

- поддержка огромным количеством средств разработки третьих фирм;
- выполняет любую программу с набором команд Thumb, может выполнять Thumb подпрограммы процессоров ARM7 и ARM9;
- снизу вверх совместим с процессором Cortex-M3;
- поддерживает работу группы команд: операции над данными, ветвление, генерации исключений, множественная загрузка сигнала;
- поддерживается всеми доступными средствами разработки для ARM процессоров.

Следующим этапом развития является серия на базе ядра Cortex-M4, лицензию приобрели: NXP, STMicroelectronics, Freescale, Winbond Electronics, Texas Instruments и др. Процессор Cortex-M4 представляет собой высокоэффективное решение для приложений управления цифровым сигналом, сохраняя ведущие в отрасли показатели семейства процессоров ARM Cortex-M по поддержке микроконтроллерных функций. Основные характеристики ARM Cortex-M4:

- самая высокая производительность – до 168 МГц;
- до 32 уровней приоритетов;
- высокая эффективность обработки данных вкуче с низким потреблением энергии;
- поддержка огромным количеством средств разработки третьих фирм;
- совместим с процессором Cortex-M3;
- реализует набор инструкций Thumb / Thumb-2; Single cycle 16,32-bit MAC; Single cycle dual 16-bit MAC;
- поддерживает работу группы команд: операции над данными, ветвление, генерации исключений, эксклюзивная загрузка сигнала;
- поддерживается всеми доступными средствами разработки для ARM процессоров.

В числе областей применения процессора ARM Cortex-M0 названы медицинские приборы, счетчики, управляющие системы, игровые аксессуары, компактные блоки питания. Основные сферы применения ARM Cortex-M1 – контрольно-измерительные встраиваемые системы, сетевое и телекоммуникационное оборудование. Основные сферы применения ARM Cortex-M3 – бытовая техника, автоматизированный электропривод, системы мониторинга, автоматизация зданий, преобразовательная техника, игровые устройства и многое другое. Область применения ARM Cortex-M4 охватывает управляющие системы, автомобильное производство, автоматизированные промышленные задачи.

Список цитированных источников

1. Меджахед, Дж. STM32: эпоха 32-битных микроконтроллеров наступила // Новости электроники [Электр. ресурс]. – 2011. – Режим доступа: <http://www.compeljournal.ru/enews/2011/2/3> – Дата доступа: 16.02.2011

УДК 51-7

ИНСТРУМЕНТАЛЬНАЯ ПОДДЕРЖКА КОЛИЧЕСТВЕННОЙ ОЦЕНКИ ПОТЕНЦИАЛЬНЫХ РИСКОВ ИТ-ПРОЕКТОВ

Неверович Ж.В.

*УО «Белорусский государственный университет информатики
и радиоэлектроники», г. Минск*

Научный руководитель – Поттосина С.А., к. ф.-м. наук, доцент

Эффективное управление рисками ИТ-проектов включает в себя не только проведение анализа рисков на качественном содержательном уровне, но и их количественную оценку. Методы количественной оценки потенциальных рисков проекта достаточно сложны в

применении и затратные по времени. Для того чтобы облегчить и ускорить процесс количественной оценки, была спроектирована программа количественной оценки потенциальных рисков IT-проекта. Разработанная программа позволяет вести учет рисков проекта, подлежащих количественной оценке, задавать модель параметров проекта для проведения количественной оценки, проводить анализ чувствительности с построением диаграммы «торнадо», осуществлять имитационное моделирование Монте-Карло и строить графики распределения значений параметров проекта.

Программа реализована на языке Java с использованием веб-сервиса на основе JAX-WS 2.0. Бизнес-логика системы количественной оценки потенциальных рисков IT-проекта реализована EJB session-компонентами версии 3.0. В качестве СУБД выбрана PostgreSQL 9.1. Доступ к данным в СУБД осуществляется с помощью Java Persistence API TopLink. В качестве сервера приложений выбран Sun Glassfish Enterprise Server v3. При построении модели параметра проекта используются символьные выражения. Для организации вычислений данных выражений применяется обратная польская запись.

С помощью разработанной программы проведем количественную оценку рисков IT-проекта по созданию каталога товаров и услуг интернет-магазина. Диаграмма Ганта данного проекта представлена на рисунке 1.

