Beyond Root Cause: AI-Assisted Learning from Incidents in COMPASS

Building a Learning Culture Through Systems Thinking, Contributing Causes Analysis, and Transparent Investigation in Multi-Agent Incident Response

> "By implying—even inadvertently—that a single root cause can be found, the term 'root cause analysis' promotes a flawed reductionist view. In complex systems, things go wrong because of multiple, interacting, contributing factors from across the system."

Document Type:	Research & Implementation Guide v2.0
Date:	November 14, 2025
System:	COMPASS - Comprehensive Observability Multi-Agent Platform
Focus:	Learning Culture & Contributing Causes Analysis
Philosophy:	Systems Thinking & Prevention over Blame

Executive Summary

This document presents a paradigm shift in how COMPASS approaches incident investigation and post-mortem generation. Moving beyond the flawed concept of "root cause analysis," we embrace a systems thinking approach that recognizes incidents result from multiple, interacting contributing factors. The focus shifts from blame to learning, from single causes to systemic patterns, and from reactive fixes to proactive prevention.

Core Philosophy Shift:

- From **Root Cause** → To **Contributing Causes**: Recognizing multiple interacting factors
- From **Blame Assignment** → To **Learning Culture**: Focus on prevention and improvement
- From **Linear Analysis** → To **Systems Thinking**: Understanding complex interactions
- From **Person-Focused Actions** → To **System-Focused Improvements**: 57% system focus vs 30%
- From **Single Incident Focus** → To **Pattern Recognition**: Aggregate learning across incidents
- From **Investigation Reports** → To **Learning Artifacts**: Documentation that drives improvement

Research shows that Learning Team approaches generate nearly twice as many improvement actions (7.5 vs 3.5) compared to traditional RCA, with 57% being system-focused versus only 30% for RCA. By implementing this approach in COMPASS's AI-assisted post-mortem generation, we create documentation that doesn't just record what happened, but actively promotes organizational learning and systemic improvement.

1. The Fundamental Problem with Root Cause Analysis

After decades of use in healthcare and technology, evidence shows that Root Cause Analysis (RCA) has failed to deliver on its promise of preventing incident recurrence. The fundamental flaws in RCA's approach limit organizational learning and perpetuate cycles of similar incidents.

1.1 Eight Critical Challenges with RCA

Challenge	Impact on Learning	COMPASS Solution
Reductionist View	Oversimplifies complex incidents to single causes	Map multiple contributing factors across system levels
Linear Thinking	Misses feedback loops and interactions	Systems thinking with causal loop analysis
Blame Focus	Creates defensive culture, limits reporting	Just culture emphasizing learning over blame
Weak Solutions	80% administrative actions that don't prevent recurrence	Hierarchy of effectiveness prioritizing system changes
Poor Feedback	Learning doesn't reach those who need it	Transparent investigation visible to all stakeholders
Isolated Analysis	Each incident analyzed in isolation	Pattern recognition across incident clusters
Goal Displacement	Report becomes end product, not learning	Living documents that drive continuous improvement
Political Hijack	Investigations manipulated to avoid difficult truths	AI-assisted objective analysis with all hypotheses shown

1.2 Evidence from Healthcare and Other Industries

Research across multiple industries demonstrates RCA's limitations:

- Healthcare studies show only 45-70% of RCA recommendations are ever implemented
- Aviation and nuclear industries have moved to systems thinking approaches decades ago
- Learning Teams generate 114% more actions with nearly double the system focus
- Organizations using RCA show persistent 'organizational forgetting' with recurring incidents
- Meta-analyses reveal RCA investigations often stop at 'last barrier to fail' missing systemic issues

2. Contributing Causes: A Systems Thinking Approach

Instead of searching for a mythical "root cause," COMPASS adopts a contributing causes approach that recognizes incidents emerge from the complex interaction of multiple factors across the sociotechnical system. This approach, proven in other high-reliability industries, focuses on prevention rather than blame.

2.1 Understanding Contributing Causes

Contributing causes represent the multiple factors that combine to create the conditions for an incident. Each factor alone may not cause failure, but together they breach safety boundaries:

```
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■■ ■ SYSTEM LEVELS ■ ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■■■■■■■■■■■■■■■■■■■■■ ■ Government/Regulatory ■ Policies, regulations,
standards ■ ■ Organization Management ■ Resources, priorities, culture ■ ■
Technical Management ■ Procedures, training, tools ■ ■ Work Environment ■
Workload, time pressure, team ■ ■ Human-System Interface ■ Displays,
controls, automation ■ ■ Equipment & Technology ■ Reliability, design,
maintenance■ ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■■■■■■■■■■■■■ Each level contains factors that contribute to incident
causation. COMPASS maps contributing causes across ALL levels, not just the
immediate technical failure or human action.
```

