Human Cooperation System (HCS)

A system model for designing, sustaining, and diagnosing human cooperation in complex work.

Viktor Jevdokimov, Vilnius, Lithuania

Table of contents

1. INTRODUCTION & QUICK START

1.1 Human Cooperation System (HCS) – From Context to Practice

A multi-modal operating system for understanding, designing, and stabilizing human cooperation.

> **30-Second Summary**
> The **Human Cooperation System (HCS)** defines the stable conditions and functions that make human cooperation possible.
> It reveals why alignment, trust, and flow break down — and provides a structure for restoring or improving them across different stages of work.

1.1.1 Introduction

Human cooperation is not powered by tools, talent, or methods alone.
It emerges from a deeper **system of shared meaning, aligned boundaries, trust, and coordinated decision-making**.

The **Human Cooperation System (HCS)** maps this system.
It exposes the invisible architecture behind all collaborative work:
the **conditions**, **needs**, and **functions** that allow groups to make sense, act, and adapt together.

Cooperation problems rarely start with skills or velocity.
They start when the **system of cooperation** becomes unstable — when people interpret work differently, operate under mismatched expectations, or lack the structural safety to speak, align, or decide.

HCS helps teams **see and repair the system**, not just the symptoms.

1.1.2 Why HCS Exists

There are two parallel paths organizations use to solve collaboration problems:

The Path of Encapsulation
Reducing friction by narrowing collaboration, separating responsibilities, and limiting interdependence.
Great for clear, predictable work — but brittle in complexity.

The Path of Integration
Managing friction through shared meaning, aligned boundaries, and transparent decision-making.
Necessary whenever work is ambiguous, cross-functional, political, or fast-changing.

Most teams unconsciously choose Encapsulation even when Integration is required — and cooperation collapses.

HCS provides the architecture for choosing the right path, and the structure for governing both.

1.1.3 Who It's For

HCS is designed for anyone responsible for ensuring people can work together effectively:

- **Project & Product Leads** managing cross-functional flow and shared accountability.
- **Engineering, Design, and Research Managers** growing autonomy and reducing misalignment.
- **Consultants & Advisors** diagnosing systemic friction across teams or organizations.
- **Practitioners of 3SF** who want to understand the foundational system beneath the framework.

If you are here from **3SF**, HCS is the **underlying system** that explains why engagement, delivery, and value succeed or fail.

1.1.4 Purpose of HCS

The Human Cooperation System is a **systemic model and decision structure** for:

- Understanding what cooperation *requires*
- Diagnosing where cooperation *breaks*
- Governing how cooperation *evolves*
- Choosing the right *mode* of intervention
- Connecting human needs, organizational intent, and work structures

HCS does **not** prescribe practices.
It provides the **systemic logic** beneath every practice, framework, and governance choice.

It defines:

- The **preconditions** needed for cooperation
- The **needs** that sustain trust and engagement
- The **functions** that transform shared intent into coordinated action
- The **dynamics** that amplify or distort cooperation
- The **modes** teams must use depending on their stage of work

Together, these form the **Systemic Basis of Work** — a way to understand and shape cooperation itself.

1.1.5 Structure of HCS

HCS is organized into three architectural layers:

Core Model — The Physics of Cooperation

The Core Model defines the stable, universal architecture of cooperation.

It answers:

“What must be true for people to work together at all?”

This includes:

- **Vision, Principles, Beliefs** — the intent and stance of HCS
- **The Conditions Matrix** — the structural landscape of cooperation
- **The Pyramid** — how stability develops from preconditions to meta-practices
- **The Level Rule** — why interventions must follow a systemic sequence

This layer is **framework-independent** and applies to all forms of human work.

Extended Human Dynamics — Real-World Complexity

Even when the structure is correct, cooperation is shaped by deeper forces:

- Psychological safety
- Power and authority
- Identity, belonging, and interpersonal patterns
- Misaligned incentives and political vectors
- Cultural and relational history

This layer answers:

“Why is cooperation difficult in real organizations?”

Extended Dynamics does not add new requirements.
It provides the **human context** needed to understand why cooperation deviates from the Core Model.

System Modes — How to Use HCS Across the Lifecycle of Work

Cooperation is not static.
Different stages of work require different systemic activities.
HCS expresses these as **modes**:

1. **Setup Mode** — establishing preconditions and governance before work begins
2. **Stabilization Mode** — fixing mismatches, breakdowns, and early friction
3. **Growth Mode** — increasing autonomy, trust, and adaptive capacity
4. **Conflict Mode** — resolving human, political, or relational breakdowns
5. **Reset Mode** — re-establishing cooperation after radical context change

Each mode applies the Core Model differently depending on what the system needs
— **design**, **repair**, **evolution**, **restoration**, or **renewal**.

1.1.6 Reading Path

HCS can be approached from two directions depending on your goals:
(1) understanding the system, or **(2) using the system to diagnose and guide real cooperation.**

Reading Path for Theorists / Learners

If your goal is to understand how HCS works as a system—its concepts, structure, and reasoning—follow this order:

1. START WITH THE CORE MODEL

1. **Vision, Principles, and Beliefs**
2. **The Matrix**
3. **The Pyramid**

These chapters explain what cooperation *is* structurally and how stability depends on the Level Rule.

2. CONTINUE WITH EXTENDED HUMAN DYNAMICS

Explore how context, psychology, and politics shape how cooperation is *experienced* in real systems.

3. MOVE TO SYSTEM MODES

Learn how cooperation behaves over time and what type of work is appropriate in each mode: - **Setup** (design)
- **Stabilization** (repair and calibration)
- **Growth** (optimization)
- **Conflict** (realignment and safety)
- **Reset** (renewal)

This path gives you a complete conceptual understanding of HCS before applying it.

Reading Path for Practitioners / Coaches

If your goal is to **analyze, stabilize, or guide real cooperation**, start with the parts of HCS designed for practice.

1. BEGIN WITH DIAGNOSTICS

Start here — this is your primary instrument. - **Diagnostic Workflow**
Shows how to move from *observation* → *structure* → *level* → *intervention*. - **Diagnostic Dynamics**
Adds the human, political, and psychological layers when structure alone is not enough.

2. FOLLOW THE WORKFLOW BACK INTO THE CORE MODEL (WHEN PROMPTED)

Use the theory *only as needed*: - **Matrix** (to identify *which function* is strained)
- **Pyramid** (to identify *which level* is unstable)

The workflow will tell you when to consult these.

3. USE SYSTEM MODES TO CHOOSE WHAT KIND OF WORK IS POSSIBLE RIGHT NOW

Modes help you avoid applying the wrong intervention type: - Setup, Stabilization, Growth, Conflict, Reset

4. CONSULT EXTENDED HUMAN DYNAMICS ONLY IF DYNAMICS FEEL "HOT"

When political or psychological fields distort cooperation, this section helps interpret patterns safely and precisely.

5. FINALLY, USE THE PRACTICES MAP AS OPTIONAL SUPPORT

Only after diagnosis is clear, pick practices that strengthen the functions you identified.

In short:
- **Learners**: read HCS top-down, from theory → dynamics → modes.
- **Practitioners**: start with **Diagnostics**, and let the workflow direct you to the required parts of the model.

1.1.7 Positioning

Think of HCS as a **cooperation operating system**:

- Methods like Agile, Lean, design thinking, DevOps, or 3SF succeed only when the **underlying system of cooperation** is stable.
- HCS reveals that system and provides a structure for improving it.
- Where 3SF operationalizes cooperation inside delivery flows, HCS explains *why* those operations function — or fail.

> **In essence:**
> HCS describes the *physics of cooperation* —
> the conditions and functions that allow human work to remain aligned, trusted, and adaptive.

2. CORE MODEL

2.1 Vision, Principles, and Beliefs

The **Human Cooperation System (HCS)** defines the systemic architecture that enables people and organizations to work together with clarity, trust, and adaptability.
It describes the **conditions** and **functions** that sustain cooperation, and the **forces** that destabilize it when misaligned.

HCS treats cooperation not as a cultural preference or interpersonal skillset, but as a **governable system** — one that can be designed, stabilized, and evolved across different modes of work.

2.1.1 Why Systems Break

Human cooperation breaks long before performance drops.
When people struggle to work together, the cause is rarely skills, motivation, or tools — it is almost always a **systemic mismatch in how interdependence is handled**.

At the root of this mismatch are two parallel, conflicting paths for solving problems in organizations:

The Two Paths of Problem Solving

The Path of Encapsulation — Reducing Interdependence
Encapsulation tries to *avoid* friction by narrowing collaboration: clearer handoffs, stronger boundaries, fixed responsibilities, predefined inputs and outputs.
This works well when the work is stable, modular, or predictable.
But it collapses when ambiguity rises, understanding must be negotiated, or decisions require shared judgment.

The Path of Integration — Managing Interdependence
Integration treats friction as **information**, not noise.
It aligns meaning, boundaries, and decisions so that people can make sense of complexity together.
Integration becomes essential when work is cross-functional, uncertain, political, or fast-changing.

The Core Reason Systems Break

Most cooperative systems fail because teams attempt to solve an **integration problem using encapsulation tools**:

- Ambiguity is high → but meaning is not aligned.
- Boundaries are fluid → but roles remain rigid.
- Decisions require shared judgment → but authority stays siloed.
- Human dynamics affect outcomes → but governance ignores them.

This mismatch produces predictable symptoms: misalignment, rework, dependency friction, escalating tension, and loss of trust.

HCS exists to reveal which path is required, when, and why — and to provide the structure for governing interdependence instead of fighting it.

2.1.2 Vision

To enable individuals, teams, and organizations to **work together intentionally and reliably**, regardless of domain, structure, or methodology.
HCS envisions a world where cooperation is understood as a **systemic discipline**, not an accidental outcome of talent, tools, or goodwill.

> **Vision Statement**
> *A cooperative world where systems of work evolve with people — not against them.*

2.1.3 Principles

The following principles form the **systemic physics** of cooperation.
They define how cooperative systems remain stable and why interventions must respect the underlying structure.

Principle	Description
Function-First	A practice is effective only when it fulfills the cooperative function it is meant to support. Tools without purpose amplify chaos.
Integration Over Encapsulation (When Required)	Use encapsulation for simple work; use integration for complex work. Treat friction as information, not noise.
The Level Rule	A dysfunction cannot be corrected from a higher layer of abstraction. Stability emerges bottom-up, from conditions to functions to practices.
Clarity Before Speed	Shared meaning is cheaper than rework. Misalignment is the most expensive form of waste.
Evidence Over Interpretation	Cooperation is observable. Behavioral evidence is more reliable than assumptions about intent or personality.
Feedback Closes the Loop	Systems drift without timely feedback. Stability requires continuous sensing and adjustment.
Trust Before Control	Excessive control compensates for missing trust. Cooperative flow emerges when reciprocal trust is structurally possible.
Autonomy with Accountability	Freedom without shared responsibility fragments; responsibility without autonomy suffocates. Balance sustains flow.
Reflection Enables Evolution	Intentional reflection transforms experience into adaptation. Without it, systems repeat avoidable patterns.

These principles guide how the HCS Core Model and System Modes should be interpreted and used.

2.1.4 Beliefs

HCS is grounded in several foundational beliefs about human work systems:

1. **Cooperation precedes performance.**
   Teams fail for systemic and relational reasons long before they fail for technical ones.
2. **Human needs shape system stability.**
   Safety, belonging, purpose, and agency are not soft concepts — they are structural conditions.
3. **Shared meaning is essential for adaptation.**
   A group cannot respond coherently to change if it does not interpret the world coherently.
4. **Trust is systemic, not emotional.**
   It is built through reliable behaviors, aligned expectations, and transparent decision-making.
5. **Frameworks depend on cooperative conditions.**
   No methodology can compensate for missing clarity, trust, or aligned boundaries.
6. **Organizations are learning organisms.**
   Their adaptability depends on the quality and speed of feedback loops across roles and structures.
7. **Friction is information.**
   Cooperation improves when teams learn to examine tension instead of avoiding or escalating it.

2.1.5 Scope and Non-Goals

HCS defines **what makes cooperation possible**, not how to run projects or structure organizations.
It provides the foundation beneath methods — not a method itself.

- **Not a Delivery Process**
  HCS is not a task-management system or workflow model. It describes the conditions and functions that make any workflow viable.
- **Not an Organizational Blueprint**
  HCS does not dictate reporting structures. It defines the cooperative relationships that must exist *regardless* of structure.
- **Not a Psychological Model**
  HCS acknowledges human psychology but does not attempt to change personal personalities or traits. It governs cooperation, not therapy.
- **Not a Replacement for Frameworks**
  HCS underlies frameworks like Agile, Lean, and 3SF. It explains why those frameworks succeed or fail depending on cooperative stability.

This section forms the philosophical foundation of HCS.
Subsequent chapters — the Matrix, the Pyramid, Extended Dynamics, and System Modes — elaborate how these principles unfold in practice.

2.2 The Human Cooperation System Matrix

Cooperation emerges when people depend on one another to achieve outcomes they cannot (or should not) accomplish alone.
Once interdependence appears, work becomes a **system of relationships**, requiring clarity of meaning, mutual commitment, coordinated roles, and the ability to adapt.

The **HCS Matrix** defines these requirements.
It describes the **conditions** of the work system and the **human needs** that must align for cooperation to be stable and adaptive.

2.2.1 Purpose of the Matrix

The Matrix captures the **minimum viable structure of cooperation** — the patterns that must be present for people to make sense, act, and adjust together.

It is not a methodology or workflow.
It is a **structural lens** that reveals:

- why cooperation stabilizes or destabilizes
- where friction originates
- which cooperation functions are strained
- when integration (not encapsulation) is required

The Matrix forms the **existential layer** of HCS — the layer beneath all tools, frameworks, and governance systems.

The **Extended Human Dynamics** section builds on this layer by explaining how psychological, political, and relational forces amplify or distort these functions.

2.2.2 Encapsulation, Integration, and the Matrix

Two fundamental strategies exist for handling cooperation challenges:

- **Encapsulation** reduces interdependence by narrowing boundaries, clarifying handoffs, and limiting meaning-sharing.
  It works when work is modular, predictable, and clear.
- **Integration** manages interdependence by aligning meaning, negotiating boundaries, and enabling shared sense-making.
  It becomes necessary when work is complex, ambiguous, cross-functional, or fast-changing.

The Matrix makes interdependence **visible**, helping teams identify where encapsulation is appropriate and where **integration becomes mandatory**.

Each matrix cell represents a **cooperation function** generated by the interaction of a *work condition* and a *human need*.
If one function is weak or absent, cooperation becomes unstable regardless of tools or processes.

2.2.3 Dimensions of the Matrix

The Matrix has two axes:

- **Vertical axis — Core Work Conditions**
  The external realities of the work environment.
- **Horizontal axis — Core Human Needs for Cooperation**
  The internal requirements for participating in cooperation.

Together, they define 25 **cooperation functions**.

Core Work Conditions (Vertical Axis)

These are the **objective features of the system** that shape how work happens.

1. **Common Purpose**
   The shared reason for the work. It provides direction, meaning, and coherence.
   Without it, local optimizations fragment and energy dissipates into unconnected goals.
2. **Interdependence**
   The degree to which people rely on each other's work to achieve outcomes.
   Interdependence makes coordination necessary and raises the cost of misalignment.
   Ignoring it leads to hidden dependencies, bottlenecks, and blame.
3. **Communication**
   The structure and flow of information, intent, and meaning between people.
   Communication is more than message exchange: it includes language, timing, channels, and accessibility.
   Poor communication distorts reality for different participants.
4. **Trust**
   Confidence in others' reliability, competence, and intent.
   Trust reduces the need for control and negotiation in everyday decisions.
   When trust is low, every interaction becomes expensive, guarded, or defensive.
5. **Change / Uncertainty Tolerance**
   The capacity to operate under shifting conditions, ambiguity, or evolving constraints.
   Change can come from markets, technology, governance, or internal decisions.
   Tolerance for uncertainty ensures that cooperation continues even when plans must shift.

Core Human Needs for Cooperation (Horizontal Axis)

These are the **subjective and relational requirements** for people to join and sustain cooperation.

1. **Shared Understanding**
   A compatible interpretation of goals, language, constraints, and context.
   This does not require full agreement — only enough overlap to coordinate based on a shared sense of reality.
2. **Mutual Commitment**
   A shared willingness to contribute to collective goals and to each other.
   It includes reliability, follow-through, and a felt sense that "we are in this together."
   Without mutual commitment, cooperation becomes transactional and fragile.
3. **Feedback Loops**
   Mechanisms for observing results, sharing signals, and adjusting behavior.
   These can be formal (metrics, reviews, ceremonies) or informal (conversations, peer correction).
   Without feedback, systems drift and small problems become systemic.
4. **Distribution of Roles**
   Clarity around responsibilities, boundaries, and contributions.
   People need to know who does what, where their authority starts and ends, and how roles relate.
5. **Autonomy & Agency**
   Freedom to act intentionally within the cooperative structure.
   Agency enables people to make decisions, take initiative, and feel ownership over their contributions.

2.2.4 Matrix Cell Descriptions (5×5 = 25 Cooperation Functions)

Each cell describes what must happen when a Work Condition meets a Human Need.

Common Purpose × Human Needs

1. **Shared Understanding → Alignment on Why**
   People interpret the goal similarly and hold a compatible view of what "success" means.
2. **Mutual Commitment → Willingness to Act**
   Individuals commit effort because the shared purpose feels meaningful and legitimate.
3. **Feedback Loops → Learning the Mission**
   Teams update their sense of purpose through real outcomes and signals, not assumptions.
4. **Distribution of Roles → Contribution Clarity**
   Each person understands how their role contributes to the shared purpose.
5. **Autonomy & Agency → Room for Initiative**
   People can act creatively in support of the purpose without waiting for permission.

Interdependence × Human Needs

1. **Shared Understanding → Task Relationships**
   People understand how their work depends on others and how others depend on them.
2. **Mutual Commitment → Responsibility to Each Other**
   Team members feel accountable for how their work affects the group.
3. **Feedback Loops → Outcome Reflection**
   Teams regularly examine dependencies to improve flow and reduce friction.
4. **Distribution of Roles → Coordination**
   Roles and responsibilities align with actual dependency structures.
5. **Autonomy & Agency → Local Decision-Making**
   People can act within dependency networks without needing constant approval.

Communication × Human Needs

1. **Shared Understanding → Common Language**
   People use terms, concepts, and frames in ways that mean the same thing.
2. **Mutual Commitment → Social Contract**
   Communication carries a baseline respect and reliability that supports cooperation.
3. **Feedback Loops → Signal/Response**
   Signals reach the right people, are interpreted correctly, and trigger meaningful adjustments.
4. **Distribution of Roles → Interaction Clarity**
   People know who to talk to, when, and for what purpose.
5. **Autonomy & Agency → Permission to Act**
   Communication norms empower action rather than reinforce hierarchy or fear.

Trust × Human Needs

1. **Shared Understanding → Meaning Consistency**
   People assume others interpret situations honestly and coherently.
2. **Mutual Commitment → Reliability**
   People trust that commitments are kept, and failures are signaled early.
3. **Feedback Loops → Safety in Feedback**
   People can share concerns or corrections without fear of retribution.
4. **Distribution of Roles → Delegation**
   Roles can be distributed with confidence; people do not micromanage.

5. **Autonomy & Agency → Empowerment**
   People act with confidence because trust supports decentralization.

Change / Uncertainty × Human Needs

1. **Shared Understanding → Scenario Awareness**
   People understand how change affects work and can interpret shifts consistently.
2. **Mutual Commitment → Resilience**
   Commitment persists even when plans shift or constraints evolve.
3. **Feedback Loops → Learning from Change**
   Teams rapidly integrate new information and adjust without panic.
4. **Distribution of Roles → Flexibility**
   Roles can shift or expand temporarily without destabilizing cooperation.
5. **Autonomy & Agency → Adaptability**
   People can act under uncertainty, making thoughtful, context-aware decisions.

