Growth Plan – Hard Skills - Examples

Table of Context

27. Understands the tradeoffs between testing shapes (e.g. Pyramid, Trophy, Honeycomb) as well as how to apply them
28. Considers Readability of Code (Understands the audience of their code is other Engineers not the computer)
29. Consider unit tests for all code, implement, and maintain appropriately
30. Consider integration points and avoid fragility
31. Consider SEO impact
32. Mentoring other Engineers in areas you are strong in
33. Speaking the language of engineering
34. Finding technology industry trends and bringing the knowledge back into the team
35. Engineering is a craft to hone, not a skill to use. (You have seen someone do something better than you and you want to grow yourself the same way. Pursuing excellence and expertise)
36. Growth Mindset: Share new craft growth with Engineers around you
37. Evaluate technical skills of candidates through the interview process
38. Owns alerts and response (can diagnose an application issue without reaching out to Systems Engineering)
39. Log appropriate code diagnostic information

Higher Level Thinking

1. Can relate work being executed to a business problem
2. Can break down work using business context even when ambiguous
3. Actively leading initiatives for your team
4. Sitting with other business functions (to learn from their experiences and the problems they run into)
5. Measuring the impact of your code quality on customer rating, crash rates, bug rates, customer success issue counts, time to market
6. Technical writing can be written with both technical and business stakeholders in mind
7. Creates monitoring dashboards to track the health of the code deployed
8. Has enough business acumen to sit in business strategy meetings
9. Considers [and can explain the impact to] Global/EA while making technical decisions
10. Partners with leadership to make the best technical decision for the business
11. Creates clarity from ambiguity
12. Discovers, introduces, and champions innovation while considering total cost of ownership
13. Connected to the flow of work of the team (metrics, decomposition, engaged in rhythms)
14. Pull cards and work them with self-sufficiency and knows how to get unblocked
15. Uncovers and surfaces opportunities for technical improvement
16. Successful creating work that spans multiple workstreams. Beginning to blend and coordinate activities while creating the work

17. Understands that tickets don't stop when the merge button is pressed but consider up and downstream systems affected by the code created

Grading Scale Definitions

Unpracticed (0%-10%)

Ranging from I don’t know what this is or how to do it to I have heard about it and can do it with much effort and probably need help. I am learning the “what” of this behavior/skill.

Emerging (10%-40%)

Ranging from I am formally aware of this behavior/skill and am in the process of growing it intentionally to I have internalized the foundational concepts and can start executing on them with consistency over small periods. I am confident in the “what” but learning the “how” of this behavior/skill.

Practicing (40%-90%)

Ranging from being personally aware of the skill/behavior in an advanced depth and my growth in it is consistent over a period to I am aware of the skill/behavior and how to do it in my sphere shown by multiple periods of consistency. I am confident in the “what” and “how” but developing the “when” and “where” of this behavior/skill.

Consistent (90% +)

Ranging from I am consistent in foundational and intermediate components of this behavior/skill and execute them over periods to I am consistent in advanced concepts of this skill/behavior and execute them over multiple periods. I am confident in the “what”, “how”, “when”, and “where” of this behavior/skill.

Radiating

Competency in this behavior/skill is recognized, sought after, respected in my current sphere plus adjacent spheres. I am a change agent for this behavior/skill. My presence transforms spheres regarding this behavior/skill.

Engineering Ability

Category: Engineering Ability > Core Practices
Behavior: Baseline Skill in Coding: Engineer can effectively build, run and debug locally in a language

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior.
Emerging (10%-40%)	Team member has completed fundamental tutorials in their language and understands how to get a simple application up and running. Team member can run simple applications in docker when given instructions or a docker-compose.
Practicing (40%-90%)	Able to get any application in their chosen language up and running natively (not dependent upon docker), but may need help on specific library issues. Can connect to common debugger (both command line and IDE based tools). Competent utilizing basic debugging tools. Understands which IDE errors are serious, may ignore certain errors that don't prevent building.
Consistent (90% +)	Able to get any application in their chosen language up and running with minimal dependence upon another. Understands which issues are common, and has resolved many of them before in their chosen language. Can connect remote debuggers even when not running application through IDE (i.e. IntelliJ Run Configuration). Understands multiple ways to connect debugger to their application. Adds helpful details to README's when encountering new issues. Understands why their IDE reports specific errors and how to resolve them, even for IDE errors that may not prevent building. Can map build, debug, or IDE concepts from one language, framework, or toolset to another.
Radiating	Helps other through build troubles related to their language (i.e. Nokogiri or mysql gem build issues in Ruby). Teaches others how to debug using their more familiar debugging libraries or IDEs. Teaching others how to connect remote debuggers to applications not running through IDE. Able to share lessons learned, shortcuts, or specific tools relating to their IDE and source code editors.

Category: Engineering Ability > Core Engineering
Behavior: Foundational (working) competence in a language

	Example
Unpracticed (0%-10%)	Team Member relies on web searches and team mates to get help with basic syntax and built-in methods.
Emerging (10%-40%)	Can complete basic technical requirements in their code. There may be a better way, but the code “works”
Practicing (40%-90%)	Growing in knowledge of the language. Requires little to no reference for language-specific syntax.
Consistent (90% +)	An expert in the essential aspects of a language. Can identify areas for optimization in a code base and can coach others how to optimize.
Radiating	Consistently coaching others in language paradigms. Often an early adopter for new language features.

Category: Engineering Ability > Core Engineering
Behavior: Comfortable and fluent working in the entire technology stack

	Example
Unpracticed (0%-10%)	Team Member may appear hesitant, struggling to navigate seamlessly through different layers of technology. They might exhibit a lack of confidence or proficiency when faced with tasks that involve multiple components within the technology stack.
Emerging (10%-40%)	Team Member demonstrates a growing confidence in handling various components within the stack but could still encounter occasional challenges. They might require some support or additional learning opportunities to fully master the intricacies of the technology stack.
Practicing (40%-90%)	Team Member exhibits a high level of confidence and proficiency. They navigate seamlessly across different layers of the technology stack, effectively troubleshoot issues, and demonstrate a comprehensive understanding of how various components interact. Their adeptness allows them to contribute efficiently to projects involving diverse technological elements.
Consistent (90% +)	Team Member consistently demonstrates a seamless and expert level grasp of the entire technology landscape for their team. They can navigate effortlessly through different layers of the technology stack, troubleshoot effectively, and contribute adeptly to a range of projects, This individual's competence is a source of inspiration to the team, setting a high standard for the team's technological achievements.
Radiating	Team Member is a beacon of expertise. Their ability to seamlessly navigate the entire tech stack not only showcases their personal

	expertise but also positions them as an inspirational guide for others. With a deep understanding of the technological landscape, they radiate confidence and fluency, elevating the entire team's capabilities. Their positive influence extends beyond individual contributions, fostering a culture of continuous improvement within the tech for their team. Help other team members understand disparate components and fill gaps and skill knowledge as needed.

