

3. Baesens B., Egmont-Petersen M., Castelo R., Vanthienen J. Learning Bayesian network classifiers for credit scoring using Markov Chain Monte Carlo search // Proceedings of International Congress on Pattern Recognition. – 2002. – P.49-52.
4. Bicer I., Sevis D., Bilgic T. Bayesian credit Scoring Model with Integration of Expert Knowledge and Customer Data // International Conference 24th in EURO Conference “Continuous Optimization and Information Technologies in the Financial Sector” (MEC EurOPT 2010). – 2010. – P.324-329.
5. Ghodselahi A.A Hybrid Support Vector Machine Ensemble Model for Credit Scoring // International Journal of Computer Applications. – 2011. – Vol.17, No.5. – P.1-5.
6. Huang J., Tzeng G., Ong C. Two-stage genetic programming (2SGP) for the credit scoring model // Applied Mathematics and Computation. – 2006. – Vol. 174. – P.1039-1053.
7. Bastos J.A. Credit Scoring with Boosted Decision Trees // City: Munich Personal RePEc Archive, P. 262-273.
8. Кочедыков Д. А., Ивахненко А. А., Воронцов К. В. Система кредитного скоринга на основе логических алгоритмов классификации // Математические методы распознавания образов-12. — М.: МАКС Пресс, 2005. — С. 349–353.

338.4

*Солдатов М.А., к.ф.-м.н., доцент,
Горбунов О.В., магистрант,
ТНУ имени В.И. Вернадского*

АНАЛИЗ ВАЖНОСТИ ИЗУЧЕНИЯ РИСКОВ ЭТАПА КОДИРОВАНИЯ ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНЫХ ПРОДУКТОВ

В современном мире наблюдается тенденция увеличения роли информационных технологий в жизни общества. Однако информационные технологии являются сравнительно весьма молодой и динамично развивающейся сферой жизни общества, поэтому объективно существует недостаток научных исследований в данной области. Как правило, информационные технологии на современном этапе развития науки рассматриваются с точки зрения следующих аспектов:

Физико-математического. Информационные процессы рассматриваются в своей глубинной сути. Изучаются физические процессы, происходящие при работе компьютерных устройств, а также математические алгоритмы, которые определяют направленность упомянутых физических процессов.

1. Программистского. Данный аспект получил мало развития как предмет научного исследования. Работы в данной области имеют в основном сугубо практический характер и не систематизированы в научном формате.

Управленческого. Процесс разработки программных продуктов рассматривается с аспекта менеджмента, как науки.

Необходимость научных знаний для управления разработкой программного обеспечения стала проявляться, когда проекты стали настолько крупными, что управлять без определенного системного подхода уже стало нельзя. Таким образом, наука менеджмента разработки программного обеспечения имеет отношения по большей части только к крупным проектам. Действительно, на практике большинство методов организации разработки для небольших проектов являются избыточными и вредными. Это связано с тем, что современный менеджмент практически не уделяет внимания вопросам организации самого процесса кодирования. А так как для небольших проектов этот процесс имеет значительно большую долю от всего процесса разработки программного обеспечения, чем в больших проектах, выясняется, что современный менеджмент эффективней не применять в формате устоявшихся книжных практик, чем применять.

Публикации на тему анализ рисков разработки программных продуктов в основном принадлежат иностранным исследователям: Don Gotterbarn (2005) “Responsible Risk Analysis for Software Development: Creating the Software Development Impact Statement”[1], Ropponen J., N. Lyytinen and N. Kalle (2000) “Components of Software Development Risk: How to Address them? A Project Manager Survey”[2], Highsmith, J. (1997) “Agile Project Management”[3], Watson R. T., Turban, E., Mclean E., and Wetherbe J. (2009) “Information Technology for Management, Improving Quality and Productivity”[4], Schmidt R. Lyytinen K., Keil M., And Cule P. (2001) ”Identifying Software Project Risks: An International Delphi Study”[5].

