**Реферат**

**«КОМПЛЕКСНАЯ АВТОМАТИЗАЦИЯ ПРОЕКТОВ РАЗРАБОТКИ ПО В УСЛОВИЯХ КРИЗИСА»**

**ВВЕДЕНИЕ**

В настоящий момент многие компании – разработчики ПО столкнулись с проявлениями кризиса в виде падения объемов продаж, замедления роста бизнеса, ужесточения требований банков к получателям кредитов. Чтобы сохранить эффективность, организациям требуется существенно повысить результативность процесса разработки ПО. Для этого необходимо решать сложные проблемы, связанные с:

 *планированием* (расхождение первоначально запланированных и фактических трудозатрат в процессе реализации проекта);

 *управлением задачами* (потеря со стороны менеджера проекта контроля над задачами и сроками их выполнения, в том числе отсутствие понимания, сколько времени сотрудники реально тратят на ту или иную работу, какова производительность проектной команды и т.д., что особенно актуально для больших распределенных команд);

 *управлением изменениями* (заказчик в процессе разработки меняет требования, при этом внесение изменений отдельно не оплачивается; как следствие, чтобы сдать проект, команде приходится увеличивать объем работ и, соответственно, трудозатраты);

 *приемо-сдаточными испытаниями* (отсутствие четких критериев сдачи-приемки проекта приводит к недовольству заказчика и, как следствие, к затягиванию сроков завершения проекта и появлению незапланированных работ);

 *управлением рисками* (отсутствие формализованного управления рисками может привести в процессе реализации проекта к событиям, которые потребуют изменения объема работ);

 *проектными коммуникациями*, как внутрикомандными, так и с заказчиком (время на коммуникации тратится неэффективно, особенно эта проблема касается географически распределенных команд);

 *конфигурационным управлением* (программный код хранится локально, не поддерживается версионность; сборка производится спорадически, в нее случайно включается устаревший, неактуальный код; силы команды тратятся на выяснение актуальной версии сборки).

Все вышеперечисленные проблемы ведут к одному результату – расхождению значений изначально запланированных и фактических трудозатрат, что снижает показатели эффективности и доходности проектов.

Решение перечисленных проблем весьма затруднительно без внедрения автоматизированной системы управления. Автоматизация поможет создать прозрачную среду управления, связав проектные активности на всем протяжении жизненного цикла проекта в единое целое, что, в свою очередь, позволит эффективно управлять проектными трудозатратами.

Считается, что автоматизация проектной деятельности – достаточно дорогостоящее мероприятие. Тем более сложно ожидать, что руководство компании пойдет на дополнительные затраты в условиях кризиса. Найти выход из подобной ситуации можно, используя готовые решения, интегрированные в единую систему.

В свое время компания «Люксофт», проанализировав и опробовав в реальных проектах ряд доступных продуктов, создала интеграционное решение для автоматизации процесса разработки – LUXproject.

При создании данного решения были поставлены следующие цели:

 комплексная поддержка жизненного цикла разработки ПО (от заключения контракта до поддержки клиента);

 поддержка распределенной разработки;

 поддержка различных методологий разработки;

 снижение затрат на разработку и внедрение системы за счет использования готовых решений (как коммерческих, так и бесплатных).

LUXproject по сути является «процессной оболочкой», которая интегрирует в себе следующие технологии и продуты: JIRA, Confluence, GreenHopper, Cruise Control, Subversion, FishEye, WebDAV и др.

Они хорошо зарекомендовали себя на рынке, накоплен богатый опыт их использования.

Одним из достоинств данных продуктов является большой потенциал их кастомизации. Существуют целые сообщества программистов, занимающихся разработкой open-source-плагинов1 для JIRA и Confluence. Также на рынке имеется много коммерческих плагинов, существенно расширяющих функционал.

В результате было создано решение, которое можно отнести к системам ALM (Application Lifecycle Management), объединяющим в себе:

 инструменты управления проектами и процессами;

 инструменты разработки и тестирования;

 средства коммуникаций для распределенных команд.

