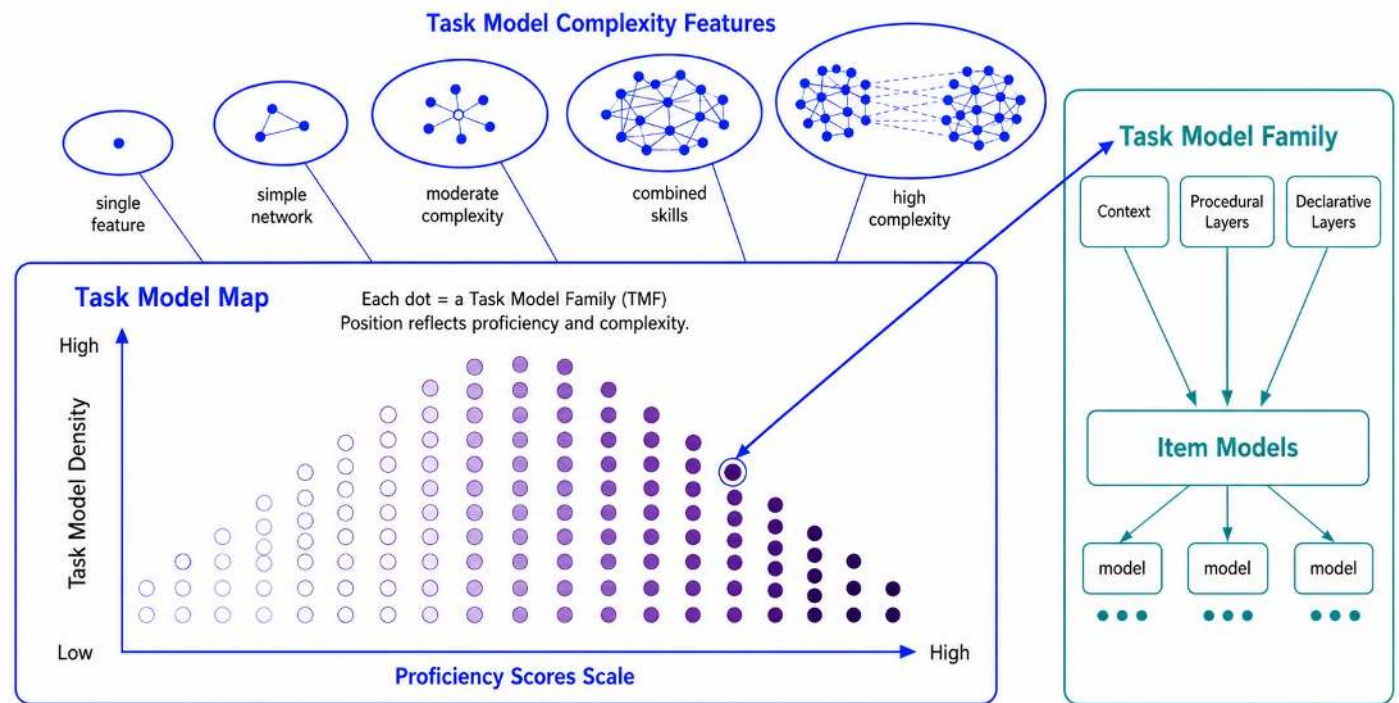
- Open site — no existing constraints
- Design your plans from scratch
- Build exactly what you need
- Everything conforms to your vision