Рассмотренные научные работы игнорируют или уделяют недостаточное внимание проблеме анализа рисков, которые возникают в процессе этапа кодирования разработки программных продуктов. Однако данные риски существуют и занимают важное место в списке рисков проектов

программной разработки продуктов, поэтому заслуживают более пристального внимания и комплексного анализа.

Целью работы является обоснование необходимости внимания к анализу факторов риска, присутствующим в процессе кодирования разработки программных продуктов. Для достижения цели данной работы рассмотрим процесс разработки и сравним уровни риска на данных этапах, а также возможность и эффективность управления ими со стороны менеджмента.

Процесс разработки состоит из множества подпроцессов, или дисциплин, некоторые из которых показаны ниже. В модели водопада они идут одна за другой, в других процессах их порядок или состав изменяется. Парадигма программирования. Парадигма программирования — это система идей и понятий, определяющих стиль написания компьютерных программ, а также образ мышления программиста[6]. Как этап разработки парадигма программирования предполагает выбор модели программирования, а также выбор подходов и приемов, которые будут эффективны в процессе создания программного продукта.

Как правило, такой выбор осуществляет архитектор проекта, причем зачастую данный вопрос не стоит перед архитектором напрямую, так как в зависимости от требований бизнес-модели очевидно на основании накопленного опыта какие подходы следует использовать. Необходимо также учитывать, что большая часть проектов выполняется в рамках существующих каркасных сред разработки, которые в себе изначально сильно ограничивают подобный выбор.

Таким образом, в данной задаче не существует существенных рисков и ей сложно управлять представителям менеджмента, так как: задача занимает очень короткое время, решается небольшим количеством людей, существует множество аналогичных прецедентов, опираясь на которые принимается решение.

Бизнес-моделирование. Данная задача решается путем анализа предоставленной предприятием документации и опроса сотрудников. Как правило, этот процесс осуществляется заказчиком.

Процесс бизнес-моделирования включает в себя минимальный уровень рисков и практически не имеет точек контроля для менеджмента.

Анализ требований. Анализ требований — это процесс сбора требований к программному обеспечению (ПО), их систематизации, документирования, анализа, выявления противоречий, неполноты, разрешения конфликтов в процессе разработки программного обеспечения[7].

Анализ требований включает три типа деятельности:

1. Сбор требований: общение с клиентами и пользователями, чтобы определить, каковы их требования.

2. Анализ требований: определение, являются ли собранные требования неясными, неполными, неоднозначными, или противоречащими, и затем решение этих проблем.

3. Документирование требований: Требования могут быть задокументированы в различных формах, таких как простое описание, сценарии использования, пользовательские истории, или спецификации процессов.

4. Разработка архитектуры. Архитектура - это набор значимых решений по поводу организации системы программного обеспечения, набор структурных элементов и их интерфейсов, при помощи которых komponуется система, вместе с их поведением, определяемым во взаимодействии между этими элементами, компоновка элементов в постепенно укрупняющиеся подсистемы, а также стиль архитектуры, который направляет эту организацию - элементы и их интерфейсы, взаимодействия и компоновку [8].

Разработку архитектуры приложения осуществляет архитектор или несколько архитекторов, предварительно разбив архитектуру на определенные модули и выполняя архитектуру каждого в отдельности.

В данном случае успех этапа зависит в основном от успешности предыдущих этапов, то есть насколько их результаты будут адекватно и доступно отображать реальность, а также от опыта и интеллекта архитекторов.

Таким образом, возможности менеджеров влиять на успех данного этапа минимальны.

5. Кодирование и отладка. Данный этап является центральным и наиболее важным в процессе разработки программного обеспечения. По сути, все остальные этапы являются лишь инфраструктурой, который обслуживают кодирование, позволяют его сделать более эффективным.

Рассмотрим эти этапы с данной точки зрения подробнее:

• Выбор парадигмы программирования определяет модель, подходы и приемы кодирования, что позволяет сделать кодирование более эффективным для достижения конкретной цели, а результат будет лучше отвечать поставленным целям.

