

# 软件层 MVP 验证落地手册

面向非技术发起人的软件验证主线手册。

## 这份文档是给谁看的

这份文档是写给非技术背景的发起人看的。

假设读者：

- 懂泳馆业务，但不懂研发
- 手里已经有一些泳池视频，或者能拿到历史录像
- 现在想先验证“软件链路能不能跑通”
- 当前不优先做摄像头部署、不优先做现场联动、不优先做自训练
- 重点是验证算法和系统链路是否 work

## 先说结论

### 这次 MVP 要验证的，不是硬件方案

这次最小验证的目标应该收缩为：

1. 把现成泳池视频输入系统
2. 系统对视频做检测、跟踪和事件判断
3. 系统输出“第几秒、哪条泳道、什么异常、对应片段”
4. 人工回看结果，判断这条链路是否具备继续做下去的价值

### 这次 MVP 暂时不验证什么

- 不验证摄像头怎么装
- 不验证 RTSP 实时拉流是否稳定
- 不验证水下摄像头部署方案
- 不验证企业微信、钉钉、声光报警
- 不优先验证自训练是否必须
- 不优先验证跨泳馆泛化能力

## 一句话版本

先做一个离线视频输入 -> 事件输出 -> 人工复盘的软件 MVP。

如果这条链路都跑不顺，就不要进入硬件接入和现场试点。

## 这次 MVP 的成功标准

第一版不要追求“真的识别出所有溺水”。

更现实的成功标准是：

1. 能稳定处理一批现成视频，不中途崩掉
2. 能为视频中的泳者生成连续轨迹
3. 能根据规则输出候选异常事件
4. 能把事件定位到时间点、区域或泳道
5. 能导出可复盘的结果文件和标注后的视频

## 这次 MVP 的最低交付物

至少应该交出这 5 样东西：

1. 一份可运行的软件工程目录
2. 一批输入视频和对应清单
3. 一份事件结果表
4. 一批带框和轨迹的输出视频
5. 一份人工复盘后的结论报告

## 非技术发起人要接受的现实

### 他不能独立完成技术实现

如果你朋友完全不懂技术，他不能独立做出这次 MVP。

但他可以很好地承担下面这些事情：

- 提供真实视频和业务场景
- 定义什么事件值得提醒
- 把“哪条泳道、哪个区域算危险”讲清楚
- 组织人工复盘
- 判断系统结果有没有业务价值

### 他应该把自己定位成什么

更准确的定位是：

- 场景负责人
- 数据和验收负责人

- 业务规则定义人

不是：

- 算法工程师
- 模型训练工程师
- Linux 运维工程师

## 这次 MVP 至少需要哪几类人

角色	最低人数	主要职责
场景负责人	1	提供视频、解释场景、定义事件、组织复盘
技术实施人	1	环境搭建、跑检测跟踪、写规则、导出结果
复盘协作人	1	对输出结果做人审，记录真阳性和误报

如果连 1 个能写 Python、会处理视频的技术人都没有，这次 MVP 就不应启动。

## 非技术发起人需要学到什么程度

### 够用就行，不要过度学习

他不需要先学机器学习。

他只需要先搞懂下面这些最低概念：

- GitHub 是什么
- ZIP 压缩包怎么解压
- README 文档怎么读
- 视频文件、输出文件、配置文件分别是什么
- 出错时怎么截图和把日志发给技术方

### GitHub、GitCode、Gitee 要怎么理解

- Git = 版本管理工具
- GitHub = 开源代码最常见的托管平台
- GitCode / Gitee = 国内可访问的替代或镜像平台

参考：

- About Git - GitHub Docs (<https://docs.github.com/en/get-started/using-git/about-git>)

- 迁移项目 | GitCode 帮助文档 (<https://docs.gitcode.com/v1-docs/docs/repo/migrate/>)
- GitHub 仓库快速导入 Gitee 及同步更新 - Gitee (<https://gitee.com/help/articles/4284>)

## 如果他不会科学上网怎么办

不要把这件事理想化。

现实做法只有这几种：

1. 技术方在能访问 GitHub 的环境里准备好代码和依赖
2. 把仓库同步到 GitCode / Gitee
3. 直接发 ZIP 压缩包和 Docker 镜像

对非技术发起人来说，最实用的方式通常是第 3 种。

## 这次 MVP 的范围怎么定义

### 建议只选 2 到 3 类事件

第一版建议只验证这些软件上容易定义的事件：

- 长时间低速或近似静止
- 在高风险区域停留过久
- 轨迹突然消失后长时间没有再出现
- 明显偏离正常游动模式的候选异常

### 第一版先不要碰这些

- 救生员玩手机识别
- 泳姿识别
- 情绪识别
- 语音播报联动
- 多摄像头跨视角融合
- 深度学习自训练闭环

## 这次 MVP 的主链路

这次应该按下面这条链路来设计：

1. 输入现成视频文件
2. 统一视频格式和命名
3. 对视频跑检测和跟踪

4. 把轨迹映射到泳道和区域
5. 用规则引擎输出候选事件
6. 生成结果表、片段和标注视频
7. 人工复盘

这条链路里，真正必须验证的是：

- 视频能不能稳定读
- 目标能不能被看见
- 轨迹能不能连续
- 规则能不能产出有意义的事件
- 结果能不能被人审

## 一次合格的软件 MVP，输出应该长什么样

至少要能输出下面这些字段：

字段	说明
video_name	来自哪段视频
event_type	异常类型
start_sec	开始时间点
end_sec	结束时间点
lane_id	所在泳道或区域
track_id	对应的目标轨迹 ID
score	风险评分或规则分数
clip_path	对应片段路径
review_result	人工复盘结论

