Question	Number
Am I the right person for the task, or is this better handled by someone else?	1.01
How might slowing down actually help us move faster?	1.02
Is this moving us towards the outcome we want? What outcome do we want, anyway?	1.03
What would happen if we simply didn’t do this task?	1.04
What’s the worst that can happen, and how would I handle it?	1.05
Are we all on the same page? Do people really know what I think they know?	1.06
What’s the next step? Is there an even smaller next step?	1.07
What should I be saying ”no” to in order to say yes to something better?	1.08
Is someone not “in the room” who should be? Are there too many people in the room?	1.09
Are there rooms you should be in that you’re not? Who could you ask about them?	1.10
Have you been clear about what you need? Are you sure?	1.11
Is something you’re doing better handled by someone more junior who could use a challenge?	1.12
What are you the go-to person for? How could you get someone else up to speed?	1.13
Are you putting off important more work by doing easier, less important work? If yes, why?	1.14
What work are you dreading? Why? Is this trying to tell you something?	1.15
Do you have your hands in the code enough? Too much? What would help?	1.16
Does your management know your accomplishments? That's not bragging, it's just stating facts.	1.17
What parts of your job do you enjoy? How can you do more of that?	1.18
Whose perspective would be useful? How can you seek that out?	1.19
Firefighting is sometimes necessary, but fire prevention is better. What does "fire prevention" look like for your work?	1.20
Are you “behind”at work? Or are theexpectations notrealistic? Either way,can you set newexpectations?	1.21

Question	Number
What do you mostwant to learn?Is there someonewho knows thatwho might be a good mentor?	1.22
Which colleaguesare ready to shinebut don’t know it?How can youencourage themor champion them?	1.23
What stories wouldyou tell at a jobinterview tohighlight your skillsand experience?	1.24
What energizes youabout writing code?What’s tedious?Can AI help?	1.25
AI can be both exciting and scary. What scares you, and what might make it safer and less scary?	1.26
Non-developersusing AI to code:Amazing? Scary?How mightdevelopers help them do it well?	1.27
If AI does morecoding, how cannew developerscontinue to learn?	1.28
Are you using AIfor tech work otherthan coding, e.g.,infrastructure,deployment, architecture?	1.29
Can experienceddevelopers modelthat it’s okay to notknow something andto ask? What elsemight make it safeto ask questions?	1.30
Do you keep a workjournal? Learnings,accomplishments,ideas, questions, challenges, stories?	1.31
The work is confusing,messy, and ambiguous.That’s why it needs you! What hard thingshave you tackled? (Not just at work!)	1.32
What does successlook like accordingto your boss?Your VP? Yourcustomer? You?	1.33
What is your idealmanager like?Think of your bestmanager(s), whatwere they like, howdid they work?	1.34
What does your team measure to see how things are going? Do these metrics encourage good practices?	1.35
What’s yourleadership style?Do you see otherleaders at yourorganization with a similar style?	1.36

Question	Number
Would you ratherbe in your role inanother companyor a different rolein your current company? Why?	1.37
Do you prefer to bean individualcontributor, amanager, or switchbetween them?What appeals to you(or not) about each?	1.38
Information flow inyour organization:Formal or informal?One to one, or oneto many? Meetings,slide decks, privateconversations?	1.39
Are you a changeadvocate? Sick of change? Both? Doyou know and talkabout the benefitsof any changescoming up?	1.40
What’s your idealrole? Ignore “theywon’t pay me forthat,” “there’s nosuch job,” “I don’ thave enough experience,” etc.	1.41
What triggers yourstress? How do youshow up when youare stressed? Whathelps you managestress?	1.42
Are you toleratingsomeone’s badbehavior? Can youget guidance orhelp from a trusted advisor?	1.43
What’s easy for usisn’t necessarilyeasy for everyone.What skills comeeasily to you? Do youundervalue them?	1.44
Are there skills orknowledge you writeoff as not for you?(I’m bad at x, I couldnever y.) Is that true?What would it meanif it wasn’t true?	1.45
Is peer review ofcode necessary?Always? Is it possibleto eliminate peerreview entirely?What would haveto be true?	1.46
Is your manageraware of all of thework that you do?Or do you do somework that goesunrecognized?	1.47
Do you dismiss “softskills” as unnecessaryfor developers? Dothe people you work with feel that way?	1.48
When did you (ormight you) knowthat you’re not“junior” anymore?	1.49
Do you take notes as you learn? Do you share your notes withothers? If not, what stops you fromwriting or sharing?	1.50
What parts of your job do you dread? How can you do less of that?	2.01
Do developers at your organization have contact with actual users?	2.02
What's the best team you've been on? Best how? Why was it like that?	2.03
Have you tried pair programming? Mob programming? What did you think?	2.04

