



# The Deeper Causes Behind Alert Fatigue in Automated Systems:

## CDSS Case Study

John Thomas, PhD

Eugenia Kim, MS

Polly Harrington, MS

Nancy Leveson, PhD

Supported by Synensys:

Stephen Powell, DHA

Abigail Harte-Williams, MPH, MS

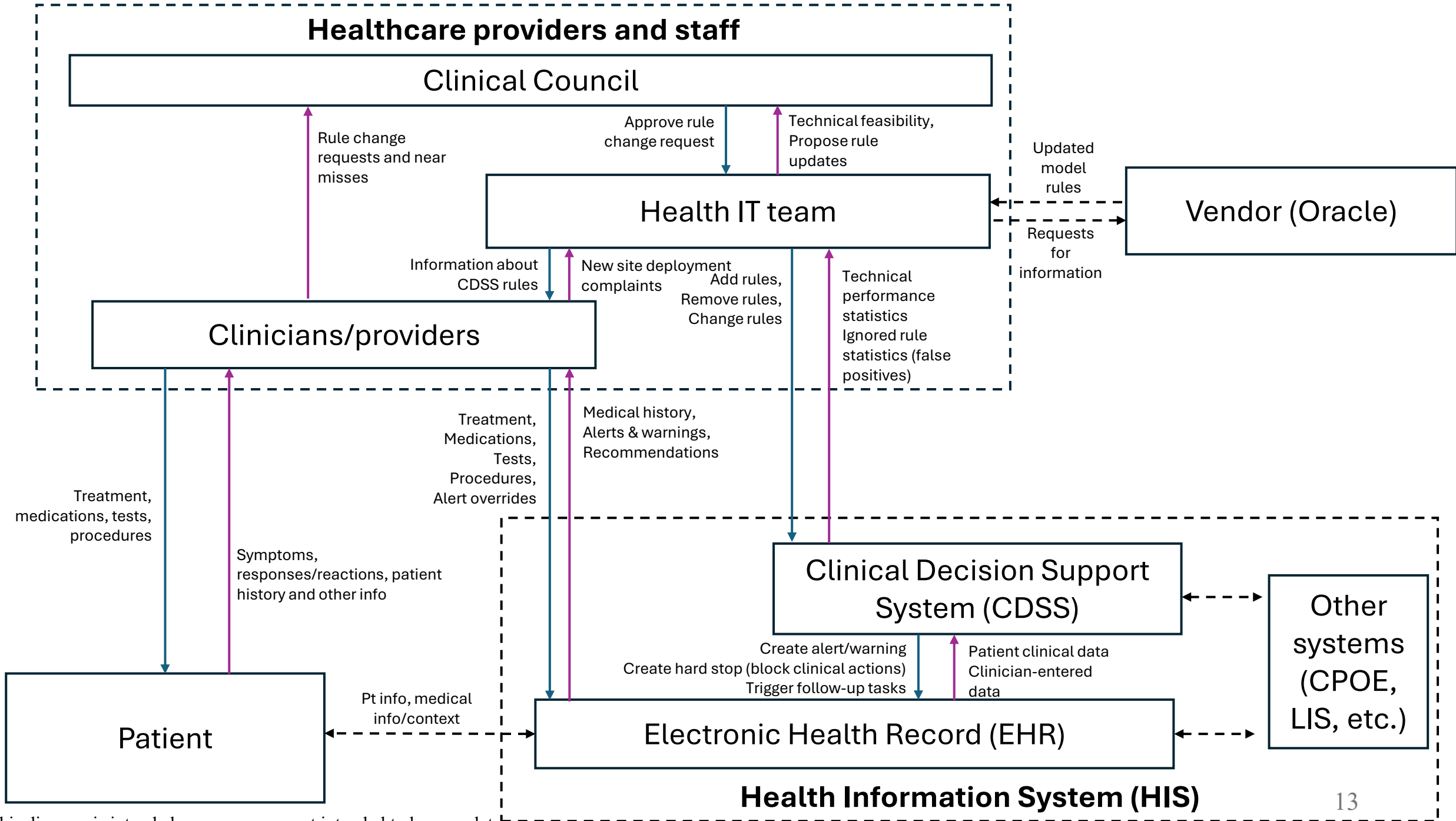
Alana Keller, MBA

Braden Burnett, BBA

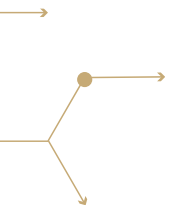


# Introduction

- How can we quickly identify major weaknesses before a loss event?
- Goal: Use STPA to find weaknesses in CDSS technologies **and their clinical environment.**
  - **CDSS: Clinical Decision Support Systems**
  - Software tools that analyze patient data to suggest potential diagnoses, issue alerts to prevent prescription errors, and provide treatment recommendations for clinicians to approve
- Application: Focused on U.S. Department of Veterans Affairs (VA) Health IT systems using Oracle Cerner Millennium.



\*This diagram is intended as a summary, not intended to be complete



# STPA Scenarios

# Example Scenarios

Unsafe Control Action (UCA)	Example Scenario
Health IT team <b>does not accept a rule request</b> when it is technically feasible	A new model rule is provided by the EHR vendor (not requested by clinical councils). The health IT team decides not to implement the rule because the configuration of the EHR doesn't match the configuration the rule is expecting. There may be no clear documentation or metadata from the EHR vendor about the model rule (e.g., the required data elements, rule logic, compatibility with the current system version). "That communication isn't there, we are several versions behind on some of their model rules, and a lot of the new rules we just don't take in at all" Feedback/information is missing from the EHR vendor about the model rule.
Health IT team <b>does not fix a rule</b> when the rule is broken	A rule is broken and not working as intended, but the health IT team does not know that the rule is broken because a ticket has not been submitted to report the broken rule. "So, then the only way we really find out is AMS gets a ticket that [says] hey, this rule broke because there was a change. So really, right now it's very reactive in terms of we find out a rule breaks because somebody starts submitting a ticket on it, and the problem is, is sometimes the rules break, but if nobody submits a ticket, we just don't know that."
Health IT team <b>pulls a data report too late</b> after a rule is not firing as intended	The data report that the health IT team pulls does not indicate when a rule is not firing as intended. The report only contains historical data going back one month. If there is a problem with a rule that was deployed in the system two months in the past, the health IT team will not be able to see that data. The report is not able to be used for prospective analysis of which rules are firing as intended. "And this is where it's that month look behind. So we have this tool, which is our audit report. We tend to use this when a rule is new in the system. [...] So we tend to use this when a rule is new in the system, but it will walk through pretty nicely for every step of the rule. It takes a little bit of time to pull, but the problem with this one is it's only a month look back. So you could see here I can debug a rule very nicely, but if somebody has an issue with the rule that was two months in the past, I'm not going to be able to see that. So this works for debugging, but it doesn't really tell us is a rule firing? Is the rule firing appropriately in that long-term look back? So it makes it harder to use as a kind of prospective analysis of what rules are firing."

# Example Scenarios

Unsafe Control Action (UCA)	Example Scenario
<p>CDSS <b>does not generate an alert</b> when meds are contraindicated</p>	<p>The CDSS receives adequate input on both new medication orders, entered together by the clinician in the same order session, with accurate data showing that these two medications are contraindicated when co-prescribed. However, due to the way the control algorithm and process model are designed, the CDSS compares each new medication only against existing active medications in the patient's chart and does not compare new medication #1 against new medication #2 if they're both part of the same transaction. So, the CDSS does not generate an alert when two contraindicated meds are ordered simultaneously — even though all relevant data (drug-drug interaction knowledge base and order details) were available in the system at the time of processing.</p>
<p>CDSS <b>does not generate an alert</b> when there is too much server latency or not enough compute</p>	<p>Due to gaps in the feedback loop (e.g., infrastructure monitoring system does not send real-time latency and compute utilization data to the CDSS, or the CDSS's process model is built to assume that compute resources are always sufficient), the CDSS does not receive feedback/input that the system is in a high-latency, resource-constrained state. However, the server latency is critically high and/or the compute capacity is insufficient. Because of this, the CDSS continues operating as though resources are adequate, does not detect the performance degradation, and does not trigger any fallback mechanisms, warnings, or prioritization of critical alerts, ultimately not generating required alerts because it silently times out or skips rule checks under the heavy load.</p>

# Why?

- **Process model belief:** Health IT team is not aware of the new model rule
- **Decision making process:** Health IT team is only tasked with model rule update after receiving a vendor-initiated notification

2

## Providers and staff

Council

Health IT Team

**Legend:**

- ↓ Control action
- ↑ Feedback
- ↔ Other communication / coordination (neither control nor feedback)

Vendor (Oracle)

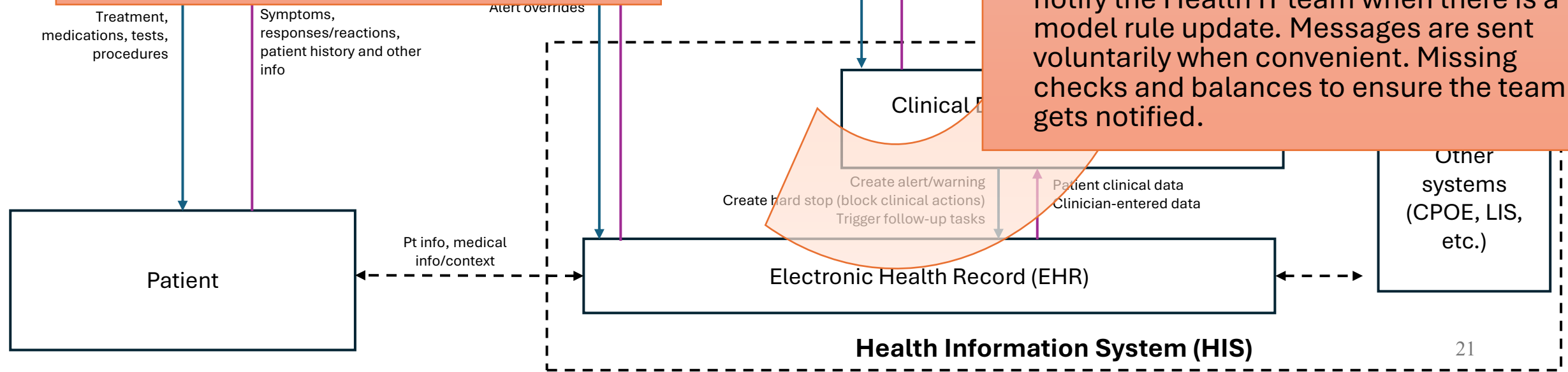
**Unsafe Control Action (UCA):**  
Health IT team does not implement / update a model rule when the EHR vendor has released a new model rule

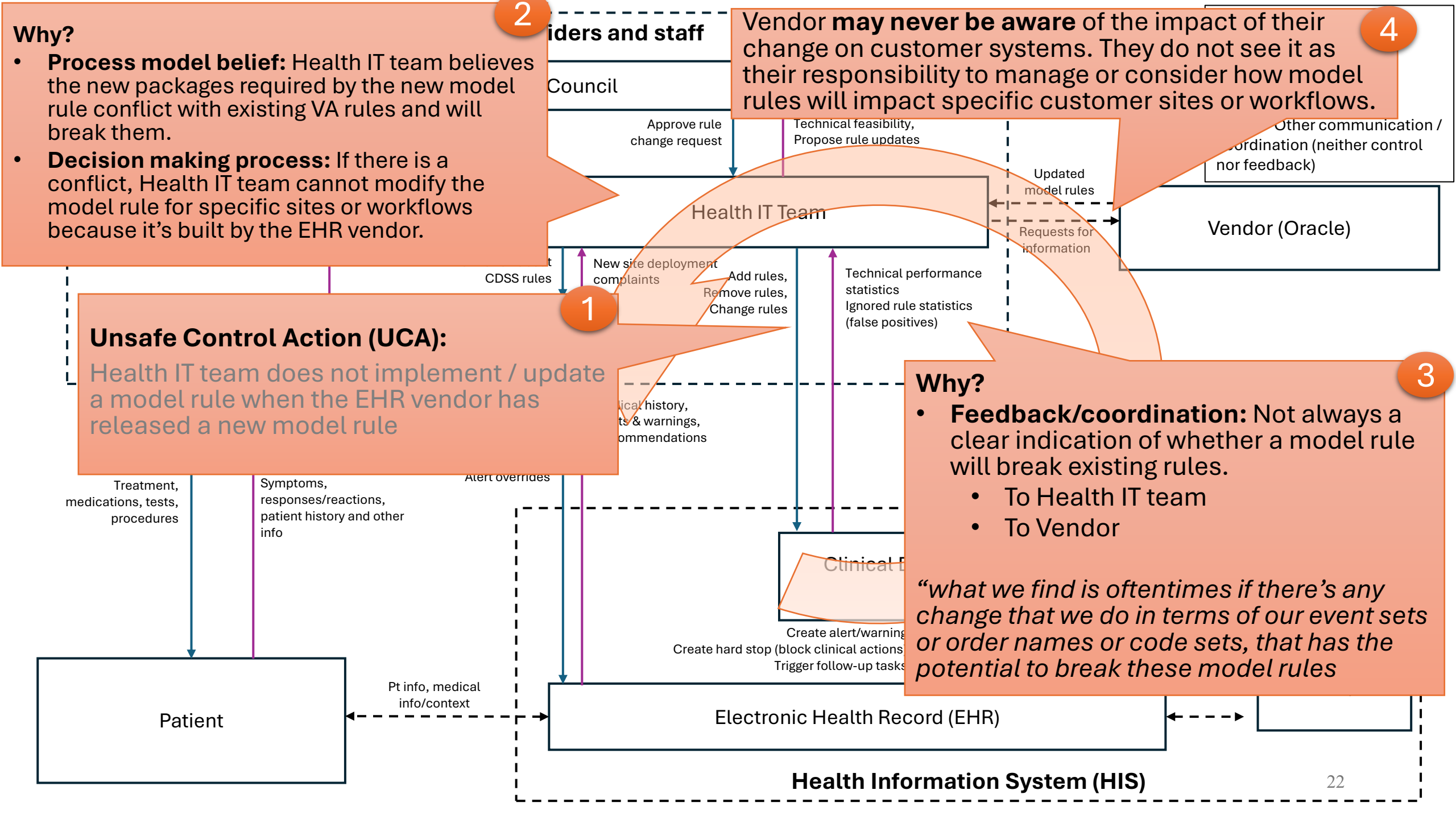
1

# Why?

- **Feedback/coordination:** There is no mechanism to ensure the vendor will notify the Health IT team when there is a model rule update. Messages are sent voluntarily when convenient. Missing checks and balances to ensure the team gets notified.

3





**Why?**

- **Process model belief:** Health IT team believes the new packages required by the new model rule conflict with existing VA rules and will break them.
- **Decision making process:** If there is a conflict, Health IT team cannot modify the model rule for specific sites or workflows because it's built by the EHR vendor.

2

**Managers and staff**

Council

**Vendor may never be aware** of the impact of their change on customer systems. They do not see it as their responsibility to manage or consider how model rules will impact specific customer sites or workflows.

