


# How to Make Toast

Stacy L. Carlson  
Director, Affiliate & Influencer  
Marketing, Automattic





I love  
**systems and**  
processes



When I came across a TED Talk by **Tom Wujec** on making toast, and his invitation to take the idea further, I knew I had to share it.

*So I started giving this presentation. Over and over.*

# We Hoard Our Knowledge

Most of us carry critical knowledge in our heads — processes, decisions, context — and never write it down.

Not because we don't want to share it.  
Because we're too busy to stop and map it out.

*The result: systems that live and die with one person. Teams that can't scale. Work that has to be explained from scratch, every time.*



# The Worst Time to Build a System

*Most teams only think about processes when something forces them to.*

*That's exactly when you have the least time and focus to do it right.*

**01**

**A new hire who needs onboarding.**

**02**

**A crisis that exposes the gaps.**

**03**

**A workload too heavy to carry alone.**

*The problem: that's exactly when you have the least time and focus to do it right.*

# Sound Familiar?

**One person holding all the knowledge. No time to document it.**

**No capacity to train someone new. The work keeps growing.**

**The systems to support it never get built. And the person who holds it all can't let go.**

*These are the kinds of deep, tangled problems that resist simple solutions, with many causes, many stakeholders, and no single right answer.*

**This is what we call a Wicked Problem**

# What is a wicked problem?

Wicked problems are messy, complex, and stubbornly resistant to simple solutions. They have many causes, many stakeholders, and no single right answer.

