

# 甲方软件质量保证体系

## 1. 项目背景

随着信息化技术对业务的支撑能力的增加，企业都通过不断推进信息化建设来服务于流程创新和战略实施。

软件是信息化的核心，软件交付质量是信息化战略实施的核心问题之一。由于各个企业会不断通过新的软件项目来支撑业务运营，因此必然有很多软件项目会外包给软件开发商来完成。然而由于目前大多软件开发商的成熟度较低，双方又缺乏好的软件外包质量保证体系的指导，因此导致诸多的软件质量问题：

- 根据软件开发商安排的需求分析过程，部分真实的需求无法被提取和实现。
- 对软件项目的过程缺乏了解和控制，对其中出现的风险无法做出合理应对。
- 软件交付时都存在很多质量和进度隐患，甚至部分项目实施过程中就陷入僵局。
- 软件交付后又由于多个承包方提交不同形式的交付件，难以统一管理。并且软件资产的维护成本很高，却无法充分的利用。
- 业务部门无法保留所有软件的验收测试环境和平台，一旦某些系统进行了升级改版，就无法对此软件进行整体的质量检测和保证。

信息化是很多企业的业务基础和核心竞争力之一，因此如何通过软件质量保证，来确保每个软件项目都能够获取成功的软件系统就成为提升企业核心竞争力和行业优势的最重要方法。

广州畅志信息技术有限公司在相关方面具有丰富的经验和积累。我们具有专门的软件质量保证体系，能够帮助甲方提高软件项目的实施质量，促进软件资产的管理和持续保证软件质量。并采用软件资产管理方式提升项目成果的重用价值。从而提高软件资产的投资回报。

## 2. 质量保证体系内容

甲方软件质量保证体系是专门用于指导甲方参与并主导软件质量保证的各个环节。通过建立沟通协作平台，覆盖软件生命周期的质量管理，建立全程测试能力三大机制，甲方就可以形成完整统一的软件质量保证平台（如图 1）从而帮助甲方提高软件项目的质量。

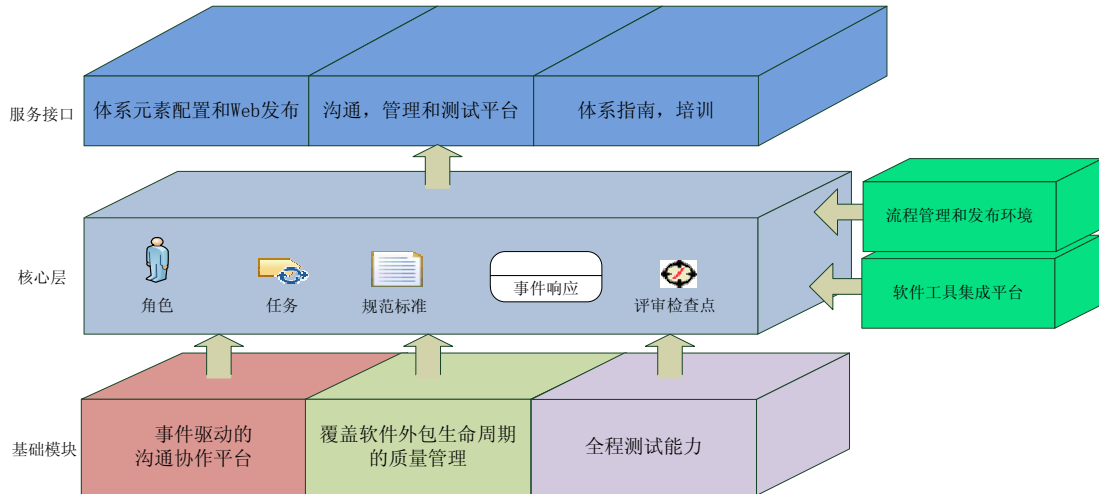


图 1 软件质量保证体系总架构

## 2.1. 三个基础模块

软件质量保证体系的基础模块部分包含三个模块，这三个模块是整个软件质量保证体系的理论基础和咨询指导。

- 沟通协作平台是由软件平台和相应的沟通流程，信息传递标准构成，是软件项目相关各方沟通的统一接口和方式。
- 覆盖软件全生命周期的质量管理包括配置管理软件等工具搭建的软件资产管理平台和各种里程碑检查点的定义和相关操作流程，指导文件。
- 帮助客户建立全程测试能力，使甲方有能力在项目各阶段进行关键成果的检验和测试。

