Methods to strengthen the first line of defence

Andy Spero, PhD
CEO at Spero Risk Associates

Outline

1. Strengthening the ~~first line of defence~~, err, don't we mean **model development**? (~20 minutes)
   i. New ideas and techniques for model development
   ii. Governance around model development

2. Data aggregation and quality (~10 minutes)
   i. The (general) implications of poor data quality
   ii. Corrections and supplemental or corroborating evidence including, enforcing uniformity & consistency through policies and procedures

3. Questions—although you are welcome to ask at any time (~10 minutes)

1.A.1. Strengthening Development: Preface

Is "**the three lines of defence**" an apt metaphor for modeling?

I would argue "no," for at least three related reasons:

1. Who are we defending against?

1.A.2 Strengthening Development: Preface

Is "**the three lines of defence**" an apt metaphor for modeling?

Two more reasons:

2. **Organizationally, it dissipates or shifts responsibility from where it should be.** Too much emphasis is placed on the checkers (in validation) and the checkers checking the checkers (in audit), and **that can be very expensive**.

3. **Individually, from a day-to-day task perspective, it can lead to the wrong incentives for developers.** There is a motivational difference between the positive, "**do the right thing**," and the negative, "don't do the wrong thing," or "**wait for validation to say if you did the wrong thing**."

1.B. Strengthening Development: the Problem

What is the overall problem we're trying to solve?

Build a conceptually-sound, well-implemented model to:

Let's stop here and ask: what is a **model**?

1.B.Aside. What is a model?

The Fed defines a model very mechanically as:

A process that apply a theory (or theories) from stats, math, econ, or finance to convert inputs (data or assumptions) into quantitative estimates… (I would add) via calculations that represent observed or assumed relationships.

It's a fine definition, and has its use, but in the spirit of Leonardo's "*Simplicity is the ultimate sophistication*," conceptually, a model is:

A simplified representation of reality.

Of course, that is reality as we understand it, which we don't very well. (Which is why a lot of the work is research, not development.)

1.B. Strengthening Development: the Problem

What is the overall problem we're trying to solve?

Build a conceptually-sound, well-implemented model to:

1. Appropriate functionality or standards
2. With available resources
3. Given a variety of **very** binding constraints

Note: having to staff development, validation and audit leads to an interesting sub-problem: how do you allocate human resources? Or, where do you put your best associates and what attributes should they have?

1.C. The Modeling Problem, restated.

Maximize the design's *representational faithfulness* (by minimizing the difference between the **world** & the math)

By deploying the appropriate resources, and

Subject to (or constrained by):

A. Available data (and data quality)
B. Our understanding (based on theories, anecdotes…)
C. Initial development costs (people and systems)
D. Subsequent validation costs
E. Failure-related costs (both internal and external, which I will define shortly—after defining "quality")

1.D. A Brief Detour through Quality Costs

There are at least two types of quality:

1. **Quality of Design** – the desired features and robustness of the item to be produced: how close is the simplified representation to **our** understanding of reality?

   In some ways, we can translate "**Quality of Design**" as "**Conceptual Soundness**."

2. **Quality of Conformance** – the ability to produce output to the desired design standards: are the mechanics both transparent and reliable?

1.E. Quality Activities & Costs

Quality-related activities can be categorized as:

1. **Prevention** – choosing inputs to improve the model's conformance quality or the chance of success, given clear design standards

2. **Appraisal** – in-process or post-construction inspection, like validation.

3. **Internal Failure** – rework and retesting and all costs associated with failures discovered prior to use, e.g., fixing findings

4. **External Failure** – the potential for adverse consequences, which frequently magnifies other risks, and are associated with failures after use—discovered or not.

   Note: there are adverse consequences in #3, too.