Команда проекта выявила наличие в проекте риска увеличения длительности всего проекта. Для проведения количественной оценки риска будем использовать имитационное моделирование Монте-Карло. Метод Монте-Карло способен значительно повысить качество принимаемых решений, предлагая четкий анализ рисков, различных сценариев и вероятности достижения цели [1].

Опираясь на диаграмму Ганта, зададим длительность проекта формулой (модель параметра проекта)

$$T = \sum_{i=1}^n t_i, \quad (1)$$

где T – длительность всего проекта; t_i – длительность i -го этапа проекта; n – количество этапов проекта.

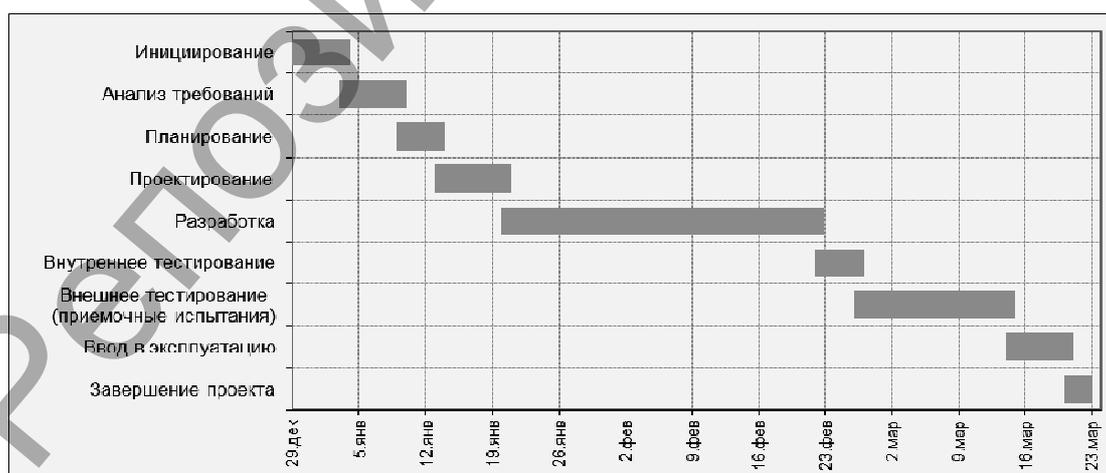


Рисунок 1 – Диаграмма Ганта проекта по созданию каталога товаров и услуг интернет-магазина

Основная идея метода Монте-Карло заключается в том, что длительность каждого этапа проекта рассматривается как случайная величина. В качестве функции распределения продолжительности этапов выбирается треугольное распределение, описание

которого требует задания трех параметров: минимального (t_L), наиболее вероятного (t_M) и максимального (t_H) значений длительности этапов проекта. Минимальные, наиболее вероятные и максимальные значения продолжительности этапов определяются экспертами, которые имеют соответствующий опыт работы в подобных проектах.

Имитация проекта подразумевает расчет по математической модели общей длительности проекта заданное количество раз. Число итераций расчета влияет на точность результата: чем больше, тем точнее, но слишком большое их число замедляет расчет. Для данного проекта общую длительность проекта рассчитаем 500 раз.

На каждой итерации продолжительности этапов проекта определяются с помощью генерируемого в диапазоне от 0 до 1 случайного числа R , которое вводится согласно треугольному распределению в формулу

$$t_i = \begin{cases} t_H - \sqrt{(t_H - t_M)(1 - R)(t_H - t_L)}, & \text{если } R > \frac{t_M - t_L}{t_H - t_L}, \\ t_L + \sqrt{R(t_M - t_L)(t_H - t_L)}, & \text{иначе,} \end{cases} \quad (2)$$

где t_i – длительность i -го этапа согласно треугольному распределению;

R – случайное число в диапазоне от 0 до 1;

t_L – минимальная (оптимистичная) длительность i -го этапа по мнению экспертов;

t_M – наиболее вероятная длительность i -го этапа по мнению экспертов;

t_H – максимальное (пессимистичное) значение длительности i -го этапа, по мнению экспертов.

Подставляя вычисленные значения продолжительности этапов в модель (формула 1), определяется длительность всего проекта. Таким образом, по результатам всех итераций получается набор значений длительности всего проекта – их распределение нам и интересно посмотреть.