其中每个模块都包含了理论体系的实施，软件工具平台构建和流程定制等具体咨询和实施活动的指导。

### 2.1.1. 事件驱动的沟通协作平台

由于事件驱动适合应用在不可预知的和异步的环境里。在软件生命周期中，对软件需求，缺陷，发布等方面都会出现很多突发事件，这些事件又都会出发相关各方的响应动作。因此我们事先收集和定制这些事件列表，并逐一建立多方的响应流程，这样一旦事件发生，就直接触发各方响应，从而减少了不必要的沟通成本和摩擦成本。

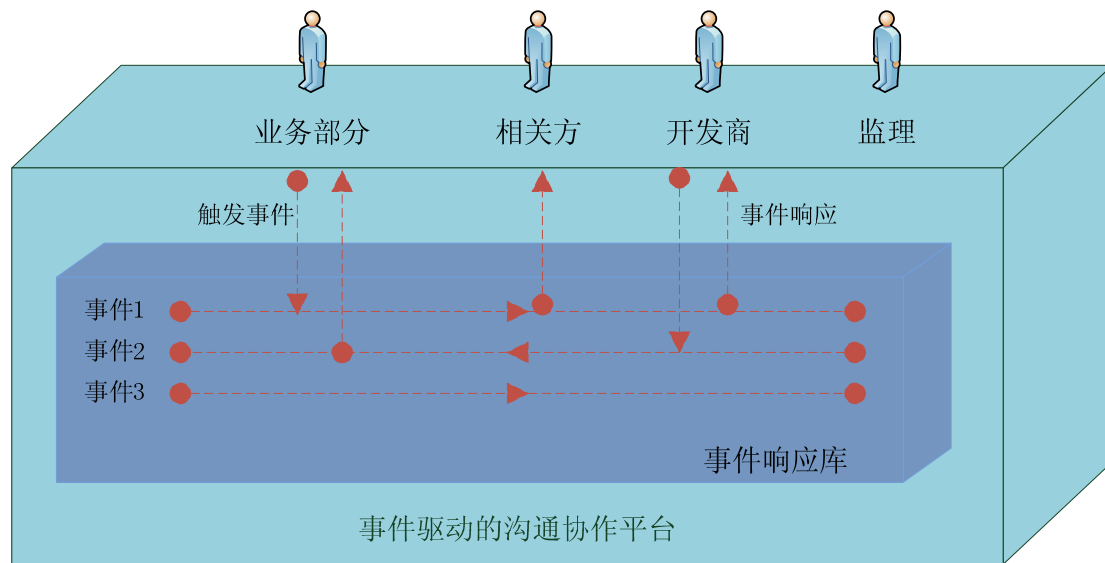


图 2 事件驱动的沟通协作平台

我们根据软件外包项目管理和软件质量保证的经验，基于企业自有特征，帮助企业定制并部署事件驱动的沟通机制。首先根据我们的相关经验，从事件库中裁减出适应本项目的大概率事件列表，并通过集成项目相关各方的事件响应方式，建立风险的识别和应对预案，以及责权清晰的高效的沟通机制。在项目各方之间建立常见事件的触发机制，然后集成项目各方的特定事件案的响应机制，为甲方需求变更，开发方架构变更，开发方进度延迟等重要的大概率事件及其缺陷跟踪等流程的高效处理提供保证（如图 2）。从而保证项目各方都能快速释放风险，共同发现问题解决问题。

同时由于事件的独立性，我们可以通过离散事件模拟，对项目的复杂情况做到足够估计和应对。

### 2.1.2. 覆盖软件外包生命周期的质量管理

在软件项目外包管理中，甲方的软件质量保证工作同样覆盖了软件外包开发生命周期的全过程。从甲方的角度看，可以将软件项目分为四个基本阶段，即项目启动阶段、项目实施阶段、项目验收阶段和项目运行维护阶段。因此企业的 IT 部门应该在对应的阶段实施软件项目需求管理、开发过程监控、软件系统验收管理以及软件后续维护开发管理，从而全面保证软件项目的成功，即在软件质量上满足要求，系统可靠运行，保证开发商按时交付系统并按时上线，进而实现投资回报最大化，在预算范围内完成系统。

