

Code Reviews

Empathetic Code-Centric Communication



*Your code is a sh***

Purpose of the presentation

- Discuss what is an empathetic code-centric approach and its benefits
- Share opinions in an open discussion

Code, in its essence, is communication.

Addy Osmani

Communication

One big important part of programming is **communication**.

Effective communication is as vital to programming as code itself.

A good communication:

- 🤝 enables developers to understand requirements
- 🤝 is focus on solutions
- 🤝 helps in document the work in ways that others can understand

Code Reviews

Code reviews matter beyond just code quality and they are more than just catching bugs 🐛.

Code reviews are conversations that shape our team's culture.

Every comment we write:

- Builds or breaks trust
- Encourages or discourages innovation
- Defines how we grow together as engineers

Code Reviews: let's start with a thought in mind

Think about the last time you got feedback that made you excited to improve your code, versus feedback that made you want to defend your choices.

Now we're going to explore together what makes that difference.

What we're going to cover

- 🎯 Code-centric communication in Code Reviews
- 🎯 Developer contexts
- 🎯 Examples of code-centric comments vs not code-centric comments
- 🎯 Some polls to share experiences all together
- 🎯 Some tips
- 🎯 Benefits of empathetic code-centric approach

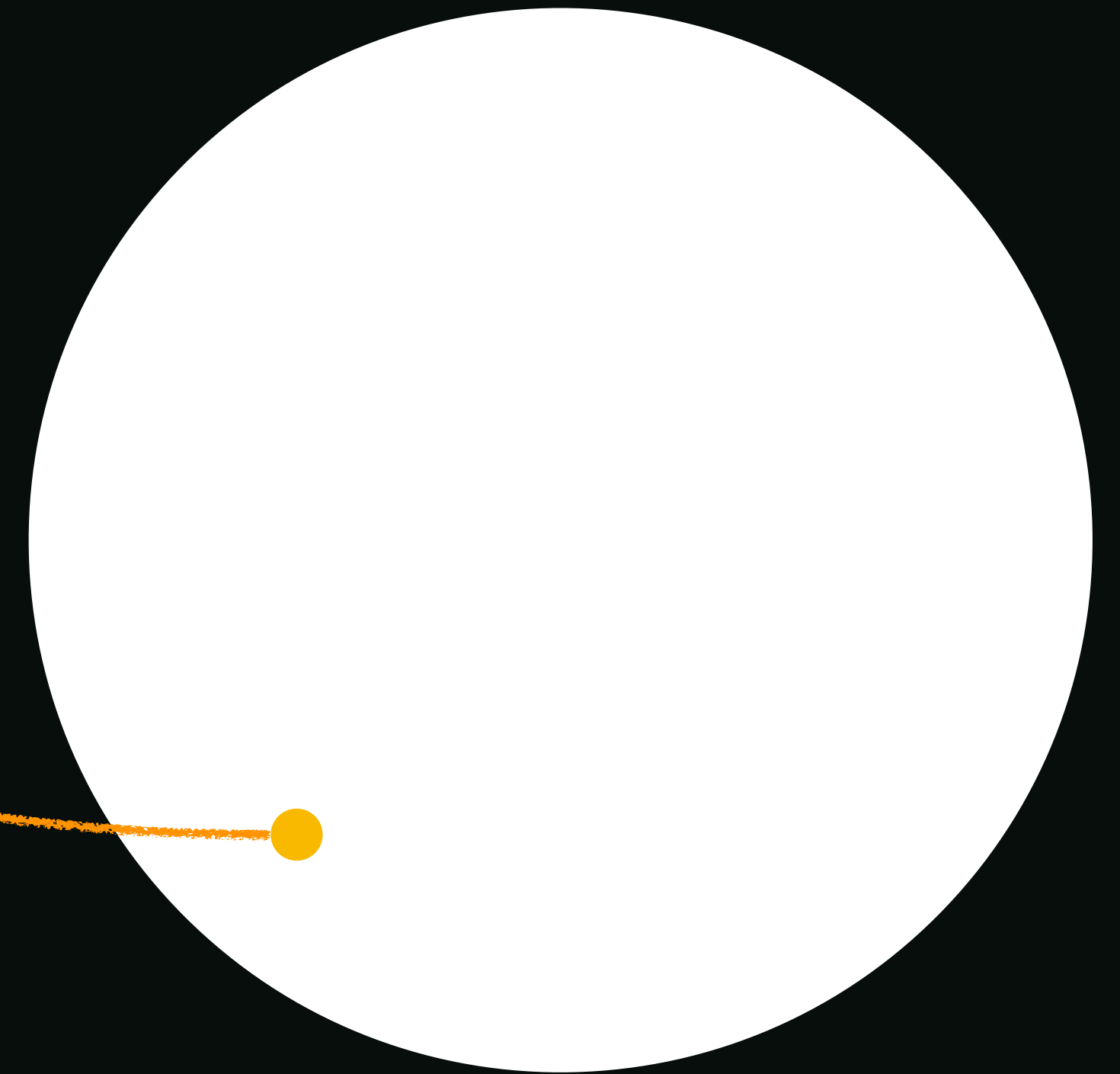
What we're NOT going to cover

- Deep analysis of the communication differences based on country, personal experiences, etc
- Different Code reviews techniques
- Decide a common protocol we all have to follow

