

Dev to Lead

Guiding Without Managing

Erik Beaulieu





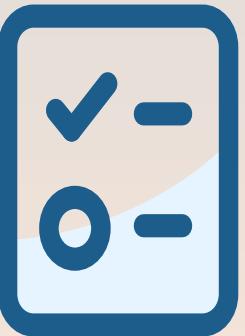
ROUNDTABLE DISCUSSION



**Who here has ever felt
like a lead...
without having the title?**



**What aspect of leadership are
you most eager to explore
today?**



What is your current job title ?

WHAT ABOUT ME ?

- Over 20 years in web development
- Bachelor degree in Software Engineering
- Tech Lead at Openmind Technologies
- Speaker at Confoo 2025



[erik-beaulieu](#)



[matrix818181](#)

PERSONAL LIFE

- Dad of 2 young kids
- Foster family with the DYP
- Proud fan of the Montreal Canadiens !



[erik-beaulieu](#)



[matrix818181](#)

MY PASSIONS

- Continuous learning and skill-sharing
- Finding simple solutions to complex technical challenges
- Mentoring and helping junior developers grow to become better individuals



[erik-beaulieu](#)



[matrix818181](#)



TODAY'S AGENDA

- O1 Understand leadership**
- O2 Listening before Leading**
- O3 Build Strong Partnerships**
- O4 Leading Through Conflict**
- O5 Balancing Code & Influence**

01

Understand leadership

LET'S DEBUNK A MYTH...

**Does leadership equal
management ?**





Leadership



Management

Leadership



Management

LET'S THINK ABOUT IT TOGETHER

**What is leadership in the context
of software development?**



LEADERSHIP

The ability to **influence** and
guide a group **to achieve**
common goals by
providing direction,
support, alignment and
ownership for results.



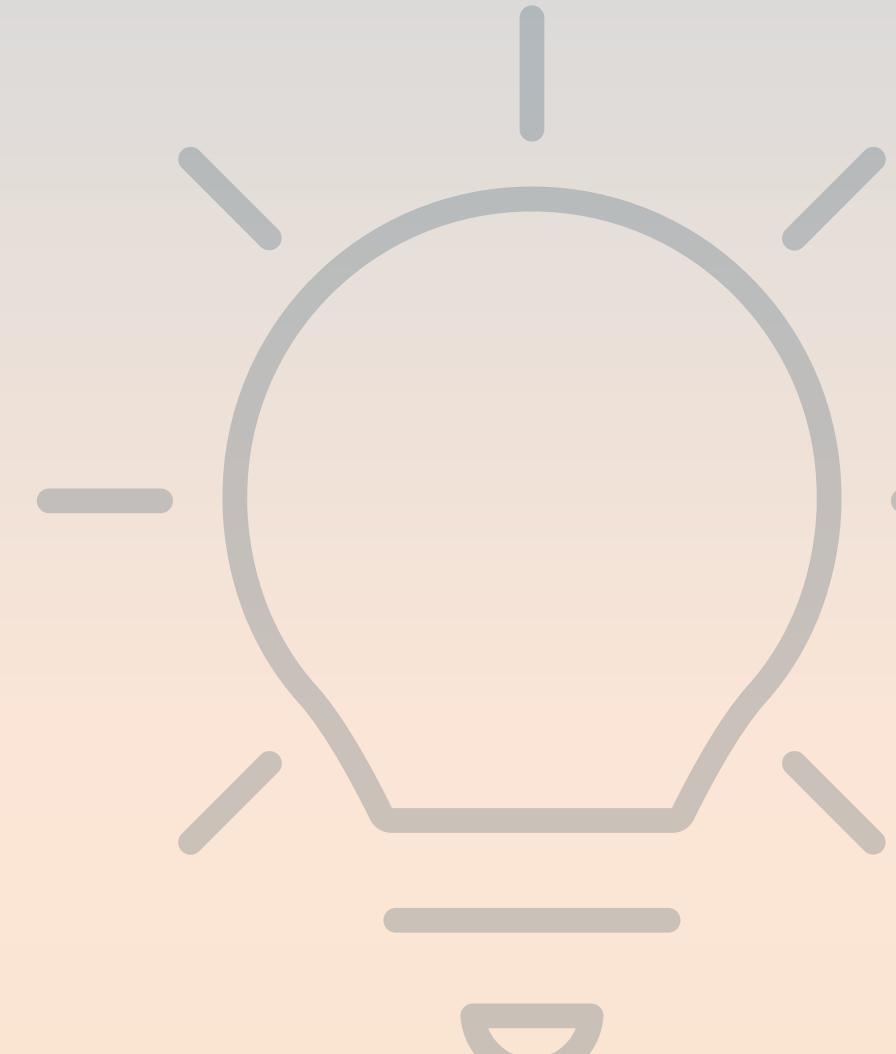
LET'S THINK ABOUT IT TOGETHER

**What is management in the
context of software development?**



MANAGEMENT

Management is the discipline of **aligning people, priorities, and processes** to deliver **reliable outcomes** within **constraints of scope, time, and quality.**



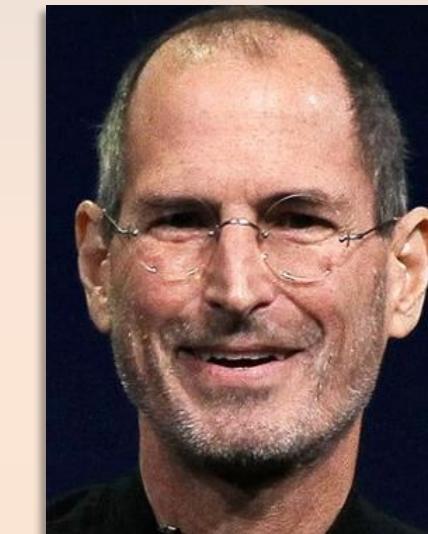
QUICK AND DIRTY...