На рисунке 2 изображен график распределения длительности проекта в целом. График распределения позволяет получить ответы на следующие вопросы:

– Насколько вероятно, что команда проекта сумеет завершить проект в срок, установленный руководством?

– Если вероятность благоприятного исхода составит менее 90% (такое значение предпочтительно с точки зрения команды), то, на сколько дней необходимо сдвинуть крайний срок?

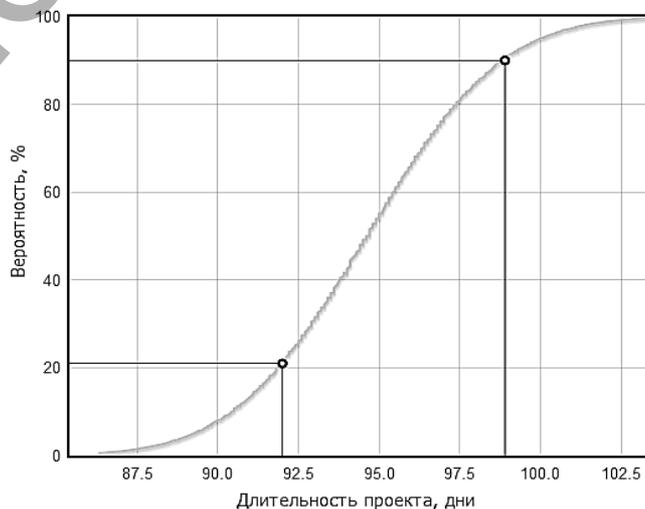


Рисунок 2 – График распределения длительности проекта

Чтобы ответить на 1-й из поставленных вопросов, на графике необходимо:

– отметить на оси X продолжительность проекта, которая соответствует сроку, установленному руководством (92 дня);

– сдвинуться от этой даты вверх до пересечения с кривой графика;

– сдвинуться от точки пересечения влево до пересечения с осью Y.

Значение, находящееся на оси Y (21%), и есть вероятность того, что проект удастся завершить за 92 дня.

Совершенно ясно, что эта вероятность значительно ниже, чем предпочитаемая в 90%. Определяется длительность проекта с 90%-ой вероятностью. По графику видно, что крайний срок исполнения проекта нужно сдвинуть на 7 дней (2-й вопрос).

Список цитированных источников

1. Милошевич, Д.З. Набор инструментов для управления проектами / Д.З.Милошевич; пер. с англ. Е.В. Мамонтова – М.: Компания АйТи; ДМК-Пресс, 2008. – 729 с.

УДК 681.3

АЛГОРИТМЫ ПОСТРОЕНИЯ ТЕСТОВЫХ СПЕЦИФИКАЦИЙ СИСТЕМ

Никонюк А.Н., Осмоловец А.М.

*УО «Брестский государственный технический университет», г. Брест
Научный руководитель – Муравьев Г.Л., к.т.н., доцент*

Для автоматизации тестирования моделей, для целей обучения аналитическому и имитационному моделированию систем, для генерации уникальных вариантов архитектур необходимо решать задачу автоматического формирования параметров систем заданной сложности и режима функционирования, соответствующих тестовых описаний. Такие спецификации, представляющие собой наборы значений параметров и характеристик систем, описываемых в терминах сетей массового обслуживания [1, 2], необходимы и для целей тестирования и для построения имитационных моделей, оценки их адекватности.

Трудоемкость формирования значений параметров большого числа систем, трудоемкость проверки их корректности, трудности обучения моделированию на реальных системах делают эту задачу актуальной.

Указанные описания должны обладать такими свойствами, как уникальность, управляемая сложность, полнота, контролируемость, документированность.

Управляемая сложность обеспечивается наличием эмпирически, либо математически обоснованных правил порождения тестовых описаний, правил их хранения и учета. Полнота обеспечивается наличием в спецификациях наборов соответствующих характеристик, выполняющих роль эталонных. Контролируемость наряду с полнотой обеспечивается автоматической генерацией соответствующих результативных моделей и их имитацией, а документированность – генерацией соответствующих отчетов.

Соответственно для компьютерной генерации тестовых описаний необходимы:

- алгоритмы генерации сетевых структур и их параметров;
- алгоритмы и правила генерации имитационных моделей сетевых структур;
- процедуры настройки алгоритмов на заданные ограничения по сложности и режимам функционирования сетей;