

УПРАВЛЕНИЕ КАЧЕСТВОМ ПРОГРАММНОГО ПРОДУКТА НА ПРИМЕРЕ СЛОЖНОЙ РАСПРЕДЕЛЕННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Доронкин Д.Ю., Медведев Е.А., Кудинов А.В.

Научный руководитель: Кудинов А.В., к.т.н.

Томский политехнический университет, 634050, Россия, г. Томск, пр. Ленина, 30

E-mail: paranamix2@mail.ru

Введение

Ввиду слабой освещенности принципов управления качеством программного продукта, в странах бывшего СССР сложилось определенное отношение к тестированию, как к роли второго плана, а к тестированию – как к конечному этапу разработки системы. Такая ситуация не соответствует современным реалиям и требует пересмотра и управление качеством должно стать непрерывным процессом на всех этапах жизненного цикла программного обеспечения (ПО). В связи с этим возникают серьезные вопросы. Как сократить расходы и повысить качество программного продукта? Какую часть бюджета выделить на тестирование и к чему приведет сокращение расходов на управление качеством? Какие ресурсы команды потребуются? Какие инструментальные средства надо использовать?

В сфере разработки ПО успешность компании определяется ее прибылью. В связи с этим, можно поставить главный вопрос: какую часть бюджета и кадровых ресурсов необходимо выделить для построения эффективного процесса управления качеством разрабатываемого продукта и как данный процесс повлияет на конечную прибыль?

Управление качеством требует систематического применения методов повышения качества, которое стало бы неотъемлемой частью процесса разработки. На сегодняшний день не существует универсальной модели управления качеством ввиду следующей зависимости: чем больше количество вариантов использования системы, тем сложнее ее протестировать.

В данной работе будут предложены решения проблемы контроля качества ПО, способствующие оптимизации процесса тестирования. Предлагаемые решения основаны на опыте работы команды разработчиков сложной распределенной информационной системы.

Описание рассматриваемой информационной системы

Система представляет собой корпоративную геоинформационную систему управления непрерывным производством компании по добыче полезных ископаемых, выполняющую широкий спектр задач: сбор, хранение, обработка, анализ, ввод-вывод, интеграция данных; ведение графиков и журналов; планирование работ и формирование производственных отчетности; мониторинг технологических процессов и др.

Система предоставляет удобный интерфейс пользователя, имеет многоуровневую сложную модель данных, развернутую на трех удаленных друг от друга серверах, объединенных системой репликации данных. Управление качеством такой системы затратный и трудоемкий процесс из-за большого количества возможных вариантов ее использования и обрабатываемых данных [1].

Основные проблемы управления качеством на примере рассматриваемой системы

Рассматриваемая геоинформационная система работает с непрерывным потоком данных, многие из которых не могут быть доступны разработчикам. В связи с этим, компании, осуществляющей разработку и сопровождение системы, необходимо иметь собственную тестовую среду, имитирующую «боевой сервер». Для этого необходимо ежедневно развертывать актуальную базу данных (БД) как разработчикам, так и тестировщикам. Данная процедура в зависимости от объема БД и мощности сервера может занимать до 30-40 минут. На настройку параметров уходит около 10 минут, время ожидания развертывания занимает до 25 минут. Эти цифры можно отнести к потерям, т.к. специалист не способен решать задачи в этот временной отрезок. Тестировщики часто развертывают актуальную версию БД более двух раз в день, что автоматически удваивает для них время бездействия. В зависимости от количества работающих сотрудников и уровня их заработной платы можно подсчитать, что убытки на время простоя составят достаточно крупную сумму.

Разработка масштабной распределенной системы требует грамотного планирования и организации работы. Помимо выполнения поставленных руководителем задач необходимо в непрерывном режиме фиксировать изменения в разрабатываемом проекте, ошибки и пожелания пользователей, контролировать процесс их устранения/выполнения, организовать централизованную среду ведения диалога между участниками команды разработки с доступом к документации и рабочим элементам проекта.

Как показывает практика, далеко не всегда данные задачи решаются своевременно, тогда, как при правильном планировании можно существенно сэкономить время и структурировать процесс разработки.

