

УДК 004.4
ББК 32.973.26-018.2

НЕКОТОРЫЕ ОБЩИЕ ПОДХОДЫ К ОРГАНИЗАЦИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

© В.Б. Федотов, 2006

*Государственная публичная научно-техническая библиотека
Сибирского отделения Российской академии наук
630200, г. Новосибирск, ул. Восход, 15*

За последнее 40-летие индустрия разработки программного обеспечения (ПО) так и не стала инженерной. Главной проблемой разработки ПО являются связанные с ней риски. Большое количество готовых программных решений ускоряет процесс разработки, однако порождает множество путей решения одних и тех же задач, что, в свою очередь, увеличивает неопределенность и риски, связанные с разработкой ПО. Главной целью в работе над программным проектом является снижение рисков, одним из таких подходов к организации разработки ПО является экстремальное программирование (XP).

Ключевые слова: XP, экстремальное программирование, ПО, парное программирование, рефакторинг, снижение рисков, коллективная собственность.

Начавшая формироваться примерно 40 лет назад индустрия разработки программного обеспечения (ПО) за счет стандартизации протоколов взаимодействия между программными продуктами постепенно вырывается из первородного хаоса, продолжая при этом бурно развиваться. Создаются и эволюционируют высокоуровневые языки программирования, средства описания и управления данными, комплексные программные среды, все более изощренным становится инструментарий разработки. Сотни тысяч специалистов по всему миру каждый день работают на мировой рынок ПО /1/.

Однако за 40 лет индустрия разработки ПО так и не стала инженерной. При наличии проработанных методик организации разработки ПО (RUP, XP и т.д.) и фундаментальных подходов к организации самого ПО (ООП, процедурное программирование и т.д.) по-прежнему отсутствуют какие-либо четкие формализованные методики написания программ под нужды конкретных типов задач. Громадное количество всевозможных компонент и готовых программных решений существенно ускоряет процесс разработки, однако при этом дает право на жизнь множеству путей решения одних и тех же задач, что, в свою очередь, увеличивает неопределенность и риски, связанные с разработкой ПО /2/.

Риски разработки ПО

Главной проблемой разработки ПО, несомненно, являются связанные с ней риски. Вот лишь некоторые из них:

1. Невыполнение сроков – день сдачи программы подходит и вам придется сказать заказчику, что программа не будет готова еще 6 месяцев.
2. Проект закрыт – после множественных задержек проект отменен, так и не завершившись.
3. Засоры системы – программа успешно эксплуатируется, но спустя несколько лет цена изменений в системе или количество ошибок вырастают настолько, что программу необходимо менять.
4. Уровень ошибок – проект завершен, но уровень ошибок настолько высок, что программа так и не доходит до практического использования.
5. Непонимание целей – проект завершен, однако он не решает поставленных проблем.
6. Изменение целей – проект завершен, но проблема, для решения которой он был разработан 6 месяцев назад, сменилась другой, более насущной проблемой.
7. Достижение ложных целей – программный продукт имеет множество потенциально интересных возможностей, которые было весело реализовать, однако ни одна из них, ни их совокупность не являются тем, что принесет много пользы в решении поставленной задачи.
8. Персонал ушел – после двух лет разработки все хорошие программисты начали ненавидеть разрабатываемую программу и ушли.

Риски разработки для нужд библиотек

Рассматривая риски разработки ПО для нужд отечественного библиотечного сообщества, необходимо принять во внимание как особенности си-

туации в индустрии разработки ПО в России, так и специфику самого библиотечного сообщества. В индустрии разработки ПО в нашей стране в последние годы наблюдается существенный дисбаланс – потребности рынка в высококвалифицированных специалистах намного превосходят имеющиеся предложения, в то же время количество слабоквалифицированных специалистов превышает потребности рынка, и с каждым годом этот дисбаланс только обостряется /3/. Говоря о библиотечной сфере, стоит отметить проявившуюся в последний год тенденцию к увеличению оплаты труда молодых специалистов в научной сфере, что, возможно, окажет свое положительное влияние и на разработку ПО для библиотечного сообщества. Однако нет оснований предполагать, что ситуация с оплатой труда изменится настолько кардинально, чтобы привлечь в библиотеки молодые высококвалифицированные кадры. Разрыв между оплатой труда специалистов в библиотечной сфере и, скажем, в торговой или банковской (которые, в свою очередь, являются далеко не самыми высокооплачиваемыми областями разработки ПО) настолько велик, что даже наметившиеся положительные тенденции едва ли в ближайшее время приведут к существенному росту заинтересованности работой в библиотеках даже со стороны выпускников вузов (слабоквалифицированных специалистов). С другой стороны, растущее перенасыщение рынка слабоквалифицированными специалистами создает предпосылки для того, чтобы выпускники вузов приходили в библиотеки с целью набраться опыта, получить практические навыки работы, повысить свою квалификацию и в дальнейшем уйти работать в другие, более привлекательные организации.