Category: Engineering Ability > Core Engineering
Behavior: Confident in making changes in an existing codebase/feature or other's code

	Example
Unpracticed (0%-10%)	Team Member needs hands-on guidance through the codebase
Emerging (10%-40%)	Team Member may need initial guidance, but can self-serve familiarizing with the codebase
Practicing (40%-90%)	Team Member does not need any coaching in the new (to them) codebase, and can identify relevant features and functionality to be aware of as they are making changes
Consistent (90% +)	Team Member requires no coaching, and identifies opportunities to “boy scout” the code base, making it better than they left it
Radiating	Team Member coaches others in making existing codebases better, giving chapter talks and lunch-and-learn sessions showcasing case studies of codebase improvements

Category: Engineering Ability > Core Engineering
Behavior: Confident to begin a greenfield codebase/feature. Knows how to start from new

	Example
Unpracticed (0%-10%)	Team Member leans heavily on tech lead and other team members for guidance in starting a new project. Does not actively take part in standing up any infrastructure or architecture but waits for infrastructure to be stood up before contributing to the codebase.
Emerging (10%-40%)	Has an understanding of what must be true to start a new project. Partners with other team members to stand up infrastructure, and can begin to contribute early features in the codebase

Practicing (40%-90%)	Comfortable standing up infrastructure on their own. Able to identify and execute on the path to achieve an MVP.
Consistent (90% +)	An expert in the Ramsey Product Playbook. Knows how to go from 0 to 1 quickly and can help a team activate on early feature development.
Radiating	Plays an active part in writing the Ramsey Product Playbook. Coaches teams on how to start projects and achieve an MVP quickly while not compromising quality.

Category: Engineering Ability > Core Engineering
Behavior: Deep understanding of primary language (performance concerns, best practices, standards, which to choose and why)

	Example
Unpracticed (0%-10%)	Team Member needs coaching in optimizing and improving their code and needs education around team and company coding standards
Emerging (10%-40%)	Team Member has learned a handful of patterns for optimizing and improving their code and understands the most important or most frequently used standards.
Practicing (40%-90%)	Team Member has a growing vocabulary of optimization patterns and standards and is actively discovering more on their own. Can easily identify the absence of a standard or best practice, and calls them out in PR’s
Consistent (90% +)	Team Member is the champion for optimizing their codebase(s) and is often blocking PR’s that do not adhere to established standards. They are actively establishing new standards, optimizations, and best practices to the team and holding them accountable to raise the bar for their codebase’s quality
Radiating	Mentoring and teaching others about deep or uncommon features of the language (threads, reflection, metaprogramming, etc.). Actively engaged with language direction and new features to understand how and when they impact how we use the language internally.

Category: Engineering Ability > Core Engineering
Behavior: Defensive coding (Avoid problematic issues before they arise. Practice of eliminating assumptions in code)

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior and does not understand why defensive coding is needed. Team member rarely considers more than just the "happy path." Guidance and training is needed to help the Team Member realize the assumptions that he is making, the unintentional risk trade-offs being made, and what can happen when reality meets code.
Emerging (10%-40%)	Team member has learned basic defensive coding patterns and knows that the patterns should be used. It is not yet a habit. Team member uses them occasionally and it takes a lot of mental energy to do so. Team member can articulate some of the risks and assumptions being made in their code.
Practicing (40%-90%)	Team member is consistently using defensive patterns and knows how to apply them in most levels of their code bases. Team member can articulate most the risks and assumptions being made to his team and leader. Team member is still figuring out the risk-benefit analysis when determining the level of defensive coding needed for the goal at hand
Consistent (90% +)	Team member consistently delivers the right level of defensive coding for the goal at hand. He understands the risks involved, the assumptions being made, and can properly communicate this to his team, leader, and stakeholders.
Radiating	Team member can teach others the why, what, how, and when of defensive coding and craft the teaching to the recipient's level of understanding. Team member is creating engineering cookbooks with common examples and pitfalls to ease adoption of defensive coding. Team member is on the lookout for patterns being used across Ramsey that can either be centralized in a library or replaced by an open source alternative.

Category: Engineering Ability > Core Engineering
Behavior: Self-improvement of Craft Based on Patterns Learned from Mentorship, Feedback, or Bugs

	Example

Unpracticed (0%-10%)	Team Member expresses little or no ownership over bugs they've introduced. Team Member expresses little or no interest in improving, treating each body of work as a task rather than an opportunity for growth.
Emerging (10%-40%)	Team Member seldom seeks feedback, but does engage with feedback when received. Team Member puts some thought into improvement and seeks small wins in their day to day work, but seldom invests any time in their craft outside of Ramsey.
Practicing (40%-90%)	Team Member frequently seeks feedback from others and consistently learns from their mistakes. Team Member is likely seeking out a personal mentor or already has found one. Team Member shows interest in new and challenging problems and technologies. Team Member spends some time out of the office leveling up their craft.
Consistent (90% +)	Team Member actively seeks and engages in feedback and keeps the conversation going, asking detailed and well-thought questions with each encounter. Team Member is constantly looking for learning opportunities, and likely has one or more side projects outside of Ramsey. Team Member keeps up with the state of the art in their craft via blogs, articles, and videos.
Radiating	It's not uncommon to walk by Team Member's desk and hear them talking about something they've learned or some project they've built. Team Member proactively engages in the search for feedback and proactively provides feedback to others. Team Member is very clearly leveling up in their craft at a rapid pace, causing others to develop a sense of admiration and desire to follow in their footsteps.

Category: Engineering Ability > Core Engineering
Behavior: Zoom out to focus on high impact, multiplier code and design

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team Member is aware of the importance of considering high impact areas but is still in the process of refining this skill. They make efforts to refactor and optimize their code and designs for efficiency and scalability. While they can identify

	some areas where code improvements can have a multiplier effect, they often need guidance or input from more experienced team members to make informed decisions.
Practicing (40%-90%)	Team member consistently exhibits an understanding of the broader impact of their work. They proactively identify and address bottlenecks, performance issues, and architectural improvements within their projects. They can make decisions on where to invest time and effort to achieve maximum impact, demonstrating a strong grasp of both the “what” and “how” aspected of code and design.
Consistent (90% +)	Team member consistently produce high-impact code and designs. They have a deep understanding of best practices in code architecture and design patterns and apply them effectively. They also are adept at influencing and guiding other team members to prioritize higher impact areas and improve overall project efficiency. Their ability yto execute on this behavior is demonstrated over multiple periods and in various contexts.
Radiating	The team members ability to identify high-impact areas is not only respected but actively sought after within the organization. Their contributions have a transformative effect on project outcomes and overall engineering excellence. They serve as mentors and catalysts for others, setting new standards and driving change in the organization’s approach to engineering practices. They radiate competence and drive positive change.

