Использование FMEA-анализа при разработке сложных программных средств

Б.М. Позднеев, А.А. Котлячков (МГТУ "Станкин")

Рассмотрены аспекты применения методологии FMEA (Failure Modes and Effects Analysis) для обеспечения качества сложных программных средств (СПС). В результате анализа потенциальных возможностей методологии FMEA предложено использовать ее в качестве основной концепции при разработке СПС.

Ключевые слова: методологии FMEA, жизненный цикл программного продукта, сложное программное средство, качество сложного программного средства.

Обеспечение качества СПС – актуальный вопрос в мире информационных технологий, решение которого возможно посредством применения методологии FMEA [1].

Методология FMEA представляет собой технологию анализа возможности возникновения дефектов в процессе эксплуатации уже готового продукта и их влияния на потребителя, позволяет снизить затраты и уменьшить риск возникновения дефектов. В настоящее время это одна из стандартных методик анализа качества технологических изделий и процессов. В ходе развития методологии были выработаны типовые формы представления результатов анализа и правила его проведения.

Рассмотрим возможность применения FMEA при разработке СПС, положительные и отрицательные стороны методологии при использовании ее в сфере информационных технологий. Качественно проектирование ПО в значительной степени обеспечивает снижение временных затрат на тестирование и исправление ошибок, допущенных на концептуальной стадии. Методология FMEA призвана помочь разработать детальные, обоснованные и корректные требования, необходимые для минимизации вероятности возникновения ошибок на стадии проектирования и, как следствие, на всех последующих стадиях жизненного цикла ПО.

Под СПС будем понимать системы, состоящие из множества взаимосвязанных компонентов с разветвленной структурой обмена и обработки данных, к которой предъявляются высокие требования в отношении надежности, быстродействия, качества и удобства для пользователей [2].

Применение FMEA при разработке ПО имеет некоторые особенности. Если рассматривать технологию производства, то здесь возможны любые внешние воздействия пользователя в отношении оборудования и его незапланированный износ при длительной эксплуатации, которые могут привести к остановке ТП. При проектировании программных продуктов отсутствует фактор износа, воздействие пользователя ограниченно интерфейсом программы, в некоторых случаях учитывается возможность использования программ для взлома. Таким образом, сужается область исследования, которую необходимо охватить при использовании методологии FMEA.

Если анализ технологических изделий является описательным, то анализ СПС легко поддается формализации и составляется в форме таблицы или рабочего листа. FMEA анализ при разработке СПС включает этапы построения моделей для объекта анализа и исследования построенных моделей.

На первом этапе строятся компонентная, структурная, архитектурная, функциональная, потоковая модели. На втором – в ходе анализа моделей определяются потенциальные дефекты для каждого из элементов компонентной модели объекта. Дефекты могут быть связаны с отказом одного из компонентов СПС, с неправильным выполнением им своих функций или с вредными воздействиями компонента на систему в целом. Определяются потенциальные причины дефектов и их потенциальное воздействие на систему в целом и на потребителя. В виду сложности программной системы каждый из рассматриваемых дефектов может вызвать цепочку отказов. При анализе последствий используются структурная и потоковая модели объекта, определяются возможности контроля выявления дефектов до наступления последствий.

Приведем пример: в систему, имеющую ограничения по потоку входящих данных, приходит сообщение об ошибке. Предположим наиболее вероятные варианты выхода из сложившейся ситуации:

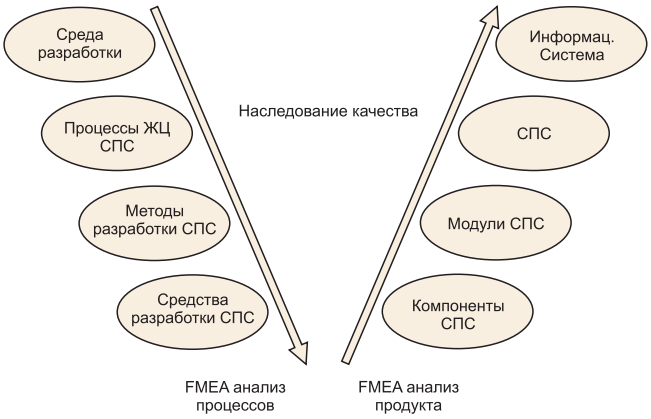
- фильтрация ошибочных данных и их игнорирование в случае, если допустима потеря данных (система имеет большой поток данных, и потеря одной строки составляет тысячную долю процента);
- попытка исправления данных на этапе их входа в систему, если поступление ошибочных данных внутрь невозможно;
- пропуск данных в систему и попытка исправления их на клиентском приложении лишь в случае запроса пользователя;
- пропуск данных в систему без исправления (ответственные модули должны уметь обрабатывать ошибочные данные).