为了能够对软件开发中的各种交付件和信息进行管理，并且使之成为日后的软件资产。在项目启动前，就要建立软件项目的资产管理平台，对各阶段的交付件进行统一的配置管理和变更管理。

针对每一个单独的软件开发项目，根据软件项目开发周期的特点，我们可以将软件项目开发的整个生命周期划分为以下几个阶段（如图 3）：

- 项目启动：针对项目情况，在质量保证体系基础上定出质量保证计划；与开发商确定软件项目开发的目标和范围及项目需求；管理需求基线。
- 软件开发：开发商根据软件需求设计、实现相应的软件系统；
- 软件验收：对开发商提交的软件系统进行验收，并投入生产运营；
- 后续维护：收集业务部门在系统使用过程中的反馈意见和新的业务需求，对系统进行改进以满足业务不断发展的需要。

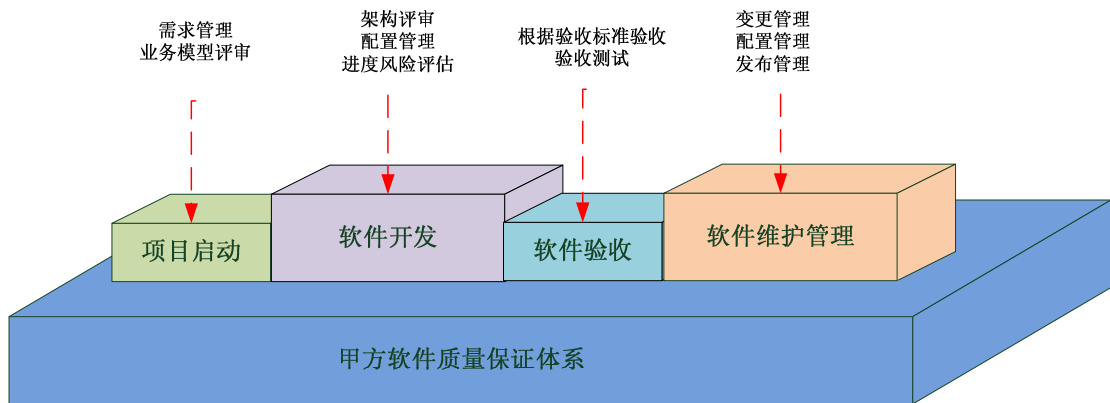


图 3 覆盖软件生命周期的质量管理

企业的软件项目由不同的开发商承接开发，所以必须有效地管理外包的多个软件项目，这些项目虽然开发目标和进度安排各不相同，但它们都是为了企业的业务发展而服务，所开发出来的系统相互配合，共同完成企业的各种业务。所以在开发的过程中，我们不仅要关心单个项目的进展情况，更要从整体的角度来综合管理这些项目的组合、统筹安排，保证紧缺的资源能够被用到最需要的项目中，保证企业的 IT 投资能够为企业的发展带来最大的回报。

针对通常的总价外包类型的软件项目，甲方质量保证的关键就是要控制好软件开发每一阶段的工作产品质量和识别并应对进度风险。

#### ● 项目启动管理

在软件项目开发的早期，首先要做的就是针对本次项目的各种属性，定制和裁减质量保证计划，从而确保软件质量保证体系能够在本次项目中高效的发挥作用。

启动阶段最主要解决的问题就是明确软件需求，但是现实中开发商往往很难理解企业的业务需求，加上业务需求会随着时间的推移而发生变化，造成软件需求一直在发生变化；另一方面，开发商提供的需求文档也很难被业务部门所理解，造成双方沟通上的障碍。因此为了确保需求的

准确完整无二义性，软件开发商应该使用业务模型和原型的方式来确定软件需求，并使用需求管理技术应对软件需求的变更。使用研讨会的方式，参与需求的讨论和评审。另外还要为需求建立甲乙双方发认可的基线。