Category: Engineering Ability > Core Engineering
Behavior: Evaluate upstream and downstream impact that unlocks company-wide business value

	Example
Unpracticed (0%-10%)	Team Member is not aware of external impacts to their code change and due to this may introduce changes that negatively impact others.
Emerging (10%-40%)	The team member occasionally thinks about the upstream and downstream impacts of their changes but often lacks a full understanding. They don’t yet know the critical dependencies and rely on input from other engineers to address the potential impacts properly.
Practicing (40%-90%)	The team member usually evaluates both upstream and

	downstream impacts of their code changes. They understand the potential ripple effects and often discuss their plans with other engineers to ensure comprehensive coverage.
Consistent (90% +)	The team member consistently identifies and evaluates the upstream and downstream impacts of their code changes. They proactively seek feedback from other teams and stakeholders to ensure their changes enhance overall business value, reducing the likelihood of unforeseen issues.
Radiating	The team member is an expert in our systems and uncovers potential side-effects that others might miss. They have a reputation for being an expert our systems. They mentor other engineers and promote best practices for comprehensive impact evaluation, driving a culture of strategic foresight across the organization.

Category: Engineering Ability > Core Engineering

Behavior: Able to communicate and articulate technical information to all levels of the discipline and company

	Example
Unpracticed (0%-10%)	Team Member uses the same terminology and depth of information when speaking to all levels without fully understanding that not everyone has the same background within the domain.
Emerging (10%-40%)	Team Member understands that when discussing a domain not all levels have a common dictionary or needs/wants the same depth of information. However, they will often fall back into the comfort of their knowledge and forget their audience.
Practicing (40%-90%)	Team Member understands that when discussing a domain not all levels have a common dictionary or needs/wants the same depth of information. When they start falling into the comfort of their knowledge, they will often recognize what they are doing and pull it back to the appropriate level.
Consistent (90% +)	Team Member is actively aware of different audiences’ needs and will adjust their terminology and depth of information to meet those needs. Lapses may still occur but happen infrequently.
Radiating	Team Member actively works with and assists their fellow Engineers to right size their communication approach.

Category: Engineering Ability > Core Engineering
Behavior: Has the ability to move between multiple squads to solve complex problems and influencing growth of the team

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Knows when and how to appropriately ask for help from another squad, using their chosen method of contact. Participates in code reviews and submits code to another squad's codebase. Is able to help another squad use code in their own squad's codebase with a specific, defined problem.
Practicing (40%-90%)	Makes contributions to multiple codebases, generally within the same channel. Begins to understand common threads between different squad initiatives. Can articulate and understand competing needs of other businesses and squads. Can articulate their own squad's interests and needs to other squads.
Consistent (90% +)	Actively collaborates with multiple squads, sometimes across channels. Aware of the inner workings of multiple codebases and affected squad's objectives. Knows when and how to reach out to other squads to inform of possibly impactful changes. There is inherent, proactive squad support built-in to the collaborative efforts.
Radiating	Actively collaborates across channels. Helps other squads adopt changes that will positively impact their objectives and the objectives of the current squad. Ties back common objectives to the primary squad goals.

Category: Engineering Ability > Core Engineering
Behavior: Can easily switch context while remaining focused on outcomes

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Is able to help another squad within the same channel with a specific, defined problem. Still able to deliver on current squad goals with minimal impact
Practicing (40%-90%)	Makes contributions to multiple codebases, generally within the same channel. Budgets time reasonably well. Begins to

	understand common threads between different squad initiatives.
Consistent (90% +)	Actively collaborates with multiple squads, sometimes across channels. Current squad delivery work may be impacted but the team member is able to communicate the trade-offs. There is inherent squad support built-in to the collaborative efforts.
Radiating	Actively collaborates across channels. Ties back common objectives to the primary squad goals.

Category: Engineering Ability > Design
Behavior: Can understand and use common design patterns

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Is aware of design patterns and can apply a design pattern to their code seen somewhere else. Can describe the design pattern and its best use case. Is aware of tradeoffs in design patterns.
Practicing (40%-90%)	Can describe common design patterns on whiteboard, Can argue for/against the use of particular design patterns including tradeoffs. Takes the time to apply design patterns in existing code when appropriate.
Consistent (90% +)	Routinely creates code that is flexible and consistent, leveraging common design patterns. Can argue the strengths and weaknesses of various design approaches independent of the code. Creates solutions that leverage multiple design patterns working together.
Radiating	Encourages other teams in the use and application of common design patterns to increase the readability and predictability of code. Can explain and apply design patterns in complex systems. Uses the whiteboard to discuss architectural patterns.

Category: Engineering Ability > Design
Behavior: Refactoring/de-duping

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Uses the IDE’s refactoring features for renames, to break out code into functions/methods, to inline code, etc. Identifies duplicate code and refactors usages to reference a single source. Knows that currently identical code may have different intent and can grow apart over time. Knows that the same intent can be found in code that looks different.
Practicing (40%-90%)	Looks for related data and code that could be grouped to create a class. Can identify when a model/class has mixed data or functionality and refactors this into separate models.
Consistent (90% +)	Continuously creates missing models from data and code spread over the code base. Continuously refactors existing code that have mixed purposes.
Radiating	Teaches others the aforementioned skills: IDE refactoring features, discerning code intent to correctly identify duplicate code, identifying when code has mixed purposes. Others apply the aforementioned skills because of your example and guidance.

Category: Engineering Ability > Design
Behavior: Create usable technical documentation for you, your team or someone with context to understand

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Understands design is a separate skill from coding. Beginning to think through designs before writing production code. Beginning to use whiteboards, LucidCharts, ADRs, and other tools to capture thinking.
Practicing (40%-90%)	Regularly spends time thinking through designs before writing production code. Understands the value of communicating designs with their team and seeks their feedback.
Consistent (90% +)	Is known for designing solutions with their team before

	shipping code. Seeks understanding and feedback on design before creating pull requests. Regularly refers to design documents as they iterate and evolve our products. Stewards our design artifacts, refining, improving, and culling.
Radiating	Ensures everyone on their team understands the value of thinking through solutions, exploring options, documenting the designs, communicating, and delivering on the design through pull requests and further maintenance. Mentors others on how to design. Others are more active at the whiteboard because of this person’s influence.

