

Engineering 360's

Engineering Cohort Section

In this section highlight the level that best reflects the engineer's performance. i) is associate engineer level, ii) is intermediate engineer, and iii) is a senior engineer. Yellow signifies working on but not mastered; while green can indicate mastery of that level.

Team work:

1. As part of the team all engineers are expected to participate in Agile Methodologies and should be active participants in all agile ceremonies.
 - a. Standup
 - i. Communicates the status of work effectively. Updates tickets with relative investigation work and updates.
 - ii. Provides accurate time updates. Consistently communicates and solves blockers on tasks. Does what is necessary to meet commitments.
 - iii. Provides guidance and help to other team members as issues arise. Takes proactive measures to help other team members when sprint goals are in danger.
 - b. Retro
 - i. Provides feedback and action items
 - ii. Holds team accountable for action items, process changes, and improvements discussed in retro/as a team
 - iii. Holds leadership accountable for the performance of team (e.g. pushing on team leads to have clear and thought out release plans, pushing on EMs to adjust process if needed)
 - c. Sprint planning
 - i. Participates in sizing. Actively asks questions to understand the systems and solutions being discussed.
 - ii. Provides meaningful estimations and guidance on tickets. Will provide broad solutioning ideas and resources that may be relevant.
 - iii. Breaks down large tickets into smaller consumable ones. Can provide very specific guidance and historical context. Calls out risks and blockers ahead of time. Will push back on unclear requirements or descriptions.
 - d. Ticket grooming
 - i. Identifies & surfaces missing or unclear requirements and Acceptance Criteria
 - ii. Surfaces suggestions for filling in requirements.
 - iii. Proactively surfaces issues/gaps as part of planning or technical design of project
 - e. Promotes an environment of learning and comradery. Actively seeks out ways to improve process/culture to foster this environment.
 - i. provides feedback on existing processes and suggestions on how to improve them. Pairs with other members to move suggestions forward.
 - ii. Improves on existing processes. Provides automation of existing manual processes. Improves painful workflows, local builds, testing with automation and documentation. Implements new technologies where appropriate, i.e. Github Actions, tracers, alerts. Improves
 - iii. Provides coaching and mentorship on agile processes. Follows team and organization established practices. Creates opportunities for self and others to try new things / learn. Creates tools to measure the success and journey of the team.
2. Teaching & Learning.
 - a. Documentation
 - i. Engineer is able to utilize existing documentation to work within complicated systems. They also contribute to updating out of date documentation.
 - ii. Engineer creates new documentation on existing and new workflows.
 - iii. Thorough documentation (system design diagrams, architecture, large scale technical design).

b. Learning

- i. Focuses on the immediate technologies used within the application. Node, js, azure, confluence, graphql, contentful, plytex, elasta etc... Utilizes Udemy and other resources to elevate knowledge.
- ii. Begins to learn more advanced programming techniques and tertiary technologies. Kubernetes, deployments, Circle CI, and other technologies focused around CI/CD.
- iii. Researches new practices and technologies to improve the project's performance, stability, or cost. I.e. Redis, Spot Instance, pod resources.

c. Teaching

- i. Supports experimentation as a way to find the best solution.
- ii. Engineer leaves meaningful comments and critiques on code reviews. Will provide alternate solutions and leave probing questions to encourage thinking of the whole system.
- iii. Engineer actively pairs with teammates when teammates are undertaking work they are SME. The scope of their knowledge transfer reaches to outside the team as well. I.e. Brown Bag sessions.

Engineering Excellence: Application of modern techniques to plan, design, develop, test, maintain and operationalize software and systems.

1. Code

a. Time management.

- i. Prioritizes and manages own work to ensure effective and timely completion of tasks.
- ii. Volunteers to assist and support others when needed.
- iii. Ensures the delivery of the team's sprint. Early intervention on tickets that appear to be blocked or needs guidance or resources.

b. Execution.

