Title: Incremental Specification for Inclusive and Systematic Problem-Solving in Software Development

1. Purpose and Scope:

- This specification outlines a problem-solving framework designed to address systematic issues through the development of computer programs. It incorporates inclusivity guidelines, especially for neurodivergent individuals, and requires responses to 15 mandatory posed questions, typically provided in a CV or clarified by an employer.
- The specification is intended to guide developers working with languages like C, Rust, Go, and web technologies (e.g., VueJS) to create solutions affecting large institutions (e.g., governments, communities, etc.).

2. Core Principles:

- **Inclusivity:** The specification promotes a problem-solving framework that accommodates neurodivergent thinkers. It emphasizes sensory-friendly environments, clear communication, and structured problem-solving to minimize cognitive overload.
- **Incrementality:** The specification follows an incremental model, ensuring that no component is overlooked or misunderstood. Each step builds on the previous one, facilitating gradual adoption.
- **Thorough Testing:** Programs created under this specification must undergo rigorous testing for functionality, security, and inclusivity.

3. Mandatory Guidelines:

- **15 Mandatory Questions:** Before commencing any project, the users must answer 15 posed questions. These questions ensure alignment with the specification’s goals and must be verified by peers or employers.
  - Examples include: *What cognitive or sensory accommodations do you require? How do you handle complex tasks under time constraints?*
- **SMART Goals:** Each sub-problem must have clearly defined SMART goals, with set timelines and built-in breaks to prevent burnout.

- **Graph Structure:** Implement a graph-based system for breaking down and visualizing problems based on difficulty, ensuring systematic tackling of each element.
- **PARA Framework:** Use the PARA structure to organize project files, resources, and related information. This aids in transparency and easy access to materials.

4. Theoretical Framework:

- The specification draws from **strength-based and neurodiversity models** that highlight cognitive diversity. This ensures that projects celebrate diverse thinking patterns and problem-solving approaches.
- For example, a developer struggling with time management can implement task breakdown methods specific to neurodivergent workflows, such as Pomodoro or Kanban boards.

5. Testing and Evaluation:

- **Continuous Testing:** The specification mandates continuous, incremental testing for programs developed under it. Tests should include functionality checks, security audits, and stress tests, especially in contexts where the application impacts large institutions or governmental systems.
- **User Feedback:** Feedback loops from neurodivergent individuals must be incorporated during testing phases to ensure the system remains accessible.

6. Specification Adoption and Review Process:

- This specification should be periodically reviewed by experts and stakeholders to ensure compliance with inclusivity and effectiveness in tackling problems.
- **Review Questions:** Users and reviewers should revisit the 15 mandatory questions periodically to check if the project has stayed true to its core goals.
- The review process allows the flexibility to make adjustments and adapt to feedback from those working under the specification.

This specification forms the foundation for building various software applications that address systematic institutional issues, with neurodivergent inclusivity at its heart. Each developer, working under it, will find a structured, inclusive, and incrementally supported environment to create impactful programs, rigorously tested to meet the needs of various stakeholders.