



智慧星空（上海）信息技术有限公司

软件开发流程规范

	软件开发流程规范	编号：Q/ISTAR-WI-35
		版本：B
		密级：三级
		页码：第 1 页 共 13 页

目 次

前言	2
1 目的	3
2 范围	3
3 术语和定义	3
4 概述	4
5 软件计划	4
6 软件开发	5
7 软件构建	6
8 软件测试	7
9 软件发布	8
10 软件部署	8
11 软件运维	9
12 相关文件标准	10
13 记录	10
14 附录	10

	软件开发流程规范	编号：Q/ISTAR-WI-35
		版本：B
		密级：三级
		页码：第 2 页 共 13 页

前 言

本标准的编写格式符合GB1.1《标准化工作导则 第1部分：标准的结构和编写》的规定。


本标准主要起草部门：软件工程部。

本标准主要起草人：赵滨。

本标准由公司标准化主管部门归口。

本标准首次发布为2023年2月1日，本次为第一次修订。

序号	修订日期	修订内容	版本号	修订人
1	2023-2-1	新建	A	赵滨
2	2024-5-13	新增CMake构建流程与目标	B	穆昌根

	软件开发流程规范	编号：Q/ISTAR-WI-35
		版本：B
		密级：三级
		页码：第 3 页 共 13 页

1 目的


该手册定义了软件开发的完整流程，明确了软件开发流程中的约束和规范等信息，用于指导开发人员进行公司产品软件系统的全生命周期的设计与开发。

2 范围

本规范描述了公司设备软件系统开发的全生命周期流程，所有软件开发相关的工作均应遵循本文中定义的流程规范。

3 术语和定义

DevOps	Development & Operations
Git	Git 代码版本控制工具
CMake	CMake 代码构建工具
SonarQube	SonarQube 代码静态测试工具
Valgrind	Valgrind 代码动态测试工具
Jenkins	Jenkins 自动化工具
ScrumMaster	敏捷开发迭代负责人
HIL	Hardware in the loop
NAS	网络文件存储器