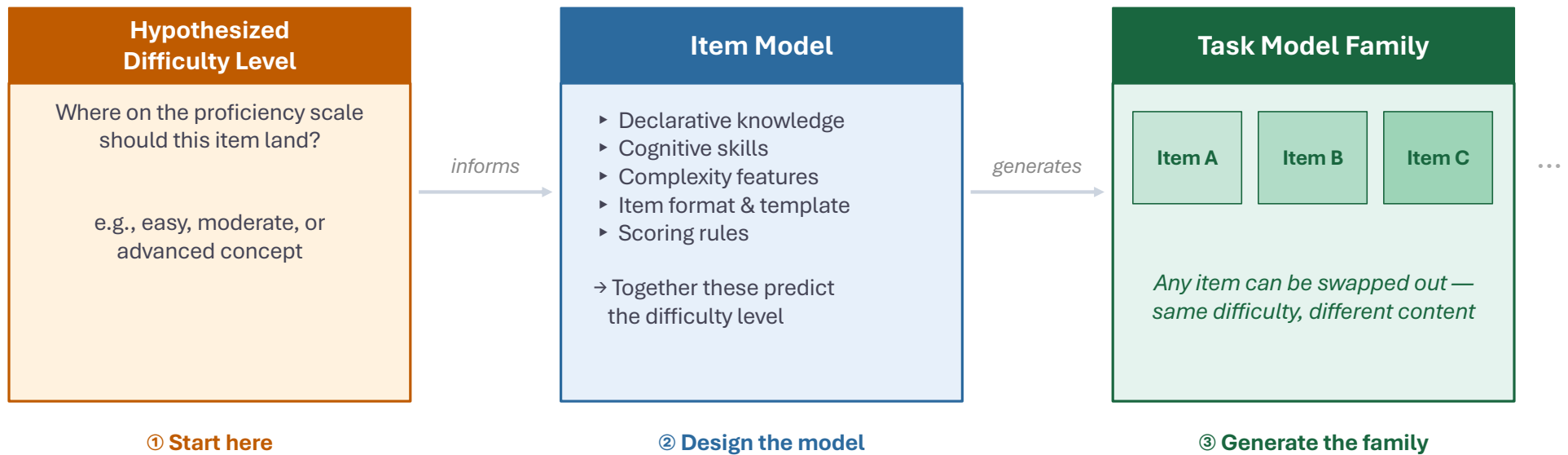
- Existing structure — can't start over
- Work within historical constraints
- Preserve what already functions well
- Adapt old to meet new requirements

Our Solution: Assessment Engineering

Luecht 2025: Task Model Map → Task Model Families → Item Models → Items



From Hypothesized Difficulty to Task Model Family

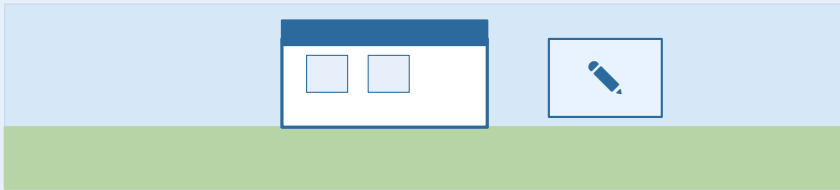


Items within a Task Model Family are isomorphic: same cognitive demands, different surface content.

