**Когда прекращать тестирование программ?**

С.Трофимов

Никто не сомневается в необходимости тестирования программ. Будь то небольшой учебный пример или целая информационная система. Вопрос только в том, сколько нужно тестировать и когда можно считать программу протестированной?

Людям свойственно ошибаться при любом виде деятельности, в том числе и при создании программ. Конечно, эти ошибки неумышленные и человек в конце концов их исправит, но как говорят, программ без ошибок не бывает, и на некотором этапе тестирования возникает вопрос, стоит ли дальше искать ошибки или смириться с их некоторым количеством до поры до времени. Этот вопрос подводит нас к определению критериев, по которым можно судить, что программа более или менее работоспособна.

Известно, что можно написать программу из одного оператора без единой ошибки. Казалось бы к одному безошибочному оператору можно добавить еще один, а затем еще один, на первый взгляд, безошибочный, однако, людям свойственно ошибаться... и результат получается не тот, которого ожидали.

Ошибки бывают разные и время на их поиск будет различно. От простых опечаток, которые находятся в первый же запуск программы, до неявных ошибок алгоритма или неправильного использования языковых конструкций, на поиск которых можно потратить не только часы, а дни. Последние найти особенно тяжело.

Современные языки программирования - это чрезвычайно сложный инструмент, на освоение которого уходят годы кропотливого труда. Иногда ошибки в документации, а чаще просто недостаточное понимание работы той или иной конструкции языка или назначения библиотеки, ведет к неправильной работе программы.

Программист смотрит в код и не понимает, почему он работает не так, как задумано. В таких случаях говорят "уперся" и зовут соседа на помощь. В этом случае "свежий" взгляд может значительно ускорить поиск ошибки.

Сократить количество ошибок можно несколькими путями:

применить специальные методы и средства написания программ, например, CASE-средства Rational Rose;

применить надежные, многократно протестированные компоненты и библиотеки;

строго соблюдать и главное контролировать соответствие создаваемых программ проектной документации.

Еще одним, достаточно эффективным, но трудоемким методом сокращения ошибок (я говорю именно о сокращении, а не полном устранении) будет тестирование. Обычно ресурсов (читай времени) для полного тестирования не хватает. Поэтому полное тестирование с проверкой системы во всех режимах и со всеми параметрами трудно реализуемо.

Одной из основных особенностей тестирования является отсутствие эталона программы, с которым можно было бы сравнить нечто уже созданное. Из этого истекает трудность определения момента, когда необходимое качество программы достигнуто.

Основным эталоном в этом случае будет проектная документация, которая приводит разницу взглядов различных людей на качество программы к одному знаменателю. При отсутствии такой документации, я считаю, что в принципе невозможно протестировать программу, поскольку взгляд на объем и качество выполняемых функций не совпадает не только у разных людей, но и может различаться у одного человека в разные промежутки времени.

Поэтому важно определить несколько уровней достижения необходимого качества программ:

отсутствие синтаксических ошибок и аварийных остановок в программе, что достигается прогоном программы с различными данными по максимальному числу ветвей. Для определения участков, которые ни разу небыли запущены при прогоне программы, существуют специальные средства, например Rational Pure Coverage. Из практики работы я могу сделать вывод, что участок кода, который ни разу не был запущен при тестировании, примерно в 80% случаев неработоспособен;

выполняемые функции программ соответствуют технической документации;

расчетные значения, полученные при помощи процедур расчета, соответствуют эталонным.

Часто первые два уровня можно тестировать одновременно, внося коррективы в сопроводительную документацию. Последний пункт самый длительный и трудно тестируемый, невозможно подготовить тест на все возможные сочетания данных, поэтому существует так называемое бета-тестирование, т.е. тестирование самими пользователями.

Теперь, когда некоторые критерии тестирование были здесь упомянуты, можно ответить на вопрос, вынесенный в заголовок статьи.

Первый этап тестирования можно прекращать, когда есть уверенность, что большая часть синтаксических ошибок и аварийный остановов устранена. Остальные постепенно будут устраняться в процессе других этапов тестирования.

Второй этап тестирования можно прекращать, когда большая часть функциональности проверена и работает в соответствии с проектной документацией. Остальные несоответствия будут устраняться в процессе написания сопроводительной документации.

И наконец, третий этап тестирования можно прекращать, когда основные расчетные тесты дают правильные результаты.

Тестирование пользователями будет продолжаться на всем этапе жизненного цикла программного обеспечения, и вплоть до смены ПО на более современное, пользователи будут находить в нем неточности и ошибки, к которыми, в конце концов, они привыкнут и начнут воспринимать их как само собой разумеющееся, пока программисты не выдадут "нагора" новую версию, в которой "исправлены старые ошибки и добавлены новые" и все начнется сначала.