- 开发过程监控

在传统的软件项目开发中，从需求分析开始到开发商交付最终软件系统,其中只有一次验收，很多问题要到最后的验收阶段才会暴露出来；同时也难于控制项目的进度，经常导致项目延期。造成这些问题的主要原因是外包项目开发过程不透明，企业很难监控开发的进展情况。

所以我们建议变一次验收为多次检验，在项目的开发过程中设置多个里程碑，在每个里程碑处设有检验点。这些里程碑可以参考 Rational 统一开发流程 RUP 中的四阶段模型（初启、精化、构建、移交）来设置；可以要求开发商采用迭代化软件开发，这样每一个迭代的结束都是一个技术上的检验点。最简单的过程监控可以在架构设计里程碑处（精化阶段结束）设置中期检验点，要求开发商按要求提交中期交付系统原型，对原型系统进行中期验收，主要考查系统功能、性能指标等，并确认系统是否能与生产环境中正常工作，同时该原型也有助于收集业务部门的反馈意见。所有检验信息和交付件都通过配置管理和变更管理平台进行统一管理，提高项目的可追溯性和透明性。

- 软件系统验收

当软件项目开发完成之后，就要对开发商所提交的交付系统进行验收，验收主要包括验证和确认两部分。

验证(Verification)主要是评估系统在功能上是否和需求定义相一致，检验系统在技术上是否达到一定质量标准(可靠性、性能等)；

确认(Validation)主要是评估系统是否真正满足了生产环境和业务运作的需要，这需要将系统部署到实际生产环境中去进行检验，并需要业务部门的参与。

对任何单位和企业都需要确定全面的软件验收标准，在定义需求和评估软件质量时，都需要从功能、易用性、可靠性、性能和可支持性等多个维度来全面考虑，使用可以验证的质量标准并尽量使用量化的质量指标。另外要注意的是除了验收软件系统本身之外，还应对系统开发过程中所有的中间工作产品如需求文档、设计模型、测试计划/报告等进行验收，RUP 为这些工件提供了详尽的模板和质量要求。

- 项目维护管理

在系统上线之后，项目团队面临的挑战在于如何快速响应业务部门的各种变更请求，一方面随着业务的变化业务部门会提出很多新的业务需求，另一方面在使用中发现的软件缺陷需要得到迅速修正。对于开发团队来说，他们需要同时维护多个软件版本，如不同地区的特殊业务要求

系统有多个版本，不同的运行平台也要求系统支持多个运行版本，这些复杂的版本管理需求往往导致软件版本管理上的错误。

在这个阶段，我们必须全面管理软件维护过程中出现的各类对软件产生变化的因素，包括缺陷、需求变更、设计变更以及文档变更等。加强开发厂商和用户的沟通，随时供项目管理人员掌握项目最新进展状态，实现软件开发量化管理。另外，我们继承项目开发中的配置变更管理，对所有软件资产进行统一版本管理，做好软件发布管理，避免版本错误以及重复开发。

由于不同的软件企业具有不同的软件过程成熟度和能力水平，对业务需求的熟悉程度也不同，因此，我们根据软件企业的能力和与业务的关系密切程度，将监控过程分为 3 个级别，分别建立强监测，中检测和弱监测三个等级的检测方式，其中等级越高，甲方的参与度也就越高，表明对甲方的管理的依赖程度越密切。而针对成熟度高的企业，我们只需要进行弱检测就可以保证软件质量。

### 2.1.3. 建立全程测试能力

我们帮助客户建立覆盖软件项目外包生命周期的全程测试能力（如图 4）。甲方依靠测试能力，可以在项目启动时进行选型测试和关键技术测试，从而释放风险。在软件开发阶段通过对重要成果的确认测试和重要工件的技术评审能力来完善技术风险的监控能力。在验收测试阶段，甲方可以通过验收测试能力来完成对验收测试标准的符合度检验。在项目转变为软件资产后，可以通过回归测试来不断检验资产变更后的质量。