2.2.5 Table View

Work Needs / Work Conditions	Shared Understanding	Mutual Commitment	Feedback Loops	Distribution of Roles	Autonomy & Agency
Common Purpose	Alignment on why	Willingness to act	Learning intent	Contribution clarity	Room for initiative
Interdependence	Task relationships	Responsibility	Outcome reflection	Coordination	Local decision-making
Communication	Common language	Social contract	Signal/response	Interaction clarity	Permission to act
Trust	Meaning consistency	Reliability	Safety in feedback	Delegation	Empowerment
Change / Uncertainty	Scenario awareness	Resilience	Learning from change	Flexibility	Adaptability

2.2.6 How the Matrix Fits Into HCS

The Matrix explains **what cooperation requires**.
The **Pyramid** describes **how these requirements develop and stabilize**.
The **System Modes** describe **how to design, repair, or evolve these functions** depending on the stage of work.

Together, they form the structural and operational foundation of the Human Cooperation System.

2.3 The Human Cooperation System Pyramid

2.3.1 Purpose of the Pyramid

The **HCS Pyramid** visualizes how cooperation develops through **five systemic levels** – from the most fundamental preconditions for working together to reflective innovation at the top.

Where the HCS Matrix defines **what must coexist** (conditions × human needs),
the Pyramid shows **how these elements depend on one another over time**:

- Lower levels provide the **foundation**.
- Higher levels express **increasing sophistication and stability**.
- Attempts to improve cooperation by acting only at higher levels fail when lower levels are weak.

In short:

> The Matrix describes **the anatomy of cooperation**.
> The Pyramid describes **its order of development**.

The Pyramid is a core part of the **Level logic** of HCS:
it explains why some interventions work and others don't, based on *which level* they target and *what that level depends on*.

2.3.2 Levels Overview

The Pyramid consists of five levels:

1. **Preconditions for Cooperation** – existential requirements to work together at all.
2. **Core Human Needs for Cooperative Work** – what people require to participate in cooperation.
3. **Cooperative System Functions** – the stable "muscles" that turn needs into coordinated work.
4. **Practices & Frameworks** – concrete ways of performing those functions.
5. **Meta-Practices & Innovation** – the ability to redesign and evolve the system itself.

Each higher level depends on the relative stability of the levels below it.
Teams can briefly "jump" ahead, but sustained cooperation requires respecting these dependencies.

Pyramid Summary Table

Level	Name	Examples	Description
5 (Apex)	Meta-Practices & Innovation	• Designing custom playbooks • Inventing new practices • Matching practices to functions deliberately • Coaching others in systemic thinking	Teams consciously reflect on, adapt, and redesign their way of working. Practices are tailored, combined, or invented to better serve cooperative needs. Knowledge is shared to elevate the whole system.
4	Practices & Frameworks	• Retrospectives • RACI matrix • Scrum sprint planning • JTBD interviews • Kanban boards	Specific, evolving methods that fulfill stable functions. Practices and frameworks change over time, but the underlying functions they serve remain stable.
3	Cooperative System Functions	• Problem Discovery • Planning & Prioritization • Monitoring & Feedback • Enablement & Empowerment • Adaptation & Learning	The stable "muscles" of cooperation — what must happen for work to succeed. Represented by the 25 cells in the Human Cooperation System Matrix.
2	Core Human Needs for Cooperative Work	• Shared Understanding • Mutual Commitment • Feedback Loops • Distribution of Roles • Autonomy & Agency	The human-level enablers that make foundational conditions operational. Corresponds to the Matrix's **horizontal axis**.
1 (Foundation)	Preconditions for Cooperation	• Common Purpose • Interdependence • Communication • Trust • Change / Uncertainty Tolerance	The existential conditions for cooperation to exist at all. Corresponds to the Matrix's **vertical axis**. Without these, there is no "working together."

2.3.3 Level 1 – Preconditions for Cooperation (Foundation)

Level 1 contains the **existential conditions** that must be present before there is any real "working together."
Without these, coordination is accidental, fragile, or impossible.

These preconditions correspond to the **vertical axis** of the Matrix:

- **Common Purpose** – there is a shared "why" behind interaction.
- **Interdependence** – outcomes depend on more than one person's contribution.
- **Communication** – a basic ability to exchange signals, language, or symbols.
- **Trust** – a minimal belief that others will not harm and will reciprocate.
- **Change / Uncertainty Tolerance** – some capacity to function despite shifts and unknowns.

When Level 1 is weak:

- people don't know why they are together,
- dependencies are hidden or ignored,
- communication channels are missing or blocked,
- trust is absent or purely transactional,
- and any change quickly destabilizes the system.

Level 1 answers the question:
"Is there enough shared ground for cooperation to exist at all?"

2.3.4 Level 2 – Core Human Needs for Cooperative Work

Level 2 contains the **human-level enablers** that make foundational conditions workable in daily interactions.
These needs correspond to the **horizontal axis** of the Matrix:

- **Shared Understanding** – people interpret goals, constraints, and context in a compatible way.
- **Mutual Commitment** – people are willing to contribute and follow through together.
- **Feedback Loops** – people can see results, exchange signals, and adjust.
- **Distribution of Roles** – people know who does what and how roles relate.
- **Autonomy & Agency** – people can act with intention and ownership.

Level 2 transforms the abstract fact of "we are together" into **workable, day-to-day cooperation**.

When Level 2 is weak:

- people talk past each other despite having a common purpose,
- commitment is shallow or one-sided,
- feedback is delayed, distorted, or unsafe,
- roles are vague or contested,
- initiative is suppressed or punished.

Level 2 answers the question:
"Do people have what they need to participate in cooperation sustainably?"

> Additional human needs (e.g., belonging, meaning, identity, recognition) are covered in the Extended Human Dynamics section.
> Level 2 focuses on the minimal needs required *specifically* for cooperative work.

2.3.5 Level 3 – Cooperative System Functions

Level 3 contains the **stable functions** that translate conditions and needs into **coordinated, repeatable work**.

These functions live in the 25 cells of the Matrix – each cell representing a specific cooperation function where a work condition meets a human need (e.g., *Common Purpose × Shared Understanding → Alignment on why*).

Examples of Level 3 functions:

- **Problem Discovery** – clarifying the problem space and aligning on what needs to be solved.
- **Planning & Prioritization** – deciding what to do, in what order, given interdependence and constraints.
- **Monitoring & Feedback** – sensing progress and adjusting based on real-world signals.
- **Enablement & Empowerment** – ensuring people have the information, trust, and authority to act.
- **Adaptation & Learning** – updating how work is done in response to change and outcomes.

These functions are the **“muscles” of cooperation**.
If Level 1–2 are reasonably present, but these functions are weak, cooperation feels:

- busy but uncoordinated,
- decision-heavy but unclear,
- full of effort but light on outcomes.

Level 3 answers the question:
“Is there a functioning system that turns intent into coordinated action?”

2.3.6 Level 4 – Practices & Frameworks

Level 4 contains the **concrete, evolving ways** in which Level 3 functions are performed.

Different organizations, teams, or domains will adopt different practices to fulfill the same function.
Practices are **implementations**, not the function itself.

Examples:

- **Retrospectives** fulfill elements of *Monitoring & Feedback* and *Learning from change*.
- **RACI matrices** fulfill *Distribution of Roles* and *Coordination*.
- **Scrum sprint planning** fulfills *Planning & Prioritization*.
- **JTBD interviews** fulfill *Problem Discovery*.
- **Kanban boards** fulfill *Flow & Focus* plus *Monitoring & Feedback*.

Level 4 changes faster than Levels 1–3:

- practices can be adopted, modified, or replaced,
- frameworks can be introduced or retired,
- tooling can be upgraded or reconfigured.

When Level 4 is strong but Levels 1–3 are weak, teams experience **“cargo cult” adoption**:
they perform the practices but do not obtain the intended outcomes because the underlying functions and needs are not supported.

Level 4 answers the question:
“How are cooperative functions currently implemented in this context?”

2.3.7 Level 5 – Meta-Practices & Innovation (Apex)

Level 5 contains the **capacity to reflect on and redesign the system itself**.

At this level, teams and organizations:

- consciously select or adapt practices to match the functions they need,
- invent new practices when existing ones are insufficient,
- retire practices that no longer serve their context,
- connect patterns across teams and domains,
- and teach others how to reason about cooperation as a system.

Examples of Level 5 behavior:

- designing custom playbooks from multiple frameworks,
- modifying rituals to better fit team maturity and constraints,
- explicitly mapping functions to practices before changing process,
- using cooperation principles to guide organizational change.

Level 5 is not about more practices;
it is about **meta-awareness** of how the system of cooperation works and the ability to evolve it deliberately.

Level 5 answers the question:
“Can this system learn to redesign itself?”

2.3.8 Interpreting and Using the Pyramid

The Pyramid expresses the **Level Rule** of the Human Cooperation System:

> **Level Rule (HCS)**
> You can experiment at any level,
> but you can only **stabilize** cooperation by restoring the **lowest unstable level** first.

In practice, this means:

- **Do not start from Level 4 or 5.**
  Changing practices, tools, or rituals cannot compensate for missing preconditions (Level 1) or unmet human needs (Level 2).
- **Stabilize from the bottom up.**
  If Levels 1–2 are fragile, Level 3 functions will keep degrading, and Level 4–5 improvements will fade or backfire.
- **Expect feedback between levels, not a strict ladder.**
  Insights at Level 5 can refine Level 4; struggles at Level 3 can reveal gaps in Levels 1–2; context shifts can destabilize Level 1.
  The Level Rule does not enforce linear progress — it enforces **dependency awareness**.

When teams repeatedly change practices or tools at Level 4 to avoid addressing deeper issues in Levels 1–3, they are attempting **Encapsulation**.
The Level Rule redirects attention toward **Integration at the depth where instability actually lives**.

- **Separate function from implementation.**
  When a practice fails, first ask which Level 3 function is under strain, and whether Levels 1–2 can support it, before replacing the practice itself.

At its core, the **Level Rule** is a **discipline of attention**:

> Look for the **lowest unstable level** first,
> and restore stability there before expecting higher-level changes to last.

In the broader HCS:

- The **Matrix** describes the detailed cooperation functions that live primarily at Levels 1–3.
- The **Pyramid** explains how these functions depend on one another over time.
- The **Extended Human Dynamics** chapter explains how psychological, political, and relational forces can erode or distort these levels, even when the structure looks correct.
- The **System Modes** describe how to work with this whole system over time — designing it, stabilizing it, growing it, addressing conflict, or resetting it.

Taken together, they let you move from:

> “We need better practices”

to the more precise and systemic question:

> **“Which level of cooperation is unstable, and what must be restored or supported there first?”**

3. EXTENDED HUMAN DYNAMICS

3.1 Introduction & Purpose

The **Human Cooperation System (HCS)** defines the structural core of how cooperation becomes possible.
The **Core Model** explains the minimum set of conditions and needs that make collaborative work stable, observable, and governable.

However, real cooperation is rarely shaped by structure alone.

Teams operate within **human, psychological, and political realities** that sit outside the minimal model, yet strongly influence how cooperation is experienced day to day.
These forces do not change the "physics" of HCS, but they do change its **behavior under pressure**:

- they can distort otherwise sound structures
- they can push systems toward Encapsulation when Integration is needed
- they can keep teams stuck in conflict or avoidance, even when the work is clear

This section exists to make those forces visible.

3.1.1 Purpose of the Extended Human Dynamics Section

The Extended Human Dynamics section introduces an **expanded collection of conditions and needs** that influence cooperation but are not required for the Core Model itself.

Its purpose is to:

- Recognize why cooperation can be difficult, even when the structural system is in place.
- Provide language for diagnosing human and political factors without assigning blame.
- Differentiate between **collective** and **individual** influences to avoid wrong-level interventions.
- Show how psychological and political vectors can distort, amplify, or suppress core HCS conditions and needs.
- Offer practitioners a way to integrate leadership tools, coaching practices, and cultural diagnostics *without* expanding or diluting the Core Model.
- Help teams decide when a problem requires **individual support**, **collective renegotiation**, or **structural correction**.

Extended Human Dynamics does not add "new laws" to HCS.
It explains why systems that are structurally sound on paper may still fail in practice.

3.1.2 Who This Section Is For

This material supports roles who work at the intersection of people and systems:

- Project and team leads navigating recurring interpersonal friction.
- Engineering managers and directors responsible for team health and culture.
- Consultants, coaches, and facilitators diagnosing systemic dysfunctions.
- Decision-makers responding to political tension, conflict, or misalignment.
- Anyone trying to understand why cooperation "feels wrong" even when work is planned and structured correctly.

3.1.3 How This Section Relates to the Core Model

The Extended Human Dynamics section **does not modify or replace the HCS Core Model**.
The Matrix and Pyramid remain the authoritative representation of the cooperation system:

- The **Matrix** defines the structural relationships between conditions, needs, and functions.
- The **Pyramid** defines how these elements depend on each other across levels.

Extended Human Dynamics adds:

- A **collective vs individual** classification of extended conditions and needs.
- **Four layers** of human dynamics: contextual, relational, structural, developmental.
- A way to examine each through **psychological** and **political** impact vectors.
- A mapping to existing leadership and motivation practices (e.g., DiSC, SCARF, Moving Motivators, Situational Leadership), clarifying where and why these tools matter.

In short:

Core Model = what cooperation requires.
Extended Human Dynamics = what cooperation must navigate.

These dynamics are especially relevant in **Stabilization**, **Growth**, and **Conflict** Modes, where human factors often determine whether structural interventions can succeed.

3.1.4 How to Use This Section

1. **Start with the Core Model.**
   Use the Matrix and Pyramid to identify what is structurally missing or misaligned.
2. **If structure is coherent but friction persists, extend the analysis.**
   Switch to Extended Human Dynamics to examine conditions and needs through **collective** and **individual** lenses.
3. **Identify psychological or political vectors affecting the situation.**
   Look for forces that amplify, distort, or suppress otherwise sound cooperative structures.
4. **Choose the correct intervention level.**
5. Individual coaching or support
6. Team agreements or renegotiation
7. Structural adjustment or boundary correction
8. Escalation to Conflict or Reset Mode when needed
9. **Use referenced practices intentionally, not by default.**
   This section clarifies when tools like DiSC or SCARF help — and when they distract from systemic issues that belong in the Core Model or System Modes.

For combining **Core** and **Extended** insights in a concrete way, see the **Integration Guide** in the System Modes section.

3.1.5 Scope of This Section

This section acknowledges human complexity without attempting to quantify, codify, or control it.
It provides a vocabulary and diagnostic aid for:

- understanding why cooperation is hard,
- recognizing non-structural influences,
- avoiding misdiagnosis and misplaced solutions,
- and bringing difficult psychological or political dynamics into safe discussion.

It is optional but highly recommended for practitioners who operate in real organizational environments where human dynamics shape every outcome.

3.2 Extended Conditions

3.2.1 Purpose of Extended Conditions

The **Core Model** defines the minimum structural conditions that make cooperation possible:
Common Purpose, Interdependence, Communication, Trust, and Change / Uncertainty tolerance – plus the human needs that interact with them in the Matrix and Pyramid.

In practice, cooperation is also shaped by a wider set of **contextual, relational, structural, and developmental** influences.
These are the **Extended Conditions**.

Extended conditions:

- do **not** change the Core Model or its Matrix,
- are **not required** for cooperation to exist at all,
- but strongly affect **how cooperation feels**, how it behaves under stress, and how fragile or resilient it becomes.

They help explain why a system that looks "correct" on paper may still be experienced as unfair, unsafe, or exhausting in reality.

Extended conditions are described along two lenses:

- **Collective vs Individual**
- **Political and Psychological impacts**

They are most relevant when deciding whether a challenge calls for **structural changes**, **collective renegotiation**, or **individual support**.

3.2.2 Collective vs Individual Conditions

Extended conditions arise at two intertwined levels:

Collective Conditions

System-wide patterns that shape the environment in which cooperation happens:

- Cultural norms and shared narratives
- Incentive structures and governance mechanisms
- How decisions are made and communicated
- How stable or volatile constraints appear to be

Collective conditions influence **what is possible** in the system.

Individual Conditions

The personal experience of living inside those collective patterns:

- How predictable the environment feels
- How safe it feels to speak, disagree, or experiment
- How much control one feels over daily work
- How much one identifies with the group or mission

Individual conditions influence **what is experienced** by each person.

Both levels matter:

- Changing only individual attitudes without addressing collective conditions leads to burnout and self-blame.
- Changing only collective structures without addressing individual experience leads to quiet disengagement.

Extended conditions help practitioners keep both in view.

3.2.3 Types of Extended Conditions

Extended conditions are grouped into four types that frequently show up in cooperation challenges.
They are **not new dimensions** of the Core Model – they are lenses to understand how the system behaves in real organizations.

Each type includes **collective** and **individual** variants, plus typical **political** and **psychological** impacts.

Contextual Conditions

These describe the broader environment surrounding cooperation.

COLLECTIVE CONTEXTUAL CONDITIONS

- Cultural climate and day-to-day tone
- Communication norms across roles and levels
- Predictability of governance and decision patterns
- Transparency of goals, constraints, and trade-offs
- Stability of priorities, timelines, and commitments

INDIVIDUAL CONTEXTUAL CONDITIONS

- Perceived stability of workload and expectations
- Personal clarity about “what matters now”
- Ability to anticipate changes that affect one’s work
- Sense of control over environment, tools, and schedule

Political impact:
Who sets the context, whose priorities define “reality,” how visible or opaque decisions are.

Psychological impact:
Uncertainty, anxiety, vigilance, or ease; perceived safety in the face of change.

Relational Conditions

These describe how people relate, interpret, and respond to one another.

COLLECTIVE RELATIONAL CONDITIONS

- Norms of reciprocity, fairness, and repair
- Predictability of behavior across roles and teams
- Accepted ways of handling disagreement and escalation
- Shared vs fragmented interpretation of problems and success

INDIVIDUAL RELATIONAL CONDITIONS

- Default trust posture toward peers and leaders
- Openness vs self-censorship in communication
- Capacity for empathy and perspective-taking
- Emotional responses to tension, conflict, or critique

Political impact:
Alliances, informal influence, exclusion, hidden agendas, and “who speaks for whom.”

Psychological impact:
Fear of loss, defensiveness, shame, guilt, emotional contagion, or genuine solidarity.

Structural Conditions

These describe how formal and informal structures support or constrain cooperation beyond the Core Model's basic boundaries.

COLLECTIVE STRUCTURAL CONDITIONS

- Boundary clarity: who decides, who contributes, who is accountable
- How feedback and progress visibility are built into the system
- Access to resources, tools, and information across roles
- Alignment or mismatch between formal organization charts and actual working patterns

INDIVIDUAL STRUCTURAL CONDITIONS

- Perceived autonomy and real decision authority
- Ability to escalate issues without backlash
- Clarity of one's own role, scope, and dependencies
- Access to information and support needed for daily work

Political impact:
Gatekeeping, veto positions, resource control, "choke points" in the system.

Psychological impact:
Overcontrol, learned helplessness, confusion, or a sense of empowerment and legitimacy.

Developmental Conditions

These describe how the cooperative system – and the people in it – learn and evolve over time.

COLLECTIVE DEVELOPMENTAL CONDITIONS

- Learning rhythms (retrospectives, reviews, open dialogue)
- How the organization remembers and reuses learning
- Adaptability to changing constraints and opportunities
- Evolution of shared narratives, identity, and rituals

INDIVIDUAL DEVELOPMENTAL CONDITIONS

- Personal reflection and self-assessment habits
- Curiosity, openness to feedback, and willingness to experiment
- Ability to integrate new insights into behavior and decisions
- Sense of progress, growth, and increasing mastery

Political impact:
Whose learning "counts," whose ideas shape direction, who is allowed to experiment.

Psychological impact:
Pride, motivation, frustration, shame, identity protection, or a sense of stagnation.

3.2.4 Why Extended Conditions Matter

Extended conditions explain **variation in cooperation quality** that the Core Model alone cannot:

- why a structurally sound setup still feels hostile or fragile,
- why teams drift into avoidance, conflict, or apathy without any obvious change in roles or process,
- why attempts to "fix" cooperation at practice level (Level 4) fail when deeper contextual or relational issues remain untouched.

