

自然语言需求如何成为可验证的 SystemSpec

Xg-Stellar 工业闭环中的需求结构化、知识引用、人工审核与冻结机制

一句话定义

SystemSpec

是把自然语言需求与已审核知识资料整理成的系统级、结构化、可验证、可追溯规格。它不是原始 TXT，而是后续 FPGA、嵌入式实现和验收共同遵守的正式依据。

1	自然语言输入	2	生成规格草案	3	人工审核与冻结
	specification.txt、任务书、聊天描述		提取需求、接口、责任分配和验证义务		确认草案内容，以 SHA-256 固定版本

TXT 与 SystemSpec 的关系

角色	原始输入材料	正式执行和验收基准
内容	自然语言，可能模糊或遗漏	需求编号、责任分配、接口和验证方式
可变性	可继续补充修改	冻结后不可原地修改；变更需生成新版本
是否直接驱动工业验证	否	是

SystemSpec 里冻结什么？

每一项信息都服务于后续实现、验证和证据追溯。

系统身份

schemaVersion、systemId，以及 FPGA 顶层模块和 HDL 语言。

需求清单

每条需求具有唯一 REQ 编号和明确、可验收的描述。

责任分配

需求必须分配给 fpga、embedded 或 shared，避免软硬件边界遗漏。

验证义务

明确 simulation、formal、host_contract、target 或 HIL 等验证方式。

接口定义

寄存器、总线、引脚、时钟、复位、中断和协议的边界约束。

知识依据

记录已审核资料的来源、版本、文档标识和内容 SHA-256。

为什么必须使用已审核知识资料？

工业规格不能依靠模型自由猜测阈值、协议或安全规则。只有 approvalStatus=approved 的资料可以进入冻结依据；系统同时记录资料内容哈希，资料发生变化时能够识别版本漂移。

一个简化示例

自然语言	“FPGA 检测按键，软件读取状态，按键需要消抖。”
结构化需求	REQ-001：输入稳定 20 ms 后更新状态，分配给 FPGA，以仿真验证。 REQ-002：软件通过只读寄存器读取稳定状态，分配为 shared，以仿真和主机契约验证。

“冻结”具体意味着什么？

冻结不是把文件设为只读这么简单，而是建立一条可审计的版本与证据链。

01	生成草案	系统从自然语言和 approved 知识资料生成结构化 JSON 草案。
02	自动检查	检查需求是否完整、责任是否分配、接口是否明确、验证义务是否存在。
03	人工审核	人员确认行为、接口、阈值和验收边界，并确认草案 approvalHash。
04	冻结版本	生成正式 SystemSpec，保存原始字节并计算 SHA-256。
05	分支执行	FPGA 与嵌入式分别实现和验证，所有证据绑定同一 SystemSpec 哈希。
06	汇总验收	共享需求必须同时具备两侧证据；只有人工确认后才能进入 ACCEPTED。

冻结后的保护规则

如果 SystemSpec 被修改、删除，或 FPGA/嵌入式证据中的 systemId、systemSpecPath、systemSpecSha256 与冻结版本不一致，系统将拒绝继续验收。若确实需要改变行为、接口、阈值或验收标准，必须回到需求澄清环节，重新审核并冻结新版本。

功能价值

防止需求漂移 实现过程不悄悄改变目标	软硬件对齐 FPGA 与嵌入式共享同一基准
证据可追溯 测试结果绑定规格和知识版本	变更可审计 新版本不覆盖历史依据

结论

specification.txt 是输入，SystemSpec 是审核后冻结的执行基准。
核心是把模糊需求变成可验证的系统合同，让知识、实现、证据与验收绑定同一版本。