# Software Requirements Specification (SRS)

## Project: Agentic Qtile

### 1. Introduction

Agentic Qtile is a fork of the Qtile window manager designed to serve as a native environment for autonomous AI agents. It implements an "Agent2UI" architecture where the desktop environment is ephemeral, generative, and goal-oriented.

### 2. Core Concepts

* **Ralph Wiggin Protocol:** A continuous iteration loop where agents execute tasks, intercept exit signals, and re-run with updated context until a success condition (e.g., test pass) is met.
* **GAD Framework:** A hierarchical structure for agent reasoning: **G**oals (User intent), **A**ctions (Tool use/WM manipulation), and **D**ecisions (Branching logic based on WM state).
* **Agent2UI:** A protocol where the UI is not a pre-defined layout but a declarative set of "cutouts" generated to facilitate a specific task.

### 3. Functional Requirements

* **FR-1: Dynamic Inlay Rendering:** The WM must be able to render real-time text/graphics (e.g., confidence scores) directly onto window borders and margins using Cairo.
* **FR-2: Sandboxed Execution:** Windows spawned by agents must be visually isolated (distinct decorators) and restricted in their interaction with the host system.
* **FR-3: Semantic IPC:** A bidirectional JSON-RPC socket allowing agents to query the WM state (window titles, focus, layout tree) and issue layout commands.
* **FR-4: Ephemeral Layouts:** The ability to "carve out" areas of the screen on-the-fly without disrupting existing tiling logic.

### 4. Non-Functional Requirements

* **Security:** High-fidelity isolation for agent-controlled processes.
* **Performance:** Cairo overlays must render with <16ms latency to ensure no desktop stutter.
* **Transparency:** Every agent action must be visually signaled to the user via "Control Inlays."