4

**Unsafe Control Action (UCA):**

Health IT team does not implement / update a model rule when the EHR vendor has released a new model rule

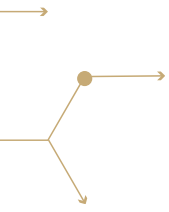
1

**Why?**

- **Feedback/coordination:** Not always a clear indication of whether a model rule will break existing rules.
  - To Health IT team
  - To Vendor

3

*“what we find is oftentimes if there’s any change that we do in terms of our event sets or order names or code sets, that has the potential to break these model rules*



# Summary of Systemic Factors: Opportunities for Improvement

# Systemic Factors

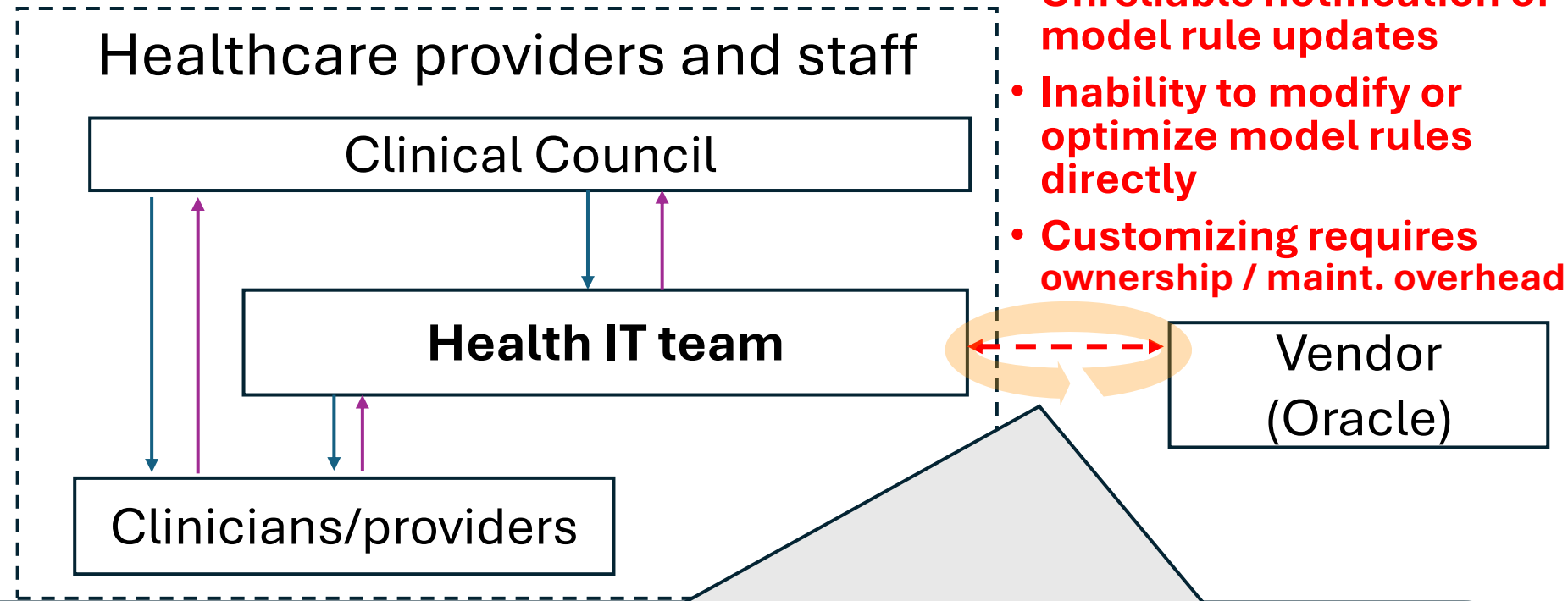
Delayed / Missing Feedback:  
Model Rules

Lack of Customization: Model Rules

Reactive vs. Proactive

Unused & Missing Feedback

Resource Constraints



- **Communication gap**
- **Unreliable notification of model rule updates**
- **Inability to modify or optimize model rules directly**
- **Customizing requires ownership / maint. overhead**

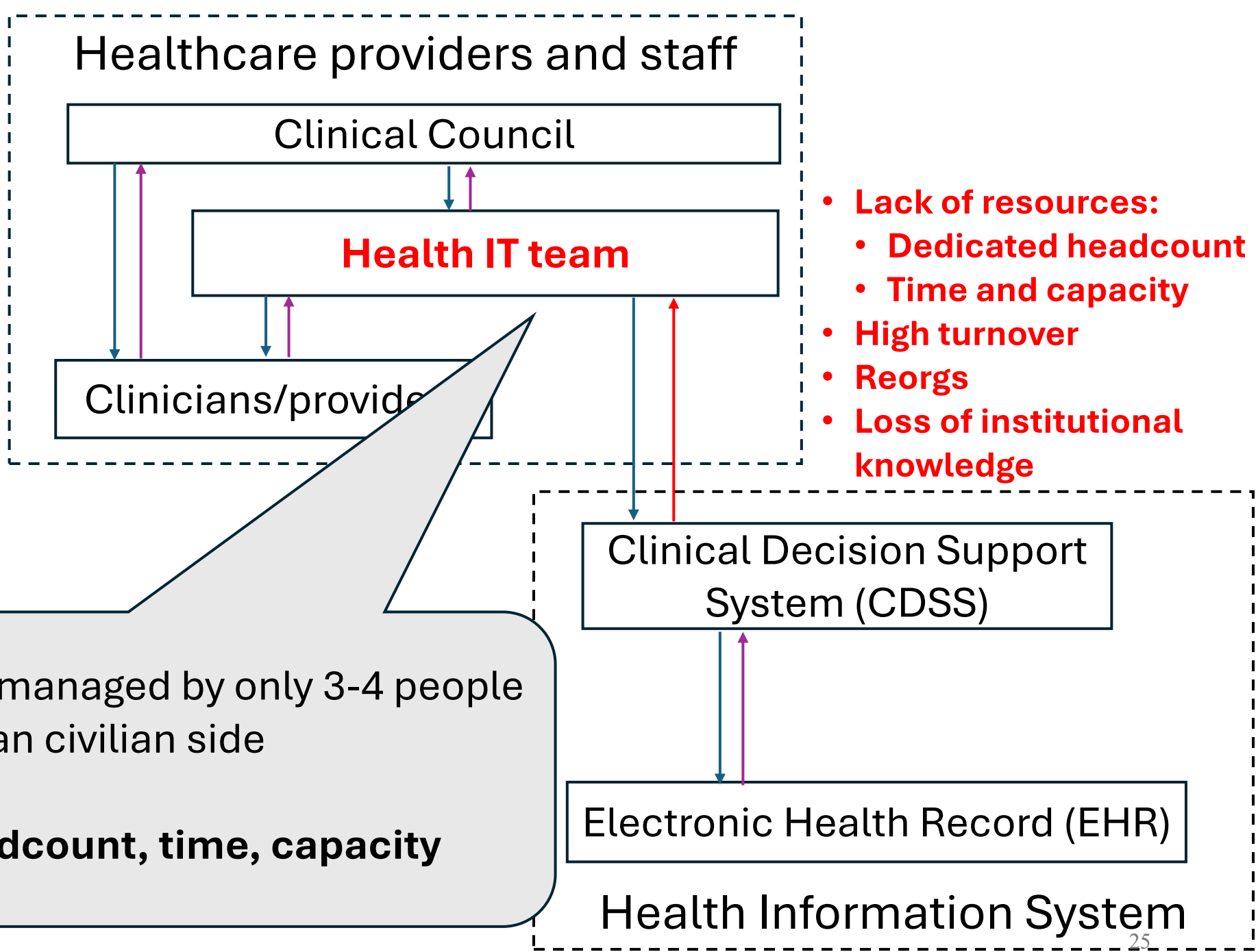
- **Model Rules**
  - Model rules account for **~1/3 of total catalog of rules**
  - VA is **several versions behind** for some model rules
  - VA **not always notified** by vendor of new model rules, so not updated
  - Many near misses from **changes that break model rules** (e.g., event sets, order names, code sets)
  - Model rules **do not always fit within standardization** (may require VA-specific tweaks)
  - Due to inability to modify, the **health IT team often needs to fix** the model rules
  - Fixes by health IT team often broken by future vendor updates
    - Typically unnoticed, impacting end users until they submit a ticket to fix

# Systemic Factors

- Delayed / Missing Feedback: Model Rules
- Lack of Customization: Model Rules
- Reactive vs. Proactive
- Unused & Missing Feedback
- Resource Constraints

• VA has **~4000 rules**, managed by only 3-4 people  
• **~70% more rules** than civilian side

**Lack of dedicated headcount, time, capacity**



- **Lack of resources:**
  - **Dedicated headcount**
  - **Time and capacity**
  - **High turnover**
  - **Reorgs**
  - **Loss of institutional knowledge**

# Systemic Factors

Delayed / Missing Feedback: Model Rules

Lack of Customization: Model Rules

Reactive vs. Proactive

Unused & Missing Feedback

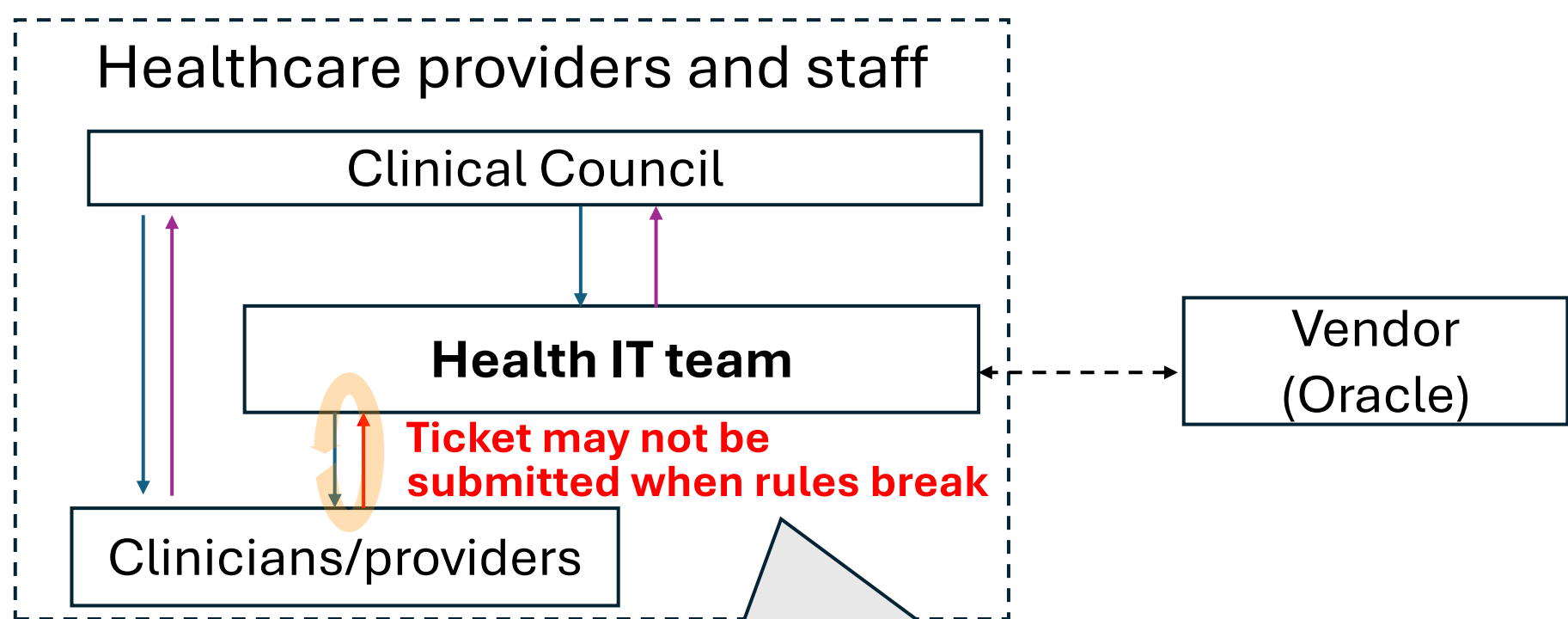
Resource Constraints

Competing Priorities

Lack of

Missing Feedback Rational

Inadequate Learning From Root Cause Analysis (RCA)



## Feedback to Health IT team

- Primarily reactive. **No proactive analysis available** about gaps before a clinical impact
- Only aware a rule is broken **\*if\* clinicians write a ticket** in a voluntary ticketing system
- Current audit report is focused on **technical measures, not clinical effectiveness**
  - No long-term analysis of whether rules are firing appropriately for clinical environment
- Inadequate resources to handle proactive analysis

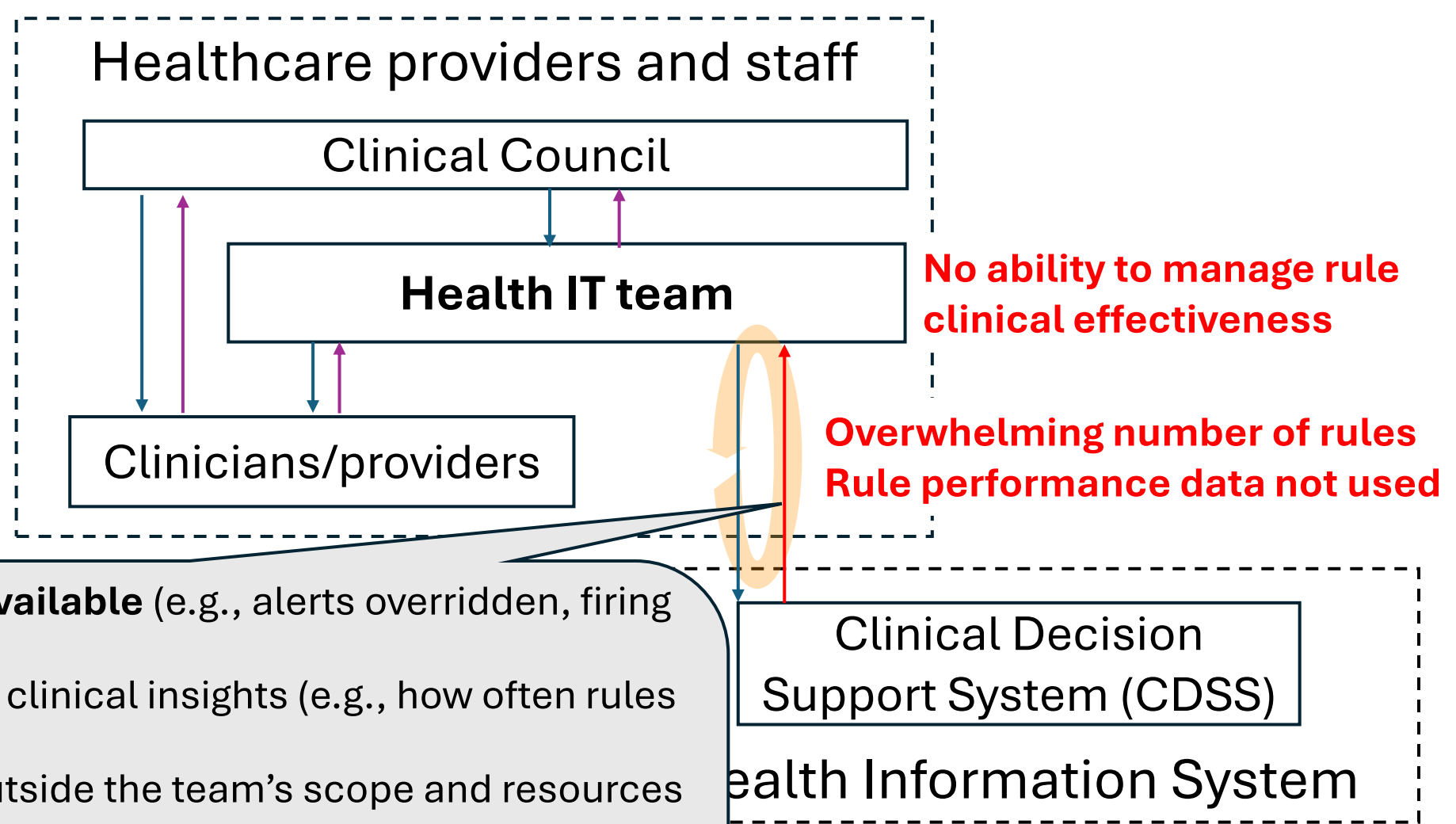
# Systemic Factors

Delayed / Missing Feedback:  
Model Rules

Lack of Customization: Model  
Rules

Reactive vs. Proactive

Unused & Missing Feedback



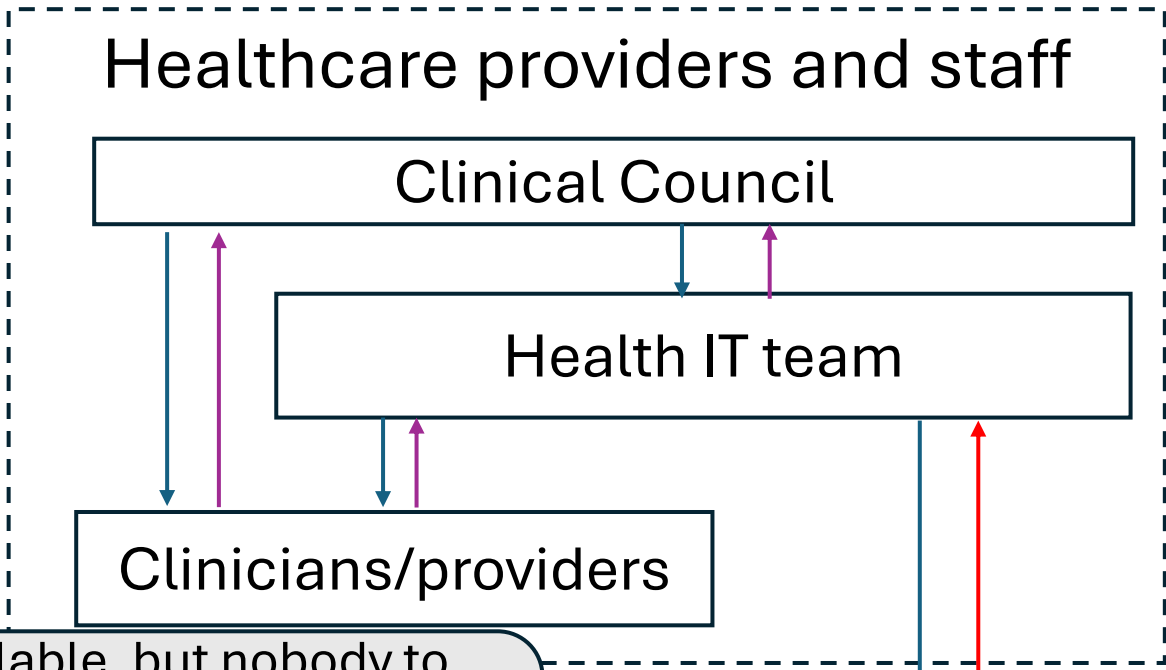
# Systemic Factors

Delayed / Missing Feedback:  
Model Rules

Lack of Customization: Model Rules

Reactive vs. Proactive

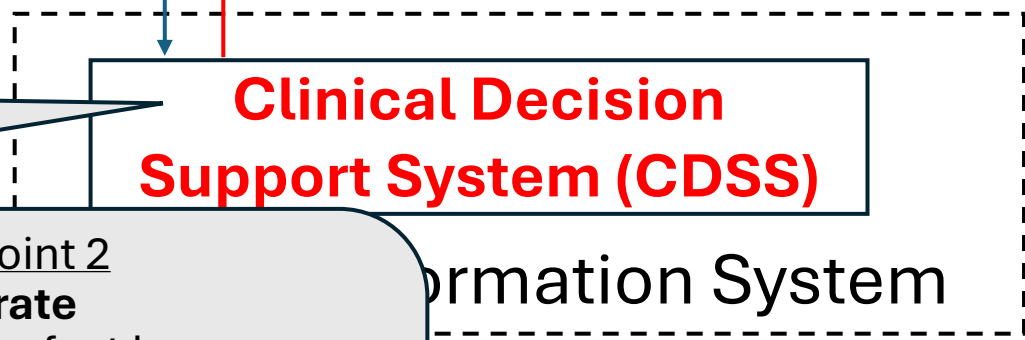
Unused & Missing Feedback



Most data not used  
Some data not clean  
(could be fixed)

Rule performance data available, but nobody to formally monitor:

- **10 million alerts** fired in last 30 days
- The vast **majority of alerts were overridden** by clinicians



Data point 1  
**85.1% alert override rate**, across all agencies and all rules.

