

СОВЕРШЕНСТВОВАНИЕ МЕТОДОВ СОЗДАНИЯ ВЫСОКОНАДЕЖНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ ТЕХНОЛОГИИ SCIENCE_SCRUM

Мелюшин П.В.

Белорусский национальный технический университет
Минск, Республика Беларусь

Автоматизация и создание информационных систем являются на данный момент одной из самых рентабельных областей деятельности в производстве. Одной из причин активного развития данной области является то, что автоматизация служит основой коренного изменения процессов, играющих важную роль в деятельности человека, и породила надежды на то, что с помощью новых научных дисциплин будут разрешены трудные проблемы и противоречия человеческой жизни. Ведется поиск эффективной индустрии разработки большого количества информационных систем: системы обработки данных, информационные системы управления, маркетинговые системы, системы бухгалтерского учета и тд.

В работе рассмотрены особенности управления процессом проектирования информационных систем и предложен подход по разработке методологических основ для построения информационных систем. Проанализированы существующие концептуальные модели и на их основе предложена технология Science_Scrum.

Современная техника развивается бурными темпами, и для удобства управления устройствами требуется все более совершенное информационное обеспечение. Создание различных информационных систем для персональных компьютеров за десять лет превратилось в важную и прибыльную сферу деятельности. Потребности людей растут, проекты становятся все более масштабными и бюджетными. В таких условиях в компаниях по разработке информационного обеспечения возникает необходимость в совершенствовании методологий для более эффективной отдачи.

В настоящее время существует много различных методологий разработки программных продуктов, рассчитанных на крупные и мелкие проекты, на большие и маленькие команды разработчиков. Для того чтобы выбранная методология приносила максимальную пользу, необходимо правильно оптимизировать и структурировать процесс разработки.

Существует много методологий разработки, однако проанализируем наиболее известные и развитые:

1. Экстремальное программирование (eXtreme Programming XP).
2. Crystal.
3. Методология адаптивных систем (ASD).
4. Функционально-ориентированная разработка (Feature Driven Development или FDD).

5. Метод разработки динамических систем (Dynamic System Development Method DSDM).

Важным критерием проекта является легкость внедрения методологии в процесс разработки. Для XP характерны жесткие правила выполнения проекта. Основными принципами методологии являются: простота решений, интенсивная разработка малыми группами, общение в группе и между группами, заказчик включен в процесс разработки, достаточная смелость и желание идти на риск.

Для семейства методологий Crystal характерна общность методик и абстрактные схемы. В ней не предусмотрены конкретные меры по организации работы в коллективе или планированию. По эффективности она проигрывает остальным методологиям, а по своей жесткости очень далеко отстоит от XP. Методология ASD построена на концептуальной базе теории сложных адаптивных систем. Она рассчитана на использование в экстремальных проектах, в которых преобладают быстрый темп разработок.

В технологии FDD ключевую роль играет понятие функции системы. Функция реализовываться не более чем за две недели и если сценарий использования достаточно мал, его можно считать функцией.

Для технологии DSDM характерно разработка проекта в короткие сроки с использованием ограниченного количества ресурсов, предусмотренного бюджетом. DSDM стремится сократить связующие звенья между заказчиком и разработчиком, аналитиком и дизайнером. Анализ легкости внедрения методик в процесс разработки требует поиска приемов упрощающих работу заказчика и исполнителя.

Важными критериями проектируемой системы являются надежность и жизненный цикл проекта (ориентировочно 10-30 лет).

При использовании eXtreme Programming тщательное предварительное проектирование заменяется постоянным присутствием в команде заказчика, готового ответить на любой вопрос и оценить любой прототип.

Семейство методологий Crystal рассматривает процесс создания системы как конечную целенаправленную игру и утверждает, что у этой игры есть всего две цели: главная и вспомогательная. Главная цель заключается в том, чтобы успешно закончить проект. Второстепенная цель - подготовиться к следующей игре.

