Question Space System (QSS)

Viktor Jevdokimov, Vilnius, Lithuania

Table of contents

1. INTRODUCTION & QUICK START

1.1 Index

1.2 010 how to use this system

2. THEORY

2.1 Vision, Principles, and Beliefs

> **QSS (Question Space System)** is a way to *design the environment in which questions live*, not just the questions themselves.
> It treats inquiry as architecture: conditions first, then functions.

2.1.1 Purpose of QSS

The Question Space System exists to answer a simple but demanding need:

> **How can we consistently create conversations and reflections that lead to real clarity, alignment, and better consequences for humans – not just more information?**

QSS is:

- A **mental model** for what a "good" question space is.
- A **design discipline** for building such spaces in different domains.
- A **shared language** for talking about questions as structures, not isolated lines.

It does **not** prescribe one "correct" way to ask questions.
Instead, it offers a way to shape:

- **Orientation** – why we're asking.
- **Topology** – which dimensions of inquiry we activate.
- **Flow** – how we move through them.
- **Recursion** – how the space corrects and evolves itself.

2.1.2 Vision

Core Vision

> **Question spaces are aesthetic, friction-reducing environments for thinking together.**

The vision of QSS:

1. **Conversations and reflections feel clear, spacious, and safe**, even when topics are complex or emotionally loaded.
2. **Misalignment and hidden consequences become visible early**, before they turn into conflict, waste, or regret.
3. **Different perspectives can coexist and interact**, without collapsing into "who is right" battles.
4. **Inquiry becomes reusable**: once a solid question space exists for a type of situation, it can be adapted and replayed in future contexts.

In other words:

> QSS aims to make *high-quality thinking* and *low-friction collaboration* more repeatable, in any domain where humans need to understand and decide together.

2.1.3 Principles

These principles define how question spaces should be conceived, designed, and used.

Intention Before Inquiry

- Every question space starts with **clear intent**:
- What are we trying to understand or change?

- For whom does this matter?
- What kind of consequences are we actually trying to influence?
- The system prioritizes **purposeful curiosity** over aimless questioning.
- If intention is fuzzy, the first task is to clarify it, not to “collect more data.”

Implication: before designing questions, design the *why*.

Conditions Before Functions

- Borrowing from systemic thinking: **conditions enable functions**.
- Good outcomes (alignment, insight, better decisions) cannot be forced; they emerge when:
- People feel safe enough to speak.
- The problem is framed clearly enough to think about.
- The space is structured enough to avoid chaos, but loose enough to allow discovery.
- Question spaces therefore focus on **setting the right conditions**, not scripting specific answers.

Implication: design the space so that the right functions *have room* to appear.

Spaces, Not Lists

- A **question list** is linear. A **question space** is structured and multi-dimensional.
- QSS treats questions as elements in a **topology**:
- Clarification
- Assumptions
- Boundaries
- Value & Impact
- Contrast & Alternatives
- Causality & Consequence
- Layering / Abstraction
- Temporal
- Meta / Reflexive
- Different situations require different mixes of these dimensions.

Implication: we design **landscapes of inquiry**, not scripts.

Friction → Curiosity, Not Conflict

- Many problems in teams, partnerships, and personal choices are not “technical failures” but **communication and meaning failures**.
- Question spaces are built to:
- Transform **blame** into shared understanding.
- Transform **defensiveness** into safe exploration.
- Transform **anxiety** into clearer options.
- The preferred move is always:

> From “Who is wrong?” → to “What are we each seeing, and where do the views diverge?”

Implication: if a question space increases interpersonal friction, it is mis-designed.

Consequence-Aware Questioning

- Not all questions are equal. Some:
- Surface crucial risks.
- Reveal misaligned expectations.
- Expose unintended harms.
- QSS gives priority to questions that:
- Make **consequences visible** (for people, systems, and time).
- Illuminate **trade-offs** instead of hiding them behind abstractions.
- "Interesting but inconsequential" inquiry is treated as optional, not core.

