

中国人寿保险（海外）股份有限公司
研发效能平台
招标清单及技术指标

2021 年 10 月

目 录

第一章	适用范围	4
第二章	背景及目标	5
2.1	项目背景.....	5
2.2	总体目标.....	5
2.3	技术要求明细.....	7
第三章	项目任务与说明	20
3.1	项目调研和需求分析.....	20
3.2	系统部署与调试.....	20
3.3	系统上线.....	20
3.4	运维移交.....	20
3.5	项目计划.....	20
第四章	项目管理	22
4.1	项目管理方法论.....	22
4.2	项目实施组织.....	22
4.3	项目实施计划.....	22
4.4	项目质量管理.....	22
第五章	供应商责任	24

5.1	供应商团队组织要求.....	24
5.2	供应商职责要求.....	24
第六章	培训与知识转移	25
6.1	系统功能、流程及使用操作培训.....	25
6.2	系统架构、运维维护管理培训.....	25
6.3	运维开发培训.....	25
第七章	项目交付与成果物	26
7.1	项目成果交付.....	26
7.2	维保服务要求.....	26
第八章	验收	29

第一章 适用范围

本技术规范书明确中国人寿保险（海外）股份有限公司研发效能平台项目建设项目（以下简称：本项目）的基本需求，也是对供应商（以下简称：乙方）的基本要求。通过本文件规范供应商的方案设计、实施过程及成果验收交付。

本技术规范书所有内容和技术要求属于信息安全保密相关，所有参与相关商务投标的供应商不得扩散或者泄露任何相关内容。

第二章 背景及目标

2.1 项目背景

为配合中国人寿保险（海外）股份有限公司金融科技建设，进一步完善研发管理体系，全面提升公司 IT 管理能力，中国人寿保险（海外）股份有限公司金融科技中心（以下简称“金科中心”）推进开发、效能、交付、运维一体化能力建设，提升快速交付效率和质量，适应新的业务发展要求，以提升代码配置管理、持续集成、持续部署、效能管理等自动化能力，进而实现交付模式规范化、关键能力线上化、研发管理数字化、决策制定智能化，进而提高研发效能。

2.2 总体目标

本项目目标将通过建立软件工程过程的统一基础框架，全面整合金科中心相关的数据，实现项目管理、需求、开发、测试、集成和发布端到端的协同服务和研发工具支撑。引入统一项目管理、代码配置管理、代码扫描、持续集成、制品安全管理、自动化测试管理、持续部署、文档管理等新技术，进一步实现 IT 研发过程的精细化管理，进而提升研发过程效率、提高交付质量。

2.2.1 范围概述

IT 研发效能方面：搭建或引入 DevOps 技术平台，通过需求、研发、测试、发布到运维端到端的管理，建设 IT 研发效能平台，实现研发到运维的打通，缩短需求交付的周期以及提升研发的效率。

2.2.2 需求概述

本期研发效能平台需求包括如下内容：

- 1、研发工具平台建设，建设统一的工具平台包含敏捷项目管理、代码仓库管理、代码扫描、测试管理、自动化测试、制品仓库、制品安全扫描、CI/CD 流水线、知识库等，对研发团队提供统一的工具平台服务。

- 2、项目管理建设，提供敏捷项目管理能力，支持需求、任务、缺陷自主流程及特性属性配置，提供看板视图、迭代管理能力。
- 3、项目集合管理，支持多团队、多项目的项目集管理。
- 4、代码管理建设，内置 Git 与 SVN 混合仓库管理，同时支持 Gitlab、Github 等主流代码仓库的集成，将自研及厂商代码通过代码托管平台进行统一管理，形成统一代码管理规范 and 分支管理策略规范。
- 5、代码质量管理，引入代码扫描能力，代码提交后能够自动触发代码扫描，扫描结果支持定位到具体问题行，并提供问题修复建议。同时，代码扫描功能需要具有开放性，能够与外部三方扫描平台集成。
- 6、测试管理建设，包括测试用例管理、用例评审、测试计划执行、测试报告功能，提供完整的测试过程管理能力。
- 7、自动化测试建设，建设统一的自动化测试平台，支持 UI 测试、接口测试、压力测试、Mock 功能等功能，提供丰富的测试报表功能，支持自定义特性化库函数便于业务自定义接口。
- 8、制品仓库建设，建设统一的一站式管理全语言软件制品仓库，对接 CI/CD 流水线，打通认证系统，根据角色和权限实现仓库隔离，支持高可用及容灾备份。制品库可以支持独立部署，并提供制品同步、晋级、代理等能力。
- 9、应用质量建设，建设统一的代码质量、制品安全、应用依赖检测、容器镜像漏洞扫描机制与规范，提升整体开发质量与开发安全管理。
- 10、CI/CD 流水线管理，建设统一的自动构建、持续集成、持续交付流水线，内置常见 SDK 及编译组件，提供自定义插件体系，提升整体研发效率。实现兼容多基础设施管理，提供多环境应用程序的自动部署，支持自定义部署计划，部署策略，部署流程的编排、服务部署触发规则。
- 11、提供文档管理能力，支持不同层级（公司、部门、项目等）知识库、知识管理多种功能，实现团队知识的有效记录和分享。
- 12、平台用户体验人性化、架构先进、性能良好，提供成熟的私有化高可用部署方案，能够满足至少 1000 用户数使用，具备 7*24 小时服务的高可靠能力，能够支持大并发用户访问，并能够根据业务规模发展支持水