- “There are a handful of rules that contribute overwhelmingly to this number and of those several are related to business processes and not necessarily clinical care.”
- **No role to find the clinical override rate.**
- **No role to fix the worst offending rules.**

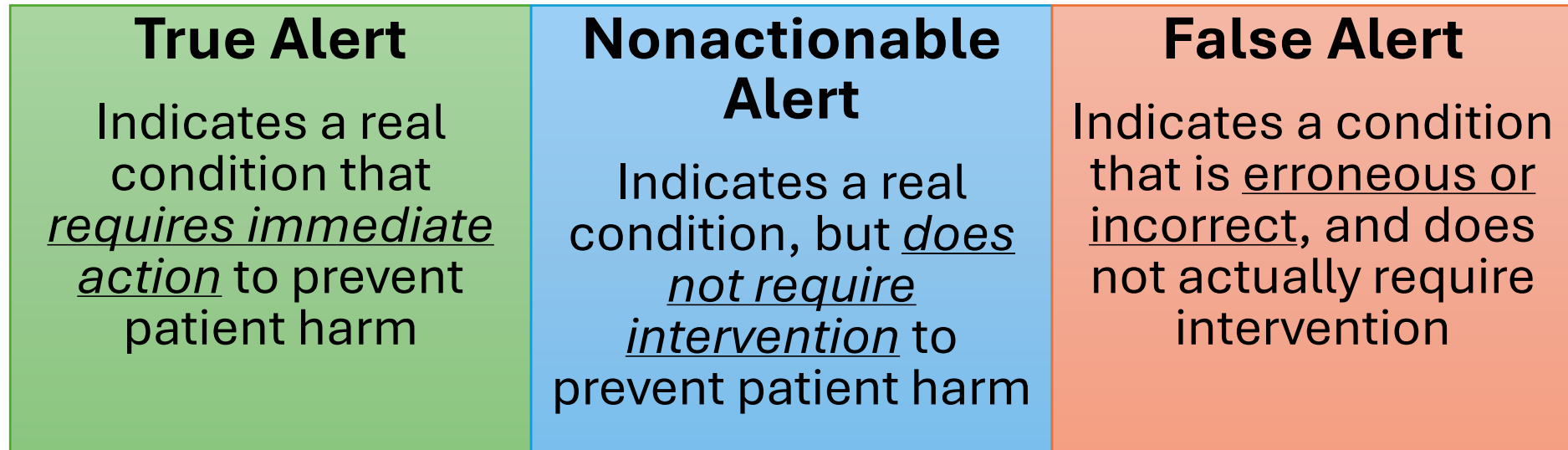
Data point 2  
**92% alert override rate**

- This figure “isn’t perfect because there are a number of alerts that are just informative and have an OK button, and in those cases, clicking the OK...it’s still counted as [being] overridden.”
- There is **no role to get override rate for actionable alerts** (excluding informative alerts)

Impact:

- Safety
- Cost/Productivity

# Alert Types and Definitions



**Nuisance Alert**: A false or nonactionable alert

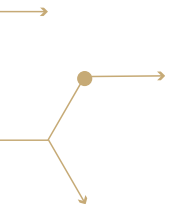
What's an acceptable "Nuisance Alert" rate?

- Depends on context
- Above **~10%** can measurably degrade human performance
- Above **~10-30%** is often worse than having all alerts disabled
- Lab testing has stringent requirements; **<1%** is common

# Alert Types and Definitions

<b>True Alert</b> Indicates a real condition that <i>requires immediate action</i> to prevent patient harm	<b>Nonactionable Alert</b> Indicates a real condition, but <i>does not require intervention</i> to prevent patient harm	<b>False Alert</b> Indicates a condition that is erroneous or incorrect
---	--	--

<b><u>True Alert Rate</u></b> CDSS: ~8-15%	<b><u>Nuisance Alert Rate</u></b> CDSS: ~85-92%
---	--



# **CDSS Management Processes:**

How are CDSS alerts and their clinical impact managed?

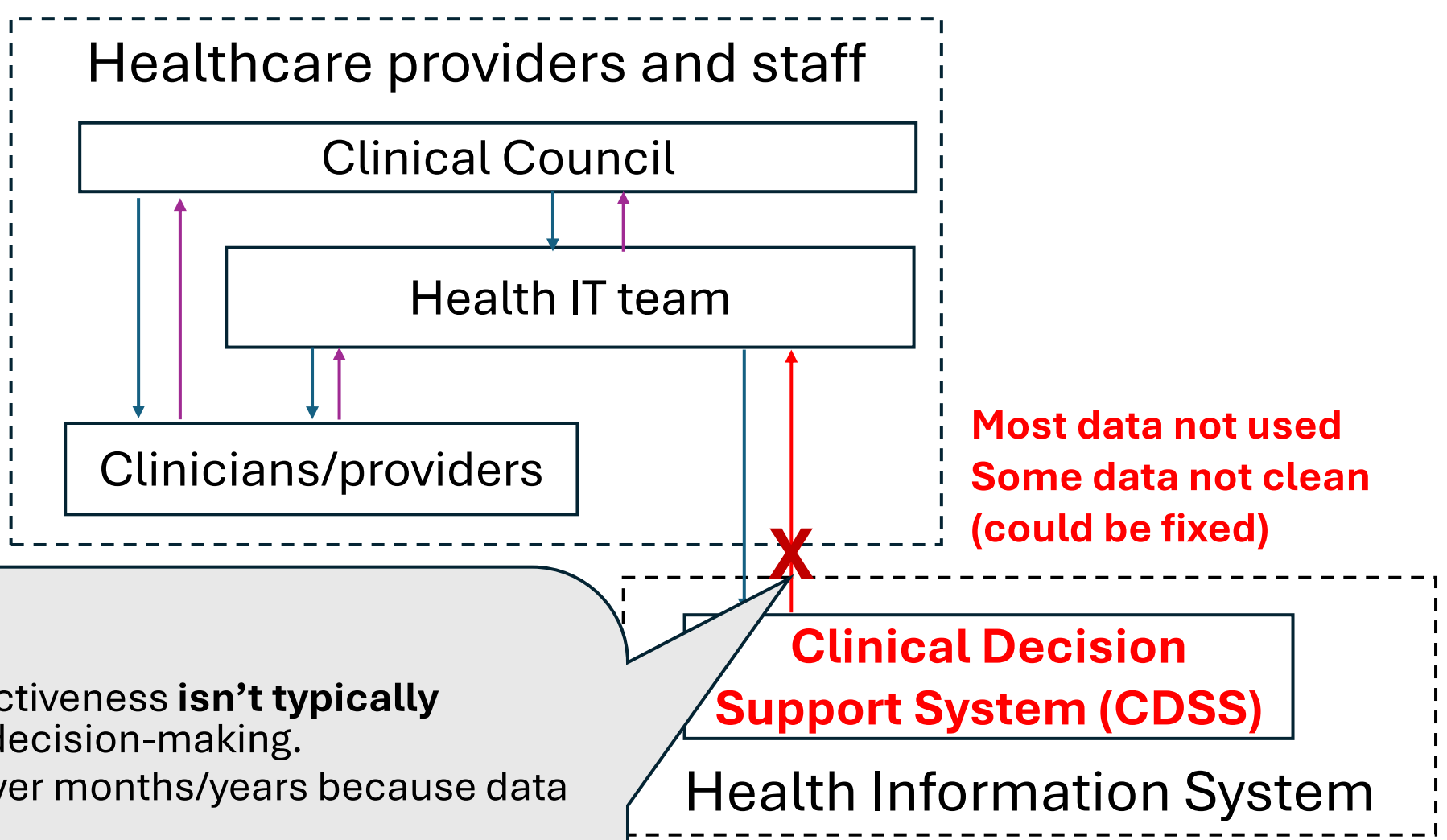
# Systemic Factors

Delayed / Missing Feedback:  
Model Rules

Lack of Customization: Model Rules

Reactive vs. Proactive

Unused & Missing Feedback



## Rule performance data:

- Data about clinical effectiveness **isn't typically monitored** or used for decision-making.
- **Trends are unknown** over months/years because data is periodically deleted
- **No group to monitor** current false alert rate
- **No group to reduce** future false alerts
- **No group to report** false alert statistics to other groups
- **No group to examine other indications** of rule clinical effectiveness

# Systemic Factors

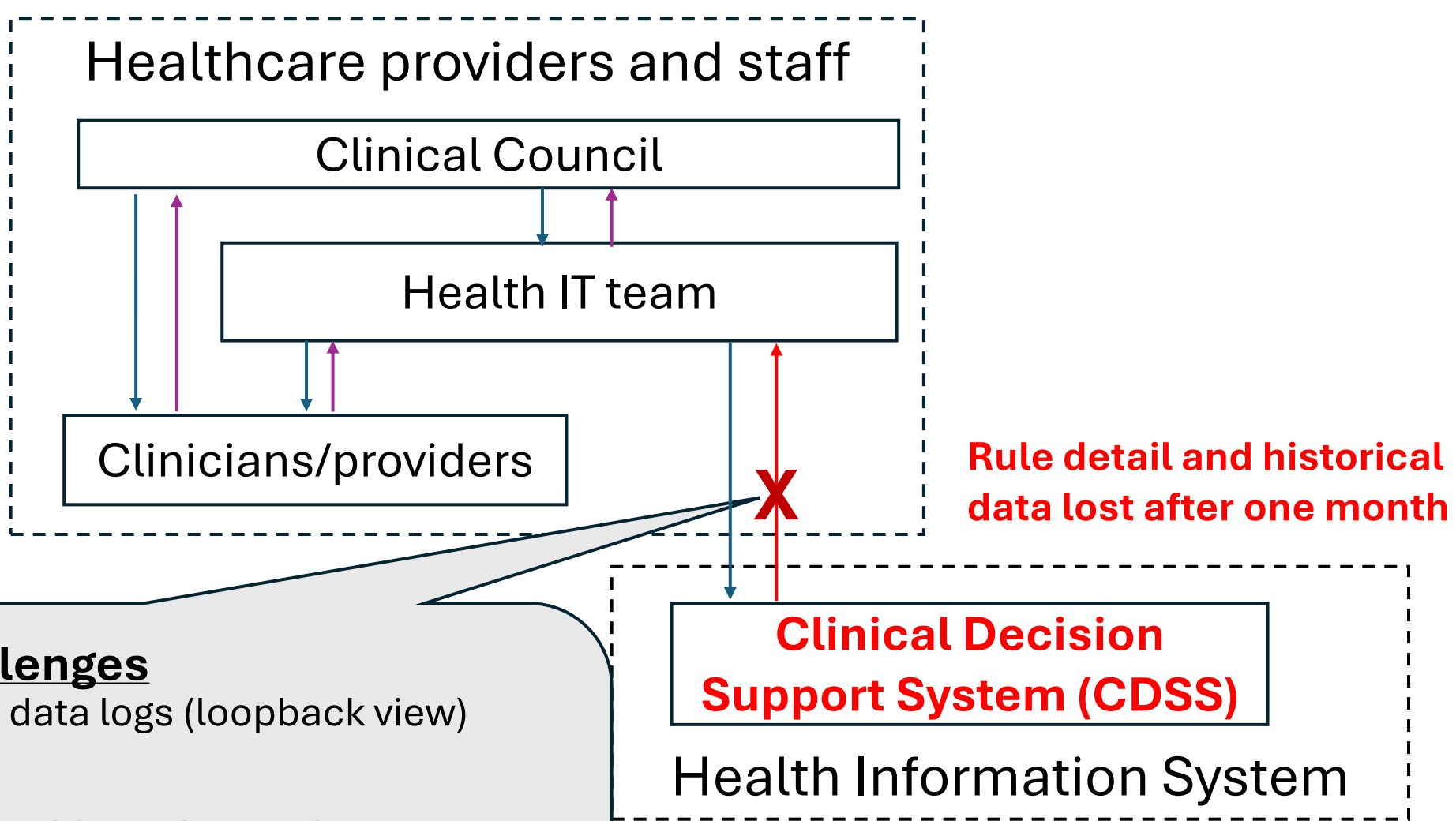
Delayed / Missing Feedback:  
Model Rules

Lack of Customization: Model  
Rules

Reactive vs. Proactive

Unused & Missing Feedback

Resource Constraints



## Diagnostic data challenges

**Logs are temporary:** Rule data logs (loopback view) disappear after 30 days.

### **No diagnostic archives**

- Difficult to diagnose a problem after 30 days
- Difficult to answer: "Has this malfunctioned before?" "How often?"