1.F. Uncertain Quality and Model Risk

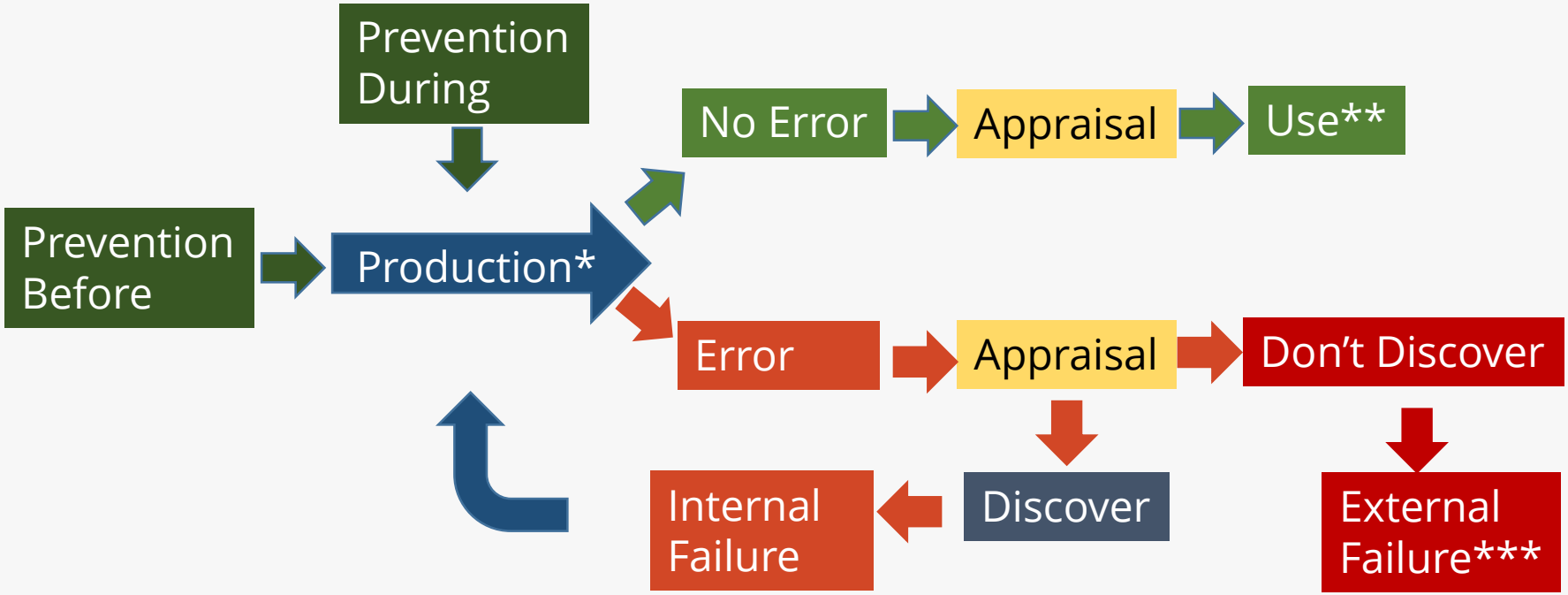
Model risk, the potential for adverse consequences from using a model in decision-making or reporting (or further processing) exists because of errors that arise from:

A. The Mostly **Uncontrollable, External Environment**:
   i. Epistemological issues, i.e., the limits of human knowledge
   ii. Volatility – the world is ever-changing and not stable

B. The **Mostly Controllable, Internal Environment**:
   i. Using (or further reporting) the output of an conceptually-unsound or poorly-implemented model or methods
   ii. Misusing (and/or misunderstanding) an otherwise sound and well-implemented model or method.

Lack of data or bad data, which is often controllable (with foresight), can't always be separated from epistemic issues.