В ходе вышеперечисленных действий определяется параметр риска потребителя: $C = B_1 \times B_2 \times B_3$ [1], где B_1, B_2, B_3 – показатели экспертной оценки по 10-балльной шкале.

Суть методологии позволяет использовать данные различного смыслового значения. При разработке программ целесообразно воспользоваться терминологией, предлагаемой Rational (RUP) [3]. В нашем случае B_1 – это критичность ошибки, то есть степень влияния на надежность системы. Высший балл ставится для ошибок, которые приводят к прекращению функ-

Таблица 1. Результаты FMEA анализа объекта

Компонент	Дефект	Причины	Последствия	Контроль	B_3	B_3	B_3	C



Наследование качества при разработке информационных систем

ционирования СПС (фатальным ошибкам); низший – ошибкам, не влияющим на работоспособность системы. B_2 – это серьезность ошибки, влияющей на пользователя в случае ее возникновения. Высший балл ставится, если ошибка привела к невосполнимым потерям или отказу от использования программы; низший – ошибкам, которые пользователь проигнорирует или сочтет за функциональные особенности. B_3 – это вероятность возникновения дефекта, наивысший балл проставляется при частоте ≥1/4.

Результирующий параметр C показывает, каким образом на текущий момент соотносятся друг с другом причины возникновения дефектов. Первоочередному устранению подлежат дефекты с наибольшим коэффициентом риска. В методическом документе и проекте стандарта [4] приведены рекомендуемые 10-балльные шкалы оценок для указанных трех критериев. Однако для каждого программного продукта и области его применения эти шкалы желательно корректировать и конкретизировать. В зависимости от требований надежности и устойчивости системы устанавливается допустимая граница значений параметра C. Риски, которые превышают граничные значения, подлежат обязательному устранению. Результаты анализа заносятся в табл. 1 [1].

При разработке СПС важно, на каком этапе при проектировании может быть допущена ошибка. На рисунке приведена иерархическая модель наследования качества СПС. Критичность возникновения отказа нарастает при движении от верхних уровней к нижним и при движении слева направо – от концептуальной стадии к конечному продукту. Критичность обратно пропорциональна процессу наследования качества [5].

Разработка любого программного средства ведется на основе ранее выработанных требований. Применение FMEA позволяет сформировать эти требования, так как дает возможность сосредоточить внимание на выявлении наиболее ответственных и уязвимых мест СПС. Для этого необходимо пройти несколько этапов:

Таблица 2. Применение методологии FMEA для обеспечения качества при разработке СПС

Величина параметра B	Трактовка величин значений параметра B, предлагаемая методологией RUP
9 - 10	Ошибка не позволяет использовать основные функции системы
7 - 8	Ошибка мешает использовать основные функции системы (то есть возможно получить обходных путей)
5 - 6	Ошибка не мешает использованию основных функций системы, но является хорошо заметной
3 - 4	Незначительная ошибка
1 - 2	Недостатки можно отнести к улучшению системы, а не к ошибке

- определить назначение системы, условий и характеристик среды ее использования, определить структуру и основные функции системы;
- сформировать группу экспертов для анализа и управления рисками проекта создания СПС на всем жизненном цикле;
- разработать предварительные требования к функциональной пригодности и конструктивным характеристикам процессов жизненного цикла.

Для полноценного анализа работы проектируемой системы необходимо построить модели функционирования ПС, отражающие структуру, функциональность, обработку и обмен данными внутри и вне системы, поведение системы при возникновении внутренних/внешних ошибок. На основе выполненных подготовительных мер необходимо произвести анализ возможных дефектов и сбоев в системе, используя рассмотренную выше методику оценки степени риска для потребителя. Для этого необходимо выделить возможные источники угрозы, определить критерии работоспособности, отказа и устойчивости системы, выделить и идентифицировать основные классы рисков в жизненном цикле ПС. Идентифицировать причины появления функциональных и структурных дефектов и анализ возможных последствий и методов их предотвращения.