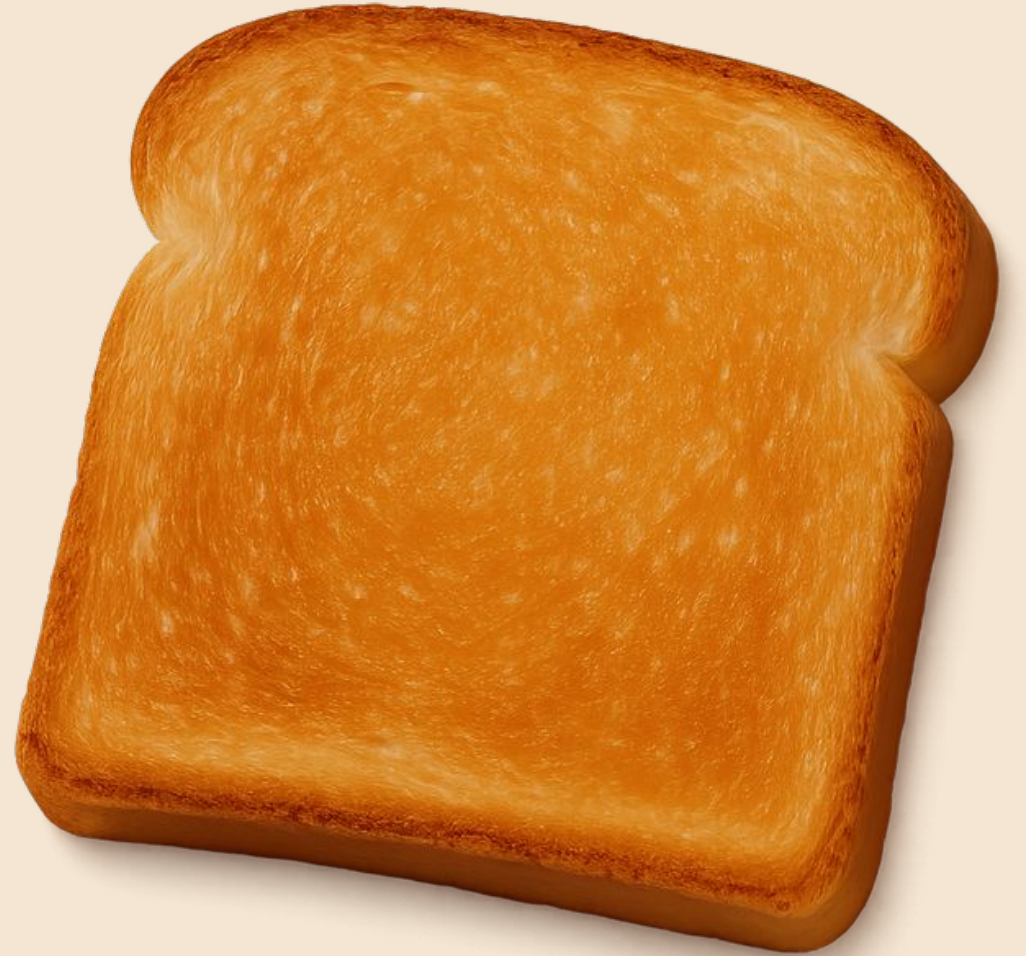
**Client scope creep**

**Developer burnout**

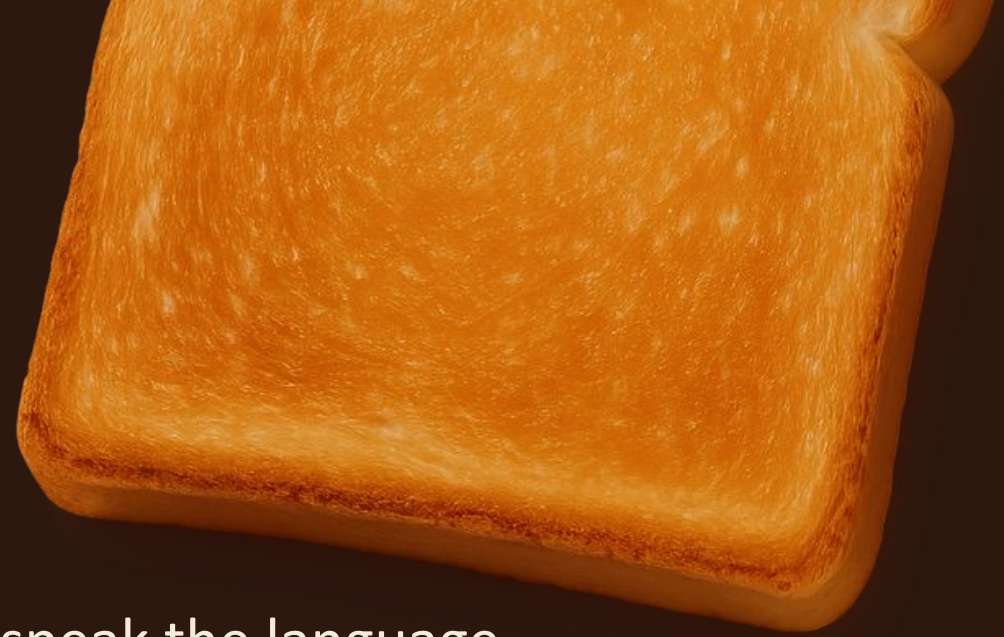
**Inconsistent code standards**

**Retaining clients long-term**

**Ok, and what  
does that have  
to do with toast?**



# You're starving. You want toast.



You're in a restaurant in a country where you don't speak the language.  
No phone. No translator.

