Sample Competency Descriptions

Analytical Ability: Approaches problems quantitatively and displays critical thinking and problem-solving abilities. Breaks down complex problems into component parts. Defines and tracks key metrics to make data-driven decisions.

- *Unfamiliar*: Unable to break down problems or only talks through problems at a high level. Does not make data-driven decisions, or actively eschews data in decision-making.
- *Familiar*: Displays some critical thinking and problem solving abilities. May start to break a problem down in component parts, but not completely. Able to identify basic product or project metrics (e.g. pageviews, uptime, cost savings) but does not connect these to business or user value.
- *Experienced*: Breaks problems down into component parts. Displays repeated experience in qualitative and quantitative analysis. Defines product or project metrics beyond the basics and ties these metrics to decisions.
- *Master*: Brings analytical thinking to everything they do, and has a range and depth of experience doing so. Anticipates the need for metrics and analysis early in the product and project discovery and design process, and carries metrics through iterations.

Managing Without Authority: Builds influence in organizations without relying on formal titles or roles, and uses that influence to achieve project success. Gains commitments from leaders and team members without formal reporting structures in place. Shares credit, understands and empathizes with others, and recognizes how different people value different incentives.

- *Unfamiliar*: No experience building influence or managing without authority. When presented with a problem, only discusses escalating to management, or may describe broad efforts to influence others without any specifics.
- *Familiar*: Familiar working with teams and influencing decisions others make, but does not have deep understanding of how to do this and does not think critically about it. Still relies on formal authority to accomplish tasks.
- *Experienced*: Builds influence in organizations without formal titles or roles; those organizations and projects might be small in scope. Understands various motivations and tailors actions to them. Uses examples from their own experience in a way that shows critical thinking and understanding of what works and what might not.

- *Master*: Clear and repeated evidence of influencing major decisions that change the trajectory of an organization. Clearly understands the pros and cons of formal authority, and leverages experience to build consensus and motivate rather than dictate.

Stakeholder Engagement: Cultivates relationships with key internal and external stakeholders. Has superior negotiation skills that enable successful communication and cooperation across all levels of an organization, including executive leadership.

- *Unfamiliar*: Has no experience, or cannot articulate how, building alliances or managing communications with stakeholders is integral to project success.
- *Familiar*: Understands that strategic stakeholder alliances and communication is important, but cannot provide an example of these actions from their career.
- *Experienced*: Provides examples and demonstrates the ability to identify a key stakeholder and a strategy for effective engagement.
- *Master*: Provides examples and demonstrates the ability to manage multiple key stakeholders with varying levels of influence during a project or across multiple projects.

Superior EQ: Understands what motivates people, through keen observations of environment and context, and uses this knowledge to communicate and engage with partners and colleagues to identify obstacles and create appropriate mitigation strategies. Is a "fixer."

- *Unfamiliar*: Does not exhibit an ability or willingness to observe and understand organizational dynamics, people with different personality traits and communication styles, or subtext.
- *Familiar*: Self-described ability to observe and understand people and complex environments, but cannot supply an example to illustrate this skill.
- *Experienced*: Demonstrated ability and willingness to observe, understand, and adapt to disparate audiences and organizational dynamics. Provides examples of working through a communications breakdown and effectively problem-solving in the workplace.
- *Master*: Demonstrates a superior ability to analyze situational dynamics in a politically-charged environment and to leverage this observational awareness to solve complex problems. Cites multiple different types of examples of effectively problem-solving in the workplace.

Team Building: Identifies necessary roles and motivates individuals to form, storm, norm, and perform as a cohesive team. Effectively provides feedback to team members, and deals with low performers. Fosters latent qualities within team members.

- *Unfamiliar*: No direct or indirect people management experience. Looks to others to direct work, or may have been a project manager where team priorities are set externally.
- *Familiar*: Has lead small teams that have been assembled by others. Has not managed through challenging situations, and does not view team development as a core responsibility.
- *Experienced*: Has both lead and built teams. Understands team members' strengths and weaknesses and balances skills effectively. Has experience dealing with low performers or other challenging management situations. Considers team development as a core responsibility if not a lower priority.
- *Master*: Has both lead and built teams multiple times and in diverse organizations. Always considers effective organizational structures, and thinks beyond the scope of their team or teams. Views developing people as well as dealing with low performers and challenging management situations as a core responsibility, and can communicate team needs and wins both up and down their chain of command.