Communication issues 🚩

🧰 Everyone possesses unique sets of skills and abilities

🧰 This is what we know about another person



Communication issues

Cultural Considerations

- Direct vs. Indirect Communication Cultures
- Language and Communication Style Variations
- Feedback Delivery Preferences

Hierarchical Impact

- Traditional hierarchies: junior vs senior
- Flat organizations: focus on expertise rather than title

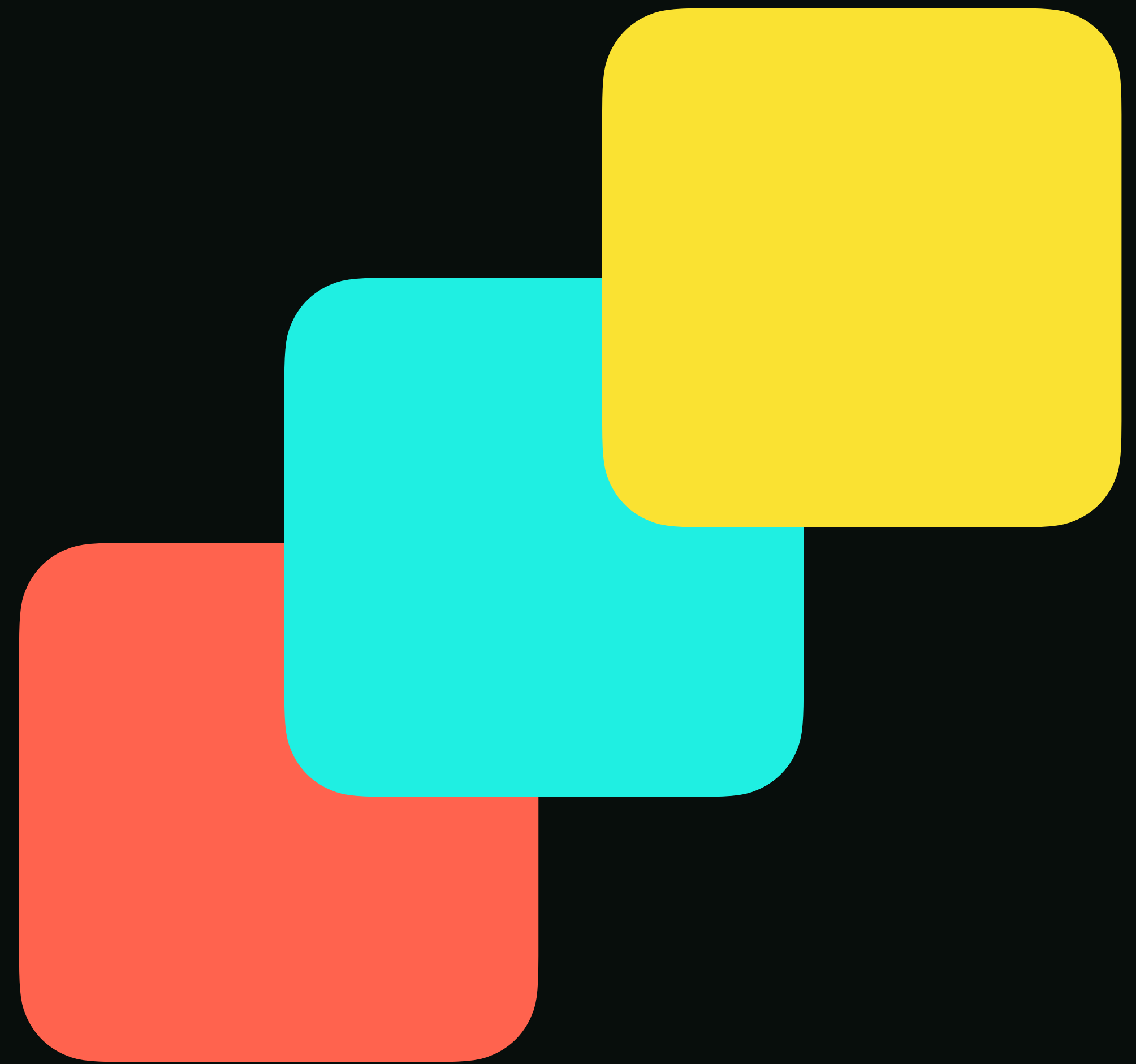
Remote vs In-person

- Async communication challenges
- Lack of body language and tone
- Time zone considerations

Communication issues

In a multinational company with non-native English speakers, communication occurs across three distinct layers:

1. Our language
 2. The other one language
 3. English language
-
1. The message we intend to convey
 2. The meaning interpreted by the receiver
 3. The words we actually use to express ourselves



Non-Native English Speakers

Code-centric approach

A common language for developers

Using code as our common language helps team members understand each other clearly, regardless of their native language or cultural background.