Management makes **things run.**



Leadership makes **people move.**



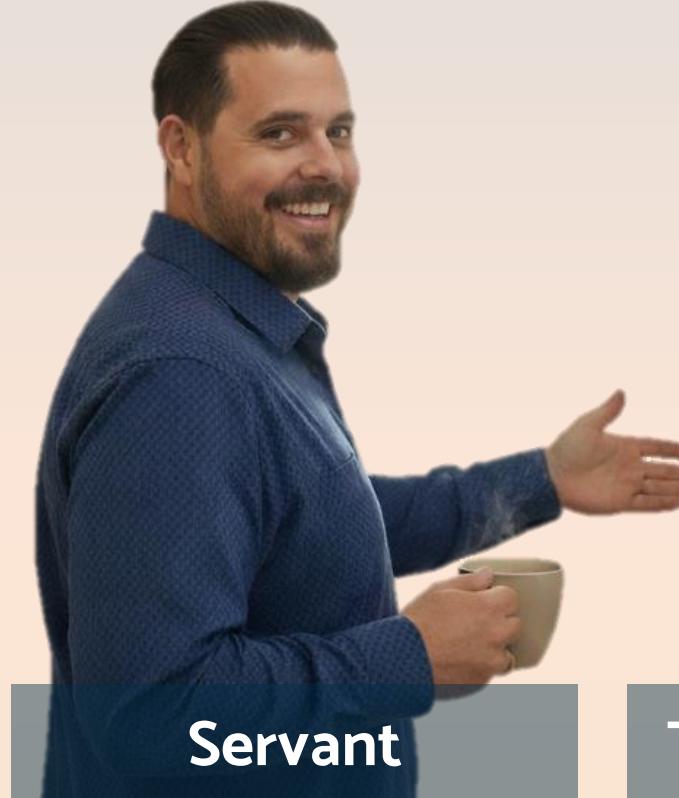
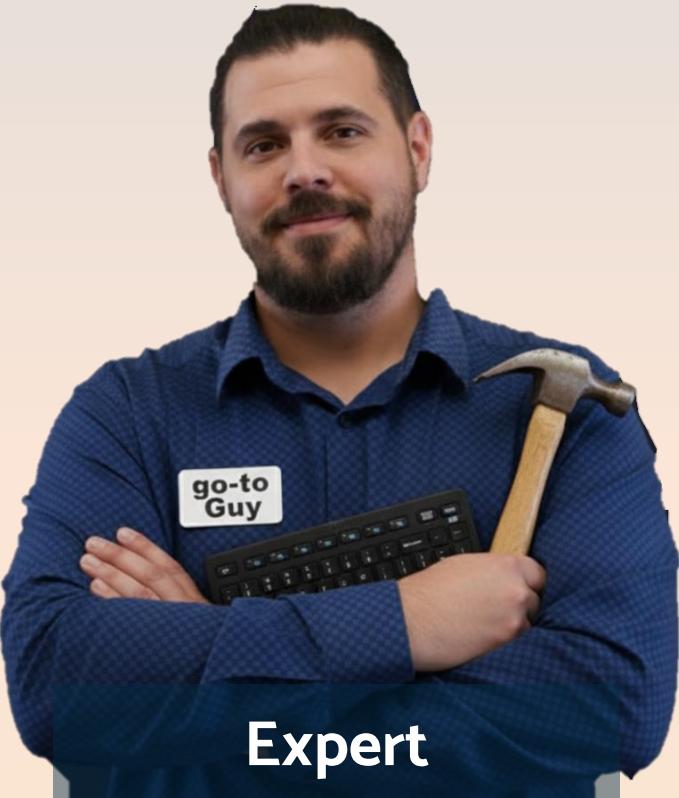
Management is about persuading people to do things they do not want to do, while leadership is about inspiring people to do things they never thought they could.