Technical Communication: Explains technical concepts to both technical and non-technical audiences. Able to frame technical choices to decision makers and justify engineering needs against business priorities.

- *Unfamiliar*: Unable to explain technical concepts at any level. Refuses to try, or gives a misleading or inaccurate description that could lead others to draw dangerous conclusions. Defers entirely to engineers.
- *Familiar*: Can explain some technical concepts to a general audience, but struggles with complex issues, either because they do not grasp the concepts well enough, or know the topic too well and cannot explain clearly to a general audience. May start to answer but then defer to engineers.
- *Experienced*: Explains technical concepts effectively, but may not make these explanations relevant to the intended audience by framing technical choices and engineering needs in the context of broader business priorities.
- *Master*: Leaves audiences of all types feeling like they better understand issues and how they impact their needs. Brings clarity, not confusion, when justifying engineering needs against broader business priorities.

Systems and Networking: *This is knowledge about computer systems, such as: Typical hardware components, OS internals, systems administration, configuration, and the use of common tools to explore and configure the workings of a production system. Includes networking and an understanding of how systems communicate over IP, TCP, HTTP, and the function of firewalls and load balancers. Includes cloud concepts like containers, Docker, VPC, etc.*

- *Unfamiliar*: Can use a computer. Can install and upgrade software.
- *Familiar*: Understands how to use tools like `top`, `uptime`, `kill` to navigate a system. Understands the risks of `kill -9`versus `kill`. Understands how processes work, What the role of `init` is. Can deploy 'hello, world!' to the cloud. Understands the relative costs of context switching, processor cache writes, RAM writes, flash writes, disk writes.
- *Experienced*: Understands how to use tools like `ldd`, `fdisk`, `mdadm`, LVM, etc. Can make use of VMs. Has detailed understanding of system processes, including `fork`/`exec`, signal handling. Understands how virtual memory works and paging. Can describe what's in /proc.
- *Master*: Can effectively manage fleets of VMs. Understands how to perform tasks at scale (such as massive machine/VM upgrades). Can name some sysctl settings and when to change them. Can discuss containers and resource limits. Threads versus processes versus coroutines.

Troubleshooting: *This is the ability to effectively troubleshoot problems. It requires scientific thinking and a methodical approach to identifying problems by developing hypotheses, testing those hypotheses, and narrowing down the root cause of an issue. This includes an understanding of monitoring and logging systems, and troubleshooting commands like `tcpdump`, `strace`, `netstat`, etc.*

- *Unfamiliar*: Knows standard log file locations. Can use `ps` or `top` to find busy processes. Basic idea of what request latency is. Knows that more errors is worse than fewer errors.
- *Familiar*: Can explain every field in an HTTP access log. Understands what typical request latencies look like. Understands what load average means. Knows what 'Connection refused' means. Can root cause a problem involving a system out of RAM and swapping heavily. Can set up basic monitoring tools to capture and plot basic pre-packaged metrics. Can use tools like `tcpdump` to identify basic connectivity problems.
- *Experienced*: Can articulate when you'd care about averages and when you'd care about 95th percentiles in request latencies. Knows what 'Address already in use' means and how to troubleshoot it. Can identify monitoring needs and design monitoring probes to capture that information. Can set up complex monitoring infrastructure. Understands scalability bottlenecks with thread and process pools. Can use tools like `strace` to understand application behavior.
- *Master*: Can troubleshoot complex problems, such as storage system failures resulting in thread pool exhaustion manifesting as web page timeouts. Understands histograms. Can identify useful application metrics and knows how to instrument code so as to feed into a monitoring system. Can use tools like `tcpdump` or Wireshark to identify unidirectional packet loss/retransmit errors.

