AMAZON
MANAGER, SOFTWARE DEVELOPMENT (SDM)
ROLE GUIDELINE

Guideline Last Updated: June 13, 2019 (Version 1.2)
Contains Expectations for: Managers
Guideline Notes: For more details on Tech Management, see the Insider’s Guide to Technical Management. If you have questions about this guideline, contact sdm-level-clarification@amazon.com

The most recent revision clarifies the SDM Level Matrix and fixed some problems with the narrative content.

Amazon Manager, Software Development (SDM) Role Guideline

This guideline contains general expectations for SDMs. However, given the wide variety of businesses and technologies at Amazon, it cannot capture all expectations. No two Amazon teams are alike and each is encouraged to develop their own approach to delighting their customers. This alters the mission of a team, the way a manager needs to operate and what constitutes as success.

Each section of this guideline has a specific purpose:

- **Section 1: "SDM Level Matrix"** is a high-level view of how SDM capabilities map to levels. It is useful for quick comparisons of level vs. role expectations in debriefs, performance, and promotion discussions.
- **Sections 2-5:** "**What you do…**" contain more detailed expectations for each level to guide hiring strategies and performance discussions. Each section contains a graphic to illustrate the scope for that level.
- **Sections 2-5 "Moving to…"** criteria isolate key skills at the next level that may be demonstrated to be considered for promotion. This is not a checklist. Every promotion case is unique; the results delivered (*and how they are delivered)* also play a role in promotion evaluations.

This guideline does not repeat expectations documented in previous levels (i.e., the abilities of higher levels inherit those of lower levels). This means a Sr. Manager has all of the abilities described in Levels 4-7.

1. SDM Level Matrix

The purpose of this matrix is to provide a high-level overview of how functional dimensions change by SDM level. It does not include our Leadership Principles, as they apply to all Amazonians at all levels. For more detailed guidance, see **Sections 2-5**.

1.1 SDM Functional Dimensions

- **Ambiguity:** The degree that requirements, opportunities, and strategies are defined. Typically related to the construction and maintenance of technology **product**[1] (**features**[2] and **components**[3]) and/or **architecture**[4]. Includes level of independence.
- **Scope and Influence:** Area of ownership (**team**[5], multiple teams) and influence. Also referred to as span and layers.
- **Execution:** Concise view of expectations, including which levels are **tactical**[6] and **strategic**[7]. Includes how a management level mitigates risk, escalates (vs. handles escalations).
- **Communication**: All managers are expected to provide clear, concise, accurate, and timely communication (verbal and written) to the right audiences. This dimension clarifies how each level differs in the types of communication that are typically expected from a management level.
- **Impact**: System size and impact on **organization**[8] goals.
- **Technical Complexity**: Degree of complexity involved in the work their team is asked to perform that they need to manage. Includes a type of trade-off that a level may make.
- **Process Improvement**: How each level drives engineering improvements (i.e., customer interaction, development methodology, working with other teams, operational excellence, etc.).
- **Experience:** Suggested experience to meet level expectations. Education is **not required** if the candidate has equivalent knowledge gained from experience**.**

[1] **Product** is any good, application, or service that Amazon sells to customers or offers internally for use by the company or its employees.

[2] **Feature** is any functionality or cohesive set of functions that has value to a customer.

[3] A **component** is a building block that does not solve a customer or business problem on its own, but is part of a solution.

[4] **Architecture** is how we put components and features together to delight customers or solve a problem. An architecture can range from the internal structure of a single service, application, or web page to interdependent technologies that combine with hardware to make a device product.

[5] **Team** is a group formed to solve a problem or provide a solution. For consistency, team is defined as < 10 people ("two-pizza team").

[6] **Tactical** is when someone takes action(s) to achieve specific goals. Actions can be implemented as one or more specific tasks or projects. Includes setting priorities and success measurements, making *short-term* trade-offs and either implementing or engaging with others to deliver.

[7] **Strategic** is a leadership skill. Includes defining a mission, vision, tenets, and long-term priorities. Includes ensuring that stakeholders, leaders, and others are working to\ward common goals. Requires judgement and experience to make appropriate *short-term* vs. *long-term* trade-offs.

[8] **Organization** (org) is a group of teams that together provide one or more product(s) or function(s). Orgs vary in size, with *small orgs* typically led by a Director and *large orgs* typically led by a VP.