Описанные приемы использования методологии FMEA при разработке программных продуктов апробированы в процессе глубокой модернизации поисковой системы информационного агентства (Москва). На основании опытных данных получены экспертные значения для параметров критичности B_1, серьезности B_2 и вероятности возникновения B_3 дефекта. Для конкретизации величины значений параметров $B_{1\text{-}3}$, было решено обратиться к градации критичности ошибок предлагаемой методологией RUP [3], которая приведена в табл. 2.

Группой анализа были установлены две основные границы. Критический уровень $C_{кр} = 220$ и уровень $C_0 = 65$. Превышение параметром C порогового значения $C_{кр}$ означает обязательную необходимость применения мер по снижению риска пользователя. Уменьшение параметра C ниже порогового значения C_0 означает желательность разработки корректирующих мер. С этой целью был разработан план модернизации поисковой систе-

мы агентства, охватывающий полный жизненный цикл [6] СПС. Отделом качества заказчика было принято решение об экспериментальном применении методологии FMEA для оценки рисков конечного пользователя, вероятных при реализации данного плана модернизации. Результаты FMEA-анализа использованы для расстановки акцентов при разработке плана тестирования обновленной поисковой системы. Они включают оценку вероятности риска пользователя C на всех этапах жизненного цикла процесса модернизации и представлены в табл. 3.

Результаты анализа позволили выявить недостатки предложенного плана модернизации. Как видно из табл. 3 работы на этапах "Архитектурное проектирование" и "Функциональное проектирование" потребовали пересмотра и переработки, так как риск потребителя превышал установленное значение Ckp. Этапы "Разработка требований", "Кодирование" и "Тестирование" потребовали разработки корректирующих мероприятий.

Выводы. Основываясь на опыте применения FMEA-анализа при разработке СПС, можно отметить его следующие положительные стороны.

• Невысокие финансовые затраты, так как не требуется создания особой материально технической базы и длительного промежутка времени для реализации.

• Применение методологии в ИТ компании не требует специальных навыков и знаний и находится в рамках компетенции отдела качества.

• Варианты использования СПС ограничены возможностями пользовательского и программного интерфейса, поэтому FMEA-анализ может быть строго формализован, что упрощает оценку параметра риска пользователя С.

Таблица 3. Применение методологии FMEA для обеспечения качества при разработке СПС

№	Операции жизненного цикла СПС	Коэффициенты			C	Примечание
		B_1	B_2	B_3		
1	Разработка требований	4	5	3	60	$C_0 < C_1 < C_{kp}$
2	Управление проектом	3	3	3	27	$C_2 < C_0$
3	Архитектурное проектирование	6	6	7	252	$C_3 > C_{kp}$
4	Функциональное проектирование	6	6	7	252	$C_4 > C_{kp}$
5	Кодирование	5	6	5	150	$C_0 < C_5 < C_{kp}$
6	Управление конфигурацией	3	3	3	27	$C_6 < C_0$
7	Тестирование	4	4	6	96	$C_0 < C_7 < C_{kp}$
8	Сертификация	3	3	3	27	$C_8 < C_0$
9	Реализация	4	4	3	48	$C_9 < C_0$
10	Поддержка	3	3	5	45	$C_{10} < C_0$
	$C_{kp} = 220,\ C_0 = 65$					

Серый цвет – процессы, требующие коррекции
Коричневый цвет – критические процессы, требующие мер по устранению отказов

Список литературы

1. Анализ видов и последствий потенциальных отказов (FMEA). М.: Приоритет. 2003.
2. *Липаев В.В.* Анализ и сокращение рисков проектов сложных программных средств. М.: Синтег. 2005.
3. *Гари Поллис, Лиз Огастин, Крис Лоу, Джас Мадхар.* Разработка программных проектов на основе Rational Unified Process (RUP). М.: Бином-Пресс. 2005.
4. Анализ видов, последствий и критичности отказов. ГОСТ 27.310-95.
5. *Позднеев Б.М.* Интегрированная информационная поддержка процессов проектирования и производства поковок конкурентоспособного качества. М.: Янус-К. 2005.
6. Информационная технология – процессы жизненного цикла программных средств ГОСТ Р ИСО/МЭК 12207.

Позднеев Борис Михайлович *– д-р техн. наук, проф.,*
Котлячков Александр Алексеевич *– старший инженер-программист кафедры "Информационные системы" МГТУ "Станкин".*

Контактный телефон (495) 972-94-27.
E-mail: nuclear_fly@mail.ru