平伸缩扩容。

13、平台需要提供丰富的开放性接口，并提供定制服务能力，能够实现与 OA 系统、一体化运维系统的打通，实现业务需求到系统需求、系统需求到应用交付的全生命周期管理。

14、提供 DevOps 落地咨询及试点辅导服务，提供 DevOps 相关实践理论培训，制定 DevOps 工程实践规范，并辅导团队接入 DevOps 平台、指导团队优化研发流程等。

15、所需工具服务描述：

所需工具服务	说明
Devops 全功能平台	Devops 全功能平台（含团队管理、项目协同、代码托管[多节点高可用]、代码扫描、持续集成、制品仓库[多节点高可用]、测试管理、自动化测试、持续部署、知识库、个人工作台）
生产区制品库	生产区制品库，制品晋级同步
DMZ 区制品库	DMZ 区代理制品库，制品代理、同步
OA 系统对接	与现有 OA 系统功能进行打通
一体化运维系统对接	与现有一体化运维系统系统功能进行打通

2.3 技术要求明细

2.3.1 功能详细要求

2.3.1.1 团队管理

DevOps 平台应该支持多租户能力，支持团队与团队之间的数据隔离，让多个团队在一个平台进行 DevOps 建设。支持团队安全设置，能够保障每个团队的数据安全。

团队管理需要具备以下能力：

- 多租户：支持多用户、多团队管理，团队之间数据能够相互隔离，租户之间能够通过域名区分。
- 权限管理：支持人员、角色的定义，支持企业级组织架构的定义，能够

快速基于角色赋予人员权限。

- 安全管理:支持访问审计,包括敏感内容标记,访问日志、操作日志等,并可以导出下载。
- 度量:支持团队仪表盘管理,可灵活配置团队项目仪表盘,自定义配置统计维度及统计周期,支持团队及项目的可视化效能度量。
- 团队目标管理:支持团队目标管理,并可将团队目标关联到团队项目中,及时跟进团队目标的完成情况。
- 项目文档模板:支持项目文档模板设置,包括需求模板、任务模板、缺陷模板、子工作项模板、版本发布模板、知识库模板、合并请求模板、发布单模板等。

2.3.1.2 项目协同

项目协同是 DevOps 平台的重要功能模块,承载前沿的敏捷&精益方法论,助力企业敏捷转型,为敏捷开发团队提供简单高效的开发协作服务,包含多项目管理、敏捷迭代、需求管理、任务管理、缺陷跟踪、测试管理、文档管理、看板、报表统计分析等功能。

项目协同需要具备以下能力:

- 项目模板:能够基于模版快速创建传统经典项目和新型敏捷型项目,支持两种协作模式的切换;支持按照项目模板来创建项目,创建后也支持按需开关项目功能模块。
- 项目管理:敏捷型项目管理,支持迭代管理、迭代报表、史诗管理、故事点评估等功能。传统经典项目管理,支持计划管理和甘特图模式;项目管理能够和代码管理、测试管理深度集成与联动。
- 项目集管理:支持多团队、多项目的协作管理。
- 迭代管理:支持迭代管理功能,包括迭代定义、时间及目标管理,并提