Поскольку учет трудозатрат осуществляется на всем протяжении разработки ПО, то автоматизация управления должна «пронизывать» все проектные процессы жизненного цикла создания ПО.

При этом чтобы не вводить читателя в заблуждение, стоит отметить, что речь идет не о специальных инструментах разработки или тестирования, а в первую очередь о средствах управления. При этом не исключается интеграция инструментов, используемых для управления и для непосредственной работы.

Далее будут описаны основные направления, концепции и способы автоматизации проектов разработки ПО, благодаря которым возможно эффективно управлять проектными трудозатратами, предотвращая появление незапланированных трудозатрат. В качестве практического примера будет рассмотрена реализация функционала LUXproject.

**1. Планирование и контроль выполнения задач**

При управлении проектами важно не только правильно спланировать работы, но и отслеживать их выполнение, постоянно контролируя ход проекта.

Применительно к проектам разработки ПО одним из недостатков популярных систем календарного планирования и управления (типа MS Project) является отсутствие возможности связывать задачи с другими проектными активностями и устанавливать зависимость между элементами их жизненных циклов.

Для решения данной проблемы все чаще используются системы с возможностью отслеживать состояние проектных активностей (issue tracking).

*Трекер* (от англ. track, что значит «след») – специальная система для отслеживания состояния проектных активностей (задач, требований, дефектов). Подобные системы появились в конце 1990-х гг. для обнаружения ошибок в коде при проведении тестирования в проектах разработки ПО. Впоследствии выяснилось, что можно успешно применять эти системы для отслеживания и других проектных активностей. Обычно в трекере для каждой проектной активности установлен определенный жизненный цикл, однако во многих представленных на рынке системах предусмотрена возможность гибкой настройки жизненного цикла в зависимости от требований проекта.

В LUXproject применяется трекер JIRA компании Atlassian. Благодаря гибкой архитектуре трекера можно создавать разнообразные жизненные циклы с учетом проектных требований.

Например, в компании используются несколько методологий ведения проектов разработки ПО:

 agile-практики;

 методологии, основанные на RUP-подобных процессах;

 проекты, связанные с поддержкой пользователей.

В этой ситуации для каждого процесса создается свой уникальный набор жизненных циклов, соответствующий конкретным требованиям методологии разработки ПО.

Возможность связывать между собой задачи, требования и дефекты позволяет создавать трассируемость проектных активностей, оперативно отслеживая цепочку изменений в проекте.

При этом трекер не заменяет систем календарного планирования, а дополняет их. Идеальная связка системы календарного планирования и трекера выглядит следующим образом: вначале руководитель проекта разрабатывает структуру декомпозиции работ (этапы, задачи, подзадачи) и формирует команду (планирует ресурсы), затем он экспортирует информацию о задачах в трекер (посредством специального модуля экспорта данных). В процессе реализации проекта рядовые сотрудники изменяют статус задач в трекере, связывают задачи с другими проектными активностями, заносят в систему данные о времени выполнения работы. Руководитель проекта периодически переносит информацию из трекера в систему календарного планирования, где проводит комплексный анализ изменений в ходе проекта и осуществляет соответствующие корректирующие и предупреждающие действия.

**2. Интеграция и единый пользовательский интерфейс**

Специфика современных проектов разработки ПО такова, что в процессе создания и внедрения ПО участники проектной команды могут возвращаться к пройденным этапам (например, выполнив тестирование, команда разработчиков может вернуться к разработке; после внедрения иногда возникают задачи, связанные с изменением требований, и т.д.). Таким образом, для команды актуально не только быстрое получение доступа к данным из различных проектных областей, но и наличие информации об истории изменений в проекте.

В LUXproject решение этой задачи достигнуто посредством единого визуального интерфейса, благодаря которому участник команды может быстро запустить тот или иной функциональный модуль системы. Сведения о различных проектных активностях передаются с помощью специального внутреннего протокола данных, что позволяет экономить время на доступ к системам и снижает риск передачи и конвертации данных между различными функциональными модулями.