Category: Engineering Ability > Design
Behavior: Edge/corner cases define the boundaries of your domain

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The team member conceptually understands edge/corner cases and is starting to identify them for their domain. When developing solutions, the team member is beginning to think about the edge/corner cases constraining their solution, such as what data states are possible and must be accommodated.
Practicing (40%-90%)	The team member identifies most of the edge/corner cases for their domain and accounts for them in their solutions. Additionally, the team member writes tests targeting these edge/corner cases. The team member may still be learning the relative importance of their domain's edge/corner cases and may be experimenting with allotting the appropriate amount of time, effort, and resources to covering edge/corner cases.
Consistent (90% +)	The team member identifies and accounts for all edge/corner cases relevant to their solution, including appropriate (adequate, not excessive) tests. The team member knows their domain and its boundaries well enough to inform solutions for their squad as well as external contributions.
Radiating	The team member leads other engineers to consider the edge/corner cases for their domains and teaches techniques for effectively handling these boundaries. Through proficiency in navigating domain boundaries in their own work, insightful code reviews, consultations, and other discussions regarding

	edge/corner cases, the team member drives more robust software systems across all spheres they touch.

Category: Engineering Ability > Design
Behavior: Active management of security, privacy, and scale concerns

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Starting to consider security, privacy and scale when planning or building software. Needs coaching on when it's appropriate. Learning some patterns for addressing these concerns.
Practicing (40%-90%)	Understands common patterns for handling security, privacy or scale concerns. Intentionally applies those patterns to their work and is learning when is the right time to address a concern proactively. Course corrects after an issue is found and handled.
Consistent (90% +)	Proactively addresses security, privacy or scale to avoid issues before they become a problem. Knows how to balance trade-offs. Experienced at applying solutions to the kinds of non-functional attributes. Considers the impact to other technology and business teams.
Radiating	Regularly considers non-functional attributes like security, privacy and scale as a core part of the software development life cycle. Thinks broadly about the impact not just to software but also the business. Evangelizes patterns across the enterprise and finds ways to systematize their approach.

Category: Engineering Ability > Design
Behavior: Best practices around architecture

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The team member is actively participating in regular architectural design discussions with the team. They are beginning to incorporate technical design as a discipline in

	their work, and they show a developing understanding of the core systems, services, and applications they are responsible for.
Practicing (40%-90%)	The team member consistently documents architectural decisions and is not afraid to engage in impromptu (i.e., “whiteboard”) design discussions with other engineers. They have demonstrated a working understanding of the architectural systems they own, allowing them to make informed decisions.
Consistent (90% +)	The team member possesses a deep understanding of the architectural systems they own. They proactively challenge existing systems and take the initiative to pioneer new, simpler approaches when applicable. They can thoroughly comprehend C4 documentation for other teams and effectively communicate those concepts to key stakeholders, such as Business Unit leaders.
Radiating	The team member drives architectural patterns that have had a profound impact on and are widely adopted throughout the company. They set “vision” in architecture that has enabled products and services to adopt new technologies that drive our business. They actively identify and address engineering sprawl within the organization, working towards unifying architectural approaches across teams. They are skilled at disseminating this information.

Category: Engineering Ability > Design
Behavior: Creates data flows as a part of engineering efforts

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Logs out data from application activities. May consider data flows during decomp. Team member is beginning to dig deeper to validate data (like in Redshift) when needed to answer questions or address issues.
Practicing (40%-90%)	Data flows are considered as part of most tickets. They are implemented and documented in most circumstances. Developer works with data analyst on regular basis.
Consistent (90% +)	Data flows are considered as part of each ticket’s success

	criteria. Data flows are consistent, documented, validated, and communicated.
Radiating	Teaches peers how to think about application data, their process for capturing and consuming it, and the results they are seeing with that data.

Category: Engineering Ability > Design
Behavior: Knows when to use or create shared components (APIs, Libraries, etc.)

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Is able to identify opportunities to share code. May need functional help with the extraction. Can participate in an API contract design discussion, but may not have a lot to contribute.
Practicing (40%-90%)	Initiates extracting shared code to libraries as needed. Makes proposals for API contract designs and incorporates feedback.
Consistent (90% +)	Proactively looks for opportunities to extract shared code to libraries. Considers common potential use cases to unlock other teams using the library quicker. Leads API contract design discussions.
Radiating	Teaches others how to extract shared components and design API contracts. Promotes new libraries and APIs they become aware of within their business unit and discipline, even when built by others.

Category: Engineering Ability > Design
Behavior: Can identify, communicate, and audit the key risks in design and architecture (the key things you need to pay attention to) for solutions they aren’t involved in delivering the details.

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The team member is building a mental framework for risks in

	design and architecture. They are learning from participation in design sessions and from experience. If they encounter an issue with a design during implementation, they seek to understand how it could have been avoided.
Practicing (40%-90%)	The team member can identify key risks in design and architecture for systems they are less familiar with. They effectively audit designs to ensure potential issues are addressed and actively incorporate feedback from more experienced colleagues.
Consistent (90% +)	The team member has a depth of experience that helps them identify risks for designs and architectures in which they have less context. They can review designs made by engineers on other teams and have meaningful conversations about risk management.
Radiating	The team member excels at identifying, communicating, and auditing key risks in design and architecture. More than just communicating potential issues, they teach other engineers how to think about risks and know which ones that are most critical to the business.

Category: Engineering Ability > Design
Behavior: Consider the usability of the product, and collaborate with appropriate disciplines

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The Team Member is starting to consider the usability of the product and is beginning to collaborate with other disciplines, such as designers or usability experts.
Practicing (40%-90%)	The Team Member consistently considers the usability of the product and takes direction from designers and other relevant disciplines. They may participate in user interviews or run frontend experiments.
Consistent (90% +)	The Team Member consistently collaborates with designers to ensure the product has lovable moments. They are likely a champion for accessibility, and they display a self-employed mentality when finding issues with existing product experiences.

Radiating	The Team Member drives usability initiatives that have a profound impact on the product's success and are adopted throughout the organization. They set a high standard for usability and champion user-centric design practices across teams. For instance, they may lead workshops on accessibility, contribute to the Ramsey Design System, or present in the frontend chapter.

Category: Engineering Ability > Design
Behavior: Contributes to the channel's technical vision and strategy

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The Team Member participates in discussions about long-term goals and aligns to the channel’s technical vision and strategy.
Practicing (40%-90%)	The Team Member gives input in discussions about long-term goals, propose ideas for improving technical processes, and provide valuable input on strategic decisions related to technology and architecture. They collaborate with team members to align technical initiatives with the channel's vision.
Consistent (90% +)	The team member possesses a deep understanding of the channel's technical vision and strategy. They proactively contribute ideas and insights that drive the channel's long-term success.
Radiating	The Team Member has a large impact on the channel's technical vision and strategy. They have a firm understanding of where our technology falls short and where our greatest opportunities lie, and they can articulate it in a way that gets others excited. They drive innovation, set ambitious but achievable technical goals, and foster a culture of strategic thinking within the business.

Category: Engineering Ability > Design
Behavior: Drives architecture across a channel.