供迭代执行情况的概览和统计能力；支持将需求/任务/缺陷纳入迭代管理，支持迭代的开启和关闭。

- 版本管理：版本管理是定期发布的需求集，犹如一辆定时开往目的地运载需求的列车。版本内容可在需求确认或需求开发中确定。版本管理系统，通过可视化需求、缺陷和项目进度，使版本交付流程更加可控和可追溯。版本管理支持版本基线管理，基线包括制品基线和集成基线。
- 业务需求管理：支持史诗的创建、修改、拆解、关联、删除、导入、导出等功能；支持评论、引用、附件上传/下载、关联代码提交记录、变更日志查看及历史版本回溯功能，可以查看到完整的事项变更情况及历史内容；支持业务需求和列表形式管理。支持与 OA 系统打通，实现业务需求的串联。
- 功能需求管理：支持需求的创建、修改、拆解、关联、删除、导入、导出、状态流转等功能；需求评估支持用户故事点、工时多种评估模式；支持评论、引用、附件上传/下载、关联代码提交记录、变更日志查看及历史版本回溯功能，可以查看到完整的事项变更情况及历史内容；事项支持以树状、平铺、看板等多种形式展现，看板模式下可以拖拉卡片改变状态，看板支持用户自定义配置。
- 任务管理：支持任务的创建、修改、拆解、关联、删除、导入、导出、状态流转等功能；支持评论、引用、附件上传/下载、关联代码提交记录、变更日志查看及历史版本回溯功能，可以查看到完整的事项变更情况及历史内容；事项支持以树状、平铺、看板等多种形式展现，看板模式下可以拖拉卡片改变状态，看板支持用户自定义配置。
- 缺陷管理：支持缺陷的创建、修改、拆解、关联、删除、导入、导出、状态流转等功能；支持评论、引用、附件上传/下载、关联代码提交记录、变更日志查看及历史版本回溯功能，可以查看到完整的事项变更情况及历史内容；事项支持以树状、平铺、看板等多种形式展现，看板模式下可以拖拉卡片改变状态，看板支持用户自定义配置。

- 项目设置：支持项目级用户组、用户及多元化的权限管理，支持自定义属性、工作流和状态，支持项目级分类标签、模块和模板配置。
- 第三方集成：支持第三方应用可以针对具体项目进行相应集成，需提供集成开放接口。

2.3.1.3 代码托管

代码托管是面向软件开发者提供的代码托管服务，包括代码克隆/下载/提交/推送/比较/合并/分支等。代码托管服务采取更细粒度的权限管控及企业级安全防护策略。分支保护防止分支被其他人提交或误删；IP 白名单控制访问区域，拦截不合法的代码下载；研发数据加密传输和存储，确保 7×24 小时高可靠性。

代码托管需要具备以下能力：

- 支持 SVN、Git 两种仓库类型，单项目下可以创建多个仓库，并可针对用户进行指定仓库授权。
- Git 仓库访问设置支持 HTTPS、SSH 两种，并支持访问方式的随时开关，Git 仓库支持设置私有或公有。
- Git 仓库支持部署公钥配置，可针对指定代码仓库，设置拥有只读或者读写权限，提高持续集成和持续部署阶段的代码访问安全性。
- Git 仓库支持文件或者目录的锁定功能，保护指定文件或目录不被别人错误修改。
- Git 仓库支持分支管理，可是设置保护分支规则、分支管理员、合并请求数量、CI 状态检查、推送及合并规则。
- Git 仓库支持版本标签管理，编写版本发布日志，支持关联资源（事项、文件等）。
- Git 仓库支持合并请求及源文件 Diff 比对，支持源码高亮显示，可以指定合并评审人、关联事项，支持代码行内评论备注。

- Git 仓库支持代码评审，能够设定完善的代码评审机制，代码评审既能够和项目管理过程联动，也可以和后续的持续集成过程联动。
- Git 仓库支持代码的在线查看、新建、修改、删除、下载等操作。
- Git 仓库支持合并请求的基础设置，包括是否默认删除源分支、是否默认以 Fast-Forward 模式合并、合并请求源分支有新提交时自动取消合并授权、合并模式选择设置、默认目标分支等。
- Git 仓库支持直接配置代码扫描能力，选择扫描分支及关联扫描方案。
- 提供高可用的代码托管服务，代码在存储的时候，能够进行分片存储，确保在平台部分服务器宕机的时候，代码仓库能够正常提供服务。

2.3.1.4 代码扫描

支持 Java、JavaScript、Web、CSS、C++、Object-C、Kotlin、.Net、Swift、PHP 等业界主流开发语言，可在线进行多种语言的代码静态检查、代码架构检查、代码安全检查、编码问题检查、质量评分、代码缺陷改进趋势分析，辅助用户管控代码质量。

