



30-60-90 Day Performance Roadmap: Software Engineer

Date: _____ Employee Name: _____

- **Role:** Software Engineer
- **Manager:** Engineering Manager
- **Goal:** Transition from "Learning the Stack" to "Shipping Features Independently."

Day 1–30: The Learner (Information Intake & Setup)

Environment Mastery & Small Wins

The focus of the first month is to become comfortable with the DevCore toolset, internal processes, and shipping the first lines of code to a non-production environment.

Key Milestones (Aligned to Week 1 Agenda)

- **Environment:** Fully configured local dev environment (IDE, Docker, VPN) with no friction.
- **Process:** Successfully attend all Sprint ceremonies (Stand-ups, Retros, Planning) and actively listen.
- **Code:** Complete 3-5 "Good First Issue" tickets (low-risk bug fixes or text changes).
- **Workflow:** Successfully clone a repo, create a branch, commit code, and open a Pull Request (PR) following the "Code Review Etiquette" standards learned on Thursday.
- **Compliance:** Complete 100% of The Company University compliance and security training in LMS.

Assessment Strategy (Day 30 Check-in)

The "First Ship" Review: Manager reviews the engineer's first merged Pull Request.

- **Criteria:** Did they follow the naming convention? Did they write a clear PR description? Did the build pass CI checks?

Knowledge Check: Informal Q&A on the architecture overview covered on Day 1 (e.g., "Explain the relationship between our API gateway and the database").



Day 31–60: The Contributor (Growing Independence)

Collaboration & Quality Assurance

In the second month, the engineer shifts from "fixing bugs" to "building components." The focus is on the Thursday agenda topics: Testing Strategies and Peer Review.

Key Milestones:

- **Development:** Pick up and complete a medium-complexity feature ticket within a single Sprint.
- **Quality:** Write unit tests for all new code submitted, achieving >80% coverage (applying the "Testing Frameworks" training).
- **Review:** Begin providing feedback on peer Pull Requests (focusing on style and logic errors).
- **Ops:** Successfully deploy code to the Staging environment and verify the fix without assistance.
- **Documentation:** Update the internal Wiki/Confluence for one outdated process discovered during onboarding.

Assessment Strategy (Day 60 Check-in)

- **Code Quality Audit:** Review of the last 3 Pull Requests.
 - **Criteria:** Are unit tests included? Is the code "clean" (DRY principles)? Are comments used effectively?
- **Velocity Check:** Is the engineer consistently completing assigned story points within the Sprint?



Day 61–90: The Owner (Autonomy & Optimization)

Autonomy & Production Readiness

By the end of the third month, the engineer should be operating as a fully productive member of the team, touching on the Friday agenda topics: Incident Management and Escalations.

Key Milestones

- **Development:** Own a feature from "Concept" to "Production Deployment."
- **Incident Management:** Shadow an "On-Call" rotation for one week (observing alerts and response).
- **Debugging:** Successfully troubleshoot and resolve a production bug using logs and monitoring tools.
- **Optimization:** Propose one refactor or tool improvement to speed up the team's workflow.
- **Culture:** Lead one segment of a Sprint Retrospective or Stand-up.

Assessment Strategy (Day 90 Graduation)

- **The "90-Day Project" Demo:** The engineer demos their "Month 3 Feature" to the product team.
 - **Criteria:** Does the feature work as intended? Can they explain the technical decisions made?
- **Probation Review:** Formal review against core competencies: Technical Skill, Communication, and Cultural Fit.