Implication: question spaces are evaluated by their impact on *real-world outcomes*, not by intellectual elegance alone.

Multi-Perspective by Design

- Any meaningful situation (project, relationship, creative work) has multiple valid perspectives.
- Question spaces should:
- Make it easy to **name the perspectives** in the system (e.g., client/vendor/user/team/individual).
- Avoid collapsing everything into a single narrative too early.
- Hold tensions between perspectives long enough for useful integration.
- QSS encourages patterns like:
- "How does this look from X's perspective?"
- "What would Y describe as success or failure here?"

Implication: a good question space **respects plural viewpoints** and uses questions to map them, not erase them.

Evolving, Not Static

- Question spaces are **living artifacts**:
- They change as more is learned.
- They can be versioned, retired, or refactored.
- QSS assumes:
- No initial design is final.
- Recurring use reveals missing dimensions and unnecessary complexity.
- Recursion (meta-questions) is a first-class principle:
- "What did this question space miss?"
- "Which dimension did we overuse or neglect?"
- "What became clearer / more tangled after going through it?"

Implication: the system bakes in *self-correction* as a normal part of practice.

Minimal Sufficient Structure

- Over-structured spaces suffocate discovery. Under-structured spaces dissolve into noise.
- QSS aims for **minimal sufficient scaffolding**:
- Enough shape to avoid getting lost.
- Enough openness to allow unexpected insights.
- Preference is always for:
- Clear, simple flows.

- Small, composable building blocks that can be reused.

Implication: if a question space feels heavy or bureaucratic, it should be simplified.

Humane Pace and Depth

- People have limited cognitive and emotional bandwidth.
- Question spaces should:
- Move at a **humane pace**.
- Offer “shallow entry, deep continuation”: quick value first, depth available if needed.
- The system respects that:
- Not every situation requires full exploration.
- Sometimes the best next step is “enough clarity to act,” not exhaustive mapping.

Implication: the design should make it easy to stop at “good enough” without guilt.

2.1.4 Beliefs

These beliefs are not enforced as “truths,” but as **explicit assumptions** behind QSS.
They explain *why* the system is built the way it is.

1. **Most failures are failures of shared meaning, not raw intelligence.**
   Smart people routinely misalign because they never shared the same problem framing or consequence map.
2. **Human impact matters more than conceptual elegance.**
   A beautiful model that does not reduce friction, harm, or confusion is a decorative artifact, not a useful one.
3. **Questions are interventions.**
   Questions do not merely extract information; they change how people see themselves, each other, and the situation.
4. **Clarity is a social resource.**
   When clarity increases in a system, people coordinate better, trust more, and waste less.
5. **Good question spaces are teachable and reusable.**
   While intuition is valuable, relying solely on “talent for asking good questions” is fragile. Structures help others reach similar quality more reliably.
6. **No framework is universal.**
   QSS is meant to be adapted and integrated with other systems (e.g., collaboration frameworks, delivery frameworks, personal reflection practices), not worshipped as a single source of truth.
7. **Exploration is a renewable source of energy.**
   When friction drops and consequences are better understood, curiosity returns. That curiosity is a key driver for growth, creativity, and better decisions.

2.1.5 How to Use This Document

Use this file as:

- A **north star** for evaluating any question space you design:
- Does it reflect these principles?
- Which beliefs is it implicitly assuming?
- A **reference** when:
- Extending QSS into new domains.
- Creating new templates or GPT instructions.
- Explaining the system to collaborators.

Subsequent chapters (Core Architecture, Dimensions, Practical Construction) turn these Vision, Principles, and Beliefs into concrete tools and methods.

2.2 What Is a Question Space

This chapter defines what a **Question Space** is within the Question Space System (QSS) and distinguishes it from more familiar artifacts like checklists, interviews, and frameworks.

The goal is simple:

> Give you a clear, practical image of “a question space” so you can recognize it, design it, and reuse it.

2.2.1 Working Definition

A **Question Space** is:

> A deliberately designed environment of inquiry that shapes how clarity, alignment, and consequences become visible for the people involved.