代码扫描需要具备以下能力：

- 支持创建代码扫描方案，可按需灵活选择扫描规则，平台需内置主流语言的常见扫描方案以供开箱即用。
- 代码扫描支持统计圈复杂度、重复代码、代码统计、问题概览及列表查看、问题代码定位到代码行和责任人，问题列表支持 Excel 格式导出，问题代码需要有相应的问题检查依据和修复建议。
- 代码扫描支持全量或者增量触发，触发方式支持手动、代码提交触发、代码合并触发等多种形式。
- 支持扫描结果查看及扫描历史的长期记录，可以查看具体的扫描日志，扫描结果可以配置通知人及通知问题类型。

- 支持设置代码扫描的质量门禁，可配置通过质量门禁的代码才允许代码合并。
- 平台具备开放性，能够支持外部第三方扫描工具的集成，支持用户自定义检查规则集。

2.3.1.5 持续集成

持续集成与代码托管无缝对接，为用户提供配置简单的混合语言集成与构建平台，实现代码拉取、代码编译、单元测试、代码扫描、代码构建等自动化过程，支撑企业实现持续交付，缩短交付周期，提升交付效率。

持续集成需要具备以下能力：

- 流水线：支持高可用、高稳定、可视化编排的流水线，支持流水线导出、导入，支持流水线配置的对比，差异部分可高亮显示。
- 可视化流程配置器需内置常用的插件，可以满足日常的绝大部分使用场景，包括单元测试、制品上传、人工确认、执行脚本、代码静态扫描等。
- 构建节点：支持构建节点分组管理，构建节点操作系统支持 Linux、Windows、MacOS，支持节点编译历史记录的查看。
- 构建环境：构建环境支持虚拟机、Docker 镜像、自定义 Dockerfile 的构建方式，内置常见的构建环境，语言栈包括 Android、Java、golang、Python、nodejs、ruby、.netcore、erlang 等，构建工具包括 Maven、Gradle、Helm、gitlfs、sshpass、yarn、composer、docker- composer、docker、pip3 等。
- 触发方式：触发规则需要支持手动触发、代码变更触发、定时周期或单次触发、API 触发等多种方式，并支持配置自动取消队列中等待构建且修订版本号相同的构建任务。
- 构建快照：构建任务支持构建快照查看，包括启动参数、环境变量、流程

配置文件等。

- 流水线执行结果统计分析:支持流水线执行结果统计分析相关功能支持,可以实时查看构建日志和当前阶段。
- 构建模板:支持构建模板创建和复制;创建构建任务时,支持从模板创建和自定义创建。
- 制品库集成:支持与制品库集成,将制品上传到制品仓库,支持自动部署。
- 平台具备开放性,提供插件体系,可以自主定制开发流水线插件。

2.3.1.6 制品仓库

制品仓库是 DevOps 实践中的重要组成部分,通过制品库能够统一的管理软件制品及其版本管理。制品管理是对软件开发过程中生成的产物的管理,一般作为最终交付物完成发布和交付。

制品管理需要具备以下能力:

- 多类型支持:需要支持 Generic、Docker、Maven、npm、PyPI、Helm、Composer、NuGet、Conan、Cocoapods、RPM 等多种制品类型;制品库支持进行代理源和可信软件源的配置,支持版本覆盖策略,Maven 支持 Snapshot 快照策略及拉取优先级。
- 权限管理:支持权限管理配置,包括项目内、团队内、公开,可支持对不同成员设置对制品的操作权限;支持制品库认证,通过令牌凭证或账号密码访问,提高制品安全性。
- 制品下载:支持在页面上进行制品库上传、下载、删除、禁止下载、设置为发布等操作。
- 制品扫描:制品库支持安全扫描能力,支持扫描结果的报告展示,对安全漏洞根据重要程度进行分类,并提供漏洞修复建议;支持制品扫描方案

的配置和管理，扫描规则配置包括漏洞等级、CVE 漏洞白名单、质量红线，并可强制设置不允许使用未通过安全扫描以及扫描未完成的制品，确保制品的安全可用性；制品扫描支持制品更新后自动触发和手动运行触发，方案应用时可以针对满足条件的制品库生效，以实现扫描的灵活定义。

- 元数据管理：支持制品的元数据管理，可以在持续集成流水线中自动生成制品元数据，也可以手动维护。
- 依赖追溯：制品库支持依赖追溯，通过分析系统中制品的依赖关系，协助用户找出系统中关键制品的使用情况。
- 独立部署：制品库能够独立部署，提供多点复制功能，支持多数据中心制品分发，包含全量和增量同步。