— Steve Jobs —

AZ QUOTES



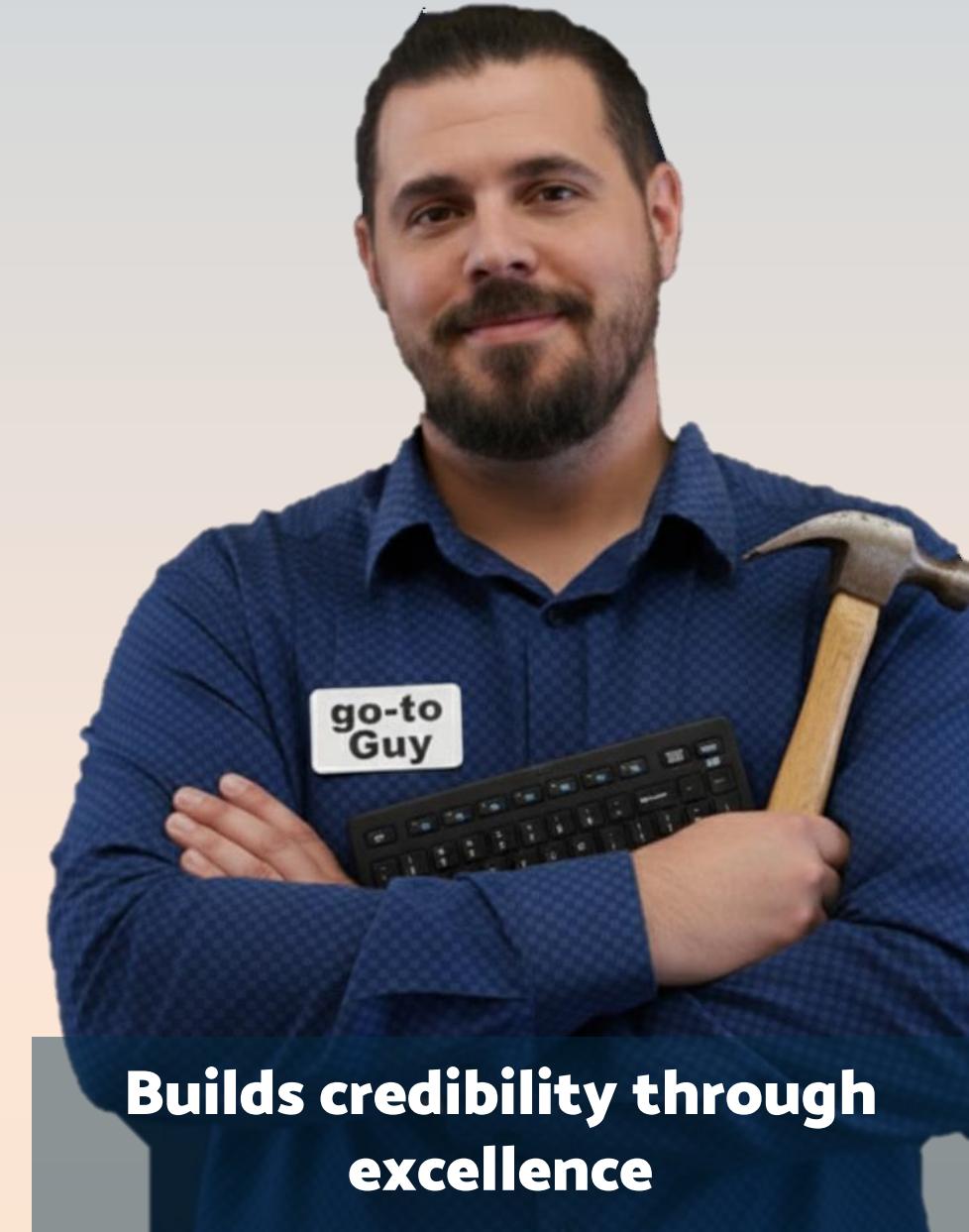
3 TYPES OF LEADERSHIP



3 TYPES OF LEADERSHIP

Expert Leader

- > **Focus:** Influence, shared responsibility, innovation, growth.
- > **Motivation:** Internal (purpose, growth).
- > **Communication:** Bidirectional (dialogue, listening).
- > **Relation:** Mutual trust. Employees are partners, collaborators who actively contribute to success.