They help practitioners:

- Distinguish **structural** problems from **cultural** or **interpersonal** ones.
- Avoid mislabeling behavior as “resistance” or “attitude” when it is a rational response to conditions.
- Identify when Encapsulation (narrowing collaboration) is a protective response to unsafe or unpredictable dynamics.
- Decide whether an issue calls for **system redesign**, **collective renegotiation**, **individual support**, or a dedicated **System Mode** (e.g., Conflict or Reset).

Extended conditions make cooperation **legible at a human level** without overloading the Core Model.
They provide the bridge between:

- the structural “physics” of HCS, and
- the lived experience of people working inside real organizations.

The next section on **Extended Needs** builds on this foundation to describe how these conditions interact with deeper human needs, shaping motivation, perception, and behavior over time.

3.3 Extended Needs

Cooperation can be **structurally sufficient** and still feel brittle, unfair, or exhausting.

The **Core Model** defines the minimum cooperative needs required for work to function (Level 2 of the Pyramid):
Shared Understanding, Mutual Commitment, Feedback Loops, Distribution of Roles, Autonomy & Agency.

Extended Needs describe the deeper human and systemic requirements that influence:

- how cooperation is experienced,
- how motivation is sustained, and
- how resilient the system is under stress.

These needs are **not part of the minimal HCS Matrix**.
Cooperation can function without them, but it will often feel strained, transactional, or fragile.

Extended Needs are:

- classified as **Collective** or **Individual**, and
- organized into five **functional categories**.

Each need can be examined through potential **political** and **psychological** impact vectors to understand how it becomes distorted under tension.

Extended Needs complement the **Extended Conditions** by focusing on **what people and groups seek**, not just **what context they inhabit**.

3.3.1 Core vs Extended Needs

It is important to distinguish:

- **Core cooperative needs** (Level 2 of the Pyramid)
- Shared Understanding
- Mutual Commitment
- Feedback Loops
- Distribution of Roles
- Autonomy & Agency

These are the **minimal human requirements** for cooperation to function at all.

- **Extended Needs** (this section)
- Purpose & Direction
- Trust & Safety
- Growth & Evolution
- Recognition & Belonging
- Autonomy & Coherence

These are **deeper motivational needs** that shape the *quality* and *emotional tone* of cooperation.

Core Needs answer:

> "Do people have enough to participate in cooperation sustainably?"

Extended Needs answer:

> "Does this cooperation feel meaningful, fair, and worth sustaining over time?"

Both are real. Only the first group is encoded in the Core Model.

3.3.2 Collective vs Individual Needs

Extended Needs arise at two connected levels:

- **Collective Needs**
  Shared meaning, fairness, legitimacy, and identity.
  These shape how groups hold purpose together and maintain cohesion during change.
- **Individual Needs**
  Personal motivation, recognition, autonomy, and emotional safety.
  These shape how each person engages with cooperation and interprets their role in it.

Both levels interact continuously:

- When **collective needs** weaken, individuals disengage or protect themselves.
- When **individual needs** are chronically unmet, collective dynamics destabilize.

Extended Needs help practitioners see both levels without collapsing everything into “culture” or “attitude”.

3.3.3 Types of Extended Needs

Extended Needs fall into five categories that influence motivation, coherence, and resilience within cooperative systems.

Purpose & Direction

Needs related to meaning, intention, and contribution.

Collective Needs

- Shared sense of purpose
- Legitimacy of direction and goals
- Relevance of the work to a broader context
- Identity as a group with a coherent mission

Individual Needs

- Personal meaning in the work
- Sense of contribution to something valuable
- Alignment between values and daily actions
- Clarity about “why my role matters”

Political impact:
Who defines purpose, whose interests shape direction, which voices are included or excluded when goals are set.

Psychological impact:
Pride, commitment, and energy when needs are met; disillusionment, cynicism, or loss of meaning when they are not.

Trust & Safety

Needs related to vulnerability, honesty, and perceived security.

Collective Needs

- Systemic fairness (rules apply consistently)
- Transparent communication about risks and constraints
- Predictable commitments and follow-through
- Safety in raising issues or challenging assumptions

Individual Needs

- Psychological safety in conversations and decisions
- Confidence that honesty does not result in punishment
- Predictability in key relationships
- Emotional security when facing uncertainty or change

Political impact:
Fear of retaliation, status risk, exclusion, and invisible power; “safe for some, dangerous for others.”

Psychological impact:
Anxiety, withdrawal, defensiveness, or hypervigilance when safety is low; openness and calm engagement when safety is high.

Growth & Evolution

Needs related to improvement, mastery, and progression.

Collective Needs

- Shared learning rhythms (retrospectives, reviews, open dialogue)
- Integration of insights into future work, not just storing them
- Collective progression toward greater mastery
- Capacity to adapt without losing coherence or identity

Individual Needs

- Skill development and increasing mastery
- Constructive feedback and support
- Opportunities for growth or new challenges
- A sense of becoming “better over time”

Political impact:
Whose growth is prioritized, who gets opportunities, who is “invested in” vs merely “used”.

Psychological impact:
Stagnation, frustration, or quiet resignation when needs are unmet; motivation and engagement when growth is visible and supported.

Recognition & Belonging

Needs related to appreciation, inclusion, and social identity.

Collective Needs

- Culture of appreciation, not just criticism
- Inclusion mechanisms that ensure representation and voice
- Fair allocation of credit and visibility
- Shared rituals that build connection and identity

Individual Needs

- Feeling valued and seen as a person, not just a resource
- Belonging to the group without constant self-protection
- Recognition for contributions and effort
- Acceptance without needing to constantly prove worth or status

Political impact:
Favoritism, visibility bias, gatekeeping, and “inner circles”.

Psychological impact:
Loneliness, envy, shame, loss of identity, or over-identification with status; versus a grounded sense of being part of “us”.

Autonomy & Coherence

Needs related to freedom, agency, and alignment.

Collective Needs

- Coherent decision-making across roles and teams
- Boundary clarity between functions and responsibilities
- Distributed authority that matches real responsibility
- Consistency of decisions with stated principles and strategy

Individual Needs

- Freedom to make informed decisions in one’s domain
- Sense of agency and ownership over outcomes
- Space to act without micromanagement
- Clarity about how personal choices affect the whole system

Political impact:
Overreach, territorialism, centralization of control, and token autonomy.

Psychological impact:
Dependency, helplessness, reactive resistance, or burnout when needs are unmet; grounded confidence and initiative when they are met.

3.3.4 Why Extended Needs Matter

Extended Needs explain variation in **motivation**, **engagement**, and **cooperation quality** that structural models alone cannot capture.

They highlight why teams with the same processes and structures can behave very differently, and why cooperation often degrades slowly rather than collapsing suddenly:

- from committed to compliant,
- from transparent to guarded,
- from adaptive to rigid,
- from “we” to isolated “me”.

Understanding Extended Needs helps practitioners:

- See the difference between **structural alignment** and **human experience**.
- Avoid over-focusing on tools or processes when motivation or meaning is the root issue.
- Detect early signals of disengagement, resentment, or quiet fragmentation.
- Choose the right level of intervention: **individual, relational, collective, or structural**.
- Recognize when political or psychological distortions, not just process gaps, are driving behavior.

Extended Needs make the human dimension of cooperation **explicit** without inflating the Core Model.
They prepare the ground for **System Modes**, where teams decide *how* to act on these insights over time — whether to design, stabilize, grow, resolve conflict, or reset the cooperation system.

4. SYSTEM MODES

4.1 System Modes – Introduction & Purpose

Cooperation is not static.
The same team can experience clear alignment one month, escalating tension the next, and the need for a full reset a year later.

The **Core Model** of HCS describes what cooperation requires.
The **Extended Human Dynamics** section describes what cooperation must navigate.

The **System Modes** section describes **how to act on this system over time**.

System Modes define distinct **operational stances** for working with cooperation:

- how we design it
- how we stabilize it
- how we grow it
- how we handle conflict
- how we reset it when the context has fundamentally changed

They turn HCS from a static model into a **longitudinal operating system**.

4.1.1 Why System Modes Exist

Even with a clear Matrix and Pyramid, practitioners still face a practical question:

> "Given the state of this team or relationship **right now**,
> what kind of intervention makes sense, and what should we avoid?"

Without a mode-based lens, organizations tend to:

- optimize practices (Level 4) when foundations (Levels 1–2) are unstable
- avoid conflict until it becomes destructive
- try to "grow" a system that is still in basic stabilization
- continue operating under a contract that is effectively obsolete

System Modes provide **named patterns** for these situations.
They ensure that actions respect:

- the **dependency order** of the Pyramid (Level Rule), and
- the difference between **structural** and **human** dynamics.

4.1.2 What a Mode Is (and Is Not)

A **System Mode** is:

- a **dominant systemic activity** required for stability at a given moment,
- a lens for choosing **what to prioritize** and **what to postpone**,
- a way to coordinate multiple roles around the **same intent** for intervention.

A mode is **not**:

- a maturity level (teams can move between modes non-linearly),
- a lifecycle phase with fixed duration (modes can recur or overlap),
- a personality label for teams ("we are a Growth Mode team").

Modes describe **what the cooperation system needs from us now**, not who we are.

4.1.3 The Five System Modes at a Glance

HCS defines five primary modes:

Setup Mode – Design & Preconditions

- **Core question:** *"What must be true before we start?"*
- **Focus:** Establishing preconditions (Level 1) and core cooperative needs (Level 2).
- **Typical use:** New engagements, major re-charters, post-reset rebuilds.
- **Risk if skipped:** The system starts with "cooperation debt" – unstable by design.

Stabilization Mode – Repair & Calibration

- **Core question:** *"Where is cooperation failing, and how do we restore basic stability?"*
- **Focus:** Identifying the **lowest unstable level** and repairing core functions (Level 3) and their prerequisites.
- **Typical use:** Early friction, recurring breakdowns, chronic misalignment, low trust.
- **Risk if skipped:** Teams normalize dysfunction and escalate problems upward instead of fixing them.

Growth Mode – Optimization & Extension

- **Core question:** *"How do we increase autonomy, speed, and learning without losing coherence?"*
- **Focus:** Strengthening and extending existing functions; expanding boundaries safely.
- **Typical use:** Stable teams aiming for higher performance, resilience, or scope.
- **Risk if misapplied:** Adding complexity to a system that is not yet stable.

Conflict Mode – Safety & Realignment

- **Core question:** *"How do we address tension and conflict without causing further harm?"*
- **Focus:** Restoring psychological safety, re-aligning expectations, repairing relationships and trust.
- **Typical use:** Escalated tensions, repeated personal conflicts, breakdowns in trust or legitimacy.
- **Risk if avoided:** Hidden conflict shifts into sabotage, disengagement, or forced exits.

Reset Mode – Existential Re-Evaluation

- **Core question:** *"Given what has changed, does the existing cooperation system still make sense?"*
- **Focus:** Acknowledging that previous assumptions, contracts, or structures are no longer valid.
- **Typical use:** Strategy shifts, major organizational change, funding or market shocks, long-term accumulated damage.
- **Risk if postponed:** Everyone continues to operate inside a contract that no longer exists in reality.

4.1.4 How Modes Relate to the Core and Extended Models

System Modes operate **under** the constraints of the Core Model and Extended Human Dynamics:

- The **Matrix** and **Pyramid** define what must be true and in what order.
- **Extended Conditions** and **Extended Needs** explain why cooperation feels the way it does.
- **System Modes** decide **where to focus attention now** and **what kind of work** is appropriate.

Modes do not change the underlying "physics" of cooperation.
They decide how to work with those physics **given the current state** of the system.

4.1.5 How to Use This Section

1. **Identify the current dominant need.**
   Are you setting up, stabilizing, growing, handling conflict, or facing a potential reset?
2. **Switch explicitly into that mode.**
   Align key participants on *which mode you are in* and *what that implies* for priorities.
3. **Use the Core Model as your map.**
   Within each mode, reason about levels and functions using the Matrix and Pyramid.
4. **Use Extended Human Dynamics to understand resistance and distortion.**
   Where structure seems correct but cooperation feels wrong, look at Extended Conditions and Needs.
5. **Use the Diagnostics section as a shared navigation pattern.**
   The **Diagnostics** section (starting with the Integration Guide) describes how to combine Core Model, Extended Dynamics, and System Modes into a coherent diagnostic and intervention flow.

Subsequent chapters describe each mode in more detail.
They do not define new theory; they explain **how to apply the Human Cooperation System** when the work requires design, stabilization, growth, conflict handling, or reset.

4.2 Setup Mode – Design & Preconditions

Setup Mode is the **architectural stance** of HCS.

It focuses on **what must be true before we start cooperating**, so that the system does not begin with hidden "cooperation debt."

Where the Core Model defines what cooperation *requires*,
Setup Mode asks:

> "Given this specific context and people,
> **how do we make those requirements explicit and agreed** before work begins?"

Setup Mode is about **designing preconditions and expectations**, not about fixing existing problems.

4.2.1 When Setup Mode Is Active

You are in Setup Mode when:

- A **new collaboration** is forming (project, initiative, partnership, team).
- An existing collaboration is being **significantly redefined** (new scope, new leader, new vendor, major restructuring).
- A previous collaboration was effectively **reset**, and you are consciously rebuilding.
- People are asking questions like:
- "Who is actually responsible for what?"
- "How will we decide and escalate?"
- "What does 'success' mean for each side?"
- "What is in and out of scope for this relationship?"

If the system already shows recurring friction, conflict, or fatigue, you are likely beyond pure Setup and closer to **Stabilization** or **Reset**.
Setup Mode works best **before** problems accumulate.

4.2.2 Core Objectives of Setup Mode

Setup Mode has three primary objectives:

1. **Establish Preconditions (Level 1)**
   Make the foundational conditions of cooperation **explicit and shared**:
2. Why we are cooperating together (Common Purpose).
3. How we depend on each other (Interdependence).
4. How we will communicate (Communication).
5. What trust will be based on (Trust).
6. How we expect change and uncertainty to be handled (Change / Uncertainty).
7. **Define Core Cooperative Needs (Level 2) in This Context**
   Translate abstract cooperative needs into **concrete agreements**:
8. What Shared Understanding looks like in practice.
9. What Mutual Commitment means for each party.
10. Which Feedback Loops we will rely on.
11. How roles and responsibilities will be distributed.
12. Where autonomy is desired and where constraints are necessary.
13. **Select an Initial Cooperation Shape**
    Decide **where to encapsulate and where to integrate**:
14. Which work can be modularized with clear contracts and handoffs.
15. Which work requires shared sense-making and joint decisions.
16. How boundaries, decision rights, and interfaces will be defined.

The outcome of Setup Mode is a **shared cooperation design** – not just a scope statement or project plan.

4.2.3 Core Model Focus in Setup Mode

Setup Mode works mainly on **Levels 1 and 2** of the Pyramid, with a light touch on Level 3:

- **Level 1 – Preconditions for Cooperation**
- Clarifying purpose for all parties.
- Making interdependence visible, not assumed.
- Choosing communication channels and rhythms.
- Stating the basis of trust (what we will rely on each other for).
- Agreeing how change will be handled, not just how to proceed if all goes well.
- **Level 2 – Core Human Needs for Cooperative Work**
- Agreeing what “enough Shared Understanding” means before starting.
- Making Mutual Commitment explicit and reciprocal.
- Defining Feedback Loops early (what we will measure, how we will talk about it).
- Outlining Distribution of Roles and how this will evolve.
- Naming where Autonomy & Agency are required and where they are constrained.
- **Level 3 – Cooperative System Functions** (lightly)
  Setup Mode does not fully design all functions, but it should:
- Decide how **Problem Discovery** will happen (who frames the problem and how often).
- Sketch how **Planning & Prioritization** will be shared between parties.
- Agree on basic **Monitoring & Feedback** mechanisms (rhythms, data, conversations).

Setup Mode respects the **Level Rule** by **not jumping directly to practices** at Level 4.
Method choices and tools can be deferred until the foundations are clear.

4.2.4 Extended Dynamics in Setup Mode

Setup Mode is also the moment when **Extended Conditions and Extended Needs** can be addressed before they become invisible “politics” later.

Common **Extended Conditions** to consider:

- **Contextual:**
- What constraints, risks, or pressures are already shaping this cooperation?
- What is the wider organizational or market context?
- **Relational:**
- What prior history exists between the people or organizations involved?
- Are there existing alliances, expectations, or unresolved tensions?
- **Structural:**
- Where are the natural gatekeepers (access, decisions, resources)?
- How will other teams or departments indirectly affect this cooperation?
- **Developmental:**
- Is this a first-time collaboration or a repeated pattern?
- What has been learned from similar past efforts?

Common **Extended Needs** to surface:

- **Purpose & Direction:**
- What does “success” look like for each side?
- What hidden objectives (career, strategy, reputation) might be present?
- **Trust & Safety:**
- What would make this cooperation feel safe or unsafe?
- How will we raise concerns without causing unnecessary escalation?
- **Autonomy & Coherence:**
- Where does each side need freedom, and where do we need tight alignment?
- How do local choices remain coherent with the overall purpose?

Setup Mode is the best time to **name these topics explicitly**, before they crystallize into unspoken assumptions.

4.2.5 What to Prioritize in Setup Mode

In Setup Mode, prioritize:

- **Clarity over speed**
  Spend effort on shared understanding of purpose, constraints, and expectations.
- **Integration where it matters**
  Decide which parts of the work require shared sense-making, not just handoffs.
- **Realistic commitments**
  Make commitments that reflect constraints, risks, and interdependence, not idealized plans.
- **Explicit boundaries**
  Define who decides what, where the edges of responsibility are, and how to handle overlap.
- **Early feedback channels**
  Commit to how you will detect misalignment or emerging friction and how you will respond.

4.2.6 What to Avoid in Setup Mode

Avoid:

- **Rushing to practices and tools (Level 4)**
  Selecting frameworks, ceremonies, or tooling before foundations are clear.
- **Assuming alignment on purpose**
  Treating a shared project label or contract as proof of shared meaning.
- **Ignoring extended dynamics**
  Pretending politics, history, or power asymmetries do not exist.
- **Over-encapsulation**
  Designing everything as a clean interface when some areas clearly require integration.
- **Over-promising and back-loading risk**
  Agreeing to outcomes and timelines that depend on assumptions you have not tested or discussed.

4.2.7 Mode Transitions

Typical transitions **into** Setup Mode:

- New initiative, engagement, or partnership.
- Major change in scope, leadership, or environment that makes the old design obsolete.
- Entering Reset Mode and deciding to rebuild.

Typical transitions **out of** Setup Mode:

- Into **Stabilization Mode**, once work starts and real-world friction appears.
- Into **Growth Mode**, if early cooperation is smooth and foundations prove robust.

Unhealthy patterns:

- Skipping Setup Mode entirely and jumping straight into execution.
- Treating Setup as a one-time ceremony instead of something that can be revisited when context or assumptions change.

4.2.8 Summary

Setup Mode gives cooperation a **deliberate starting shape**.
It does not guarantee the absence of problems, but it ensures that:

- when problems appear,
- they appear **against a clear, shared baseline**
- rather than inside a fog of unspoken expectations.

4.3 Stabilization Mode – Repair & Calibration

Stabilization Mode is the **mechanic stance** of HCS.

Where Setup Mode designs the initial shape of cooperation, Stabilization Mode asks:

> "Given how things are actually working now,
> **where is cooperation failing and how do we restore basic stability?**"

It focuses on **repairing and calibrating** the system so that everyday cooperation no longer depends on constant heroics, escalation, or informal workarounds.

Stabilization Mode is not about optimization or growth.
Its goal is to reach a point where **normal work can proceed reliably**.

4.3.1 When Stabilization Mode Is Active

You are in Stabilization Mode when:

- The cooperation system exists and is already in motion.
- Friction, confusion, or breakdowns are **recurring**, not occasional.
- People are saying things like:
- "We keep having the same issue over and over."
- "We agreed on this, but everyone interprets it differently."
- "We spend more time resolving misunderstandings than doing the work."
- "Nobody is sure who should decide or fix this."