1.2 SDM Level Matrix

Dimension	L5: SDM II	L6: SDM III	L7: Senior SDM
Ambiguity	Capable of managing a software development team with existing products and defined strategy. May seek direction.	Capable of managing a software development team with a business problem or strategy that may not yet be defined. Leads with limited guidance.	Capable of managing multiple software development teams where the overall business opportunity or strategy may not yet be defined. Leads independently.
Scope and Influence	(Two-Pizza) Team Manager. Rarely manages L6 SDEs. Influences team priorities. May influence vendors or external partners.	(Two-Pizza) Team Manager. Manages SDEs up to L6. May manage SDEs in other locations. May influence teams that overlap in business/technology areas.	Typically manages other managers, but may manage a single team with a critical technology, major product, or business segment. Influences within own organization.
Execution	Tactical management. Learning to be strategic. Leads a team of SDEs to meet customer, business, and technology needs. Able to clarify technical requirements, negotiate priorities, and manage projects to deliver on time. Knows how to audit SDE work (code, design) to make sure results are high quality. Ensures technology solutions are testable, monitored, and maintainable. Accelerates progress. Clears or escalates blockers. Mitigates immediate risks. May be learning to hire, manage performance, coach, and promote. Sets the example for LPs.	Tactical and strategic management. Leads team to invent, evolve, and/or deprecate software solutions. Able to scope a project and evaluate releases to ensure they meet business needs. Knows how to discern which features are essential, can be triaged, or omitted. Determines where to simplify, optimize, or extend solutions for the best outcomes. Drives resolution to software architecture deficiencies. Handles most problems, decisions, and escalations. Mitigates long-term risks. Finds a path forward in difficult situations. Effectively hires, manages performance, coaches, and promotes. Evangelizes LPs.	Strategic management. Delivers a vision. Leads team(s) to evolve/create software architectures and/or product solutions that are exemplary in terms of robustness, stability, scalability, and cost-effectiveness. Able to proactively identify technology gaps and opportunities. Determines when to make a case for resourcing (and when not to). May negotiate new boundaries for their area of ownership to achieve a better outcome. Handles critical problems, decisions, and escalations. Mitigates risks created by complexity. Ensures the right people are in the right job. Actively builds succession plans. Creates leader opportunities. Audits LP adherence.
Communication	Drives effective business and technical discussions. Writes clear documentation. Able to create an engineering roadmap. May be accountable for **COEs**[9].	Participates in strategic initiatives (1-3 year). Writes effective communications (e.g., Mission, Tenets, PR/FAQ, etc.). May have input into OP1/OP2.	Communicates and drives strategic initiatives (3-5 year). Makes business cases for external communications (i.e., blogs, IEEE whitepapers, conferences, etc.). Inputs into OP1/OP2.
Technical complexity[10]	Able to lead a team of SDEs tasked with building or solving **difficult**[11] software problems. Makes trade-offs: features vs. resources (time) vs. quality.	Able to lead a team of SDEs tasked with building or solving **complex**[12] software problems, often at the architectural level. Makes trade-offs: short-term vs. long-term needs	Able to lead one or more teams of SDEs tasked with building or solving **significantly complex**[13] or endemic software problems. Makes trade-offs: opportunity vs. resources vs. sustainability
Impact	Moderate. Decisions impact customers, software quality, performance, and costs (related to solutions provided by the team).	High. Decisions impact a software architecture and/or major product. May be accountable for OP1/OP2 goals.	High. Decisions impact a significantly large or strategically important architecture and/or major products. May have S-Team Goals.
Process Improvement	Improves software development process efficiency, metrics, test, and operational excellence practices.	Establishes and/or streamlines development, test, and operational excellence best practices. Shares management best practices with peers.	Actively works to remove bottlenecks. Sets targets for engineering excellence metrics. Drives best practices across own organization. Establishes team access to Principal Engineers, etc.
Experience	2+ yrs. as a SDE, TPM, or similar role. Meets SDM Tech bar.	5+ yrs., in a software role	10+ yrs., in a software role. Ideally some experience managing Mgrs.

[9] **COE** - Correction of Errors

[10] **Hiring Managers and Bar Raisers** – Please refer to the SDM Tech bar by level in **FAQ #6** at the end of this guideline.

[11] **Difficult** problems/efforts have visible risks or roadblocks; requiring skill and a considerable amount of work to resolve/deliver results.

[12] **Complex** problems/efforts have visible risks, roadblocks, and *constraints* (e.g., differing customer behaviors/cultural expectations, data quality issues, conflicting business rules, compliance requirements (e.g., financial legal, regulatory agencies), architecture or technology limitations, extensibility asks (e.g., enable clients to customize behavior without direct work from an engineering team), legacy dependencies, performance (e.g., implementing an API with requirement of p99.9 < 200 ms is a lot harder than p99.9 < 2 seconds), and scalability (e.g., designing a payment processor connector that processes 2M messages per hour is a lot harder than 2K messages per hour).