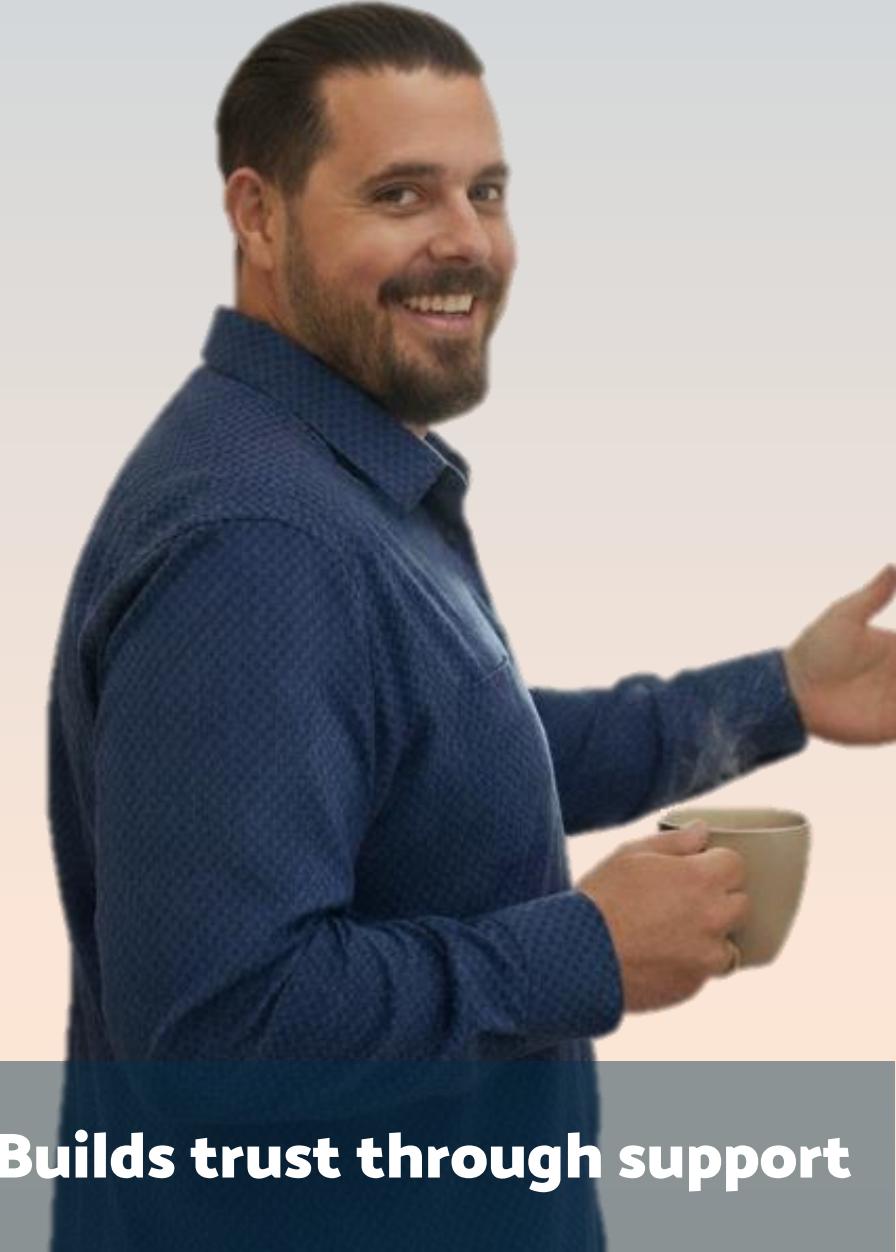


Builds credibility through
excellence

3 TYPES OF LEADERSHIP

Servant Leader

- > **Focus:** Empowering others and removing obstacles
- > **Motivation:** Help the team succeed before yourself
- > **Communication:** Empathetic, listening-driven, supportive
- > **Relationship:** Built on trust, humility, and service.



Builds trust through support



3 TYPES OF LEADERSHIP

Transformational Leader

- > **Focus:** Inspiring change and innovation
- > **Motivation:** Elevate people and vision beyond comfort zones
- > **Communication:** Visionary, purpose-driven, energizing
- > **Relationship:** Built on inspiration, growth, and shared ambition.



Builds momentum through
purpose

so... ?

Which type of leader do you relate to most?

- > Expert ?
- > Servant ?
- > Transformational ?
- > All of them ?





IT'S A CAREER CHOICE, NOT THE NEXT LEVEL

Leadership is a direction, not a promotion

- > Senior Developer → deeper technical mastery
- > Leader Developer → broader human impact
- > Both paths require discipline
- > Both paths are honorable
- > **The difference is where you expand your responsibility**



NOT A PATH FOR EVERYONE

Is Leadership for you ?

Leadership might be for you if:

- > You care how decisions are made
- > Broken collaboration frustrates you
- > You want impact beyond your own work
- > You enjoy helping others grow

Leadership might not be for you if:

- > You prefer depth over breadth
- > Interpersonal complexity drains you
- > You value technical solitude
- > You want ownership of code, not culture

WHAT ACTUALLY CHANGES?

The Shift From Developer to Leader

When you're a Developer:

- > Optimize for correctness
- > Optimize for performance
- > Optimize for elegant code

When you start Leading:

- > Optimize for clarity
- > Optimize for alignment
- > Optimize for momentum

02

Listening before Leading

LET'S START WITH...

What is active listening ?