**Minimal records saved:** The only persistent data is a simple log of popup events that occurred (not other events, not diagnostic info)

# Systemic Factors

Delayed / Missing Feedback:  
Model Rules

Lack of Customization: Model  
Rules

Reactive vs. Proactive

Unused & Missing Feedback

Resource Constraints

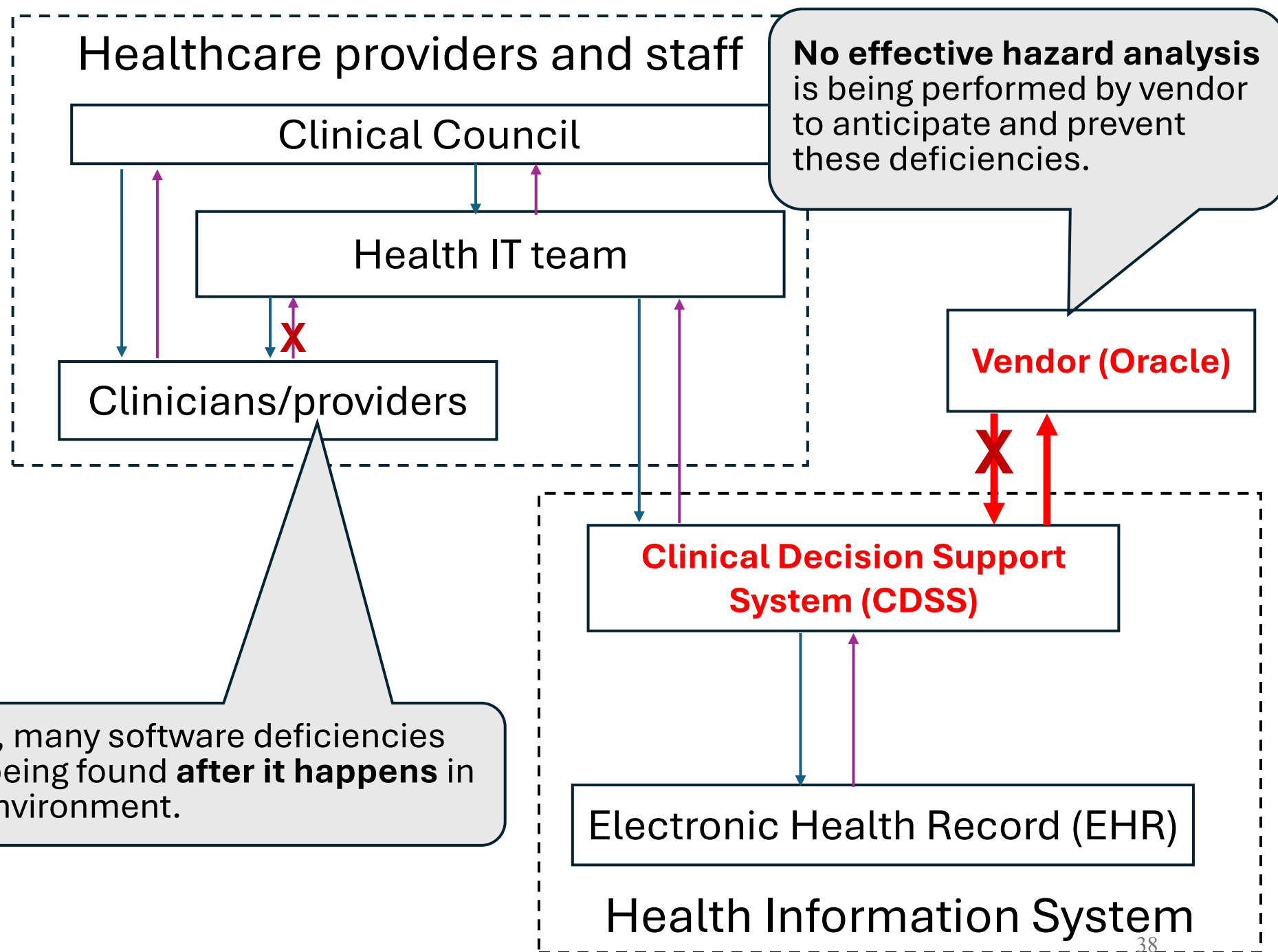
Competing Priorities

Lack of Rule Ownership

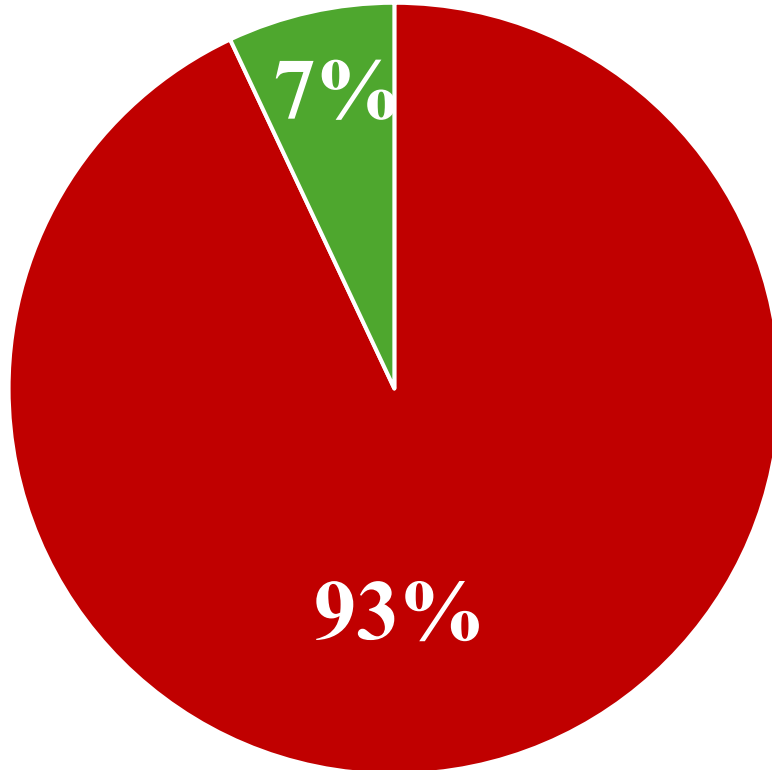
Missing or Inaccessible  
Feedback: Rule Intent  
Rationale

Inadequate Learning From  
Root Cause Analysis (RCA)

Currently, many software deficiencies are only being found **after it happens** in clinical environment.



## CDSS Errors & Malfunctions

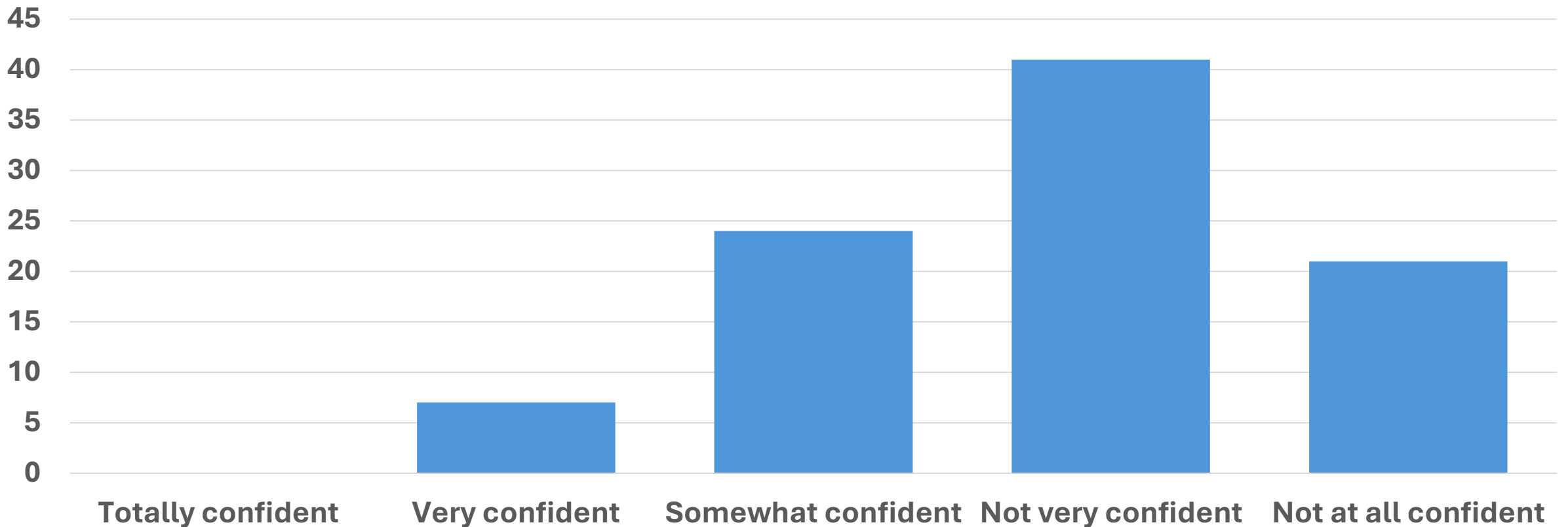


93% of Chief Medical Information Officers (CMIOs) report experiencing at least one CDSS malfunction in the last year

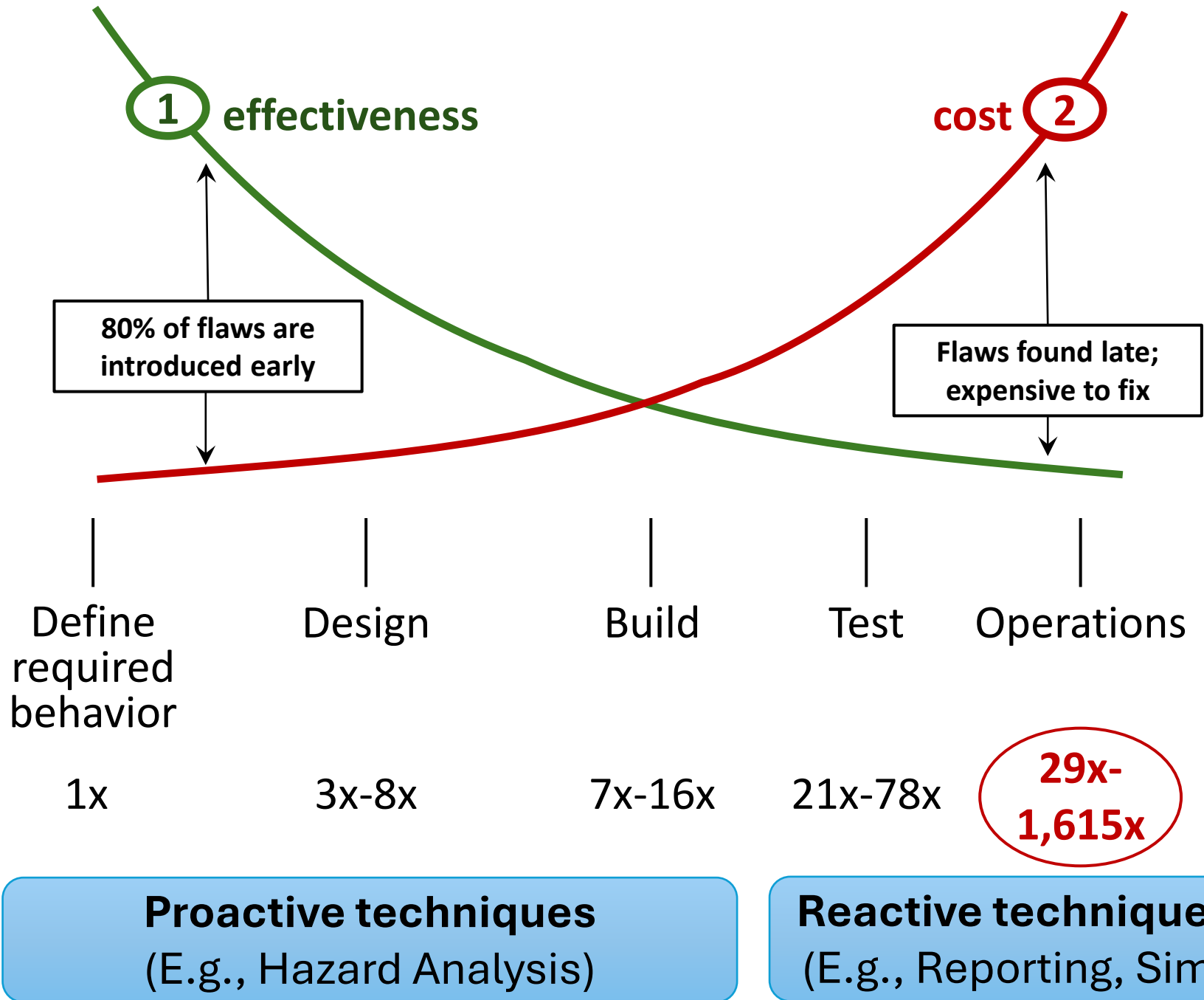
Most CDSS errors are overlooked during rule development, design, and testing.

Most errors are only caught in operation when users voluntarily decide to submit a ticket.

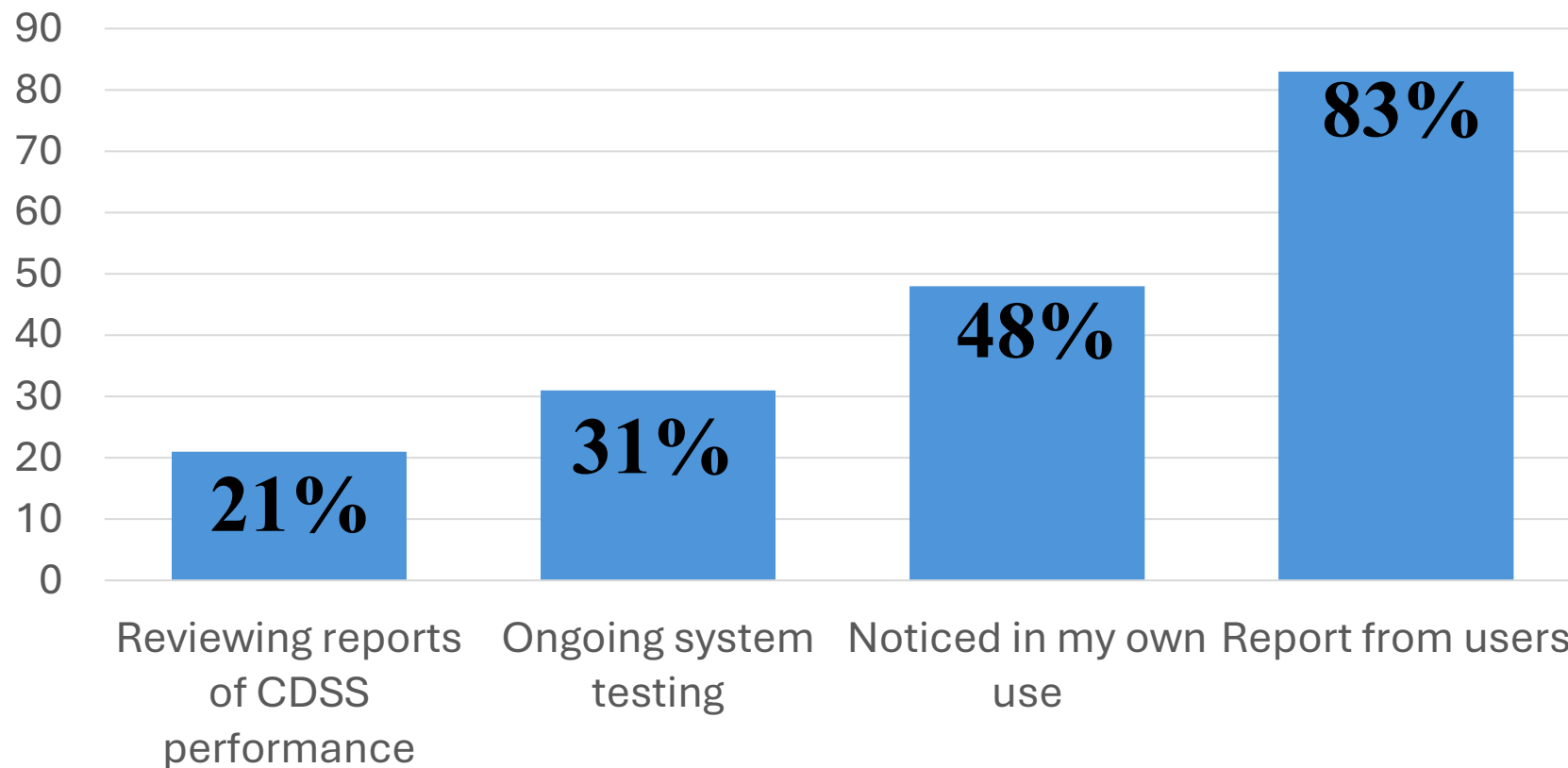
# How confident are you that your existing processes are sufficient to prevent or detect all CDSS malfunctions before they reach the user?