1.G.1. Quality Activities in Modeling Flow Chart

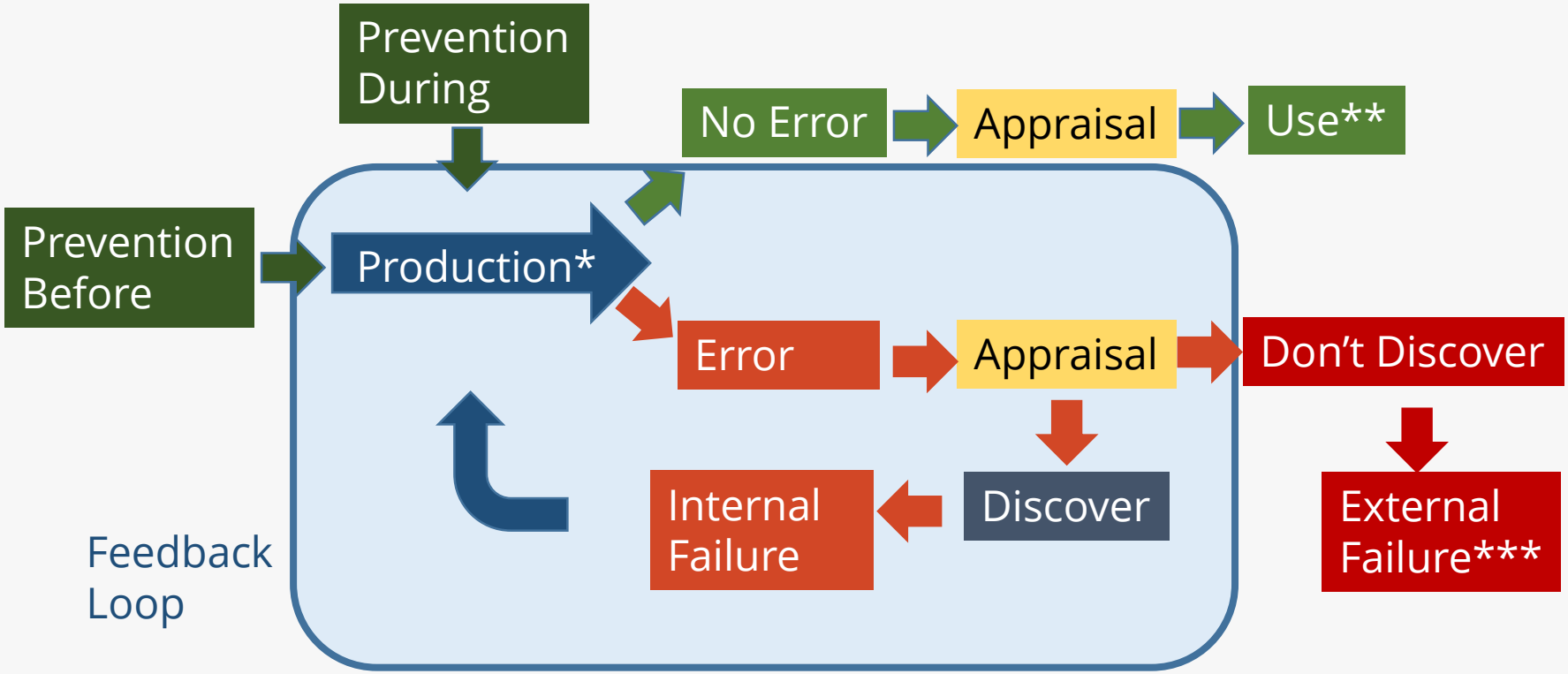


*Model Development: includes design and construction.