ACTIVE LISTENING

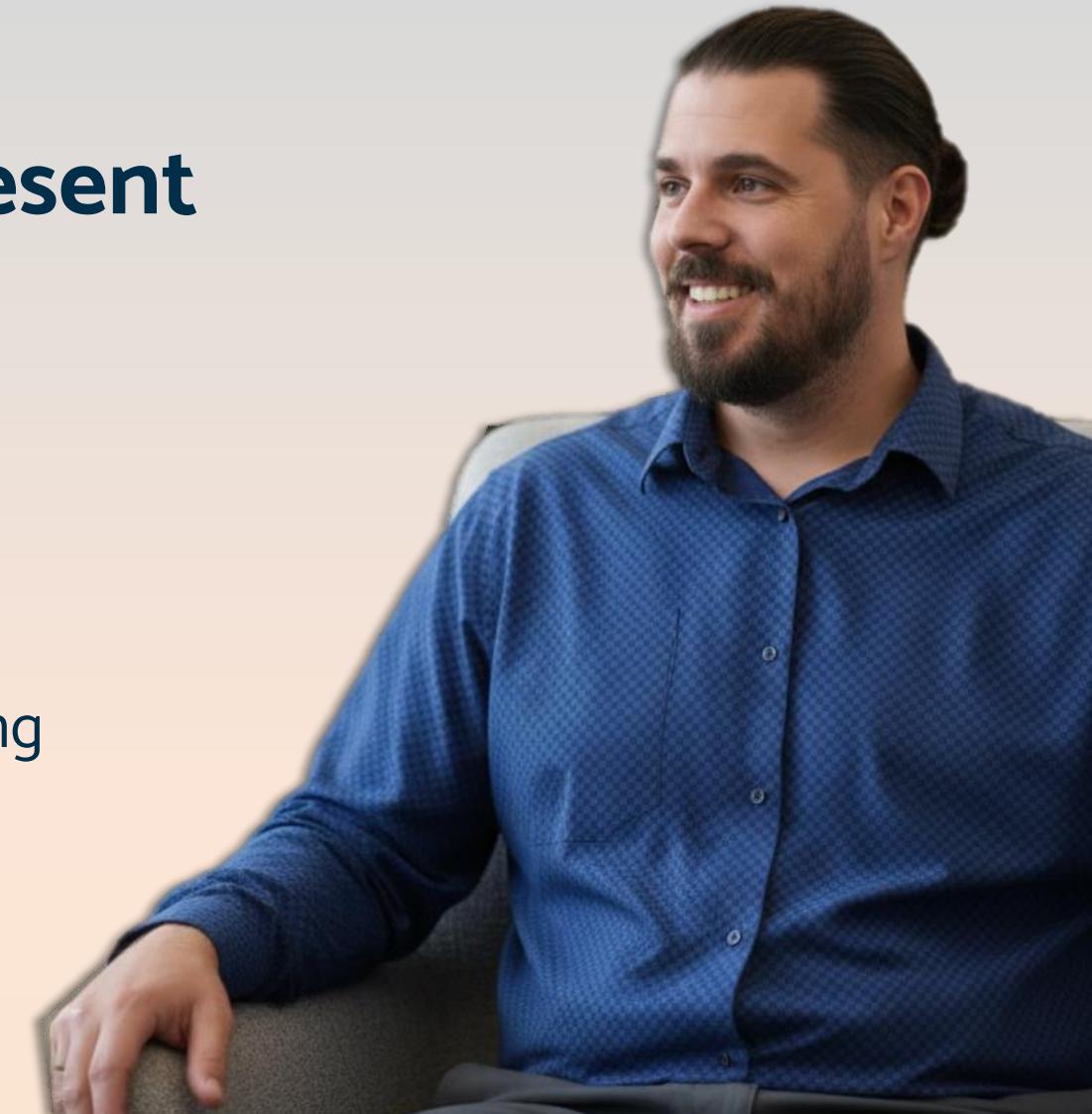
Practice of listening with **full attention** and empathy **to truly understand** what the other person is expressing, whether it be their words, emotions, or intentions, **rather than judging or responding too quickly.**



BECOME AN ACTIVE LISTENER

Stop thinking, be fully present

- > Turn down your mental chatter and pause your inner dialogue
- > Create a space for the other person
- > Maintain eye contact... without staring
- > Avoid multitasking (no typing, no notifications) and adopt an open, grounded posture, show that you care





BECOME AN ACTIVE LISTENER

Show that you're listening

- > Use subtle signs to demonstrate engagement
- > Use verbal cues



BECOME AN ACTIVE LISTENER

Paraphrase to confirm your understanding

- > “So, if I understand correctly...”
- > “Let me make sure I got this right...”
- > “Just to confirm, you’re saying that...”
- > “What I’m hearing is that...”
- > “From your point of view, it sounds like...”
- > “To put it another way, you mean that...”





BECOME AN ACTIVE LISTENER

**Curiosity creates understanding,
speed creates assumptions**

- > Quick answers kill understanding
- > Curiosity opens the door to meaning instead of “What should I say next?”
- > Try “What’s really behind what they’re saying?” without assuming
- > Proper mindset can turn debates in explorations





BECOME AN ACTIVE LISTENER

Acknowledge emotions without trying to fix them

- > Emotions are information, not problems
- > Resist the urge to minimize or rationalize it
- > Acknowledge gives people permission to feel and it can defuse tension



BECOME AN ACTIVE LISTENER

Listen to what's not being said...

- > Words only tell part of the story
- > Pay attention to unspoken signals like pauses, sighs, hesitations or tone shifts
- > Often, it reveals what the person is reluctant or unable to express



INFLUENCE WITHOUT AUTHORITY REQUIRES BACKBONE.

Leadership Without Courage Becomes People-Pleasing

- Say “I don’t know” publicly
- Challenge respectfully
- Admit mistakes first
- Protect the team under pressure
- Take responsibility before credit



THEY'RE ABOUT IDENTITY.

Most Technical Conflicts Aren't About Code

- > Architecture becomes personal
- > Code reviews trigger ego
- > Debate turns into defense
- > “Best practice” can hide insecurity
- > **Leaders separate problem from ego**



03

Build Strong Partnerships

LET'S GET THIS STRAIGHT...

What is partnership ?



PARTNERSHIP

A **mutual agreement** between equals where the **parties involved** are **considered autonomous**.

They choose to **share skills and resources** to **achieve a common goal**.



HOW TO BUILD A WORK AGREEMENT ?