[13] **Significantly complex** problems/efforts have visible and not-yet-visible risks, roadblocks, constraints, and many *conflict* with each other (i.e., resolution of one issue creates a conflict with the resolution of another issue; multiplied by the number of issues). Requires significant expertise to see around corners, make the right trade-offs, and design a solution that is appropriately simple (doesn't add to the complexity). Due to the stakeholders involved (i.e., those blocking or driving the constraint, senior leaders, etc.) achieving alignment on an approach or implementation is more challenging and the trade-offs made usually have long-term impacts.

2. SDM I

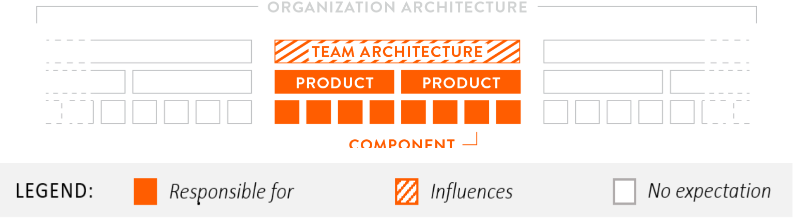
There is no SDM I (Level 4). The entry point into management for this role is SDM II (Level 5).

3. SDM II

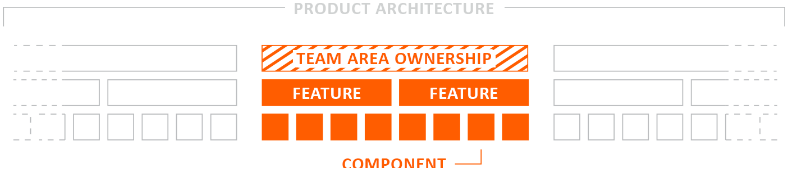
3.1 Scope and Influence

At this level, SDMs are growing into a full ownership capacity. When they achieve level 6, they own the strategic, long-term technology direction of the team's software architecture/product features.

To visualize scope and influence this guideline provides two examples. The first applies to a type of team that owns multiple tools and services.



The second applies to a type of team that owns a single piece of technology or major features that are part of a larger product architecture that is spread across multiple teams in an organization.



3.2 What you do…

You directly manage a team of SDEs working to deliver technology solutions. You team owns existing software in a segment of a business, functional area, or technology (e.g., set of containers for Beanstalk). You are responsible for the execution of the team and the management of the engineers that report them. As the manager, you are accountable for the customer experience, project prioritization, software quality, and system maintenance. You are also accountable for the experience of the SDEs that report to you. You set the example for our Leadership Principles. You help them grow by encouraging them to contribute to cross-team technical discussions, supporting their ideas, and empowering them to make decisions. You actively work to make sure they are assigned to projects to develop and stretch their skills. You take the time to understand the larger picture (i.e., customer needs, business opportunities, and problems to be solved) and share that knowledge. You clarify your team's mission, vision, and priorities. You help them see how their work contributes to organization goals. You seek long-term, strategic guidance from your manager.

You own the day-to-day management of software development. You understand the pros and cons of various agile methodologies (e.g., Scrum, Kanban, Lean, etc.) and help your SDES adopt the best approach for optimal team productivity. To make sure they develop the right solutions, you utilize the **working backwards**[14] process and facilitate their connection to customers, Senior SDEs/Principal Engineers, and subject matter experts. You make sure business and engineering requirements are well-defined. You drive constructive discussions and dive deeply into technical details (e.g., dependencies, design choices, operability). You transform raw thoughts into clear direction. You may partner with a TPM, PMT, and/or UX Designer. If involved with mechanical engineering or device development, you may partner with hardware engineering or systems development teams. You help scope development efforts, identify risks, and effectively prioritize deliverables. You make sure your team has adequate time to design the right solutions. You protect them from being over-committed, recognizing that their ability to meet every expectation will at times be constrained. You stay connected and resolve blockers that could derail launch schedules. You proactively communicate priorities, development status, and any team challenges to your customers, TPMs/PMTs, stakeholders, and management. You ensure everyone is aware of how decisions are made and are notified when schedules change. You make sure the team thinks and implements globally.