# System Engineering: Cost Savings

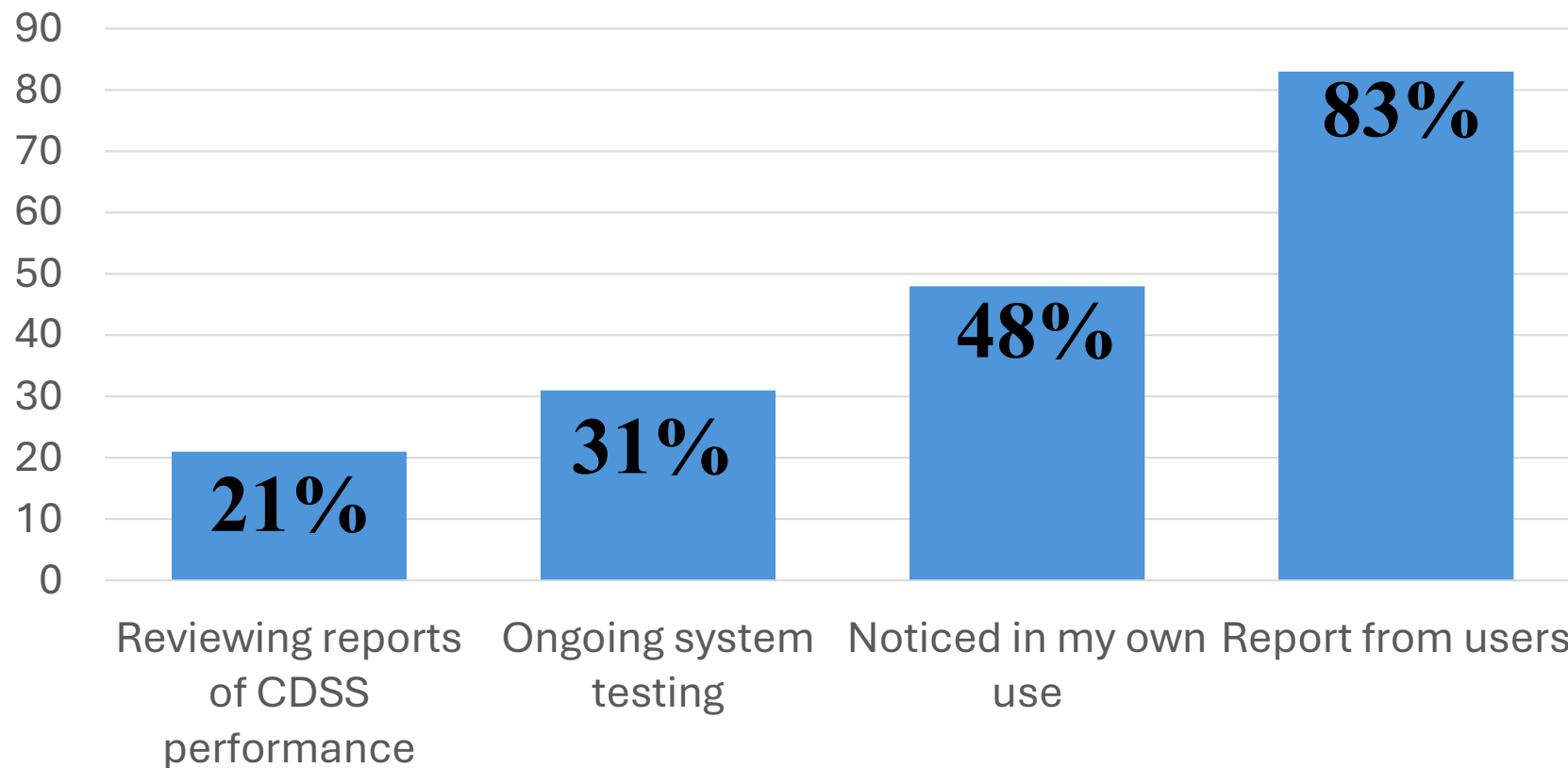


# How are CDSS malfunctions found today?



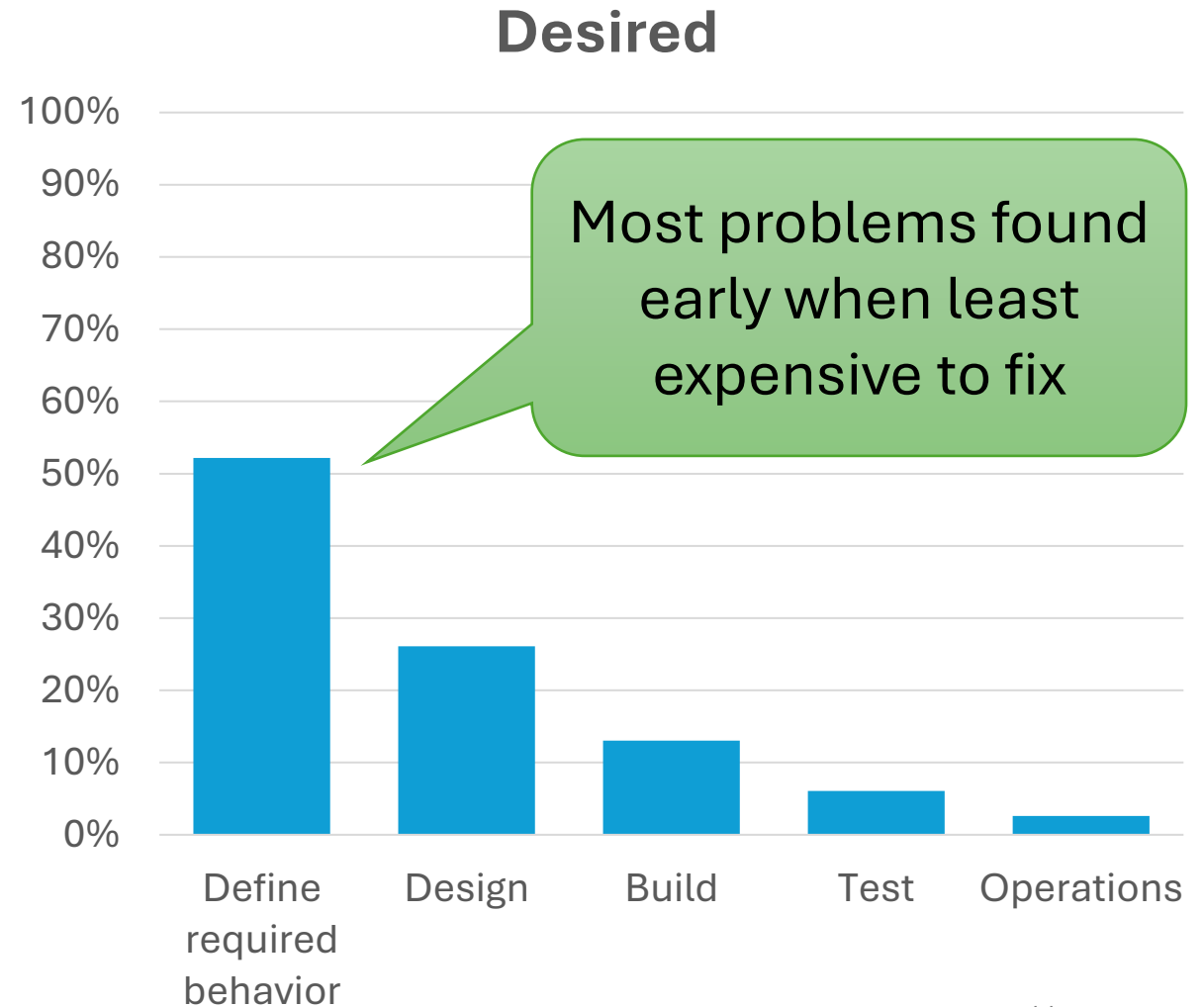
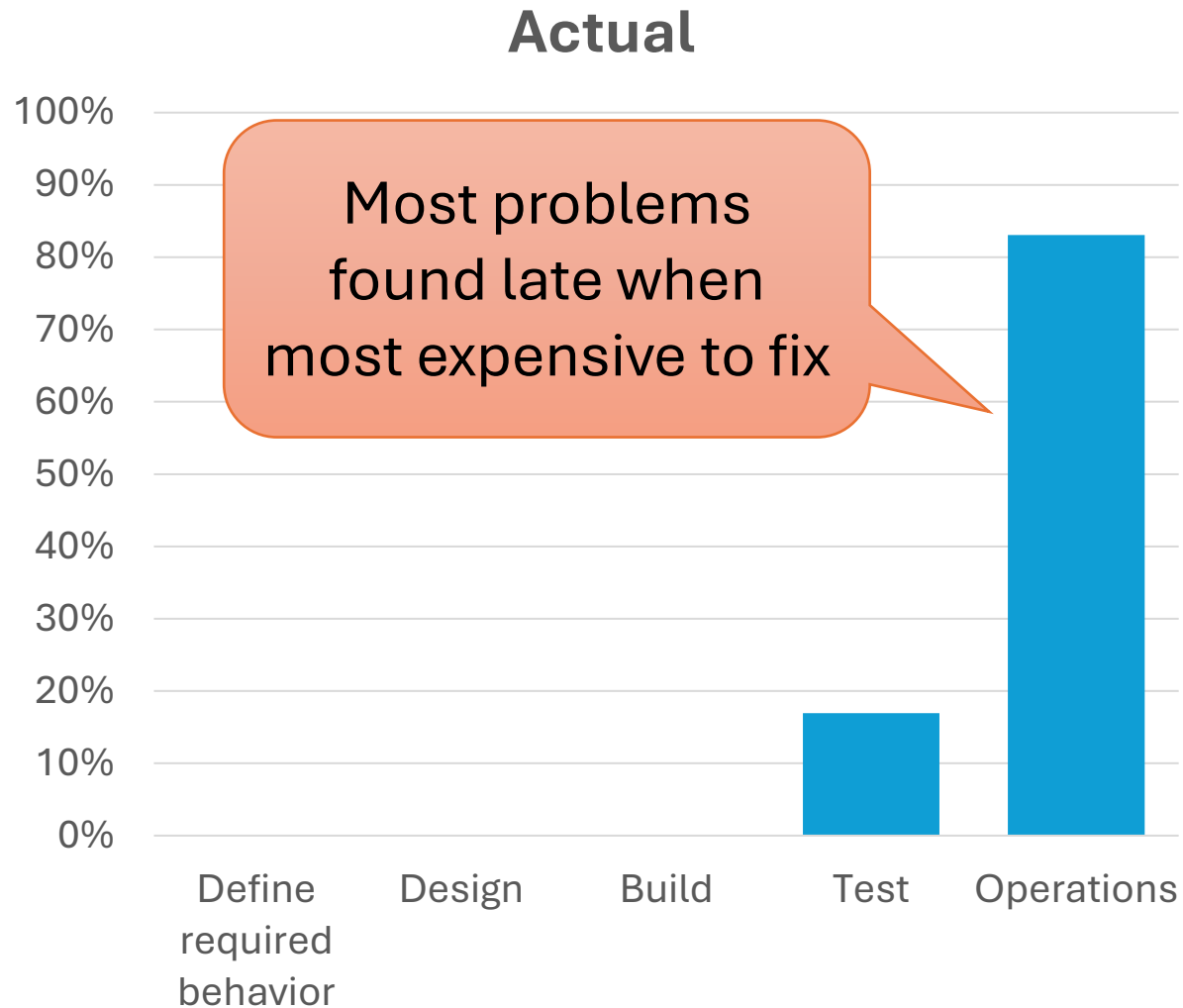
The vast majority of CDSS malfunctions are being **overlooked by testing** and existing CDSS performance metrics

# How are CDSS malfunctions found today?



Voluntary reports from **thousands of clinical staff** only amount to about twice the number independently encountered by a **single expert user** (CMIO)

# Preventing CDSS Errors: Actual vs. Desired



# Systemic Factors

Delayed / Missing Feedback: Model Rules

Lack of Customization: Model Rules

Reactive vs. Proactive

Unused & Missing Feedback

Resource Constraints

Competing Priorities

Lack of Rule Ownership

Missing or Inaccessible Feedback: Rule Intent and Rationale

Inadequate Learning From Root Cause Analysis (RCA)

## Healthcare providers and staff

- There appears to be **no government requirement for effective hazard analysis** by the vendor to prevent these software deficiencies.
  - Hazard analysis **anticipates deficiencies before they occur in operation** and requires proactive measures to prevent or mitigate them.
- Hazard analysis is a **standard practice in all safety-critical industries**.
  - Why? Hazard analysis is **far less costly** than waiting for a flaw to be discovered in operation.

Government contracting



Vendor (Oracle)

Clinical Decision Support System (CDSS)

Electronic Health Record (EHR)

Health Information System

# Systemic Factors

Delayed / Missing Feedback: Model Rules

Lack of Customization: Model Rules

Reactive vs. Proactive

Unused & Missing Feedback

Resource Constraints

Competing Priorities

Lack of Rule Ownership

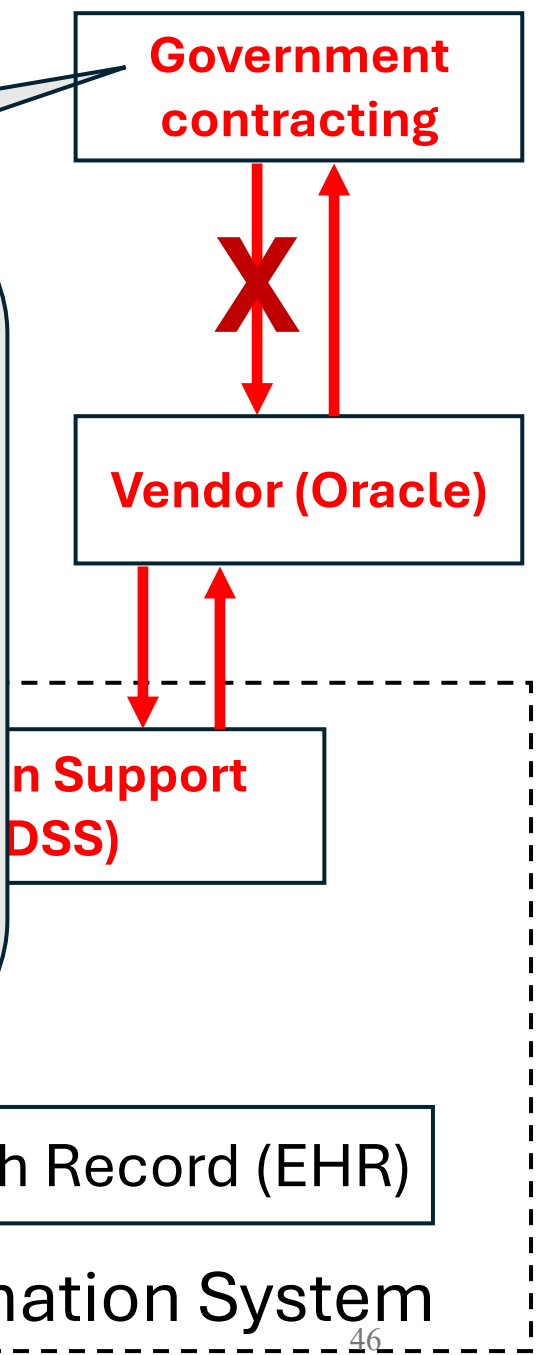
Missing or Inaccessible Feedback: Rule Intent and Rationale

Inadequate Learning From Root Cause Analysis (RCA)

Healthcare providers and staff

Clinical Council

- There appears to be **no requirement for effective Human Factors analysis** by the vendor to prevent these deficiencies.
- **Human Factors (HF) analysis is a standard practice** in all safety-critical industries.
  - Why? Software without HF consideration **tends to increase** human workload, confusion, etc.
  - HF analysis addresses problems like **EHR alert fatigue** and high false-positive rates
  - HF analysis produces clear requirements to limit these well-known factors. They **must be measured, monitored, and fixed** in operation.



