

Жизненный цикл программного обеспечения

JSP & Servlets
15 уровень, 0 лекция

ОТКРЫТА

Этапы жизненного цикла программного продукта

Разработка качественного программного обеспечения требует наличия целого ряда факторов: квалифицированной команды, планирования рабочих процессов, соответствия продукта ожиданиям заказчика, соблюдение дедлайнов.

1. Анализ требований

Этот этап можно считать одним из наиболее важных. Именно от него зависит успех проекта. Все начинается с того, что формируется цель проекта. Затем ставится перечень задач, которые необходимо выполнить, и сфера применения будущего ПО. После этого проясняются условия, срок выполнения и бюджет проекта. На финальной стадии первого этапа утверждается техническая задача команде разработчиков.

2. Этап проектирования

Проектирование начинается с определения архитектуры приложения, его функций, требований к функционалу и интерфейсу. Затем распределяются функции между программой и пользователями, учитываются требования к различным компонентам. Проектирование продукта должно учесть ожидания заказчика и возможность их практической реализации.

Далее начинается разработка спецификации ПО, выбирается наиболее оптимальная архитектура системы, СУБД, вариант хранения данных, определяется совместимость с оборудованием, процесс внедрения ПО и список документов по эксплуатации.

3. Написание кода

На этом этапе создается прототип готового продукта и его компонентов, выполняется разработка структуры данных и программных кодов. Затем проводится тестирование и пишется техническая документация. Итогом данного этапа становится появление жизнеспособной версии продукта, доступной для тестирования и отладки.

4. Тестирование и отладка

Этот процесс неотъемлемо связан с проектированием и эксплуатацией. В нем применяются механизмы, позволяющие выполнять тестирование софта на соответствие требований к нему. В этот этап также входит проверка сопроводительной документации.

Успешным итогом тестирования следует считать устранение всех обнаруженных ошибок в приложении и создание отчета о его качестве.

5. Эксплуатация и сопровождение

Переход к эксплуатации софта предполагает его установку, обучение пользователей и документирование. Поддержка работы ПО осуществляется командой техподдержки разработчика.

Сопровождение включает в себя адаптацию приложения к новым требованиям работы, добавление изменений в код и сопроводительную документацию, если это требуется по причине обнаружения багов или исходя из требований в модификации. Внесение изменений в ПО предполагает сохранение его основного функционала.

Вывод из эксплуатации софта может происходить по причине его морального старения, появления на рынке более современных продуктов или по иным соображениям.

Технические проблемы, которые встречаются при разработке ПО

Для чего необходимы модели жизненного цикла? Разве не лучше просто создать надежный продукт с безотказной работой? Оказывается, разработка модели жизненного цикла ПО помогает решить сразу четыре проблемы разработки:

Неверное понимание того, что нужно пользователям. У разработчика продукта может быть ошибочное представление о состоянии рынка и потребностей в его продукте среди пользователей.

Отладка. При обнаружении бага в программе наличие модели жизненного цикла ПО помогает быстрее определить, в чем состоит проблема и наиболее вероятные способы ее устранения.

Слишком быстрое изменение рыночной конъюнктуры. Продукт, который актуален сейчас, через полгода уже может устареть. На рынке может появиться приложение с более широким функционалом или более удобным интерфейсом. Цикл разработки ПО помогает следить за тенденциями, что облегчает понимание того, как нужно улучшить приложение, если пользовательские предпочтения изменились.

Безопасность. Это одна из основных проблем разработки. Как правило, ошибку трудно найти до начала эксплуатации продукта. Обнаружение ошибки после релиза ПО приводит к ощутимым финансовым убыткам для бизнеса. Благодаря жизненному циклу разработки ускоряется поиск и устранение лазеек в безопасности. Хотя это не исключает, но заметно снижает угрозы.

В результате на начальных этапах разработка становится эффективнее, снижается ее себестоимость и ускоряется релиз продукта.

Подходы к разработке ПО

Существует несколько подходов к разработке программных продуктов. Довольно часто их адаптация происходит исходя из текущей ситуации: требований к соблюдению дедлайнов, надежности, безопасности, стоимости работ, квалификации участников команды. Среди наиболее известных подходов стоит выделить:

- Code and fix — написание кода и устранение багов в нем;
- Waterfall Model — каскад или “водопад”;
- V-model — разработка через тестирование;
- Incremental Model — инкрементная модель;
- Iterative Model — итеративная модель;
- Spiral Model — спиральная модель;
- Agile Model — гибкая методология разработки.

Модель code and fix самая простая. Разработчик пишет программный код, запускает его. Затем смотрит, как он работает. При обнаружении бага устраняет его. Скорее всего, эту модель вы уже освоили, поэтому давайте перейдем к остальным.

< Предыдущая лекция

Следующая лекция >

− +15 +

Комментарии

популярные

новые

старые

JavaCoder

Введите текст комментария



ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

