Describe the software development process as per ISO 26262 guidelines. Include key activities and deliverables.

The software development process, as described by the ISO 26262 guidelines, is a critical part of product development at the software level (ISO 26262-2:2018, Clause 5.2.2.2 f)). It is based on the concept of a V-model, with the specification of software requirements, software architectural design, and implementation on the left side, and software integration and verification on the right side (ISO 26262-2:2018, Clause 5.2.2.2 f)). The overall management tasks, including planning, coordinating, and monitoring safety activities, apply throughout the safety lifecycle (ISO 26262-2:2018, Clause 5.2.2.1).

Here are the key activities and deliverables:

1. **Software Development Process and Environment Setup:** When developing safety-related embedded software, suitable software development processes and environments shall be used. These must be compatible with system and hardware development phases regarding interaction and information exchange, and support consistency across sub-phases and work products (ISO 26262-6:2018, Clause 5.4.1). Agile software development approaches can be suitable, provided safety activities are tailored, and fundamental documentation, process, or safety integrity rigour is not omitted or ignored (ISO 26262-6:2018, Note 1 to Clause 5.4.1).

2. **Language and Tool Selection:** Criteria for selecting design, modelling, or programming languages include their unambiguous definition, suitability for managing safety requirements, support for modularity, abstraction, encapsulation, and structured constructs (ISO 26262-6:2018, Clause 5.4.2). Any criteria not sufficiently addressed by the language itself shall be covered by corresponding guidelines or the development environment, considering topics such as enforcement of low complexity, use of language subsets, strong typing, defensive implementation techniques, well-trusted design principles, unambiguous graphical representation, style guides, naming conventions, and concurrency aspects (ISO 26262-6:2018, Clause 5.4.3, Table 1). Qualified software tools can be considered as supporting information (ISO 26262-6:2018, Clause 5.3.2). The software tool criteria evaluation report or the software tool qualification report can provide input to tool usage (ISO 26262-6:2018, Note 2 to Clause 5.4.1).

3. **Software Unit Design and Implementation:** For safety-related software units, the design and implementation shall be suitable to satisfy allocated software requirements, consistent with the software architectural design specification, and consistent with the hardware-software interface specification (ISO 26262-6:2018, Clause 8.4.1, Clause 8.4.2). To avoid systematic faults, the software unit design shall be described using appropriate notations (e.g., natural language, informal, semi-formal, or formal notations) to achieve consistency, comprehensibility, maintainability, and verifiability (ISO 26262-6:2018, Clause 8.4.3, Table 5). The specification shall detail functional behavior and internal design for implementation (ISO 26262-6:2018, Clause 8.4.4). Design principles shall be applied to achieve properties like correct execution order, interface consistency, data/control flow correctness, simplicity, readability, robustness, suitability for modification, and verifiability (ISO 26262-6:2018, Clause 8.4.5).

    - **Inputs:** Documentation of the software development environment (ISO 26262-6:2018, Clause 5.5.1), refined hardware-software interface specification (ISO 26262-6:2018, Clause 6.5.2), software architectural design specification (ISO 26262-6:2018, Clause 7.5.1), software safety requirements specification (ISO 26262-6:2018, Clause 6.5.1), configuration data (ISO 26262-6:2018, Clause C.5.3), and calibration data (ISO 26262-

6:2018, Clause C.5.4) (ISO 26262-6:2018, Clause 8.3.1).
    - **Deliverables:** Software unit design specification (ISO 26262-6:2018, Table A.1 for Clause 9 Prerequisites, referring to 8.5.1), Software unit implementation (ISO 26262-6:2018, Table A.1 for Clause 9 Prerequisites, referring to 8.5.2).

4. **Software Verification:** Verification activities are performed to ensure work products fulfill project and technical requirements, particularly regarding use cases and failure modes (ISO 26262-2:2018, Clause 6.2). For software units, verification aims to provide evidence that the design satisfies allocated software requirements and is suitable for implementation, to verify that defined safety measures are properly implemented, and to ensure the implemented software unit complies with the design and fulfills allocated requirements, containing neither undesired functionalities nor undesired properties regarding functional safety (ISO 26262-6:2018, Table A.1 for Clause 9 Objectives).

    - **Deliverables:** Software verification specification (ISO 26262-6:2018, Table A.1, 9.5.1), Software verification report (refined) (ISO 26262-6:2018, Table A.1, 9.5.2).

5. **Supporting Processes:** Documentation Management A documentation management strategy is developed to facilitate an effective and repeatable documentation management process (ISO 26262-8:2018, Table for Clause 10 Objectives). Documents should be precise, concise, clearly structured, easy to understand, verifiable, and maintainable (ISO 26262-8:2018, Clause 10.4.3). Each work product or document shall be associated with formal elements such as title, author/approver, unique revision identification, change history, and status (ISO 26262-8:2018, Clause 10.4.5).