*You are CRAVING golden brown toast.*

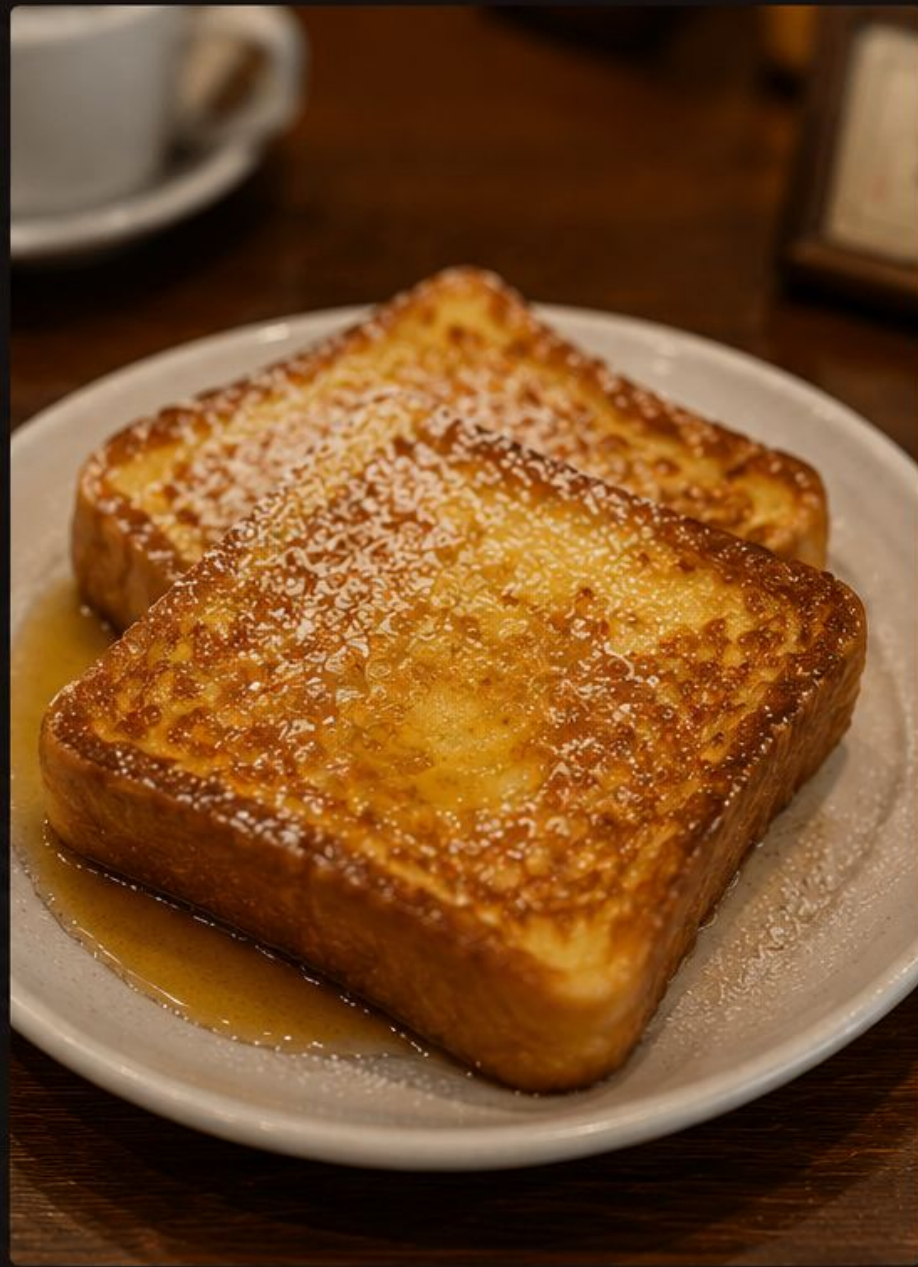
How do you tell the waiter exactly what you want (Without a translator!?)



?

食事

料理





# Draw a picture of how to make toast.

*Three minutes. No instructions. No words.*



## 1 Draw alone

Take a piece of paper.  
No talking. You have 3  
minutes.

## 2 I'm not artistic

Don't worry about how  
it looks. You won't be  
graded on it.

## 3 Keep it simple

One toaster, one piece  
of bread. No butter, no  
jelly. Just show how to  
make toast.



# Who wants to share?

Walk us through your drawing.