Question	Number
Have you tried test driven development (TDD)? What did you think?	2.05
Do you take a midday break at work? Do you actually get away from the computer?	2.06
How can tech workers stay physically active during the workday?	2.07
Do you keep a praise file? (Saving a copy when people say good things about you or your work.)	2.08
Are there skills you're great at but dislike? (Just because you're skilled doesn't mean it's the job for you.)	2.09
Agile vs. Waterfall: is there a time and place for each? Or are you fed up with one?	2.10
The boss keeps piling the work on. It's too much. What do you do?	2.11
Will the thing you're working on ever be "done"? Or will it need work indefinitely?	2.12
You're presenting at a big meeting in a month. How do you prepare?	2.13
How do you keep track of what you need to do and what is most urgent? To do list? An app? Tickets? Just in your head?	2.14
What helps you troubleshoot when handling a production outage? What helps you stay calm?	2.15
Does your organization treat long hours as a sign of devotion (even if they tell people not to work extra)? Do you feel pressure to work more hours?	2.16
Do you feel like you aren't productive enough, even if the boss says you're doing fine?	2.17
Think of times when you were very busy. When was it draining, when was it energizing?	2.18
How do you know if you are burned out? What signs do you look for? What do you do about it?	2.19
How do you meet other developers at your level?	2.20
What do you feel like you should know, but struggle with? (Mine include CORS, character encoding, and REST)	2.21
If someone says "Kubernetes," your first thought is...?	2.22
What's the opposite of "what you do all day"? (Sitting -> running, inside -> outside...)	2.23
What's your favorite use of AI so far for development? For other work? In general?	2.24
What wild stories have you heard about legacy code or systems?	2.25

Question	Number
What does "developer experience" mean to you? to your boss?	2.26
Have you read any books that changed how you think about being a developer?	2.27
Do you have a favorite way to keep up on new industry trends?	2.28
Are there any tech blogs or podcasts you like, or interesting tech folks to follow on social media?	2.29
How do you think AI might change organizations? Team size? IT and business integration? New roles?	2.30
Who are or were your tech role models? What did you learn from them?	2.31
In the future, will most developers not look at code, just like how most of us don't look at 0s and 1s now?	2.32
What would you create if you could work 10x or more faster? For work or otherwise.	2.33
What do you think of "you build it, you run it" where the developers do production support? Are you in that world now?	2.34
What influence do staff+ engineers have at your organization? Are they truly peers of management? (Come talk to me.)	2.35
You're an IC, fired up about agile, devops, etc. Your boss isn't on board. What can you do?	2.36
What are some tools you could not imagine doing your job without?	2.37
Your build spits out lots of insignificant warnings and errors. Do you ignore them or try to clean them up?	2.38
"I can't get any work done because I'm in meetings." Which meetings ARE your work? Can you stop attending ones that aren't?	2.39
How do you know you're "productive" or not? What is a productive day like?	2.40
Have you ever seen bad measures of productivity (e.g. lines of code)? What made them bad?	2.41
Do you have a personal mission at work? How about a secret personal goal?	2.42
Think of tough decisions you've made at work. What values have guided those decisions?	2.43
What's your ideal workspace? Does it depend on what you're working on?	2.44

Question	Number
Do you block focus time in your calendar? Do you block lunch? Do people respect your blocked time?	2.45
Think of your favorite developer colleagues. What makes them great to work with?	2.46
How do you want to be remembered by your colleagues after you leave? How is that going?	2.47
Do you solicit feedback on your performance other than from your boss? Do you get honest feedback?	2.48
Have you worked on a team that measures team happiness? How was it measured?	2.49
The product you've been working on has been abruptly canceled. How do you react?	2.50