	软件开发流程规范	编号：Q/ISTAR-WI-35
		版本：B
		密级：三级
		页码：第 4 页 共 13 页

4 概述

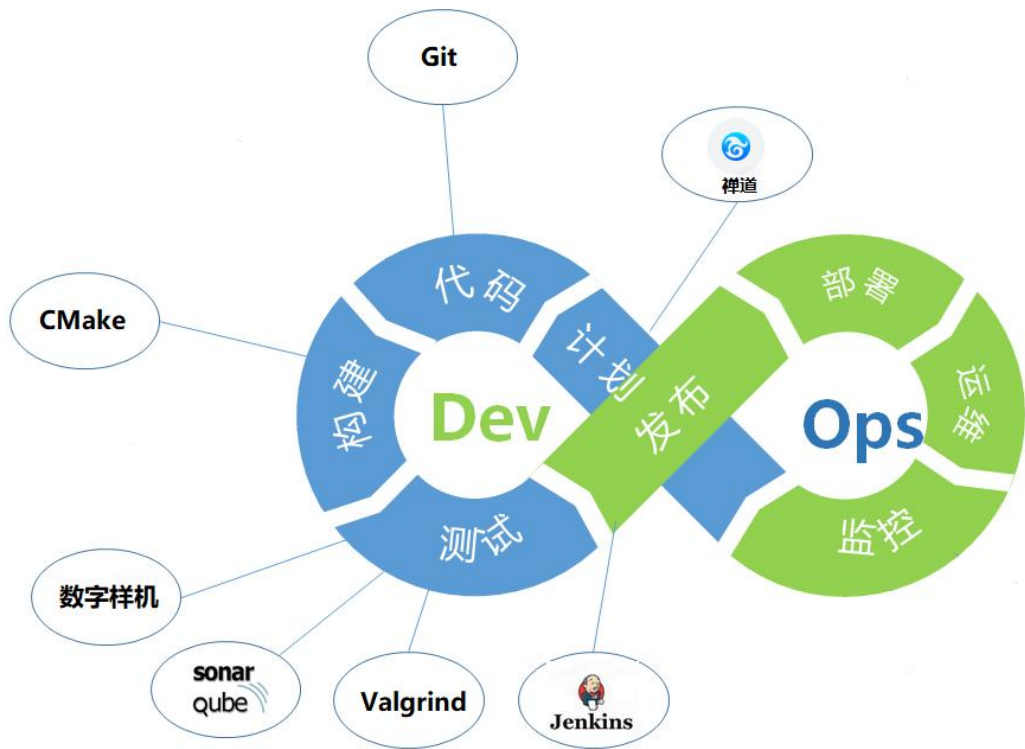


图 1 软件开发流程

公司的软件开发流程按DevOps的过程与方法组织实施，参见图1。其总体的目标为：

- a) 促进软件开发、技术运营、质量保障业务之间的沟通、协助与整合；
- b) 使得构建、测试、发布软件能够更加的快捷、频繁和可靠。

5 软件计划

软件计划的环节包含需求的管理、问题的管理、以及软件开发任务的发布与跟踪。软件计划采用“禅道”工具进行相关活动的管控。

5.1 需求管理


所有软件开发活动的起点为软件需求，需求有如下来源：

- a) 产品对软件提出的需求；
- b) 合规性需求，例如：公司统一的软件架构要求、行业的标准符合性需求等；
- c) 软件运维过程中，提出的改进和变更需求。

每个软件产品的需求，统一录入至“禅道”对应的产品结构下。需求细化至最小的软件功能实现单元。所有新增需求及变更需求，需提交产品软件对应的负责人或部门经理评审确认。

需求按重要程度进行分级：

- a) ① 关键需求，产品功能性能实现必备的需求；
- b) ② 重要需求，作为一个完整软件产品必备的需求；
- c) ③ 一般需求，增强软件功能或提升软件使用体验的需求；
- d) ④ 可选需求。

	软件开发流程规范	编号：Q/ISTAR-WI-35
		版本：B
		密级：三级
		页码：第 5 页 共 13 页

5.2 问题管理

所有软件产品的问题，统一录入至“禅道”对应的产品结构下。

问题的来源如下：

- a) 代码错误；
- b) 代码规范性问题；
- c) 软件设计缺陷；
- d) 软件性能问题；
- e) 代码优化改进；
- f) 其它。

软件问题按严重程度和解决优先级分级管控，分级按①②③④进行。严重级别①表示最严重，解决优先级①表示需最优先解决。

5.3 任务管理

软件开发的任务使用“禅道”工具进行管理。

任务管理的逻辑如下：

- a) 在对应的软件产品下创建计划，例如产品的阶段开发计划或产品的定期版本发布计划；
- b) 创建项目，关联对应的产品和计划；
- c) 在项目内创建一个或多个迭代，关联对应的产品和计划；
- d) 在迭代内创建一个或多个任务，每个任务可以关联相应需求，也可为离散任务；
- e) 由产品负责人或 Scrum Master 将任务指派给相应的开发人员，并跟踪迭代至所有任务完成。

6 软件开发

6.1 开发环境

软件开发的环境参见表1，详细的软件开发技术栈参见附录B《iStar软件开发技术栈》。

表 1 软件开发环境

软件类型	开发环境	说明
前端界面	Qt PyQt	设备的前端人机交互软件使用 Qt 开发； 部分工具软件可采用 PyQt 开发。
应用	Qt VisualStudio Python	设备的应用软件主要采用 C++开发，允许混用部分 Python；设备的测校软件采用 Python 开发为主。
驱动	Qt VisualStudio Cordis TwinCAT	设备的驱动软件根据具体需求，可采用 C++及基于 Cordis 及 TwinCAT 的软 PLC 开发。
固件	Vivado 伪代码	设备的固件采用 C、Verilog（FPGA）、以及基于控制器特定的伪代码进行开发。