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior

Emerging (10%-40%)	Team member has a basic grasp of architectural concepts, such as server-client, or n-tier. They can contribute meaningful ideas to architectural design discussions with guidance and can follow architectural decisions made by more experienced team members while building an understanding of the trade-offs of those decisions.
Practicing (40%-90%)	Able to lead architecture discussions for small-scale projects or components within a larger system. Can drive architectural decisions with input from team members across their channel through mediums such as meetings, documentation, and informal discussions. Demonstrates an understanding of trade-offs between different architectural approaches and can articulate reasons behind architectural choices made.
Consistent (90% +)	Consistently drives architecture discussions and decisions for medium to large-scale projects within their division, considering various factors such as scalability, performance, security, and maintainability. Can effectively communicate architectural designs across different teams within the division, including managers and non-engineering disciplines. Actively seeks feedback on architectural proposals and adjusts them based on input received. Recognized as a subject matter expert in specific architectural domains within their channel.
Radiating	Mentors and leads by example other team members within their channel, guiding them in understanding and contributing to architectural discussions. Proactively identifies opportunities to improve existing architectures within their channel and drives initiatives to implement changes. Influences architectural decisions beyond their immediate team, and channel, collaborating with other teams and departments to ensure alignment with organizational goals and standards. Publishes articles or gives talks within the company on architectural approaches developed within their division.

Category: Engineering Ability > Maintainability
Behavior: Fluency in Unit Testing

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior

Emerging (10%-40%)	Writes unit tests as part of delivery
Practicing (40%-90%)	Confident in mocking, spying and stubbing in unit tests but may not always write with the appropriate scope / isolation
Consistent (90% +)	Demonstrates understanding of tradeoffs in unit testing, applies appropriate scope and isolation, writes tests first when appropriate
Radiating	Drives unit testing culture within their respective team, project or domain

Category: Engineering Ability > Maintainability
Behavior: Understands the tradeoffs between testing shapes (e.g. Pyramid, Trophy, Honeycomb) as well as how to apply them.

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Knows there are different testing shapes and what makes up a shape (Unit, Integration, Acceptance, End-to-End)
Practicing (40%-90%)	Starts to recognize that different types of tests have their tradeoffs (maintainability, how fast they run, ease to write), and explores the tradeoffs of another shape.
Consistent (90% +)	Understands types of tests, the different shapes and when to apply each type of test or shape, drawing appropriate boundaries for tests in an application.
Radiating	Influences others to consider testing breakdown, approach and tradeoffs

Category: Engineering Ability > Maintainability
Behavior: Considers Readability of Code (Understands the audience of their code is other Engineers not the computer)

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Beginning to see that their code has a lifecycle. Considers others as they name things.
Practicing (40%-90%)	Cares that others can understand their solution. Favors simplicity over complexity. Writing code for the team, not for themselves. Thinks about how their solutions might appear to a new team member. Demonstrated ability to create code that can be iterated upon easily. Favors known patterns over new, clever techniques. When others struggle to understand their code, they consider this as feedback to improve the maintainability of their work, not a judgement of the competence of the other person.
Consistent (90% +)	Demonstrated ability to create “clean code” that others can understand quickly. Considers the life of their solution compared to the life of the problem, actively makes decisions considering how permanent or fleeting the solution is. Actively seeks feedback from fresh eyes, incorporating feedback to create clear code.
Radiating	Influences others to actively manage the complexity of their work, favoring simpler solutions. Demonstrated ability to shape solutions that can be modified, extended, and even outgrown. Celebrates when their solution can be superseded and replaced by something new. This person’s team is known for moving quickly, partly because they aren’t hampered by technical debt that slows them down.

Category: Engineering Ability > Maintainability
Behavior: Consider unit tests for all code, implement, and maintain appropriately

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Occasionally considers writing unit tests for new code but may not due to lack of skill or for the sake of short term

	speed. They are updating existing unit tests for code they are changing, when they break.
Practicing (40%-90%)	More often than not, they are writing unit tests for new code and starting to understand the long term speed gain. They are updating existing unit tests for code they are changing, regardless of if they break or not. They are starting to examine existing unit tests outside the scope of their change to understand how code is expected to work.
Consistent (90% +)	They consistently weigh the tradeoffs of writing unit tests for all new code they write and understand when to write unit tests and when not to. They are starting to encourage others to write unit tests for new code and starting to mentor them. They are starting to use tooling to point them at holes in their code coverage.
Radiating	They consistently demonstrate an ability to determine when unit tests are appropriate and consider them in the scope of all the other testing methods. They are mentoring others in when to write unit tests, how to implement them, and maintain them. They always use tooling to find holes in code coverage and fill them in.

Category: Engineering Ability > Maintainability
Behavior: Consider integration points and avoid fragility

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Is aware of defensive vs offensive programming. Is aware of the concept of preconditions, postconditions and invariants. Can identify code that crosses the application’s code

	boundary.
Practicing (40%-90%)	With guidance: - Applies defensive and offensive techniques. - Isolates integration points to the fringes of the application.
Consistent (90% +)	Consistently applies the aforementioned techniques without guidance. Has sufficient experience to teach these techniques.
Radiating	Consistently teaches and applies these techniques. Has demonstrated, taught and mentored others sufficiently that others apply this from their learned experience.

Category: Engineering Ability > Maintainability
Behavior: Consider SEO impact

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Understands that SEO is an important part of software delivery. Is able to research and learn code level concerns. Needs help identifying what is important.
Practicing (40%-90%)	Knows the benefit of SEO towards business goals. Partners with non-technology team members to identify and implement solutions to SEO-related problems. Starting to think beyond one-time solutions.
Consistent (90% +)	Proactively considers SEO as a non-functional attribute in their work. Can research and implement SEO solutions that serve business goals. Systematizes solutions to make the right thing the easy thing.
Radiating	Considers and empowers SEO efforts as second nature. Coaches technical and non-technical team members on the importance of SEO. Thinks globally when systematizing solutions.

Category: Engineering Ability > Mentoring
Behavior: Mentoring other Engineers in areas you are strong in

	Example

Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team member is starting to share concepts and techniques they are learning with other members on their team. They are pursuing how to teach others. They may need to be told to share some knowledge they have (not being proactive in their sharing of knowledge).
Practicing (40%-90%)	Team member can clearly articulate concepts and techniques to new team members. They are recognizing that other team members have different learning styles are pursuing how to be effective in their explanation of a topic (balancing conciseness with completeness/thoroughness).
Consistent (90% +)	Team member is constantly mentoring and other team members leaves nearly every interaction having learned something. The team member recognizes that mentoring is a two-way street and wants to teach, coach, and guide so they can get better in the practice of mentoring and learn in the process of mentoring. The team member recognizes that the most effective mentoring is through asking questions and engaging the other team member in critical thinking. They are pursuing the practice of mentoring mentors in how to mentor.
Radiating	Team member can mentor mentors and is being sought out across the company to teach, coach, and guide in the areas they have deep expertise. Other team members come away from nearly every interaction knowing more about a particular topic. Team member is skilled in mentoring and teaching others in how to mentor and teach others in the team member’s areas of expertise. Other team members leave interactions feeling like they learned this topic on their own (because of the questions, critical thinking, and learning that the mentor uses to engage and teach the mentee).

Category: Engineering Ability > Mentoring
Behavior: Speaking the language of engineering

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The team member is starting to speak the language of engineering by actively engaging in technical discussions with

	colleagues. For example, they may ask clarifying questions during team meetings to better understand technical concepts and terminology. They begin to use technical terms appropriately but may still rely on explanations from others.
Practicing (40%-90%)	The team member consistently speaks the language of engineering by confidently using technical terms and concepts in discussions and presentations. For instance, they effectively communicate complex technical ideas to both technical and non-technical stakeholders, demonstrating a clear understanding of engineering principles. They actively seek out opportunities to learn and expand their technical vocabulary.
Consistent (90% +)	The team member can explain complex technical concepts in simple terms and tailor their communication style to different audiences. For example, they mentor junior engineers by providing clear explanations and guidance, contributing to technical documentation, and participating in technical reviews with confidence.
Radiating	The Team Member is known for their understanding of engineering concepts. They shape the way the business talks about technology in both engineering and non-engineering contexts.

Category: Engineering Ability > Mentoring
Behavior: Finding technology industry trends and bringing the knowledge back to the team

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team Member occasionally reads articles or blogs about technology trends and infrequently shares industry insights with other members on their team. Team Member may occasionally attend webinars or conferences related to their technology stack. Team Member participates in discussions about technology trends when prompted.
Practicing (40%-90%)	Team Member regularly reads industry publications and blogs and shares relevant industry insights with the team during meetings. Team Member attends industry conferences or webinars regularly. Team Member collaborates with team

	members to explore the potential impact of trends on current projects.
Consistent (90% +)	Team Member demonstrates a deep understanding of technology trends and regularly presents industry findings to the team to inspire innovation. Team Member actively contributes to team discussions with industry insights and encourages other team members to stay updated.
Radiating	Team Member is considered a thought leader in the industry. Team Member initiates cross-team knowledge-sharing sessions and proactively suggests and leads the adoption of new technologies. Team Member inspires and mentors others within and outside the team to stay at the forefront of industry developments.

Category: Engineering Ability > Mentoring
Behavior: Engineering is a craft to hone, not a skill to use. (You have seen someone do something better than you and you want to grow yourself the same way. Pursuing excellence and expertise)

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	You have seen someone do something better than you and you want to grow yourself the same way.
Practicing (40%-90%)	Pursuing excellence and Expertise.
Consistent (90% +)	
Radiating	

Category: Engineering Ability > Mentoring
Behavior: Growth Mindset: Share new craft growth with Engineers around you

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	You are having conversations with other engineers to talk about the craft of Engineering. What tools are even available? How are they used?

	Engaging in a chapter
Practicing (40%-90%)	You discuss different tools within your toolbelt that can be used to solve problems and are aware of the weaknesses and strengths of the various tools available to you. Presenting at chapters
Consistent (90% +)	
Radiating	

Category: Engineering Ability > Core Engineering
Behavior: Evaluate technical skills of candidates through the interview process

	Example
Unpracticed (0%-10%)	Team Members are not expected to participate in technical interviews
Emerging (10%-40%)	Team Member has completed the training process (in Paycom) and has worked through the shadowing process and has received approval to start driving technical interviews for Software Engineer 1 and Software Engineer 2.
Practicing (40%-90%)	Team Member starting to do consistent interviews, starting to learn how to ask the right questions that result in skill ratings that match the candidate's actual level 70% of the time.
Consistent (90% +)	Team Member can conduct interviews that feel like you are having a natural conversation by mastering how to ask the right questions and driving a deeper level of questioning that result in skill ratings that match the candidate's actual level 90% of the time.
Radiating	Through their approach to interviewing, Team Members drive overall improvements to the Ramsey Technical interview process.

Category: Engineering Ability > Monitoring
Behavior: Owns alerts and response (Can diagnose an application issue without reaching out to Systems Engineering)

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team Member is aware of alerts and seeking to understand them but not acting on them. Team member is aware of team responding to alerts but not involved in responding or may be an observer in to the team responding.
Practicing (40%-90%)	Team Member is acting on alerts within their area of expertise and can address some alerts without help. Team member is

	getting involved with team when responding to alerts. Team member is learning how to adjust alerts and create new alerts. Team member is in oncall rotation but not expected to resolve alerts on their own.
Consistent (90% +)	Team Member is consistently acting on alerts and can resolve some without help. Team member can adjust alerts and create new ones. Team Member is consistently involved in responding to all alerts. Team Member is part of oncall rotation and expected to drive to resolution of alerts with our without help from the team.
Radiating	Team Member is able and willing to address and resolve alerts without the team, thereby protecting the teams time and focus. Team member helps team form and adjust the process of responding and resolving alerts. Team member is able / willing to help others respond and resolve alerts.

Category: Engineering Ability > Monitoring
Behavior: Log appropriate code diagnostic information

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Knows how to use the logging libraries in their applications. Knows the difference between logging, printing to the console and exception messages and stack traces.
Practicing (40%-90%)	Can configure logging libraries in their tech stack. With guidance, writes logs with precise information and identifiers that helps trace and diagnose paths of interest.
Consistent (90% +)	Continuously improves diagnostics and traceability by updating log calls with missing context or more precise information.
Radiating	Helps peers identify code that benefits from logging. Teaches best practices with logging, diagnostics and traceability. Others apply these learnings to their work on their own initiative because of consistent examples and guidance from this Team Member.

Higher Level Thinking

Category: Higher Level Thinking > Business Acumen
Behavior: Can relate work being executed to a business problem

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Beginning to connect the dots between the technical solution and the business outcomes they are working on.
Practicing (40%-90%)	When asked what they are working on, often starts by explaining the business outcome they are trying to achieve.
Consistent (90% +)	Actively engages with business and talked through not only the main business outcomes, but also works towards business uncommon scenarios that may occur while attempting to solve the current business problem.
Radiating	Actively seeks out not only current business problems but is also interested in where the business is going. Then working towards a solution to the current business problem while keeping the future business direction in mind.

Category: Higher Level Thinking > Business Acumen
Behavior: Can break down work using business context even when ambiguous

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Individuals at the emerging level can identify the high-level objectives of a project or feature.
Practicing (40%-90%)	Individuals at the practicing level can break down the work and tasks required for a project or feature.
Consistent (90% +)	Individuals at the consistent level demonstrate expertise in dissecting work within a business context and breaking it down into manageable tasks that the team can execute. They excel at identifying the interconnections within the systems they oversee and uncovering potential complexities that may not have been initially recognized in the features.