2.3.1.7 测试管理

测试管理是面向开发测试团队提供的一体化测试管理服务，覆盖测试需求、用例管理、缺陷管理，多维度评估产品质量，帮助用户高效管理测试活动，保障产品高质量交付。

测试管理需要具备以下能力：

- 测试概览：支持测试概览查看，包括项目活跃度概览、我的待办等。
- 用例管理：支持测试用例管理，包括分组管理、关联需求、附件上传、用例历史版本追溯及恢复；支持用例导入、导出，支持 Excel 或其他文档格式。
- 用例评审：支持用例评审功能，可以指定多位用例评审人员，支持评审意见填写。
- 测试计划：支持创建测试计划，可以关联迭代、勾选测试用例范围、指派测试人员，支持测试计划的执行、复制、归档、删除；支持测试计划执行

过程中，可以记录测试结果、发表注释，也可以快速创建问题缺陷。

- 测试报告:支持测试报告管理,可以设置测试报告模板,定期生成测试报告发送给指定的关注人;也支持手动创建测试报告。

2.3.1.1 自动化测试

自动化测试管理是质量自动化基础设施之一,软件开发者提供的一体化自动化测试管理服务,覆盖 WebUI 自动化测试、接口自动化测试、压力测试,多维度评估产品质量,提高应用整体交付的质量。

WebUI 自动化:提供在线脚本录制、在线维护、公共脚本、数据库读取、脚本调试和维护等功能。支持在 UI 自动化执行过程中完成数据准备和数据清理。

接口自动化:提供在线编写的接口用例的自动化测试产品,支持 HTTP、Dubbo、Websocket 等多种接口测试。支持在接口自动化执行过程中完成数据准备和数据清理。

压力测试:需要能够提供可视化性能测试和监测等功能,创建并模拟大量用户并发访问业务的场景,紧密结合监控、流控等,收集性能数据并提供丰富的汇总报告。

Mock 功能:需要能够 Mock 各类第三方环境,支持在线配置 Mock,支持 HTTP/S、Socket、Dubbo 等多种协议。提供在线灵活的 Mock 服务。

测试数据管理:支持测试数据的管理。

2.3.1.2 持续部署

持续部署服务提供可视化、一键式部署服务,支持并行部署和流水线无缝集成,实现部署环境标准化和部署过程自动化。

持续部署需要具备以下能力:

- 应用管理,在持续部署模块中可以直接管理需要部署的应用。

- 部署环境管理，在持续部署模块中，能够管理应用的部署环境，能够管理的环境类型包括物理机/虚拟机环境，容器集群环境，各类云环境等等。
- 环境管理通道支持横向扩展，需要保留对接后续其他基础资源的通道。在底层基础资源变动的时候，可以无缝的扩展环境类型。
- 部署策略：支持常见的部署策略，比如灰度发布、蓝绿部署、滚动部署等等。
- 部署流程：支持自定义部署流程，部署流程中的各个动作支持脚本的方式定义和执行。
- 部署审批：部署到生产环境支持审批，需要通过特定权限的人员审批之后，方可部署到测试环境。
- 部署触发：支持多种部署触发方式，包括手动触发、定时触发、持续集成任务触发、WebHook 触发等等。
- 部署窗口：支持定义部署窗口，预先设置好部署活动。待到达部署窗口时间的时候，经过审批后部署到生产环境。
- 平台具备开放性，可以支持与一体化运维平台衔接打通，实现应用的交付上线。

2.3.1.3 知识库

团队知识库是团队成员进行知识共享的专属空间，为团队提供信息整理与知识沉淀服务，使组织中不同粒度的团队都能通畅地进行信息共享与协作。

知识库需要具备以下能力：

- 支持知识库的在线创建、查看、编辑、预览、权限设置等；
- 支持知识库的导入、导出、分享功能，分享时支持选择是否分享子页面、公开访问还是团队内访问，也可以设置访问权限；

- 支持知识库的历史版本追溯，可以恢复至指定版本；
- 删除知识库时，支持回收站功能，以便意外删除而需要恢复数据；
- 支持知识库模板的创建和管理，以便统一文档使用规范；
- 支持知识库以列表形式或目录树形式展现，可自由选择切换；
- 支持公司、部门、项目等不同级别文档共享及知识库功能；
- 文件管理用于存储团队中的文件资料，可以支持所有文件格式，包括 office、视频、图片等，文件能够进行分享、预览，设置访问权限。需要具备以下能力：
- 支持目录/文件的创建、删除、重命名、下载、移动、复制等功能；
- 支持文件的历史版本追溯，可以恢复至指定版本；
- 支持文件的添加星标、分享等功能，分享时支持外部分享和团队内分享，并支持设置访问权限；
- 支持文件回收站功能，以便意外删除而需要恢复数据。

