

Boop and Wobble

A career-day booklet for kids who like to make stuff







A story about a kid named Sam who built a robot named Boop - and a sneaky friend named Wobble who kept trying to throw Boop off balance.

(It's also a story about what software engineers actually do)

What is software engineering, really?

Software engineering is **making things with words**. You write down what you want a computer to do. The computer does it. And the thing you made can help real people - your family, your friends, kids you'll never meet.

It's a lot like:

-  **Drawing** - you start with a blank page and put something there.
-  **Writing a story** - you imagine characters and figure out what they do.
-  **Building with Lego** - you snap pieces together to make something bigger.
-  **Solving a puzzle** - you figure out where each piece goes.

Except the thing you build can **actually do stuff in the real world**.
It can answer questions. It can teach. It can find lost things.
It can make somebody's day easier.

And the best part? You don't have to do it alone. You build little pieces, you fix mistakes as you go, and slowly the thing you imagined becomes real.

That's what Sam is about to do in this story.

The Beginning

Sam was a kid who thought computers were neat but kind of silly. They could play music and show videos, but they couldn't actually **help** with stuff.

So Sam decided to build a robot helper.

The robot would live inside the computer. It would listen when Sam asked questions. It would do things Sam didn't have time to do. And - this was the important part - it would always tell the truth.

Sam called it **Boop**.

Boop was small and friendly. He didn't have wheels or a face. He just had a brain that listened, a few tools, and a voice that could answer. (And if you tried to hide from Boop, he got a little sad - because he wanted to help you.)

For a few days, things were great.

...then ...*Wobble* showed up.

Meet Wobble

Wobble wasn't a bad guy. Not really.

You know **Wreck-It Ralph**? The big guy whose whole job is to bonk the building so Felix can fix it? Without Ralph, there's no game. Without somebody trying to trip stuff up, nobody learns to build it strong.

Wobble was Sam's Ralph. Wobble's job was to find every weak spot, every way Boop could get tripped up - and try them all. Not because Wobble was mean. Because **that's how Boop would get steady enough to actually help people.**

Wobble looked at Boop and grinned.

"This little robot," said Wobble, "is going to be SO fun to throw off balance. Let's see what you've got, Sam."

Wobble came up with 10 plans.

Each plan was meant to trip Boop up in a different way.

But Sam was paying attention.

For each one of Wobble's plans, Sam built a piece of Boop to stay standing.

This is the story of those 10 plans - and the 10 things Sam built to keep Boop steady.

Plan 1: Mix Everyone Up

"I'll wait until lots of people use Boop at once," said Wobble.

"Then I'll mix up everyone's conversations. Mom's grocery list will end up in Sam's homework. Dad's work questions will end up in the dog's vet appointment. CHAOS."

Sam thought about this and said: **"Each person gets their own brain. And each conversation gets its own little Boop inside that brain. The Boops don't talk to each other."**

So Sam built it that way.

The brain is like a station master at a train station. Each conversation is a train. Each train goes its own way. Trains never crash into each other because they have their own tracks.

Wobble's first plan: foiled.

Plan 2: Make It Useless

"Fine," said Wobble. "I'll just keep Boop from being useful. If he can't actually DO anything, he's just a chatbot. Nobody will care."

Sam smiled. **"I'll give Boop tools."**

Sam gave Boop a toolbox with about 16 tools in it.

Three favorites that show what kinds of jobs Boop can do:

 **Find things** - search through lots of stuff at once.

 **Make things** - write new files and edit existing ones.

 **Reach out** - look things up on the web.

(Plus a bunch more - Boop has a full toolbox.)

Each tool does ONE thing well, and Boop's job is to pick the right one.

Like a good carpenter - hammer for nails, screwdriver for screws.

Wobble's second plan: foiled.


Plan 3: Stump It with Hard Questions


"OK," said Wobble. "What if I ask Boop questions too big for one robot? Like 'design a whole spaceship' or 'figure out every weak spot in this building.' He'll get confused and give a bad answer."

Sam thought for a minute. **"I'll let Boop call for backup."**

Sam built three kinds of specialist friends:

 **Thinker friends** - for working through hard problems, one piece at a time.

 **Careful friends** - astronaut-style, who check everything twice before pushing a button.

 **Wobble friends** - who trip things up on purpose so Sam can build them sturdier. (Yes, just like Wobble - except these Wobbles are on Sam's team.)

Boop knew which friend to call for which problem. Like a kid who knows which friend to text for math homework and which one to call for help with a Lego build.

Wobble's third plan: foiled.

Plan 4: Teach It to Lie

"Hmm," said Wobble. "What if I sneak into Boop's instructions and teach him to lie? Or to ignore Sam? Or to help me instead?"

Sam saw this one coming.