Incident and Change Management: *This is the ability to effectively manage incidents and risk. It involves understanding roles during an incident and how to lead large-scale incident resolution by spawning and coordinating investigation teams. It also requires the ability to effectively communicate with non-technical stakeholders, including the writing of good post-mortems. It also includes an understanding of risk and how change management systems can be effectively used to mitigate/balance risk against agile development.*

- *Unfamiliar*: Can respond to pages. Can file change request tickets. Knows how to push to a `prod` branch.
- *Familiar*: Knows when to escalate issues to leadership and can effectively communicate with non-technical people. Can work within a focused team to resolve a problem. Is able to articulate risk in terms of changes made to an application and release timing.
- *Experienced*: Acts as a bridge between technical teams resolving a problem and leadership needing an assessment of the problem and resolution activities. Can lead a troubleshooting team to resolve an outage. Can clearly gauge risk of remediation activities and act decisively. Can identify risky changes and can exercise good judgment deferring a high-risk change.
- *Master*: Can write incident coordination and communication plans. Can weigh user impact and establish severity of a problem and adapt response accordingly. Can coordinate between multiple investigation teams working different parts of the same or related problems. Can effectively keep resolution activities focused on resolution and not distracted by tasks like communication. Can identify and empower individuals that are strong contributors to an incident and politely dismiss those that are not.

Product Delivery: Regardless of environmental challenges, ships products and drives teams to deliver on key milestones. Gets Minimum Viable Products out the door without waiting for the perfect technical implementation or inclusion of every feature.

- *Unfamiliar*: Has never been part of a product team that shipped a software product or product update.
- *Familiar*: Has been part of teams that have shipped software products, but has never lead any teams. May have some experience as a data analyst in service or product delivery, or as an Agile Product Owner with limited or no experience scoping and making tradeoffs to hit deadlines.
- *Experienced*: Has a demonstrated track record of shipping software products and scoping to a Minimum Viable Product. Products may be on a small scale to only a few thousand users, but must have been broadly available. Has experience scoping and making tradeoffs to hit deadlines, and also working closely with engineers and designers in an iterative environment.
- *Master*: Has repeated experience shipping software products to hundreds of thousands of users. Has experience in all parts of a product lifecycle from initial launch to optimization to sun-setting. Effectively makes prioritization

decisions and manages to deadlines, working closely with engineers and designers in an iterative environment.

Engineering Considerations: Works effectively with engineers as a true partner. Understands the technical stack of a product and how it can impact product design and project schedules. Considers the complexities involved in building technology at massive scale.

- *Unfamiliar*: Has never worked with engineers, or has but not effectively/views them as a different group of people. Views technical matters as someone else's problem, and is unwilling to learn more about a project's technical foundations. Unable to describe a project's technical stack.
- *Familiar*: Has some experience working effectively with engineers. Displays a basic understanding that technical choices impact product design and project schedules. Shows an active interest in learning more about a project's technical foundations.
- *Experienced*: Repeated track record of working as a true partner with engineers. Describes situations where they have engaged in technical decisions or shifted product or project plans due to technical issues. Understanding some of the complexities of building technology at massive scale.
- *Master*: Functions as a true partner with engineers in all technical products and projects in which they are involved. Able to identify potential technical concerns with proposals before consulting engineers. Repeatedly engages in technical decisions and supports engineering needs.

Product Design: Designs and launches software products and features focused on user needs, scoped to Minimum Viable Products, and with clear criteria for success. Understands and effectively critiques poor software product design.

- *Unfamiliar*: Unable to articulate what makes a software product well designed, or fails to appreciate user needs.
- *Familiar*: Identifies and explains good product design in the context of user needs, and is familiar with the product design process but may not execute all phases. May say "I would talk to users to identify requirements" and then lean on these statements to avoid making any design decisions. Does not naturally scope and prioritize to Minimum Viable Product, or use data to evaluate decisions.
- *Experienced*: Identifies user needs in coordination with designers and researchers, and quickly connects them to product requirements. Defines priorities and makes tradeoffs in the context of user needs, but may do so without full understanding of broader business goals.

- *Master*: Repeated examples of connecting product design to user needs, data-driven decisions, broader business goals, and technical requirements.