6.2 交付件

软件开发在设备产品的各开发阶段的交付件参见表2。

表 2 软件开发交付件

开发阶段	交付件	说明
PDR	软件操作概念 SDS	软件应用场景和操作概念
	软件需求设计 SPDS	软件需求, 由“禅道”内导出, 需求简要说明
CDR	软件需求设计 SPDS	软件设计, 由“禅道”内导出, 包含各需求的设计方案、流程图等
	人机交互界面概念	界面的草图
	测试需求 STPS	软件测试用例, 由“禅道”内导出
	软件 UML 模型	基于 Cordis 开发的部分, UML 静态模型和动态模型
	处方 Schema	处方文件的格式 Json
	机器常数 Schema	机器常数的格式 Json
编码	源代码	源代码文件
	处方文件	样版处方文件
	机器常数文件	样版机器常数文件
	组件测试报告	
ATP	集成测试报告	

6.3 版本控制

软件源代码的版本控制采用 Git 进行。

软件开发过程中, 可 Branch 出不同分支, 测试通过后 Merge 至 Master。

特定的版本在 Git 中打上 Tag 标签, 标签即为版本号。

软件产品的版本有一个主版本号, 其下属的组件允许有其各自的版本号。

7 软件构建

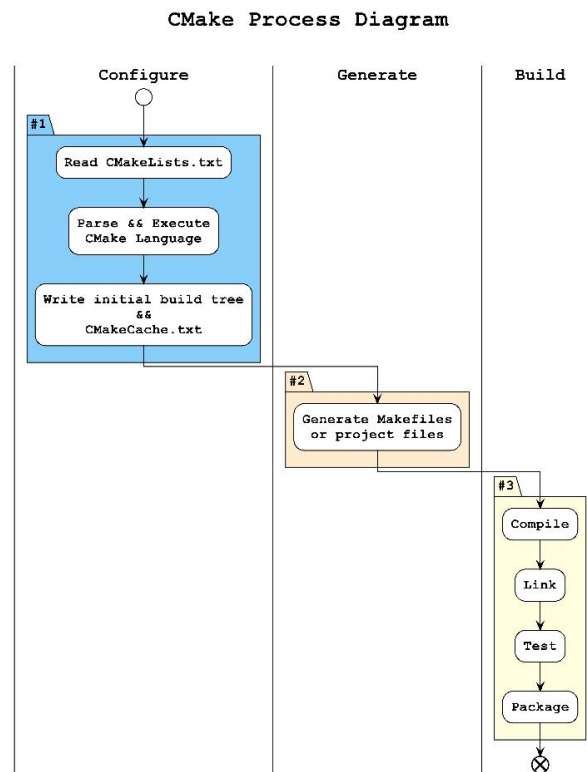



图 2 软件构建流程

	软件开发流程规范	编号：Q/ISTAR-WI-35
		版本：B
		密级：三级
		页码：第 7 页 共 13 页

公司软件采用CMake控制构建与打包过程，参见图2。其总体目标为：

- a) 统一所有 C/C++项目构建过程，提高跨项目的协作效率和复用性；
- b) 利用 CMake 的跨平台设计和交叉编译特性，简化编译和打包过程；
- c) CMake 作为持续集成/部署的重要构成，提高软件开发和迭代效率。

7.1 代码编译

代码的编译通过 CMake 脚本进行，过程如下：

- a) 通过 CMake 在 Git 仓库中下载指定版本的源代码，然后搭建成代码编译所需的目录结构；
- b) 通过 CMake 脚本生成工程文件；
- c) 调用编译器将源码编译成可执行文件，并输出至指定目录下；编译过程可由 CMake 脚本自动调用进行，不具备自动调用的特定工程采用人工方式进行。

Python 脚本和不需要编译的伪代码等，直接由 CMake 脚本下载后输出至指定目录内。

7.2 软件打包

软件的可执行文件打包成两个部分：主软件包和依赖程序包。

- a) 主软件包包含以下内容：
 - 1) 完整的软件可执行程序及其目录结构；
 - 2) ChangeLog：完整的、到目前发布版本的更新说明；
 - 3) 软件的配置文件：主要包括软件的版本信息、运行环境需求等信息，供部署工具使用；
 - 4) 数据更新工具：旧版本软件相关数据（处方、机器常数、报告等）的格式更新工具。
- b) 依赖程序包包含以下内容（部署时按需使用）：
 - 1) 所有软件产品运行所需的第三方工具、驱动等的安装程序。

8 软件测试

8.1 测试环境

软件的测试环境，根据软件的特性，分为多个层面，对应于不同的测试需求：

- a) 人机交互界面测试环境：采用消息收发的方式进行测试；分析界面发出的消息是否符合接口定义；向界面发送消息，检查界面的响应是否符合预期；
- b) 数据仿真层测试环境：对软件处理的数据进行仿真，测试软件的输入输出是否符合要求；如果待测软件为完整的软件产品，则数据仿真层为一个完整的数据虚拟机；
- c) HIL 仿真层：硬件在环仿真层，用于测试包含控制器的完整的软件系统。

8.2 代码审查

代码审查的目的：


- a) 检查代码中是否有明显的设计错误；
- b) 检查代码是否存在规范性问题。

原则上，代码提交测试前应先进行代码审查，审查方式可以会议评审的方式进行，也可使用合适的自动化工具，如 SonarQube 进行。

8.3 测试用例

对每项软件需求，均需设计测试用例对该条需求进行测试。

测试用例由“禅道”系统管理。

	软件开发流程规范	编号：Q/ISTAR-WI-35
		版本：B
		密级：三级
		页码：第 8 页 共 13 页

8.4 组件测试

具备可独立验证功能的软件组件开发完成后，进行组件测试，测试报告录入“禅道”系统。

8.5 集成测试

完整的软件产品将各组件集成在一起，在对应的测试环境中进行整体的测试，包含全部功能、部分性能以及对各类故障、异常场景的响应的测试。

测试报告作为软件产品的交付验收报告。

8.6 硬件联调

软件产品完成集成测试后，部署在实际的硬件系统上，与硬件进行联合调试，确保全部的功能性能满足产品的要求。

9 软件发布

9.1 版本规则

公司的软件产品的版本规则采用标准的Sematic Versioning模式，即三段式的版本号（Major.Minor.Patch）。版本号的每一段均由数字组成，按序列升版。版本的发布可根据实际情况，跳序进行版本升级。

各版本段的具体定义如下：

- Major：可能产生不向下兼容的重大变更，升级主版本；此段为0时，表示非稳定版本，可能会快速迭代变更；
- Minor：保持向下兼容的新功能增加，升级次版本；
- Patch：所有保持兼容性的变更，如Bug修复，功能调整等，升级修订版本。

公司的软件内部版本，可省略Patch段，在Minor段后添加a（Alpha），b（Beta）或rc（Release Candidate）和数字，表达正式发布前版本。例如：1.12b1。