Key aspects:

- It is an **environment**, not a single question or a linear list.
- It is **deliberate**: it exists because someone designed it with intent, not by accident.
- It is built to improve **clarity**, **alignment**, and **consequence awareness**.
- It is **context-specific**: you design a different space for client–vendor misalignment than for a personal career reflection.

You can think of it as the cognitive equivalent of **room acoustics**: - The room does not tell you *what to say*. - But it strongly influences *how well* you hear each other and *what becomes audible*.

2.2.2 Core Characteristics

A question space has a few essential characteristics that distinguish it from ad hoc questioning.

It Has Orientation (Intent)

A question space starts with a clear **why**:

- What are we trying to understand or change?
- For whom does this matter?
- What kinds of consequences are we trying to influence or avoid?

If intention is fuzzy, the space is weak.
If intention is explicit, the space can be evaluated against it.

It Has Topology (Dimensions)

Instead of being a flat list, a question space has **dimensions of inquiry**, such as:

- Clarification
- Assumptions
- Boundaries & Ownership
- Value & Impact
- Contrast & Alternatives
- Causality & Consequence
- Layering / Abstraction
- Temporal (past–present–future)
- Meta / Reflexive

A given space chooses a **subset** of these dimensions and organizes questions around them.

It Has Flow (Pathways)

A question space defines **how you move** through the dimensions:

- Where to start (e.g., context and intent).
- Which dimensions to visit first.
- When to go deeper vs when to move on.
- How to “land” on a usable level of clarity.

The same dimensions can be arranged into different flows depending on the context (kickoff, conflict, visioning, personal reflection, etc.).

It Has Recursion (Self-Correction)

A question space includes **meta-questions** that revisit the space itself:

- What became clearer after this round?
- What still feels fuzzy?
- Which perspective did we ignore?
- Which dimension did we overuse or avoid?

These recursive moves allow the space to **evolve** instead of staying static.

2.2.3 What a Question Space Is Not

It is important to contrast question spaces with related but different artifacts.

Not Just a List of “Good Questions”

A list of questions can be useful, but:

- It has **no explicit orientation** (why this list, for what purpose?).
- It has **no topology** (no indication of dimensions or relationships).
- It has **no flow** (no guidance on where to start, when to stop, how to adapt).
- It has **no recursion** (no built-in self-correction).

A question list is a pile of tools.
A question space is a **designed workshop** where those tools are used with intent.

Not Just an Interview Script

An interview script often:

- Optimizes for **information extraction**, not shared meaning.
- Is designed from **one side’s perspective** (the interviewer).
- Has a fixed sequence that may not adapt well to what emerges.

A question space, in contrast:

- Optimizes for **mutual clarity, alignment, and consequence awareness**.
- Explicitly holds **multiple perspectives** (e.g., client, vendor, user).
- Allows **adaptive movement** through dimensions based on what appears.

Not Just a Framework or Canvas

Frameworks and canvases (e.g., popular business canvases) provide fields to fill in.
They are useful, but:

- Often assume a fixed set of categories and a standard sequence.
- Can hide underlying assumptions about how problems “should” be structured.

A question space can integrate frameworks, but it remains:

- More **fluid** (dimensions can be added, removed, or reordered).
- More **context-driven** (you select dimensions that fit the situation, not the template).

2.2.4 What a Question Space Does in Practice

When used well, a question space:

- **Reduces friction**
  By turning confusion, accusation, and talking past each other into structured curiosity.
- **Reveals misalignment early**
  By surfacing hidden assumptions, conflicting expectations, and unspoken constraints.
- **Makes consequences visible**
  By connecting choices to likely impacts on people, systems, and time.
- **Supports better decisions**
  Not by providing answers, but by making the decision landscape clearer and shared.

In short:

> A question space changes the quality of thinking and relating, not just the quantity of information.

2.2.5 Three Short Illustrative Examples

These are intentionally simplified to make the shape visible.

Example 1 – Client–Vendor Misalignment (Software Delivery)

Orientation
- Understand why the client and vendor have different expectations about scope and speed.

Topology (chosen dimensions)
- Clarification, Assumptions, Boundaries, Value & Impact, Causality & Consequence.

Flow (sketch)
- Start with Clarification: “What problem are we each trying to solve?”
- Move to Assumptions: “What do we each assume about roles, constraints, and decision rights?”
- Then Boundaries: “What is in / out of scope for each side?”
- Then Value & Impact: “What does success/failure look like for each party?”
- Close with Consequences: “If we keep operating like today, what happens in three months?”

Recursion
- “What changed in our understanding after this space?”
- “Where do we still disagree, and is that acceptable or risky?”

This is a **question space**, not a one-time meeting agenda.
It can be reused and adapted for future client–vendor engagements.

Example 2 – Personal Role Transition Reflection

Orientation
- Decide whether to move from a senior IC role into a hybrid leadership role.

Topology (chosen dimensions)
- Value & Impact, Temporal, Layering, Meta.

Flow (sketch)
- Temporal: “Looking back, which work gave you the most durable satisfaction?”
- Value & Impact: “Whose lives do you want your work to impact most in the next 5 years?”
- Layering: “What changes at the system level if you shift roles (team, org, clients)?”
- Meta: “Are you framing this as ‘gain’ or ‘loss’, and what does that reveal?”

Recursion
- “After walking through this, what feels more stable in your decision? What still feels unsettled?”

Again, this is a **reusable reflective space**, not just a set of coaching questions.

Example 3 – Music Production Direction (Live vs Studio Vibe)

Orientation
- Decide whether to mix a live recording for “hi-fi precision” or “live energy and vibe.”

Topology (chosen dimensions)
- Contrast & Alternatives, Value & Impact, Causality & Consequence, Layering.

Flow (sketch)
- Contrast: “If we lean into ‘audiophile precision’ vs ‘raw live feel’, what changes in sound and perception?”
- Value & Impact: “What matters more to this audience: accuracy or emotion?”
- Layering: “How does this choice affect band identity, future gigs, and your own satisfaction?”
- Consequence: “If you choose one direction now, how does it influence expectations for the next release?”

Recursion
- “Did these questions make your preferred direction clearer or more conflicted?”
- “Is there a hybrid direction we didn’t initially consider?”

This space helps **surface trade-offs** and **align intent** with artistic and audience impact.

2.2.6 How This Chapter Connects to the Rest of QSS

- The **Vision, Principles, and Beliefs** explain *why* question spaces matter and what values they serve.
- This chapter defines *what a question space is and isn’t*, including how it behaves.
- The next chapters will:
- Describe the **Core Architecture** (Orientation, Topology, Flow, Recursion) in more detail.
- Catalogue the **Dimensions of Inquiry** and how to use them.
- Show how to **construct and navigate** question spaces in practice across different domains.

You can treat this chapter as the reference point whenever you ask:

> “Are we actually designing a question space here,
> or just collecting a list of questions?”

2.3 Core Architecture: Orientation, Topology, Flow, Recursion

This chapter describes the **core architecture** of a Question Space in QSS.

Every question space, regardless of domain, can be understood as four interacting layers:

- **Orientation** – Why we are asking.
- **Topology** – What dimensions of inquiry we activate.
- **Flow** – How we move through those dimensions.
- **Recursion** – How the space learns and corrects itself.

You can think of it as:

> Orientation sets the intent →
> Topology shapes the landscape →
> Flow guides the journey →
> Recursion improves the map while you travel.

The rest of QSS builds on this structure.

2.3.1 Orientation

Orientation is the layer that defines *why* the question space exists at all.

Without Orientation, questions become either random or manipulative. With Orientation, the space can be evaluated and adjusted against a clear purpose.

Purpose of Orientation

Orientation:

- Clarifies what we are trying to understand or change.
- Connects inquiry to **human and system consequences**.
- Makes the space accountable to a concrete intent.