# Systemic Factors

Delayed / Missing Feedback: Model Rules

Lack of Customization: Model Rules

Reactive vs. Proactive

Unused & Missing Feedback

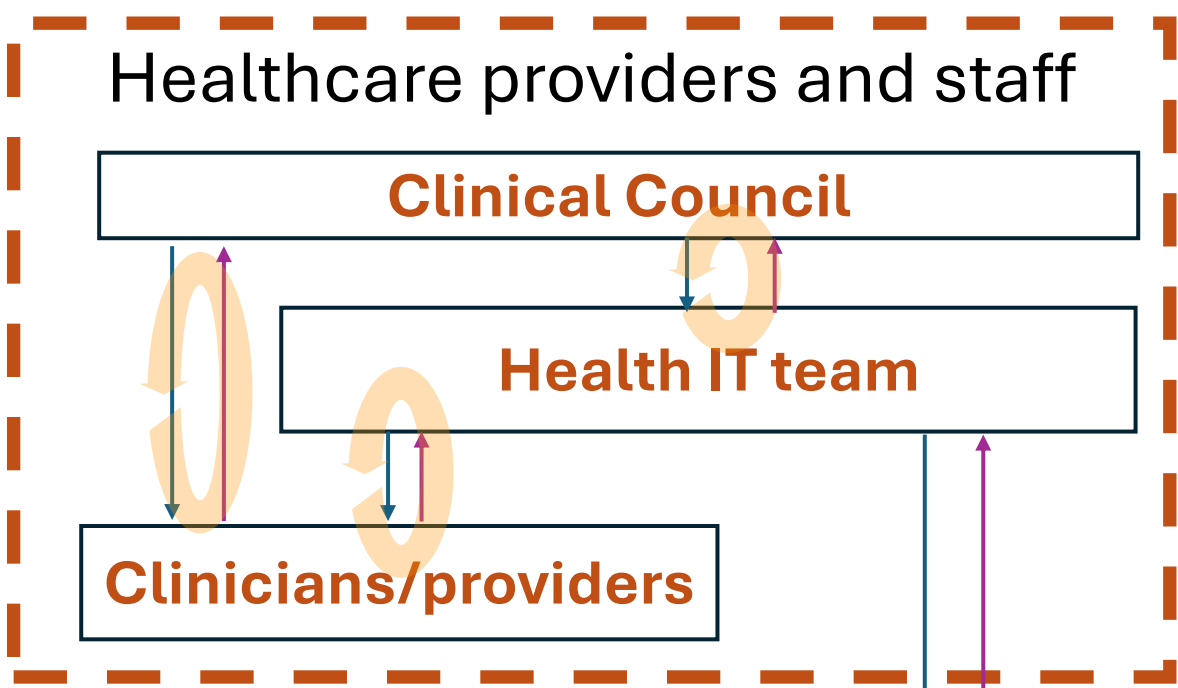
Resource Constraints

Competing Priorities

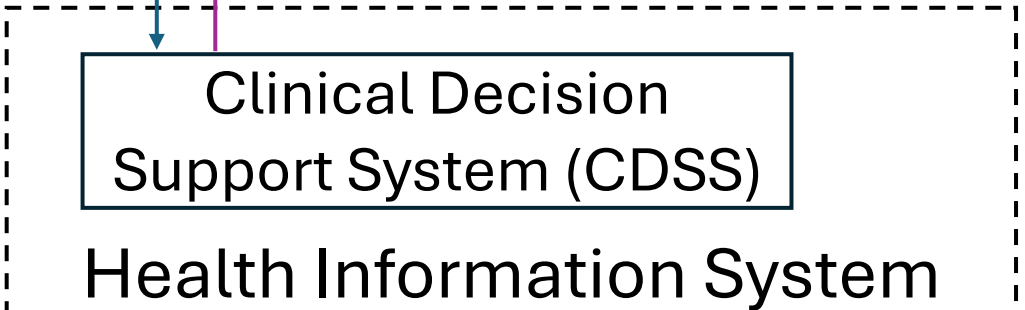
Lack of Rule Ownership

Missing / Inaccessible Feedback: Rule Intent and Rationale

Inadequate Learning From Root Cause Analysis (RCA)



*“One of the big things we struggle with is [...] finding ownership [...], essentially a clinical community that will be responsible for updates”*



**No Clinical Rule Ownership** (Missing Responsibility)

- No **tracking of changes**, such as clinical guidelines, that affect rules
- No continuous **monitoring of clinical relevance**, periodic reassessment
- No systematic checks of **rule effectiveness**; reliant on users submitting tickets
  - E.g., false-positive and false-negative rates
- No checks for **unintended consequences** after implementation

# Systemic Factors

Delayed / Missing Feedback:  
Model Rules

Lack of Customization: Model  
Rules

Reactive vs. Proactive

Unused & Missing Feedback

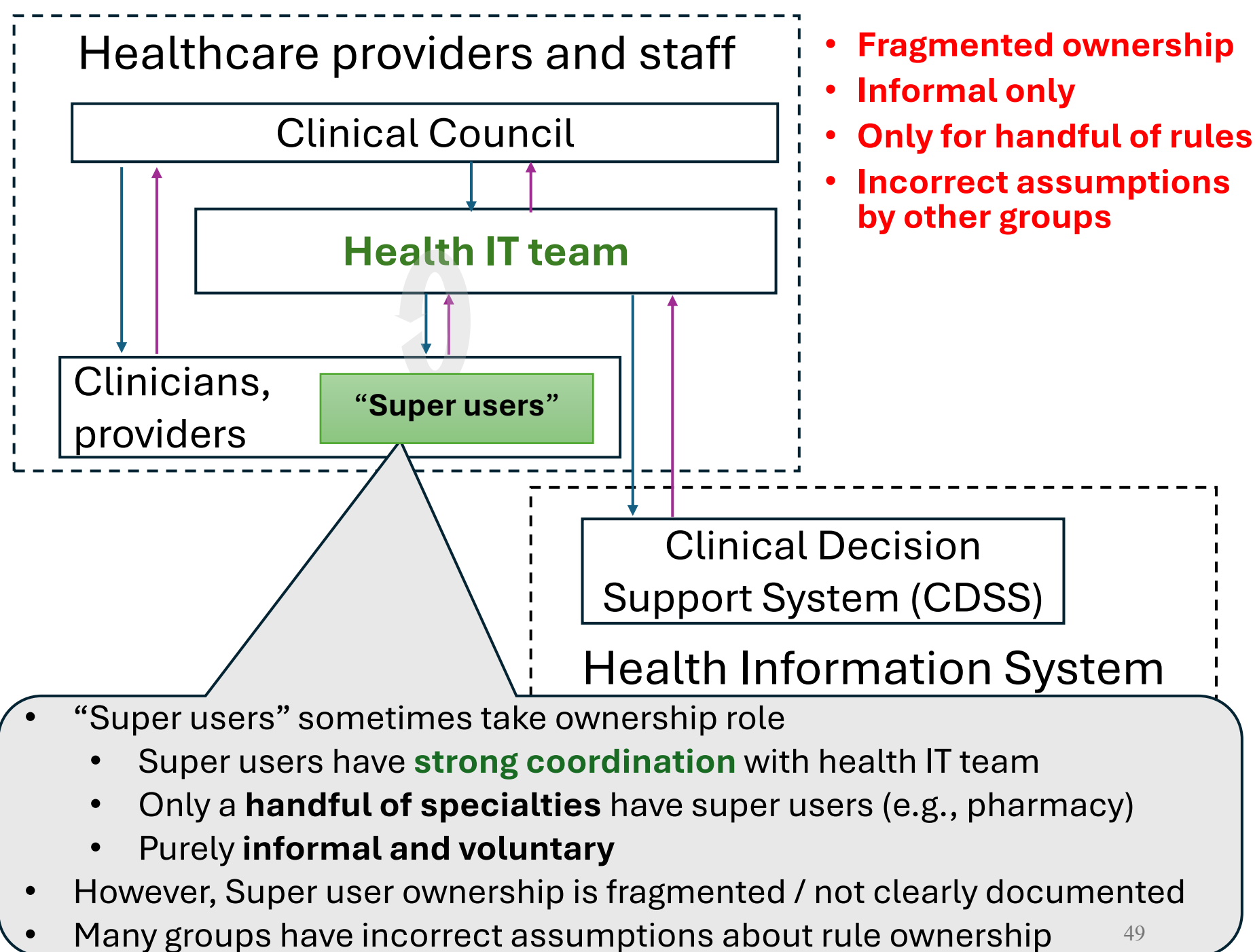
Resource Constraints

Competing Priorities

Lack of Rule Ownership

Missing / Inaccessible  
Feedback: Rule Intent and  
Rationale

Inadequate Learning From  
Root Cause Analysis (RCA)



# Systemic Factors

Delayed / Missing Feedback:  
Model Rules

Lack of Customization: Model  
Rules

Reactive vs. Proactive

Unused & Missing Feedback

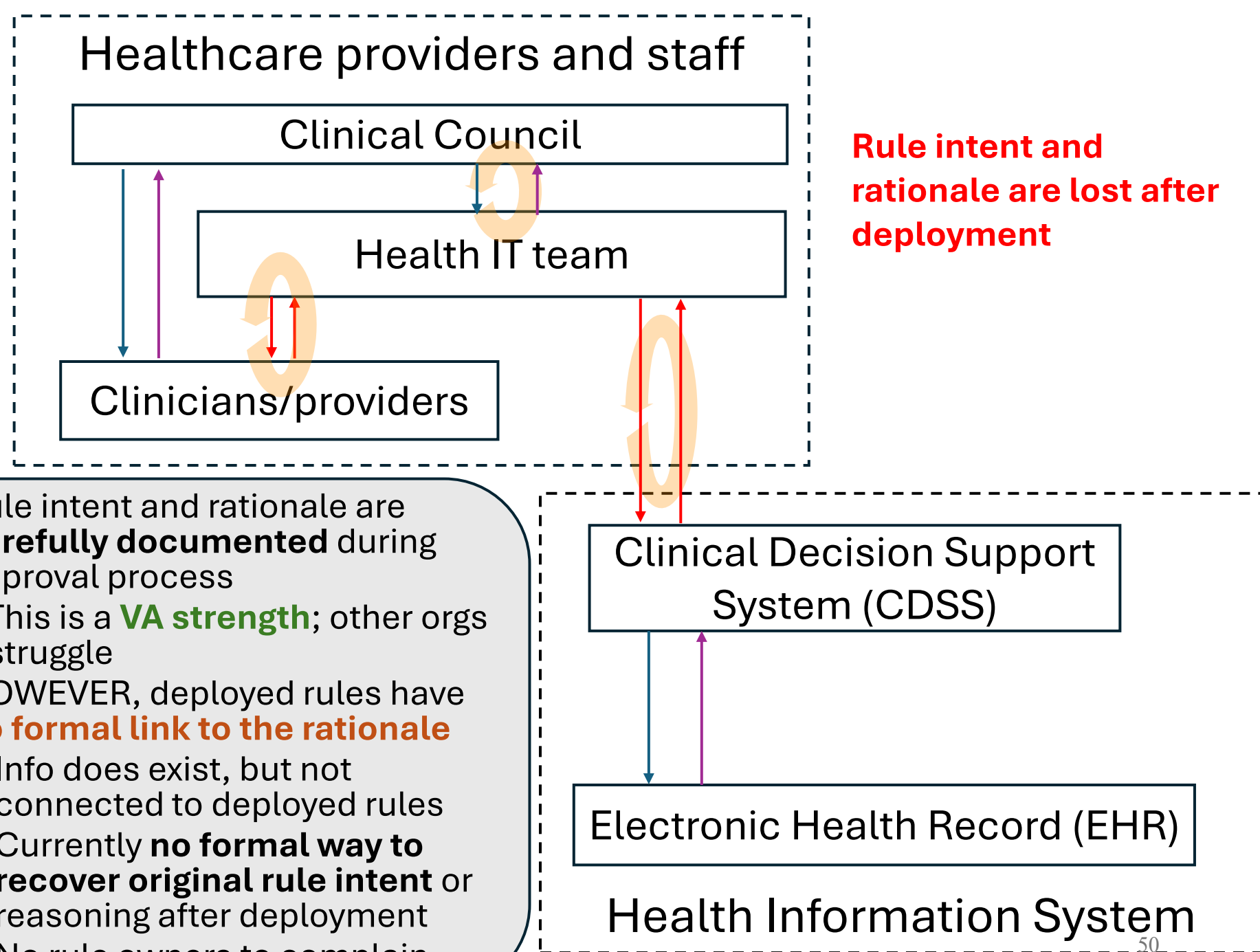
Resource Constraints

Competing Priorities

Lack of Rule Ownership

Missing / Inaccessible  
Feedback: Rule Intent and  
Rationale

Inadequate Learning From  
Root Cause Analysis (RCA)



# Systemic Factors

Delayed / Missing Feedback: Model Rules

Lack of Customization: Model Rules

Reactive vs. Proactive

Unused & Missing Feedback

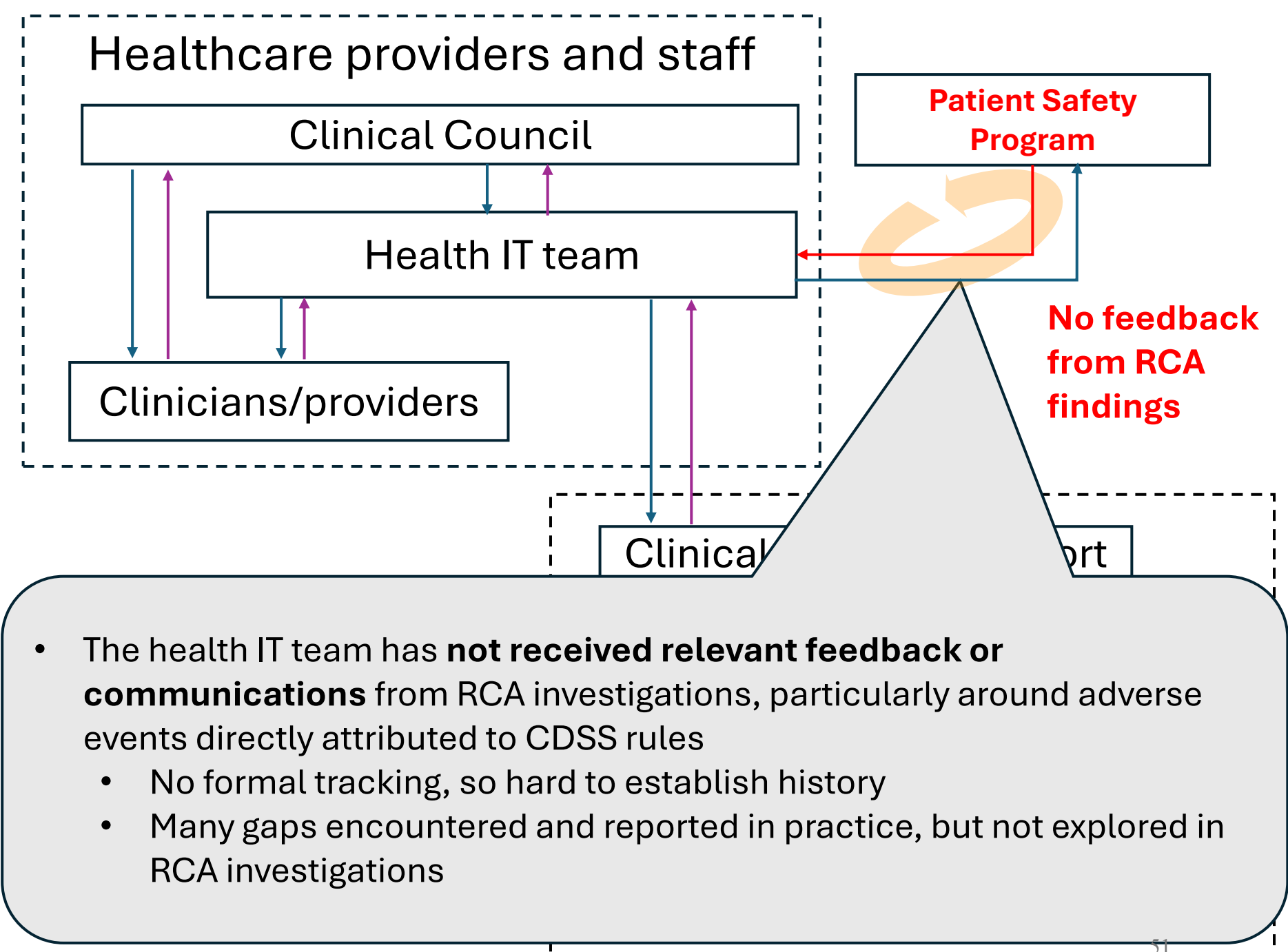
Resource Constraints

Competing Priorities

Lack of Rule Ownership

Missing / Inaccessible Feedback: Rule Intent and Rationale

Inadequate Learning From Root Cause Analysis (RCA)



