СОДЕРЖАНИЕ

Введение…………………………………………………………………………….3

История возникновения методологии…………………………………………….4

Содержание методологии………………………………………………………….5

Известные применения…………………………………………………………….7

Выводы……………………………………………………………………………...8

Список использованной литературы……………………………………………...9

ВВЕДЕНИЕ

До появления комплексного подхода к разработке программного обеспечения пользовались такой методологией, как “code-and-fix” (кодирование и правка), однако в 30-е года XIX века в трудах Шеварда упоминается итеративная модель разработки.

“code-and-fix” – это та методология, которая требует только написания кода, не требует моделирования систем и тщательного документирования готового продукта. У этой методологии есть один весомый недостаток: конечный продукт невозможно поддерживать. Внесение правок в систему могло привести к неисправности системы и нарушению работы модулей.

Каскадная модель является первой попыткой решить проблемы прошлого и обратить внимание на то, что создание программных продуктов состоит не из обычного “кодинга”, а из целостного цикла от анализа предметной области до вывода программы из эксплуатации.

2.ИСТОРИЯ ВОЗНИКНОВЕНИЯ МЕТОДОЛОГИИ

Впервые каскадную модель описал Уинстон Ройс в своей статье 1970 года “MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS”(“Управление развитием больших программных систем”).

Успехи науки в области микроэлектроники удешевили стоимость аппаратуры, а стоимость программного обеспечения все еще была равна стоимости “железа”. В 70-80 года XX века к программированию относились как к кодированию. Но уже в то время программное обеспечение начинают рассматривать как имеющее полный жизненный цикл, начинающийся с появления концепции и проходящий стадии проектирования, разработки, ввода в действие, сопровождения и развития. Смена вектора внимания с простого кодирования позволило разрабатывать различные методологии разработки.

Предшественников водопадной методологии была такая методология, как “code ans fix”.

Конкурентом водопадной методологии является итеративная модель разработки. Истоки концепции итеративной разработки прослеживаются в относящихся к 30-м годам работах эксперта по проблемам качества продукции Уолтера Шеварта из Bell Labs.

3.СОДЕРЖАНИЕ МЕТОДОЛОГИИ

Этапы каскадной модели разработки включают в себя этапы: определение требований, проектирование, реализация (кодирование), интеграция компонентов, тестирование и отладка, ввод в эксплуатацию и аттестация, сопровождение и эксплуатация, вывод системы из эксплуатации. Смыслом такой модели разработки является то, что переход от одной фазы к другой происходит только после полного и успешного завершения предыдущей.

3.1 Определение требований. На основе полученной предметной области формируются и формализуются требования к системе, как к единому целому. Результатом этого этапа является определение общих функциональных требований к программному обеспечению, ограничений, главной цели системы.

3.2 Проектирование. Определение системных компонентов и связей между ними на основе системных требований. Так же включает в себя разбиение требований на группы, определение подсистем, Распределение требований по подсистемам, определение функциональных характеристик каждой подсистемы, определение интерфейса подсистем.

3.3 Кодирование. Перевод в программный код проектных решений.

3.4 Интеграция компонентов. Сборка подсистем и модулей в единую функционирующую систему.

3.5 Тестирование и отладка. Проверка системы на соответствие требованиям заказчика и функциональным требованиям

3.6 Ввод в эксплуатацию и аттестация. На этом этапе решаются проблемы в несовместимости программного обеспечения исполнителя и заказчика, проблемы совместного использования программного обеспечения. Так же решается задача обучения персонала использовать внедряемый программный продукт.

3.7 Сопровождение программного продукта и его эксплуатация. Процесс улучшения, оптимизации и устранения дефектов программного обеспечения после передачи в эксплуатацию. Сопровождение. В ходе сопровождения в программу вносятся изменения, с тем, чтобы исправить обнаруженные в процессе использования дефекты и недоработки, а также для добавления новой функциональности, с целью повысить удобство использования и применимость ПО.

3.8 Вывод системы из эксплуатации. Во время этого этапа решается проблема вывода системы из использования на предприятии. Проводится в соответствии с законодательством страны, в котором используется. Основные задачи: избежать потери данных и решить проблему их переиспользования.

Преимущества каскадной модели :

- стабильность требований в течение всего жизненного цикла разработки.

- возможность последовательного устранения возникающих сложностей.

- определенность и понятность шагов модели и простота ее применения.

- упрощение возможности осуществления планирования, контроля и управления проектом.

- доступность для понимания заказчиками.

- эффективность для проектов с четкими и понятными, но трудно реализуемыми требованиями.

- эффективность для проектов с высокими требованиями к качеству при отсутствии жестких ограничений затрат и графика работ.

Недостатки Каскадной модели

- сложность четкого формулирования требований в начале жизненного цикла и невозможность их динамического изменения на его протяжении;

- последовательность линейной структуры процесса разработки, в результате возврат к предыдущим шагам для решения возникающих проблем приводит к увеличению затрат и нарушению графика работ;

- непригодность промежуточного продукта для использования;

- невозможность гибкого моделирования систем, не имеющих аналогов;

- позднее обнаружение проблем, связанных со сборкой, в связи с одновременной интеграцией всех результатов в конце разработки;

- недостаточное участие пользователя в создании системы – только в самом начале (при разработке требований) и в конце (во время приемочных испытаний);

- невозможность предварительной оценки качества системы пользователем;

- проблемность финансирования проекта, связанная со сложностью единовременного распределения больших денежных средств.

4.ИЗВЕСТНЫЕ ПРИМЕНЕНИЯ:

Таблица 1 – Известные применения каскадной методологии в разработке программных продуктов

|  |  |
| --- | --- |
| Wüstenrot & Württembergische (W&W) | Разработка ERP-системы для финансового сектора |
| Cisco | Разработка систем безопасности |
| Microsoft IT, AT Consulting,  Parallels, SAP, Toyota. | Разнообразные продукты и решения или их части |
|  |  |

ВЫВОДЫ

На ранних стадиях проекта можно достаточно точно прогнозировать и время и материальные затраты на его реализацию. По окончании проекта заказчик получает тот продукт, который был изначально запланирован.

Только этапы проектирования требуют наличия высококвалифицированных специалистов, реализация может осуществляться менее опытными программистами.

Такие проекты хорошо масштабируются. После завершения проектирования работа по созданию разных модулей может проходить независимо друг от друга.

Несмотря на огромный успех в те времена, у водопадной модели были серьезные ограничения и недостатки:

Заказчик получает готовый продукт только в самом конце разработки. В то время как проект с применением данной методологии может длиться от одного года до двух

В большинстве проектов возникают проблемы с формулированием требований на раннем этапе разработки.

Эти недостатки привели к провалу огромного количества проектов и возникла потребность разрабатывать новые методологии разработки программных продуктов.

Несмотря на существенные недостатки, водопадная модель испрользуется в современной разработке. Только водопад применяют для создания программ, к качеству которых предъявляются высокие требования: системы управления критическими производствами, высококачественные системы информационной безопасности.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ:

[Royce, W.E. Managing the Development of Large Software Systems](https://web.archive.org/web/20160315200930/http:/www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf)//Wayback machine. – 1970. – C. 1-13.

Shewhart, W. A. Economic control of quality of manufactured product/ , W. A. Shewhart, 2010 – 151.

Shewhart, W.A.Statistical method from the viewpoint of quality control/ W.A. Shewhart, 2011 – 176c

McConnell, S.A. Rapid Development: Taming Wild Software Schedules// Microsoft Press. – 1996. – C.3-7