Align on intent

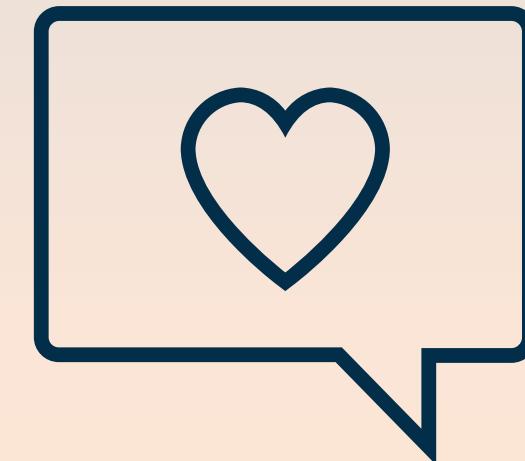
- > You need to clarify what you're both trying to achieve and why it matters
- > It's about shared purpose and not about deliverables
- > “What do we both need from this partnership to make it work sustainably?”



HOW TO BUILD A WORK AGREEMENT ?

Surface individual needs

- > Bring the implicit into the explicit world
- > Clarity prevents 80% of future conflicts
- > It's not about pleasing each other, it's about understanding each other's operating system



HOW TO BUILD A WORK AGREEMENT ?

Define concrete commitments

- > Turn principles into observable behaviors
- > Best agreements are measurable and actionable
- > Make expectations explicit so accountability feels fair, not personal
- > “What behaviors will help us uphold our intentions and how will we know if we’re off track?”



HOW TO BUILD A WORK AGREEMENT ?

Review and adjust

- > Keep the agreement alive and adaptive
- > It isn't a static statement, it's living contract of trust
- > “How do we keep this partnership healthy as we grow?”



04

Leading Through Conflict

**CONFFLICT IS NOT A FAILURE OF LEADERSHIP.
IT'S A TEST OF IT.**

Leading Through Conflict

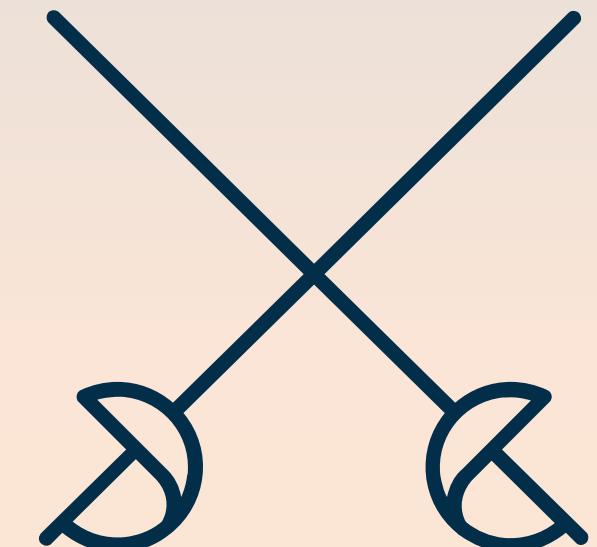
- > Technical disagreement
- > Architectural tension
- > Code review friction
- > Resistance to change
- > Emotional escalation



**TWO SENIOR DEVELOPERS.
ONE ARCHITECTURAL DECISION.**

A Real Scenario

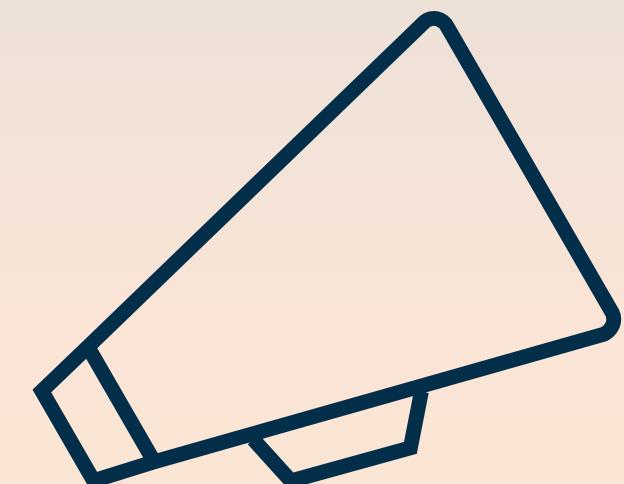
- > Both technically strong
- > Both convinced they're right
- > The team starts taking sides
- > Delivery slows down
- > Tension rises



WHEN CONFLICT APPEARS.

What Most People Do

- > Argue harder
- > Escalate to authority
- > Shut down the discussion
- > Avoid the topic
- > Let ego take control



RELUCTANCE ≠ UNWILLINGNESS.

Resistance Is Information

- > Fear of risk
- > Fear of losing control
- > Attachment to identity
- > Lack of clarity
- > Lack of trust



NOT ALL FRICTION IS ABOUT CODE.

Diagnose Before You Decide

- > Is this about architecture... or ego?
- > Is this about quality... or trust?
- > Is this about risk... or comfort?
- > Is this about speed... or fear?



WITHOUT AUTHORITY.