# Systemic Factors

Delayed / Missing Feedback:  
Model Rules

Lack of Customization: Model Rules

Reactive vs. Proactive

Unused & Missing Feedback

Resource Constraints

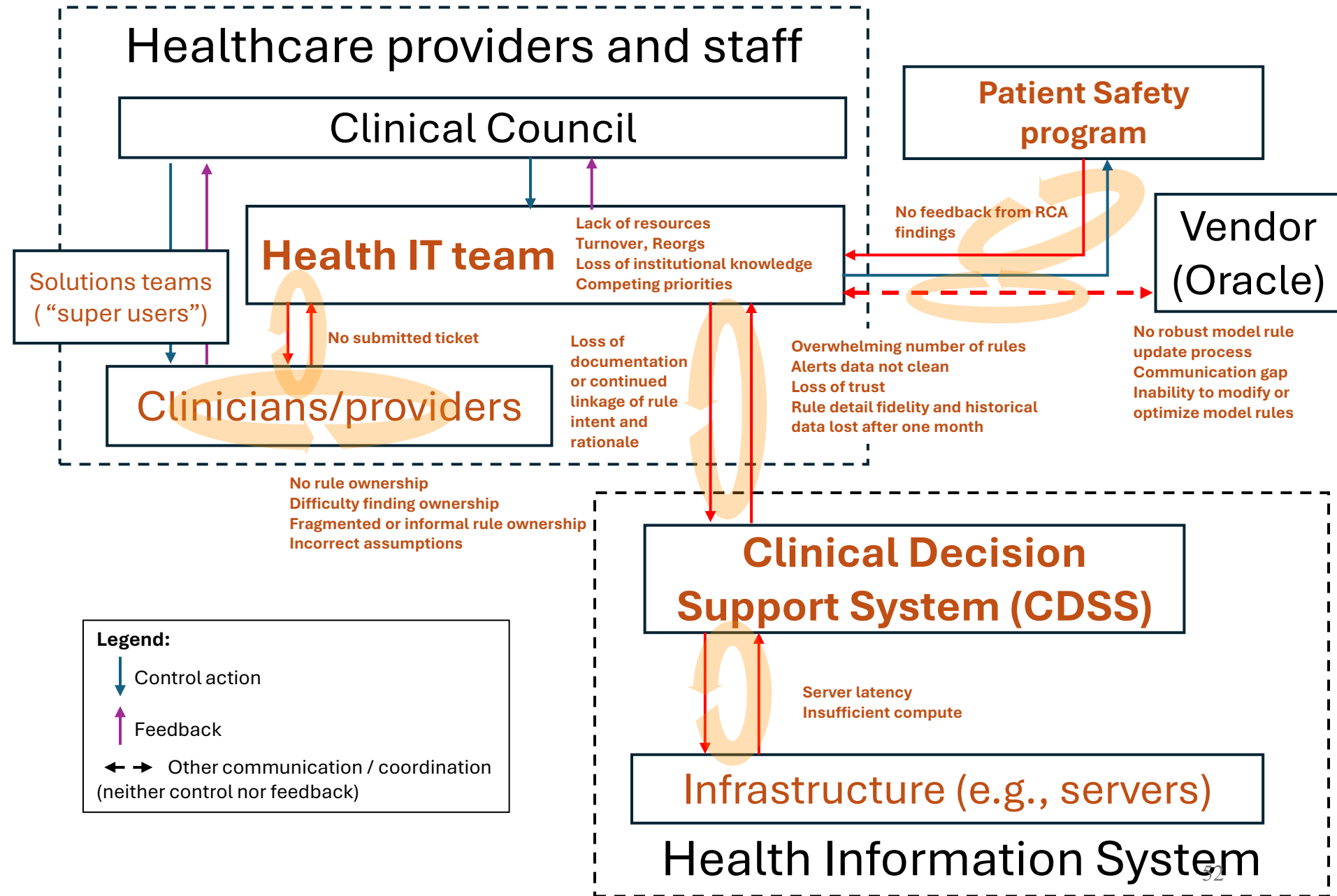
Competing Priorities

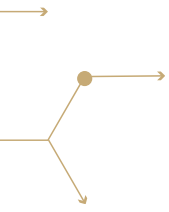
Lack of Rule Ownership

Missing / Inaccessible  
Feedback: Rule Intent and  
Rationale

Inadequate Learning From  
Root Cause Analysis (RCA)

# Opportunities to Strengthen SMS





# Recommendations

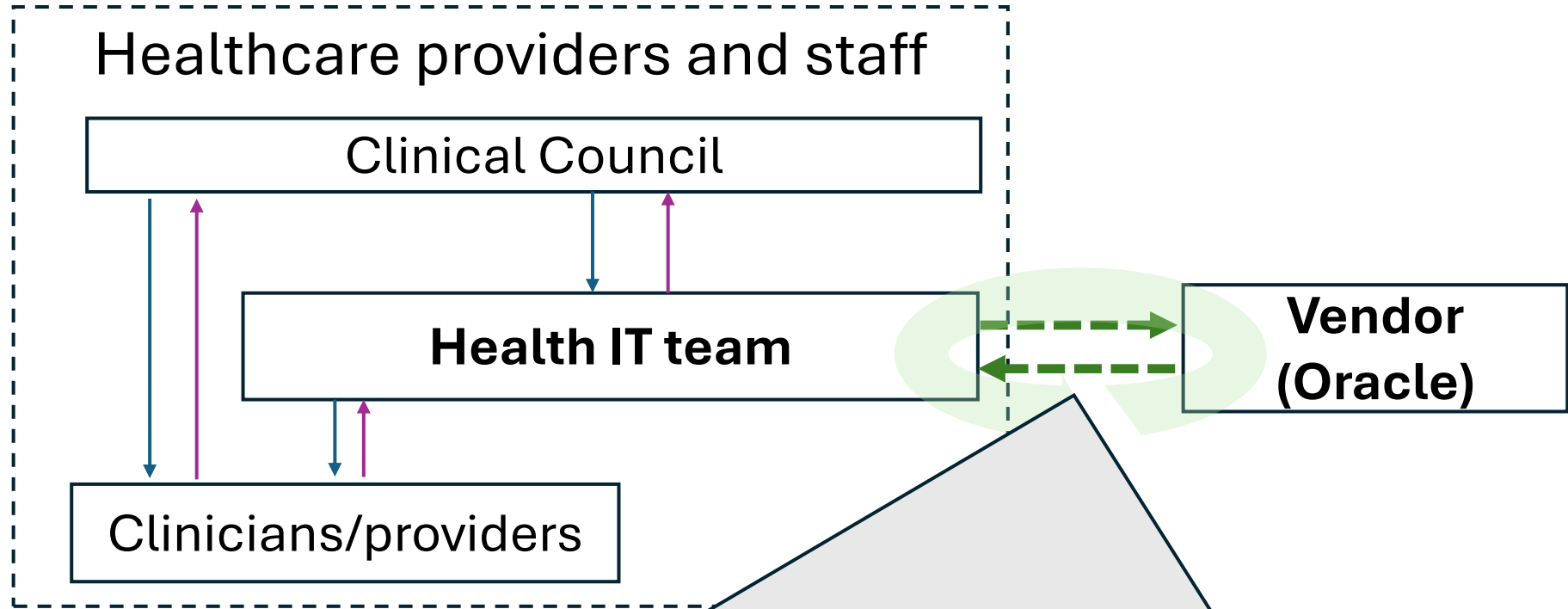
# Opportunities

Improve model rule coordination with vendor

Establish clear rule ownership

Incorporate strong proactive measures

Add feedback currently missing



## Improve Model Rule Coordination with Vendor

- Investigate the reasons for the communication gaps
  - Why aren't the relevant VA teams always notified of an update?
  - Is there a technology solution for automatic notification?
  - Is it better for the Health IT team to contact vendor to get status instead of waiting?
  - Is there a better way for vendor to roll out updates?
- Establish a simple and easy-to-use process for the vendor to send model rule changes
- Document and commit to a clear process for vendor updates

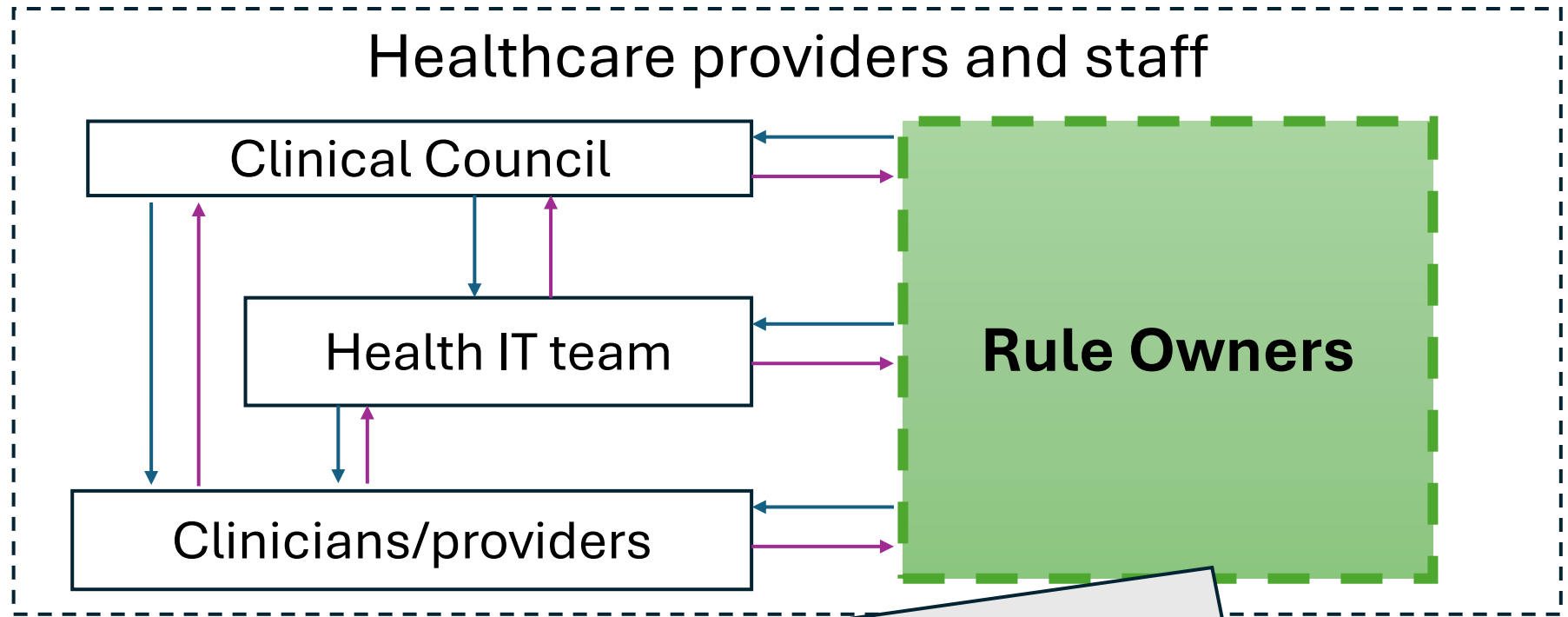
# Opportunities

Improve model rule coordination with vendor

Establish clear rule ownership

Incorporate strong proactive measures

Add feedback currently missing



## Establish clinical rule ownership role

Assign responsibilities:

- Monitor relevant **clinical guidelines** for changes that require rule updates
- Monitor **clinical effectiveness** of rules
  - Identify measures to **evaluate clinical impact**, like override rates and false-positive rates
  - If a rule is ineffective, take appropriate response
- Monitor **unintended consequences** of rules
- Clearly document and **link rule metadata**: rule owners, change history, rationale, etc.

# Opportunities

Improve model rule coordination with vendor

Establish clear rule ownership

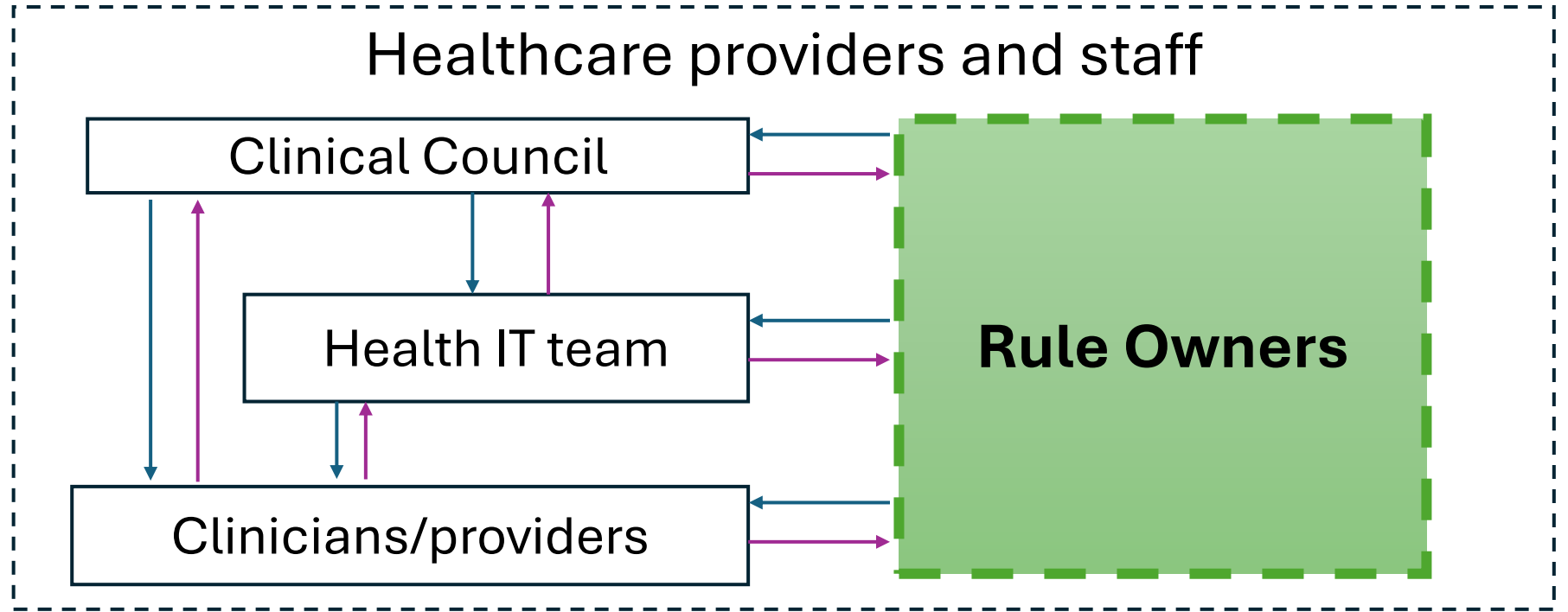
Incorporate strong proactive measures

Add feedback currently missing

Match resources to responsibilities

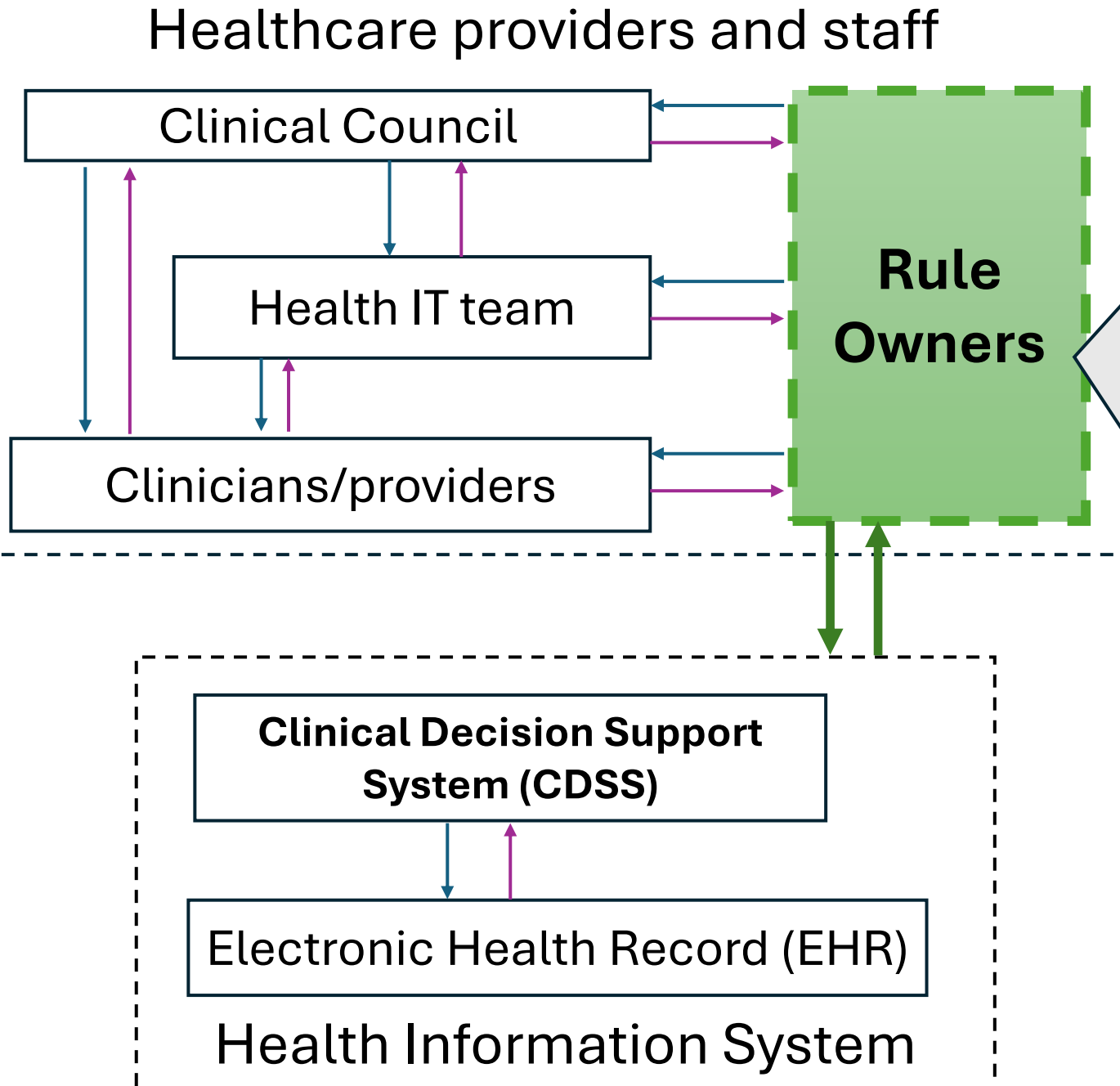
Establish clear priorities

Improve Learning From Root Cause Analysis (RCA)



Rule owners:  
A good role for clinical informaticists?

# Recommendations



- 1) Establish clinical ownership role
- 2) Improve feedback to manage rules
  - Positive clinical impact of rules
    - **How often is a rule followed/ accepted by clinicians?**
  - Negative clinical impact of rules
    - **How often is a rule ignored/ rejected by clinicians?**
    - **Unintended clinical consequences** of rules
  - Clinical basis of the rule
  - Clinical data that triggers the rule
    - E.g., PIQI
  - The rationale and intent of the rule
  - Changelog (history) of the rule

# Examples of improving feedback

## Positive clinical impact of rules



- **How often is a rule followed/accepted by clinicians?**
- When creating a rule, identify how to track if clinicians accept or follow the rule's recommendation (e.g., they click "Accept", they submit an order that matches the recommended action, they cancel a draft order after a rule warns about a conflict, etc.)
- Monitor the **clinical effectiveness** of rules.
  - a) When creating a rule, specify the expected changes in clinical outcomes if the rule is effective.
  - b) Enable automatic checks to verify if rules are achieving their clinical goals over time.
  - c) Identify rules that aren't working and evaluate whether to correct or remove the rule.

## Negative clinical impact of rules



- **How often is a rule ignored/rejected by clinicians?**
- When creating a rule, define a means to determine whether clinicians ignored/rejected a rule.
  - a) Ideally, design all rules to enable simple feedback about whether it was helpful or not (e.g., "Accept" or "Ignore" button).
  - b) Avoid rules that only provide "OK" button. In addition to eliminating feedback, it reduces critical thinking and tends to trigger a habitual response (making the rule less effective).
  - c) Alternative indicators (e.g., a draft order is submitted as-is after a rule warns about a conflict).
- When a rule is ignored/rejected, ask the clinician why. Provide this feedback to the rule owners.
- If a rule is not helping, then rule owners should evaluate whether to fix the rule or potentially remove it.
- **Unintended clinical consequences** of rules
  - Identify what changes in clinical outcomes would indicate a rule is not working as intended.
  - Set up automatic flagging of rules that aren't working as intended.
  - Review why the rule isn't working take action to fix

## Clinical basis of the rule



- Identify any clinical guidelines that the rule is based on
- Flag rules that need to be updated when clinical guidelines change
- Maintain a rule status showing which rules:
  - a) Require an update (we know the clinical guidelines have changed)
  - b) Do not require an update (we know the clinical guidelines have not changed)
  - c) Status unknown (we do not know if relevant clinical guidelines have changed)
- Report summary of rule status to leadership.

## Clinical data that triggers the rule



- Automatically identify the reliability of the data driving each rule (e.g., from **PIQI, Komet, DeX**).
- Before a new rule is approved, provide feedback on the quality of data that the rule would be based on.
- Enable checks to flag when future changes impact relevant data (e.g., drug id changes when generic becomes available).
- Identify which rules have not been re-evaluated since the data or data structure or data quality changed.

## The rationale and intent for the rule when it was created

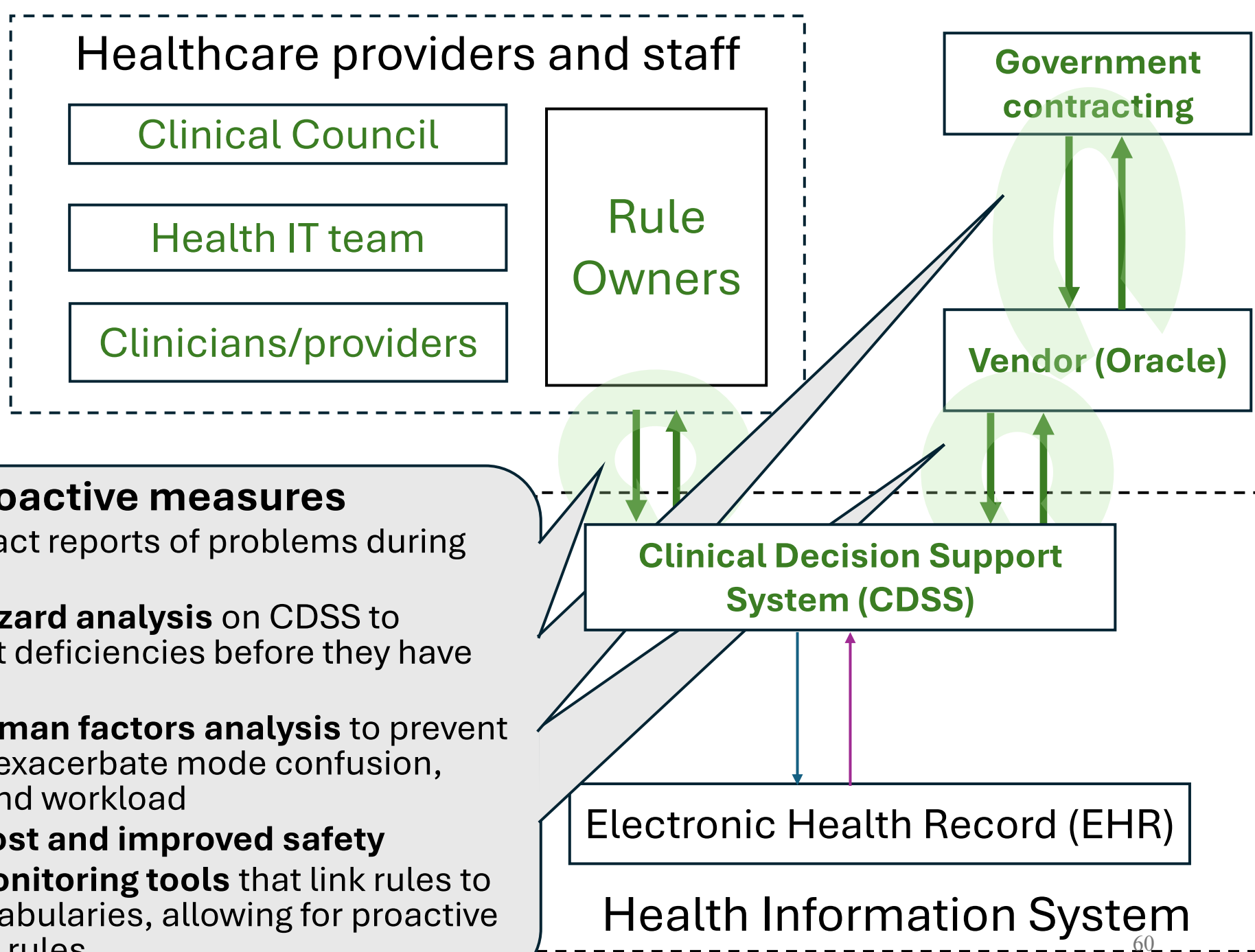
## Changelog/history of the rule

# Opportunities

Improve model rule coordination with vendor

Establish clear rule ownership

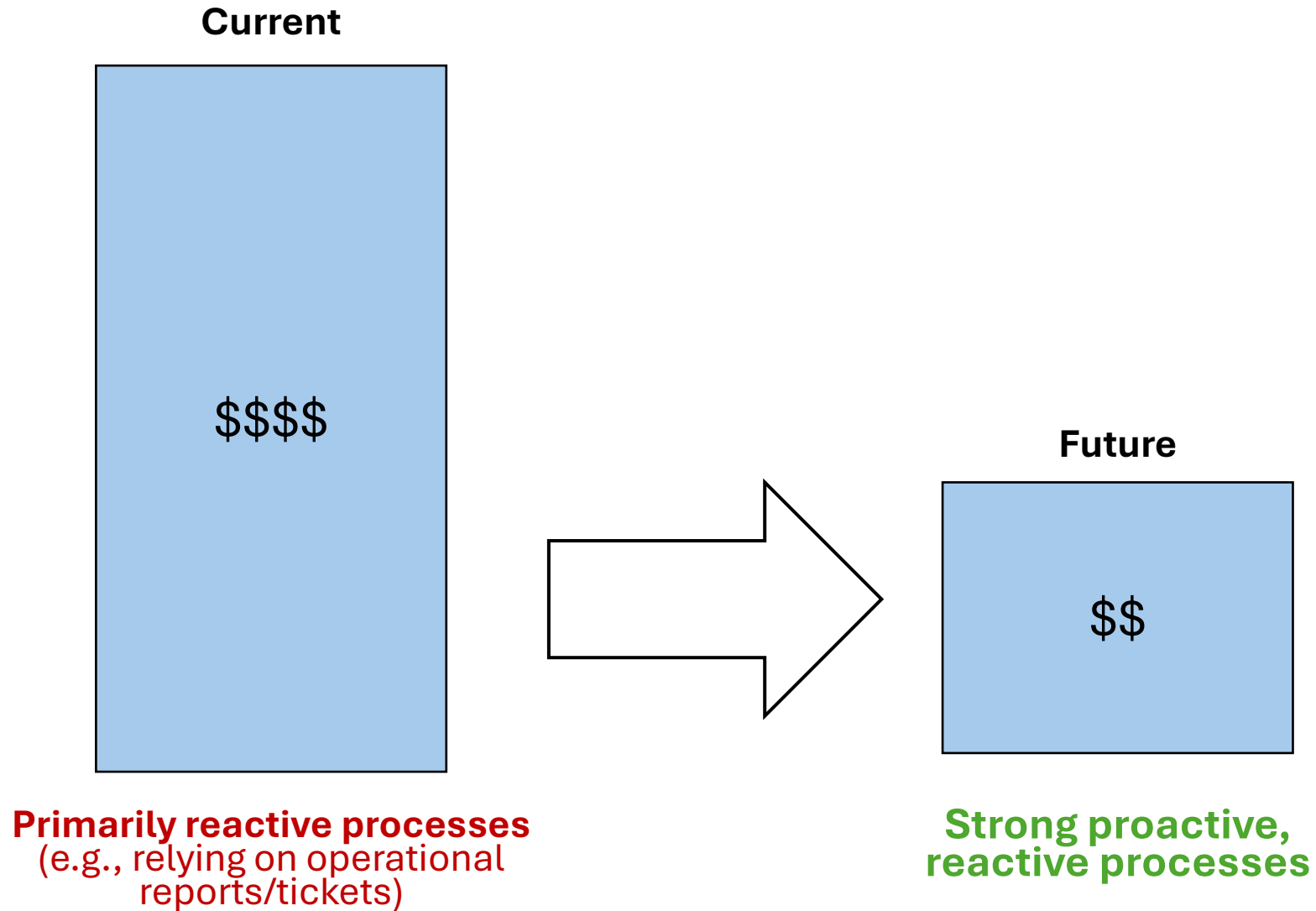
Incorporate strong proactive measures



## Incorporate strong proactive measures

- Move beyond after-the-fact reports of problems during clinical care
- Perform, or require, a **hazard analysis** on CDSS to **reduce cost** and prevent deficiencies before they have clinical impact
- Perform, or require, a **human factors analysis** to prevent software behaviors that exacerbate mode confusion, human error, usability, and workload
  - Impact: **Reduced cost and improved safety**
- Consider **automated monitoring tools** that link rules to source guidelines or vocabularies, allowing for proactive identification of affected rules.

# VA CDSS Current vs. Future State



# Recommendations

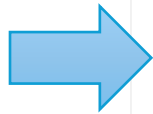
Systemic Factor	Recommendation
Model Rules: Delayed / Missing Feedback & Lack of Customization	<p>Strengthen vendor communication about model rule changes</p> <ul style="list-style-type: none"> <li>• Improve vendor processes to notify VA when model rules are updated.</li> <li>• Investigate the reasons for the communication gaps, and establish a simple and easy-to-use way for the vendor to notify of any model rule change</li> </ul>
Reactive vs. Proactive	<p>Improve proactive monitoring processes</p> <ul style="list-style-type: none"> <li>• Move beyond purely reactive change requests; develop systems that actively monitor for external changes (e.g., guideline updates, new drug codes, position changes).</li> <li>• Consider automated monitoring tools that link rules to source guidelines or vocabularies, allowing for proactive identification of affected rules.</li> </ul> <p>Strengthen coordination between sites, councils, and deployments</p> <ul style="list-style-type: none"> <li>• Prevent rule divergence across sites by improving coordination between councils, deployment teams, and DoD partners.</li> <li>• Avoid redundant or duplicate rules across VA sites.</li> </ul>
Unused & Missing Feedback	<p>Create ways to monitor rule effectiveness (e.g., override rates, false positives)</p> <ul style="list-style-type: none"> <li>• Analyze how often rules are fired and overridden to identify ineffective or burdensome rules.</li> <li>• Consider ways to collect feedback from clinicians about rule usefulness (E.g., “this rule was useful” button)</li> <li>• Use this feedback to guide rule refinement and reduce alert fatigue.</li> </ul> <p>Build tools to extract and analyze rule metadata</p> <ul style="list-style-type: none"> <li>• Continue expanding reporting tools that allow solution teams to proactively analyze rules (e.g., searching for drug codes, facilities, positions affected by rules).</li> <li>• This approach has already shown success for pharmacy teams and should be expanded.</li> </ul>

# Potential Recommendations

Systemic Factor	Recommendation
Resource Constraints	<p>Ensure that the health IT team is appropriately staffed and able to not only react to tickets and build rules based on tickets, but also to be able to monitor what's already in the system and prospectively.</p> <ul style="list-style-type: none"> <li>• Adequately staff team and hire and design team for cross-collaboration</li> </ul>
Lack of Rule Ownership & Competing Priorities	<p>Establish formal ownership for each rule</p> <ul style="list-style-type: none"> <li>• Assign responsibility to specific clinical teams or solution groups for monitoring and maintaining rules after implementation.</li> <li>• Ensure each rule has a clearly documented owner responsible for monitoring relevant clinical guideline changes.</li> </ul> <p>Improve tracking of organizational responsibility over time</p> <ul style="list-style-type: none"> <li>• Account for organizational changes that make rule ownership ambiguous over time.</li> <li>• Build a way to track ownership that survives reorganizations, staff turnover, and changing structures.</li> </ul>
Rule Intent and Rationale Not Available	<p>Improve rule metadata and linkage to rationale</p> <ul style="list-style-type: none"> <li>• Capture and store rule rationale, origin, clinical objective, source documents, and requestor in a structured, easily searchable format.</li> <li>• Link rule records directly to change request records (e.g., JIRA tickets)</li> </ul>
Inadequate Learning from Root Cause Analysis (RCA)	<p>Improve RCA integration with CDSS review</p> <ul style="list-style-type: none"> <li>• Ensure RCA investigations examine whether CDSS rules contributed to adverse events or could prevent similar events in the future.</li> </ul>

# Other Healthcare Examples with STAMP

Search for  
“health”



## Papers and Publications

[< Other materials available](#)

Search:

Note: This searches the text below and a few keywords. It does not search the content of the linked files.

Found 22 results

---

### Safety-III A Systems Approach to Safety and Resilience in Healthcare and Other Complex, Adaptive Systems

Jan 2026

Nancy Leveson

*MIT Whitepaper*

[▷ Show description](#)

[Paper](#) | [Google Scholar](#)

---

### System Safety for Health Information Technology

Sept 2025

John Thomas, Eugenia Kim, Polly Harrington, Nancy Leveson

*MIT Technical Report*

[▷ Show description](#)

[Paper](#) | [Google Scholar](#)

---

### A Safety Management System for Health Information Technology HIT

Sep 2025

Nancy Leveson, John Thomas, Stephen Powell, Abigail Williams, Alana

Keller

*MIT Technical Report*

[▷ Show description](#)

[Paper](#) | [Google Scholar](#)

---

### Challenges in Deploying the Oracle Cerner EHR and Potential Paths Forward

April 2025

Nancy Leveson, Stephen Powell

*MIT Technical Report*

[▷ Show description](#)

[Paper](#) | [Google Scholar](#)

---

### CAST of an Adverse Event Involving Laboratory Data

Sep 2024

Polly Harrington, John Thomas, Nancy Leveson, Stephen Powell, Alana Keller

*MIT Technical Report*

[▷ Show description](#)

[Paper](#) | [Google Scholar](#)

---

### System Safety of Over-the-Counter and Point-of-Care Testing

Jul 2024

Nancy Leveson, John Thomas, Polly Harrington, Rodrigo Rose

*MIT Technical Report*

[▷ Show description](#)

[Paper](#) | [Google Scholar](#)

---

### System Safety within Laboratory Data Exchanges Report

Sept 2023

Nancy Leveson, John Thomas, Polly Harrington, and Rodrigo Rose

*MIT/Synensys Technical Report to the FDA*

[▷ Show description](#)

[Paper](#) | [Google Scholar](#)

More:

[mit.edu/psas](https://mit.edu/psas)