- i. Consistently delivers 2-4 small (1,2 pts), well documented tasks. Engineer delivers 1 and 2 point stories without much guidance, and 3 point stories with guidance.
- ii. Consistently delivers 2-4 medium tasks (1,2,3 pts). Engineer delivers 1, 2, and 3 point stories without guidance. Engineer gathers any missing requirements, missing acceptance criteria, and 5 point stories with guidance.
- iii. Consistently delivers 2-4 large tasks (1,2,3,5 pts). Engineer delivers 1, 2, and 3 point stories without much guidance, and 5 point stories with guidance.

c. Quality.

- i. Follows coding standards, learns from code review comments, and minimal back and forth on pr's before merger.
- ii. Code not only solves the problem, but also attempts to refactor for readability and testability.
- iii. Thorough test plans; able to write comprehensive unit/integration tests.

2. Operational Excellence

a. Monitoring:

- i. Engineer is able to find, read, and interpret important graphs and metrics.
- ii. Engineer adds new monitoring to existing workflows.
- iii. Engineer adds new technologies and monitoring to new workflows. Includes monitoring in their solutions.

b. Incident response

- i. Engineer should be able to read logs and metrics and provide support to active incidents.
- ii. Be able to proactively monitor and be the point-person responding to incidents. Should be able to escalate appropriately, understand priority and be able to execute under pressure.
- iii. Should be able to monitor, keep stake holders updated and solve the incident.

3. System Architecture

a. Technical Design

- i. Participates in technical design reviews

- ii. Contributes to parts of technical design for a component
- iii. Lists and understands the major components and concerns of technical design (requirements, database schema, class diagrams, component diagrams, apis, rollout, monitoring, work breakdown, etc.) Defines and documents major design alternatives and design considerations, recommending preferred options Estimates the cost (development effort/cloud spend) of a specific design of a proposed product

4. Technologies:

a. Core Technology skills:

- i. Has a workable knowledge of the technologies and best practices used within the optum store. i.e. node, react, mysql, graphql, elasticsearch
- ii. Has an in-depth knowledge of the technologies and best practices. Possess a workable knowledge of our support infrastructure. i.e. kubernetes, Circle Ci, dev-ops, GCP, redis, elasta etc.
- iii. Posses in-depth knowledge of all technologies used within the optum store. Is able to incorporate new technologies to improve the platform.

Open Feedback section:

1.Quality of deliverables: Accuracy, consistency, follow-through, meets deadlines, keeps trying to work smarter, not harder.

2.Teamwork: Collaborates with internal and external teams across positions, proactively helps others, puts team targets higher than personal achievements, contributes and takes initiative to social activities

3.Innovation: Propose/use digital technologies to change internal processes, generates ideas and gives input that makes our product more efficient

4.Professional working attitude: Help create a good atmosphere and working environment, accepts constructive criticism, follows standards.

5.Job knowledge: Create, maintain, audit, and improve systems to meet particular needs. Continuously update new coding knowledge.

6.Fosters team environment where members can self-organize and contribute independently. Supports experimentation as a way to find the best solution. Fosters a team environment that encourages and rewards learning. Keeps abreast of industry standards and new tech development.

7. Accountability: Holds self accountable by following through on commitments. Takes responsibility for successes and failures in work. Monitors sprint and team progress against goals. Incorporates feedback in their work. Learns from mistakes and adjusts accordingly. Shares lessons learned from successes & failures across the team and platform.

Feedback for Leadership:

1. Career Development in the long-term:

- a. Reflections on job opportunities, other tasks, new disciplines, etc. Do you have adequate support from your manager to achieve your development goals?

2. Work-life balance:

- a. How do you feel about your work life balance?

3. Shared vision:

- a. Do you feel aligned with the goals of the company

4. Transparency:

- a. You are aware of how your contributions lead to the companies success.
- b. You are aware of the state of the company and the reasons behind decisions leadership makes.

5. Was this helpful? Please leave feedback on questions and levels that you liked and what you didn't like. Point out anything you feel redundant. Did you like the format? Were there elements you feel are important that were not discussed.

Company Engineering Values:

