

BES III 离线软件质量管理

(李卫东 2004-12-20)

1 引言

BES III 离线数据处理和分析系统的主要任务是将 BES III 探测器和模拟产生的原始数据进行离线处理，生成物理分析使用的 DST 数据，同时提供物理分析所需的其它工具。该软件系统将以欧洲核子中心 (CERN) 开发的通用高能物理实验底层软件 GAUDI 作为基础，采用世界上其它高能物理实验使用的先进软件技术，按照 BES III 实验的具体需求，以 C++ 语言为主开发的全新离线软件系统。BES III 离线数据处理和分析软件系统分为：总体框架和物理分析核心软件、GEANT4 模拟软件、刻度和数据库应用软件以及事例重建软件四个子系统。

为了实现 BES III 的物理目标，必须要有高质量的 BES III 离线软件作为保证。同时，高质量的软件系统应该是容易维护的，这也是周期为十几年的 BES III 物理实验所必不可少的。像 BES III 离线软件系统这样的大型软件项目，质量管理的重要性和加强软件管理所带来的益处已成为高能实验物理学家们的共识。

下面将通过定义 BES III 离线软件的质量标准和软件开发过程，来探讨加强质量管理的具体措施。这些措施将会帮助开发人员提高软件质量，而不是成为他们额外的约束和负担。所以在目前和将来的离线软件开发实践中，每个开发人员必须坚持实施这些质量管理措施。

2 软件质量标准

BES III 离线软件的质量标准包括以下几个方面：

2.1 设计 (Design)

定义

软件要有清晰的设计，软件实现要模块化，并和 BES III 离线软件的整体结构兼容。

说明

在设计时，不仅要考虑软件的内部结构，而且要考虑与 BES III 整体软件框架集成的方案。在设计阶段，要明确软件模块和 BES III 整体软件框架的接口。

2.2 编程 (Coding)

定义

软件开发人员要遵循规定的 C++ 编程规范，使得生成的程序代码符合下列要求：

- 没有 C++ 常见的错误
- 容易被不同的开发人员维护
- 具有良好的可读性
- 具有统一的风格
- 能被移植到不同的操作系统中运行

说明

BES III 离线软件编程规范是在 CERN SIPDER Project 研究成果基础上编制的，共有 70 余条，分必需的（必须遵循的条例）和推荐的（建议遵循的条例）两个级别。

2.3 性能 (Performance)

定义

算法或其它软件必须达到 BES III 要求的性能标准。

说明

软件性能分为技术性能和物理性能两个方面内容，技术性能包括代码的运行速度、使用内存的大小、可移植性等。物理性能包括效率、计算物理量的分辨、物理量分布的尾巴（Tail）情况等。原则上，BES III 各子探测器软件的整体性能应该和相应子探测器的设计指标一致。而且在软件性能优化时，要选取合适的效率、精度和运行速度的平衡点（trade-off）。

2.4 健壮性 (Robustness)

定义

软件能正确地、稳定地在规定的各种环境中运行。

说明

要求软件代码在正确运行的同时，能够有效地使用内存、CPU 等计算机资源，而且不能够对其它软件模块的运行产生不良的影响。

在规定的操作系统和软件环境（离线和在线环境）中，软件代码应能正确编译和运行；同时能够正确处理遇到的异常情况，即一些不正常的条件出现时不会引起程序运行崩溃。

2.5 可维护性 (Maintainability)

定义

当软件代码需要更新时，能很容易地找出需要修改的地方和做出相应的改动，同时能够很容易地检验改动是否正确。

说明

由于软件开发人员的流动性比较大，软件代码的可维护性十分重要。当软件作者离开时，要求其它人员在一段合理的交接时间后能够承担维护任务。为了达到该目标，要求软件代码的可读性好，同时软件要尽可能提供内部错误诊断和跟踪调试的工具。

2.6 文档 (Documentation)