建立全程测试能力后，能够满足多项目的各种测试环境要求，从而减少每次软件项目中验收环境搭建的成本，提高企业整体的软件质量。

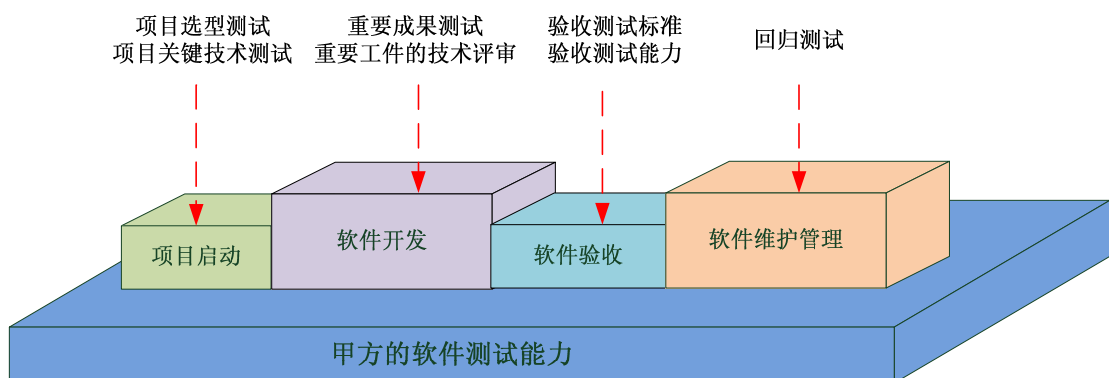


图 4 覆盖软件生命周期的全程测试能力

甲方建立全程测试能力可以通过两种方式实现，一种是甲方通过建立自己的测试工具集成平台和测试团队来形成测试能力。我们帮助甲方集成软件测试环境，并培训测试团队，制定测试团队参与软件生命周期的流程和指导。另一种是采用测试外包的方式。针对这种方式，我们帮甲

方建立对开发方，测试方的多方协调的流程，明确测试方能够在适当的时候参与质量控制。另外还会建立对测试方的管理流程，确保测试服务的质量。

## 2.2. 核心层

### 2.2.1. 集成平台

在实施质量保证体系时，需要集成的工具平台的支撑，同时也只有将体系中的各种流程都配置到统一平台中，才能支持体系的高效运转。

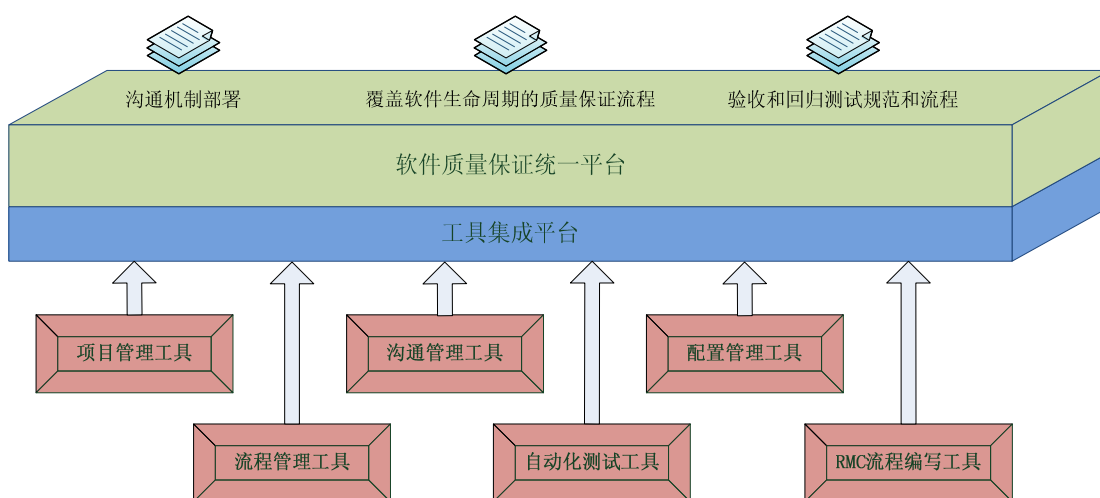


图4 质量保证的工具集成平台和流程配置

使用项目管理工具、流程管理工具、沟通管理工具、配置管理工具共同集成搭建统一的工具平台。在此平台上部署质量保证体系的所有流程、规范，形成固化的质量保证体系，保证了体系的执行效率。

