## Методы обеспечения надежности программ для контроля их технологической безопасности.

При исследовании методов и средств оценки уровня технологической безопасности программных комплексов учитываются факторы, имеющие, как правило, чисто случайный характер. Следовательно, показатели, связанные с оцениванием безопасности ПО лучше всего выражать вероятностной мерой, а для их вычисления можно использовать вероятностные модели надежности ПО [[56]](http://citforum.ru/security/articles/kazarin/4.shtml#56), которые при осуществлении замены условия правильности функционирования программ на условие их безопасности можно использовать для этих целей.

Исходные данные, определения и условия

В данном разделе будем считать, что безопасность программного обеспечения - это вероятность того, что преднамеренные программные дефекты, вызывающие критическое поведение управляемой КС, будут обнаружены при определенных условиями внешней среды и в течение заданного периода наблюдения при испытаниях.

Под определенными условиями внешней среды следует понимать описание входных данных и состояние вычислительного процесса в момент выполнения программы при испытаниях. Под заданным периодом функционирования понимается время, необходимое для выполнения поставленной задачи. Выделение определенного интервала времени целесообразно в случае систем реального времени, в которых неопределенными являются количество прогонов любой из действующих программ, состояние баз данных и моменты выполнения той или иной программы. В условиях, когда состояние программы достоверно известно в качестве периода наблюдений следует выбрать рабочий цикл или прогон. В любом случае перед каждым повторным выполнением программы необходимо либо восстанавливать состояние памяти , либо осуществлять серию последовательных прогонов, при котором последовательным образом изменяется состояние базы данных.

Интуитивное определение безопасности ПО может быть уточнено в статистическом смысле на основе следующих простых соображений:

* машинная программа p может быть определена как описание некоторой вычислимой функции F на множестве E всех значений наборов входных данных, таких что каждый элемент Ei множества Eпредставляет собой набор значений данных, необходимый для выполнения прогона программы: E=(Ei:i=1,2,...,N);
* выполнение программы p приводит к получению для каждого Ei определенного значения функцииF(Ei);
* множество E определяет все возможные вычисления в программе p, то есть каждому набору входных данных Ei соответствует прогон программы p, и наоборот, каждому прогону соответствует некоторый набор входных данных Ei;
* наличие дефектов в программе p приводит к тому, что ей на самом деле соответствует функция F', отличная от заданной функции F;
* для некоторого Ei отклонение выхода F'(Ei), полученного в результате выполнения программы не должно превышать уровень безопасности программного обеспечения S(Ei), то есть безопасность обеспечивается при соблюдении ограничения: F'(Ei), S(Ei). (Вопрос о том, приводит ли некоторое отклонение выхода к нарушению условия безопасности, должен решаться в каждом конкретном случае отдельно, поскольку все определяется конкретными особенностями поведения системы после нарушения ее работы)

.

Совокупность действий, включающая ввод Ei, выполнение программы p, которое заканчивается получением результата F'(Ei) называется прогоном программы p. Необходимо также отметить, что значения входных переменных, образующие Ei, не должны все одновременно подаваться на вход программ p. Таким образом, вероятность P того, что прогон программы приведет к обнаружению дефекта, равна вероятности того, что набор данных Ei, используемый в данном прогоне, принадлежит множествуEe. Если обозначить через neчисло различных наборов значений входных данных, содержащихся в Ee, тоP=ne/N - есть вероятность того, что прогон программы на наборе входных данных Ei, случайно выбранном из E среди равновероятных, закончится обнаружением дефекта. При этом R=1-P - есть вероятность того, что прогон программы p на наборе входных данных Ei, случайно выбранном из E среди априорно равновероятных, приведет к получению приемлемого результата.