2.3.1.4 个人工作台

支持个人信息的维护管理，支持个人提醒；提供丰富的 OpenAPI 接口，方便实现数据交互和系统集成接入；支持个人访问令牌配置，可以设置操作权限以及 IP 访问白名单；可以查看相关的工作事项、合并请求、构建记录、待发布确认等，提供个人便捷办公能力。

2.3.2 非功能要求

1.1.1.1.1 国产化要求

为响应国产化的建设方针及目标，要求乙方提供的研发效能平台必须为国产化产品，并提供相应的软著以及知识产权相关文件并公司盖章证明。

1.1.1.1.2 高可用要求

平台需支持分布式高可用架构，并且支持跨数据中心容灾架构，当任一中心发生故障时，另一中心能够切换并接管全部功能，并且需要提供相应的实际落地应用案例。

1.1.1.1.3 性能要求

系统性能指标的设计应满足业务量出现跨跃式增长的要求，满足未来 5-10 年的业务发展要求。

系统结构设计和数据库设计合理，具备较高的系统处理能力。资源占用率低，能够将系统资源消耗情况控制在较低程度，避免过多的数据冗余。

并发在线用户数量：至少支持 1000 在线用户。

平台本身具备 7*24 小时服务的高可靠能力。平台支持热升级能力。平台需提供自管控能力，如自恢复性，平台本身的发布可视化，平台组件可以进行监控等。

DevOps 平台支持私有化部署方式。

1.1.1.1.4 安全性要求

系统要配置和构建可靠实用的安全策略和机制，保证系统安全和数据安全，满足甲方和相关监管部门对安全和风险管理的要求。

根据不同的业务要求和应用处理方式，设置不同的安全措施和环节，满足业务管理、业务授权以及流程管理中的业务安全性要求。

对应用系统用户采用身份验证和权限控制，包括机构、角色、用户组、用户、权限管理等，用户登录认证，具有完善的安全机制以防止非授权的访问和使用，具有完善的机构、用户、角色、权限管理体系以确保客户信息能够在授权范围内合理地使用。

敏感数据信息(密码等)须加密存储，不能以明文方式保存。

系统必须具有严谨周密的安全体系结构，必须能够提供有效、全方位、多层次的安全机制，抵御可能产生的恶意攻击和病毒侵蚀，并且在运行安全、网络安全和应用系统安全等方面有合理可靠的策略。

系统必须满足护网的安全要求，应积极根据护网安全要求对系统进行安全加固。

1.1.1.1.5 自主性要求

为打破过往受厂商约束的技术壁垒，本着双方合作共赢的态度，乙方所提供的产品需有足够的灵活性让甲方能否自主把控产品技术，并向合作共研的方向努力，因此要求乙方提供的产品需具备让甲方自主开发的可能性，应具备让甲方维护人员也能掌握的足够低门槛的前后端开发框架，以及所采用的开发语言应偏向主流开发语言。

1.1.1.1.6 生态开放要求

产品开放性良好，需要提供丰富的 OpenAPI，易于集成。

第三章 项目任务与说明

本项目，分为项目调研和需求分析、系统部署与调试、系统上线阶段、项目总结及上线后运维移交阶段。乙方各阶段所执行的主要工作任务和内容包括但不限于如下：

3.1 项目调研和需求分析

- 项目启动会议，建立项目计划；
- 系统环境及外围系统关联性调研；
- 软、硬件资源的建议及项目各项需求的建议。

3.2 系统部署与调试

- 按照甲方流程进行调试；
- 妥善处理调试部署后的各种问题。

3.3 系统上线

- 按照甲方流程进行系统上线；
- 对系统上线后的使用问题进行跟踪排查；
- 对系统进行优化调整。

3.4 运维移交

- 对项目进行总结，整理并交付所有要求的项目文档、技术材料；
- 建立针对所部属内容的安装、配置、监控、巡检、日常运维、应急等运维体系；
- 出具功能方面的成果报告；
- 安排本项目的知识转移；
- 对于由乙方在项目实施过程中负责部署的内容，如果在部署后出现问题，由乙方负责进行解决。

