Table 1 Attributes of great software engineers, along a scale of *must have*, *should have*, *nice to have*, *irrelevant*, and *should not have*

# Higher	Mode	Attribute and description
53	Must	Pays attention to coding details, such as error handling, memory, performance, style
52	Must	Mentally capable of handling complexity; can comprehend multiple interacting software components
49	Must	Continuously improving: improves themselves, their product, or their surroundings
49	Must	Honest: provide credible information and feedback that others can act on
49	Must	Open-minded: lets new information change their thinking
46	Must	Executes: knows when to stop thinking and to start doing
45	Must	Self-reliant: gets things done independently and does not get blocked easily
45	Must	Self-reflecting: recognizes when things are going wrong with a plan and pivots
43	Must	Persevering: not dissuaded by setbacks and failures
41	Must	Fits together with other pieces around it: code accounts for surrounding constraints and products.
41	Must	Knowledgeable about their technical domain, including product, platform, and competitors
39	Must	Makes informed trade-offs: code is responsive to time to market goals, critical needs of the business
39	Must	Updates their decision making knowledge: does not let their understanding stagnate
36	Must	Curious: desires to know why things happen and how things work
36	Must	Evolving: code is structured to be effectively built, delivered, and updated incrementally.
35	Should	Knowledgeable about tools and building materials: knows strengths and weaknesses of code
35	Should	Grows their ability to make good decisions: builds understanding of possible outcomes of decisions
34	Should	Sees the forest and the trees: reasons through situations at multiple levels of abstraction
31	Should	Craftsmanship: wants their output to be a reflection of their skills and abilities
30	Should	Does due diligence beforehand: examines available information before deciding
30	Should	Elegant: designs solutions that others can understand and appreciate
29	Should	Asks for help: knows the limits of their knowledge and supplements it with knowledge of others
28	Should	Desires to turn ideas into reality: takes pleasure in building software
28	Should	Long-termed: considers costs and benefits over time, not just short-term goals
25	Should	Willing to go into the unknown: can step outside of comfort zone to explore a new area
24	Should	Is a good listener: effectively obtains, comprehends, and understands others' knowledge
22	Should	Passionate: intrinsically interested in the area they are working in
22	Should	Manages expectations: clearly communicates what they are going to do and by when
22	Should	Focused: prioritizes time for the most impactful work
21	Should	Systematic: address problems in an organized, principled manner
21	Should	Adapts to new settings: continues to be valuable to the organization as environment changes
19	Should	Integrates understandings of others: can build a more complete understanding with others

Table 1 (continued)

# Higher	Mode	Attribute and description
19	Should	Does not make it personal: avoids deciding based on individual goals and feelings
19	Should	Creative: generates novel and innovative solutions based on the context and its limitations
18	Should	Walks-the-walk: acts as an exemplar for others to follow
18	Should	Knowledgeable about software engineering processes: knows the bests practices and techniques
13	Should	Anticipates needs: proactively determines potential problems and needs
13	Should	Uses the right processes during construction: uses best practices and techniques to construct software
13	Should	Resists external pressure for the good of the software product: stands firm against outside pressures
11	Should	Has a good reputation: has the belief, respect, trust, and confidence of others
11	Should	Productive: achieves the same results as others faster
10	Helps	Knowledgeable about customers and business: understands their product's value proposition
10	Helps	Creates shared understanding with others: shapes others' knowledge via effective communication
9	Helps	Creates shared success for everyone: establishes long-term goals that everyone can buy into
8	Helps	Aligned with organizational goals: takes actions for the good of the product and the organization
7	Helps	Well-mannered: treats others with respect
7	Helps	Data-driven: lets data drive actions, not solely intuition
6	Helps	Creates a safe haven for others: frees others to make decisions based on what is right, not fear
5	Helps	Mentoring: teaches, guides, and supports other developers
4	Helps	Knowledgeable about people and the organization: aware of others' responsibilities and knowledge
2	Helps	Challenges others to improve: encourages expanding capabilities and goals
2	Helps	Personable: establishes trusting, positive social relationships
1	Helps	Hardworking: is willing to work more than 8 hour days to deliver
0	Helps	Trades favors: builds personal equity with others allowing them to call upon others later

We tried to ask a single informant about multiple attributes (when the informant qualified to answer questions about multiple attributes, per the selection criteria above), in order to uncover insights that spanned multiple relationships. The email started with a salutation and a thank you for participation in the study. It followed with: "Something interesting came up when I analyzed the data, and I hope you can help me understand it better". The email then describe the attribute(s) or relationship(s), the reason for needing further understanding (e.g. is one of the most important attributes, may not be an important attribute, having X is associated with lower importance ratings for Y), the informant's rating, and then: "Why did you choose this answer? Can you help me understand your reasoning?".

We qualitatively analyzed the responses to gain understandings, selected representative quotations, and then asked the informants' permission to quote them anonymously.