定义

每一部分的代码都要有相应的文档，代码文档应该包括以下主要内容：

- 文档要清楚说明代码所完成的任务，以及完成该任务所使用的方法 (method) 和算法 (algorithm)。
- 文档应包括详细的设计内容。
- 文档应包含“用户指南”部分，即输入 / 输出接口描述。
- 尽可能提供可运行的例子以及运行例子所需的数据或配置文件。

说明

文档应该用中文书写。在描述软件设计时，要采用标准的建模语言（UML），设计中所使用的类和类间的关系要用 UML 语言进行正确的表示。设计的文字部分应包含软件高层概念设计的描述，UML 部分应详细描述实现细节。对已存在或从其他实验移植的 C++ 代码，要采用“逆向工程”的方法，编写设计和实现文档。

输入 / 输出接口描述应予以充分重视，文档应详细说明所有输入和输出变量的定义。同时能够提供将软件集成到现有开发环境的其它有用信息。

3 软件开发过程

BES III 离线软件是一个大型的软件系统，参与项目的人员约 25 名。项目从 2001 年开始启动，必须在 2007 年提供满足实验要求的软件。开发人员以在读研究生为主，在地理上分布在几个不同的城市和院校。

软件开发过程一般包括需求分析、设计、实现、测试和发布五个基本过程。根据项目和人员的实际情况，BES III 离线系统采用迭代增量式开发模型。同时强调开发过程中的人员的合作和沟通；要求开发人员要“拥抱变化”，聆听软件用户或团队其它成员的意见，根据共识对设计和代码不断进行改进。

BES III 离线软件项目计划已在工作分解结构中进行了详细的描述，包括人员、内容、时间和里程碑，共计 250 余项。各个子系统以半年为单位的软件任务也以 BEPC II 技术通知单的方式加以了明确。BES III 离线软件项目的活动将围绕工作分解结构和任务通知书两个核心开展。

如果软件开发人员要从事 BES III 离线软件工作分解结构中定义的工作，首先要获得子系统负责人的许可。在进行软件模块的设计之前，要分析该模块的需求、在整个系统中的位置、应该具备的功能以及和其它软件模块之间的关系。应该采用 UML 语言和 UML 建模工具（Micro soft Visio、IBM Rational Rose 等）来进行软件设计，设计要求简单并符合 BES III 离线软件整体框架的要求。设计的方案要在 BES III 离线软件会议上进行讨论，并根据反馈信息进行修改。只有获通过的设计方案才能进入下一个实现的阶段。软件开发人员

在编程时，要严格遵守 BES III 离线软件编程规范，以避免容易发生的软件 Bug。编写代码要坚持渐进和测试驱动的原则，在编写软件模块代码的同时要考虑测试方案和测试代码。完成软件模块开发后，软件开发人员要在 BES III 离线软件会议报告详细的实现细节。软件系统管理员根据提供的模块代码和测试代码进行可接受测试，只有通过测试的软件，才能进入 BES III 离线软件正式版本发布。对于进入正式版本的软件，开发人员仍要根据用户使用的反馈信息不断进行改进，同时要编写详细的软件文档，准备软件评审。

4 质量管理实施

4.1 配置管理

软件配置管理是一套规范的、高效的软件开发管理方法，同时也是提高软件质量的重要手段。BES III 离线软件开发采用 CMT 软件作为配置管理工具，CMT 支持以下主要功能：

- 并行开发支持：因开发和维护的原因，能够实现开发人员同时在同一个软件模块上工作，同时对同一个代码作不同的修改，也不失去控制。
- 修订版管理：跟踪每一个变更的创造者、时间和原因，从而加快问题和缺陷的确定。
- 版本控制：能够简单、明确地重现软件系统的任何一个历史版本。
- 发布管理：管理软件开发，与软件计划保持一致。
- 构建管理：有基于软件包的版本控制功能，实现构建自动化。
- 代码共享：开发人员可以共享各自开发的代码。