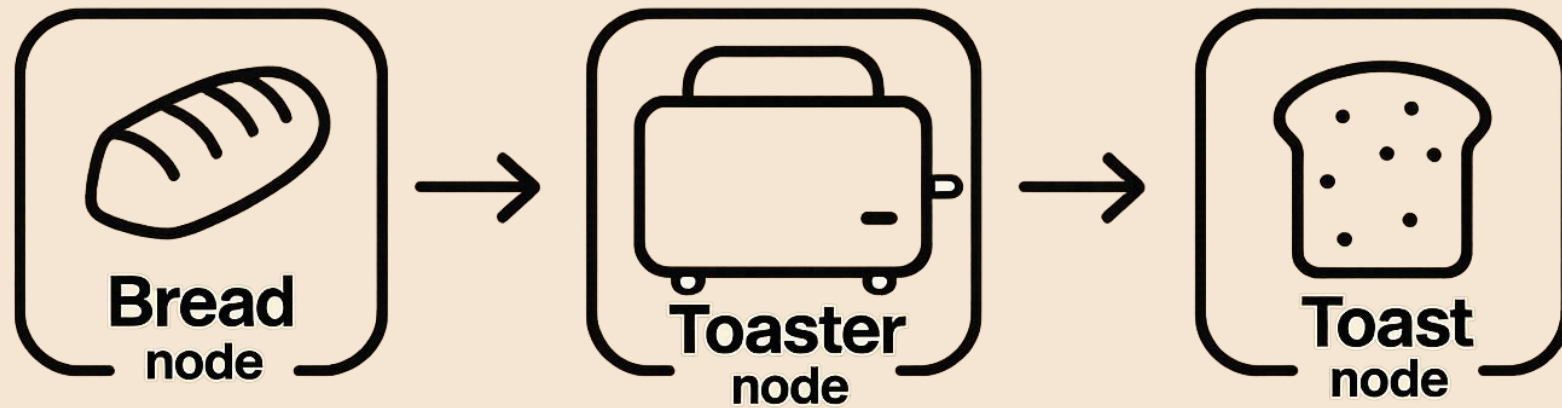
*What did you include? Where did you start? What did you leave out?*



There are  
**24 steps**  
to making toast by the  
way...

# Nodes and links

*Every system breaks down into the same two things.*



## Nodes

The things in the system — bread, toaster, person, plug, plate.

## Links

The connections between them — the dashed lines. Relationships, flows, sequences.

# Now do it **together**.

Look at the person to your right, or left, or behind you. Find a partner (or two).

## **New Problem: How to open a letter**

Tear up the piece of paper (or two) so you have individual pieces of paper to write out each step. **Rearrange each step until it feels right.** Feel free to talk, and let the pieces of paper speak more than you!






# Who wants to share?

Walk us through your map.

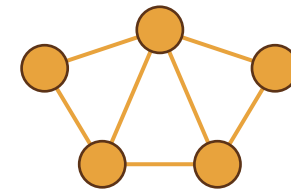
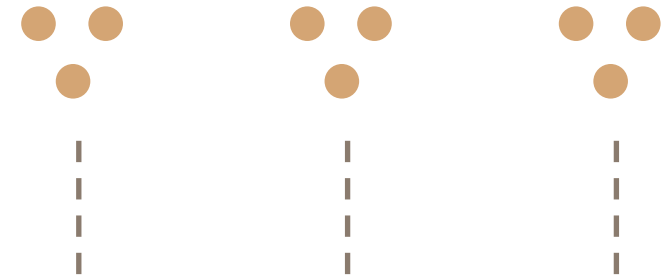
*How did you open your letter??*

# Build it together

When a group merges its individual node models, something striking happens. The diversity of viewpoints — the marketer, the engineer, the data team — doesn't cancel out. It compounds.

-  Each person sees a different angle of the system.
-  Each round of building reveals what was missing.
-  The final model is richer than anyone's draft.

From individual to collective



***One shared model.***

# What toast teaches us about thinking.



## **Make it visible**

Drawing forces the implicit to become explicit. You can't argue clearly about a system you can't see.



## **Make it iterative**

Models on sticky notes get better. Models in heads stay stuck. Movability beats memory.



## **Make it collective**

No single perspective captures a wicked problem. Only the merged map gets close.



**How to  
Get Started  
after you leave  
this room**

# One process. Not ten.

- 👁️ Pick the thing your team does most often, or the thing that breaks most often.
- 🔄 Document that first.
- 💡 A single clear process beats a half-finished library every time.



# Every person who does the work writes the process.



## Time-box it

One hour per session max. Longer than that and you're writing a manual, not a process.



## Review it

A process no one checks becomes wrong fast. Build in a review date from day one.