2.2 Causal Factor Analysis Framework

COMPASS implements a sophisticated framework for identifying and analyzing contributing causes:

```
class ContributingCausesAnalyzer: def __init__(self): self.system_levels = [
'regulatory', 'organizational', 'technical', 'work_environment',
'human_system', 'equipment' ] self.causal_factors = [] def
analyze_incident(self, incident_data): """Identify contributing causes across
system levels""" factors = [] # Don't stop at first cause - explore all
branches for level in self.system_levels: level_factors =
self.identify_factors_at_level( incident_data, level )
factors.extend(level_factors) # Analyze interactions between factors
interactions = self.analyze_factor_interactions(factors) # Identify patterns
across similar incidents patterns = self.identify_cross_incident_patterns(
incident_data.type, factors ) return { 'contributing_causes': factors,
'interactions': interactions, 'systemic_patterns': patterns,
'prevention_opportunities': self.generate_preventions(factors) } def
generate_preventions(self, factors): """Focus on prevention, not blame"""
preventions = [] for factor in factors: # Evaluate prevention efficiency
prevention = { 'factor': factor, 'interventions':
self.identify_interventions(factor), 'effectiveness':
self.evaluate_effectiveness(factor), 'implementation_cost':
self.estimate_cost(factor) } preventions.append(prevention) # Prioritize by
prevention efficiency return sorted(preventions, key=lambda p:
p['effectiveness'] / p['implementation_cost'], reverse=True)
```

2.3 Moving from Blame to Prevention

The contributing causes approach fundamentally shifts the investigation focus:

Traditional RCA Focus	Contributing Causes Focus

Who made the error?	What system conditions enabled the error?
What rule was violated?	Why did the rule violation make sense at the time?
How do we prevent this exact scenario?	How do we strengthen system resilience?
What's the root cause?	What factors combined to breach safety boundaries?
Who should be retrained?	What system improvements prevent recurrence?
Single point of failure	Multiple interacting contributing factors

3. Building a Learning Culture Through AI-Assisted Documentation

Learning from incidents requires more than just documentation—it demands a culture that values transparency, psychological safety, and continuous improvement. COMPASS's AI-assisted post-mortem generation actively promotes this learning culture through its design and implementation.

3.1 Elements of a Learning Culture

Research identifies key elements necessary for effective learning from incidents:

- **Just Culture:** Focus on system improvement rather than individual blame
- **Psychological Safety:** Team members feel safe reporting incidents and near-misses
- **Transparency:** Investigation process and findings visible to all stakeholders
- **Double-Loop Learning:** Not just fixing problems but questioning underlying assumptions
- **Systemic View:** Understanding incidents as emerging from complex system interactions
- **Continuous Improvement:** Each incident contributes to organizational knowledge

3.2 Learning Teams Methodology

COMPASS incorporates Learning Teams methodology, proven to generate more effective improvements:

Aspect	Traditional RCA	Learning Teams	Impact
Participation	Only those involved in incident	All relevant staff, even if not involved	Reduces bias, increases perspectives
Focus	What went wrong	How work normally happens	Understanding work-as-done vs work-as-imagined
Questions	Why did you make that error?	What made sense at the time?	Psychological safety, richer information
Actions Generated	3.5 average	7.5 average	114% more improvement opportunities
System Focus	30%	57%	Nearly double system improvements
Culture Created	Defensive, blame-oriented	Open, learning-oriented	Increased reporting and engagement

3.3 AI's Role in Promoting Learning Culture

COMPASS's AI-assisted documentation actively promotes learning culture through:

- **Objective Analysis:** AI provides neutral, fact-based analysis reducing blame attribution
- **Pattern Recognition:** Identifies systemic issues across multiple incidents

• **Comprehensive Coverage:** Ensures all contributing factors are documented, not just obvious ones

• **Consistent Framework:** Applies systems thinking uniformly to all investigations

• **Knowledge Preservation:** Captures tacit knowledge that might otherwise be lost

• **Accessible Language:** Translates technical findings for diverse stakeholders

4. Implementation: Systems-Based Post-Mortem Generation

Implementing contributing causes analysis and learning culture in COMPASS requires fundamental changes to how the system captures, analyzes, and presents incident data. The implementation focuses on systemic understanding rather than linear causation.

