

МЕТОДЫ РАЗРАБОТКИ ВЕБ ПРИЛОЖЕНИЙ И САЙТОВ

В статье рассматривается применение различных методологий разработки веб-приложений и сайтов. Описываются наиболее популярные методологии.

ВВЕДЕНИЕ

При разработке сайтов, приложений и прочих программных продуктов используется достаточно много отличных методологий. Какую из них выбрать, во многом зависит от конкретной ситуации — особенностей проекта, бюджета, сроков, личных предпочтений разработчика и т.д. В рамках этой статьи я хотел бы рассказать вам о пяти наиболее популярных методологиях.

I. «Водопад» или КАСКАДНАЯ МОДЕЛЬ (WATERFALL MODEL)

Классическая методология, используемая «с незапамятных времен». Представляет собой строго последовательное выполнение всех стадий разработки. Иными словами, новая стадия не начинается до тех пор, пока не будет полностью закончена предыдущая.

Каскадная модель очень удобна для управления проектом, поскольку процесс разработки легко отслеживается. Это дает возможность жесткого контроля над процессом разработки, что позволяет достаточно точно заранее определить сроки окончания и общую стоимость проекта.

Однако такая жесткость имеет и негативную сторону. «Водопад» хорошо подходит для проектов с предельно четкими требованиями и заранее продуманными способами реализации. Но если в техническом задании есть «туманные» моменты, которые можно трактовать двусмысленно, каскадная методология становится крайне неудобной. «Водопад» не предусматривает возможности откатить разработку на одну-две стадии назад, отсутствует возможность протестировать отдельный аспект до полного окончания разработки. По этой причине невозможно вносить изменения, поправки и корректировки в уже сделанную часть работы, либо внесение таких поправок резко повышает стоимость проекта.

Таким образом, каскадная методология подходит исключительно для тех проектов, где требования в техническом задании предельно точны, понятны и зафиксированы на бумаге, а какие-либо разночтения или недопонимания отсутствуют. Также очень желательно использовать данную методологию только для относительно небольших проектов.

II. V-ОБРАЗНАЯ МОДЕЛЬ (V-MODEL)

Данная модель имеет в целом те же принципы последовательной «шаг-за-шагом» разработки, что и каскадная, но отличается от нее одним принципиальным моментом — на каждом этапе осуществляется тестирование готовой части проекта.

V-образную модель обычно используют при разработке программного обеспечения, предназначенного для важных систем, где недопустимы перебои в их работе. К примеру, при создании программного обеспечения для мониторингового медицинского оборудования, различных систем безопасности и т.д. Словом везде, где ошибки и недочеты в программном продукте могут иметь серьезные последствия.

В целом можно уверенно говорить о предпочтительности V-образной модели для тех проектов, которые требуют тщательного тестирования всех аспектов от удобства интерфейса до системной стабильности и отсутствия уязвимостей для внешнего вмешательства.

III. ИНКРЕМЕНТНАЯ МОДЕЛЬ (INCREMENTAL MODEL)

Данная методология используется для проектов, предусматривающих несколько вариантов (сборок) готового продукта. Зачастую разработка ведется несколькими циклами, то есть в итоге получается своего рода «мульти-водопад». При этом в каждом цикле имеются свои этапы и создаваемые модули. Для каждого модуля предусмотрены собственные этапы уточнения требований, создания проекта, кодирования, тестирования и т.д.

Инкрементная модель предполагает особую последовательность создания сборок: сначала реализуется основной проект (базовая сборка), затем на ее основе создаются новые сборки с новыми функциями, называемыми «инкрементами».

Разработка по инкрементной модели хороша для тех проектов, в которых четки и ясны не только базовые требования к системе, но и запросы на внесение изменений тоже ясны, а сами изменения легко реализуются. При этом вполне допускается, что отдельные функции и новые сборки могут дорабатываться уже после внедрения на практике базовой сборки.

IV. БЫСТРАЯ РАЗРАБОТКА ПРИЛОЖЕНИЙ ИЛИ «RAD MODEL»

Представляет собой разновидность описанной выше инкрементной модели. Ключевым отличием является то, что компоненты проекта (модули) или разные сборки разрабатываются не поочередно одной командой, а параллельно несколькими командами. В условиях жестко лимитированного времени созданные одновременно модули собирают в единый рабочий прототип. В итоге удается предоставить заказчику рабочую систему в предельно сжатые сроки.

Важным условием применения данной методологии является наличие нескольких высококвалифицированных команд специалистов. Следствием такого подхода становятся высокие расходы на оплату услуг большого количества задействованных спецов и рабочих инструментов, которые они используют.

V. ГИБКАЯ МОДЕЛЬ РАЗРАБОТКИ (AGILE MODEL)

Ключевая особенность данной методологии заключается в максимальной прозрачности процесса разработки для заказчика, у которого есть возможность отслеживать буквально каждую итерацию и одобрять ее либо требовать переделки. Таким образом, полностью исключается малейшая вероятность сделать совсем не то, чего хотел клиент.

Дроздович Денис Андреевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, denisdrozdvovich@gmail.com.

Научный руководитель: Ломако Александр Викторович, Доцент кафедры ИТАС, кандидат технических наук, доцент, lavlot@bsuir.by

Очевидным недостатком гибкой модели является сложность предварительной оценки трудозатрат и стоимости проекта. Однако в условиях отсутствия четких требований и невнятного ТЗ, когда заказчик сам весьма смутно понимает, что ему нужно, гибкая модель является единственно возможной для использования.

Важным атрибутом гибкой методологии является проведение непродолжительных ежедневных встреч, которые именуются «Scrum», а также регулярных собраний раз в неделю или реже, именуемых «Sprint».

Методология хорошо себя показывает при разработке больших проектов, либо проектов, которые нужно постоянно адаптировать к меняющимся условиям рынка.

VI. ВЫВОДЫ

В современной практике методы разработки веб приложений и сайтов многовариантны. Нет единственно верной для всех проектов, стартовых условий и моделей оплаты. Даже столь любимая всеми нами Agile не может применяться повсеместно из-за неготовности некоторых заказчиков или невозможности гибкого финансирования. Методологии частично пересекаются в средствах и отчасти похожи друг на друга. Некоторые другие концепции использовались лишь для пропаганды собственных компиляторов и не привносили в практику ничего нового.