Исходя из анализа ситуации, можно сделать определенные выводы и по поводу рисков разработки ПО в библиотечной сфере. Имея в качестве исполнителей в основном лишь слабоквалифицированных специалистов, можно с уверенностью сказать, что растут практически все риски, связанные с разработкой ПО. Учитывая же высокую текучесть кадров, можно сделать вывод, что особенно вырастает риск «ухода персонала», со всеми вытекающими из этого последствиями, а именно:

1. Задержки сдачи проекта.
2. Долгий период адаптации новых кадров, приводящий к еще большим задержкам.
3. Сложный, неочевидный дизайн программных систем.
4. Полная «смерть» проекта, когда ушедший специалист не оставил исходников программы или оставил исходники, в которых никто ничего не может понять.

Практика показывает, что «смерть» проекта в случае ухода одного конкретного специалиста, несмотря на кажущуюся нелепость этой ситуации,

более чем реальна и такая ситуация встречается довольно часто.

Применение техник XP как способ снижения рисков

Естественно, одной из основных целей любого управления проектом является в первую очередь снижение рисков. В случае же, когда риски высоки, а в разработке ПО для нужд библиотечного сообщества они особенно высоки, управлению проектом и правильной организации работ над проектом следует дать особый приоритет. Одним из способов такой организации работ над ПО, который имеет целью убрать большую часть рисков, является экстремальное программирование (Extreme Programming, далее по тексту – XP).

Что такое XP? Некоторым людям принципы экстремального программирования напоминают обычный здравый смысл. Отчего же слово «экстремальное» присутствует в его названии? XP основывается на принципах здравого смысла и применяет их до чрезвычайного уровня /4/. Примеры такого здравого смысла:

- если обзоры кода – это хорошо, то мы будем рассматривать код все время (парное программирование);
- если тестирование хорошо, то каждый будет тестировать все время (тестирование модулей), даже заказчики (функциональное тестирование);
- если проектирование – это хорошо, мы сделаем это частью ежедневной работы (рефакторинг);
- если простота хороша, мы будем всегда оставлять систему с самым простым дизайном, который поддерживает все функциональные возможности (самая простая вещь, которая могла бы работать);
- если архитектура важна, каждый будет работать, определяя и оттачивая архитектуру все время (метафора);
- если тестирование интеграции полезно, то мы будем интегрировать и тестировать несколько раз в день (непрерывная интеграция);
- если короткие рабочие циклы хороши, мы будем делать циклы минимальными – секунды, минуты и часы, не недели, месяцы и годы (игра планирования).

XP делает два набора обещаний:

1. Программистам XP обещает, что они каждый день будут работать над вещами, действительно имеющими значение. Им не придется сталкиваться с непредсказуемыми ситуациями в одиночку. Они будут способны делать все возможное для того, чтобы сделать их систему успешной. Они будут принимать решения, которые могут делать лучше всего и не наоборот.

2. Заказчикам и руководителям проектов XP обещает, что они получат максимально возможный эффект от каждой недели программирования. Каждые несколько недель они будут способны видеть конкретное продвижение к цели. И, что особенно важно, они будут способны изменить направление проекта в середине его развития без непомерных затрат на это /4/.

Говоря коротко, XP обещает уменьшить риск проекта, улучшить возможности отклика на изменение целей, повысить производительность рабочего процесса в течение всего цикла жизни системы и добавить интерес к созданию программного обеспечения в командах – все это одновременно.

Вот основные техники XP:

- **игра планирования** – определяется масштаб следующего релиза, объединяя стоящие задачи и технические оценки. Если реальность не соответствует плану, план модернизируется;

- **маленькие релизы** – быстро делается простая система, затем выпускаются новые версии через очень короткие интервалы;

- **метафора** – направляет всю разработку системы простым доступным описанием того, как вся система работает;

- **простой дизайн** – система разрабатывается настолько просто, насколько возможно в любой данный момент времени. Дополнительная сложность удаляется, как только она будет обнаружена;

- **тестирование** – программисты непрерывно пишут тесты модулей, которые должны проходить безупречно для того, чтобы разработка продолжалась дальше. Заказчики пишут тесты, демонстрирующие, что функциональные особенности закончены;

- **рефакторинг** – программисты реструктурируют систему, не изменяя ее поведение, в целях удаления дублирования, улучшения связей, упрощения или добавления гибкости;

- **парное программирование** – весь программный код пишется двумя программистами на одной машине;

- **коллективная собственность** – любой может изменить любой код в системе в любое время;

- **непрерывная интеграция** – как только какая-то задача реализована, она сразу же интегрируется в систему, интеграции происходят много раз в день;

- **40-часовая рабочая неделя** – не больше чем 40 часов в неделю, как правило. Никогда не работайте сверхурочное время две недели подряд;

- **доступность заказчика** – в команду включается настоящий пользователь, способный ответить на вопросы на протяжении всей рабочей недели;

- **стандарты кодирования** – в целях улучшения общения между разработчиками на уровне

программного кода программисты пишут весь код в соответствии с принятыми правилами /5/.