Leadership Moves Under Conflict

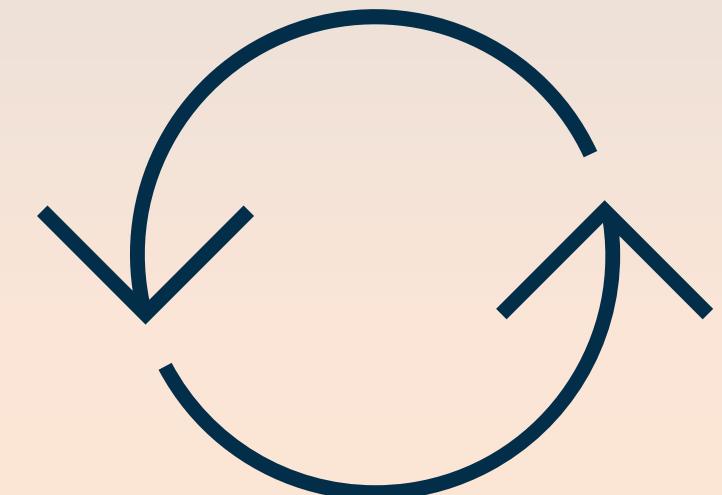
- > Slow down the tempo
- > Surface the real concern
- > Reframe around shared goals



CLARITY BEATS INFINITE DEBATE.

Leadership Isn't Endless Consensus

- > Clarify decision ownership
- > Explain the reasoning
- > Invite dissent, once
- > Close clearly
- > Protect the relationship



NOT THE DECISION. THE WAY YOU HANDLED IT.

What the Team Will Remember

- > Your tone
- > Your calm
- > Your fairness
- > Your transparency
- > Your consistency



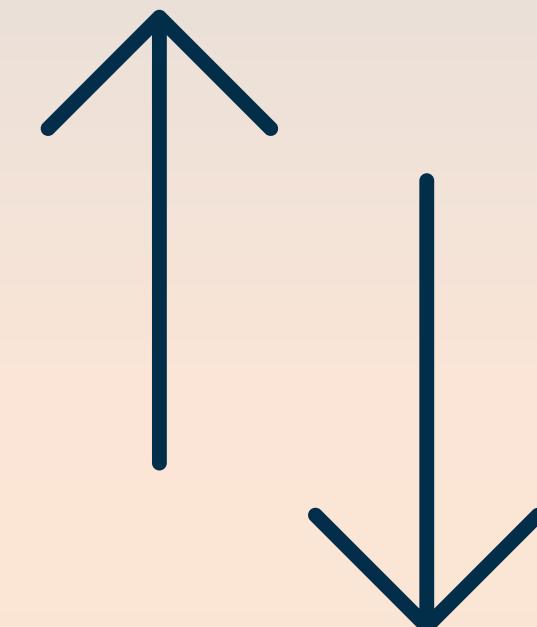
05

Balancing Code & Influence

FIND BALANCE

Leadership in tech isn't about coding less

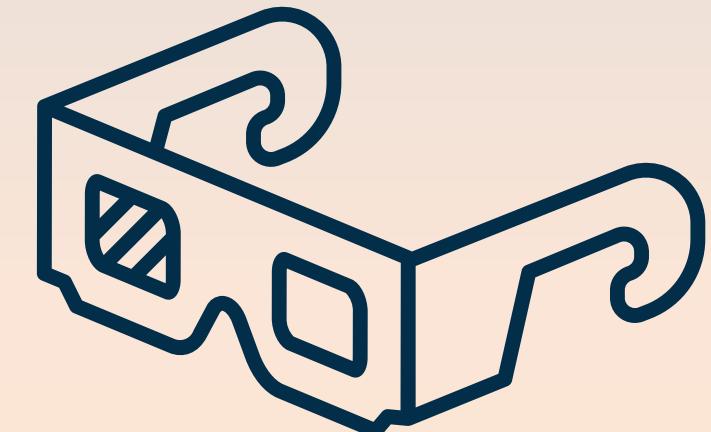
- > It's about amplifying impact
- > You need to stay technically credible while creating clarity and momentum for others
- > Think of it as a signal you send, not a position you hold



FIND BALANCE

Replace control with shared understanding

- > Translate complexity into understandable direction
- > Ask “why” before “how”
- > Frame decisions so everyone, not just seniors, understands the reasoning
- > Remove ambiguity before removing blockers.
- > A good lead clarifies the framework but lets the team own the decisions.



FIND BALANCE

Model the behavior you expect

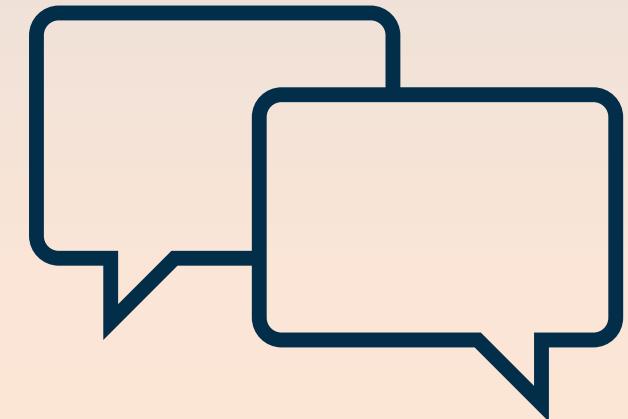
- > People copy what you do, not what you say
- > Admit your mistakes publicly. Don't be afraid to show vulnerability
- > Give code reviews that teach, not that shame
- > Prioritize long-term quality over short-term delivery and explain why
- > Your tone under pressure defines your culture



FIND BALANCE

Elevate conversations, not just code

- > Shift team discussions from “how to implement” to “why it matters.”
- > Ask context questions like: “How does this align with our architecture vision?”
- > Encourage debate without ego to create psychological safety
- > Redirect style wars toward shared understanding and value.



