

1 Minimal Viable Team: Why Five

1.1 The Central Question

How many people does a team need so that the departure of one person does not kill the project? Not “how many is convenient” and not “how many is conventional”—but how many is *structurally necessary* for the system to survive the loss of any single element?

The answer: **five**.

1.2 Why Not Three

Any meaningful project requires three things: someone must *see* the whole, someone must *translate* the vision into a plan, someone must *execute* with hands. Vision, design, execution. Three is the minimum for existence. But if one person leaves—the loop breaks. No vision—the team becomes blind. No designer—the vision remains a dream. No builder—the plan remains paper. A triad is a skeleton, not an organism. It works while everyone is in place. But the moment one person gets sick, burns out, or leaves—everything falls apart.

1.3 Why Not Four

A quartet is already more resilient: even with maximum disagreement within a team of four, consensus cannot fall to zero. There is a guaranteed “floor.” But when one of the four leaves, the quartet becomes a triad—and a triad, as we have seen, is on the edge. Four is a one-time safety margin: it will withstand one blow, but not two.

1.4 Why Exactly Five

When one of five leaves, a quartet remains—and a quartet already has guaranteed consensus above zero. All three basic functions (vision, design, execution) remain covered. The loop deforms but does not break. The team is sick, but does not die—and is capable of *self-regeneration*: finding a new person or redistributing roles.

Five is a triad of structure plus a pair of redundancy. Skeleton plus immune system.

1.5 Five Roles: Stations of One Loop

Any project is a cycle. Someone sees the possibility. Someone turns it into a plan. Someone realizes it. Someone checks the result. Someone ensures that everyone sees the same thing. And then—again: refined vision, refined plan, refined result. A spiral, not a line.

Five stations of this spiral—five roles.

1.5.1 Visionary

The one who holds the whole. Not a manager—but a person who sees *the space of possibilities*. He does not point where to go, but shows where one could end up. His work is not to prematurely narrow the world but to keep the horizon open. Founder, strategist, product owner.

His main question: “*What are we creating and why?*”

The Visionary’s danger: if he stops believing in the project—nothing will happen, because there is no direction. If he believes *absolutely*, without a shred of doubt—he becomes blind and drags the team into a wall, unable to see that the world has changed.

1.5.2 Analyst

The one who turns vision into a plan. From the infinite space of possibilities, he *selects* a concrete path. This is inevitable loss: choosing one thing means giving up everything else. The quality of this choice determines whether the team will reach its destination. Architect, systems analyst, lead designer.

His main question: “*How exactly should this be implemented?*”

The Analyst’s danger: if he processes every input signal identically—he is useless. A good analyst *distinguishes*: he sees the difference between “looks like the right thing” and “is the right thing.”

1.5.3 Builder

The one who does the work. Realizes the plan in matter—code, text, product, result. He works against inertia: the more that is already done, the harder it is to change. Every line of code written, every process fixed—this is simultaneously both result and resistance to future changes. Developer, executor, technical leader.

His main question: “*What exactly is already done?*”

The Builder’s danger: he creates not only the result but also its weight. A conscious builder makes the result *light*—minimum inertia with maximum function. An unconscious builder cements every step, and after six months the team cannot move.

1.5.4 Validator

The one who checks. The most underestimated role—and the most necessary. Without him, the result is created but not *evaluated*. The team does not know whether what was made matches what was intended. Without feedback, there is no learning, no refinement, no spiral—there is only a straight line into the unknown. Tester, QA, reviewer, critic.

His main question: “*Does the result match what was intended?*”

The Validator’s danger: if he always says “everything is good”—feedback is zero, the team does not learn. If he always says “everything is bad”—feedback is maximum but meaningless, the

team is demoralized. A good validator *measures the distance* between expectation and reality—concretely, precisely, without emotional coloring.

1.5.5 Coherence Keeper

The one who synchronizes. His work is to ensure that all five see *one and the same project*. Not the same task (that is too narrow), but one and the same *reality*: shared understanding of goals, terms, priorities, state of affairs. Facilitator, scrum master, team lead, coach.

His main question: “*Are we still seeing one and the same thing?*”

The Coherence Keeper directly impacts the *lifetime of the project*. The higher the consensus—the longer the configuration lives. The lower—the more “noise”: random failures, conflicts, misunderstandings, lost tasks. The Coherence Keeper transforms the team’s chaotic Brownian motion into a meaningful path.

The Coherence Keeper’s danger: he cannot be absolutely certain that he “knows how it should be.” Such a Coherence Keeper suppresses disagreement instead of working with it. The ideal Coherence Keeper is one who can *accept disagreement* and help the team pass through it, rather than silence it.