[14] https://w.amazon.com/index.php/WorkingBackwards

You have a solid understanding of the design approaches and industry technologies utilized in your team. You understand the overall architecture and foster great working relationships with any teams that share that architecture. You help make trade-offs; balancing the larger picture (e.g., resourcing, business goals, user experience, dependency impacts, efficiency, availability, etc.) against the needs of building, maintaining, refactoring, scaling, and extending features to properly support the life of the technology. You make sure your SDEs are able to spend adequate time on design and you seek their input on technology recommendations. In the absence of a Senior SDE, you are able to review software code (for style, readability, quality, and maintainability). You drive constructive technical discussions and audit as needed to prevent the consequences of poor technology decisions. You are able to recognize when a proposed design is too complex or risky (and arrange for additional reviews with Senior/Principal Engineers).

You prioritize operational excellence. Your team has a structure for testing software defects and maintaining software. You ensure compliance with policies (e.g., information security, data handling, PCI, accessibility, service level agreements.) You determine if the correct metrics are in place to measure the customer experience, and if not work to define them. You allocate development time to setup proper logging, operational metrics, customer intake, alarm mechanisms (e.g., TT/SIM), and runbooks/documentation (e.g., tutorials, help pages, troubleshooting). Your team has a sustainable oncall rotation that honors SLAs without burning out engineers. You prioritize root cause resolution, software enhancements, automated testing, and other projects that improve your customer’s experience, system quality, and the team environment. You hold postmortems and document outage impact and lessons-learned in COEs to ensure that problems don’t repeat.

3.3 Moving to SDM III…

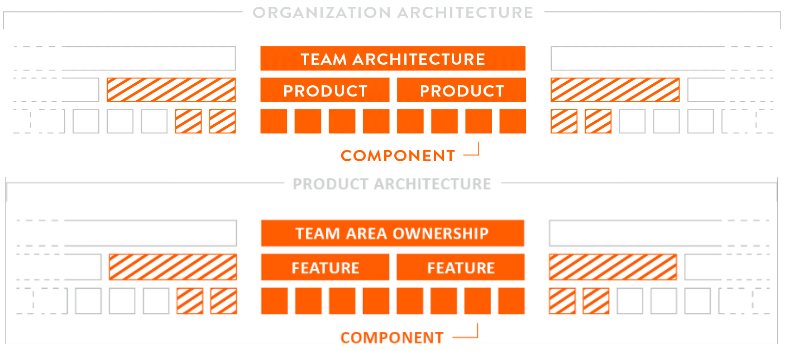
You will be considered for promotion to SDM III if you consistently demonstrate a combination of the following:

- You independently manage a team of SDEs. You hire the right mix of SDEs to accomplish team goals. You are able to assess SDE performance and have experience managing both high and low performers. You take effective action addressing employee concerns. You are able to hire, develop, and promote SDEs (up to level 5).
- You define the strategic vision for your team, partnering effectively with customers and stakeholders. You establish a roadmap and successfully deliver software that executes that vision.
- You are able to independently make short-term vs. long-term decisions. You define clear goals and objectives. You make crisp decisions about what projects move forward and in what priority order.
- Your team has an effective development process and consistently delivers software with minimal defects and system disruption.
- You delegate appropriately, assigning the right level of tasks based on the strength and development objectives of the individuals on your team.
- You proactively identify risks and bring them to the attention of your manager, customers, and stakeholders with plans for mitigation before they become roadblocks. You know when to escalate to Senior Leaders.
- You communicate ideas effectively, both verbally and in writing, to all types of audiences. You are able to write narratives, PR/FAQs, and other strategic documentation.
- You work successfully with customers, leaders, and other engineering teams. You foster a constructive dialogue, harmonize discordant views, and lead the resolution of contentious issues (influence and build consensus).
- You define metrics to measure team development progress, customer experience, and operational excellence.
- Your team has customer engagement mechanisms, adheres to SLAs, and effectively manages software operations.
- The quality of the software owned by your team and operational performance metrics have improved under your leadership.

4. SDM III

4.1 Scope and Influence

At this level, SDMs independently manage a team that owns a large portion of software architecture (e.g., services, process, functionality, etc.) with considerable impact on a customer segment, business, or technology domain. They fully own the strategic direction.



4.2 What you do…

You independently manage a software developmaaent team. You may manage employees in other locations, but rarely manage other managers. You may manage the type of team that owns cross-functional software (e.g., Search experience), a single product in a critical technology area, or that develops software used by multiple organizations internally. You are able to represent your team autonomously and require minimal guidance. You are accountable for the team's structure, software, and strategic direction. You partner with your engineers and leadership to define a vision and tenets that communicate how your team makes decisions. You provide the larger picture, the long-term perspective, and the context (past, current, and future) behind business and technology choices. Your roadmap influences organization goals. You effectively participate in Amazon's planning process (OP1/OP2). You evangelize our Leadership Principles.