3.5 项目计划

- 1、引入研发效能工具，构建公司级在线 DevOps 体系，试点 OnePartner 产

品线研发、测试、发布端到端全流程线上全自动化贯通。

2、建立研发效能平台，与一体化运维平台对接，形成研运流程闭环，同时增强自动化测试、安全检测、性能测试等自动化能力；

3、强化研发效能平台服务，完善数据监控，增加埋点分析，增强数据分析度量及可视化，产出研发管理看板及报告。

第四章 项目管理

4.1 项目管理方法论

乙方须依据自身的项目管理方法论，结合甲方项目实际情况，提出合理优化的适应本项目的项目管理方法论，并在项目实施过程中, 包括项目启动、执行与控制、收尾等阶段贯彻执行。

4.2 项目实施组织

为有效保证基础架构自动化运维平台建设项目的顺利实施, 乙方应根据项目实施要求派驻足够的稳定的技术人员到甲方现场进行项目实施, 并且在现场期间遵守甲方的工作纪律和要求。同时, 甲方将根据项目实施需要, 组织相关领导、业务人员、技术人员参加此项目, 与乙方人员紧密结合成项目实施小组。

乙方应提出详细的项目实施组织建议。项目实施组织的具体形式、人员组成及分工由双方在项目启动阶段根据项目实施需要协商决定, 并报甲方批准执行。在具体项目实施各阶段, 项目组可根据需要, 提出项目实施组织或/和人员组成变更申请, 经甲方批准执行。

4.3 项目实施计划

本项目中, 乙方需要按照甲方要求的时间点安排实施计划, 按时完成项目相应内容。

4.4 项目质量管理

质量控制队伍应该独立于项目组, 作为项目成功的重要因素和保障, 在项目执行过程中始终与项目组保持密切联系。从另一个角度观察和监督项目的开展, 帮助项目组发现和解决项目执行中的问题, 确保项目的成功。

在组织结构上, 将设置质量控制小组, 专门负责项目实施的质量控制。质量控制小组由项目双方项目经理、业务人员及双方领导组成, 乙方专门指定一位高级经理作为质量总监。

质量管理小组将定期或不定期的举行检查会议, 进行项目各阶段提交文件的评审, 听取项目经理及项目其他人员的汇报, 对项目的进展和质量进行监督和控制, 提出有关建议和意见。

质量管理小组的每次会议的内容，结论和决定，都将形成文字备忘录，为项目文档保存，并作为以后工作的依据，随时对项目的进展和质量进行修正。

第五章 供应商责任

供应商须确认并承诺能够完成本技术规范中所要求的全部项目内容以及按要求组织项目团队，确认并承诺承担本技术规范书中所要求供应商承担的所有责任。

5.1 供应商团队组织要求

为保障项目以及产品落地成功，同时为了确保乙方提供的技术服务响应的及时性、有效性，要求乙方具备 100 人以上技术人员规模的本地团队，需提供本地化团队社保证明，拟参与本次项目的团队成员还应提供个人简历以及本地项目的建设经验。

有至少 3 家及以上大型金融企业、保险企业的服务实施案例。

5.2 供应商职责要求

供应商须对项目的全部内容进行应答，按照本技术规范要求提交全部资料，并对本技术规范各方面做出实质性响应（确认或提出异议）。

- 1) 对本项目范围、内容、所承担的任务的理解与确认；
- 2) 本项目管理方案建议包括但不限于：
 - 设计项目组织结构，明确双方职责和主要任务；
 - 详细工作进度计划/策略，包括对工时、人力、费用等资源需求的预期；
 - 拟提交成果清单及说明；
 - 培训方案；
 - 项目个人简历及资质证明；
 - 项目验收标准；
 - 项目实施经验；
 - 项目实施组织、工作职责。

第六章 培训与知识转移

在项目中，为确保项目的可持续性发展，保证从设计阶段到实施阶段和上线维护的平滑过渡，以及降低不同阶段过渡过程的不确定性和可能的执行偏差，通过多种方式提供项目技术培训。

知识转移是将专家的特定能力转移到项目组的客户成员，并最终转移到客户整个组织。根据本次项目总体内容和时间进度要求，乙方安排有关项目方法、项目成果等方面的培训，相关知识培训内容包括：