CMT 是以软件包为基本单元来组织和管理整个软件系统，软件代码存储于 CVS 仓库中。开发人员在软件的开发过程中，要频繁使用软件包的创建(create package)、检入(check in)、检出(check out)、更新(update)、合并(merge)等操作。由于一个发布的 BES III 离线软件版本是由一些通过测试的软件包组成，所以要实现软件版本控制，只要对每一个软件包的版本进行跟踪即可。BES III 离线软件使用统一的标识 包名—Major 号—Minor 号—Patch 号，Major 号、Minor 号和 Patch 号的变更需符合下列规则：

- 去除软件 bug，但对外的软件接口没有变化，Patch 号增加 1。
- 以向下兼容的方式增加新的属性，用户只需重新编译该软件包就能使用，Minor 号增加 1。
- 软件包中任何影响对外接口的变更，Major 号增加 1。

每个软件包必需具有修改纪录 (ChangeLog)，软件包作者或维护人员所作的改动要在修改纪录中作扼要的说明。

BES III 离线软件系统规定的 CVS 仓库的访问权限如下：

- 任何人可做检出操作。
- 作者、软件包维护人员、软件管理员、子系统负责人可做检入操作。

BES III 离线软件系统使用的外部程序包应标明本身的版本以及对应的 GCC 编译器和操作系统版本，并将外部软件包集中安放于一个目录之下。外部程序包在 CMT 系统中表示为 CMT 接口，该 CMT 接口应该为用户提供灵活选择版本的自由。

BES III 离线软件系统的最小软件管理单位是软件包。当检入一个新的软件包时，包的命名需获得系统负责人的同意。一个完整的软件包应该包括放置在不同目录中的头文件、源文件、配置文件、测试文件等。不同软件包间要避免相互依赖，同是整个软件系统应该很容易安装和部署。

BES III 离线软件系统的备份要求定人定时完成。

4.2 进度管理

BES III 离线软件出品日十分明确，要在 2007 年提供处理 BES III 实验数据的软件。对于一个软件项目，虽然精确估计开发进度非常困难，但必须采取有实效的措施来保证最终目标的实现。

我们采用的方法是将进度安排和工作管理的责任落实到每个开发人员。这保证每个人除了作为小组的一部分外，还负有个人的责任。要求每个开发人员设立自己的进度表，子系统负责人把单独的进度表汇总起来，对提供的日期进行调整并加上缓冲的时间，制定出一个全面的项目进度表，即 BES III 离线软件任务书。

进度管理采用的第二个方法是在完成任务过程中，要进一步对每一项实现进行进度的估计和管理。具体的措施是要求每个开发人员每周填写统一格式的周工作报告，有本周总结和下周计划两个方面的内容。本周总结包括工作描述、完成情况、涉及人员和遇到困难等项，下周计划包括任务描述、进度安排、涉及人员和所需支持等项。系统负责人根据每个开发人

员递交的周报告，编写整个 BES III 离线软件的周报告，并在每周一次的“BES III 系统负责人”会议上进行汇报。

BES III 离线软件系统每两周举行一次软件会议，要求每个开发人员定期在软件会议上报告进展情况。同时系统负责人和子系统负责人一起根据实际情况，不定期和每一个开发人员进行交流，解决开发过程中遇到的困难；而且还要根据进展情况调整计划，在需要时考虑增加人力投入，以完成预定的目标。

4.3 设计和实现

面向对象技术的特点是通过对象之间的职责分工和高度协作来完成任务。这样的好处是代码量较少，系统布局合理，重用程度高。但是当对象的个数大量增加的时候，对象之间的高度耦合的关系将会使得系统变得复杂，难以理解。BES III 离线软件是以 GAUDI 作为底层支撑框架的软件系统。GAUDI 采用了组件编程的思想，并定义了数据对象(DataObject)、算法 (Algorithm)、服务(Service)、工具(Tool)等基本组件和组织组件的方式。在 GAUDI 中，接口和实现是分离的，这能解除面向对象带来的高度耦和关系，同时用户只需要了解接口，而不需要了解实现细节，能做到轻松的抽换实现，而不用修改用户程序。所以开发人员在设计和编程时，应该充分利用 GAUDI 带来的优势，开发与框架兼容的高质量软件。