Typical entry signals:

- Misaligned expectations between client and vendor, or between business and engineering.
- Chronic rework, stalled decisions, or constantly shifting priorities.
- Conflicting interpretations of scope, roles, or success criteria.
- Growing distrust, but not yet open hostility or escalation (that's where Conflict Mode may be needed).

If cooperation is still being designed, you are closer to **Setup Mode**.
If tensions are high, personal, and emotionally loaded, **Conflict Mode** may be more appropriate.

4.3.2 Core Objectives of Stabilization Mode

Stabilization Mode has three main objectives:

1. **Identify the Lowest Unstable Level**
   Use the Matrix and Pyramid to locate where instability starts:
2. Are Preconditions (Level 1) unclear or contested?
3. Are Core Cooperative Needs (Level 2) unmet?
4. Are Cooperative Functions (Level 3) missing, distorted, or overloaded?
5. **Restore Basic Reliability of Cooperation**
   Bring the system to a point where:
6. People know why they are working together and what “good” means.
7. Dependencies and roles are clear enough to coordinate.
8. Feedback loops are working and safe to use.
9. Trust is at least sufficient for normal risk-taking.
10. **Reduce Dependence on Manual Intervention**
    Shift from:
11. constant escalation, heroic problem solving, and ad-hoc fixes
    toward:
12. predictable routines, clear decision paths, and stable agreements.

Once these objectives are met, the system is ready for **Growth Mode** or, if conditions change, for deliberate **Reset**.

4.3.3 Core Model Focus in Stabilization Mode

Stabilization Mode is where the **Level Rule** is most actively applied.

- **Primary focus: Levels 1–3 of the Pyramid**
- **Level 1 – Preconditions for Cooperation**
  - Check for clarity and alignment on purpose, interdependence, communication basics, trust foundations, and how change is handled.
  - Typical issues: hidden assumptions about “why we’re here”, invisible dependencies, mismatched expectations of responsiveness or availability.
- **Level 2 – Core Human Needs for Cooperative Work**
  - Examine Shared Understanding, Mutual Commitment, Feedback Loops, Distribution of Roles, Autonomy & Agency.
  - Typical issues: different interpretations of the same goal, one-sided commitment, feedback as performance judgment only, vague or overlapping roles, blocked autonomy.
- **Level 3 – Cooperative System Functions**
  - Look at the 25 Matrix functions: where they are missing, weak, or overloaded.
  - Typical issues: poor Problem Discovery, weak Planning & Prioritization, absent Monitoring & Feedback, unclear Enablement & Empowerment, weak Adaptation & Learning.
- **Secondary focus: Level 4 (Practices & Frameworks)**

Stabilization Mode may adjust or simplify practices, but **only after** the lower levels are understood.
The intent is: - not to “install a new framework”,
- but to ensure that existing or chosen practices **actually serve the needed functions**.

Level 5 (Meta-Practices & Innovation) is generally **not the focus** in Stabilization Mode.
Reflection may happen, but the priority is **reliability**, not redesigning the entire system.

4.3.4 Extended Dynamics in Stabilization Mode

Stabilization Mode is where **structural issues and human experience intersect most visibly**.

Typical **Extended Conditions** to examine:

- **Contextual**
- Are priorities changing too fast to stabilize anything?
- Do different groups live in different perceived realities (what is urgent, what is safe to ignore)?
- **Relational**
- Are misunderstandings turning into narratives (“they never listen”, “they always change their mind”)?
- Are there patterns of avoidance, blame, or quiet resentment?
- **Structural**
- Are there hidden gatekeepers or veto points that constantly slow work?
- Are some roles carrying invisible coordination work that is not recognized?
- **Developmental**
- Is the system trying to run at “high maturity” patterns without having built the basics?
- Are lessons from past breakdowns actually integrated?

Typical **Extended Needs** that surface:

- **Trust & Safety**
- People may comply but no longer speak honestly.
- Issues are reported late because signaling feels risky.
- **Autonomy & Coherence**
- People feel either micromanaged or abandoned.
- Local decisions conflict with global direction.
- **Recognition & Belonging**
- Some participants feel like “outsiders” whose constraints are ignored.
- Work that keeps the system alive (coordination, translation) is undervalued.

Stabilization Mode uses Extended Dynamics not to psychologize individuals, but to understand **why the structure is experienced the way it is**, and why certain functions are resisted or overloaded.

4.3.5 What to Prioritize in Stabilization Mode

Prioritize:

- **Finding the lowest unstable level**
- Ask: “Where does this actually start?”
- Address Level 1 issues before expecting Level 3 functions to stabilize.
- **Clarifying expectations and commitments**
- Make implicit assumptions explicit.
- Document agreements in accessible, lightweight form.
- **Rebuilding basic trust in the system**
- Follow through on small, visible commitments.
- Create safe, bounded spaces to surface misalignments and past disappointments.
- **Simplifying where needed**
- Remove or pause practices that add complexity without solving the core problem.
- Reduce the number of parallel “improvement initiatives” until stability is restored.
- **Restoring working feedback loops**
- Make it normal and safe to signal risks, misalignment, and confusion early.
- Ensure that feedback leads to adjustment, not punishment or endless discussion.

4.3.6 What to Avoid in Stabilization Mode

Avoid:

- **Jumping straight to new practices or tools**
- Replacing frameworks, tooling, or ceremonies without addressing underlying levels.
- **Over-optimizing while unstable**
- Introducing “advanced” practices (e.g., complex OKR systems, intricate PI planning) while basic coordination is still failing.
- **Personalizing systemic issues**
- Blaming individuals or teams for behaviors that are rational responses to unclear structure or unsafe conditions.
- **Overusing escalation**
- Relying on escalation as the primary way to get things done, instead of stabilizing normal pathways.
- **Pretending everything is fine**
- Forcing positivity or “solution talk” when people have not yet seen that their real concerns are understood.

Stabilization Mode requires **honest acknowledgment** of what is not working, without sliding into blame.

4.3.7 Mode Transitions

Typical transitions **into** Stabilization Mode:

- After Setup, once real work starts and gaps become visible.
- From Growth, when extending autonomy or speed reveals structural cracks.
- From Conflict, after de-escalation, when the system needs to be rebuilt with clearer agreements.
- From Reset, when a new design is in place but must be tested and calibrated.

Typical transitions **out of** Stabilization Mode:

- Into **Growth Mode**, when cooperation is reliable enough to optimize and extend.
- Into **Conflict Mode**, if attempts to stabilize reveal deep, unresolved tensions.
- Into **Reset Mode**, if stabilization repeatedly fails because the underlying contract or context is no longer valid.

Unhealthy patterns:

- Remaining in permanent “almost stable” mode without ever addressing foundational issues.
- Cycling through tools and frameworks instead of examining conditions, needs, and functions.
- Using Stabilization as a way to avoid naming necessary Conflict or Reset.

4.3.8 Summary

Stabilization Mode returns cooperation to a state where **normal work can proceed reliably**, instead of depending on escalation, heroics, or constant workaround.

It does not promise high performance or rapid growth.
Its job is to ensure that:

- recurring issues are traced back to the **lowest unstable level**, not treated as isolated incidents,
- the basics of cooperation (purpose, roles, expectations, feedback) are **clear enough to trust**,
- people can **signal problems early** without fear of punishment or futility,
- everyday coordination no longer relies on a few individuals compensating for systemic gaps,
- the system is ready either to **grow** (by extending autonomy and capability) or to **face harder truths** in Conflict or Reset Mode if needed.

When deeper investigation is required, the **Diagnostics** section provides structured ways to walk the Core Model and Extended Human Dynamics – but Stabilization Mode’s core outcome is simple:

> A cooperation system that works **as it is**,
> rather than one that only works when someone is constantly rescuing it.

4.4 Growth Mode – Optimization & Extension

Growth Mode is the **gardener stance** of HCS.

Where Stabilization Mode restores basic reliability, Growth Mode asks:

> "Now that cooperation is stable enough,
> **how do we increase autonomy, speed, and learning without losing coherence or safety?**"

It focuses on **strengthening what already works**, **extending boundaries** where appropriate, and **removing friction** that is no longer necessary.

Growth Mode is not about starting over or firefighting.
It is about **deliberate, sustainable improvement** of a working system.

4.4.1 When Growth Mode Is Active

You are in Growth Mode when:

- The cooperation system is **mostly stable**:
- recurring breakdowns are rare or manageable,
- roles and dependencies are understood,
- work can proceed without constant escalation.
- People are asking questions like:
- "How can we move faster without burning out?"
- "Where can we decentralize decisions?"
- "What can we improve so we're not always at the limit?"
- "How do we learn more from what we're already doing?"

Typical entry signals:

- Stabilization work has reduced visible chaos.
- Teams feel **safe enough** to talk about improvement, not just survival.
- Autonomy is requested from the people doing the work.
- There is appetite to experiment with new practices or tools, not as a rescue, but as an upgrade.

If basic cooperation is still unreliable, you are likely still in **Stabilization Mode**.
If tension and mistrust dominate, **Conflict Mode** may be more appropriate.

4.4.2 Core Objectives of Growth Mode

Growth Mode has three main objectives:

1. **Strengthen Core Functions**
   Identify which cooperative functions (from the Matrix) already work and:
2. remove residual friction around them,
3. make them more robust under load,
4. ensure they do not depend on a few individuals.
5. **Expand Autonomy and Capacity Safely**
   Extend:
6. decision-making closer to where information lives,
7. ownership toward teams and individuals,
8. scope where the system can handle it
   without undermining shared direction or safety.
9. **Build a Stable Learning Rhythm**
   Turn one-off fixes into:
10. repeatable improvement cycles,
11. shared learning across teams,
12. growing confidence in adapting the cooperation system itself.

Growth Mode prepares the ground for healthy use of **Meta-Practices & Innovation** (Level 5) without destabilizing lower levels.

4.4.3 Core Model Focus in Growth Mode

Growth Mode primarily works on **Levels 3–5** of the Pyramid, while respecting the foundations of Levels 1–2.

- **Level 1 – Preconditions for Cooperation** (must remain stable)
- Purpose, interdependence, communication basics, trust, and change tolerance should be clear enough not to require constant renegotiation.
- If Level 1 starts to wobble during Growth, you must temporarily drop back toward **Stabilization Mode**.
- **Level 2 – Core Human Needs for Cooperative Work**
- Growth should **strengthen**, not erode:
    - Shared Understanding (especially as scope expands),
    - Mutual Commitment (to fairness and shared benefit),
    - Feedback Loops (more, not less, visibility),
    - Distribution of Roles (avoiding overload and confusion),
    - Autonomy & Agency (expanded with clear boundaries).
- **Level 3 – Cooperative System Functions**
- Focus on making key functions **more effective and scalable**:
    - **Problem Discovery** – better discovery practices, wider participation.
    - **Planning & Prioritization** – more realistic, dependency-aware planning.
    - **Monitoring & Feedback** – richer, more timely signals; fewer surprises.
    - **Enablement & Empowerment** – ensuring people have what they need to act.
    - **Adaptation & Learning** – faster, safer iteration in response to change.
- **Level 4 – Practices & Frameworks**
- Growth Mode is where you:
    - tune or evolve existing practices,
    - introduce new practices selectively,
    - simplify or retire practices that no longer fit.
- The test of any practice change:
  "Does it strengthen the needed function without destabilizing lower levels?"
- **Level 5 – Meta-Practices & Innovation**
- Growth Mode may cautiously expand Level 5 capabilities:
    - more people involved in designing/improving ways of working,
    - shared language for reasoning about cooperation,
    - increased comfort with experimenting and iterating on the system itself.

The **Level Rule** still applies:
if experimentation at Level 4–5 reveals instability at Level 1–2, Growth Mode should temporarily yield to Stabilization.

4.4.4 Extended Dynamics in Growth Mode

Growth Mode activates a more optimistic side of **Extended Conditions and Extended Needs**, but also introduces new risks.

Common **Extended Conditions**:

- **Contextual**
- More ambitious goals, new opportunities, higher expectations.
- External pressure to “scale” or “move faster”.
- **Relational**
- Increased cross-team interaction, more interfaces, more stakeholders.
- Potential for both stronger collaboration and renewed tension.
- **Structural**
- Changing boundaries, new teams, new capabilities.
- Possible creation of new gatekeepers or bottlenecks if expansion is not deliberate.
- **Developmental**
- Stronger learning rhythms (retros, reviews, post-mortems).
- Opportunity to spread learning horizontally, not just within one team.

Common **Extended Needs** in focus:

- **Growth & Evolution**
- People want visible progress, mastery, and meaningful challenges.
- The organization wants to improve its capacity and reputation.
- **Autonomy & Coherence**
- Teams want more ownership; leadership wants alignment.
- The tension between “freedom” and “togetherness” becomes central.
- **Recognition & Belonging**
- Success introduces questions of who gets credit and who feels left behind.
- New roles and units may shift identity and status dynamics.

Growth Mode should treat these dynamics as **design inputs**, not afterthoughts.

4.4.5 What to Prioritize in Growth Mode

Prioritize:

- **Strengthening what already works**
- Identify stable functions and make them smoother and more resilient.
- Protect the mechanisms (rituals, roles, relationships) that keep them healthy.
- **Explicit expansion of autonomy**
- Deliberately decide where decisions will move closer to the work.
- Clarify new decision rights and responsibilities, not just “trust people more”.
- **Improvement loops, not one-off projects**
- Establish regular rhythms for reviewing cooperation quality, not just output.
- Make improvements small, frequent, and reversible where possible.
- **Guardrails for experimentation**
- Define safe-to-try areas vs. high-risk areas.
- Ensure that experiments do not silently undermine purpose, safety, or fairness.
- **Sharing learning across boundaries**
- When one team finds a better pattern, share the *function* and *conditions* first, not just the practice.
- Let others adapt the practice to their context.

4.4.6 What to Avoid in Growth Mode

Avoid:

- **Scaling instability**
- Expanding scope, autonomy, or speed while unresolved issues from Stabilization linger.
- **Framework tourism**
- Importing new methodologies or practices simply because they are popular, without checking:
    - which function they serve,
    - which levels they touch,
    - and what conditions they assume.
- **Over-loading high performers**
- Relying on the same individuals to drive all improvements and all delivery.
- Growth that depends on a few people is not systemic.
- **Invisible trade-offs**
- Increasing speed at the cost of safety or sustainability.
- Piling on initiatives without removing anything.
- **Treating Growth as permanent mode**
- Growth Mode is not the default of a “good team”.
- Systems cannot be in continuous improvement without periods of stabilization and consolidation.

4.4.7 Mode Transitions

Typical transitions **into** Growth Mode:

- From **Stabilization Mode**, once basic reliability is restored.
- From **Setup Mode**, if early cooperation proves smoother than expected and the system feels ready to extend.
- After **Conflict Mode**, when trust has been repaired enough to talk about improvement, not just harm.

Typical transitions **out of** Growth Mode:

- Back to **Stabilization Mode**, if expansion reveals deeper instability.
- Into **Conflict Mode**, if growth attempts amplify unresolved tensions or perceived unfairness.
- Into **Reset Mode**, if external conditions or strategic shifts make the current growth direction obsolete.

Unhealthy patterns:

- Treating Growth Mode as a badge of honor and refusing to acknowledge when stabilization or conflict work is needed.
- Using “continuous improvement” language to mask chronic instability or overwork.

4.4.8 Summary

Growth Mode helps a cooperation system become **more capable, more autonomous, and more adaptive** without losing its foundations.

It does not guarantee endless acceleration.
Its role is to ensure that:

- improvements are built on **real stability**, not wishful thinking,
- autonomy increases in ways that **preserve coherence and safety**,
- learning becomes a **normal part of work**, not a rare event,
- success does not rest on a few individuals but on the **system as a whole**,
- the organization can recognize when it must pause growth to stabilize, confront conflict, or reset.

When used well, Growth Mode turns cooperation from “barely working” into **increasingly resilient and self-improving**, ready for whatever the next mode demands.

4.5 Conflict Mode – Safety & Realignment

Conflict Mode is the **mediator stance** of HCS.

Where Stabilization Mode focuses on repairing functions and agreements, Conflict Mode asks:

> "Given the tension, mistrust, or hurt we're facing now,
> **how do we restore enough safety and realignment for cooperation to be possible again?**"

It focuses on **making conflict visible, safe to work with, and structurally actionable** – without collapsing into blame or pretending nothing is wrong.

Conflict Mode is not about winning arguments or forcing agreement.
Its purpose is to **protect people and the system** long enough to decide whether to stabilize, grow, or reset.

4.5.1 When Conflict Mode Is Active

You are in Conflict Mode when:

- Tension and frustration are **no longer incidental**, but recurring and emotionally charged.
- The story about the other party shifts from "they are confused" to "they are unreasonable / careless / malicious".
- People are saying or implying things like:
- "I can't trust them anymore."
- "We tried to fix this, but they always go back to the same behavior."
- "I don't feel safe raising this in the open."
- "At this point, I'd rather go around them."

Typical entry signals:

- Repeated escalations with no lasting resolution.
- Meetings where the **real conversation** happens after the meeting, in side channels.
- Individuals showing signs of withdrawal, resentment, or burnout.
- "Us vs them" narratives hardening between roles, teams, or organizations.

If issues are mostly about unclear structure and expectations, you are likely still in **Stabilization Mode**.
If the cooperation contract itself is fundamentally invalidated (e.g., strategy, budget, or purpose has changed drastically), **Reset Mode** may be more appropriate.

4.5.2 Core Objectives of Conflict Mode

Conflict Mode has three main objectives:

1. **Restore Minimum Psychological Safety**
2. Create conditions where people can speak about the conflict
   without fear of retaliation, humiliation, or dismissal.
3. Make it possible to distinguish between:
   - what happened,
   - how it was experienced, and
   - what it means for cooperation.
4. **Clarify What the Conflict Is Really About**
5. Separate:
   - structural issues (roles, decisions, incentives),
   - relational issues (trust, respect, fairness),
   - and individual issues (capacity, fit, behavior).
6. Identify which **levels of the Pyramid** and which **Extended Needs** are actually at stake.
7. **Decide Whether to Repair, Contain, or Reset**
8. Agree on whether:
   - the relationship should be repaired and stabilized,
   - boundaries should be tightened or cooperation reduced,
   - or the cooperation contract should be significantly redefined or ended.
9. Prevent ongoing harm while this decision is being made.

Conflict Mode does **not** assume that reconciliation is always possible or desirable.
It aims for **honest, humane decisions** about what the system can and should sustain.

4.5.3 Core Model Focus in Conflict Mode

Conflict Mode mainly surfaces **Level 1–2** issues that have accumulated over time, often disguised as Level 3–4 problems.

- **Level 1 – Preconditions for Cooperation**
- Conflicts frequently reveal that:
  - purpose is no longer shared (or was never truly aligned),
  - interdependence is resented or denied,
  - communication channels are weaponized or avoided,
  - trust foundations have eroded,
  - change is experienced as unilateral imposition rather than shared reality.
- **Level 2 – Core Human Needs for Cooperative Work**
- In conflict, these needs are often at the heart of the pain:
  - **Shared Understanding** – "We don't even agree on what happened."
  - **Mutual Commitment** – "We are carrying more risk/effort than they are."
  - **Feedback Loops** – "It's not safe to tell the truth."
  - **Distribution of Roles** – "We are treated as less important/less competent."
  - **Autonomy & Agency** – "Decisions that affect us are made without us."
- **Level 3 – Cooperative System Functions**
- Certain functions are usually under severe strain:
  - **Problem Discovery** – the conflict itself is poorly framed.
  - **Monitoring & Feedback** – signals have been ignored or punished.
  - **Enablement & Empowerment** – some parties feel structurally disempowered.
  - **Adaptation & Learning** – old harms repeat because learning is not integrated.

Level 4 and Level 5 (practices and meta-practices) are **not the primary levers** in Conflict Mode.
Retrospectives, workshops, or process changes can support the work, but **they are not the work**.

4.5.4 Extended Dynamics in Conflict Mode

Conflict Mode is where **Extended Conditions and Extended Needs** are most visible and sensitive.