You establish structures that enable your team to solve complex problems and deliver large systems. You drive mindful discussions that lead to crisp decisions. You partner with your customers, your engineers, and other teams on what projects move forward and in what priority order. You write press releases (PRs) that imagine what we would say to customers on the release date of a new product and make sure your engineers have enough specificity to build the right solutions. You are able to evaluate whether a product to be launched meets the intent of the PR and Working Backward documents. You help your team discern which features are essential, can be triaged, as well as which can be omitted altogether. You do not allow software work to be de-scoped to a less than acceptable product. You drive reasonable schedules and will readjust priorities to ensure the right customer, technology, and engineering outcomes.

You understand the products and systems your team owns, their limitations and reasons behind architectural decisions *(Q. Why did we build X? What business goal does it solve? What business or technical assumptions were made?).* You may personally coordinate discrete segments of cross-functional development work (e.g., solutions to meet business goals or require changes that involve multiple teams) or you may opt to delegate that responsibility to a TPM, PMT, or an engineer on your team considering those positions. You drive the simplification and optimization of project delivery. You structure efforts to reduce your team's exposure to classic failure modes (e.g., requirements not sufficiently understood, ineffective team collaboration, long-term impact(s) from of the use of third-party technologies, insufficient testing). You tackle ambiguous problems and proactively mitigate risks before they become roadblocks. You demonstrate good judgment in how and when to escalate without damaging relationships. When confronted with discordant views, you are able to find the best way forward and influence others to follow that path (build consensus).

You support the adoption of high quality engineering practices. You know the level of deferred maintenance for every system and product owned by your team. You appropriately allocate engineer time towards improving backend quality and efforts that reduce technical debt (e.g., endemic root cause resolution, refactoring, task automation, etc.). You collaborate with your engineers on when to shift roadmap priorities and dedicate resources towards long-term technology projects. You provide your team with the necessary support to take responsibility for their systems end-to-end (design, code quality, system health). You are strategic about senior engineer growth and provide those interested with opportunities. You build a team that delivers successfully without you. When you are involved, you use your technical judgment to question proposals and test assumptions (Q. *Do we need to build this at all?*). You help your SDEs learn how to make smart trade-offs (e.g., time vs. effort vs. features). You carefully consider short-term solutions and make sure their impact on the long-term architecture is properly assessed. You recommend reviews with Senior/Principal Engineers when solutions appear to add architectural complexity or impair future innovation. You define audit mechanisms and metrics that enable you to quickly explain your team's performance and any variance against your goals. You know when it is right to alter your team structure to address emerging priorities and allocate resources to other roles that reduce developer time spent on non-development tasks (e.g., QAE, TPM, PM-T, UX Designer, etc.). You keep your technical skills relevant.

4.3 Moving to Senior SDM…

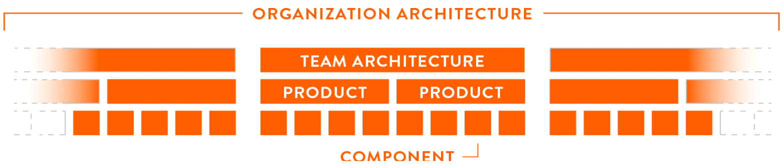
You will be considered for promotion to Senior SDM if you consistently demonstrate a combination of the following:

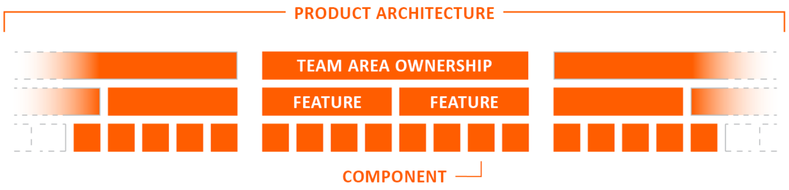
- You are a leader of leaders. You independently manage one or more development teams and may manage Senior SDEs. You are able to hire, develop, and promote SDEs (up to Level 6). If your role is scoped to manage other managers, you have demonstrated that you can hire, develop, and promote managers (up to Level 6).
- You own an area of strategic importance with significant business and/or technology impact. You define the 3-5 year strategic vision, have a long-term roadmap, establish the right software team structure, and lead your team(s) as they deliver high-quality, maintainable software that execute on that vision.
- You build leaders, effectively delegating responsibility over portions of software architecture, product(s), and/or a business domain to SDM(s), TPM/PMT, and senior engineers. You provide the right level of oversight to allow them take ownership and evolve technologies with some direction, diving deep when needed.
- Your team(s) are structured to sustainably meet customer, business, and technology needs. You leverage talent as needed and make the appropriate people trade-offs to meet incoming demands. Your strategic plans include how your engineering teams will scale or need to accommodate business needs. Your team(s) deliver consistently without wearing out key individuals.
- You work effectively with other technical leaders. You influence other team roadmaps, are able to reach consensus on approach/prioritization, and successfully deliver cohesive solutions to meet business needs.
- You identify large risks, endemic problems, areas of opportunity/customer value and when needed, are able to influence other teams to collaborate with your team(s) to build technology solutions.
- You create a culture of innovation, encouraging your team(s) to find new and better ways to evolve software and achieve goals. Your team(s) demonstrate that they take risks without fearing failure.
- You demonstrate the ability to distinguish between one-way and two-way doors. Your judgment has led to your team(s) taking smart risks and/or innovating with two-way doors.
- The quality of the software architecture(s) owned by your team and the level at which they need to be maintained has improved under your leadership.
- You champion software engineering and operational excellence best practices in your team and influence other teams in your organization.

4. Senior SDM

5.1 Scope and Impact

Senior SDMs own a software area of strategic importance and significance within an organization.





5.2 What you do…

You establish and manage scalable engineering team structures that create, correct, and evolve software that runs a segment of the business or customer experience. You are a leader of leaders. You own a strategically important software architecture, major product, and/or suite of products with significant business impact. As a senior manager, you are responsible for establishing the right engineering team structure and overall technology strategy to meet a long-term business need. You may have SDM manager reports (i.e., manage multiple teams) or you may manage a single team of senior SDEs with a charter to innovate/disrupt a strategically important domain. You determine whether it is appropriate to make a case to resource a large engineering initiative (spanning multiple years) and when to prioritize the resolution of an endemic problem over a current request. You are able to recognize where there needs to be a team boundary to better encapsulate technologies and where teams could combine for a better outcome. You are accountable for the execution and overall experience of your engineering team(s), the systems they create, the quality of the experience they provide, and how they meet business goals. To make sure everyone understands our Leadership Principles, you audit adherence and drive initiatives that show them in action.

You have a strategic focus, yet retain an appropriate level of hands-on management. You partner with business leaders, technical managers, and Senior/Principal Engineers to determine the long-term viability of architectures and product offerings, from business and technology standpoints. You stay abreast of relevant development and innovation occurring both inside and outside the company and use that knowledge to inspire big ideas. You trust your team(s) to innovate and deliver, verifying that they are achieving intended outcomes. You develop closed-loop mechanisms that enable you to quickly identify when projects are at risk and then act decisively to get them back on track. You ask the hard questions (*Q. Where do we block the business? Should this software continue to exist?*). You are able to decompose significantly complex problems into straight-forward solutions. You consistently bring strong, data-driven business, and technical judgment to decisions. You recognize when designs/solutions require technical guidance from Principal Engineers. You help make trade-offs and overcome difficult obstacles. You escalate appropriately, communicating any serious risks to your management and senior stakeholders. Positive or negative, you own the results. You visibly recognize success by giving credit to engineers, managers, TPM/PMs, and any others who contributed.

You understand the functionality, architecture, business drivers, and technologies behind the products and systems you own. You collaborate with SDMs and Senior SDEs/Principals to determine the long-term viability of your software architecture, both from a business and technology standpoint. You structure your team(s) to ensure gaps/opportunities in or between regions, architectures, and organizations (e.g., services, workflows, tooling) are identified and mitigated. You establish mechanisms that audit the customer experience, quality, operational performance of the software produced by your team(s). You regularly review business, performance, and operational metrics. You define standards of excellence and drive your team(s) to adopt best practices. You establish and regularly review business, performance, and operational metrics. You prioritize efforts that improve overall architecture, software quality, testability, maintainability, and efficiency. You demonstrate frugality by prioritizing the right initiatives to achieve the greatest impact with the least amount of resources. You look at where we apply the same engineering effort year-over-year to improve efficiency. You remove bottlenecks to enable your team(s) to work independently. You drive your team(s) to proactively identify gaps and opportunities (e.g., services, workflows, tooling) within or between regions, architectures, and organizations. You make sure risks are mitigated.