BES III 离线软件系统的开发过程中，除 GAUDI 外还会引进国外其它实验的一些优秀算法软件例如径迹的模式识别、径迹拟合软件等。对于引进的算法，首先要采用“逆向工程”的方法还原出软件的原始设计。由于实验的探测器存在差异，必需在原始代码的基础上进行二次开发。在达到性能要求的前提下，还有可能对二次开发后的软件实行“重构”以达到和软件框架的良好结合。

在软件开发中，应采用简单原则，不要仅仅为了使用新的技术语言或者方法而使用它们。虽然 BES III 离线软件系统支持 Fortran 和 C++ 语言，但为了避免系统的复杂性，应该尽量使用 C++ 语言编写程序。

4.4 测试

测试是检查软件质量的直接手段。软件设计为编码活动制定了一个标准，但是这个标

准不可能精细到代码级的程度,标准不够明确的则会产生一些问题。再加上其它一些因素(软件的复杂程度大、时间的压力、缺乏或没有进行沟通等)使得错误难以避免。我们提倡测试驱动开发,软件开发人员要一边写代码一边对新产生的代码进行测试,并对设计作必要的调整。

BES III 离线软件系统采用的测试类型主要有下面三种:

- **单元测试 (Unit Testing):** 单元测试是典型的针对代码逻辑的黑盒测试,完成单元测试是开发人员的职责。
- **集成测试 (Integration Testing):** 把通过单元测试的软件模块放入系统时,要进行集成测试来检查模块间以及模块和系统间的作用,以保证模块满足设计需求。
- **回归测试 (Regression Testing):** 对一个正常工作的模块进行修改或升级后,要进行回归测试以保证运行的结果和改动前是一致的。如果不一致,结果应该是合理正确的。

除了单元测试、集成测试和回归测试外,还要对 **BES III** 离线软件的性能进行测试、分析和优化,以达到 **BES III** 实验对软件的要求。并且 **BES III** 离线软件的性能优化将是一项长期的任务。

4.5 文档管理

BES III 离线软件项目的文档管理应该予以充分的重视。软件文档应和对应的软件保持同步。每个文档应包含作者、创建日期、修改历史和对应代码的版本等信息。文档应有统一的模板和统一的编号,文档应进行集中管理。

对于外部软件包所附带的文档(**GAUDI**、**ROOT**、**GEANT4** 等)和外部工具的使用手册(**CMT**、**AFS** 等)要加以收集和整理。

每一个开发人员和用户应该能非常方便地通过网页浏览文档。

4.6 评审

设计和开发评审的目的是由一组评审人员对软件设计和开发的输出进行评价,以判断确定设计和开发的输出能否实现预先定义的规格,同时通过评审标识出与规格和标准的偏差。

评审前,开发人员要准备好详细的文档。评审的内容可根据软件研制周期、技术难度、复杂程度以及使用要求有所侧重和适当的增减。一般评审人员需检查以下内容:

- 软件是否满足需求,设计和实现的整体描述能否被容易理解。

- 软件模块的输入、输出接口以及主要的功能；
- 主要类的类图（包括类间的关系）、表明不同类间相互作用的协作图和顺序图。
- 软件模块的性能、可维护性、与整体结构的兼容性以及是否容易使用。

设计和开发评审应分级进行。设计和开发评审视具体情况可一次进行，也可分段进行。

评审结论应明确，评审资料应及时归档。

5 小结

质量管理和控制是 **BES III** 离线软件开发的一个重要环节。只有拥有高质量的离线软件才能够正确地、高效地处理和分析 **BES III** 实验的数据，取得物理研究成果。本文首先定义了 **BES III** 离线软件的主要质量标准，然后通过分析离线软件的开发过程，得出了配置管理、进度管理、软件设计、软件实现、测试、文档管理和评审等方面质量管理应采取的方法和措施，希望能为软件开发人员提供一些帮助。