Однако в процесс функционирования программы выбор входных данных из E обычно осуществляется не с одинаковыми априорными вероятностями, а диктуется определенными условиями работы. Эти условия характеризуются некоторым распределением вероятностей pi, того, что будет выбран набор входных данных Ei. Распределение P может быть определено через pi с помощью величины yi, которая принимает значение 0, если прогон программы на наборе Ei заканчивается вычислением приемлемого значения функции, и значением 1, если этот прогон заканчивается обнаружением дефекта. Поэтому http://citforum.ru/security/articles/kazarin/8.gif - есть вероятность того, что прогон программы на наборе входных данных Ei, выбранных случайно с распределением вероятностей pi, закончится обнаружением дефекта. При этом R=1-P есть вероятность того, что прогон программы p на наборе входных данных Ei, выбранных случайно с распределением вероятностей pi, приведет к получению приемлемого результата.

Введем также определения и обозначения, связывающие структурные характеристики программ с их безопасностью. Структурными характеристиками программы p являются множество ветвей Lj (j=1,...,n), подмножества входных наборов данных Gj, соответствующие ветвям Lj, множества сегментов Segj, из которых состоят отдельные ветви, совокупность операторов ветвления, которые обеспечивают переход от одного сегмента к другому при движении по отдельной ветви программы.

Оценка технологической безопасности программ на базе метода Нельсона

Перед тем как перейти непосредственно к методу оценки, необходимо сделать несколько замечаний. Следует заметить, что реальные условия испытаний программ всегда существенно отличаются от тех, которые требуются для представительного измерения уровня безопасности ПО. Так, например, тестовые прогоны выполняются на входных наборах данных, выбранных не совсем случайным, а выбранных некоторым определенным образом: обычно выбор производится так, чтобы соответствующую ошибку можно было найти как можно быстрее. При этом выбор основывается на опыте и интуиции испытателей, либо осуществляется с учетом функциональных возможностей, которые должна обеспечивать программа, или возможностей соответствующих методик испытаний. Поэтому контрольные примеры, как правило, не являются представительными с точки зрения моделирования реальных условий работы программы и далее описывается процедура грубой оценки величины R, предусматривающая использование результатов испытаний и включающая следующие шаги:

1. Определение множестваE входных массивов.
2. Выделение в E подмножеств Gj, связанных с отдельными ветвями программы.
3. Определение для каждого Gj в предполагаемых условиях функционирования значений вероятностиPj.
4. Определение подмножества Gj для каждого входного набора данных, используемого в контрольных примерах.
5. Выявление проверенных пар и непроверенных в ходе испытаний сегментов и пар сегментов.
6. Определение для каждого j величины P'=ajPj, где aj определяется в соответствии со следующими правилами [[56]](http://citforum.ru/security/articles/kazarin/4.shtml#56).
   * aj=0,99, если подмножество Gj включает более одного контрольного примера;
   * aj=0,95, если подмножество Gj включает ровно один контрольный пример;
   * aj=0,90, если подмножество Gj не включает ни одного контрольного примера, но в процессе проверки программы были найдены все сегменты и все сегментные пары ветви Lj;
   * aj=0,80, если в ходе испытаний были опробованы все сегменты, но не все сегментные пары;
   * aj=0,80-0,20m, если m сегментов (1 http://citforum.ru/security/articles/kazarin/b.gif m http://citforum.ru/security/articles/kazarin/b.gif 4) ветви Lj не были опробованы в ходе испытаний;
   * aj=0, если более чем 4 сегмента не были опробованы в процессе испытаний.
7. Вычисление грубой оценки R" осуществляется по формуле http://citforum.ru/security/articles/kazarin/9.gif, где k представляет собой общее число ветвей программы.

Приведенные выше параметры ajбыли определены интуитивно [[56]](http://citforum.ru/security/articles/kazarin/4.shtml#56) на основе анализа теоретических результатов исследования и экспериментальных результатов тестирования различных программ. Для того, чтобы получить более точные оценки величины R необходимо провести измерения с использованием подходящего метода формирования выборки.

Оценка технологической безопасности ПО осуществляется посредством проверки условия R"*http://citforum.ru/security/articles/kazarin/b.gif*S", где S"установленная нормативными документами граница безопасности ПО. Отметим также, что для систем критических приложений такая граница должна быть достаточно высокой, то есть стремится к 1.