4.1 Data Collection Across System Levels

```
class SystemsDataCollector: def __init__(self): self.data_sources = {
'regulatory_level': [ 'compliance_status', 'recent_policy_changes',
'industry_standards', 'audit_findings' ], 'organizational_level': [
'resource_allocation', 'competing_priorities', 'organizational_changes',
'culture_metrics' ], 'technical_management': [ 'procedures',
'training_records', 'tool_availability', 'technical_debt' ],
'work_environment': [ 'workload_metrics', 'team_dynamics', 'time_pressures',
'communication_patterns' ], 'human_system_interface': [ 'ui_complexity',
'automation_surprises', 'alert_fatigue', 'cognitive_load' ],
'equipment_technology': [ 'failure_rates', 'maintenance_history',
'design_limitations', 'known_issues' ] } async def
collect_incident_context(self, incident): """Gather data across all system
levels""" context = {} # Parallel collection across levels tasks = [] for
level, sources in self.data_sources.items(): for source in sources: task =
self.collect_source_data( incident, level, source ) tasks.append(task) #
Gather all data simultaneously results = await asyncio.gather(*tasks) #
Organize by system level for result in results: level = result['level'] if
level not in context: context[level] = [] context[level].append(result) # Add
temporal context context['temporal'] =
self.collect_temporal_factors(incident) # Add interaction patterns
context['interactions'] = self.identify_cross_level_interactions(context)
return context
```

4.2 Contributing Causes Identification

The system uses multiple analytical techniques to identify contributing causes:

- **Causal Loop Analysis:** Identify feedback loops and systemic structures
- **Barrier Analysis:** Which barriers failed and which held?
- **Work-As-Done vs Work-As-Imagined:** Gap between procedures and practice
- **Migration Analysis:** How did practices drift toward failure boundaries?
- **Coupling Analysis:** How tightly coupled were the contributing factors?
- **Resilience Assessment:** What adaptive capacity existed or was missing?

4.3 Post-Mortem Structure for Learning

The AI-generated post-mortem follows a structure designed to maximize learning:

Section	Content	Learning Purpose
Incident Overview	What happened, when, impact	Context setting without blame
Normal Work Description	How work usually happens	Understanding work-as-done

Contributing Causes Map	All factors across system levels	Systems thinking visualization
Hypothesis Journey	All tested paths, including disproved	Transparency and learning from exploration
Causal Interactions	How factors combined	Understanding emergence
Similar Patterns	Links to related incidents	Pattern recognition
System Improvements	Hierarchy of interventions	Focus on prevention
Learning Points	Key insights for teams	Knowledge transfer
Questions for Reflection	Prompts for team discussion	Continued learning

5. Cross-Incident Learning and Pattern Recognition

One of RCA's greatest failings is analyzing incidents in isolation. COMPASS aggregates learning across incidents to identify systemic patterns and emerging risks before they result in major failures.

5.1 Pattern Recognition Framework

```
class CrossIncidentAnalyzer: def __init__(self): self.incident_database = []
self.pattern_library = {} def identify_patterns(self, new_incident):
"""Identify patterns across incident clusters""" patterns = {
'recurring_factors': self.find_recurring_contributing_causes(),
'temporal_clusters': self.identify_temporal_patterns(), 'migration_trends':
self.detect_practice_drift(), 'weak_signals': self.identify_emerging_risks(),
'systemic_vulnerabilities': self.map_system_weaknesses() } # Analyze pattern
strength for pattern_type, pattern_data in patterns.items():
pattern_data['strength'] = self.calculate_pattern_strength( pattern_data )
pattern_data['incidents_involved'] = self.link_related_incidents(
pattern_data ) return patterns def find_recurring_contributing_causes(self):
"""Identify factors that appear across multiple incidents""" factor_frequency
= {} for incident in self.incident_database: for factor in
incident.contributing_causes: key =
f"{factor.system_level}:{factor.description}" if key not in factor_frequency:
factor_frequency[key] = { 'count': 0, 'incidents': [], 'contexts': [] }
factor_frequency[key]['count'] += 1
factor_frequency[key]['incidents'].append(incident.id)
factor_frequency[key]['contexts'].append(incident.context) # Identify
significant patterns recurring = { k: v for k, v in factor_frequency.items()
if v['count'] >= 3 # Appears in 3+ incidents } return recurring def
generate_systemic_recommendations(self, patterns): """Generate
recommendations addressing systemic issues""" recommendations = [] for
pattern in patterns['recurring_factors'].values(): if pattern['strength'] >
0.7: # Strong pattern rec = self.create_systemic_intervention(pattern)
recommendations.append(rec) return sorted(recommendations, key=lambda r:
r['potential_impact'], reverse=True)
```

5.2 Organizational Learning Metrics

COMPASS tracks metrics that indicate organizational learning effectiveness:

Metric	What It Measures	Target
Repeat Incident Rate	Similar incidents recurring	< 5%
Time Between Similar Incidents	Learning retention	Increasing trend
System-Focused Actions %	Quality of improvements	> 60%
Action Implementation Rate	Follow-through on learning	> 90%
Cross-Team Learning	Knowledge dissemination	> 80% teams aware
Weak Signal Detection	Proactive risk identification	> 70% before incident
Practice Drift Rate	System degradation	< 2% per quarter

Learning Velocity	Speed of improvement adoption	< 1 week

6. Practical Implementation Roadmap

Transitioning from root cause analysis to contributing causes and learning culture requires careful change management. This roadmap provides a practical path for organizations to transform their incident learning processes.