FIND BALANCE

Build autonomy, not dependence

- > Leadership success = team performs even when you're away
- > Delegate decision with clear boundaries and a strong feedback loop
- > Explain the why before the what
- > Let others present in reviews to allow them to shine
- > Create a space where people can take risks because progress happens when failure is safe.



HOW TO DESCRIBE A TECH LEAD

A **great** Tech Lead **doesn't lead people away from code**, they use **code, clarity and calm to lead people toward impact.**

You don't stop coding to lead...
You code the culture through your actions!

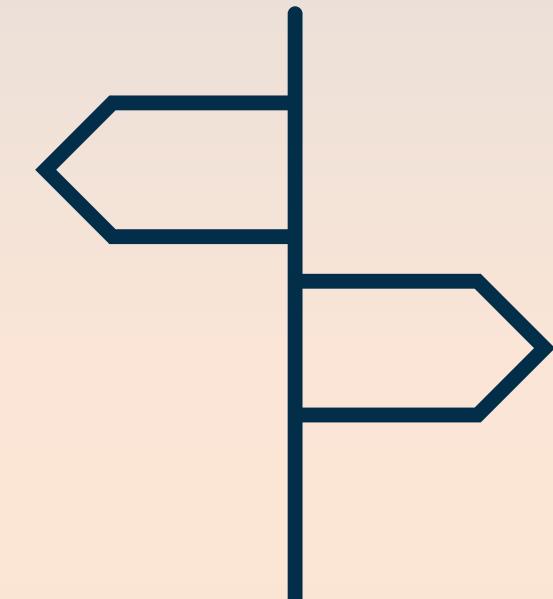


It's time to wrap up.

LET'S STEP BACK

Before we close

- > Leadership isn't a promotion
- > It's not a reward
- > It's not a title
- > It's not mandatory
- > **It's a choice**



HERE'S WHAT IT REALLY MEANS.

If you choose this path

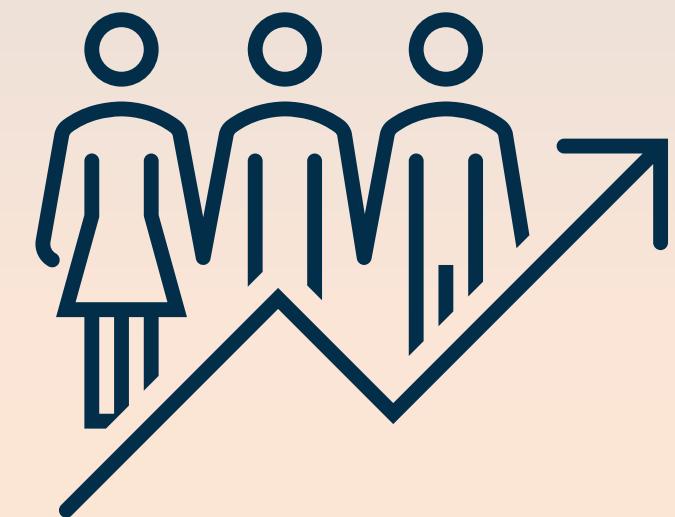
- > You'll deal with conflict
- > You'll face resistance
- > You'll be misunderstood
- > You'll make imperfect decisions publicly
- > You'll carry emotional weight



EXPERIENCE SOMETHING DIFFERENT.

But you'll also

- > See people grow
- > Watch the team move without you
- > Create clarity under pressure
- > Build trust through consistency
- > Shape culture through behavior



WHAT KIND OF IMPACT DO YOU WANT?

The real question...

- > Depth of code?
- > Breadth of influence?
- > Individual excellence?
- > Collective growth?



WHY NOT ANOTHER MANIFESTO?

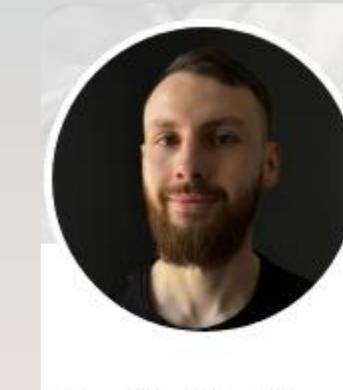
Dev to Lead Manifesto

-  Influence is earned through consistency
-  Clarity beats control
-  Courage beats comfort
-  Conflict tests your character
-  Autonomy is the real measure of leadership
-  Leadership is a choice of impact
-  It begins the moment you care beyond your own code



QUESTIONS ?

**WANT
TO
LEARN
MORE?**



Daniil Shykhov



Anton Martyniuk

THANK YOU!



erik-beaulieu



matrix81818