Следует отметить, что интеграция wikiдвижка, трекера, версионного репозитория и инструментов разработки в едином визуальном интерфейсе (рис. 2) имеет существенные достоинства, такие как возможность создавать прямые ссылки между различными функциональными модулями и наличие прозрачных переходов из одного функционального модуля в другой.

Например, в LUXproject интегрирован wikiдвижок Confluence компании Atlassian, который позволяет выводить на страницу информацию из других систем, формируя ее расположение и параметры отображения путем простейших конфигурационных настроек, благодаря чему можно быстро адаптировать проект к конкретным требованиям внутренней и внешней среды.

**3. Проектные коммуникации**

У распределенных команд, особенно находящихся в различных часовых поясах, могут возникнуть проблемы с передачей оперативных данных в ходе проекта и представлением актуальных проектных артефактов. Самый распространенный способ решения этих проблем заключается в предоставлении доступа к некой информационной системе с помощью различных протоколов и методов, например доступ через VPN. Однако последний не всегда удобен, особенно если надо организовать доступ с территории заказчика, где существует жесткая политика информационной безопасности.

Более удобен доступ через Интернет по протоколу HTTPS. Благодаря использованию этого протокола и единого веб-интерфейса возможно организовать круглосуточный доступ к системе из любой географической точки, где имеется подключение к Интернету. Тем самым экономится время на получение информации.

Одним из наиболее удобных средств коммуникаций для распределенных команд является вышеупомянутый wiki-движок, благодаря которому можно использовать такие инструменты обмена информацией, как форумы, комментарии к проектным активностям, базы знаний. В результате сокращается время на переписку, вся информация хранится в едином месте, можно организовать для заказчика доступ к данным с возможностью оставлять комментарии в проектных активностях.

**4. Управление рисками**

В современных условиях, когда на проект оказывает влияние как внутренняя, так и внешняя среда, управление рисками становится обязательным компонентом любого проекта независимо от отрасли.

Успешное управление рисками в проекте заключается не только в создании списка рисков и их оценке, но и в непрерывном отслеживании рисков с точки зрения их актуальности для проекта и оперативном принятии решений при наступлении рисковых событий.

Как показывает практика, отслеживание рисков с использованием подручных средств, например с помощью таблиц MS Excel, вполне осуществимо, но зачастую к концу проекта такой контроль практически сходит на нет, т. к. руководитель проекта уделяет основное внимание другим, более актуальным проектным задачам. Поэтому при автоматизации управления рисками в проекте автоматизация процесса оценки актуальности рисков находится на первом плане.

В системе заложен следующий механизм контроля за рисками и управления ими. Риск оценивается с точки зрения «вероятности свершения» и «влияния на проект». Оценка осуществляется по балльной шкале. Затем автоматически вычисляется цена риска (произведение его вероятности и влияния). В зависимости от полученного значения устанавливается срок оценки актуальности риска. При наступлении даты проверки соответствующий риск отображается на специальной странице руководителя проекта и система автоматически высылает по электронной почте письмо-уведомление. Описанный механизм обеспечивает непрерывный контроль за рисками на всем протяжении жизненного цикла проекта.

Механизм интеграции wiki и трекера позволяет давать ссылки из каждого описания риска на тот или иной проектный артефакт или связывать риск с другой проектной сущностью (задачей, требованием, дефектом). Этот механизм обеспечивает сквозной контроль за источниками риска и при наступлении рискового события позволяет оперативно составить план действий по минимизации его последствий.

**5. Проектные шаблоны**

После того как система поддержки проектов разработки ПО приобретена и развернута, пользователи могут столкнуться со следующей ситуацией. В компании реализуется множество проектов с использованием различных моделей разработки, будь то классические модели, базирующиеся на RUP-подобных процессах, или модели, основанные на agile-практиках. К тому же довольно час то специфика проекта определяется требованиями заказчика, поэтому проектная команда может потратить довольно много времени на конфигурирование системы в соответствии с требованиями конкретного проекта.