Typical **Extended Conditions**:

- **Contextual**
- External pressure, scarcity, or high stakes increase emotional load.
- Conflicting narratives about what "reality" is (market, risk, importance).
- **Relational**
- Past disappointments and perceived betrayals accumulate into stories.
- Cross-team stereotypes ("they always…", "they never…").
- **Structural**
- Power asymmetries (who can say no, who can walk away).
- Incentives that reward behavior harmful to cooperation.
- **Developmental**
- Old conflicts that were patched but not resolved.
- Patterns of avoidance or forced "harmony" instead of genuine repair.

Typical **Extended Needs** stressed:

- **Trust & Safety**
- Fear of speaking honestly; expectation that vulnerability will be used against someone.
- Sense that rules are not applied fairly.
- **Recognition & Belonging**
- Feeling disrespected, invisible, or treated as “less than”.
- Group identities hardening into camps.
- **Autonomy & Coherence**
- Feeling coerced or trapped in a system one cannot influence.
- Experiencing decisions as arbitrary or incoherent with stated values.

Conflict Mode uses these insights not to pathologize people, but to understand **what it would take for cooperation to stop being experienced as harmful or illegitimate**.

4.5.5 What to Prioritize in Conflict Mode

Prioritize:

- **Stabilizing emotional and psychological safety**
- Create spaces and formats where people can speak without immediate judgment or retaliation.
- Acknowledge impact before debating intent or “the facts”.
- **Naming the conflict clearly**
- Distinguish:
    - events,
    - interpretations,
    - and patterns over time.
- Agree on which issues are systemic and which are situational.
- **Separating layers of the problem**
- Untangle:
    - structural design issues (roles, decisions, incentives),
    - relational issues (trust, fairness),
    - individual issues (behavior, performance).
- Avoid collapsing everything into one explanation (“they are the problem”).
- **Making harm visible**
- Allow people to say what has felt harmful or unfair.
- Document this in a way that can inform structural decisions, not just emotional closure.
- **Exploring options honestly**
- Discuss what repair would require from *each* side.
- Consider containment or separation where repair is unrealistic.

4.5.6 What to Avoid in Conflict Mode

Avoid:

- **Rushing to solution or compromise**
- Quick deals that ignore underlying pain or power asymmetries will not hold.
- Premature “let’s move on” creates deeper resentment.
- **Using process to bury conflict**
- More meetings, rituals, or frameworks that never name the real issue.
- Forcing “professionalism” as a way to suppress emotional reality.
- **Treating conflict as purely interpersonal**
- Blaming individuals for behavior that is shaped by structure, incentives, or leadership signals.
- Sending people to coaching while preserving the system that generates the behavior.
- **Weaponizing the Core Model**
- Using HCS language to score points (“you’re breaking Level 2 again”).
- Turning diagnostics into an argument rather than a shared lens.
- **Turning Conflict Mode into a permanent state**
- Living in constant re-litigation without ever deciding to repair, contain, or reset.

Conflict Mode must be **time-bounded and purpose-driven**:
it is there to help the system decide **what happens next**.

4.5.7 Mode Transitions

Typical transitions **into** Conflict Mode:

- From **Stabilization Mode**, when attempts to fix structure keep surfacing strong emotions, mistrust, or narratives of unfairness.
- From **Growth Mode**, when expansion creates visible winners and losers, and tensions become personal.
- From **Setup or Reset**, when early expectations are violated in ways that feel like betrayal.

Typical transitions **out of** Conflict Mode:

- Into **Stabilization Mode**, when safety is restored enough to work on structure and functions again.
- Into **Growth Mode**, when conflict has been meaningfully addressed and relationships can support higher autonomy or speed.
- Into **Reset Mode**, when conflict reveals that the cooperation contract or context is no longer viable or legitimate.

Unhealthy patterns:

- Cycling between Conflict and Growth without Stabilization – trying to “grow through” unresolved pain.
- Ignoring the need for Reset when both sides know the cooperation should end or radically change.
- Treating Conflict Mode as a sign of failure instead of a necessary function in complex systems.

4.5.8 Summary

Conflict Mode makes cooperation **safe to question**.

It does not guarantee reconciliation or continued partnership.
Its role is to ensure that:

- harm and mistrust are **acknowledged**, not denied or minimized,
- structural, relational, and individual layers of the problem are **distinguished**,
- people can speak honestly **without being punished for it**,
- decisions about repair, containment, or reset are based on **reality, not wishful thinking**,
- the system can either stabilize, grow, or reset **without carrying hidden fractures forward**.

Used well, Conflict Mode turns inevitable tensions into **moments of truth** that either deepen cooperation or clarify that it must change – instead of letting conflict quietly hollow the system from within.

4.6 Reset Mode – Existential Re-Evaluation

Reset Mode is the **renovator stance** of HCS.

Where Stabilization Mode repairs, Growth Mode extends, and Conflict Mode makes tension visible, Reset Mode asks:

> "Given what has changed and what we've learned,
> **does this cooperation system still make sense in its current form – and if not, what needs to end or be rebuilt?**"

It focuses on **acknowledging that the old contract is no longer valid** and **designing the conditions for a different future** – whether together or apart.

Reset Mode is not an admission of failure.
It is a recognition that **context, purpose, or legitimacy have moved on**, and the system must either transform or be consciously closed.

4.6.1 When Reset Mode Is Active

You are in Reset Mode when:

- The **original basis of cooperation has fundamentally changed**:
- Strategy, market, or funding shifts make previous goals obsolete.
- Key stakeholders, sponsors, or decision-makers have changed.
- The intended value is no longer relevant or worth the cost.
- Or, the **relationship itself is no longer workable**:
- Repeated Stabilization and Conflict efforts have not led to sustainable improvement.
- Parties remain technically cooperative but are emotionally or politically disengaged.
- People privately agree: "If we were starting today, we wouldn't design it like this."

Typical entry signals:

- Long-running projects or partnerships that feel "too big to stop" but no longer make sense.
- Initiatives continued out of habit, sunk cost, or fear of political consequences.
- A sense of **stagnation, cynicism, or quiet exit** (people leaving, disengaging, or "ghosting" the work).

If the core purpose is still valid and trust is repairable, you may be in **Stabilization** or **Conflict** rather than Reset.
Reset Mode is appropriate when **continuing "as is" is the more dangerous option**.

4.6.2 Core Objectives of Reset Mode

Reset Mode has three main objectives:

1. **Face Reality About the Current System**
2. Name clearly:
    - what has changed in context and purpose,
    - what is no longer true or legitimate,
    - what damage or fatigue has accumulated.
3. Stop pretending the old system is still appropriate.
4. **Decide What Ends, What Continues, and What Transforms**
5. Clarify:
    - which agreements, roles, and structures should be **ended**,
    - which parts still have value and can be **preserved**,
    - what needs to be **redesigned from first principles**.
6. Avoid “zombie cooperation” – arrangements that exist on paper but not in reality.
7. **Create Conditions for a Clean Start (or Clean Closure)**
8. If cooperation should continue in a different form:
    - define a clear path back to **Setup Mode**.
9. If cooperation should end:
    - close it with clarity and respect, avoiding unnecessary harm and ambiguity.

Reset Mode is about **integrity**: aligning the cooperation system with what is actually true now.

4.6.3 Core Model Focus in Reset Mode

Reset Mode primarily revisits **Level 1 and Level 2** of the Pyramid, with deliberate decisions about what happens to Levels 3–5.

- **Level 1 – Preconditions for Cooperation**
- The central questions:
    - Is there still a **shared, legitimate purpose** for this cooperation?
    - Is the **interdependence** still real, or has it become forced or symbolic?
    - Can we still meaningfully **communicate** as partners?
    - Is there enough **trust** to justify continued cooperation?
    - Has the **change/uncertainty landscape** shifted so much that old assumptions are invalid?

If the answer to these is mostly “no”, remaining in the existing system is often worse than resetting.

- **Level 2 – Core Human Needs for Cooperative Work**
- In Reset Mode, you examine:
  - whether Shared Understanding is possible or constantly dissolves,
  - whether Mutual Commitment is genuinely reciprocal or deeply unbalanced,
  - whether Feedback Loops are functional or systematically ignored,
  - whether Distribution of Roles remains legitimate,
  - whether any real Autonomy & Agency is left on both sides.
- **Levels 3–5 – Functions, Practices, Meta-Practices**
- Reset Mode does **not** try to optimize these.
- It asks which functions and practices:
  - should be **retired** with the system,
  - can be **salvaged** and reused elsewhere,
  - reveal insights that should inform future Setup and Stabilization.

The **Level Rule** is invoked here in its strongest form:
if Level 1 is fundamentally broken, working harder on Levels 3–4 is a way of avoiding reality.

4.6.4 Extended Dynamics in Reset Mode

Reset Mode operates in a landscape of **strong Extended Conditions and Extended Needs**, often accumulated over time.

Typical **Extended Conditions**:

- **Contextual**
- Market shifts, mergers, reorganizations, funding cuts, regulatory changes.
- New strategic priorities that make the old cooperation peripheral or obstructive.
- **Relational**
- History of unresolved conflicts, broken commitments, or repeated disappointments.
- Loss of goodwill, even if people remain “professional”.
- **Structural**
- Power shifts: who can exit, who is locked in, who decides to end or renew the contract.
- Misaligned incentives that reward continuation even when it makes no sense.
- **Developmental**
- The system has reached the end of a natural lifecycle.
- Attempts to “revive” it feel more like resuscitating something that wants to end.

Typical **Extended Needs** in focus:

- **Purpose & Direction**
- People need clarity about **why** any continuation or termination is happening.
- They need to see that decisions match stated values.
- **Trust & Safety**
- People want to know they will not be punished for acknowledging reality.
- They need assurances about how changes will affect them.
- **Recognition & Belonging**
- There is a need to honor contributions and effort, even if the outcome is closure.
- Individuals and teams may fear loss of identity or status tied to the existing system.

Reset Mode treats these needs as **central**, not peripheral, to ending or transforming cooperation responsibly.

4.6.5 What to Prioritize in Reset Mode

Prioritize:

- **Telling the truth about viability**
- Clearly state when the existing cooperation no longer serves its intended purpose.
- Share assessments and constraints transparently with key stakeholders.
- **Honoring what has been invested**
- Acknowledge the work, learning, and relationships built.
- Distinguish between **failed outcomes** and **valuable efforts**.
- **Designing a deliberate ending or restart**
- Define:
    - what exactly stops and when,
    - what transitions to a different form,
    - what must be kept running temporarily to avoid harm.
- **Protecting people during transition**
- Consider psychological and political impact:
    - How will people be informed?
    - Who might feel blamed or abandoned?
    - How can you avoid unnecessary stigma or shame?
- **Capturing learning for future Setup and Stabilization**
- Document key insights:
    - what worked,
    - what failed,
    - what assumptions proved wrong.
- Feed this back into **Setup Mode** for any new cooperation that follows.

4.6.6 What to Avoid in Reset Mode

Avoid:

- **Silent decay**
- Letting the system die informally without explicit decisions, leaving people confused and stuck in limbo.
- **Sunk-cost insistence**
- Continuing cooperation purely because “we’ve already invested so much”.
- **Cosmetic rebranding**
- Renaming or relabeling the system without changing purpose, structure, or relationships.
- **Scapegoating**
- Blaming individuals or one party for a situation driven by context shifts or joint design choices.
- **Rushing into a new Setup without integration**
- Jumping straight into a new cooperation design without integrating what the old one taught you.

Reset Mode is about **clean decisions**, not cosmetic change.

4.6.7 Mode Transitions

Typical transitions **into** Reset Mode:

- From **Growth Mode**, when expansion runs into strategic or contextual walls that cannot be worked around.
- From **Conflict Mode**, when attempts at repair reveal that continued cooperation would be harmful or pointless.
- From **Stabilization Mode**, when repeated attempts to stabilize fail because the underlying purpose or context has fundamentally shifted.

Typical transitions **out of** Reset Mode:

- Into **Setup Mode**, when a new or radically redesigned cooperation is justified.
- Into **no cooperation**, when the best option is to stop working together in this domain.
- Into **limited, re-scoped cooperation**, with much narrower boundaries and expectations.

Unhealthy patterns:

- Treating Reset as a purely technical or contractual event, ignoring emotional and relational impact.
- Cycling between Growth and Stabilization to avoid acknowledging the need for Reset.
- Treating Reset as a “failure to hide”, instead of a necessary adaptation to reality.

4.6.8 Summary

Reset Mode allows a cooperation system to **end or transform with integrity**.

It does not guarantee that everyone will be happy with the outcome.
Its role is to ensure that:

- the decision to continue, transform, or end cooperation is based on **current reality**, not past intent,
- foundational conditions (purpose, interdependence, trust) are not faked or assumed,
- people are **not trapped** in obsolete or harmful arrangements out of habit or fear,
- valuable learning and relationships are **carried forward consciously**,
- any new cooperation starts from **Setup Mode**, not from the unspoken debris of the old system.

Used well, Reset Mode prevents organizations and teams from living in “zombie cooperation” –
freeing energy and attention for systems that are genuinely alive, needed, and possible.

5. DIAGNOSTICS

5.1 Integration Guide

The **Core Model** defines the structural foundation of cooperation.
Extended Human Dynamics describes the human and political forces that shape how cooperation is experienced.
System Modes define how we act on the system over time.

The Diagnostics section — starting with this Integration Guide — explains how to use these three parts of HCS **together** in a coherent way.

It is not a new model.
It is a **navigation pattern** that keeps:

- the Core Model clean and universal, and
- Extended Dynamics grounded in structure rather than personality or vague “culture”,
- System Modes anchored in real observations rather than wishful thinking.

The goal is to make cooperation **understandable and actionable** without oversimplifying human systems.

5.1.1 The Integration Principle

At the highest level:

- Use the **Core Model** to diagnose **structural gaps**.
- Use **Extended Human Dynamics** to diagnose **human and political distortions**.
- Use **System Modes** to choose **what kind of work** is appropriate right now.

All three must be considered, but never confused.

If you treat a structural issue as a purely human one, you drift into blaming individuals for systemic design choices.
If you treat a human or political issue as purely structural, you redesign processes while leaving the real tension untouched.
If you ignore mode, you try to **grow** when you should **stabilize**, or **stabilize** what should be **reset**.

The Integration Principle keeps these dimensions **separate but connected**.

5.1.2 What Each Layer Is Responsible For

Core Model (Matrix + Pyramid)

Use the Core Model to reason about:

- **What cooperation requires** (conditions, needs, functions).
- **How these requirements depend on each other** (levels and Level Rule).
- **Where in the system** a breakdown is happening structurally.

Typical questions:

- Which **conditions** are missing or unstable (Common Purpose, Interdependence, Communication, Trust, Change/Uncertainty)?
- Which **core needs** are not being met (Shared Understanding, Mutual Commitment, Feedback Loops, Distribution of Roles, Autonomy & Agency)?
- Which **functions** from the Matrix are under strain (e.g., Problem Discovery, Coordination, Monitoring & Feedback)?
- At which **level** does instability first appear (Preconditions, Needs, Functions, Practices, Meta-Practices)?

Extended Human Dynamics (Conditions + Needs)

Use Extended Dynamics to reason about:

- **How cooperation is felt and interpreted** by individuals and groups.
- **Which contextual, relational, structural, or developmental patterns** are amplifying or suppressing cooperation.
- **Which deeper needs** (Purpose & Direction, Trust & Safety, Growth & Evolution, Recognition & Belonging, Autonomy & Coherence) are at play.

Typical questions:

- How do **collective conditions** (culture, incentives, decision norms) affect behavior?
- How do **individual conditions** (perceived safety, control, predictability) shape engagement?
- Which **extended needs** are most salient in this conflict or disengagement?
- How are **political vectors** (power, visibility, gatekeeping) and **psychological vectors** (fear, shame, identity, pride) interacting with the structure?

System Modes (Setup, Stabilization, Growth, Conflict, Reset)

Use System Modes to decide:

- **What kind of work** is appropriate now (design, repair, optimization, conflict handling, reset).
- **What to prioritize** and **what to postpone** at this moment.
- **How far** you can safely go at each level of the Pyramid without breaking the Level Rule.

Typical questions:

- Are we **starting something** that needs a deliberate shape? (Setup)
- Are we **trying to repair recurring issues** and restore reliability? (Stabilization)
- Are we **extending autonomy and capability** on a stable base? (Growth)
- Are we **facing tensions and mistrust** that must be addressed directly? (Conflict)
- Has the **context or purpose changed so much** that the old system no longer makes sense? (Reset)

Modes do not tell you *what is true* about the system.
They tell you *what kind of response* is appropriate once you understand what is true.

5.1.3 A Simple Integration Pattern

The Integration Guide describes a repeatable four-step pattern.

You can apply it **within any System Mode**, but it is especially powerful in Stabilization, Conflict, and Reset.

Step 1 – Start From Observation, Not Theory

Begin with what is **actually happening**:

- repeated friction between specific roles or teams
- confusion about priorities or success criteria
- stalled decisions, escalating conflict, or quiet disengagement
- a sense that “we’re working hard but not moving together”

Write down **concrete examples**, not just interpretations:

- “X said Y in meeting Z.”
- “Access ticket waited 14 days before anyone picked it up.”
- “Three people privately said they don’t trust the roadmap.”

This is your **observable surface**.
Diagnostics starts from **evidence**, not from favorite models.

Step 2 – Map to the Core Model

Ask:

> "If I ignore personalities for a moment, what is structurally happening?"

Use the Matrix and Pyramid to identify:

- Which **conditions** might be weak or misaligned?
- Which **core cooperative needs** are clearly unmet?
- Which **functions** (Matrix cells) are failing or missing?
- At what **level** does the instability appear first?

Examples:

- Misaligned expectations → likely failure in **Common Purpose × Shared Understanding** (Alignment on why).
- Chronic rework across teams → likely problems in **Interdependence × Distribution of Roles** (Coordination).
- Fear of raising issues → likely fragility in **Trust × Feedback Loops** (Safety in feedback).

At this step, stay with **structure**, not personal motives.

This is also where the **Level Rule** applies most strongly:
stabilize lower levels before expecting higher-level practices or innovation to stick.

Step 3 – Add the Extended Lens

Once you have a structural hypothesis, ask:

> "Given this structure, **why might people be behaving this way**?"

Use Extended Conditions and Extended Needs to explore:

- **Contextual** patterns: volatility, opaque decisions, shifting priorities.
- **Relational** patterns: alliances, avoidance, blame, emotional load.
- **Structural** patterns: gatekeepers, bottlenecks, invisible veto power.
- **Developmental** patterns: lack of learning rhythm, repeated unclosed loops.

Then consider:

- Which **extended needs** are driving the strongest reactions?
- Purpose & Direction?
- Trust & Safety?
- Growth & Evolution?
- Recognition & Belonging?
- Autonomy & Coherence?

This step explains *why* specific structural points are "hot" for people:

- why some functions meet resistance,
- why some agreements feel unfair,
- why some changes trigger disproportionate reactions.

The goal is **understanding**, not excuse-making:
human dynamics do not override structural issues, they explain how those issues are lived.

Step 4 – Choose the Right Mode and Intervention Channels

With both layers in view, decide:

1. **Which System Mode fits the current situation best?**
2. Are you mostly lacking a proper design? → **Setup Mode**
3. Are you trying to restore basic reliability? → **Stabilization Mode**
4. Are you trying to extend autonomy/capability on a stable base? → **Growth Mode**
5. Are you dealing with strong tension, mistrust, or harm? → **Conflict Mode**
6. Has the cooperation's purpose or context fundamentally shifted? → **Reset Mode**

You can change modes later as you learn more.
The point is to avoid mixing conflicting intentions (e.g., trying to Grow while you should Reset).

1. **Within that mode, choose appropriate intervention channels:**
2. **Structural interventions**
    - Clarify purpose, boundaries, roles, decision paths.
    - Adjust dependencies, interfaces, or information flow.
    - Redesign practices or governance to better support key functions.
3. **Collective agreements**
    - Align expectations about behavior, feedback, and repair.
    - Make implicit norms explicit and renegotiate where needed.
    - Establish shared rules for escalation, conflict handling, or change.
4. **Individual support**
    - Coaching, mentoring, or guided reflection.
    - Support for people carrying disproportionate load or risk.
    - Space to process change, conflict, or identity shifts.