Two Approaches to Test Design

An architectural analogy

Building New



Open site: start from scratch

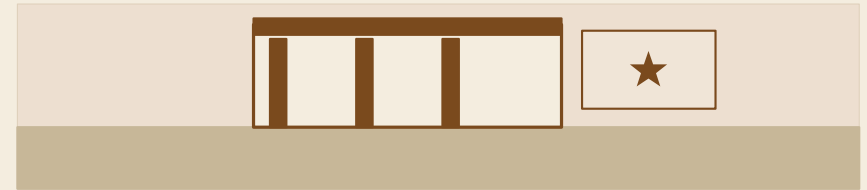
Fresh blueprint: design what you need

No constraints from existing structures

Build items directly from task models

Ideal for new test programs

Retrofitting



Existing structure: ECL item banks

Preserve what works: calibrated items with known difficulty/discrimination

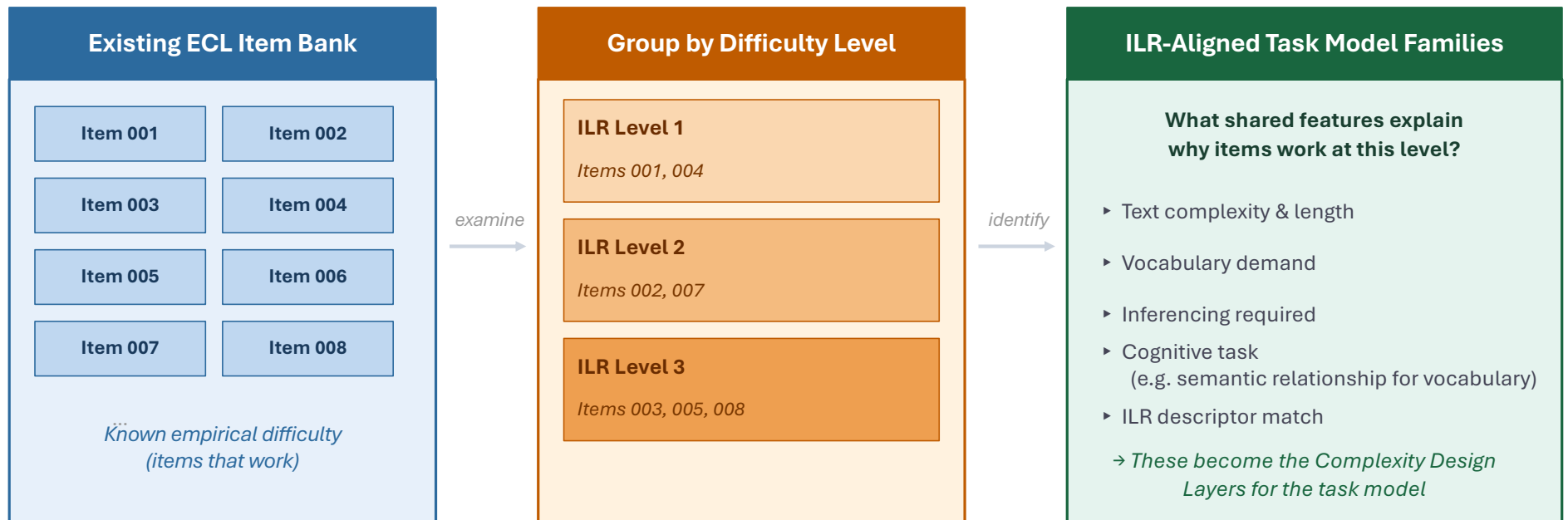
Work within constraints of ECL infrastructure and ILR scale & history

Reverse-engineer: find features that explain difficulty

Our situation with the ECL

Retrofitting: From ECL Items to ILR-Aligned Task Model Families

Working backwards from what already works



Goal: use existing ECL items to ground-truth the task model features before writing new items from scratch.

The Leap to ECL 2.0 and Its Limitations

2PL Model

Equated Content

Reliable Scores

Dynamic Seeding

Lack of Score Descriptors

Case Study

Deconstructing ECL Vocabulary Items: From “What” to “How”

Semantic Relationship	Task Models
Synonymy & Antonymy	Recognize semantic equivalence or opposition
Definition	Map a description to its specific lexical label
Cause/Effect & Function	Perform logical inference Perform functional inference
Part of a Whole & Location	Access schematic knowledge of structure or place
Classification & Trait	Perform hierarchical or associative reasoning
Grammatical	Perform syntactic-semantic analysis

Anatomy of a Task Model: TMF-CE-BEG-01



Task Model ID: TMF-CE-BEG-01



Task Family: Cause/Effect



Target Proficiency: ECL 0-49
(Projected ILR 0+ / 1)



Cognitive Task

Identify the direct, physical cause of a concrete, observable effect.

The Item Blueprint



Stimulus (The Prompt)

Template: A 'Why' question about a concrete, observable state.

Example: "Why is there glass on the floor?"



Stem (The Question)

Template: A blank sentence stem for the answer.

Example: "_____."



Options (The Choices)

Key: A noun phrase describing the direct physical cause.