**Ignoring false positives and epistemic issues.

***There's a propensity of failure, not a guarantee, and a feedback loop.

1.G.2. The Feedback Loop

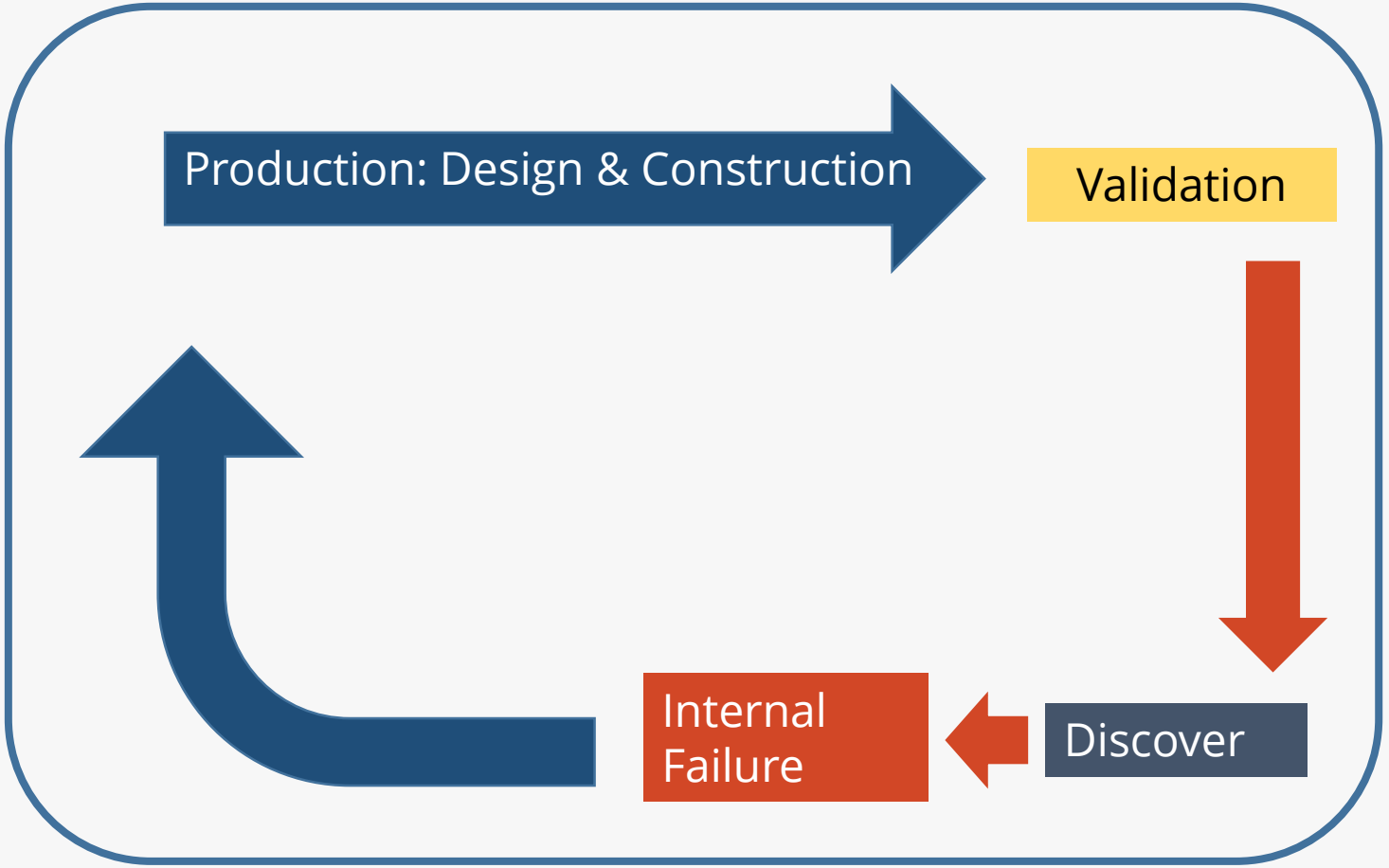


*Model Development