使用集成自动化测试工具，缺陷管理工具、配置管理工具，构建自动化验收和回归测试平台，并部署软件的验收测试流程和规范。

### 2.2.2. 流程管理和发布

使用 RMC 的作为流程编写和发布平台，定制质量保证体系的多个插件。

建立体系是为了使软件全生命周期涉及到的各方人员都按照标准的方式进行沟通和执行。为了促进各个角色都能完整的了解自己在质量保证体系中的任务和相关的指导信息，我们采用国际标准的统一方法定义架构来描述这些机制和流程，并通过 Web 方式发布到整个团队（如图 4）。从而确保引入的体系能够高效实施。

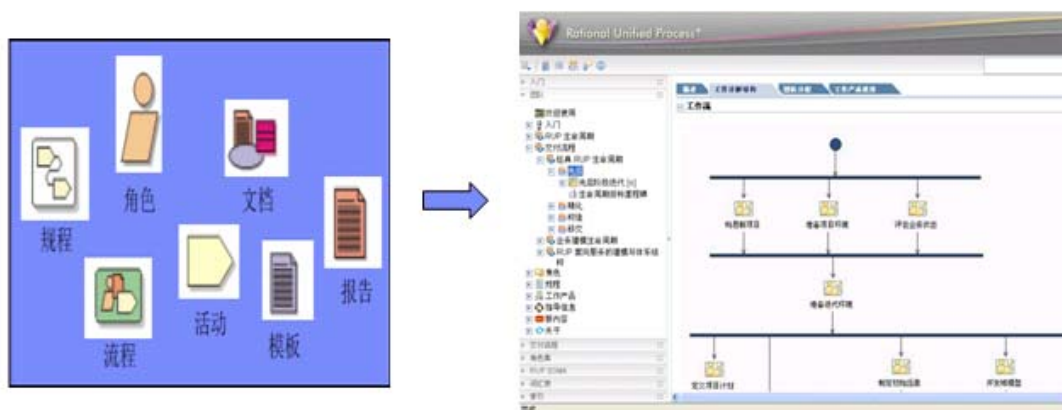


图 4 质量保证体系管理和 Web 站点

采用 IBM 提供的方法编写工具 Rational Method Composer 来完成所有流程及其角色、工作、指导规范等组件的编排和重新组装，帮助客户不断的改进体系内容，适应不同的软件管理环境。

### 2.3. 服务接口

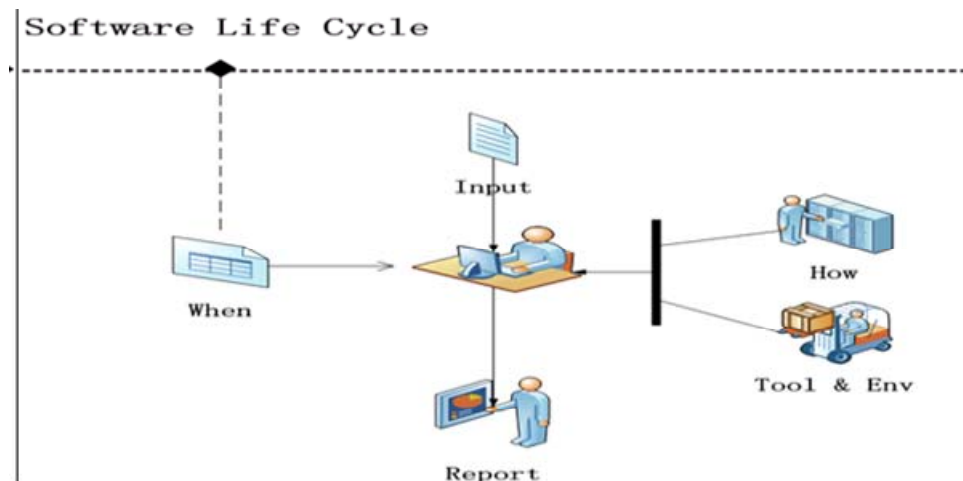
通过质量保证体系的集成实施，将形成三方面服务接口。

- 体系元素配置插件和 Web 发布的体系 Web 站点
- 部署了沟通机制的协作平台，部署了管理流程的平台，部署了测试流程的测试平台
- 各种流程，机制的文档模板，指南，培训。提高体系涉及人员的能力。