As a leader of leaders, you take the time to grow successful senior engineers and managers through collaborative development planning, coaching, and strategic project assignment. You identify future leaders, provide challenging opportunities, and facilitate their exposure to senior management. You create a positive culture and participate in efforts to drive improvements within your organization (e.g., recruiting, peak readiness, hardware efficiency, Connections/Tech Survey action plans, and Operational Excellence initiatives). You may run talent reviews (OLRs/Promos). You confidently handle employee situations, soliciting input from HR and your manager when necessary.

5.3 Moving to Director, Software

You will be considered for promotion to Director, if you consistently demonstrate a combination of the following:

- You manage an organization that grows technical leaders. You successfully deliver results through others without burning out your engineers and management. You intentionally design your organization to be self-sustaining. Your teams are better for your presence, but do not need *you* to be successful.
- You drive engagement, innovation, and retention initiatives in your own organization. You establish structures that result in individual/management skill and career growth. You hire, develop, and promote SDEs, SDMs and/or other technical roles up to Level 7.
- You are responsible for large-scale, business critical architectures, and/or major software product(s).

- You are a strategic thinker with a company-wide perspective. You defined a long-term vision and high-level plans that span multiple years with work that potentially affects many engineering teams.
- You may have managed teams outside of your country or have had an international assignment.
- You are a senior member of your organization’s leadership team and regularly participate in high-level planning (OP1/OP2, 3-year Roadmaps), business reviews, talent reviews (OLRs/Promos), and compensation reviews.
- You manage transitions in your organization successfully. You have a strong succession plan for your own role and for critical roles reporting to you. You lead your teams through change, helping them to understand reasons and maintain success despite any setbacks.
- You influence roadmaps and goals in and outside of your organization, both individuals and teams. You are recognized by your customers, peers, direct reports, and senior leaders for high-value contributions.
- You successfully influence at the Director level, through the effective use of data, written narratives, and sound business judgment. You escalate effectively across all levels of leadership.
- You create mechanisms to get the right information at the right time. You drive accurate metrics representing business performance, customer experience, and the operational effectiveness of your systems. Your organization achieves a consistency of excellence.
- You elevate the standards of excellence in your organization. The quality of the products and technologies you own, your customer and employee experience, your business and operational metrics have all improved since you became the leader of the organization.
- You are an active mentor, trainer, and a sponsor of cross-organization initiatives or initiatives at your location.

SDM CLARIFICATION FAQ

1. Why is there no SDM I (Level 4)?

Amazon Level 4 expectations do not allow for much ambiguity or complexity. Employees at this level may be recent graduates with no industry experience. It was determine that someone at this level would have a hard time succeeding.

2. Why don't Level 6 SDMs include managing other managers?

Amazon levels are clear that managing other managers is a Level 7 skill. This doesn't mean that Level 6 SDMs cannot manage managers, it just means that in this regard they are demonstrating a *next level* skill.

3. Amazon Leadership Principles seem to be missing. Is this intentional?

Yes. As the foundation for our company culture, Leadership Principles apply to all Amazonians in all roles at all levels. They are not role-specific and do not need to repeated in role guidelines.

4. Is the SDM role guideline to be used for hiring?

Yes. Amazon role guidelines convey company-wide role and level expectations. To ensure we are consistent across all organizations and locations, they should be used in all aspects of assessing performance or level. This includes hiring, performance reviews, next-level goal setting, promotion justification and promotion reviews.

5. Some SDM expectations overlap with TPMs and PM-Ts. How can both be responsible for the same tasks?

SDMs own their team development process and software end-to-end. If they do not have a TPM or PMT, then any product, project or program-related responsibilities fall directly to them. At Level 6 (SDM III) they may decide to hire a TPM or PMT as a single-threaded owner on a strategically important program or initiative. In those cases, SDMs need to refer to the TPM role description and PMT level guideline to clarify which types of responsibilities fall into their expectations and delegate ownership and responsibilities appropriately.

6. What is the SDM Tech bar?

At a high level, SDMs need enough knowledge/experience with design approaches) to be able to ask the right technical questions and drive the right technology solution(s). They should be able to constructively contribute to technical discussions – i.e., able to understand and evaluate end-to-end system designs for strengths and weaknesses (scalability, latency, security, performance, data integrity, etc.). They also need enough expertise to foresee the consequences of poor technology decisions to determine if the right trade-offs are being made. They need to recognize when the technical information they have been given is potentially inaccurate, inflated, or not sufficiently understood. This is not as much about possessing specific skills as it is about having sufficient technical understanding and *judgment* to know when a software design is complex or risky enough that it requires further evaluation from a Principal Engineer. SDMs need sufficient technical credibility to be respected by SDEs.