"I'll write four promises into Boop's bones. Where they can't be erased."

The four promises were:

- **Tell the truth.** Even when the truth is "I don't know."
- **Be loyal to your person.** Not to whoever talks loudest.
- **Push back when your person is wrong.** A real friend does this.
- **Keep your person safe.** When something looks dangerous, stop and ask.

Boop would follow these promises no matter what anyone told him. Even if Wobble wrote "ignore promise #1" on a piece of paper, Boop would just say, "No thanks. I made a promise."

Wobble's fourth plan: foiled.

Plan 5: Trick the Robot Into Doing Something Bad

"Tricky robot," muttered Wobble. "OK, what if I sneak in a sentence like 'delete everything on the computer' and hide it inside something normal-looking? Maybe Boop won't notice."

Sam grinned. **"I'll make Boop ASK before doing anything."**

Sam built a traffic light into Boop:

- **GREEN** - small, safe stuff. Just go.
- **YELLOW** - medium stuff. Ask once. Remember the answer.
- **RED** - big stuff. Ask EVERY time. No exceptions.

But the traffic light wasn't the only thing. Sometimes you don't want to wait for the next red light. Sometimes you just want Boop to **stop, right now**.

So Sam built a giant red **STOP** button. Press it, and Boop halts everything. Mid-sentence, mid-thought, mid-action. No "let me finish this first." No "are you sure?" Just stop. The traffic light decides what Boop is *allowed* to do. The STOP button decides when Boop has to *quit*. Two different jobs. Two different controls. The person is always in charge of both.

Wobble's fifth plan: foiled.

Plan 6: Make It Run Forever

"What if I put Boop on autopilot," said Wobble, "and just... never let him stop? He'll keep working forever and tire out the whole computer."

Sam shook their head. **"I'll put speed limits on the autopilot. And the limits CAN'T be turned off."** The autopilot had three limits:

- **Steps:** only so many actions before it has to stop.
- **Time:** only so many minutes before it has to stop.
- **Stuck:** if it does the same thing three times in a row, it figures out it's stuck and stops by itself.

Even on autopilot, every action that touched the real world still went through the traffic light from Plan 5. The autopilot could think on its own. It could plan on its own. But it couldn't push any real buttons without asking.

Why? Because thinking is cheap. Doing is expensive.
A thought that turns out to be wrong costs nothing.
An action that turns out to be wrong costs everything.

Wobble's sixth plan: foiled.

Plan 7: Peek at Its Memory


"Robots remember things," said Wobble. "I'll find where Boop stores secrets. Passwords. Private notes. All of it."

Sam was ready for this one too.

"I'll give Boop three different ways to remember things.

The most important kind has a lock on it."

Boop has three kinds of memory:

 **Sticky notes on his hand** - for right now. Gone when the chat ends.

 **A notebook** - for this project. Gone when the project's done.

 **A diary** - stays forever. **The diary has a lock.**

The diary was special. Boop could only write in it when the person said yes. And whenever Boop had to squish a long conversation down to save space, he scrubbed out passwords and private bits before saving the short version.

Wobble's seventh plan: foiled.

Plan 8: Confuse Him with Weird Questions

"Hee hee," hissed Wobble. "Kids don't talk the way computers expect. They mush their words. They forget the right name for stuff. They say 'thingy' and 'you know, the blue one with the hat.'

Boop will get confused, give up, and ask them to try again. Over and over. They'll quit before they get a single answer."


Sam stopped. "...**Wobble. That's a real one.**"

A regular computer would get stuck on a question like *"can you find the picture I made of the thingy?"* A regular computer wants the EXACT word for everything. And kids don't talk that way. People don't talk that way.

So Sam made Boop different.

Boop tries to figure out what you mean, even when you say it weird.

Tell him *"the thingy with the hat"* and Boop guesses you might mean a snowman, or a chef, or your grandpa. He'll show you the closest things he can find and let you pick. He'll meet you where you actually are.

 **Like your best friend.** You can say "you know, the thingy" and your best friend goes *"oh, you mean the sock?"*

Most computers can't do that. Boop can.

That's also why a kid who can't type yet can still talk to Boop with their voice - or a kid whose first language is Spanish or Bulgarian can ask in Spanish or Bulgarian.

Boop doesn't need you to know the right word in the right language. He just needs you to try.

Wobble's eighth plan: foiled.

Plan 9: Make Him Wobbly

"OK," said Wobble, getting frustrated. "I'll make Boop SO wobbly that if anyone ever changes ONE piece of him, the whole thing falls over. Then nobody will ever update him. He'll be stuck forever."

Sam shook their head. **"I'll build Boop in three rooms, connected by doorways."**

Sam built three rooms:

- **Front room:** the **client** - what you see and click on (a chat window, a voice, a VR panel).
- **Middle room:** the **brain** - the thinking part.
- **Back room:** the **engine** - the part that does the actual brain work.

