

Team Technological Fitness for the Project (AT): The extent to which the team possesses the technical skills necessary to successfully execute the project.

AT (Technology Fit)	
Value	Description
VL	<p>POOR FIT</p> <p>Characteristics:</p> <p>Significant Lack of Skills: The team has little to no experience with the essential Core Technology, resulting in a critical lack of technical knowledge.</p> <p>High Need for External Support: The team cannot progress independently and requires continuous training or assistance to understand the project's fundamentals.</p> <p>Risk of Delays and Errors: The lack of technical skills creates a high risk of delays, frequent rework, and low-quality deliverables.</p> <p>Quality: All team members lack understanding of the criteria for quality and acceptance of the project.</p> <p>Example: In an Artificial Intelligence project, where the Core Technology is Machine Learning, the team has no experience with Python (the main language for AI) and has never used TensorFlow or PyTorch (key ML ecosystems). They only have limited knowledge in unrelated programming languages, like PHP, and depend heavily on external support to set up the environment and develop basic algorithms.</p>
L	<p>BELOW AVERAGE FIT</p> <p>Characteristics:</p> <p>Basic Knowledge with Significant Gaps: The team has some familiarity with the Core Technology, but lacks competence in essential areas, compromising its ability to deliver independently.</p> <p>Moderate Dependency on Support: Although they have a foundational understanding, the team still needs additional support to fill significant gaps and handle technical challenges.</p> <p>Limited Capacity to Solve Complex Problems: The team can handle basic tasks but struggles greatly when facing more advanced requirements.</p> <p>Quality: Some team members have a basic understanding of the quality criteria, but this knowledge is incomplete or inconsistently applied across the team. As a result, the team often misses key quality benchmarks, and their work may require frequent adjustments to meet the project's standards. There are gaps in the team's ability to produce work that consistently meets acceptance criteria.</p> <p>Example: In a Cloud Computing project, where the Core is Distributed Systems, the team has moderate experience with AWS (ecosystem) but no knowledge of languages like Go or JavaScript, commonly used for cloud microservices development. They have some experience with Microsoft Azure, but struggle to apply that knowledge to AWS. This lack of key language expertise results in a dependency on technical support for basic service configuration and implementation.</p>

M	<p>AVERAGE FIT</p> <p>Characteristics:</p> <p>Adequate but Incomplete Technical Foundation: The team covers key technologies associated with the Core Technology but lacks competence in specific areas that may pose challenges.</p> <p>Moderate Progression Capability: The team can make progress in most of the project but may need occasional support in areas where skills are still developing.</p> <p>Moderate Risk of Rework and Delays: While the team has a sufficient technical base, missing critical skills can lead to delays and rework.</p> <p>Quality: Most team members have a good understanding of the project's quality criteria and acceptance standards. The team can generally deliver work that meets the required quality levels, but there are occasional oversights or inconsistencies. While quality is a priority, some improvement in adherence to standards is needed to ensure consistent output.</p> <p>Example: In an Internet of Things (IoT) project, where the Core Technology is IoT, the team has solid experience with Python (language) and has worked with Arduino devices (ecosystem), but lacks familiarity with MQTT, a common IoT protocol. They make progress on the project but face communication challenges between devices. Their knowledge of Raspberry Pi aids them, yet the lack of expertise in IoT communication protocols results in occasional delays.</p>
H	<p>ABOVE AVERAGE FIT</p> <p>Characteristics:</p> <p>Mastery of Most Required Technologies: The team covers the Core Technology and most complementary technologies, with only minor gaps.</p> <p>High Technical Capacity and Problem-Solving Skills: The team is self-sufficient in most aspects and can efficiently solve complex problems with minimal support.</p> <p>Minimal Need for External Support: The team is able to deliver the project effectively, quickly learning to fill any small gaps.</p> <p>Quality: The team has a strong grasp of the quality and acceptance criteria for the project, and they apply this understanding consistently. Deliverables are typically of high quality, meeting or exceeding expectations. Any deviations from quality standards are minor and addressed quickly. The team's work is closely aligned with project goals and accepted with minimal revisions.</p> <p>Example: In a Real-Time Systems project, where the Core Technology is Embedded Systems, the team is experienced with C++ (language) and regularly uses FreeRTOS (ecosystem). Although they haven't worked with VxWorks, a common real-time OS, they quickly adapt. Their familiarity with C and Ada helps them effectively implement components, with minimal risk of performance issues.</p>
VH	<p>PERFECT FIT</p> <p>Characteristics:</p> <p>Comprehensive Skill Coverage: The team is highly experienced in the Core Technology as well as in both the Ecosystem and Programming Language, allowing them to execute the project autonomously and effectively.</p> <p>Self-Sufficiency and Proactivity: The team applies best practices and proactively solves</p>

problems, anticipating technical challenges and resolving them efficiently.

Independent Execution Capability: The team delivers high-quality work while maintaining efficiency and minimizing the risk of rework.

Quality: All team members have a comprehensive understanding of the quality and acceptance criteria, and they consistently deliver work of the highest standard. The team proactively ensures that deliverables exceed expectations, often setting the bar for quality within the organization. Their work is fully aligned with project objectives, rarely requiring any rework or adjustments.

Example: In a Cloud Computing project, where the Core is Distributed Systems, the team is highly experienced with AWS (ecosystem) and Python and Go (programming languages), used for building and managing cloud microservices. They also have strong proficiency in Kubernetes for container orchestration, enhancing system scalability and reliability. This combination of skills covers all project needs, enabling agile, high-quality delivery with little to no external assistance required.