Typical intentions include:

- Diagnose misalignment.
- Explore options and trade-offs.
- Clarify values and priorities.
- Design a transition or reset.
- Reflect on experience and learn.

Core Orientation Questions

Before designing any question space, QSS encourages answering questions such as:

- What situation is this space for?
- What decision, change, or understanding do we want to enable?
- For whom does this matter most?
- What kinds of consequences are we trying to influence or avoid?
- What is explicitly *out of scope* for this space?

These do not need to be long. A few sharp sentences are usually enough.

Orientation Examples

Client–Vendor engagement: - “Enable both sides to see where expectations, constraints, and success criteria differ, so we can adjust the engagement before it fails.”

Personal role reflection: - “Understand whether moving into a hybrid leadership role is aligned with my values, energy, and desired impact over the next five years.”

Creative decision (mix direction): - “Clarify the trade-offs between ‘hi-fi precision’ and ‘live energy’ for this recording, and choose a direction that best serves the music and its audience.”

In practice, Orientation is often captured as a short paragraph at the top of a question space artifact.

2.3.2 Topology

Topology describes the **shape of the space**: which dimensions of inquiry are present and how they relate.

Instead of a flat, undifferentiated list of questions, a question space is structured around **dimensions** such as:

- Clarification
- Assumptions
- Boundaries & Ownership
- Value & Impact
- Contrast & Alternatives
- Causality & Consequence
- Layering / Abstraction
- Temporal (past–present–future)
- Meta / Reflexive

These are detailed in the next chapter. Here, we focus on how they work together as a topology.

Purpose of Topology

Topology:

- Ensures that inquiry covers **relevant dimensions**, not just the loudest or most comfortable ones.
- Reduces the chance of blind spots (for example, talking only about features and never about consequences).
- Makes it easier to adapt and reuse spaces by swapping or reweighting dimensions.

Selecting Dimensions

A question space rarely uses all possible dimensions. Instead, it deliberately chooses a subset based on:

- The **nature of the situation** (diagnosis, strategy, reflection, creative choice).
- The **main risks** (misalignment, hidden constraints, emotional overload, unclear impact).
- The **time available** (quick scan vs deep exploration).

Examples:

- For early project misalignment:
- Clarification, Assumptions, Boundaries, Value & Impact, Causality & Consequence.
- For personal reflection:
- Value & Impact, Temporal, Layering, Meta.
- For creative direction:
- Contrast & Alternatives, Value & Impact, Layering, Causality & Consequence.

Multi-Perspective Topology

Topology can also carry **perspectives** explicitly:

- Client vs Vendor vs End User.
- Individual vs Team vs Organization.
- Artist vs Audience vs Market.

For example, the same dimension (Value & Impact) can be explored from multiple perspectives:

- “What does success look like for the client?”
- “What does success look like for the vendor?”
- “What does success look like for end users?”

Good topology design makes it easy to see where perspectives align and where they diverge.

2.3.3 Flow

Flow defines **how we travel** through the topology.

A question space is not just a set of dimensions; it also provides a sensible **pathway**:

- Where to start.
- What to explore first.
- When to deepen.
- When to switch dimensions.
- How to land on usable clarity.

Purpose of Flow

Flow:

- Prevents the conversation from becoming chaotic or exhausting.
- Matches the **cognitive/emotional load** of participants.
- Creates a sense of progress: “We moved from confusion to some kind of shared understanding.”

Flow does not have to be rigidly linear, but it should be **legible**.

Typical Flow Patterns

Some common patterns QSS recognizes:

- **Context → Clarification → Boundaries → Value & Impact → Consequences → Meta**
  Useful for project kickoffs and alignment sessions.
- **Situation recap → Assumptions → Boundaries → Causality & Consequence → Trade-offs → Next steps**
  Useful for conflict or escalation.
- **Future vision → Value & Impact → Contrast with present → Causality → Experiments / options → Meta reflection**
  Useful for strategy or visioning conversations.
- **Past → Present → Future**
  Combined with Value & Impact and Meta, useful for personal reflection.