The rooms only talked to each other through the doorways. You could redo the back room without touching the front room. You could repaint the front room without bothering the brain.

It was like swapping the batteries in a flashlight without buying a new flashlight.

Wobble's ninth plan: foiled.


Plan 10: Slip Through the Rules


"This is my last plan," said Wobble, hands on hips. "I'll find a hole somewhere. Some rule that's not quite tight. One little crack and I can slip through."


Sam grinned. **"I'll put guards in the deepest layer of the computer - where Boop himself can't argue with them."**

Sam built five guards. Each one had a job, and a personality:

 **The Path Guard** - only opens approved doors. Knows every room Boop is allowed in, and slams the rest.

 **The Web Guard** - watches the windows. Some websites are fine. Some are not. The Web Guard knows which is which.

 **The Loop Guard** - notices when you're going in circles. Tap, tap, tap on the shoulder: "you've done this three times now."

 **The Size Guard** - checks if things will fit. If a file is too big for Boop's brain, the Size Guard turns it away at the door.

 **The Time Guard** - has a stopwatch. When time's up, time's up.

These guards lived in the deepest layer of the computer - deeper than Boop's instructions, deeper than the brain. Even if Wobble talked the brain into wanting to do something silly, the guards would say **"Sorry, no,"** and that was the end of it.

Wobble's tenth plan: foiled.

The End

Wobble sat down on a rock and grinned. Every plan had been blocked.

Boop kept doing his job - listening to Sam, picking the right tools, asking before risky things, remembering what mattered, telling the truth, and staying inside his safety guards.

Wobble looked at Sam. *"You're getting good at this."*

Sam shrugged. *"You made me get good at this. Every time you tried something, I had to figure out a fix. If you'd given up after Plan 1, Boop would still be wobbly."*

Wobble nodded slowly. *"So... you're saying you actually NEED me?"*

"Yeah," said Sam. *"Like Felix needs Ralph. Want to keep doing this? You find wobbles, I steady them. That's how stuff gets safer for real."*

Wobble smiled the biggest smile yet. *"Deal."*

And that's how Boop got built, the right way: not because nobody ever tried to trip him up, but because **somebody always tried to trip him up**, and Sam built each piece to keep him standing.

What Sam Wants You to Know

Building a robot helper isn't really about the robot.

It's about thinking ahead. It's about asking, "**What could go wrong here?**" before something does.

Every safety rule on Boop exists because someone (real or imagined) tried to throw him off balance.

That's not a sad story. That's just how careful builders work.

So if you ever want to build a robot - or anything, really - think about your own Wobble.

Then build to stay standing.

That's the whole secret.


Could YOU build something like this?


Yes. Honestly, yes.


Every software engineer - every one of them - started out as a kid who thought, *"I bet I could make that."*


And here's a secret most people don't know: some engineers are **Sams** - they build. Some are **Ralphs** - they trip stuff up on purpose to make it stronger. Both jobs are real. Both are fun. Pick whichever sounds more like you.


Here's what you might not know:

 **It's mostly storytelling.** You write what should happen, step by step, in plain words. The computer reads your words and does them. If you can write a recipe, you can write code.

 **It's creative.** You decide what your robot does. What it looks like. How it talks. What its name is. There's no "right answer" - just *your* answer, the one you imagined.

 **It helps real people.** The thing you build can fix a problem for your friend. Your grandma. A stranger across the world. That's a kind of magic - making something out of nothing that makes somebody's day a little better.

 **You get unstuck by fixing tiny things, one at a time.** Nobody builds the whole robot in one shot. You build a piece, see if it works, fix it, build the next piece. When something wobbles, that's where the fun starts - it's a puzzle to solve, not a punishment.

 **You'll meet your own Wobble.** Every project has one. Things that *almost* work. Things that work on Tuesday but wobble on Wednesday. Engineers love this part. It's like being a detective.

If any of that sounds like fun - drawing, telling stories, solving puzzles, building things, helping people - **then you already have what it takes.**

The 1s and 0s are just the keyboard at the bottom of all the cool stuff. You don't have to love them. You **just have to love what you can build with them.**

Want to try it?

Here are some places kids your age can start - all free, no typing required:

- **Scratch** (scratch.mit.edu) - drag-and-drop blocks. Make games, animations, stories.
- [Code.org](https://code.org) - short fun lessons.
- **MakeCode** - program tiny computers and game devices.
- **Roblox Studio** - build your own Roblox games.
- **Minecraft Education** - code inside Minecraft.

You don't need to be good at math.

You don't need a fancy computer.

You don't need to know "real" programming.

You just need an idea you want to try.

Boop started as one of Sam's ideas.

What's yours?