Еще одной важной проблемой при разработке ПО можно выделить отсутствие непрерывного тестирования требований к системе. Поскольку

чаще всего каждый разработчик несет ответственность за свою часть системы, необходимо тестировать как каждый блок по отдельности, так и работоспособность всей системы, т.к. при интеграции блоков в единую систему могут проявиться дефекты. При их своевременном обнаружении имеется возможность систематизировать исправление однотипных ошибок в разных частях программы, что сократит трудозатраты на разработку и позволит исключить их появление в будущем.

Предлагаемые решения поставленных проблем управления качеством

Одним из основных решений имеющихся проблем в области управления качеством ПО является использование специализированных инструментов разработки. По мнению экспертов в области контроля качества, таких как М. Мескон Д. Эванс [2], использование специализированного инструментария и структуризация процесса тестирования позволяет достичь максимального эффекта управления качеством. Идеи экспертов легли в основу концепции Test-drive development (TDD – разработка через тестирования), разработанной Кентом Бекон. Данная техника разработки ПО является одним из ярких примеров достижения высокого уровня качества разрабатываемого продукта при соблюдении установленных заранее принципов.

Однако внедрение TDD накладывает определенные затраты по ресурсам и времени, которые может себе позволить не каждая компания. В связи с этим было решено оптимизировать управление качеством за счет использования специализированных средств разработки, таких как Team Foundation Server (TFS).

TFS позволяет: автоматизировать развертывание среды тестирования; обеспечить непрерывный диалог между всеми участниками команды разработки; обеспечить доступ к задачам проекта; организовать процесс фиксирования статуса работы по рабочим элементам (задачи, ошибки, заявки).

Данное решение позволяет существенно экономить затраты на выпуск продукта. Если в разработке задействовано 15 человек, в тестировании – 3, а максимальное время простоя достигает 25 (для тестировщиков – 50) минут в сутки, то верхнюю границу экономии временных ресурсов можно оценить в:

$$\frac{15 \text{ сотрудников} * 25 \text{ минут} * 22 \text{ рабочих дня}}{480 \text{ рабочих минут в день}} = 17 \text{ рабочих дней, для разработчиков}$$

$$\frac{3 \text{ сотрудника} * 50 \text{ минут} * 22 \text{ рабочих дня}}{480 \text{ рабочих минут в день}} = 6 \text{ рабочих дней, для тестировщиков}$$

Суммарная экономия составляет до 23 человеко/дней в месяц.

Кроме этого, TFS упрощает взаимодействие разработчика и тестировщика, предоставляет удобный интерфейс для работы с проектом. Автоматизирование некоторых видов работ позволяет тестировщику перераспределить свое рабочее время.

Во-первых, тестировщик может участвовать в планировании проекта наравне с разработчиками. Это позволяет изначально сформировать свое видение разрабатываемой системы, принципы ее работы, определить возможные слабые места в разрабатываемой системе и уделить им особое внимание при тестировании, а также начать процесс тестирования параллельно разработке.

Во-вторых – взять на себя часть организационных работ: распределение обязанностей между разработчиками, документирование рабочих элементов. Тем самым тестировщик не только управляет качеством ПО, но и структурирует процесс разработки, что упрощает работу всей команды.

Изначальное представление о разрабатываемой системе позволяет тестировщику управлять качеством параллельно этапу разработки. Данный подход к управлению качеством позволяет находить и исправлять ошибки практически сразу после их обнаружения. Это удобно как для разработчика, допустившего ошибку, которому не нужно спустя определенное количество времени возвращаться на несколько этапов назад для исправления «старого бага», так и для других разработчиков, знающих природу ошибки.

Своевременное обнаружение, исправление и систематизация ошибок позволяет существенно сократить трудозатраты на разработку ПО и повысить качество минимизацией возможного количества дефектов.

Заключение

Применение предложенных решений по контролю качества разрабатываемого продукта существенно сокращает затраты на производство и сопровождение программных продуктов. Внедряются современные инструменты работы с проектом, повышается квалификация тестировщика за счет его внедрения в процесс проектирования разрабатываемой системы.

Работа выполнялась по тематике госбюджетной НИР № 4.1316.2014 в рамках государственного задания «Наука».

Список использованных источников

1. Проблемы автоматизации производства газодобывающих компаний: монография – А.В. Кудинов, Н.Г. Марков, Томск: изд-во ТПУ, 2012 – 247 с.
2. Управление качеством: учебное пособие – Д. Эванс, Изд.: Юнити-Дана, 2007 – 672 с.