

Программная поддержка контрольных точек жизненного цикла информационных систем

Минич А.М.

Кафедра интеллектуальных информационных технологий
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
e-mail: anastacia1988@inbox.ru

Аннотация—В работе рассматривается поддержка контрольных точек жизненного цикла информационных систем

Ключевые слова: информационная система; контрольная точка; программное средство; жизненный цикл; методологии; проект

I. ВВЕДЕНИЕ

На современном этапе существования информационных систем и технологий человек стремится автоматизировать, механизировать свою деятельность во всех сферах общественной жизни. Качество программного продукта характеризуется набором свойств, определяющих, насколько продукт "хорош" с точки зрения заинтересованных сторон, таких как заказчик продукта, спонсор, конечный пользователь, разработчики и специалисты по качеству продукта, инженеры поддержки, сотрудники отделов маркетинга, обучения и продаж. Для контроля разработки программного обеспечения зачастую руководствуются уже существующими моделями жизненных циклов проектов. В крупных компаниях управление IT- проектами осуществляется по четко отлаженной схеме, которая позволяет заказчику (customer) не усомниться, что данное ПО будет выполнено качественно и в определенные сроки.

Модели жизненного цикла программного обеспечения предназначены в первую очередь для организации методических ориентиров для разработчиков. Часто на этом уровне применения они и остаются, т.е. только иллюстрируют процесс, указывая, на какие его моменты нужно обращать внимание. Однако было бы ошибкой считать, что этим и ограничиваются возможности моделирования развития программных систем. Напротив, одна из целей моделирования заключается в такой поддержке процесса развития проекта, которая в конечном итоге приводит к повышению производительности труда и качества результатов.

Одним из главных преимуществ данной системы является то, что в ней предусматривается наряду с базовыми функциями уже существующих программных средств реализация экономической оценки каждой контрольной точки жизненного цикла информационных систем.

Новое программное средство должно удовлетворять следующим свойствам:

- Удобство интерфейса пользователя (Usability). Проектируемое программное средство должно

обладать интуитивно понятным интерфейсом для человека компетентного в этой области (менеджер, разработчик и т.д.).

- Кроссплатформенность. Проектируемая система должна обладать возможностью беспрепятственного переноса и последующей работы на любой аппаратной платформе и не зависеть от операционной системы, где будет происходить развёртывание приложения.

II. МЕТОДОЛОГИИ ЖИЗНЕННОГО ЦИКЛА ИНФОРМАЦИОННЫХ СИСТЕМ

Сегодняшний заказчик информационной системы хочет, чтобы она работала безотказно, быстро, а её стоимость и качество соответствовали действительности. Для таких критериев и будет существовать разрабатываемое программное средство.

Для организации разработки информационных систем IT – компании руководствуются современными существующими методологиями жизненного цикла информационных систем.

Наиболее популярные на сегодняшний день:

- XP (экстремальное программирование) XP стоит на четырех китах: Коммуникация, Обратная связь, Простота и Смелость. Из них следуют двенадцать практик, которым должны следовать проекты, использующие XP. Многие из этих практик представляют собой старые проверенные техники, которые, тем не менее, многие успели забыть (включая большинство предсказуемых процессов). XP не только воскрешает к жизни такие техники, но и соединяет их таким образом, что все они поддерживают и усиливают друг друга [1].
- SCRUM. Эта методология, в отличие от экстремального программирования, не задает технических правил и методик, а позволяет использовать уже существующие в компании практики кодирования. Главная идея Scrum - эмпирический подход к разработке и упор на планирование и отслеживание. Согласно методу SCRUM, проект делится на итерации (которые здесь называются "спринт"), по 30 дней каждая. Перед началом спринта вы определяете функциональность, которая требуется на данном этапе, после чего уступаете место команде разработчиков, которые выполняют поставленную вами задачу. Весь фокус в том, чтобы в течение одного спринта требования оставались неизменными.

