

*Д. Ю. Волканов, ассистент факультета
Вычислительной математики и кибернетики МГУ им. М. В. Ломоносова*

Д. А. Зорин, студент МГУ им. М. В. Ломоносова

Исследование применимости моделей оценки надежности для разработки программного обеспечения с открытым исходным кодом

В статье рассматриваются различные модели оценки надежности программного обеспечения с целью определения их применимости к проектам с открытым исходным кодом и уточнения механизма использования. Описываются эксперименты с некоторыми крупными проектами с открытым исходным кодом, в результате которых получены различные характеристики надежности таких проектов и сформулированы гипотезы об их общих свойствах.

Введение

В настоящее время в мире разрабатывается большое количество программных продуктов с открытым исходным кодом (*Open Source*). В этих проектах участвует много разработчиков, добавляющих свои фрагменты кода в общий репозиторий (хранилище, где находятся различные версии файлов, из которых состоит программа).

Проектом будем считать одну или несколько программ, чьим созданием управляет одна команда разработчиков. Большой проект может состоять из подпроектов, реализующих отдельные части, работающие независимо друг от друга; подпроекты — из компонентов, которые выполняют определенный набор функций и не используются отдельно. В проекте участвуют программисты, задействованные на одном или нескольких подпроектах, тестировщики и руководители. Одна из основных задач руководителя проекта — определить, когда программу можно предоставлять конечным пользователям.

Перед руководителями стоят задачи оценить:

- количество ошибок, оставшихся в проекте, и их критичность;

- количество ошибок, оставшихся в выbranном компоненте, и их критичность;

- время, за которое текущая версия станет стабильной, т. е. количество ошибок и их критичность будут меньше заданного порога.

Для решения перечисленных задач можно использовать два подхода: первый основан на изучении исходного кода программы, второй — на применении моделей оценки надежности (МОН) ПО (англ. *Software reliability growth model — SRGM*), т. е. данных, полученных при тестировании [1]. Далее в статье подробно рассматривается второй подход.

Основным результатом расчетов в этом случае является *функция количества ошибок*, или *функция количества* (англ. *mean function*) — неубывающая функция времени, показывающая количество ошибок, найденных к заданному моменту времени. Применение МОН к проприетарному программному обеспечению достаточно хорошо исследовано (например, в работах [2], [3], [4]). Проприетарное программное обеспечение в большей степени, чем открытое, удовлетворяет заданным ограничениям, так как команда разработчиков и тестировщиков централизованная, и поэтому процесс