6.1 Phase 1: Foundation Building (Weeks 1-6)

- **Cultural Assessment:** Evaluate current blame vs learning culture
- **Stakeholder Alignment:** Build consensus on moving beyond RCA
- **Terminology Shift:** Replace 'root cause' with 'contributing causes' in all documentation
- **Training Program:** Systems thinking and just culture for investigation teams
- **Pilot Selection:** Choose high-frequency, lower-severity incidents for initial implementation
- **Metrics Baseline:** Establish current state for learning effectiveness metrics

6.2 Phase 2: Systems Integration (Weeks 7-12)

- **Multi-Level Data Collection:** Implement collection across all system levels
- **Learning Team Formation:** Establish cross-functional investigation teams
- **AI Prompt Engineering:** Develop prompts focusing on contributing causes
- **Pattern Database:** Create repository for cross-incident patterns
- **Feedback Loops:** Establish mechanisms for learning dissemination
- **First Learning Reports:** Generate and review initial AI-assisted post-mortems

6.3 Phase 3: Scale and Optimize (Weeks 13-24)

- **Full Deployment:** Extend to all incident types and severities
- **Pattern Analytics:** Implement cross-incident pattern recognition
- **Systemic Interventions:** Design improvements addressing recurring patterns
- **Learning Velocity:** Optimize speed from incident to implemented improvement
- **Culture Metrics:** Measure and reinforce learning culture behaviors
- **Continuous Improvement:** Refine based on effectiveness data

6.4 Critical Success Factors

- **Executive Sponsorship:** Leadership must model learning behavior, not blame
- **Psychological Safety:** Teams must feel safe reporting failures and near-misses
- **Time Investment:** Allow sufficient time for thorough systems analysis

- **Cross-Functional Participation:** Include diverse perspectives in investigations
- **Action Follow-Through:** Commit resources to implement system improvements
- **Transparency:** Share learning broadly, including failures and uncertainties

Conclusion: The Path Forward

The evidence is clear: Root Cause Analysis has failed to deliver on its promise of preventing incident recurrence. After decades of use, organizations continue to experience the same types of failures, suggesting that the fundamental approach is flawed. The problem isn't in the execution of RCA—it's in the underlying philosophy that assumes complex system failures can be reduced to singular causes. By adopting a contributing causes approach grounded in systems thinking, COMPASS transforms incident investigation from a blame-assignment exercise into a learning opportunity. This shift isn't merely semantic—it represents a fundamental change in how we understand failure and improvement in complex sociotechnical systems. The integration of Learning Teams methodology, which generates 114% more improvement actions with nearly double the system focus, demonstrates that better approaches already exist. When combined with AI's capability for pattern recognition and objective analysis, we can finally move beyond the limitations that have plagued incident learning for decades. Most importantly, this approach recognizes that incidents are not failures to be punished but opportunities to understand how our complex systems actually work. By showing all investigation paths—including those that proved fruitless—we create transparency that builds trust and accelerates learning. By focusing on prevention rather than blame, we create psychological safety that encourages reporting and participation. The journey from root cause to contributing causes, from blame to learning, from isolation to patterns, represents more than a technical upgrade—it's a cultural transformation. Organizations that make this shift will not only reduce incident recurrence but will build resilient systems capable of adapting to unexpected challenges.

> "As an engineer, stay away from using the terms 'root cause' and 'proximate cause'—those terms are laden with poorly defined baggage. Instead, look for causal factors, engage in prevention analysis, and provide a nuanced discussion on prevention of the accident rather than assignment of blame." — Martin Ottaway

Key Takeaways for COMPASS Implementation:

1. Abandon the search for single root causes—embrace multiple contributing factors
2. Build investigation teams that include those not involved in the incident
3. Focus relentlessly on system improvements over person-focused actions
4. Make all investigation paths transparent, including disproved hypotheses
5. Track patterns across incidents to identify systemic vulnerabilities
6. Measure learning effectiveness, not just investigation completion

7. Create psychological safety that encourages reporting and participation
8. Use AI to reduce bias and identify patterns humans might miss

The future of incident learning lies not in finding who or what to blame, but in understanding the complex interplay of factors that shape system behavior. COMPASS, with its AI-assisted documentation and systems thinking approach, represents a significant step toward that future— a future where every incident makes our systems stronger, safer, and more resilient.