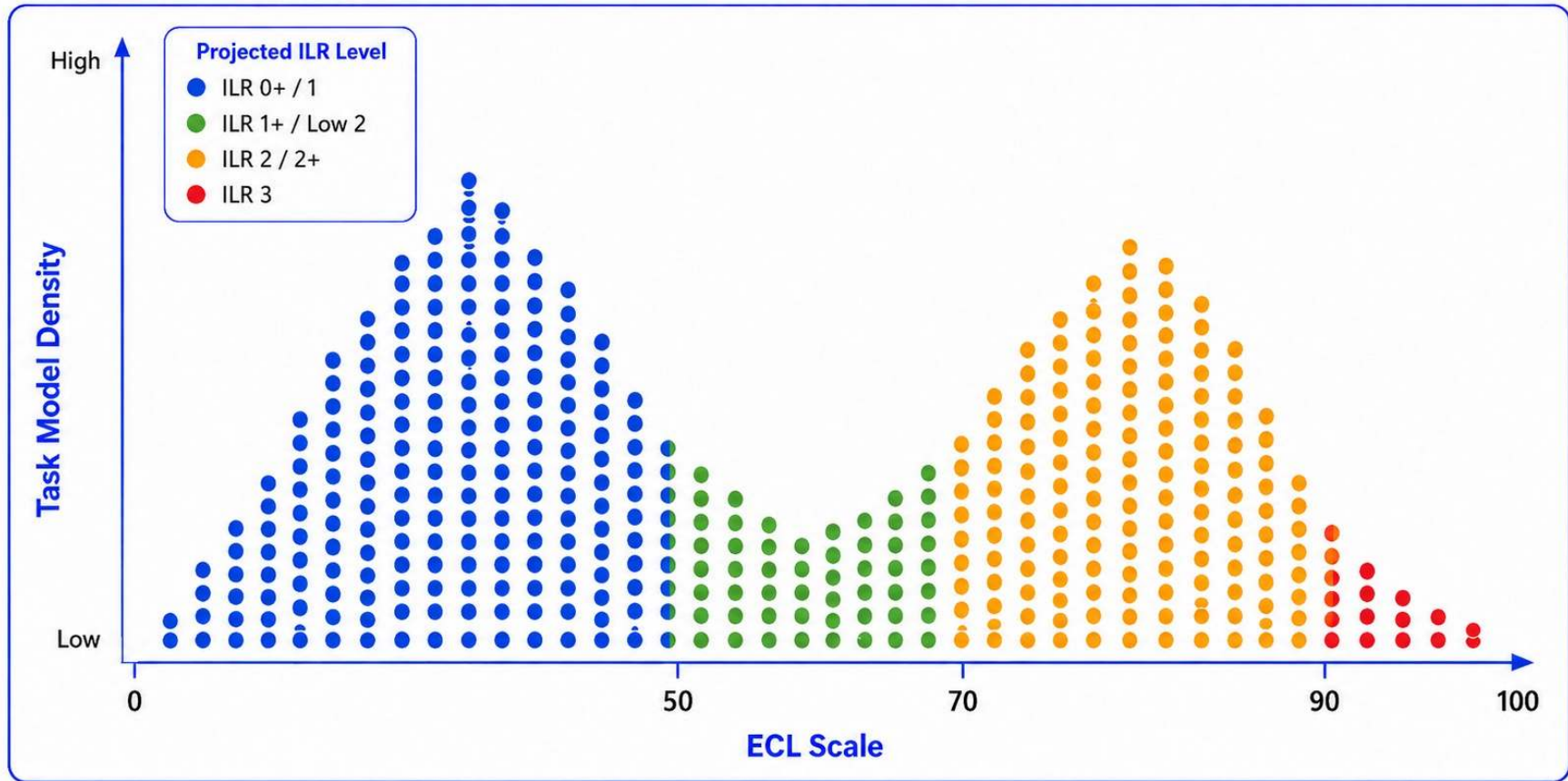
Distractors: Unrelated but plausible events from a different context.



Difficulty Drivers (Engineered for 'Low' Difficulty)

- ✓ **Concrete Topic:** Uses everyday objects and events.
- ✓ **High-Frequency Vocabulary:** No lexical surprises.
- ✓ **Direct Causal Link:** Requires no complex inference.
- ✓ **Thematically Distant Distractors:** Easy to eliminate wrong answers.

Task Model Map: Cause/Effect Vocabulary



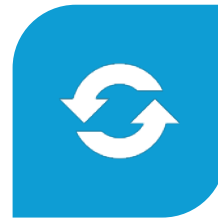
Our Path Forward with AE



DEVELOP



VALIDATE



REFINE



DEFINE



Reverse-Engineered Validity: Retrofitting a Legacy Test Through Assessment Engineering

Questions & Discussion