Depth Management

Flow can also be designed at multiple depths:

- **Shallow pass**: one or two questions per dimension, to get quick shared picture.
- **Deep dive**: more detailed questions within a single critical dimension (for example, assumptions or consequences).

A well-designed flow makes it possible to stop after a shallow pass if time or energy is limited, without collapsing the whole space.

2.3.4 Recursion

Recursion is the self-correcting layer of a question space.

It introduces **meta-questions** and **revision moves** that allow the space to evolve in response to what is discovered.

Purpose of Recursion

Recursion:

- Prevents the space from becoming rigid or dogmatic.
- Ensures that unexpected insights are integrated back into the structure.
- Allows reuse: question spaces can be refined across multiple uses.

In practice, recursion is what turns a question space from a static template into a **living tool**.

Meta-Questions

Recursion is often implemented through short, simple meta-questions, such as:

- What became clearer after this round of questions?
- What feels more confusing or unsettled now?
- Which perspectives did we not hear or explore?
- Which dimension did we overuse? Which dimension did we ignore?
- Did we drift away from our original Orientation? Should we adjust it?

These can be asked:

- After a specific dimension.
- At the end of a flow.
- When the conversation feels stuck or emotionally loaded.

Structural Adjustments

Recursion can also change the **structure** of the question space:

- Updating Orientation if the real problem turns out to be different.
- Adding or removing dimensions in the topology.
- Reordering flow to reflect what actually works better.
- Splitting a complex space into two simpler ones.

Example:

- You design a question space for “project risk diagnosis”.
- In practice, you notice that people get defensive in the Assumptions segment.
- Recursion: you insert a short Values & Impact segment before Assumptions to establish shared goals and reduce defensiveness.

Over time, this turns into a **new version** of the question space that is more humane and effective.

2.3.5 How the Layers Work Together

The four layers are not steps; they are **aspects** of the same artifact.

A simple way to keep them in mind:

- **Orientation** – Are we clear on why this space exists?
- **Topology** – Have we chosen the right dimensions and perspectives?
- **Flow** – Is there a humane, understandable way to move through them?
- **Recursion** – Do we have mechanisms to notice and correct misfits?

When designing or evaluating a question space, you can use these as a quick checklist:

- If conversations feel aimless → Orientation is weak.
- If blind spots keep appearing → Topology is incomplete or unbalanced.
- If people feel overwhelmed or lost → Flow needs adjustment.
- If the space never improves across uses → Recursion is missing.

The next chapters go deeper into:

- The catalog of **Dimensions of Inquiry** (Topology).
- The **Modes and Lifecycle** of question spaces (how they evolve through setup, stabilization, and growth).
- Practical methods to construct and refine spaces using this architecture.

2.4 130 dimensions of inquiry

2.5 140 modes and lifecycle

2.6 150 patterns and anti patterns

3. PRACTICE

3.1 200 practical overview

3.2 210 quickstart builder

3.3 220 context intake and orientation

3.4 230 designing the topology dimensions

3.5 240 designing flow and pathways

3.6 250 recursion and iteration loops

3.7 260 domain playbooks software delivery consulting

3.8 261 domain playbook personal reflection

3.9 262 domain playbook music production

4. REFERENCE

4.1 300 glossary

4.2 310 question pattern cheatsheets

4.3 320 templates and checklists

4.4 330 example question spaces case studies

4.5 Version and Licensing

This section documents the current version, license, and attribution principles for the **Question Space System (QSS)**.

4.5.1 Version Information

Attribute	Description
System Name	Question Space System (HCS)
Version	**V1.0**
Status	WIP — work in progress.
Release Date	November 2025
Maintained by	3in3.dev
Repository	GitHub – vitar/qss

Version 1.0 Summary

Version 1.0 consolidates the **foundational architecture** of the QSS.

4.5.2 Licensing

The **Question Space 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:

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

4.5.3 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.
- 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.

4.5.4 Attribution Guidelines

If reusing or adapting QSS 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 QSS 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”).

4.6 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**.