**Ignoring false positives and epistemic issues.

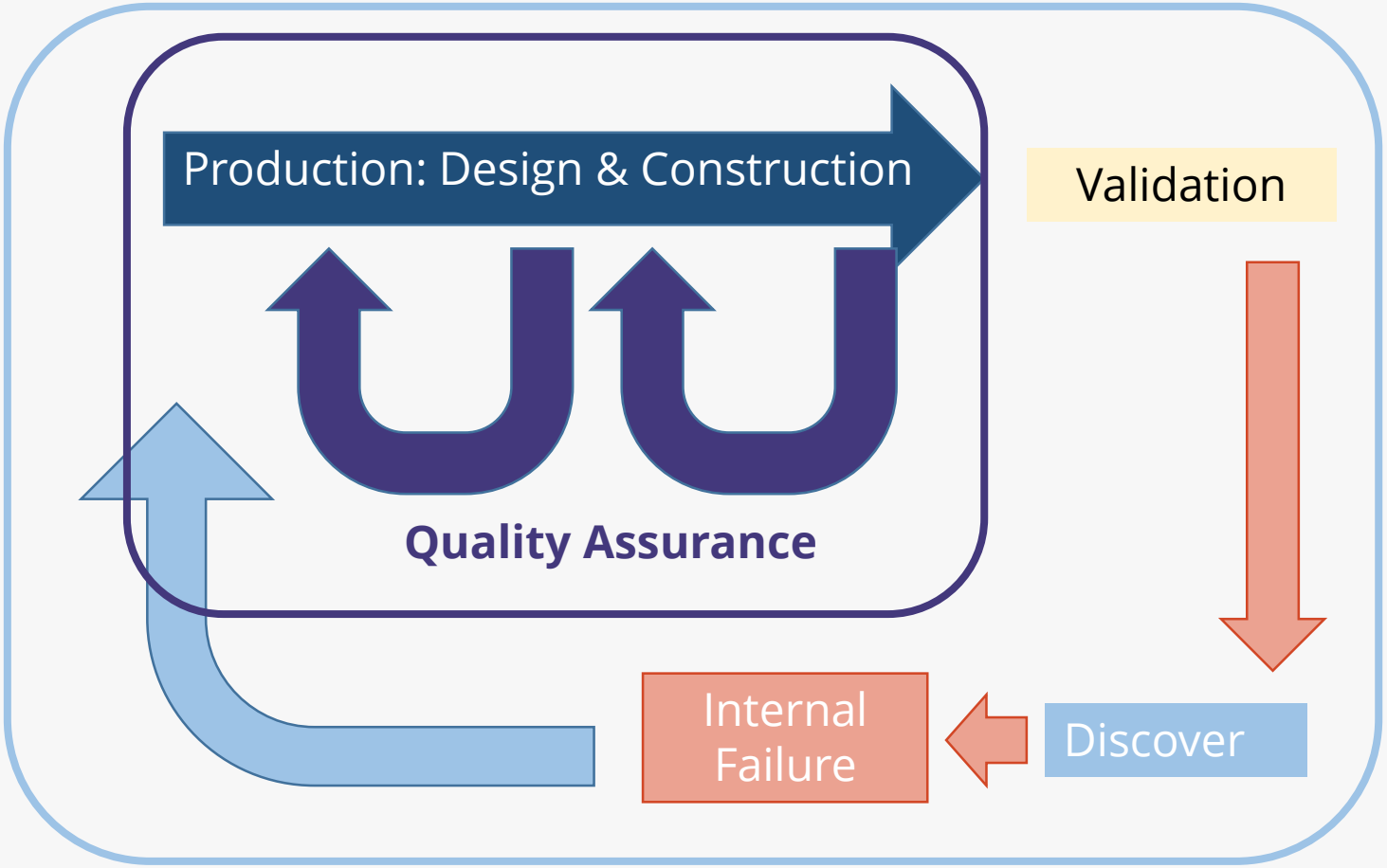
***There's a propensity of failure, not a guarantee, and a feedback loop.

1.G.3. The Feedback Loop: Can It Be Shortened?



*Both initial production and rework.