The key is **alignment**:

- Don't use individual coaching to compensate for **structural injustice**.
- Don't redesign process to fix what is fundamentally a **trust or safety breakdown**.
- Don't treat political power plays as if they were only misunderstandings about process.
- Don't choose a mode whose intent contradicts the system's actual needs.

5.1.4 Using Practices and Tools Safely

Extended Human Dynamics naturally invites leadership tools:

- personality / style lenses (e.g., DiSC),
- social neuroscience models (e.g., SCARF),
- motivation and needs tools (e.g., Moving Motivators),
- feedback and mediation techniques.

Use such tools to:

- generate insight and shared vocabulary,
- surface perceptions, fears, and expectations,
- create space for reflection and dialogue,
- support personal and collective awareness.

Do **not** use them to:

- override or ignore structural problems identified in the Core Model,
- assign fixed identity labels or stereotypes,
- “fix” political issues solely through individual personality work,
- “fix” structural issues solely through emotional or motivational interventions.

The system dictates which tools are appropriate.
Tools do not dictate what the system is.

5.1.5 How This Guide Fits Within HCS

The Integration Guide is **not** an additional model.
It is a practical bridge that ensures:

- the **Core Model** stays clean and universal,
- Extended Dynamics do not dilute structural clarity,
- System Modes are chosen consciously, not implicitly,
- practitioners avoid misdiagnosis (blaming people for system design, or vice versa),
- leadership and coaching tools are used intentionally, not by reflex,
- interventions land at the **correct systemic level** and through the right channels.

Within **System Modes**:

- **Setup Mode** uses this guide to ensure initial design considers both structure and human reality.
- **Stabilization Mode** uses it to separate structural faults from emotional or political reactions.
- **Growth Mode** uses it to scale autonomy and learning without eroding safety or coherence.
- **Conflict Mode** uses it to name what hurts without losing sight of structural leverage points.
- **Reset Mode** uses it to design a new system (or end the old one) while honoring what people have lived through.

Subsequent files in the Diagnostics section build on this guide:

- **Diagnostic Workflow** – a step-by-step walk through the Matrix and Pyramid to locate structural instability.
- **Diagnostic Dynamics** – ways to layer Extended Conditions and Extended Needs onto that workflow.

Used consistently, the Integration Guide aligns **human experience** with **structural design** and **mode choice**,
helping cooperation become both **stable and humane**, both **clear and adaptive**.

5.2 Diagnostic Workflow – From Observation to Intervention

The **HCS Diagnostic Workflow** is a small loop that turns the Core Model into a usable lens.

It provides a structured way to trace **observable issues** in cooperation back to their **systemic root causes** in the Matrix and Pyramid — before you decide how to intervene or which System Mode to operate in.

Most improvement efforts fail because they start with *practice changes* (Level 4) rather than asking:

> "At which level of cooperation is this actually unstable?"

This workflow keeps analysis grounded and ensures each corrective action strengthens the **right layer** of the system.

5.2.1 Why This Workflow Exists

The workflow exists to:

- move from vague problem labels ("communication issue", "planning is bad") to **specific matrix cells and levels**;
- respect the **Level Rule** (stabilize lower levels before adjusting higher ones);
- avoid jumping straight into tools and methods without understanding what function they are supposed to serve.

It focuses on the **Core Model** first:

- Conditions × Needs → Functions (Matrix)
- Levels 1–5 (Pyramid)

Extended Human Dynamics and System Modes are layered on **after** this basic structural pass.

5.2.2 When to Use the Diagnostic Workflow

Use this workflow when:

- you observe **recurring friction**, not one-off incidents;
- multiple people describe the same issue in different words;
- you are tempted to "roll out a new practice" but cannot clearly say which function it should support;
- you are in **Stabilization Mode** and need to understand where to repair first;
- you are in **Growth, Conflict, or Reset Mode** and want a clearer view of the underlying structure.

It works at multiple scales:

- within a single team;
- across client–vendor relationships;
- across functions (e.g., business, product, engineering, operations).

5.2.3 Workflow Overview

The diagnostic loop has five steps:

1. **Observation** – Capture neutral evidence.
2. **Matrix Mapping** – Locate the function (Condition × Need).
3. **Level Check** – Identify the lowest unstable level.
4. **Function → Practice** – Decide what to strengthen and how.
5. **Trial & Learn** – Run a small experiment and review.

You can run this loop in minutes for a single issue, or more formally for bigger patterns.

5.2.4 Step 1. Observation – Capture Evidence

Start with what you **see, hear, or measure** — without interpretation or blame.

Focus on **observable behavior**, not motives or personality.

Examples:

- "Critical dependencies were discovered three days before release; downstream team blocked twice this sprint."
- "Access requests for test environments waited 10–14 days before approval in the last three cycles."
- "In three consecutive steering meetings, we changed the priority of the same initiative."

Good observations answer:

- what happened,
- when and where,
- who was affected,
- how often it occurred.

This ensures diagnosis starts from **data**, not opinion.

5.2.5 Step 2. Matrix Mapping – Locate the Function

Next, ask:

> "If I ignore individual motives for a moment, **where does this live in the Matrix**?"

Map the observation to the **HCS Matrix**:

- Pick the most relevant **Condition** (vertical axis):
  Common Purpose, Interdependence, Communication, Trust, Change/Uncertainty.
- Pick the most relevant **Core Need** (horizontal axis):
  Shared Understanding, Mutual Commitment, Feedback Loops, Distribution of Roles, Autonomy & Agency.

This gives you a specific **function** (cell) to work with.

Example:

> Issue: Dependencies discovered late; downstream team repeatedly blocked.
> Likely cell: **Interdependence × Feedback Loops → Outcome Reflection / Coordination Feedback**

Another example:

> Issue: Different leaders give conflicting statements about what "success" means.
> Likely cell: **Common Purpose × Shared Understanding → Alignment on Why**

This step prevents vague labels like "communication problem" and anchors the issue to **one or a small set of functions**.

If several cells seem plausible, note the top 2–3 and keep them as hypotheses.

5.2.6 Step 3. Level Check – Find How Deep the Root Is

Now, decide at **which level of the Pyramid** the instability appears first.

Use this as a quick guide:

If the problem is mainly about…	It likely belongs to…
No shared reason to cooperate, no real trust, no basic communication ground	**Level 1 – Preconditions**
Misalignment, unclear expectations, weak feedback, low perceived agency	**Level 2 – Core Needs**
Broken coordination, planning, or learning cycles	**Level 3 – Functions**
Ineffective or over-complicated methods	**Level 4 – Practices & Frameworks**
Lack of reflection on how we work, inability to adapt the system itself	**Level 5 – Meta-Practices & Innovation**

Examples:

- Repeated late dependency discovery, even after checklists exist →
  likely **Level 3 – Monitoring & Feedback function**, with a possible **Level 2 – Feedback Loops** issue beneath.
- Constant disagreement about what success means, despite having KPIs →
  likely **Level 1 – Common Purpose** and **Level 2 – Shared Understanding** problem; KPIs are Level 4 artifacts.

The **Level Rule** applies here:

> Stabilize **lower levels** before expecting higher levels (functions, practices, innovation) to work.

If you find signs of instability at multiple levels, treat the **lowest one** as the primary diagnostic target.

5.2.7 Step 4. Function → Practice – Decide What to Strengthen

Once you know:

- **which function** is under strain (Matrix cell), and
- **which level** is unstable (Pyramid),

you can choose or design a **practice** to support that function at the correct level.

Use the 25 Matrix functions as a reference for what “healthy cooperation” looks like. The practice can be:

- an existing method you already use;
- a method borrowed from a known framework;
- a lightweight, custom agreement that fits your context.

Examples:

1. Matrix cell: **Communication × Feedback Loops → Signal & Response**
2. Observed issue: risks and blockers appear “out of nowhere”.
3. Practice ideas:
    - Add a daily or twice-weekly “risk surfacing” moment.
    - Introduce a simple, visible “stop-the-line” signal for serious blockers.
    - Define who must respond and within what time.
4. Matrix cell: **Interdependence × Distribution of Roles → Coordination**
5. Observed issue: no one is clearly responsible for cross-team dependencies.
6. Practice ideas:
    - Introduce a rotating “dependency coordinator” role.
    - Add a simple rule: “No new work started while unresolved cross-team blocker > X days.”

The question is always:

> "What **practice or agreement** would directly strengthen this function,
> at this level, in this context?"

If the function is missing entirely, start with something **small and explicit**, not a heavy framework.

5.2.8 Step 5. Trial & Learn – Validate and Iterate

Treat each diagnostic outcome as a **hypothesis**, not a verdict.

For each chosen practice:

1. Define a **simple signal or metric** to watch.
2. Time-to-unblock, frequency of rework, number of escalations, etc.
3. Set a **review date** (e.g., after two sprints, one month, one release).
4. Run the practice and then review:
5. Did the specific observable issue improve?
6. If not, did something else become visible (e.g., resistance, new constraints)?

If the practice does not help:

- Re-check whether you targeted the **correct level**.
- Ask whether a **lower level** needs attention first (e.g., trust or shared understanding).
- Use the **Diagnostic Dynamics** file to explore contextual, relational, or political factors that may be blocking change.

The point is not to find the perfect practice on the first try,
but to keep the loop **short, explicit, and learnable**.

5.2.9 Output Template

You can use this lightweight template for documentation or reflection:

```
Observation:
Matrix cell(s):
Lowest affected level:
Function to strengthen:
Practice(s) selected:
Signal/metric to watch:
Review date:
Outcome / next step:
```

It fits naturally into retrospectives, 1:1s, coaching sessions, or system reviews.

5.2.10 Practical Example (Full Loop)

Observation
"Design team's updates rarely align with development progress; misinterpretations surface during QA. This has happened in the last three sprints."

Matrix Mapping
Likely cell:
Communication × Shared Understanding → Language / Terms

Level Check
- Level 2: Shared Understanding is weak (different mental models of the same work).
- Level 3: Coordination function between design and dev is under strain.

Function to Strengthen
Create and maintain shared language and explicit handoff criteria.

Practice(s)
- Introduce a short weekly alignment session between design and dev leads focused only on “what changed and what that means”.
- Create a shared “definition of ready” checklist for design handoffs.

Signal / Metric
- Reduction in QA rework caused by misunderstanding.
- Fewer clarifications needed after design handoff.

Review Date
- After three sprints.

Outcome / Next Step
- If signal improves: consider light optimization or extension.
- If not: revisit Level 1–2 assumptions (are purpose, priorities, or roles actually aligned?).

5.2.11 Summary Table

Step	Question	Output
1. Observation	What is happening, concretely?	Neutral description of events/pattern
2. Matrix Mapping	Where does this live structurally?	Condition × Need → Function (cell)
3. Level Check	How deep is the instability?	Lowest affected level in the Pyramid
4. Function → Practice	What should we strengthen? How?	Target function + candidate practice(s)
5. Trial & Learn	Did it help? What did we learn?	Measured outcome + next adjustment

5.2.12 Essence

Every cooperation issue can be traced through the same Core Model lens:

> **Observation → Matrix → Level → Function → Practice → Learning**

This workflow does not replace judgment, System Modes, or Extended Human Dynamics.
It provides a **stable structural backbone** for all of them — a way to keep diagnosis grounded while you decide how to act.

5.3 Diagnostic Dynamics – Working With Political and Psychological Fields

The **Diagnostic Workflow** focuses first on the **Core Model**:

- Conditions × Needs → Functions (Matrix)
- Levels 1–5 (Pyramid)
- Level Rule (stabilize lower levels first)

Often, that is enough to locate where cooperation is structurally unstable.

However, there are many situations where:

- structure looks correct **on paper**,
- practices are in place,
- but cooperation still feels tense, fragile, or distorted.

This is where **Diagnostic Dynamics** comes in.

It adds the lens of **Extended Human Dynamics** to diagnosis, by making visible:

- how **political fields** (power, legitimacy, visibility, gatekeeping) and
- how **psychological fields** (fear, identity, fairness, safety)

are shaping **conditions and needs**, and therefore **behavior**, inside an otherwise sound structure.

Diagnostic Dynamics does **not** change the Core Model or the Level Rule.
It helps you understand **why the system behaves the way it does**, so that interventions land at the right level.

5.3.1 Political and Psychological Fields

The Core Model defines **what cooperation requires**.
Extended Human Dynamics describes **what cooperation feels like**.

Political and psychological fields explain **why cooperation becomes difficult**, even when the basic design is in place.

Political Field

Political influences emerge from:

- Power and authority distribution
- Control of resources and access
- Legitimacy battles (“who is the real owner?”)
- Visibility and reputation
- Representation and agenda-setting
- Hidden coalitions, veto points, and gatekeepers

They shape:

- whose interests define reality when there is ambiguity,
- whose work is prioritized or ignored,
- who can say “no” without consequences.

In short: political fields shape *permission*.

Psychological Field

Psychological influences emerge from:

- Fear, anxiety, loss of safety
- Identity and status threat
- Emotional triggers and personal history
- Perceived fairness or unfairness
- Trust posture (default trust or default suspicion)
- Vulnerability and self-protection patterns

They shape:

- how people interpret behavior and decisions,
- whether they speak up or stay silent,
- whether they invest energy or withdraw.

In short: psychological fields shape *participation*.

Together, these fields can **distort** or **amplify** any condition or need:

- Cooperation can be structurally sound but **politically impossible**.
- Cooperation can be well-designed but **psychologically unsafe**.

Diagnostic Dynamics makes these forces discussable without turning HCS into a personality or politics model.

5.3.2 How Fields Act on Extended Conditions and Needs

Extended Human Dynamics describes:

- **Extended Conditions** – contextual, relational, structural, developmental patterns that shape the "climate" of cooperation.
- **Extended Needs** – deeper human needs that show up around purpose, safety, growth, belonging, and autonomy.

Political and psychological fields can act on *any* of these.

Examples:

- **Contextual + Political**
- Strategy and priorities are defined by a small group; others feel decisions are opaque.
- Formally, purpose is clear; in practice, **whose purpose counts** is contested.
- **Relational + Psychological**
- Past incidents (broken promises, public shaming) make people cautious.
- Structurally, feedback channels exist; psychologically, feedback feels dangerous.
- **Structural + Political**
- A team owns a critical platform but has no formal say in roadmap decisions.
- On paper, roles and dependencies are defined; politically, they are ignored.
- **Developmental + Psychological**
- Repeated "failed initiatives" create learned helplessness.
- People no longer believe change efforts will last, even if design looks better.

These forces do not nullify the Core Model.
They change **how conditions and needs are lived**, which in turn affects whether functions can operate as designed.

5.3.3 Four Diagnostic Quadrants

To keep diagnosis precise, it helps to distinguish **where** the issue is most visible.

Think in terms of two axes:

1. **Condition vs Need** – Is this mostly about the **situation** we are in (conditions) or about **what people require** to stay engaged (needs)?
2. **Collective vs Individual** – Is the pattern mostly **system-wide** or is it concentrated in **specific people or roles**?

This gives four quadrants:

1. **Collective Condition**
2. Examples: overall culture of risk avoidance; cross-team blame habits; chronic overload.
3. Often expressed as “this is how things are *around here*”.
4. **Individual Condition**
5. Examples: one person has low perceived safety; another is isolated from key information; one role is structurally exposed.
6. Often expressed as “this role / person is always in a difficult spot”.
7. **Collective Need**
8. Examples: a department needs recognition; a partner needs legitimacy; a group needs real involvement in decisions.
9. Often expressed as “they never listen to us” or “we’re treated as second-class”.
10. **Individual Need**
11. Examples: a leader needs more clarity to lead; an engineer needs growth opportunities; someone needs assurance they will not be punished for honesty.
12. Often expressed in 1:1s as “for me personally, this is what hurts”.

Why this matters:
If you misidentify the quadrant, you will likely intervene at the wrong level:

- giving coaching to an individual when a whole group is structurally excluded,
- changing process when a specific person needs support and feedback,
- running a team workshop for what is fundamentally a cross-organization political conflict.

Quadrants help you be **precise about where the issue lives**.

5.3.4 Three-Layer Diagnostic Sequence

When Extended Dynamics are clearly in play, use this sequence after the basic Diagnostic Workflow:

1. **Confirm the structural baseline (Core Model)**
2. **Locate the quadrant (Extended Conditions/Needs)**
3. **Select the intervention layer**

Step 1. Confirm the Structural Baseline

First, use the **Diagnostic Workflow**:

- map the observation to the Matrix (Condition × Need → Function),
- identify the lowest unstable level in the Pyramid,
- form a hypothesis about the **structural** issue.

Ask:

- Is there a real design problem here (unclear purpose, missing roles, broken feedback)?
- Or is the structure *technically* present but not used as intended?

Only once this is clear should you lean into Extended Dynamics.

Step 2. Locate the Quadrant

Ask:

- Where is this most visible right now?

Examples:

- "People across multiple teams are afraid to speak up in large forums."
  → likely **Collective Condition** (fearful climate) + **Collective Need** (Trust & Safety).
- "One team feels consistently sidelined from key decisions."
  → likely **Collective Need** for Recognition & Belonging, with a **Structural** condition pattern.
- "A single senior engineer resists all changes and dominates discussions."
  → could be **Individual Condition** (identity threat) and **Individual Need** (Autonomy & Coherence), but check whether they are carrying **systemic risk** that nobody else is owning.

Naming the quadrant prevents you from treating **systemic** issues as **personal** or vice versa.

Step 3. Select the Intervention Layer

Once the quadrant is clear, choose the appropriate **intervention layer**:

- **Structural**
- Adjust boundaries, roles, decision paths.
- Add or repair feedback loops, transparency, access.
- Example: change who approves work, who is at which meeting, or how information flows.
- **Collective**
- Align purpose and narratives.
- Establish or renegotiate shared agreements.
- Run inclusion or representation checks ("who is affected vs who is in the room?").
- Example: joint sessions between client and vendor leadership to reset expectations.
- **Relational**
- Facilitate repair between specific roles or groups.
- Make reciprocity and respect explicit topics.
- Example: structured dialogue between two teams with history of mutual blame.
- **Individual**
- Coaching, mentoring, tailored support.
- Help individuals process change, conflict, or identity shifts.
- Example: supporting a key person who is over-exposed to risk or conflict.

Often you will need a **combination** (e.g., structural + collective + relational).
The point is to be intentional about **where you start**.

5.3.5 Mode-Specific Signals

Diagnostic Dynamics plays differently depending on **System Mode**.

In Setup Mode

Signals:

- Stakeholders hesitate to commit, even when design seems clear.
- Some groups are “quiet” in early conversations; their risks or needs are not voiced.

Focus:

- Extended Conditions: who holds power, who feels vulnerable, who is missing.
- Extended Needs: Purpose & Direction, Trust & Safety, Recognition & Belonging.

Intention:

- Use Extended Dynamics to design a **realistic starting shape**,
  not an idealized cooperation contract that ignores politics and psychology.

In Stabilization Mode

Signals:

- Same structural fixes are proposed repeatedly but never land.
- People comply in meetings but revert to old patterns in practice.
- Escalations are emotional, not only technical.

Focus:

- Extended Conditions: relational history, hidden gatekeepers, over-exposed roles.
- Extended Needs: Trust & Safety, Autonomy & Coherence.

Intention:

- Distinguish between “we don’t know what to do” and “we don’t believe it’s safe or fair to do it”.
- Adjust interventions accordingly: structural where needed, but also relational and individual.

In Growth Mode

Signals:

- Some teams welcome increased autonomy; others show quiet resistance.
- Perceived winners and losers emerge as scope expands.
- Improvement initiatives are seen as “their agenda”, not “our evolution”.

Focus:

- Extended Needs: Growth & Evolution, Recognition & Belonging, Autonomy & Coherence.
- Extended Conditions: structural and contextual fairness of growth.

Intention:

- Ensure growth does not erode safety or legitimacy.
- Adjust growth pace and scope to what the system can integrate.

In Conflict Mode

Signals:

- Strong narratives about “them” vs “us”.
- Emotional reactions disproportionate to the immediate trigger.
- People reinterpret events through a lens of past harm.

Focus:

- Both fields at full strength: political (who benefits, who suffers) and psychological (hurt, fear, shame).
- Extended Needs: Trust & Safety, Recognition & Belonging, Purpose & Direction.

Intention:

- Make conflict **sayable** without punishment.
- Separate structural, relational, and individual elements.
- Decide honestly whether repair, containment, or Reset is appropriate.

In Reset Mode

Signals:

- Cynicism and resignation, even when new proposals are sound.
- People privately say “this should have ended a while ago”.
- Old harms are carried into new conversations.

Focus:

- Extended Conditions: developmental history, accumulated fatigue.
- Extended Needs: Purpose & Direction, Recognition & Belonging, Autonomy & Coherence.

Intention:

- Close or transform the system with **integrity**.
- Ensure any new Setup does not inherit unspoken trauma or political debt.

5.3.6 Practical Heuristics

A few quick rules of thumb:

- **“If structure is clear but behavior is erratic, check psychological fields.”**
  Fear, identity threat, or learned helplessness often override clarity.
- **“If individuals keep failing in the same way, check collective conditions.”**
  People adapt to the system they are in; repeated patterns are rarely purely personal.
- **“If the story makes sense to leadership but not to the team, check political fields.”**
  Narratives often serve those who shape them.
- **“If conflict repeats, examine unmet needs, not personalities.”**
  Unmet needs create stable patterns of resistance or withdrawal.
- **“If people stop speaking up, treat this as a system alarm, not a personal flaw.”**
  Silence is both a political and a psychological signal.

These heuristics keep you from defaulting to the most comfortable explanation.

5.3.7 How to Use Diagnostic Dynamics with the Workflow

In practice, combine the pieces like this:

1. **Run the Diagnostic Workflow**
2. Observation → Matrix → Level → Function → Practice.
3. **If friction persists or feels "hot"**, layer in Diagnostic Dynamics:
4. Identify quadrant (collective / individual × condition / need).
5. Notice political and psychological fields at play.
6. **Pick a System Mode** that matches the overall situation:
7. Setup, Stabilization, Growth, Conflict, or Reset.
8. **Choose intervention layers** (structural, collective, relational, individual) that match:
9. the quadrant,
10. the fields,
11. and the mode.
12. **Run a small, explicit experiment**, then review:
13. Did structural changes help?
14. Did relational work shift anything?
15. Are different needs now visible?

Diagnostic Dynamics does not turn you into a therapist or a political strategist.
It gives you enough language to:

- **see** how political and psychological fields shape cooperation,
- **name** their effects without blaming individuals,
- and **act** at the right level, with the right expectations.

That is often enough to turn "this is just how it is here" into something once again **changeable**.

6. REFERENCE

6.1 Glossary

The **Human Cooperation System (HCS)** defines a shared language for describing **cooperative work systems** – how people and organizations align, commit, and adapt under changing conditions.

This glossary supports consistency across:

- the **Core Model** (Matrix + Pyramid),
- **Extended Human Dynamics**,
- **System Modes**, and
- the **Diagnostics** section.

6.1.1 Core System Terms

Term	Meaning
Human Cooperation System (HCS)	A system model that explains how cooperative work remains stable and adaptive across changing conditions. It describes the "physics" of cooperation using the Matrix, Pyramid, Extended Human Dynamics, System Modes, and Diagnostics.
Core Model	The structural heart of HCS, composed of the **Matrix** (Conditions × Needs × Functions) and the **Pyramid** (five levels of cooperative stability). It defines what cooperation *requires* at a minimum.
Matrix	A 5×5 map that crosses **Preconditions for Cooperation** (vertical axis) with **Core Human Needs for Cooperative Work** (horizontal axis). Each cell represents a **Cooperative Function** that must be supported for cooperation to remain healthy.
Pyramid	A layered model with five levels: (1) Preconditions for Cooperation, (2) Core Human Needs, (3) Cooperative System Functions, (4) Practices & Frameworks, and (5) Meta-Practices & Innovation. It defines **dependency order** between layers.
Level Rule	The principle that **dysfunction at a lower level cannot be fixed by interventions at higher levels**. Stability must be restored from the bottom up (Level 1 → Level 2 → Level 3 → Level 4 → Level 5).
Extended Human Dynamics	A lens that describes how **contextual, relational, structural, and developmental** patterns (Extended Conditions) and deeper **human needs** (Extended Needs) shape how cooperation is experienced. It explains variation without changing the Core Model.
System Modes	Five systemic "stances" that describe **what kind of work** is appropriate right now: Setup, Stabilization, Growth, Conflict, Reset. Modes guide priorities and guardrails for intervention.
Diagnostics	A set of workflows and patterns that combine the Core Model, Extended Human Dynamics, and System Modes into a **repeatable way** of observing issues, locating root causes, and choosing interventions.
Diagnostic Workflow	A small loop that moves from Observation → Matrix Mapping → Level Check → Function → Practice → Learning. It anchors diagnosis in the Core Model before adding human dynamics or mode choices.
Diagnostic Dynamics	The extension of the Diagnostic Workflow that incorporates Extended Conditions, Extended Needs, and political/psychological fields, helping distinguish structural issues from human and political distortions.

6.1.2 Two Paths of Problem Solving

Term	Meaning
Path of Encapsulation	The tendency to reduce friction by creating **clear interfaces, contracts, and modular components** so that parties need to understand each other as little as possible. Effective for well-bounded problems; harmful when used for deep cooperation needs.
Path of Integration	The tendency to reduce friction by investing in **shared understanding, joint sense-making, and coordinated governance** so that parties can adapt together. Essential when work is interdependent, novel, or politically sensitive.
Encapsulation vs Integration Balance	The deliberate choice of **where to encapsulate** work with stable contracts and **where to integrate** through joint sense-making and shared decision-making. HCS helps make this choice explicit instead of accidental.

6.1.3 Pyramid – Levels and Core Concepts

Level 1 – Preconditions for Cooperation

These are the **existential conditions** required for cooperation to exist at all.

Term	Meaning
Common Purpose	The shared intent and direction that makes cooperation worthwhile. Answers “Why are we doing this together rather than separately?”
Interdependence	Recognition that outcomes depend on others, making collaboration both necessary and consequential.
Communication (Ground)	A minimal shared ground for exchanging information – channels, language, and norms that make basic mutual orientation possible.
Trust	Confidence that others will act with sufficient reliability, integrity, and care within shared norms. Without some trust, cooperation collapses into control and avoidance.
Change / Uncertainty Tolerance	The ability of the system to live with variability (in context, priorities, constraints) without losing coherence or legitimacy.

Level 2 – Core Human Needs for Cooperative Work

These are the **human-level enablers** that make Level 1 operational in day-to-day work.

Term	Meaning
Shared Understanding	A sufficiently aligned mental model of goals, constraints, and current reality. People may disagree, but they know what they disagree about.
Mutual Commitment	Reciprocal willingness to invest effort and uphold agreements. Both sides recognize and value each other’s contribution and risk.
Feedback Loops	Safe and reliable ways to surface information (progress, risks, errors, tensions) and see that it leads to appropriate response or adjustment.
Distribution of Roles	Clarity on who does what, who decides what, and who is accountable for what. Includes explicit edges and overlaps.
Autonomy & Agency	The ability of people and teams to act intentionally within agreed boundaries, shaping outcomes rather than merely complying.

Level 3 – Cooperative System Functions

Level 3 contains the **stable functions** represented by the 25 Matrix cells (Condition × Need). Each cell describes **what must happen** for cooperation to work (e.g., Alignment on Why, Coordination, Signal & Response, Enablement & Empowerment, Adaptation & Learning).

Term	Meaning
Cooperative Function (Matrix Cell)	A specific, stable capability of cooperation (e.g., discovering problems, coordinating interdependent work, making decisions, learning from outcomes). Functions describe *what* must happen, independent of specific methods.
Function Group	A loose cluster of related functions (e.g., Problem Discovery, Planning & Prioritization, Monitoring & Feedback, Enablement & Empowerment, Adaptation & Learning) used as a mental shortcut when working with the Matrix at a higher level.

Level 4 – Practices & Frameworks

Term	Meaning
Practice	A concrete, repeatable way of working (e.g., retrospectives, sprint planning, RACI, incident reviews) that serves one or more Cooperative Functions. Practices are **replaceable**; functions are not.
Framework	A named collection of practices and roles (e.g., Scrum, Kanban, SAFe, OKRs) that assumes certain Core Model conditions are already in place. Frameworks live at Level 4 and depend on Levels 1–3.

Level 5 – Meta-Practices & Innovation

Term	Meaning
Meta-Practice	Any activity where people **reflect on and redesign their way of working** (e.g., designing custom playbooks, mapping cooperation systems, coaching others in system thinking).
System Innovation (Cooperation)	The deliberate creation or evolution of new practices and structures to better support cooperative functions. It is innovation focused on *how* we cooperate, not just *what* we deliver.

6.1.4 System Modes

Term	Meaning
Setup Mode	Mode focused on **designing preconditions and expectations** before work begins or when cooperation is fundamentally re-chartered. Establishes shared purpose, interdependence, roles, and basic governance.
Stabilization Mode	Mode focused on **repair and calibration** when cooperation exists but suffers recurring friction. Aims to restore basic reliability by fixing issues at the **lowest unstable level**.
Growth Mode	Mode focused on **optimization and extension** once cooperation is stable enough. Increases autonomy, speed, and learning while preserving safety and coherence.
Conflict Mode	Mode focused on **safety and realignment** when tension, mistrust, or harm are prominent. Makes conflict discussable, separates structural and relational issues, and clarifies whether to repair, contain, or reset.
Reset Mode	Mode focused on **existential re-evaluation** when context or purpose has shifted so much that the current cooperation system no longer makes sense. Decides what ends, what transforms, and what restarts via Setup.

6.1.5 Diagnostic Terms

Term	Meaning
Observation (Diagnostics)	A neutral description of what is happening (events, patterns, frequencies) without interpretation or blame. Starting point of the Diagnostic Workflow.
Matrix Mapping	The act of locating an observation in the Matrix by identifying the most relevant Condition, Need, and Cooperative Function.
Level Check	Determining the **lowest Pyramid level** at which instability appears (Preconditions, Needs, Functions, Practices, Meta-Practices).
Function → Practice Mapping	Choosing or designing a practice that directly supports a specific Cooperative Function at the correct level, instead of adopting practices by imitation.
Trial & Learn Loop	The small cycle of trying a targeted change, observing its effects, and adjusting based on evidence, rather than treating any diagnostic conclusion as final.

6.1.6 Extended Conditions and Needs

Extended Conditions

Term	Meaning
Extended Conditions	Contextual, relational, structural, and developmental patterns that shape how cooperation is experienced (e.g., incentives, history, decision norms). They do not change the Core Model but strongly influence its quality.
Contextual Conditions	External and environmental factors (market, regulation, strategy, crises, organizational priorities) that affect pressure, risk, and attention.
Relational Conditions	Patterns in how people and groups relate over time (alliances, avoidance, trust history, blame habits).
Structural Conditions	How power, access, decision rights, and resources are arranged (gatekeepers, bottlenecks, representation).
Developmental Conditions	The maturity and history of cooperation (past initiatives, learning rhythms, accumulated fatigue or trauma).

Extended Needs

Term	Meaning
Extended Needs	Human motivational and relational needs that shape engagement and emotional experience in cooperation. They go beyond the minimal Core Needs and are vital for **sustained, humane collaboration**.
Purpose & Direction	The need to understand and endorse the “why” of cooperation and see how one’s work contributes to something meaningful.
Trust & Safety	The need to feel physically, emotionally, and politically safe enough to participate honestly and take reasonable risks.
Growth & Evolution	The need for development, mastery, challenge, and the sense that the system is improving rather than stagnating.
Recognition & Belonging	The need to feel seen, valued, fairly treated, and included in the community of cooperation.
Autonomy & Coherence (Extended)	The need to experience freedom of action **and** coherence with one’s values, identity, and the broader direction of the system.

Collective vs Individual

Term	Meaning
Collective Conditions / Needs	System-level patterns, narratives, and needs that emerge from groups (teams, departments, organizations). Examples: a team's need for recognition, an organization's need for strategic clarity.
Individual Conditions / Needs	A person's lived experience within cooperation (their safety, clarity, growth, recognition, autonomy). These can differ greatly from the group's averages.
Quadrant (Diagnostics)	The four-way classification used in Extended Dynamics: Collective Condition, Individual Condition, Collective Need, Individual Need. It guides **where** an issue lives and helps pick the right intervention layer (structural, collective, relational, individual).

6.1.7 Influence Fields

Term	Meaning
Political Field	The pattern of power, legitimacy, resource control, representation, and informal hierarchies that shapes who gets heard, who can say no, and whose interests define "reality". Political dynamics primarily shape **permission**.
Psychological Field	The pattern of emotions, identity, fear, trust posture, and personal narratives that shapes how people interpret events, whether they speak up, and how much energy they invest. Psychological dynamics primarily shape **participation**.
Distortion Vector	Any political or psychological influence that **amplifies or suppresses** conditions and needs, making cooperation difficult even when structure looks correct on paper.
Intervention Layer	The level at which an issue must be addressed: **structural** (system design), **collective** (group narratives and agreements), **relational** (between specific roles or groups), or **individual** (support and coaching).

6.1.8 Principles and Rules

Term	Meaning
Function-First Principle	Practices and tools should be chosen or designed to **serve a specific Cooperative Function**, not adopted by imitation or trend.
Lowest-Level Fix Rule	Interventions should always target the **lowest level** where instability appears. Fix Level 1–2 issues before expecting Level 3–5 improvements to hold.
Whole-System View	The stance that all conditions, needs, and functions interact; changing one element will affect others. Diagnostics should always consider **systemic side-effects**.
Mode Alignment	The principle that **what you are trying to do** (setup, stabilize, grow, resolve conflict, reset) must match the system's actual state. Misaligned modes lead to wasted effort and additional tension.
Reflective Practice	Regular examination of how cooperation itself functions (not only what is delivered), leading to deliberate adjustments and, at Level 5, to system innovation.

6.1.9 Relationships to Other Frameworks

Term	Meaning
3-in-3 SDLC Framework (3SF)	A separate framework that applies HCS-inspired principles to client–vendor ecosystems and software delivery. HCS provides the underlying cooperation model; 3SF focuses on concrete contracts and practices.
Agile / Lean / Scrum / Kanban	Families of Level 4 frameworks that define practices, roles, and cadences. Their success depends on the stability of Levels 1–3 (Preconditions, Needs, Functions) as described by HCS.
Systems Thinking	The broader discipline of viewing organizations as interconnected, adaptive systems. HCS is a **systems-thinking application** focused specifically on human cooperation.

6.1.10 Reference Purpose

This glossary exists to:

- provide a **shared vocabulary** for reading, teaching, and applying HCS;
- keep diagnostic conversations **grounded and precise**, not vague or personal;
- bridge **theory and practice** across HCS, 3SF, and other delivery or leadership frameworks.

Use it as an anchor whenever terms start drifting or being reused with local meanings.

6.2 Sources and Frameworks

The **Human Cooperation System (HCS)** builds upon a wide range of theories, models, and disciplines that have shaped how humans understand **cooperation, trust, group dynamics, and adaptive systems**.
This section lists the **academic and conceptual influences** that inform HCS theory and its derivatives (including 3SF).

Each source is categorized by its **theoretical contribution area** — systemic, psychological, communicative, or organizational — highlighting how HCS integrates existing knowledge into a unified diagnostic model of human cooperation.

6.2.1 Systems Thinking and Cybernetics

Source / Framework	Core Idea	Relevance to HCS
General Systems Theory (Bertalanffy)	All systems — biological, social, or technical — share structural and functional patterns.	Provides the foundation for viewing cooperation as a *system of interacting elements* rather than isolated behaviors.
The Fifth Discipline (Peter Senge)	Organizations learn and adapt through feedback and systemic awareness.	Inspires HCS's focus on *feedback loops* and *collective learning* as stabilizing forces.
The Viable System Model (Stafford Beer)	Describes how systems maintain internal stability and external adaptability through recursive control loops.	Informs HCS's concept of *systemic governance* and *Autonomy–Control balance*.
Cybernetics (Wiener / Ashby)	Studies self-regulating systems and the role of feedback and control.	Underpins HCS's diagnostic logic and *Level Rule* — stability emerges from functional feedback, not hierarchy.
Complex Adaptive Systems (Holland / Gell-Mann)	Systems evolve through adaptation and local interaction under uncertainty.	Reinforces HCS's treatment of *Change and Uncertainty* as natural, not exceptional, conditions.

6.2.2 Human Motivation and Organizational Psychology

Source / Framework	Core Idea	Relevance to HCS
Hierarchy of Needs (Maslow)	Human motivation progresses from basic to self-actualizing needs.	Provides conceptual ancestry for the *HCS Pyramid* and the *Level Rule* — higher forms of cooperation rely on lower-level stability.
Self-Determination Theory (Deci & Ryan)	Autonomy, competence, and relatedness drive intrinsic motivation.	Directly informs HCS's dimensions of *Autonomy* and *Mutual Commitment*.
Theory X and Theory Y (McGregor)	Management assumptions influence employee motivation and trust.	Supports HCS's framing of *Trust* and *Agency* as emergent, not enforceable, conditions.
Psychological Safety (Edmondson)	Teams learn and innovate when members feel safe to take interpersonal risks.	Embedded in HCS's *Learning and Adaptation* functions and Extended Needs around *Trust & Safety*.
Social Exchange Theory (Blau)	Relationships are sustained through reciprocal benefit and fairness.	Grounds HCS's view of *Mutual Commitment* as both emotional and contractual.
Self-Determination Theory / Modern Motivation Models	Motivation emerges from autonomy, mastery, and belonging rather than external control.	Supports HCS Extended Needs around autonomy, growth, and recognition.
SCARF Model (David Rock)	Social threat and reward shape behavior through status, certainty, autonomy, relatedness, and fairness.	Provides psychological foundations for Extended Needs and psychological vectors.
Intrinsic Motivation Models (Pink / Oldham / Hackman)	Engagement increases when work is meaningful, self-directed, and feedback-rich.	Aligns with Extended Needs such as purpose, autonomy, and growth.

6.2.3 Communication and Shared Meaning

Source / Framework	Core Idea	Relevance to HCS
Shannon–Weaver Model of Communication	Information transmission depends on reducing noise and distortion.	Forms basis for HCS's *Communication Fidelity* and *Signal & Response* functions.
Double-Loop Learning (Argyris & Schön)	True learning requires questioning underlying assumptions, not just correcting errors.	Informs *Reflective Practice* and Level 5 *Meta-Practices & Innovation*.
Sensemaking (Weick)	People construct meaning retrospectively to understand ambiguous situations.	Explains why *Shared Understanding* is a dynamic, co-created process rather than a static alignment.
Nonviolent Communication (Marshall Rosenberg)	Needs-based, empathetic dialogue that improves relational clarity and trust.	Supports Core Trust (feedback clarity) and Extended Relational Conditions (emotional safety, conflict repair).
Radical Candor (Kim Scott)	Honest, caring communication strengthens accountability and relationships.	Informs relational repair and feedback practices within Extended Dynamics and Conflict Mode.

6.2.4 Organizational Design and Governance

Source / Framework	Core Idea	Relevance to HCS
Sociotechnical Systems Theory (Trist & Emery)	Optimal performance arises when social and technical subsystems are jointly optimized.	Validates HCS's integration of human and procedural stability layers in the Core Model.
Lean Thinking (Womack & Jones)	Continuous removal of waste to improve flow and value.	Reflects in HCS's *Execution & Coordination* and *Flow & Focus* functions.
Kaizen	Continuous small improvements by all participants.	Embedded in *Adaptation & Learning* functions and the *Diagnostic Workflow*.
Adaptive Leadership (Heifetz)	Leadership is about enabling systems to adapt, not control.	Mirrors HCS's *Systemic Governance* and *Agency* principles across System Modes.
Organizational Learning (Argyris, Senge)	Organizations evolve when individuals learn within systemic feedback structures.	Reinforces *Systemic Learning*, *feedback closure*, and Level 5 meta-practices.