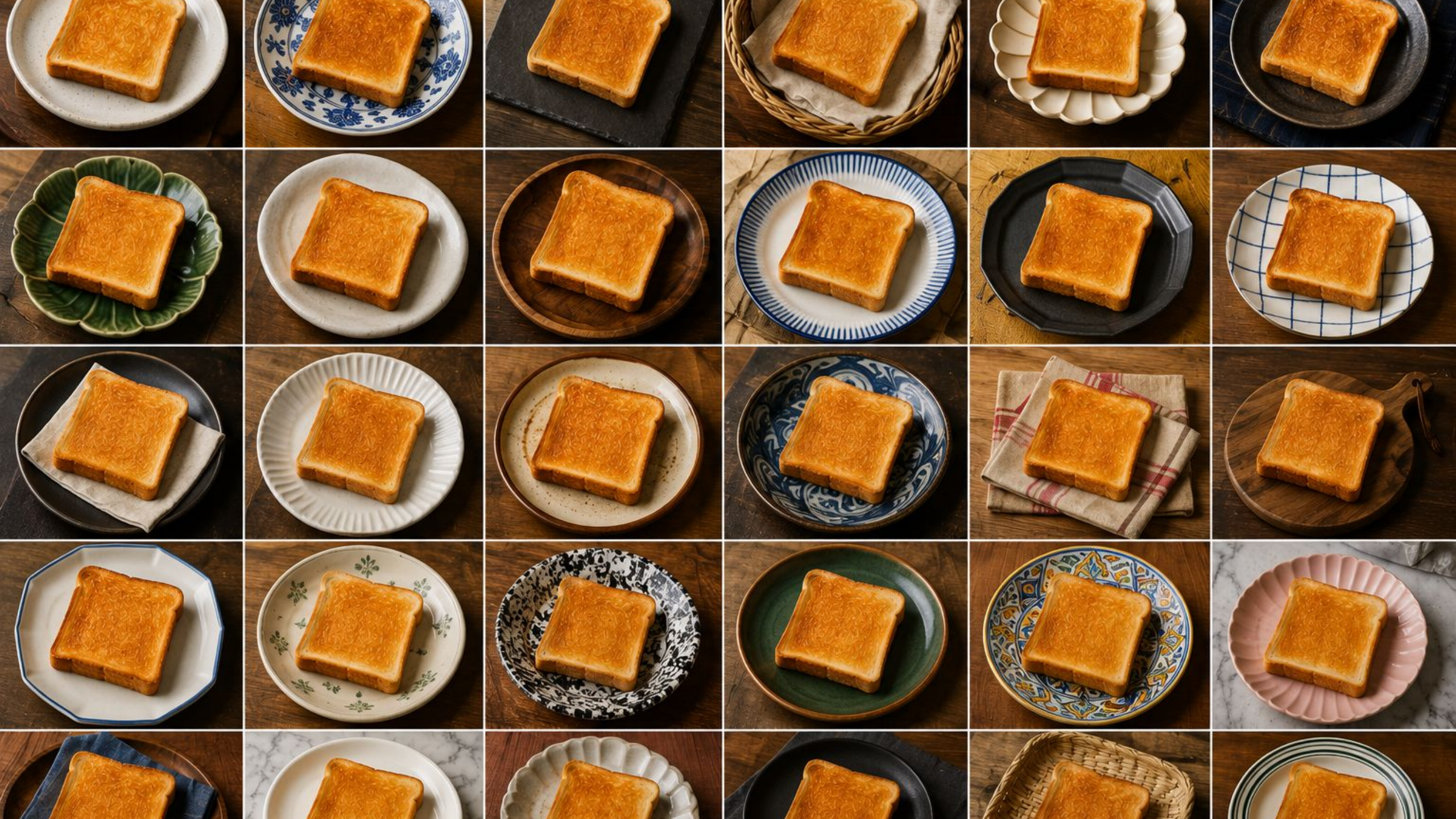
## Pair up

Have someone else follow your draft. If they get stuck, the process needs work.



# Done beats perfect.

A rough process someone actually uses is worth more than a polished one still in draft. Version 1 does not need to be beautiful. It needs to exist.



**THANK YOU!**