1.G.4. Yes, the Feedback Loop Can Be Shortened



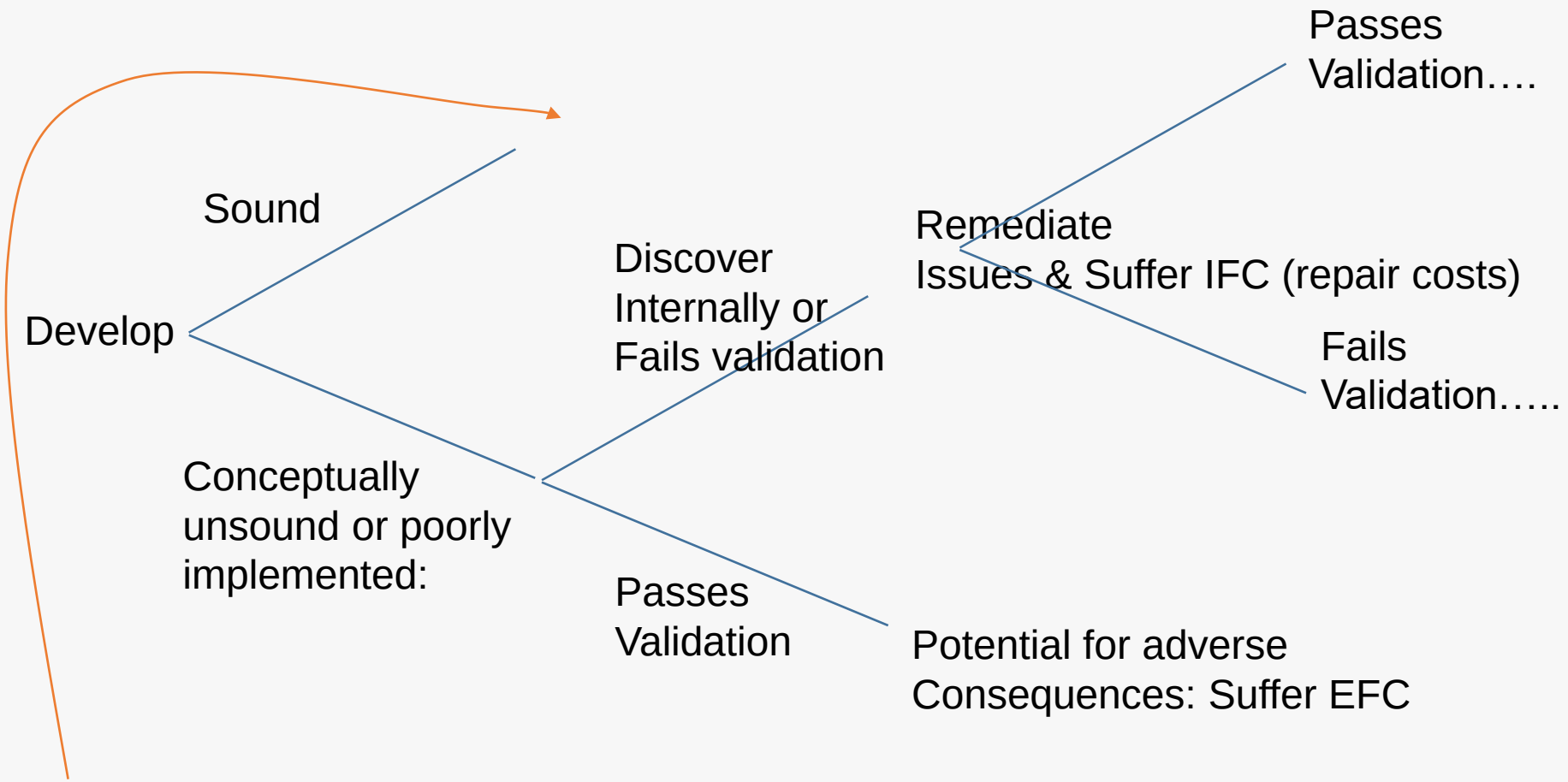
*Both initial production and rework.

1.H. Benefits of QA during Development

1. **Eliminate mistakes, especially silly ones** (that can compound through time)
2. **Get *appropriate* effective challenge early** (often immediately), particularly about *reality*
3. **Keep associates engaged** and not "pigeon-holed" (satisfy their curiosity)
4. **Eliminate key man risk** and **build institutional knowledge**.
5. **Improve efficiencies** by making others aware of existing methodologies, techniques, etc. Improve speed of subsequent development (for similar products or portfolios or on same model for other purposes). Prioritization is crucial, here.