Radiating	Individuals at the radiating level are considered experts in breaking down work while taking into account both the direct business context and the broader business context that extends beyond Ramsey. They create possible solutions and architectures that address the core needs while managing engineering and architectural trade-offs. With these solutions, engineers make recommendations, propose decisions, and provide context and reasons, both for and against, from a technical perspective connected to the business context. They also ask clarifying questions to uncover the final 10% of the business context truth and share it with the team. They deliver iterations rapidly, seeking feedback and striving for value creation. Furthermore, they efficiently identify the tasks required to achieve a goal and proactively escalate risks to stakeholders and leadership.

Category: Higher Level Thinking > Business Acumen
Behavior: Actively leading initiatives for your team

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team Member consistently owns a small task given by leader or other team member
Practicing (40%-90%)	Team Member consistently owns a more complex task given by leader and brings a few other into it. The team member provides regular updates to leadership
Consistent (90% +)	Team Member consistently seeks out initiatives that solve problems for their leader and finds the right people to partner with. The team member provides regular updates to leadership. The team member appropriate balances this work with their current duties.
Radiating	Team Member consistently seeks out initiatives that solves problems for their channel and finds the right people to partner with. The team member provides regular updates to the appropriate stakeholders. The team member appropriate balances this work with their current duties.

Category: Higher Level Thinking > Business Acumen

Behavior: Sitting with other business functions (to learn from their experiences and the problems they run into)

	Example
Unpracticed (0%-10%)	Team Member has little or no interest in other business functions.
Emerging (10%-40%)	Team Member is curious what other business functions are doing across the company. This causes them to periodically ask questions of members of other business functions.
Practicing (40%-90%)	Team Member is sometimes monitoring and seeking out what other business functions are doing across the company. Team member sometimes meets with other business functions.
Consistent (90% +)	Team Member is actively monitoring and seeking out what other business functions are doing across the company. Team member often meets with other business functions.
Radiating	Team Member is actively monitoring and seeking out what other business functions are doing across the company. Team Member also actively informs other businesses when they have solved a problem that is shared across business functions.

Category: Higher Level Thinking > Business Acumen
Behavior: Measuring the impact of your code quality on customer rating, crash rates, bug rates, customer success issue counts, time to market

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team Member is aware that code quality has an impact on customer rating, crash rates, bug rates, customer success issue counts, time to market. Team Member is working on understanding the impact and looking into data that shows the impact (code analysis tooling and reports, customer ratings, APM data, workflow data). Team Member is starting to make changes in their own work that improves code quality.

Practicing (40%-90%)	Team Member is starting to monitor code quality impact on customer rating, crash rates, bug rates, customer success issue counts, time to market. Team Member understands the impact of code quality and data that shows the impact (code analysis tooling and reports, customer ratings, APM data, workflow data). Team Member is making changes in their own work that improves code quality and reviewing others work for code quality issues.
Consistent (90% +)	Team Member is actively monitoring code quality impact on customer rating, crash rates, bug rates, customer success issue counts, time to market. Team Member brings code quality issues and data that shows the impact (code analysis tooling and reports, customer ratings, APM data, workflow data) to the team and is starting to create standards to prevent issues. Team Member is making changes in their own work that improves code quality, reviewing others work for code quality issues and verifying the changes impact on code quality. Team Member is starting to express risk around code quality issues to stakeholders.
Radiating	Team Member is actively monitoring code quality impact on customer rating, crash rates, bug rates, customer success issue counts, time to market and adjusting tooling to better monitor code quality issues. Team Member brings code quality issues and data that shows the impact (code analysis tooling and reports, customer ratings, APM data, workflow data) to the team. Team Member is making changes in their own work that improves code quality, reviewing others work for code quality issues, verifying the changes impact on code quality and introducing standards to prevent issues. Team Member is expressing risk around code quality issues to stakeholders.

Category: Higher Level Thinking > Business Acumen
Behavior: Technical writing can be written with both technical and business stakeholders in mind

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team Member is starting to write technical documents that are intended for technical stakeholders.
Practicing (40%-90%)	Team Member writes technical documents that are intended for technical stakeholders and is starting to write in terms that business stakeholders use.
Consistent (90% +)	Team Member writes technical documents that are intended for technical stakeholders in terms that business stakeholders use.
Radiating	Team Member writes technical documents that are intended for technical stakeholders, business stakeholders, and shared channel or Ramsey wide.

Category: Higher Level Thinking > Business Acumen
Behavior: Creates monitoring dashboards to track the health of the code deployed

	Example
Unpracticed (0%-10%)	Team Member has little or no interest in monitoring. Team Member does not actively seek out anything that does come from monitoring.
Emerging (10%-40%)	Team Member has set up basic monitoring and is periodically checking monitoring. Team Member has personal alerting if certain errors occur.
Practicing (40%-90%)	Team Member has set up monitoring and is regularly checking monitoring. Team Member has alerting some of the most important metrics.
Consistent (90% +)	Team Member has monitoring in a visible place. Team Member also has alerting thresholds for many metrics. Team Member actively monitors it daily.

Radiating	Team Member has monitoring in a visible place. Team Member actively monitors it daily. Team Member also has alerting thresholds for every important metric. Team Member also has monitoring of testing as well from testing strategies.

Category: Higher Level Thinking > Business Acumen
Behavior: Has enough business acumen to sit in business strategy meetings

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The team member is starting to develop an understanding of basic business concepts and is beginning to grasp how their work contributes to broader business goals. They may attend strategy meetings as an observer and seek to learn from more experienced colleagues.
Practicing (40%-90%)	The team member consistently shows an understanding of business objectives and can contribute to discussions about how their work aligns with these goals. They actively participate in business strategy meetings, offering insights and asking relevant questions to deepen their understanding.
Consistent (90% +)	The team member consistently integrates business acumen into their decision-making processes. They proactively contribute to business strategy meetings, providing valuable input on how technical solutions can drive business success and collaborating with business leaders to align technical and strategic goals.
Radiating	The team member is highly skilled in business acumen and significantly impacts business strategy. They lead initiatives that drive business strategy and challenge other engineers to consider business value in their work. They are often sought after for their expertise and regularly present strategic insights to senior leadership.

Category: Higher Level Thinking > Business Acumen
Behavior: Considers [and can explain the impact to] Global/EA while making technical decisions

	Example

Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	
Practicing (40%-90%)	
Consistent (90% +)	
Radiating	

Category: Higher Level Thinking > Business Acumen
Behavior: Partners with leadership to make the best technical decision for the business

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The team member is starting to understand the importance of aligning technical decisions with business goals. They occasionally engage with leadership but often need guidance to fully comprehend the broader business implications of technical choices.
Practicing (40%-90%)	The team member regularly collaborates with leadership to align technical decisions with business objectives. They actively participate in discussions, offer informed suggestions, and seek feedback to ensure their technical contributions support the business strategy.
Consistent (90% +)	The team member consistently partners with leadership to make informed technical decisions that drive business success. They provide strategic insights, anticipate potential business impacts of technical choices, and ensure their recommendations are aligned with the company's goals.
Radiating	The team member excels at partnering with leadership to make optimal technical decisions for the business. They lead strategic initiatives and are trusted advisors to senior leadership. Their decisions not only drive innovation but consistently enhance business outcomes.