Технология ASD базируется на принципе непрерывной адаптации, благодаря которой возни-

кает другой жизненный цикл проекта. Изменения в проекте становятся нормой.

Технология DSDM основана на непрерывном вовлечении пользователя в итерационный процесс разработки для которого не страшны изменения требований, но в то же время достаточно приспособлен к использованию с формальной системой управления проектом.

Для анализируемых технологий есть один существенный недостаток, из-за которого большинство заказчиков от них отказываются. Чтобы сделать процесс разработки предсказуемым, необходимо наладить дисциплину внутри коллектива, а для этого нужно точно следовать предписаниям методологии.

Альтернатива - использовать методологию разработки, которая превращает создание информационных систем в упорядоченный процесс, с помощью которого можно сделать разработку информационных систем более прогнозируемой и эффективной [1].

Автором предлагается подход на основе технологии Science_Scrum, основанный на принципе восходящей прогнозной разработки [2], когда система строится вокруг ключевых моделей, которые создаются на ранних стадиях разработки. Результаты предъявляются заказчику, который может оценить будущие и текущие тенденции. В проекте выделяется некоторый набор задач, которые должны быть реализованы для успешного его завершения. Для сложных проектов этот набор будет иметь достаточно большой вес. Декомпозиция концептуальных требований может быть глубокой и нужна система, структурирующая требования.

В начале проектирования владелец продукта и заказчик формируют концепцию информационной системы, которая показывает, для кого предназначен продукт, какие преимущества получают пользователи и какие существуют конкуренты. Результатом является схема, обладающая только частью необходимых функций. Анализ моделей разворачивается у заинтересованных лиц для предварительного анализа соответствия ожиданиям заказчика.

Помимо конечного продукта, всегда создаются вспомогательные промежуточные продукты: модели, схемы, описания. Их должно быть ровно столько, сколько необходимо для достижения конечной цели. Проблема состоит в том, что очень сложно заранее предсказать, какие промежуточные продукты нужны, а какие – нет.

Не стоит забывать и о том, что чем больше система, тем больше в ней со временем появляется ошибок, которые все сложнее и сложнее исправлять. А это означает, что большинство крупных создаваемых систем имеют долгий тестовый период. Редко какие компании закладывают в план проекта этот отрезок времени, а потому очень часто происходит срыв сроков сдачи проекта. Для достижения этой цели создается детальное описание процесса, важное место в котором занимает прогнозирование.

Предлагаемый подход оказывается эффективнее по нескольким показателям. Он приспособлен к меняющимся требованиям. Изменение требований и добавление новых свойств, определяемых заказчиком или нуждами технологии, оценивается на ранней стадии проекта. Интеграция будущих требований в начале проекта исключаются переработки и потери времени – иногда до 40% всего объема проекта [1]. Риски обнаруживаются и устраняются на ранних итерациях. Итеративный подход минимизирует риски на ранних стадиях, когда тестируются все компоненты. Дефекты можно найти и исправить за несколько итераций, что обеспечивает создание четкой архитектуры и высококачественного приложения. Узкие места обнаруживаются еще на ранних итерациях, а не в конце проекта при глобальном тестировании.

Лучшее использование персонала в проекте. Совмещается использование каскадного подхода с организацией по типу конвейера. Прогнозисты аналитики посылают будущие требования проектировщикам, которые отсылают свой продукт программистам, которые посылают компоненты специалистам по интеграции, которые отсылают систему тестировщикам. Такие переходы исключают ошибки и недопонимание разработчиков.

1. Мелюшин П.В., Хмель С.А., Казачинская Е.А. Оптимальный алгоритм проектирования устойчивых к сбоям информационных систем обработки электронных медицинских записей. НАУКА И ТЕХНИКА Мн. N 3. 2014.
2. Мелюшин П.В., Якубчик Е.О. Моделирование надежных информационных систем обработки электронных медицинских записей, Материалы 8-ой международной конференции «Приборостроение -2015».