Note: #2 hints that quant skills are necessary, but not sufficient.

1.I. Modeling as a Tree Diagram



1. To save space, we're ignoring the validation of good models as well as individual findings or issues.
2. Like a Greek Tragedy, this tree need not end on any failure branch.

1.J. Strengthening Development through TQM

I was supposed to discuss:

i. New ideas and techniques for model development
ii. Governance around model development

In sum:

1. View development as a production problem with uncertain quality (of both design and conformance)
2. Consider the long-term Sisyphean costs related to validation, redevelopment and revalidation… ad infinitum or nauseum.
3. Mechanical objects need mechanics but simplified representations require conceptual skills and experience with and expertise in the real world.
4. Prioritize, do it right, and then apply elsewhere.
5. QA during development forces transparency (documentation), which is crucial for several reasons.

2. Data Quality

What do we mean by "data quality?"

1. **Short-term**: the lack of missing or incorrect records, given the available field. (This is the narrowest or most tactical definition.)
2. **Long-term**: the richness or comprehensiveness of the data set. (Are we record-keeping or capturing data that can be processed into information, including, say, loss forecasts in different settings?)
    I. Risk factors and other characteristics (breadth)
    II. Length of sample (depth)
3. **Long-term Paradox**: usually, the longer the sample period, the less relevant the earliest observations become.

Note: These aren't problems, they are opportunities… As long as they exist, they will be a need for judgment, experience and expertise.

2.A. Data Quality (through the lens of CCAR)

1. CCAR as a test of risk management. The basic idea is simple: if you can't credibly forecast your losses across different important events, how can you manage your LOB or portfolio?
2. If the data are good enough for modeling, then they are good enough for everything else.
3. Unfortunately, too often, they are not (and that creates implicit—and probably ignored—compromises in day-to-day management and analysis).
4. One of the advantages of (being forced to do) CCAR is that it is a test of risk management capabilities, including (1) real understanding of each LOB and portfolio and (2) data.

Seems reasonable to infer that if you can't forecast your losses because of poor data then luck plays a key role in your risk management. (Implies your not sure what's worth keeping.)

2.A. Caution about Data-mining

Without a theory or sequence of observations or intuition, monkeys can find strong statistical relationships—even in bad or poor-quality data, especially if they lack experience.

2.B. Data Quality (through the lens of CCAR)

There are also advantages to: (1) being compared to your peers and (2) having the Fed estimate losses with their own industry-based models and your firm's data. Both provide valuable feedback loops.

1. **Peer Comparison**: why is your firm's loss rate different than its peers? Is it the (1) modeling approach; (2) differences in business approach, including risk-taking tolerances, or other factors like geography or industry concentration; or (3) in data quality?

2. **Fed Estimates**: why is your firm's loss rate different than Fed's? Is it the model or poor/missing data that the Fed penalizes?

Note: #1 involves database design: is your database rich enough to capture all salient factors (risk drivers)?

#2 involves execution: are you capturing what you should?

2.C. Data Fixes and Considerations

Short-term:

1. **Actual fixes during data prep**, i.e., data cleaning: best if done by developers or—at least—led by development team. Especially if your developers are inexperienced; otherwise, they my find the relationships used to complete missing data or fix errors.
2. **Supplemental or Corroborating Evidence**: Find anything that is available that supports your hypothesis or beliefs. Examples:
    I. Industry-wide call report data to look at losses over longer horizons
    II. The behavior of corporate bond defaults within industries.
3. **Require adherence to policies and procedures:** this is where validation and audit can really help.

2.D. Data Fixes and Considerations

Long-term:

1. Fix it at the source: there is no alternative.
2. Backfill, if possible.
3. Use associates who understand that data are inputs to (creating) information.

3. Questions?