As the functional safety manager of autonomous vehicle manufacturer supplier, your company will be given a Software (IP protected, source not available for you) for inclusion in the autonomous driving functionality (your responsibility) what would you do? Which options does ISO26262 provide to help?

As the functional safety manager, when integrating an IP-protected software component, where the source is not available, into an autonomous driving functionality, your primary approach within the ISO 26262 framework would be to treat this as the integration of a safety-related system or component not developed according to the ISO 26262 series of standards. This scenario is specifically addressed in ISO 26262-8:2018, Clause 16.

Here's what you would do and the options ISO 26262 provides:

1. **Understand the Objective for Integration (ISO26262-8:2018 Clause 16.1):** The objective is to achieve confidence that the software component, though not developed according to ISO 26262, satisfies the required level of functional safety necessary for its integration into your ISO 26262-compliant item (ISO26262-8:2018 Clause 16.1 Objectives). This approach is particularly relevant for scenarios where a company following ISO 26262 integrates a system or component developed to another standard (ISO26262-8:2018 Clause 16.2 General).
2. **Ensure Prerequisites are Met (ISO26262-8:2018 Clause 16.3.1):** Before proceeding, you must ensure that an item definition for the system or array of systems on the integrator side that includes the third-party software is available in accordance with ISO 26262-3:2018, 5.5.1 (ISO26262-8:2018 Clause 16.3.1 Prerequisites).
3. **Develop a Safety Rationale (ISO26262-8:2018 Clause 16.4.2):** You will need to provide a clear rationale within your integrator safety case that justifies the application of this specific clause (ISO26262-8:2018 Clause 16.4.2). This rationale is a critical work product (ISO26262-8:2018 Clause 16.5 Work products; ISO26262-8:2018 Clause 16.5.1).
4. **Define Criteria for Functional Safety (ISO26262-8:2018 Clause 16.4.3):** Since the source code is unavailable, you cannot perform detailed code-level analysis or verification according to ISO 26262-6. Instead, the integrator shall define robust criteria to demonstrate that the safety-related software component, developed to another safety standard or internal process, meets the required level of functional safety (ISO26262-8:2018 Clause 16.4.3). These criteria could include:
    - **Process Assessment:** Evaluating the supplier's development process (e.g., if it adheres to another safety standard like IEC 61508 or ISO 13849, or a robust quality management system).
    - **Product Design Analysis:**Reviewing architectural design documentation, safety mechanisms, and fault tolerance strategies, even without source code access.
    - **Qualification Measures:** Requiring evidence of rigorous testing (e.g., black-box testing, robustness testing, environmental testing) and validation activities performed by the supplier (ISO26262-8:2018 Clause 16.4.4, EXAMPLE).
    - **Failure Rate Analysis:** Requesting data on observed failure rates or proven-in-use arguments, if applicable.
    - **Agreement on Safety Requirements:** Ensuring a clear understanding and agreement with the supplier on the safety requirements allocated to this software component.
5. **Agree on Verification Measures with the Supplier (ISO26262-8:2018 Clause 16.4.4):** You and the supplier shall mutually agree upon a relevant set of measures to verify that the defined criteria are met (ISO26262-8:2018 Clause 16.4.4). This could involve:
    - Review of documentation provided by the supplier (e.g., specifications, test reports, safety analyses like FMEA/FTA performed by the supplier).
    - Witnessing of tests or reviewing test results to confirm compliance with its requirements.

    - Analyzing field data if the software has been proven in use in other applications.
    - Ensuring the supplier's product release is based on an adequate approval process (e.g., Production Part Approval Process - PPAP).
6. **General Compliance and Tailoring (ISO26262-2:2018 Clause 4.2):** The overarching principle of ISO 26262 compliance allows for tailoring of safety activities or providing a rationale for non-compliance if it's acceptable and evaluated according to ISO 26262-2 (ISO26262-2:2018 Clause 4.2). The approach described in ISO 26262-8, Clause 16, is a specific instance of this general principle for integrating non-ISO 26262 developed components.

By following these steps, you can establish the necessary confidence in the functional safety of the IP-protected software component for its integration into your autonomous driving functionality, even without access to its source code.

**Additional List of Clauses:**

- ISO 26262-2:2018 Clause 4.2 General requirements
- ISO 26262-3:2018 Clause 5.5.1
- ISO 26262-8:2018 Clause 16.1 Objectives
- ISO 26262-8:2018 Clause 16.2 General
- ISO 26262-8:2018 Clause 16.3.1 Prerequisites
- ISO 26262-8:2018 Clause 16.4 Requirements and recommendations
- ISO 26262-8:2018 Clause 16.4.2
- ISO 26262-8:2018 Clause 16.4.3
- ISO 26262-8:2018 Clause 16.4.4
- ISO 26262-8:2018 Clause 16.5 Work products
- ISO 26262-8:2018 Clause 16.5.1

Correctness = 5P
Reinterpreting clause 16, which is meant for T&B components, for autonomous driving that is part of the AI realm (maybe ISO/IEC TR 5459 or ISO 22898) is an interesting rational, which may actually be quite applicable.

Completeness = 2P
-1P only implicit mention of DIA. Even tho clause 16 and the proposed clause hints at a DIA (Developer Interface Agreement) or a similar document, it is never explicitly mentioned.
-2P no mention of a Dependent failure analysis. This is a crucial step to ensure freedom from interference between the IP-protected software and other system components