- Бизнес-моделирование определяет среду, в которой будет функционировать программа. Эти знания необходимы для выполнения анализа требований, что в свою очередь необходимо для обеспечения процесса кодирования, и непосредственно позволяет разработчикам лучше четко осознавать требования к структуре и функциональности своего кода.

- Анализ требований является необходимым источником информации для осуществления процесса кодирования. Результатом этапа анализа требования является техническое задание – документ, в котором подробно описаны все детали выполняемого проекта. Задание считается полностью выполненным, когда созданный продукт отвечает каждому пункту требований технического задания. Таким образом анализ требований является вектором кодирования, придавая ему конкретное направление.

- Разработка архитектуры конечной целью имеет определение будущей архитектуры приложения. Архитектура призвана обеспечивать гарантию того, что все части приложения, написанные разными программистами в разное время будут слажено взаимодействовать между собой.

- Тестирование также обслуживает этап кодирования. Тестирование необходимо для того, чтобы выявить отклонения показателей приложения от минимальных требований, поставленных в техническом задании, что позволит заказчику на основании договора требовать исправления найденных недочетов и, следовательно, избежать убытков, связанных с обнаружением данных недочетов в процессе коммерческой эксплуатации приложения. Таким образом, тестирование позволяет реализовать дополнительные объемы кодирования, что увеличит качество создаваемого продукта.

- Документирование приложения призвано облегчить использование созданного программного продукта, позволяя эффективней достигать целей пользователей данного программного продукта, а, следовательно, повышая его ценность. Таким образом, данный этап увеличивает потенциал программного кода, который, по сути, представляет собой любое приложение, быть использованным.

- Внедрение программного обеспечения необходимо для того, чтобы код приложения мог выполнять обслуживающие конкретный бизнес функции. Таким образом, внедрение программного продукта позволяет приложению, созданному на этапе кодирования, реализовать свое предназначение на практике.

- Сопровождение программного обеспечения необходимо для того, чтобы в дальнейшем код приложения мог продолжать приносить пользу в условиях динамически меняющихся окружающих условий среды. В состав проекта сопровождения программного обеспечения обычно входят такие этапы: анализ требований, кодирование и отладка, тестирование и документирование. Так как выше было отмечено, что данные этапы (кроме кодирования) являются вспомогательными для этапа кодирования, то и этап сопровождения программного обеспечения в целом является вспомогательным для этапа кодирования также.

6. Тестирование. Как правило, большая часть тестирования выполняется по уже существующим алгоритмам или же программами, для которых тестировщик проводит некоторую настройку. Отсутствие широкого спектра факторов, которые могут помешать успешному тестированию, ограничивает возможности менеджмента по оптимизации процесса тестирования.

7. Составление документации. Составлением документации занимается технический писатель. Часто роль технического писателя выполняет один из участников проекта, поэтому, как правило, технический писатель изначально хорошо проинформирован об особенностях структуры и функциональных особенностей приложения.

8. Внедрение. Риски внедрения программного обеспечения со стороны разработчиков напрямую связаны со свойствами самого программного обеспечения. Риски, связанные с определенными условиями фирмы-клиента относятся к зоне обслуживания самой фирмы клиента. Обучение пользователей работе с программным продуктом не является зоной высокого риска, так как результат данного мероприятия зависит в основном от подбора сотрудников для осуществления данной задачи и от квалификации обучаемых сотрудников фирмы-клиента.

9. Сопровождение. Сопровождение программного обеспечения является неотъемлемым этапом процесса разработки программного обеспечения, однако является отдельным проектом, так как одним из неотъемлемых свойств проекта является его конечность, в то время как сопровождение программного обеспечения осуществляется на протяжении всего существования проекта, величина которого (протяжения) за исключением отдельных случаев не известна заранее. Так как

сопровождение программного обеспечения является отдельным проектом, состоящих из нескольких этапов, перечисленных выше, не имеет смысла оценивать уровень риска данного этапа в отдельности в данном контексте.

На основании приведенных выше рассуждений можно сделать вывод, что предельно важной и одновременно малоизученной сферой наличия риска в процессе разработки программного обеспечения следует рассматривать этап кодирования.

