**Committee Board**

**Description of general functions within the project**

**Key Stakeholders:** They can be key stakeholders for the project, such as main client, investor or end users, they can also make important decisions by providing feedback and validation of the implemented functionalities.

**Quality Engineer:** is responsible for ensuring that the software meets quality standards and regulations. Perform tests, identify issues, and collaborate with the development team to ensure software is bug-free for a quality product.

**Project Manager:** Responsible for defining the product vision, prioritizing features, managing the task backlog, and representing customer needs. Make decisions about which features will be implemented and in what order.

**Manager:** Responsible for the planning, monitoring and general management of the project. Ensures the team meets established deadlines and budgets, as well as ensuring human resources and logistics.

**Software Architect:** The software architect is responsible for designing the structure and architecture of the system. He makes important technical decisions, such as the choice of technologies and design patterns, to ensure the technical quality and scalability of the software.

**Lead Developer:** The team leader is an experienced developer who leads a subteam within the development group. He is responsible for assigning tasks, monitoring progress, and ensuring that coding best practices are followed.

**Decision Making Policy for Project Change Requests**

**Objective:**

This policy is intended to establish a clear and effective process for evaluating and making decisions regarding requests for changes to the software development project. It seeks to ensure that changes are managed efficiently and that an appropriate balance is maintained between flexibility and stability of the project.

**Participants in decision-making and tasks:**

**Project Manager:** present the analysis of change requests to demonstrate whether or not they are aligned with business objectives.

**Manager:** present the strategic perspective regarding the logistics required for the change.

**Lead Developer:** present the evaluation of the technical feasibility of the proposed changes.

**Quality Engineer:** present the evaluation and quality impact of the proposed changes.

**Software Architect:** provide your experience in system architecture and design to present the compatibility evaluation of the proposed changes.

**Key Stakeholders:** present their opinions and points of view regarding the change

**Decision-making process:**

1. Change Request: Any team member or key stakeholder can submit a change request using a specific change request form.
2. Registration and Initial Evaluation: The Project Manager records the change request and distributes it to the relevant team members. The team initially reviews the request to understand its scope and potential impact.
3. Technical Evaluation: The Lead Developer and Software Architect evaluate the technical feasibility of the change and determine if it aligns with the existing architecture and design.
4. Quality Assessment: The Quality Engineer evaluates the quality impact of the proposed change and determines if additional testing is required.
5. Strategic Evaluation: The Manager and Key Stakeholders evaluate the strategic impact of the change in terms of business objectives, budget and deadline.
6. Review Meeting: The team meets to discuss the findings of the technical, quality and strategic evaluations. The risks of change are evaluated and their urgency and criticality determined.
7. Decision: The team makes a decision on the change request. Possible decisions include:

Approval: The change is approved, and its implementation is planned.

Rejection: The change request is rejected.

Postponement: The decision is postponed for future review or for consideration in planning for the next cycle.

1. Communication: All interested parties are informed about the decision made and the next steps.

**Measurement of Exchange Risks:**

The risks associated with change requests must be measured in terms of urgency and criticality:

**Urgent:** Changes that must be implemented immediately due to critical issues in production or regulatory requirements.

**Critical:** Breaking changes that significantly affect the functionality or stability of the system.

**Medium:** Changes that have a moderate impact on the system and can be planned in the current development cycle.

**Low:** Minor changes or improvements that have minimal impact on the system and can be planned in future releases.

**Guidelines for Deciding (Point 7)**

**Voting to approve, reject, or postpone a change must be done considering multiple factors.**

* Complete Change Request Review: Before voting, all team members should carefully review the change request and understand its scope and implications. This includes technical, quality and strategic evaluation.
* Defining Approval Criteria: Establishing clear criteria for approving changes is essential. Criteria should be specific and measurable, and should reflect project objectives and business requirements. This helps ensure that the decision is based on objective data.
* Risk Assessment: Prior to voting, a full risk assessment must be conducted to determine the urgency and criticality of the change. **Urgent or critical changes may require faster and more rigorous approval.**
* Identification of Key Stakeholders: Identify key stakeholders who may be affected by the change and consider their opinions. Stakeholders should be given the opportunity to express their views before the vote.
* Deliberate Voting: During voting, each team member must have the opportunity to express their opinion in a deliberate and reasoned manner. This may include discussions to clarify doubts or concerns before casting a vote.
* Informed Majority: The decision must be based on an informed majority. This means that team members must vote based on available information, established criteria, and risk analysis.
* Postpone if in Doubt: If there is a significant degree of uncertainty or lack of information, consider postponing the vote to obtain more data or conduct additional research.
* Record of Decisions: Carefully document the decisions made and the reasons behind each vote. This provides transparency and a basis for future references.

**Clear criteria for approval**

**Impact on Customer Requirements**: Does the change satisfy a valid customer need or request? Is it aligned with the agreed requirements in the contract or project specifications?

**Impact on Quality:** Does the change improve the overall quality of the software? Does it increase system stability, security or performance?

**Technical Feasibility:** Is it feasible to implement the change from a technical point of view? Are additional resources or specialized skills required to make the change effectively?

**Cost and Resources:** Is the change cost-effective and within the project budget? Does it require a significant investment of time or resources?

**Impact on Schedule:** Does the change affect the project schedule? Can it be done without delaying the delivery of the product?

**Compatibility with Existing Architecture:** Is the change consistent with the existing architecture and design of the software? Does it introduce conflicts or integration problems?

**Associated Risks:** What are the potential risks associated with the change? Can it have a negative impact on other components or functionalities of the system?

**Added Value:** Does the change add significant value to the project or the client? Does it improve the user experience or satisfy a critical need?

**Consistency with Business Strategy**: Is the change aligned with the strategic objectives of the organization or the client? Does it contribute to achieving the desired results?

**Alignment with Project Objectives:** Is the change consistent with the project objectives and vision? Does it contribute to meeting planned milestones and deliverables?

**Key Stakeholder Support:** Do key stakeholders, including the customer, support the change? Have they expressed their approval or disapproval?

**History of Similar Changes:** Are there similar changes that have been approved or rejected in the past? What lessons can be learned from those experiences?

In short, a vote to approve, reject, or postpone a change to a software development project should be a rigorous, data-driven process. Consideration of technical, quality and strategic information, together with risk assessment and participation of key stakeholders, contributes to making informed decisions that benefit the project as a whole.

**This decision-making policy helps ensure that change requests are comprehensively evaluated, considering both technical and strategic and quality aspects, and that informed decisions are made for the benefit of the project and stakeholders.**