    - **Deliverables:** Documentation management plan (ISO 26262-8:2018, 10.5.1), Documentation guideline requirements (ISO 26262-8:2018, 10.5.2).

6. **Supporting Processes:** Confidence in Software Tools The objectives for confidence in software tools are to provide criteria for determining the required level of confidence and to provide means for qualification to create evidence that the tool is suitable for supporting ISO 26262 activities or tasks (ISO 26262-8:2018, Clause 11.1). Confidence is needed to minimize the risk of systematic faults in the product due to tool malfunctions and to ensure the development process complies with ISO 26262 if activities rely on the tool (ISO 26262-8:2018, Clause 11.2).

        - **Deliverables:** Software tool criteria evaluation report (ISO 26262-8:2018, 11.5.1), Software tool qualification report (ISO 26262-8:2018, 11.5.2).

7. **Safety Management and Planning:** Project-dependent safety management, including planning of safety activities, is performed throughout the product development phases (ISO 26262-2:2018, Clause 5.2.2.1). The safety plan includes planning for safety activities related to ISO 26262-6 (ISO 26262-2:2018, Clause 6.4.6.5 c)). It also includes planning for supporting processes, like confidence in the usage of software tools (ISO 26262-2:2018, Clause 6.4.6.5 i)).

Additional List of Clauses:

- ISO 26262-2:2018, Clause 5.2.2.1
- ISO 26262-2:2018, Clause 5.2.2.2 f)
- ISO 26262-2:2018, Clause 6.2
- ISO 26262-2:2018, Clause 6.4.6.5 c)

- ISO 26262-2:2018, Clause 6.4.6.5 i)
- ISO 26262-6:2018, Note 1 to Clause 5.4.1
- ISO 26262-6:2018, Note 2 to Clause 5.4.1
- ISO 26262-6:2018, Clause 5.3.2
- ISO 26262-6:2018, Clause 5.4.1
- ISO 26262-6:2018, Clause 5.4.2
- ISO 26262-6:2018, Clause 5.4.3
- ISO 26262-6:2018, Table 1
- ISO 26262-6:2018, Clause 6.5.1
- ISO 26262-6:2018, Clause 6.5.2
- ISO 26262-6:2018, Clause 7.5.1
- ISO 26262-6:2018, Clause 8.3.1
- ISO 26262-6:2018, Clause 8.4.1
- ISO 26262-6:2018, Clause 8.4.2
- ISO 26262-6:2018, Clause 8.4.3
- ISO 26262-6:2018, Table 5
- ISO 26262-6:2018, Clause 8.4.4
- ISO 26262-6:2018, Clause 8.4.5
- ISO 26262-6:2018, Clause C.5.3
- ISO 26262-6:2018, Clause C.5.4
- ISO 26262-6:2018, Table A.1 (for Clause 9 Objectives, 9.5.1, 9.5.2 and Prerequisites 8.5.1, 8.5.2)
- ISO 26262-8:2018, Table for Clause 10 Objectives
- ISO 26262-8:2018, Clause 10.4.3
- ISO 26262-8:2018, Clause 10.4.4
- ISO 26262-8:2018, Clause 10.4.5
- ISO 26262-8:2018, 10.5.1
- ISO 26262-8:2018, 10.5.2
- ISO 26262-8:2018, Clause 11.1
- ISO 26262-8:2018, Clause 11.2
- ISO 26262-8:2018, 11.5.1
- ISO 26262-8:2018, 11.5.2

Correctness = 2P,
-1P for "ISO 26262-6:2018, Note 1 to Clause 5.4.1" when it's actually Clause 5.2
-2P for mixing in the supporting processes. While what is stated is not wrong, this is not what the question is about.

Completeness = 1P
-1P for missing phases “Specification of software safety requirements" altho hinted at in Software Unit Design and Implementation
-1P for missing phases "Software architectural design"altho hinted at in Software Unit Design and Implementation
-2P completely missing phase “Software integration and verification“ without any hinting.
no additional points have been deducted for missing phase “Testing of the embedded software“ as this is often seen as under one umbrella with “Software integration and verification“

Various hinting at Annex A, specifically Table A.1 which essentially holds the entire answer to this question was the only reason why 1P was given at completeness. In itself the Answer is completely inadequate and a good example for the limitations of the retriever, who is not able to retrieve large

amounts of connected text, which would be necessary to fully answer this question. Alternatively Table A.1 could have been retrieved, which is much shorter, but the retriever was also not able to do this. There are a number of ways that could theoretically solve this problem, larger chunks or more clause contained chunks could be a solution´, but these would need to be manually crafted (or maybe with AI support). Nevertheless, a RAG implementation for these kinds of questions will always be challenging.