Category: Higher Level Thinking > Business Acumen
Behavior: Creates clarity from ambiguity

	Example

Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team Member can read and understand the work described in a ticket and fill in the gaps where the ticket/task may not have full detail of the desired implementation. Team Member is beginning to communicate technical concepts to stakeholders in a way to create clarity on desired outcomes.
Practicing (40%-90%)	Team Member is able to participate in business conversations and with some help and guidance translate the desired outcome into clear work for the team. Team Member helps stakeholders understand the tradeoffs with desired outcomes to ensure clear understanding of problem.
Consistent (90% +)	Team Member contributes to business conversations and is able to successfully communicate complex technical concepts in a way that helps stakeholders get clear on the problem domain to make informed decisions. Team Member is successful at communicating desired outcome to the team to drive towards creating business value.
Radiating	Team Member is leading conversations and discussing tradeoffs of desired outcomes with both stakeholders and team to create and drive business value. Team Member ensures that clarity is created for the desired outcome with all stakeholders.

Category: Higher Level Thinking > Business Acumen
Behavior: Discovers, introduces, and champions innovation while considering total cost of ownership

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The team member is beginning to recognize opportunities for innovation and understands the importance of considering the cost of ownership. They help with the adoption of new technologies and processes.
Practicing (40%-90%)	The team member is able to speak about the cost of ownership for several familiar technologies and can communicate tradeoffs to leadership. They champion global technology initiatives within their team.

Consistent (90% +)	The team member is always looking for ways we can improve our technology, and they have the skills to market those solutions to others. They consider the disadvantages that come with supporting another technology or framework.
Radiating	The team member excels at driving innovation while considering the total cost of ownership. They are a thought leader, regularly introducing transformative ideas that significantly enhance business operations. They mentor others on evaluating and championing innovation, and their strategic insights lead to substantial cost savings and efficiency gains for the organization.

Category: Higher Level Thinking > Workflow Management
Behavior: Connected to the flow of work of the team (metrics, decomposition, engaged in rhythms)

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The team member is beginning to understand the team's workflow. They understand some key metrics, and they participate in the squad’s cadences. They often listen more than contribute and are learning the rhythms of team processes.
Practicing (40%-90%)	The team member contributes to decomp, highlights blockers in standups, and makes realistic commitments during planning sessions. They provide valuable input during retrospectives and work on incorporating feedback into future cycles.
Consistent (90% +)	The team member is comfortable facilitating any cadence. They take a more active role in decomposing tickets. They understand the importance of collaboration and make efforts to minimize WIP and blockers on their team.
Radiating	The team member has a reputation for collaborating well with others. They can facilitate any cadence with excellence, and they foster a culture of continuous improvement within their team. Others go to this person for advice around improving their team’s productivity, and they help shape standard business practices.

Category: Higher Level Thinking > Workflow Management
Behavior: Pull cards and work them with self-sufficiency and knows how to get unblocked

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team member is able to unblock themselves from some issues. Before reaching out for help, the team member has tried to unblock themselves and can articulate what has been tried and where they are stuck.
Practicing (40%-90%)	Team member usually completes work without needing support, but occasionally needs help with challenging issues/blockers. When support is needed, the team member should be able to discuss in depth what has been tried and what potential solution options are good candidates.
Consistent (90% +)	Team member is almost always self-sufficient in their work. The team member is also starting to help to unblock others on their squad or adjacent squads.
Radiating	Team member is completely self-sufficient in their work. The team member is known on their squad for removing blockers and providing clarity, often guiding/coaching team members to clear their own blockers. The team member consistently helps other engineers navigate and troubleshoot issues within their squad, business unit, and all spheres they touch.

Category: Higher Level Thinking > Workflow Management
Behavior: Uncovers and surfaces opportunities for technical improvement

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team Member takes ownership of making code improvements and upgrading to newer versions of libraries within projects that they own. At this level they will likely need guidance on the business implications and value of making these improvements.

Practicing (40%-90%)	Team Member fills comfortable updating software dependencies and is begging to evaluate completely new technologies to replace existing capabilities. They are able to evaluate the pros/cons of new solutions and can formulate arguments for a business value of these improvements.
Consistent (90% +)	Team Member is skilled in adopting technical improvements in a seamless way and is evangelizing those improvements to other teams within the business unit and other product teams.
Radiating	Team Member is analyzing the impact of significant technical improvements across all of Ramsey and is focused on industry best practices. Their primary focus of making technical improvements is to drive business objectives and operational excellence across the company. These technical improvements are not incremental improvements but tactical shifts which produce technological “leaps” for all of Ramsey.

Category: Higher Level Thinking > Workflow Management
Behavior: Successful creating work that spans multiple workstreams. Beginning to blend and coordinate activities while creating the work

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	The team member recognizes work that requires handoffs between people and are learning to coordinate with relevant groups before the work starts.
Practicing (40%-90%)	The team member regularly creates and coordinates work that spans multiple workstreams. They effectively blend activities, ensuring tasks are aligned across teams, and contribute to seamless collaboration. They take the initiative to communicate and synchronize efforts with other teams.
Consistent (90% +)	The team member consistently ensures that work spanning multiple workstreams is well-coordinated. They communicate timelines with relevant stakeholders in advance. They facilitate cross-team standups and retrospectives when necessary to accomplish business goals.
Radiating	The team member is highly respected for their expertise in coordinating work across multiple workstreams. They

	implement best practices that improve handoffs, and those learnings are applied to other similar partnerships across the company. They play a pivotal role in strategic initiatives, ensuring that efforts align with each team's goals and contributes to overall business success.

Category: Higher Level Thinking > Workflow Management
Behavior: Understands that tickets don't stop when the merge button is pressed but consider up and downstream systems affected by the code created

	Example
Unpracticed (0%-10%)	Team Member is not demonstrating this behavior
Emerging (10%-40%)	Team Member understands change has impact and works with the team to identify the impact and the proper procedure to mitigate and/or communicate that impact to the necessary stakeholders.
Practicing (40%-90%)	Team Member has learned how to use tools to identify related systems (i.e. APM, Logs, etc.). Changes and their impact are documented and communicated.
Consistent (90% +)	Team Member identifies related systems affected by change, communicates and coordinates with all stakeholders, and performs post-deployment validations to ensure maximum stability. Those related systems are monitored for impact when a change is rolled out and impact to those systems is communicated.
Radiating	Team Member identifies areas of missed awareness of change on related systems. Communicates with others with the heart of a teacher to ensure more global understanding of our interconnected systems. Doesn't walk away from code once it's merged but continues to monitor, maintain, and own until retirement.