При планировании спринта участвуют заказчики, пользователи, менеджмент, в конце

каждого спринта (они длятся около месяца) команда представляет продукт, возможно не законченный полностью, но готовый для демонстрации заказчику [2].

Разрабатываемое программное обеспечение будет опираться на методологию SCRUM.

III. ПОДДЕРЖКА КОНТРОЛЬНЫХ ТОЧЕК ЖИЗНЕННОГО ЦИКЛА ИНФОРМАЦИОННЫХ СИСТЕМ

Процесс создания новой информационной системы можно разделить на следующие четыре основных этапа:

- Этап начала.

На этом этапе составляется бизнес-план, включающий описание бизнес-среды, факторов успеха и финансовых прогнозов. Артефакты обычно включают документы, описывающие требования проекта, UML-модель сценариев использования, план проекта и оценку рисков. Очередная итерация проекта завершается, а проект переходит к следующему этапу при прохождении *контрольной точки цели жизненного цикла проекта*: завоевано доверие и получено согласие руководителей относительно области действия, затрат и планируемого графика работ; требования считаются подтвержденными актуальностью основных сценариев использования; утвержден какой-либо разработанный прототип архитектуры; утверждена величина фактических затрат, отличающаяся от планируемой.

- Этап проектирования.

На этом этапе выполняется анализ предметной области и разрабатывается архитектура проекта. Очередная итерация проекта завершается, а проект переходит к следующему этапу при прохождении *контрольной точки архитектуры жизненного цикла проекта*: определены сценарии использования и их участники и разработаны описания большинства сценариев использования; задокументировано описание архитектуры ПО и процесса разработки программной системы; задокументированы экономическое обоснование и перечень рисков; задокументирован план разработки для проекта в целом – разработаны прототипы, демонстрирующие техническую осуществимость.

- Этап построения.

На этом этапе разработка компонентов и других составляющих системы разбивается на несколько

итераций, позволяющих создать первый внешний выпуск продукта в рамках проекта, называемый *контрольной точкой функциональной готовности*.

- Этап внедрения.

На этом этапе конечный продукт передается разработчиком пользователю. Этап включает обучение пользователей и специалистов по сопровождению и бета-тестирование решения, призванное проверить, соответствует ли решение ожиданиям пользователей. Итерации повторяются, пока не будет достигнут уровень качества, указанный на этапе *Начало*. При достижении этого уровня проект проходит *контрольную точку выпуска продукта*.

Итерация длится не более месяца, а быть их может при создании информационной системы множество в зависимости от масштабов проекта и последующих желаний заказчика [3].

Если все этапы по созданию нового программного обеспечения выполнены успешно, и желания заказчика не изменились, то проект завершается и переходит к конечному этапу внедрения [4].

Для поддержки всех этапов, а также итераций и будет создаваться данное программное средство, в которое будут включены вышеуказанные контрольные точки.

IV. ЗАКЛЮЧЕНИЕ

В работе приведены методологии проектирования программных продуктов, на основе которых будет строиться программное средство по поддержке контрольных точек жизненного цикла информационных систем.

- [1] Колтунова Е. Управление разработкой ПО [электронный ресурс]. – Электронные данные.– Режим доступа: <http://www.koltunova.com/Publications/ITMethodologyClassification.pdf>
- [2] Фаулер М. Новые методологии программирования [Электронный ресурс]. – Электронные данные.– Режим доступа: <http://siliconsigma.ru/home.asp?artId=4889>
- [3] Agile Journal [Электронный ресурс] .– Электронные данные – Режим доступа: <http://agile.techwell.com/> – Дата доступа: 7.10.2012
- [4] Philippe Kruchten. What Is the Rational Unified Process? – Технический документ. – Режим доступа: <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/jan01/WhatIsTheRationalUnifiedProcessJan01.pdf> 2001. – Дата доступа : 7.10.2012