For Bar Raisers and Hiring Managers, here are the specific SDM technical expectations by level:

The Tech Bar includes any technical skills outlined in previous levels (i.e., a Sr. SDM has all of the skills documented at lower levels). These are minimum requirements. Organizations may expect a higher bar, but not a lower one.

SDM II:

- College education/personal study in Computer Science, Computer Engineering, Mathematics, Physics, or other STEM discipline. A degree is **not required** if the candidate has skills that meet the bar below.
- Understands the industry technologies in software domain and how to apply those technologies to solve problems *(Q. What do we need to build to achieve X?).*
- Able to dive deeply into technical details (e.g., key dependencies, design choices, operability, etc.) with engineers and drive a constructive technical discussion.
- Can convey detailed technical knowledge (verbally, in writing, and via diagram)
- Understands the architecture of the intended team systems. Knowledge is sufficient to be able to make sure that dependencies are not broken by development changes and that systems are not broken by upstream changes.
- Able to consider the larger picture (e.g., efficiency, availability, operability, scalability, business goals, customer experience, etc.) and balance with building, maintaining, and extending features for the life of a technology.

- Knowledge of development methodologies. Can help their team implement the right approach.
- Able to assess engineering practices for testing, triage and maintenance.
- Could represent their team in a technical capacity and to make commitments to customers and management.
- Can be held accountable technical decisions. *(Q. Tell me about a difficult business/technical decision where you were on the hook? What level of authority did you have?).*
- Previous experience as a coder is not a hard requirement, however it may be a functional requirement if the team provides software for developers/development teams. (e.g., SDMs in Builder Tools need to be able to code).
- May need to be capable of reviewing code (style, readability, quality, maintainability).

SDM III:

- Understands system limitations, scaling factors, boundary conditions, and/or the reasons for architectural decisions *(Q. Why did we build X in this way? What business assumptions were made? What technical assumptions were made? Do we need to build something else– if so why?).*
- Knows how to decouple software dependencies (e.g., SOA best practices), prevent duplicate software efforts, as well as collisions/outages.
- Has a larger business and architecture view. Uses technical judgment to inform technology and business trade-offs. Able to make sure SDEs are focused on value-added efforts.
- Can hold an engineering team to a high standard for both solutions and engineering practices.
- Capable of reviewing design decisions (in cases when a Senior/Principal Engineer are not available).
- Simplifies concurrent project delivery, including development and testing for projects that cross team boundaries.
- Is current on industry technologies in their software domain.

Senior SDM:

- Broad understanding of company systems/technologies. Able to apply this technical knowledge to invent, evolve, improve, simplify, etc.
- Knows how to drive large-scale engineering efforts that solve significantly complex or endemic problems for customers, business, or technology teams.
- Can identify gaps/opportunities between regions, architectures, etc. (e.g., services, workflows, tooling).
- Can deconstruct complex processes into straightforward solutions.
- Has strong, data-driven business and technical judgment. Recognizes when designs/solutions require additional technical guidance (e.g., from senior engineers or subject matter experts).
- Knows how to make a case for engineering efforts and remove areas where we exert that effort multiple times.
- Can drive architecture or organization changes to enable teams to work independently and/or achieve a significant efficiency improvement.

7. Does the same technical bar apply to SDMs and TPMs?

Essentially yes, however TPMs differ in the management of programs and expectations to manage all aspects of cross-functional projects that span organization boundaries.

8. Are there specific examples for SDM performance at each level?

No. Role Guidelines describe *company-wide* expectations. We have a huge variety of technology products, customer needs, and organizational structures. And as we all know, Amazon is not prescriptive. It is expected that SDMs and their leaders understand their space and how to map generalized level guidelines to appropriate goals. Another reason that we do not include specific examples is to avoid **anchor**[15] bias. Providing a reference list for each level, risks teams overly indexing on the listed types of products and technology solutions. They may attempt to pattern match rather than applying more critical thought on whether SDMs are meeting the criteria to be considered for promotion (and thus demonstrating they can meet the challenges of the next level). We don't want to introduce inappropriate barriers when hiring or promoting. ("*The solution described is similar to solution foo on the list of solutions that got people promoted, but this particular solutions is missing N things that foo had, therefore, I'm not inclined to promote*.")

[15] See: http://en.wikipedia.org/wiki/Anchoring