版本适用于软件的最小的独立功能组件，组成一个完整软件产品的各组件可以有各自独立的版本号。

9.2 发布内容

一个完整的对外的软件发布需包含以下内容：

- 主软件包：完整的发布的软件包；
- 依赖程序包：软件正常运行依赖的第三程序。


10 软件部署

10.1 部署环境

设备的软件产品部署的目标平台参见表3，各平台的具体配置参见附录A《iStar设备软件部署规则》。

表3 软件部署平台

平台名称	说明
界面	人机界面计算机（可并入主控计算机）
主控	设备的主控计算机，设备软件的主体和所有数据均部署在此

	软件开发流程规范	编号：Q/ISTAR-WI-35
		版本：B
		密级：三级
		页码：第 9 页 共 13 页

图像	图像处理控制器，部署图像处理的应用软件，以及图像数据的存储
NAS	数据备份网络存储器
调试电脑	用于调试设备的计算机

软件在各平台上的部署目录参见表4，具体配置参见附录A《iStar设备软件部署规则》。

表 4 软件部署目录


根目录	文件夹		说明
System	iStar	Bin	主控应用程序，人机交互软件，辅助工具软件
		Diagnostics	设备问题诊断的工具软件
		Metrology	各分系统、设备的测校软件
		Runtime	软件运行需要用到的库：Qt, Opencv 等 Twincat runtime, 数据库软件, Cordis 的 MCS, 以及第三方设备厂商提供的运行程序
		SnapShot	存于硬盘上的临时文件，供系统恢复使用
		Config	设备的配置文件
		MachineConstants	机器常数
		Firmware	由主控下发的固件
		Updates	更新文件
		Locales	语言文件
		Resources	程序运行所需的图像、文本资源
		Docs	帮助文件，设备相关资料等
		License	软件 license
Data	UserData	Recipe	处方文件
		Report	各类报告文件，包括生产报告、校验报告、维护报告
		Diagnostics	输出的诊断文件
		Log	运行日志文件
		Maintenance	维护的记录文件
		Database	生产的数据，测校数据，报警、异常信息
		Images	拍摄的图像文件
	iStarBackup	System	系统盘恢复文件，iStar 应用程序完整备份
		Application	iStar 应用更新后的备份，供回滚使用
		MachineConstants	MC 更新后的备份，供回滚使用
		Recipe	Recipe 更新后的备份，供回滚使用

10.2 自动化部署工具

软件部署时配备自动化部署脚本，其主要功能如下：

- 检查软件的部署环境是否正常，各依赖库或第三方案程序是否完备；
- 部署或更新软件；
- 如为更新软件，将前一版本的软件进行备份，供回滚使用。

11 软件运维

	软件开发流程规范	编号：Q/ISTAR-WI-35
		版本：B
		密级：三级
		页码：第 10 页 共 13 页

软件运维的流程如下：

- 收集维护的需求，来源为售后人员反馈的客户问题、客户提出的新需求、项目经理提出的软件优化需求等；
- 由软件产品负责人将维护需求转化为新的软件需求或对已有需求的变更，提交至“禅道”系统内，并进行评审；
- 软件产品负责人制定发布计划，包含更新版本需实现的需求列表，并形成开发任务；
- 新版本软件开发完成并测试通过后，构建打包，对外发布；
- 由售后人员进行新版本软件的现场部署。

按上述流程持续进行，与软件计划、开发、构建、发布、部署环节形成闭环，维护软件产品直至软件产品的生命周期结束。

12 相关文件标准

Q/ISTAR-T0112-2022 《软件开发手册》

13 记录


无

14 附录

附录 A 《iStar 设备软件部署规则》

A.1 计算机部署


平台名称	操作系统	版本	系统工具配置	账户配置	根目录配置	备注
界面	Ubuntu/Windows	ubuntu-21.10 win10		iStar (Admin, 内部账户) User (启动缺省, 限制权限)	/Data	简配界面计算机可并入主控计算机 客户可选 Win 系统, 相关配置参考主控 /Data 为独立分区, link 除指定系统工具配置外, 可选项均不安装
主控	Windows	win10	twincat3 Cordis MCS	iStar (Admin, 内部账户) User (启动缺省, 限制权限)	c:\ (System) d:\ (Data) z:\ (NAS)?	Data 为独立分区 除指定系统工具配置外, 可选项均不安装
图像	windows	win10	Halcon Runtime			除指定系统工具配置外, 可选项均不安装

	软件开发流程规范	编号: Q/ISTAR-WI-35
		版本: B
		密级: 三级
		页码: 第 11 页 共 13 页

			OpenCV Runtime			
NAS	/	/	/	/	/	Raid 备份
调试电 脑	Windows	win10	qt-5.12 Cordis Modeler Cordis MCS twincat3			