Для дальнейших исследований необходимо проанализировать разновидности существующих рисков и описать характер их влияния на успешность выполнения заданий.

Автор подразделяет такие проблемы, которые сильно влияют на результат кодирования: мотивация сотрудников (mot), соответствие ума и опыта с поставленными задачами (exp), форс-мажорные обстоятельства (интернет, электричество) (fm), взаимопонимание с заказчиком (und), изменение внешних модулей (exm), степень грамотности менеджмента (men), скорость и качество программного обеспечения и физических компьютерных устройств и сетей (com), слаженность работы сотрудников на одном проекте (int), синергия сотрудников (sin), количество и степень изменений от старта проекта (ch).

Так как данные зависимости довольно сложно смоделировать математически, автор планирует использовать методы имитационного моделирования. Исходными данными будут оценки перечисленных факторов, а также опорное время выполнения задания, рассчитанное при условии идеального сочетания всех влияющих обстоятельств. В результате осуществления процесса выполнения задания, получаем фактическое время выполнения задания при конкретном стечении обстоятельств:

$$\text{Real_time} = f(\text{Ideal_time}, \text{mot}, \text{exp}, \text{fm}, \text{und}, \text{exm}, \text{men}, \text{com}, \text{int}, \text{sin}, \text{ch})$$

Данные результаты при успешном построении модели с достаточным уровнем адекватности отражения действительности могут быть использованы разработчиками в производственном процессе для оценки сроков выполнения выставленных заказчиками заданий.

Литература

1. Gotterbarn D., Rogerson S. "Responsible Risk Analysis For Software Development: Creating The Software Development Impact Statement" // Communications of the Association for Information Systems (Volume 15) – 2005. – P. 730-750.
2. Ropponen J., Lyytinen N. and Kalle N. Components of Software Development Risk: How to Address them? A Project Manager Survey // Software Engineering, IEEE Transactions on. – 2000. - Volume 26. – P. 98-112.
3. Highsmith J. Agile Project Management // Publisher: Addison-Wesley Professional. - 2009. - 824p.
4. Watson R.T., Turban E., Mclean E., and Wetherbe J. Information Technology for Management, Improving Quality and Productivity // John Wiley & Sons Inc. – 1997. - 1031 p.
5. Schmidt R. (2005) Identifying Software Project Risks: An International Delphi Study // Journal of Management Information Systems. – 2001. - №4. – P. 5-34.
6. Floyd R.W. The Paradigms of Programming Communications of the ACM, 22(8):455—460, 1979. Русский перевод см. в кн.: Лекции лауреатов премии Тьюринга за первые двадцать лет (1966—1985), М.: МИР, 1993.
7. Иан Соммервилл. Инженерия программного обеспечения / Пер. с англ. — 6-е изд. — М.: Вильямс, 2002. — 624 с.
8. Роберт У. Себеста. Основные концепции языков программирования / Пер. с англ. — 5-е изд. — М.: Вильямс, 2001. — 672 с.
9. Канер Кем, Фолк Джек, Нгуен Енг Кек Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. — Киев: ДиаСофт, 2001. — 544 с.

*Павленко І.Г., к.е.н., доцент,
ТНУ імені В.І. Вернадського*

ПРОФЕСІЙНІ КОМПЕТЕНЦІЇ В УПРАВЛІННІ ЕФЕКТИВНИМ РОЗВИТКОМ ПІДПРИЄМСТВА

На сучасному етапі розвитку економічних реалій в Україні та світі можна констатувати, що забезпечення довгострокового успіху підприємства вже визначається не тільки наявністю та можливістю залучення матеріальних і фінансових ресурсів. Ера зростання компанії внаслідок розвитку матеріальних активів закінчується. В умовах відкритості ринків матеріальних ресурсів, розширення можливостей і подолання бар'єрів їх залучення суттєво зменшується їх значущість для забезпечення конкурентних переваг. У теперішні часи роль таких нематеріальних ресурсів як