Code-centric Language in Code Reviews

Code-centric

- Focus on the code and outcomes
- Uses technical specifications and requirements
- References concrete examples and documentation
- Discusses measurable results and behaviours

Promotes collaborative problem-solving

Non-Code-centric

- Focus on the developer ("You did...", "Your code...")
- Uses abstract descriptions and subjective language
- References vague concepts without examples
- Emphasizes personal preferences over technical standards

Often leads to defensive responses

Let's dive in Code Review Topic

✗ Non-Code-centric

✓ Code-centric

“This loop condition could lead to infinite execution, potentially impacting server stability”



"You wrote an infinite loop that could crash our server"



1# Poll

How often are you able to write a code-centric comment?



Always



Frequently



It requires a lot of
effort

Example



✗ "Why did you you write the function the wrong way?"

✓ "Is this an alternative syntax for a function? I'm familiar with the standard implementation pattern, but I'm interested in understanding if there are specific advantages to this approach."

- Creates defensiveness
- Lack of explanation
- Doesn't acknowledge multiple valid approaches may exist



- Uses neutral and objective language
- Considers different implementation options
- Expresses genuine interest in the technical decision
- Invites explanation without presuming error





Example



✗ "Why did you implement it this way?"

✓ "Could you explain the reasoning behind this implementation?"



✗ "This code needs to be completely rewritten"

✓ "Here are some suggestions for improving maintainability"



Example



✗ "You're not following our standards"

✓ "This differs from our established patterns - here's a link to our guidelines"



✗ "You always use variables with those names. This makes the code hard to read."

✓ "Consider using a more descriptive variable name like 'timestamp' or 'currentTime' to make the code's intent clearer."

Examples: key points

Focus on Solutions

- ✗ "This is wrong"
- ✓ "Here's an alternative approach"

Be Specific

- ✗ "The code needs improvement"
- ✓ "Adding error handling would help catch file access issues"

Provide Examples

- ✗ "Use better names"
- ✓ "Consider names like 'timestamp' instead of *"

Explain Why

- ✗ "Don't do this"
- ✓ "This approach could cause issues because..."

Offer Options

- ✗ "Rewrite this"
- ✓ "Here are two ways we could handle this..."

Empathetic communication 🤝

When working closely with a partner, **the stakes are high.**

Empathetic communication helps us to do two important things:

- Keeping high code standards
- Maintaining the relationship

Taking time to leaving clear comments is one way to code with empathy.

Code-centric Empathetic communication results 🤝

A code-centric approach which fosters empathetic communication brings to our team:

- Clarity
- Maintainability
- Collaboration
- Speed
- Knowledge
- Creativity

You are not your code

2# Poll

Have you ever heard about this sentence?



Yes and I used it



They told me



Never heard of it

You are not your code

...And don't take it personally

✓ Used Well:

- Encourages constructive feedback
- Helps separate personal identity from work, fostering growth
- Promotes a collaborative, code-centric approach

✗ Used Poorly:

- Justifies harsh or dismissive criticism
- Ignores the emotional impact of feedback
- Undermines a colleague's effort instead of supporting improvement

You are not your code

...And don't take it personally

💡 Key Takeaway

- Use this phrase to build a supportive, respectful coding culture—never as an excuse for unkindness
- Remember code-centric communication is not about removing responsibility. You are not your code, **but the code you write is your own responsibility.**

When is *personal*

“Please don’t add inline style”



```
<div class="bg-blue-500 text-white p-4" style="margin: 20px;">  
  <!-- other tags here -->  
</div>
```

Experience Bias

When we’re experienced in a specific area, we assume others share the same knowledge of best practices.


Implicit Knowledge Trap

We often take best practices for granted, forgetting that not everyone has the same background.

Constructive Guidance Over Correction

A review should educate, not just enforce.

 “Please don’t add inline style”

 “Inline styles can make maintenance harder. I would use the corresponding Tailwind class here. See this [reference].”

Code-centric **EXTREME** 🔥

Writing comments **purely based on the code**, without considering how the receiver will interpret or react to it.

“The code is unacceptable”

3# Poll

What would be your first reaction if you received a comment like this?



It makes me laugh



It makes me angry



I'm confused

Code-centric + empathetic approach

=

**Better knowledge sharing and a
stronger team culture**

Benefits of adopting empathetic code-centric approach

- **Company Culture**

Code reviews should reflect shared values, collaboration, and growth.

- **Foster a Learning Mindset**

Reviews are not just for catching mistakes but for continuous learning and improvement.

- **Build a Sustainable Process**

Over time, a good review system boosts code quality and team dynamics.

From a wise Grandma



*Do something well done or do something badly:
it will take you the same amount of time*

Tips

- ✨ Focus on requirements over preferences
- ✨ use WE instead of YOU
- ✨ avoid the usage of “I prefer”
- ✨ understand the importance of explaining the "why" behind suggestions
- ✨ share alternative ideas providing help
- ✨ propose to have a call or a pair session