Для минимизации времени настройки систем существуют так называемые проектные шаблоны.

Это совокупность настроек функциональных модулей системы (состоящих из issues (проектных активностей с жизненным циклом), wiki-контента, версионного репозитория, шаблонов документов и базы знаний) под конкретную методологию разработки и требования заказчика. В результате настройка системы сводится к простому выбору руководителем проекта проектного шаблона из соответствующего каталога во время инициализации проекта и применению его к конкретному проекту в системе.

Например, в LUXproject существуют преднастроенные проектные шаблоны для управления проектами по классическим моделям, аgile, шаблоны для управления группой проектов (проектный офис), шаблоны для службы поддержки. Это позволяет руководителю проекта оперативно развертывать проектную среду в системе. Продемонстрировать экономию времени на настройку модели проекта можно, сравнив продолжительность настройки модели проекта в LUXproject и в трекере, где отсутствуют проектные шаблоны. В LUXproject создание нового проекта и применение проектного шаблона занимает от трех до десяти минут, и осуществить их может руководитель проекта. Для настройки трекера в соответствии с конкретными требованиями проекта может понадобиться конфигурационный инженер, и, как показал опыт компании «Люксофт», этот процесс может занять до двух рабочих дней.

**Выводы**

В условиях кризиса особо остро стоит вопрос о сокращении издержек производства. В проектах разработки и внедрения ПО основными затратами являются:

 трудозатраты участников проекта;

 затраты на покупку лицензий и поддержку ПО, необходимых для разработки продукта;

 затраты на аренду помещений.

Внедрение автоматической системы, которая базируется на веб-интерфейсе и интегрирует в себе различные продукты и технологии, поддерживающие разработку ПО на всем жизненном цикле проекта, позволяет снижать издержки за счет:

 построения более прозрачной схемы управления;

 распределенного доступа к инструментам всей проектной команды и заказчика;

 быстрого поиска информации;

 непрерывного управления рисками.

Управление активностями проекта становится более прозрачным при использовании в планировании и управлении связки системы календарного планирования и трекера. Высокоуровневое планирование проекта, выбор ресурсов, расчет трудозатрат и бюджета осуществляются в системах календарного планирования, а оперативное управление работами команды при реализации проекта выполняется с помощью трекерсистемы.

Связывание проектных активностей в трекере с помощью различного типа ссылок обеспечивает взаимосвязь всех компонентов проекта, что экономит время при отслеживании изменений в проекте, снижает вероятность ошибок и рисков (за счет более точной оценки рисков и выявления их возможных источников).

Веб-интерфейс системы позволяет организовать работу распределенной команды и оперативное взаимодействие с заказчиком в любой географической точке. Заказчик, получив доступ с определенными правами, видит, что реализация проекта осуществляется по плану и все запросы и эскалации имеют подтверждение в соответствующей проектной активности, в результате чего доверие исполнителя и заказчика друг к другу повышается.

Интеграция всех инструментов в единый интерфейс позволяет также организовать работу с региональными офисами и разработчикамифрилансерами. Разработчики, где бы они ни находились, могут войти в систему и получить доступ ко всем инструментам разработки и тестирования, не устанавливая их на свой компьютер, а менеджер проекта может осуществлять прозрачный контроль за их работой.

Использование при интеграции wiki-системы обеспечивает информационную составляющую рабочей среды, что существенно сокращает время, затрачиваемое исполнителями на изучение регламентирующей и рабочей документации, и снижает вероятность ошибок при выполнении задач.

При этом отпадает необходимость в каждом офисе компании «городить огород» систем для управления и непосредственной разработки ПО – достаточно установить систему на один или несколько серверов, что, в свою очередь, упрощает ведение лицензионной политики в компании.