1.6 What Happens When One Leaves

Who Left	What Breaks	Who Picks It Up	How
Visionary	Horizon Narrows	Coherence Keeper + Analyst	Coherence Keeper Maintains Consensus, Analyst Recovers Vision from Experience
Analyst	Planning Stops	Builder + Visionary	Builder Plans near Horizon, Visionary Adjusts Direction
Builder	Realization Stops	Analyst + Validator	Analyst Takes Execution (Worse, but Not Death), Validator Maintains Quality
Validator	No Feedback	Coherence Keeper + Visionary	Coherence Keeper Gives Process Feedback, Visionary Gives Meaning Feedback

Who Left	What Breaks	Who Picks It Up	How
Coherence Keeper	Consensus Falls, Noise Grows	Visionary + Validator	Visionary Holds Common Picture, Validator Calibrates through Checking

The key: when *any one* person leaves, the remaining quartet maintains consensus above zero and all three basic functions remain covered. The loop deforms but does not break.

1.7 Four Qualities of Each Participant

Each person in the team has four “dimensions,” and all four are important *simultaneously*. If even one equals zero—the person’s contribution is nullified entirely, no matter how high the other three are.

Focus—the ability to concentrate on the task. If there is no focus—a person is physically present but does not work. Body in the office, mind on the phone.

Emotional Resilience—the ability to work under stress, uncertainty, conflict. If there is no resilience—any crisis nullifies the contribution. A person is paralyzed or explodes.

Internal Consistency—alignment of words and actions, overt and hidden goals. If a person says one thing and does another—he destroys team consensus from within. This is the most dangerous: invisible from outside, but the system rots.

Scale of Vision—the level from which a person sees the system. A junior sees the task. A senior sees the module. An architect sees the system. A CEO sees the market. If scale is zero—a person does not see the context of their work and makes decisions blindly.

Profile by roles:

Role	Focus	Resilience	Consistency	Scale	Most Important
Visionary	Medium	High	High	Maximum	Scale
Analyst	Maximum	Medium	High	High	Focus
Builder	High	Maximum	Medium	Medium	Resilience
Validator	High	Medium	Maximum	Medium	Consistency
Coherence Keeper	High	High	High	High	Balance of All Four

1.8 Two Ways to Die

A team dies in two ways—and both are connected to extremes.

1.8.1 Burnout: Belief Falls to Zero

When a key participant stops believing in the project completely—he “goes to work but does not work.” His contribution to the overall result is nullified. Worse: if his state is far from others, he *lowers* team consensus. Not just does not help—actively harms. A burned-out person on a team is more dangerous than an absent one.

1.8.2 Dogmatism: Belief Has Grown to Absolute

When a participant “knows everything” and cannot doubt—he is frozen. Cannot learn, cannot adapt, cannot accept that the world has changed. Absolute certainty and absolute disbelief are equally deadly: both halt learning.

1.8.3 Condition for Life

A team is alive when each participant is *sufficiently convinced to act*—and *sufficiently open to learn*. Not zero and not one. Between them. The living space between doubt and certainty—the only place where knowledge is possible.

1.9 Numerical Example: Builder Leaves

Suppose a team of five. Everyone believes in the project, but to varying degrees: the Visionary—more than anyone (0.85 out of 1), the Validator—slightly more cautious (0.70). Small variation—high consensus (0.93 out of 1). Probability of success—99.1%.

The Builder leaves. A quartet remains. Consensus drops to 0.90—but still high. Probability of success—98.1%. Project lifetime—ten times the baseline. The team is deformed, but alive. The Analyst temporarily takes on part of the Builder’s work, the Coherence Keeper maintains synchronization, the system seeks a replacement.

If the team were a triad—the departure of one would nullify everything.

1.10 Summary

Five is not a magic number and not a management hack. It is the *minimal configuration at which the loss of one element does not destroy the system*. A triad is the skeleton (vision, design, execution). A pair is the redundancy (feedback and synchronization), allowing the loop not to break under stress.

A project is never “completed” fully—every sprint, every iteration, every release is a turn of the spiral, refining the configuration. And at every turn, all five are needed: the one who sees; the one who designs; the one who does; the one who checks; and the one who ensures that all five—including himself—see one and the same thing.

A self-consistent team is one that reproduces itself: closes the loop “vision → plan → result → check → synchronization → refined vision”—and enters a stable orbit. It does not stand still (that is death) and does not fly into chaos (that is also death). It moves in a spiral—each turn slightly more precise than the previous, but never the last.