sgClaw 团队管理标准（V1.0）

> 适用范围：P1a、P1b、P2、P3、P4 五个角色并行开发与联调。 管理原则：**接口先行、清单驱动、里程碑验收、变更可追溯**。

1. 五角色统一工作清单（每人必遵循）

1.1 每日清单（Daily）

- [ ] 站会前更新昨日产出、今日计划、阻塞项（最多 3 条）
- [ ] 当日代码提交至少 1 次，提交信息含角色前缀（如 `P2: ...`）
- [ ] 对齐接口文档变更（若有）并在群内公告
- [ ] 完成最小可运行验证（本地或联调环境）

1.2 里程碑清单（DoD）

- [ ] 代码通过本角色测试（单测/集成）
- [ ] 产出物齐全（代码 + 配置 + 文档 + 示例）
- [ ] 错误码与日志可定位（含 `seq` / `action` / `error.code`）
- [ ] 通过对应上下游角色联调验收

2. 五个角色分工清单（按职责）

P1a（核心通信，Rust）

- [ ] 实现 Pipe 协议：JSON Line、`seq` 递增、消息体 ≤ 1MB
- [ ] 实现 BrowserPipeTool 与 MAC/HMAC 校验
- [ ] 提供 command 发送与 response 关联能力（按 `seq`）
- [ ] 提供协议测试样例（成功 + 失败）

P1b（业务支持，Rust）

- [ ] Skill 加载、签名校验、执行沙箱落地
- [ ] 记忆分层（L0/L1/L2）可读写与检索
- [ ] AgentRuntime 与 P1a 工具链路打通
- [ ] Critic 与熔断策略生效

P2（浏览器对接，Chromium C++）

- [ ] SgClawProcessHost 生命周期管理（start/stop/crash）
- [ ] PipeListener 正确收发并严格按 Schema 校验
- [ ] MAC 白名单检查与 CommandRouter 映射一致
- [ ] 回传标准错误码（`PIPE_*` / `MAC_*` / `CMD_*`）

P3（业务技能，JS）

- [ ] 产出 skill 元数据、参数 Schema、签名文件
- [ ] 每个 skill 提供最小可运行示例

- [ ] 对关键场景补齐降级与异常处理
- [ ] 与 P1b 联调加载、执行、回滚路径

P4（前端与发布，Vue/DevOps）

- [ ] 管理面板支持启动/停止/状态展示
- [ ] 人工确认（human-in-the-loop）交互闭环
- [ ] 打包脚本可一键产物（deb/exe）
- [ ] 发布文档与回滚步骤完整

2.6 分发给个人的任务卡模板（直接复制）

```
负责人：
角色：P1a / P1b / P2 / P3 / P4

本周必须交付：
1)
2)
3)

联调对象：
阻塞项：

验收证据（必填）：
- 提交记录：
- 测试结果：
- 日志/截图：
```

3. Chromium ↔ sgClaw 接口标准（联调强约束）

3.1 协议边界与版本冻结

- 单一事实来源：`docs/L2-核心模块与接口契约层.md` 第 5 章（5.1~5.4）。
- 协议版本：固定 `1.0`；字段增删、action 增删、错误码增删均视为“协议变更”。
- 协议变更流程：先 RFC 评审（P1a+P2+管理者）→ 更新 L2 → 再改代码。

3.2 线级契约（Wire Contract，双方 MUST）

项目	强约束	违规处理
传输	STDIO + JSON Line（每行一条完整 JSON）	`PIPE_INVALID_JSON`
编码	UTF-8	`PIPE_INVALID_JSON`
大小	单条消息 `<= 1MB`	`PIPE_MESSAGE_TOO_LARGE`
时序	`seq` 从 1 开始严格递增，不重复不乱序	`PIPE_SEQ_DUPLICATE` / `PIPE_SEQ_OUT_OF_ORDER`

项目	强约束	违规处理
安全字段	command 必带 `security.expected_domain` 与 `security.hmac`	`PIPE_HMAC_INVALID` / `MAC_*`
一致性	一个 command（`seq=N`）必须且只能对应一个 response（`seq=N`）	`INTERNAL_UNKNOWN` + 熔断

3.3 握手契约（启动即校验）

1. Browser 拉起 sgClaw 子进程后，立即发送：

```
{"type":"init","version":"1.0","hmac_seed":"<hex>"}
```

2. sgClaw 必须返回：

```
{"type":"init_ack","version":"1.0","agent_id":"<uuid-
v4>","supported_actions":["click","type","navigate"]}
```

3. Browser 等待 `init_ack` 超时 `5000ms`：Kill 子进程，状态置为 `Crashed`，通知 UI。
4. version 不一致：立即失败，不进入 Running。

3.4 命令/响应契约（字段级）

- command 必填：`seq`、`type=command`、`action`、`params`、`security`。
- response 必填：`seq`、`type=response`、`success`。
- 失败响应必填：`error.code`、`error.message`（禁止仅返回字符串错误）。
- `action` 仅允许 L2 §5.1 枚举；`params` 必须通过 L2 §5.2 对应 Schema 校验。

标准 command 示例：

```
{"seq":12,"type":"command","action":"click","params":
{"selector":"#submit"},"security":
{"expected_domain":"erp.example.com","hmac":"<hex>"}}
```

标准失败 response 示例：

```
{"seq":12,"type":"response","success":false,"error":
{"code":"CMD_SELECTOR_NOT_FOUND","message":"selector #submit not found"}}
```

3.5 错误处理与重试矩阵（执行统一）

错误类型	是否重试	规则

错误类型	是否重试	规则
`PIPE_*`	否	直接失败，修协议/编码/消息体
`MAC_*`	否	直接失败，修白名单/域名/人工确认
`CMD_SELECTOR_TIMEOUT`	是	最多 2 次，退避 500ms/1000ms
`CMD_NAVIGATION_FAILED`	是	最多 1 次，退避 1000ms
`INTERNAL_*`	是	最多 1 次，失败即上报

- 同一 action 连续失败 `>10` 次：触发熔断（停止执行 + 人工介入）。

3.6 联调验收清单（全部通过才算完成）

- [ ] `init -> init_ack` 成功率 100%（连续 100 次启动）。
- [ ] 版本不匹配可稳定失败，且 UI 能看到明确错误。
- [ ] `seq` 重复/乱序可稳定复现并返回标准错误码。
- [ ] 超大消息（>1MB）被拒绝并返回 `PIPE_MESSAGE_TOO_LARGE`。
- [ ] 核心 action（click/type/navigate/getText）成功率 `>=99%`。
- [ ] 所有失败场景均有结构化 `error.code` 与 `error.message`。
- [ ] 日志可按 `seq` 串联：发出 command、浏览器执行、返回 response。