## 详细执行路径

### 阶段 0：先把目标说清楚

先写一句话版本的 MVP 目标。

推荐写法：

用现成的泳池视频，验证软件系统能否输出候选溺水异常事件，并定位到时间点和泳道，供人工复盘。

这一阶段的交付物：

- 一页纸目标说明
- 事件定义表
- 不做事项清单

## 阶段 1：准备输入视频

第一版不需要先采新视频。

优先用这几类现成素材：

- 正常游泳视频
- 高峰期馆内视频
- 训练课视频
- 演练视频
- 历史录像中的复杂场景片段

建议先准备：

- 10 到 20 段短视频做第一轮跑通
- 再准备 30 到 50 段视频做第二轮验证

每段视频都至少要记录：

- 视频名
- 时长
- 拍摄视角
- 是否已知有异常演练
- 泳道信息是否可见

## 阶段 2：把视频整理规范

这一阶段不要急着跑模型。

先把视频统一到能处理的状态。

至少要做到：

- 文件命名统一
- 格式尽量统一为 mp4
- 帧率、分辨率、时长有清单
- 明确哪些视频能看清泳道，哪些不能

推荐命名：

```
2026-04-08_poolA_overhead_lane-visible_normal_01.mp4
2026-04-09_poolA_overhead_lane-visible_drill-low-motion_01.mp4
2026-04-09_poolA_underwater_lane-unknown_drill-sink_01.mp4
```

这一阶段的交付物：

- video\_manifest.csv
- 标准化后的视频目录

### 阶段 3：先跑最基础的检测和跟踪

这里不要一上来就训练。

先用开源现成模型跑出基线。

更现实的顺序是：

1. 先看模型能不能把人稳定框出来
2. 再看跟踪 ID 能不能连续
3. 最后再判断规则是否成立

如果第一步都做不到，就说明：

- 视频质量不够
- 视角不合适
- 当前开源模型不适合这个素材

这时候还不该先谈报警逻辑。

### 阶段 4：做泳道和区域映射

你要的输出里有“哪条泳道”，所以必须有一份区域映射。

这一步不需要 AI。

它本质上是：

- 在画面中画出泳池范围
- 画出每条泳道的多边形区域
- 定义高风险区域

然后把目标轨迹中心点落到这些区域里。

这样系统才有可能输出：

- 第 123 秒
- 第 4 泳道
- 长时间低速停留

## 阶段 5: 把轨迹转成候选异常事件

这一步也不一定要训练模型。

第一版更建议用规则：

- 某个目标在同一区域停留超过阈值
- 某个目标长时间速度很低
- 某个目标轨迹突然消失
- 某个目标长时间处于低位或边缘异常区域

然后把多个弱信号组合成一个事件。

第一版不要追求“直接判断溺水”。

更合理的输出应该是：

- 候选异常事件
- 值得人工优先回看的片段

## 阶段 6: 导出输出结果

每跑完一段视频，至少导出这几类结果：

- events.csv
- events.json
- annotated\_video.mp4
- clips/ 目录
- review\_sheet.xlsx 或 review.csv

只有把结果导出来，非技术发起人才有东西可看，有东西可验收。

## 阶段 7: 人工复盘

这一步非常关键。

没有人工复盘，就没有 MVP 结论。

每条事件都至少要判断：

- 是真的值得提醒，还是明显误报
- 时间点是否接近真实发生时刻
- 泳道定位是否正确
- 规则描述是否有解释性

建议复盘表里至少保留这些列：

- event\_id
- reviewer
- true\_or\_false
- useful\_or\_not
- lane\_correct
- time\_correct
- comment

## 阶段 8：只调规则，不急训练

如果第一轮结果不好，先排查这 4 类问题：

1. 视频本身看不清
2. 检测漏人
3. 跟踪断裂
4. 规则阈值不合理

只有当你确认：

- 视频质量还可以
- 基础检测也能看见人
- 但特定动作还是长期识别不到

这时才值得进入“收集数据做微调”。

## 阶段 9：形成结论

最后不能只说“感觉还行”。

必须输出一份结论表。

建议至少写清楚：

- 跑了多少段视频

- 总时长多少
- 输出了多少候选事件
- 其中多少是有价值的提醒
- 哪类误报最多
- 哪些场景完全看不见
- 是否值得进入下一阶段

## 非技术发起人每周应该投入多少时间

如果只是配合这次软件 MVP，建议按这个量级预期：

- 第 1 周到第 2 周：每周 6 到 8 小时
- 第 3 周到第 6 周：每周 8 到 12 小时

这些时间主要花在：

- 整理视频
- 定义事件
- 解释业务场景
- 做人工复盘
- 和技术方对齐结果

不是花在写代码。

## 什么情况下可以进入下一阶段

只有满足下面这些条件，才建议从软件 MVP 进入“硬件接入 / 现场试点 / 自训练”阶段：

1. 离线视频链路已经稳定
2. 输出结果能定位到时间点和区域
3. 至少一部分候选事件被业务方认为有价值
4. 误报来源已经基本看清
5. 团队知道下一步是补规则、补视角还是补训练

如果这些都还没弄清，就不要先谈摄像头部署。

## 读完这份文档后，下一份该看什么

推荐顺序：

1. 先看 软件层 MVP 工具链与开源项目清单 (/toolchain/)

2. 再看 离线视频软件 MVP 操作手册 (/software-mvp/)

3. 最后再看 室内泳馆现场试点技术规划 (/pilot-plan/)

这样顺序才不会反。