6.2.5 Power, Influence, and Organizational Politics

Source / Framework	Core Idea	Relevance to HCS
French & Raven's Bases of Power	Power derives from position, expertise, relationships, information, and personal influence.	Forms the basis of HCS's *political field*, explaining how influence affects cooperation beyond formal roles.
Organizational Politics Research (Mintzberg / Pfeffer)	Informal networks, hidden agendas, and resource negotiation shape real decision-making.	Validates HCS's distinction between *formal structure* and *actual influence flows* in Extended Conditions.
Stakeholder Theory (Freeman)	Organizations must manage diverse interests and legitimacy claims.	Supports Collective Extended Needs around fairness, representation, and legitimacy in shared systems.

6.2.6 Team Development, Interpersonal Dynamics, and Conflict

Source / Framework	Core Idea	Relevance to HCS
Group Development Stages (Tuckman)	Teams evolve through Forming, Storming, Norming, Performing (and later Adjourning), with each stage bringing characteristic tensions and needs.	Provides a developmental lens for team dynamics. HCS **System Modes** (Setup, Stabilization, Growth, Conflict, Reset) echo these stages but are defined at a **system level** by governance and cooperative function, not just intra-team behavior. Tuckman informs how modes may feel from the inside; HCS Modes define what kind of systemic work is appropriate.
Conflict Styles (Thomas–Kilmann)	People favor competing, avoiding, accommodating, compromising, or collaborating under tension.	Helps interpret relational distortions in Extended Conditions and supports Conflict Mode diagnostics.
Attachment Theory (Bowlby / Ainsworth)	Safety and past relational patterns influence trust and conflict behaviour.	Provides psychological grounding for individual Extended Needs (safety, belonging) and for interpreting reactions under stress.
Emotional Intelligence (Goleman)	Self-awareness and empathy shape communication effectiveness and conflict handling.	Aligns with relational conditions and leader behaviour in Extended Dynamics and Conflict Mode.

6.2.7 Philosophical and Ethical Foundations

Source / Framework	Core Idea	Relevance to HCS
Aristotle – Nicomachean Ethics	Virtue arises from practiced balance between extremes.	Influences HCS's pursuit of equilibrium between autonomy and control, stability and change.
Kantian Ethics	Human cooperation is grounded in respect for rational agency.	Echoes in HCS's concept of *Agency* and moral interdependence between cooperating parties.
Ubuntu Philosophy ("I am because we are")	Human identity and well-being are inseparable from community.	Embodies the relational worldview behind *Interdependence* and *Mutual Commitment*.
Phenomenology (Husserl / Merleau-Ponty)	Meaning is constructed through lived experience.	Supports HCS's principle that *Shared Understanding* emerges through interaction, not instruction.

6.2.8 Bridging Toward Practice

Source / Framework	Core Idea	Relevance to HCS
Agile / Lean / Scrum	Frameworks that operationalize adaptability, feedback, and incremental delivery.	Represent Level 4 practices that rely on stable lower-level HCS conditions (Preconditions, Needs, Functions).
3-in-3 SDLC Framework (3SF)	Systemic governance model for client–vendor ecosystems and software delivery.	Serves as an **applied derivative** of HCS: it uses HCS principles to design concrete contracts, roles, and practices across client–vendor–product triangles.
Team Topologies (Skelton & Pais)	Structures teams for flow and cognitive load balance.	In practice, can be evaluated through HCS's *Execution & Coordination* and *Boundary Management* functions.
VMOSA / OKR / Wardley Maps	Strategic planning and situational awareness frameworks.	Connect to *Strategic Alignment* and *Problem Discovery* functions at Level 3, translating cooperative intent into measurable direction.
Situational Leadership (Hersey–Blanchard)	Leadership behaviour must match follower readiness and capability.	Supports Extended Needs around growth, autonomy, and developmental alignment, especially in Growth Mode.
Liberating Structures	Creates equal-opportunity participation environments.	Supports fairness, belonging, and balanced contribution in Extended Dynamics and collective interventions.
RACI / Decision Records (ADR/CDR)	Makes decisions, ownership, and roles explicit.	Reinforces Core Conditions: clarity, coordination, and boundary safety; can be mapped directly to Distribution of Roles and related functions.

6.2.9 Reference Philosophy

HCS does not seek to **supersede** these theories — it **synthesizes** their enduring principles into a unified, practical system for diagnosing cooperation.

> **In essence:**
> **HCS = (Systems Thinking + Organizational Psychology) × (Communication + Adaptive Governance)**
> structured through the **Pyramid** (stability hierarchy) and **Matrix** (functional map),
> extended by **Human Dynamics** and **System Modes**,
> and made usable through **Diagnostics** and applied frameworks like **3SF**.

6.3 Practices Map

This chapter provides a **reference-level overview** of commonly used practices that support either:

- **Core Model functions** (Matrix + Pyramid Levels 1–3), or
- **Extended Human Dynamics** (Extended Conditions and Extended Needs).

Practices are included only to illustrate **where teams often look for help** once they have:

1. Used the **Diagnostic Workflow** to locate a structural function or level, and
2. Optionally applied **Diagnostic Dynamics** to understand political and psychological fields.

They are **optional**, **framework-neutral**, and **not prescriptive**.

Each practice is listed with:

- A brief **functional purpose** (what it helps with)
- Its **HCS mapping** (Core / Extended / Mixed and, where helpful, key needs or conditions)
- A simple **category** for navigation

The intent is to offer **direction**, not detailed instruction or endorsement.

6.3.1 Legend

Core – Primarily supports minimal structural conditions and needs of the HCS Core Model
Extended – Primarily supports motivation, relational dynamics, psychological or political influences
Mixed – Addresses both Core and Extended dynamics

Use this map **after** you know:

- which **Matrix cell / function** you are strengthening, and
- which **System Mode** you are in (Setup, Stabilization, Growth, Conflict, Reset).

6.3.2 Motivation & Values Alignment

Moving Motivators
Surfaces individual intrinsic motivators behind behavior and engagement.
HCS Mapping: Extended (Purpose & Direction; Recognition & Belonging; Autonomy & Coherence)
Category: Motivation

SCARF (David Rock)
Identifies status, certainty, autonomy, relatedness, and fairness drivers that shape threat/reward responses.
HCS Mapping: Mixed – Core (Trust, Feedback Loops); Extended (Trust & Safety; Recognition & Belonging)
Category: Motivation / Psychological Triggers

CliftonStrengths
Highlights personal strengths and energy sources to frame work around capabilities.
HCS Mapping: Extended (Growth & Evolution; Recognition & Belonging)
Category: Strengths Awareness

6.3.3 Interaction Styles & Communication

DiSC
Provides a vocabulary for communication and interaction preferences across styles.
HCS Mapping: Extended (Relational Conditions; Autonomy & Coherence); Mixed (Trust & Safety via predictability)
Category: Communication Style

MBTI / 16Personalities
Describes cognitive and interpersonal preference patterns for self-reflection and empathy.
HCS Mapping: Extended (Relational Understanding; Recognition & Belonging)
Category: Communication / Empathy

6.3.4 Developmental Readiness & Leadership Adaptation

Situational Leadership
Aligns leadership behavior with individual readiness and capability.
HCS Mapping: Extended (Growth & Evolution; Autonomy & Coherence); Mixed (Core: Distribution of Roles, Mutual Commitment)
Category: Leadership Adaptation

Competency Models
Clarify expectations, maturity levels, and developmental pathways for roles.
HCS Mapping: Core (Shared Understanding; Distribution of Roles); Extended (Growth & Evolution; Recognition & Belonging)
Category: Expectations & Development

6.3.5 Safety, Belonging, and Relational Health

Psychological Safety Surveys (Edmondson)
Measure perceived safety in communication and risk-taking within teams.
HCS Mapping: Core (Trust; Feedback Loops); Extended (Trust & Safety; Recognition & Belonging)
Category: Safety Assessment

Nonviolent Communication (NVC)
Provides a structured approach for respectful, needs-based dialogue.
HCS Mapping: Extended (Recognition & Belonging; Trust & Safety); Mixed (Core: Communication × Feedback Loops)
Category: Communication / Emotional Clarity

Empathy Mapping
Makes emotional states, expectations, and experiences visible in a group.
HCS Mapping: Extended (Relational Conditions; Recognition & Belonging; Purpose & Direction)
Category: Empathy & Insight

6.3.6 Conflict, Repair, and Difficult Conversations

Radical Candor
Supports direct yet caring communication to address tension and misalignment.
HCS Mapping: Mixed – Core (Feedback Loops; Mutual Commitment); Extended (Trust & Safety; Recognition & Belonging)
Category: Feedback & Conflict

Crucial Conversations
Offers a framework for navigating high-stakes or emotionally charged dialogue.
HCS Mapping: Extended (Trust & Safety; Autonomy & Coherence); Mixed (Core: Communication × Shared Understanding)
Category: Conflict Management

Conflict Styles Inventory (TKI)
Identifies preferred conflict responses to improve self-awareness and collaboration.
HCS Mapping: Extended (Relational Conditions; Trust & Safety)
Category: Conflict Behavior

6.3.7 Sensemaking, Purpose, and Collective Narrative

Team Canvas
Aligns teams on purpose, values, roles, and expectations in a visual way.
HCS Mapping: Core (Common Purpose; Shared Understanding; Distribution of Roles); Extended (Recognition & Belonging)
Category: Alignment

Appreciative Inquiry
Strengthens shared identity through exploring what works and why.
HCS Mapping: Extended (Purpose & Direction; Recognition & Belonging; Growth & Evolution)
Category: Narrative & Identity

Storytelling Workshops
Help teams articulate and align personal and collective narratives about the work.
HCS Mapping: Extended (Purpose & Direction; Recognition & Belonging); Mixed (Core: Shared Understanding)
Category: Sensemaking

6.3.8 Structural & Decision-Making Clarity

RACI / RASCI
Clarify roles, responsibilities, and decision ownership.
HCS Mapping: Core (Interdependence × Distribution of Roles → Coordination; Boundary Management); Extended (Autonomy & Coherence via clarity)
Category: Structural Clarity

Decision Records (ADRs / CDRs)
Make significant decisions visible and traceable over time.
HCS Mapping: Core (Communication × Shared Understanding → Language / Terms; Trust × Feedback Loops → Transparency); Extended (Fairness; Autonomy & Coherence)
Category: Decision-Making

Working Agreements
Create explicit local rules for collaboration, behavior, and expectations.
HCS Mapping: Core (Shared Understanding; Mutual Commitment; Feedback Loops); Extended (Predictability; Recognition & Belonging)
Category: Team Norms

6.3.9 Representation, Fairness, and Inclusion

Stakeholder Mapping
Makes influence, representation, and interests visible across roles and organizations.
HCS Mapping: Extended (Structural Conditions; Purpose & Direction; Recognition & Belonging; Political field awareness)
Category: Influence & Representation

Inclusion Audits
Evaluate whose voices, roles, or perspectives are missing or under-weighted.
HCS Mapping: Extended (Fairness; Recognition & Belonging; Structural Conditions); Mixed (Core: Interdependence × Mutual Commitment)
Category: Inclusion

Liberating Structures
Ensure equitable participation and more distributed contribution in group discussions.
HCS Mapping: Extended (Recognition & Belonging; Trust & Safety); Mixed (Core: Communication × Feedback Loops → Signal & Response)
Category: Group Facilitation

6.3.10 How to Use This Map

These mappings describe **where practices are often helpful**, not what you *must* use.

When working with HCS:

1. **Start with structure**
2. Use the **Diagnostic Workflow** to locate the issue in the Matrix and Pyramid.
3. Identify the **lowest unstable level** and the Cooperative Function to strengthen.
4. **Layer human dynamics if needed**
5. Use **Diagnostic Dynamics** to understand Extended Conditions, Extended Needs, and political/psychological fields.
6. Determine whether the issue is primarily **structural**, **collective**, **relational**, or **individual**.
7. **Check your System Mode**
8. In **Setup**, use practices that help clarify purpose, roles, and expectations.
9. In **Stabilization**, focus on practices that make feedback and coordination safer and clearer.
10. In **Growth**, use practices that extend autonomy, learning, and shared narrative without eroding safety.
11. In **Conflict**, choose practices that support safe confrontation and repair.
12. In **Reset**, use practices that help integrate learning and close or redesign cooperation with integrity.
13. **Select practices as hypotheses, not prescriptions**
14. Use this map to suggest **candidate practices** for experimentation.
15. Combine structural changes (roles, decisions, flows) with **targeted human practices** where appropriate.
16. **Avoid common misuses**
17. Do not use interpersonal tools to **hide structural injustice** or broken agreements.
18. Do not adopt practices “because others use them” without a clear **function-first** justification.
19. Do not expect any practice to compensate for missing **Level 1–2** foundations (purpose, trust, basic safety).

The Practices Map is intentionally **lightweight**.
Its purpose is to **orient** practitioners toward appropriate tools once the **systemic diagnosis is clear**,
not to replace the design and governance decisions defined by HCS.

6.4 Version and Licensing

This section documents the current version, license, and attribution principles for the **Human Cooperation System (HCS)**.
It ensures transparency, traceability, and consistency across all derivative works, educational materials, and frameworks built upon it — including the **3-in-3 SDLC Framework (3SF)**.

6.4.1 Version Information

Attribute	Description
System Name	Human Cooperation System (HCS)
Version	**V2.0**
Status	Stable — consolidated system release integrating Core Model, Extended Human Dynamics, System Modes, and Diagnostics.
Release Date	November 2025
Maintained by	3in3.dev
Repository	GitHub – vitar/hcs

Version 2.0 Summary

Version 2.0 is the first fully consolidated release of HCS, integrating the Core Model, Extended Human Dynamics, System Modes, and Diagnostics into a single, unified cooperation system.
While it preserves conceptual continuity with V1.0, it supersedes V1.0 as the primary reference.

Key additions and refinements:

- **Introducing System Modes as a new section**
- Defines five **System Modes** — Setup, Stabilization, Growth, Conflict, Reset — as **system-level stances** that describe *what kind of work* is appropriate given the current state of cooperation.
- Makes mode choice explicit so practitioners avoid:
  - trying to **grow** a system that still needs **stabilization**,
  - using **conflict tools** where a **reset** is needed,
  - or attempting a **setup** change when a **conflict** must be acknowledged first.
- Connects modes directly to the Core Model and Extended Dynamics, so each mode has clear structural focus and human-dynamics implications.
- **Dedicated Diagnostics section**
- **Diagnostic Workflow** – a structural loop from observation → Matrix mapping → Level check → Function → Practice → learning, grounded in the Core Model.
- **Diagnostic Dynamics** – a complementary view that layers Extended Conditions, Extended Needs, and political/psychological fields onto the structural diagnosis, without changing the Core Model itself.
- Diagnostics is framed as a **navigation pattern**, not a new theory layer.
- **Clear separation between theory and practice references**
- Practice examples and tools are moved into a neutral **Practices Map** in the Reference section.
- The Core Model remains explicitly **function-first** and framework-independent; practices are treated as optional examples, not prescriptions.
- **Positioning clarified**
- HCS is described explicitly as a **cooperation operating system**:
  - a stable conceptual base for diagnosing and designing cooperative systems,
  - able to support multiple applied frameworks (such as 3SF),
  - without becoming a delivery methodology or team process framework on its own.

Overall, V2.0 unifies all HCS elements into a coherent operating system and supersedes V1.0 as the authoritative model.
(Version 1.0 remains available as historical context and early formulation of the Core Model.)

Version 1.0 Summary

Version 1.0 consolidates the **foundational architecture** of the Human Cooperation System across two complementary layers:

- **Core Model** — the structural foundation of cooperation, including the 5×5 Matrix, Pyramid, Level Rule, and Diagnostic Workflow.
- **Extended Human Dynamics** — classification of collective and individual conditions and needs, and the influence of psychological and political vectors on real-world cooperation.
- **Reference Practices Map** — a non-prescriptive orientation linking common leadership and communication practices to relevant HCS functions.
- **Reference Section** — Glossary and theoretical sources covering systemic, psychological, relational, and governance influences.

This release establishes HCS as a **complete V1.0 system**:
a stable theoretical base (Core Model) and a complementary human-centered layer (Extended Dynamics), suitable for diagnostics, education, and derivative frameworks such as 3SF.

6.4.2 Licensing

The **Human Cooperation System** and all related documentation are licensed under the:

Creative Commons Attribution 4.0 International (CC BY 4.0)

You are free to:

- **Share** — copy and redistribute the material in any medium or format.
- **Adapt** — remix, transform, and build upon the material for any purpose, including commercial use.

Under the following terms:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made.
- **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

License Reference

Creative Commons Attribution 4.0 International (CC BY 4.0)

To attribute this work, please use the following reference:

> *Human Cooperation System (HCS) by 3in3.dev — licensed under CC BY 4.0 via GitHub repository vitar/hcs.*

6.4.3 Versioning Policy

- **Major versions (V2, V3, …)** introduce new theoretical constructs or expanded diagnostic models.
- **Minor revisions (e.g., V2.1)** include refinements, clarifications, or terminology alignment with derivative frameworks (e.g., 3SF).
- All published versions will remain **permanently available** for reference and citation.
- Future releases will aim to maintain **backward compatibility** with the foundational definitions, rules, and models of HCS.

6.4.4 Attribution Guidelines

If reusing or adapting HCS content:

1. Include a visible credit line referencing *3in3.dev* and the license type.
2. Retain section numbering and core definitions where possible to preserve structural consistency.
3. When combining HCS content with other frameworks or methods, clearly separate attribution and derived materials.
4. For translations or derivative works, add a note identifying the adaptation (e.g., “Adapted from the original Human Cooperation System V1.0 documentation licensed under CC BY 4.0”).

> © 2025 **3in3.dev**
> Licensed under **Creative Commons Attribution 4.0 International (CC BY 4.0)**
> https://creativecommons.org/licenses/by/4.0/

6.5 About the Author

Viktor Jevdokimov, Vilnius, Lithuania — Creator of 3in3.dev, HCS, and 3SF

Viktor Jevdokimov is a software engineering leader, systems thinker, and framework designer with over 30 years of experience in software product delivery, modernization, and team alignment.

He is the creator of the **Human Cooperation System (HCS)** and the **3-in-3 SDLC Framework (3SF)**, and founder of the **3in3.dev** initiative — an independent platform dedicated to advancing collaboration and alignment between **Client**, **Vendor**, and **Product** ecosystems.

Professional Background

- Began career supporting distributed banking software on DOS and Windows, developing a deep appreciation for troubleshooting and system design.
- Progressed through roles of **developer**, **architect**, **delivery lead**, and **practice lead**, working with international clients on modernization and cloud migration initiatives.
- Specializes in **Client–Vendor relationship design**, **project leadership**, and **delivery system diagnostics**.
- Advocates for *"Context before Method"* and *"Trust before Control"* as guiding principles of effective collaboration.

Creative and Personal Work

Beyond software, Viktor is an **active musician and live sound engineer**, performing and mixing with the *Great Things* cover band.
He approaches both sound and systems with the same mindset: striving for **clarity, balance, and authenticity**.

About 3in3.dev

3in3.dev is an independent research and publishing initiative founded by Viktor Jevdokimov.
It consolidates his experience and experimentation into open frameworks that help organizations improve how they **engage, deliver, and measure value** across collaborative ecosystems.

3in3.dev publishes:

- The **Human Cooperation System (HCS)** — theoretical foundation for cooperative system design.
- The **3-in-3 SDLC Framework (3SF)** — practical application of HCS principles in software delivery.
- Supporting tools, templates, and learning materials under an open license.

> "These systems aren't about control — they're about clarity, trust, and the shared intent that makes collaboration work."
> — Viktor J., Creator of 3in3.dev

Connect and follow on **LinkedIn** for updates and professional discussions.
For contact, collaboration, or speaking requests, visit **https://3in3.dev**.