A.2 软件目录部署


根目录	文件夹	说明	界面	主控	图像	NAS	调试
System	iStar	Bin	主控应用程序, 人机交互软件, 辅助工具软件	√	√	√	√
		Diagnostics	设备问题诊断的工具软件	√	√		√
		Metrology	各分系统、设备的测校软件		√		
		Runtime	软件运行需要用到的库: Qt, Opencv 等 Twincat runtime, 数据库软件, Cordis 的 MCS, 以及第三方设备厂商提供的运行程序。	√	√	√	√
		SnapShot	存于硬盘上的临时文件, 供系统恢复使用	√	√		
		Config	设备的配置文件		√	√	
		MachineConstants	机器常数		√	√	
		Firmware	由主控下发的固件				
		Updates	更新文件				√
		Locales	语言文件	√	√		√
		Resources	程序运行所需的图像、文本资源	√	√		√
		Docs	帮助文件, 设备相关资料等	√	√		√
		License	软件 license	√	√	√	
Data	UserData	Recipe	处方文件	√	√		√
		Report	各类报告文件, 包括生产报告、校验报告、维护报告	√	√	√	√
		Diagnostics	输出的诊断文件	√	√	√	
		Log	运行日志文件	√	√	√	√
		Maintenance	维护的记录文件	√	√	√	√
		Database	生产的数据, 测校数据, 报警、异常信息	√	√	√	√
		Images	拍摄的图像文件	√	√	√	√
	iStarBackup	System	系统盘恢复文件, iStar 应用程序完整备	√	√		√

	软件开发流程规范	编号: Q/ISTAR-WI-35
		版本: B
		密级: 三级
		页码: 第 12 页 共 13 页

		份					
	Application	iStar 应用更新后的备份, 供回滚使用	√	√		√	
	MachineConstants	MC 更新后的备份, 供回滚使用		√		√	
	Recipe	Recipe 更新后的备份, 供回滚使用		√		√	

附录 B 《iStar 软件开发技术栈》

模块	工具/语言/过程	说明
iStar 软件开发规范	禅道	禅道管理工具的使用
	Git	结合 Git 的软件开发流程
软件系统开发	系统架构	构建设备的软件系统
	操作概念	制定设备的软件操作概念
	平台化软件开发	开发通用平台软件组件
	问题诊断	软件疑难问题的诊断
界面开发	SEMI	SEMI E95
	QT	基于 QT 界面开发
	PyQT	基于 PyQT 界面开发
应用开发	SEMI	软件相关 SEMI 标准 (SEMI E**)
	UML 建模	软件 UML 模型建立
	Cordis	基于 Cordis 的设备驱动开发
	Digital Twin	数字虚拟机开发
	工厂自动化	SECS/GEM 开发
	DB	Influx 数据库应用及开发
	C++	C++应用开发
	Python	Python 应用开发
嵌入式开发	FreeRTOS	基于 FreeRTOS 的嵌入式应用开发
	Bare Metal	裸核应用开发
	FPGA	基于 HDL 的 FPGA 应用开发
	ACS	ACS 运动控制器应用开发
机器视觉	OpenCV	基于 OpenCV 的图像处理应用开发
	商用控制器	商用图像控制器的应用与开发
	前沿算法	先进前沿图像算法的预研与开发
深度学习	PyTorch	基于 PyTorch 框架的深度学习应用开发
控制算法	Controller	控制器的设计与仿真 (Matlab)
	前沿控制算法	先进前沿控制算法的预研与开发
信号处理	Signal Processing	信号处理算法的设计与仿真 (Matlab)
	前沿信号处理算法	先进前沿信号处理算法的预研与开发
Devop	系统运维	软件开发环境及系统的运维

	软件开发流程规范	编号：Q/ISTAR-WI-35
		版本：B
		密级：三级
		页码：第 13 页 共 13 页

	Script	自动化脚本、配置脚本的编制
	Code Review	代码审查
	CI	自动化持续集成的应用与开发
	CD	自动化持续部署的应用与开发