Сведя эти техники вместе, про разработку в стиле XP можно сказать, что:

а) Пары программистов работают вместе.

б) Разработка направляется непрерывным тестированием. Вы сначала тестируете, затем кодируете. Пока все тесты не пройдут, вы не считаете задачу выполненной. Когда все тесты пройдены, и вы не можете придумать тест, который мог бы упасть, это значит, что вы закончили добавлять функциональную возможность.

в) Пары программистов не только придумывают тесты и заставляют их работать. Они также постоянно развивают дизайн системы. Любой код может быть просмотрен и изменен любым человеком в команде. Парная работа повышает эффективность анализа, проектирования, реализации и тестирования системы, т.е. везде, где система в этом нуждается.

г) Интеграция немедленно следует за реализацией, включая тестирование самой интеграции.

Парная работа за одним компьютером, непрерывное тестирование, постоянная доработка дизайна и кода (рефакторинг), интеграция сразу после реализации новой функциональности – все эти техники XP вместе призваны в первую очередь добиться того, чтобы и дизайн, и сам код программы было легко модифицировать в любое время жизни проекта, как на этапе первичного проектирования системы, так и спустя годы после запуска программы в промышленную эксплуатацию.

Плюсы XP в разработке ПО для нужд библиотечного сообщества

Чтобы понять возможные плюсы от применения техник XP, сопоставим вместе техники XP и основные факторы, характеризующие разработку ПО для библиотек. Низкая квалификация исполнителей, высокая текучесть кадров, зачастую заинтересованность исполнителей не столько в успешном завершении проекта (за который им все равно мало платят), сколько в повышении квалификации и смене работы на более выгодную – со всеми этими негативными факторами можно успешно справляться, применяя XP.

Так, например, парное программирование позволяет повысить квалификацию исполнителей намного быстрее, чем одиночное. Лучше взять двух выпускников вуза и посадить их вместе делать модуль за модулем, чем, пытаясь распараллелить работу, посадить тех же выпускников делать два разных модуля параллельно. Во втором случае низкая квалификация исполнителей обусловит высокую вероятность того, что какой-нибудь из модулей будет сильно задержан или во-

обще не будет сделан и поставит под сомнение выполнение всего проекта.

В свою очередь, коллективная собственность на программный код также позволяет программистам быстрее повышать квалификацию. Кроме того, общий программный код полностью исключает возможность возникновения абсурдной ситуации, когда один из программистов уходит и уносит с собой часть кода, убивая тем самым долго разрабатываемый и возможно почти законченный проект. И сильно уменьшает эффекты от похожей ситуации, когда один из исполнителей уходит, код остается, но никто не может понять, что этот код делает и как его изменить. На самом деле, коллективная собственность на код и парное программирование вместе полностью исключают возможность такой ситуации в ХР – оставшиеся члены команды обязательно в той или иной степени будут разбираться во всем коде программы, что позволит им быстро вникнуть даже в те участки кода, которые разрабатывались преимущественно кем-то другим в команде.

Простой дизайн и принятые стандарты кодирования обеспечат возможность быстро разобраться в проекте новым членам команды. А постоянно разрабатываемые и поддерживаемые тесты модулей обеспечат уверенность в работоспособности

кода, сделанного даже несколько лет назад совершенно другим программистом.

В заключение можно сказать, что ХР – это весьма широко применяемый в мире подход к разработке ПО. В нашей стране ХР также нашло широкое применение в индустрии, однако конкретно в сфере разработки ПО для библиотек техники ХР применяются мало. В случае же, когда разработкой программного обеспечения занимаются преимущественно слабоквалифицированные специалисты, применение современных методик управления программным проектом становится особенно актуальным.

Список литературы

1. 3DNews – Daily Digital Digest [Электронный ресурс]. – Режим доступа: <http://www.3dnews.ru/>
2. *Игумнов, Е.* Software Project Manager среднего проекта – кто он [Электронный ресурс] / Е. Игумнов. – Режим доступа: <http://www.citforum.ru/programming/digest/spm.shtml>
3. Новосибирский городской сайт «НГС» [Электронный ресурс]. – Режим доступа: <http://www.ngs.ru/>
4. *Beck, K.* Extreme Programming explained : Embrace change / К. Beck. – Addison-Wesley, 2000. – 224 p.
5. *Auer, K.* Extreme Programming Applied : Playing to Win / К. Auer, R. Miller. – Addison-Wesley, 2002. – 384 p.

Материал поступил в редакцию 27.09.2006 г.

Сведения об авторе: *Федотов Вадим Борисович* – аспирант ГПНТБ СО РАН, инженер отдела автоматизированных систем, тел. (383) 266-71-33, e-mail: fedot_st@online.nsk.su