6.1 系统功能、流程及使用操作培训

面向平台用户的培训，目标是使平台用户快速熟悉系统的功能特点，熟练掌握相关的功能操作及使用流程。

至少提供 3 场次的培训，不限参与人数，由乙方安排经验丰富的培训讲师在项目交付阶段或上线阶段完成培训。

6.2 系统架构、运维维护管理培训

面向平台系统架构及运维维护管理的培训，目标是使平台运维管理人员能快速熟悉系统的部署架构、所用技术及组件、日常运维管理、维护、故障排查等技能。

至少提供 2 场次的培训，且不限参与人数，由乙方安排经验丰富的培训讲师在项目交付阶段或上线阶段完成培训。

6.3 运维开发培训

面向技术人员及运维开发人员的培训，目标是让相关技术人员能够掌握系统的设计思想、系统架构、开发框架，具备独立的系统设计、应用开发的能力。

至少提供 2 场次的培训，不限参与人数，由乙方安排经验丰富的运维开发谁培训讲师在项目交付阶段或上线阶段完成培训。

第七章 项目交付与成果物

7.1 项目成果交付

为了确保项目实施的成功，在项目中的各级交付文件非常重要，乙方需要按照双方约定的格式提交项目相关文档。

为确保项目按照项目预期的计划执行，在项目实施过程中，一些重要文件需要甲方项目经理或相关人员进行审批和确认。

在项目各阶段交付物内容与要求（包括但不限于以下内容）：

- 项目实施方案；
- 项目设计方案；
- 项目实施手册；
- 项目总结报告；
- 系统测试报告；
- 系统运维管理手册；
- 系统应急手册；
- 工作遗留项及关注事项；
- 技术问题处理记录。

7.2 维保服务要求

乙方在项目终验后需提供至少 2 年的维保服务，包括平台升级服务以及技术支持服务，具备如下：

7.2.1 升级服务

乙方在项目维保服务期内提供免费平台升级服务。

7.2.2 技术支持服务

在项目维保服务期内，当运维平台出现问题时，乙方提供电话技术指导、邮件指导或远程技术支持，如发生紧急故障，乙方提供现场紧急技术服务，对现场服务工程师未能解决的问题，乙方需积极组织相关技术人员，包括二线专家，分析问题，制定解决方案，直至问题解决。

具体技术支持服务内容及要求如下：

服务范围	服务内容	服务人员及时间	支持方式	服务响应时间	完成时间(含排除故障)
常规技术支持服务	1. 提供本系统相关的技术支持、业务咨询服务； 2. 负责软件的日常管理、巡检、监控、优化等运维工作； 3. 系统相关的缺陷修复工作。	运维工程师 工作时间 58h	电话、邮件或微信沟通、远程技术支持	10 分钟以内	8 个小时以内
升级服务	1. 提供软件产品兼容版本的免费升级服务。	运维工程师 工作时间 58h	电话、邮件或微信沟通、远程技术支持	10 分钟以内	24 个小时以内
健康巡检服务	1. 提供定期的系统性能评估和健康检查, 查找、分析存在的隐患, 并提供相应的服务以消除隐患。	运维工程师 工作时间 58h	电话、邮件或微信沟通、远程技术支持	10 分钟以内	24 个小时以内
配合支持服务	1. 对本系统相关联的其他系统升级、例行维护、变更等提供相应的配合支持服务； 2. 特殊时段(春节、国庆节、年终、重大应用系统测试、投产、灾备演练等), 以及产品安装、硬件升级、操作系	运维工程师 工作时间 58h	电话、邮件或微信沟通、远程技术支持、现场支持	10 分钟以内, 到达现场时间为响应后 1 小时内	24 个小时以内

	统或中间件变更、迁移、升级时等的现场支持服务。				
紧急技术支持服务	1. 提供处理各种突发事件的应急预案。	二线专家 724h	电话、邮件或微信沟通、远程技术支持、现场支持	10 分钟以内，到达现场时间为响应后 1 小时内	4 个小时以内（按到达现场起算）

第八章 验收

本项目的验收标准为：乙方完成研发效能平台建设工作，完成场景功能的实施，确保所建设的平台满足非功能要求，平台平稳运行，完成研发过程流程体系建设，完成建设行业内 Devops 最佳实践，完成项目成果交付以及知识移交。

符合验收标准后，乙方须提前十四天用书面方式向甲方提出验收申请，由甲方组织项目验收小组对项目成果物进行验收，如验收通过，则由甲方出具最终验收证书。

如果双方对产品的质量、完成情况有意见分歧，甲方可委托权威机构对产品进行重新鉴定，此机构出具的鉴定证明即作为甲方向乙方提出修改、补齐、更换和索赔的有效证据。乙方除承担修改费用之外，还需承担重新鉴定费用。

上述各验收阶段均与付款直接相关，具体条款经甲乙双方商定后，体现在具体合同文本中。