本体系覆盖的软件项目中相关人员，都可以根据从这三个服务接口中获知

项目的质量保证人员通过以上三方面功能，在项目启动时与项目的各方负责人一起定制软件项目相关的各方人员都可以根据以上三方面的用户界面，了解在不同的软件生命周期中，何时以何种方式做事，以及输入、输出、工作指导、环境工具等。并通过具体的操作环境来方便地完成自己的工作。





### 3. 体系实施

软件质量保证体系的实施，是对不同客户做具体的定制。在帮助企业引入此体系时，会按照企业的现状和实际需要进行裁减。其具体实施步骤为：

阶段	工作
访谈需求	<ul style="list-style-type: none"> <li>➤ 深入访谈各角色，记录并总结客户当前的实际的质量保证流程，工具，方法，问题等现状</li> <li>➤ 访谈并获得各方面对软件质量保证改进的期望</li> <li>➤ 以事实和现状为基础分析差距，确定改进目标，制定改进路线</li> </ul>
体系定制	<ul style="list-style-type: none"> <li>➤ 根据改进路线，为客户量身裁剪体系内容。定制出软件测试能力改进的方案</li> <li>➤ 不断与客户沟通并改进方案，直至达到改进目标</li> </ul>
体系实施	<ul style="list-style-type: none"> <li>➤ 实施质量保证体系，</li> <li>➤ 进行质量保证体系的标准化管理和快速 web 发布</li> <li>➤ 根据改进方案对相关人员进行培训和其他能力提升</li> <li>➤ 构建工具平台，并部署相关流程</li> </ul>

	<ul style="list-style-type: none"> <li>➤ 建立持续优化机制</li> </ul>
跟踪优化	<ul style="list-style-type: none"> <li>➤ 跟踪实施效果，并进行适当优化</li> <li>➤ 评估改进成果，为客户提供投资回报分析</li> </ul>

体系实施后的交付件为：

模块	交付件
质量保证体系	<ol style="list-style-type: none"> <li>1. 《事件驱动的多方沟通机制》体系文件</li> <li>2. 《覆盖软件生命周期的质量保证体系》体系文件</li> <li>3. 全程测试的相关指导和模板文件</li> </ol>
Web 网站	<ol style="list-style-type: none"> <li>1. 事件驱动的多方沟通机制的 Web 站点</li> <li>2. 覆盖软件生命周期的质量保证体系的 Web 站点</li> <li>3. 全程测试的指南</li> </ol>
工具平台	<ol style="list-style-type: none"> <li>1. 项目管理工具、流程管理工具、配置管理、沟通管理工具共同搭建的质量保证统一平台。</li> <li>2. 集成自动化测试工具，缺陷管理工具、配置管理工具的自动化验收和回归测试平台</li> <li>3. RMC 的流程编写和发布平台，及软件质量保证体系的多个定制插件</li> </ol>
培训	<ol style="list-style-type: none"> <li>1. 事件驱动的多方沟通机制的培训</li> <li>2. 覆盖软件生命周期的质量保证体系培训</li> <li>3. 质量保证体系的 RMC 管理和发布培训</li> </ol>
跟踪实施	跟踪项目实施情况，并在关键点做审查监控

## 4. 客户价值

通过我们提供的方案，客户将获取以下价值：

1. 建立了统一的质量保证规范和平台，规范开发商行为，提高项目质量。
2. 增强项目透明性，加强项目监控，减少风险。
3. 获得了多方一致的质量保证计划发布方式。多方人员可通过一致的界面了解自己的角色岗位和工作职责及工作时间。
4. 提高了企业的软件资产管理能力和可重用行，从而加强了对业务战略的支撑能力，提高了软件项目的投资回报率。
5. 持续优化内部质量保证体系的机制和能力。