

Методика структурной идентификации рисков разработки программного обеспечения

Смирнов А.А., д.т.н., проф., dr.smirnova@gmail.com, Коваленко А.В., к.т.н., доц.
Кировоградский национальный технический университет, г. Кировоград

Используя результаты исследований, мнения экспертов, маркетинговые данные, а также базы знаний таких известных фирм как *Eram Systems* и *Nix Solutions Ltd*, идентифицируем риски разработки ПО. Основные риски разработки программного обеспечения можно представить в виде совокупности множеств организационных, управленческих, операционных, технологических, эксплуатационных, социальных и правовых рисков.

Отличительной особенностью представленной классификации является учет эксплуатационных рисков. Особенную важность эти риски приобретают в условиях повышенного уровня киберпреступности, когда пренебрежение уязвимостями программного обеспечения может привести к эксплуатационным проблемам, а зачастую и невозможности эксплуатации («краху») ПО.

Кроме этого, в условиях украинского правового поля наблюдаются отдельные случаи неадекватности и несоответствия правовым нормам действий должностных лиц государственного аппарата.

Практика ряда известных фирм-разработчиков ПО (*Nix Solutions Ltd*, и др.) показывает, что указанный фактор риска целесообразно учитывать при разработке ПО, наряду с фактором возможного изменения украинского законодательства.

Большинство из рассматриваемых рисков разработки ПО (организационные, операционные, управленческие и др.) могут оказывать непосредственное влияние как на процесс разработки ПО, так и на процесс его эксплуатации. В то же время, например, эксплуатационные риски непосредственного влияния на процесс разработки ПО не оказывают. Но пренебрежение этими рисками ведет зачастую к провалу эксплуатации ПО и потерям будущих заказов и проектов (простоям разработчиков ПО). Именно этим фактором вызвана связь между блоками «Провал при эксплуатации ПО» и «Провал при разработке ПО».

Однако, несмотря на это в целом можно выделить множество рисков непосредственно влияющих на процесс разработки ПО и множество рисков непосредственно влияющих на процесс эксплуатации ПО.

Следует заметить, что выделенные факторы в достаточной степени описывают перечень возможных рисков разработки ПО. Однако, они не дают представления о взаимном влиянии и соответственно возможном изменении конечного результата. Кроме этого приведенные множества рисков разработки ПО в разной степени влияют на конечный результат. Поэтому следующим шагом идентификации рисков разработки ПО целесообразно выполнить процедуры ранжирования и выделения наиболее приоритетных (важных) рисков разработки ПО.

Проведенные исследования показали, что для решения задачи определения взаимовлияния рисков целесообразно использовать инструмент анализа причинно-следственных связей между различными факторами и рисками, разработанный Каору Исикава (диаграмма Исикавы). В соответствии с известным принципом Парето, среди множества потенциальных причин (причинных факторов, по Исикаве), порождающих проблемы (следствие), лишь две-три являются наиболее значимыми, их поиск и должен быть организован. Для этого осуществляется:

- сбор и систематизация всех причин, прямо или косвенно влияющих на исследуемую проблему;
- группировка этих причин по смысловым и причинно-следственным блокам;
- ранжирование их внутри каждого блока;
- анализ получившейся картины.

Поэтому данный инструмент позволяет прояснить и учесть все существенные факторы, влияющие на результат разработки ПО.

Применение диаграммы Исикавы позволяет выяснить причины каких-либо проблем в организации или, например, причины возникновения эксплуатационных «багов» ПО. При этом диаграмма Исикавы имеет ряд достоинств:

- помогает наглядно показать связи между полученным результатом и вызвавшими его причинами;
- позволяет провести анализ цепочки факторов, воздействующих на проблему.

Изображение диаграммы Исикавы дает возможность получить более подробную информацию о возможности взаимовлияния различных видов риска друг на друга, что так же даст уточняющие данные для количественного анализа рисков. Однако задачу выбора наиболее приоритетных рисков диаграмма решить не может.

Для решения этой задачи в работе предлагается использовать математический аппарат многокритериальной оптимизации, основанной на локальной геометрии множества Парето.

Анализ литературы показал, что существуют, по крайней мере, три формулировки многокритериальной оптимизации, основанной на локальной геометрии множества Парето:

1. Локальная. Найти одно Парето-оптимальное решение (ближайшее к заданной начальной точке).
2. Глобальная. Найти конечное множество Парето-оптимальных решений, достаточно хорошо описывающее (покрывающее) истинный Парето-фронт.
3. Интерактивная. Найти Парето-оптимальное решение, максимально удовлетворяющее предпочтениям лица принимающего решение (ЛПР).

Проведенные исследования показали, что в процессах, построенных на принципах постоянных коммуникаций между участниками, использования «мозговых штурмов» с привлечением мнений экспертов, целесообразным представляется использование интерактивной формулировки многокритериальной оптимизации.

В этих условиях абстрактную задачу выбора наиболее важных рисков разработки ПО из имеющегося исходного множества возможных (допустимых) вариантов (решений) X можно сформулировать следующим образом.

Обозначим множество всех заранее определенных рисков разработки ПО через $S(X)$. Очевидно, $S(X) \subset X$. Таким образом, в задаче выбора дано множество X , содержащее, по крайней мере, два элемента, а требуется найти некоторое его непустое подмножество $S(X)$. Предполагается, что выбор производится ЛПР, в роли которого может выступать как отдельный человек, так и целый коллектив разработчиков. Для того, чтобы совершаемый выбор в наибольшей степени соответствовал достижению имеющейся цели (т.е. был «наилучшим» или «оптимальным» для данного ЛПР), необходимо в процессе выбора учитывать мнение экспертов.

Проведенные исследования показали, что в настоящее время существует множество подходов учета мнения экспертов (метод анализа иерархий, реализованный в программном продукте *EXPERT CHOICE*, метод «искусственного» отношения предпочтения, и др.) однако все они обладают существенными недостатками, главный из которых заключается в том, что, несмотря на многообразие и детальную изученность иерархий и «искусственных» отношений, крайне редко какое-либо из них можно считать удовлетворяющим конкретное ЛПР в полной мере. Характерным примером, подтверждающим данный факт является пренебрежение оценкой уязвимостей разработанного ПО (недостаточность или полное отсутствие реп-тестирования).

Таким образом, получила дальнейшее развитие методика структурной идентификации рисков разработки ПО, отличающаяся от известных построением оценки рисков разработки ПО «сверху» в виде множества, при наличии произвольного непротиворечивого конечного набора «квантов информации».