

Программирование и основы алгоритмизации

Лекция 12

Технологии программирования

Понятие технологии

Техноло́гия (от греч. *téchne* — искусство, мастерство, умение и греч. *логия* — изучение) — совокупность методов и инструментов для достижения желаемого результата; способ преобразования данного в необходимое. Целью технологии является производство продукта: материальная технология создаёт материальный продукт, информационная технология (ИТ) — информационный продукт. Технология это также научная дисциплина, разрабатывающая и совершенствующая способы и инструменты производства [1].

Хронология развития технологии разработки программ

- Конец 1960-х начало 1970-х - Зарождение концепции структурного программирования
 - *для записи любой программы в принципе достаточно только трех конструкций управления – последовательного выполнения, ветвления и цикла и ... о необходимости использовании оператора перехода (GOTO)*
- Следующий шаг – введение в структурное программирование аппарата функций и переход к процессу нисходящего программирования
 - *При этом подходе программа пишется в терминах вызова функций верхнего уровня, которые реализуются при помощи функций более низкого уровня*
- *Возникновение парадигмы объектно-ориентированного программирования, позволяющей в структуре программ отображать сущности предметной области.*
 - *При этом подходе облегчается модификация программы за счет ..., что ... в условиях изменяющихся требований*
- *Появление современных систем и языков программирования, основанных на ООП – Java и CORBA*
 - *CORBA позволяет приложениям использовать функции, написанные на различных языках, и выполнять их на различных платформах.*

Жизненный цикл программного обеспечения

Период от момента появления идеи создания некоторого программного обеспечения до момента завершения его поддержки фирмой-разработчиком или фирмой, выполнявшей сопровождение

Состав процессов жизненного цикла регламентируется международным стандартом ISO/IEC 12207: 1995 «Information Technologies – Software Life Cycle Processes» («Информационные технологии - Процессы жизненного цикла программного обеспечения»).

ISO - International Organization for Standardization – Международная организация по стандартизации.
IEC -International Electrotechnical Commission – Международная комиссия по электротехнике.

Модель жизненного цикла (ЖЦ) ПО

- Модель ЖЦ ПО включает в себя:
 - Стадии;
 - Результаты выполнения работ на каждой стадии;
 - Ключевые события — точки завершения работ и принятия решений.
- Стадия — часть процесса создания ПО, ограниченная определенными временными рамками и заканчивающаяся выпуском конкретного продукта (моделей, программных компонентов, документации), определяемого заданными для данной стадии требованиями.
- На каждой стадии могут выполняться несколько процессов, определенных в стандарте ГОСТ Р ИСО/МЭК 12207-99, и наоборот, один и тот же процесс может выполняться на различных стадиях. Соотношение между процессами и стадиями также определяется используемой моделью жизненного цикла ПО.

Процессы жизненного цикла ПО

- Основные:
 - Приобретение (действия и задачи заказчика, приобретающего ПО)
 - Поставка (действия и задачи поставщика, который снабжает заказчика программным продуктом или услугой)
 - Разработка (действия и задачи, выполняемые разработчиком: создание ПО, оформление проектной и эксплуатационной документации, подготовка тестовых и учебных материалов и т. д.)
 - Эксплуатация (действия и задачи оператора — организации, эксплуатирующей систему)
 - Сопровождение (действия и задачи, выполняемые сопровождающей организацией, то есть службой сопровождения). Сопровождение — внесений изменений в ПО в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы или требованиям.

Процессы жизненного цикла ПО (2)

- Вспомогательные
 - Документирование (формализованное описание информации, созданной в течение ЖЦ ПО)
 - Управление конфигурацией (применение административных и технических процедур на всем протяжении ЖЦ ПО для определения состояния компонентов ПО, управления его модификациями).
 - Обеспечение качества (обеспечение гарантий того, что ИС и процессы ее ЖЦ соответствуют заданным требованиям и утвержденным планам)
 - Верификация (определение того, что программные продукты, являющиеся результатами некоторого действия, полностью удовлетворяют требованиям или условиям, обусловленным предшествующими действиями)
 - Аттестация (определение полноты соответствия заданных требований и созданной системы их конкретному функциональному назначению)
 - Совместная оценка (оценка состояния работ по проекту: контроль планирования и управления ресурсами, персоналом, аппаратурой, инструментальными средствами)
 - Аудит (определение соответствия требованиям, планам и условиям договора)
 - Разрешение проблем (анализ и решение проблем, независимо от их происхождения или источника, которые обнаружены в ходе разработки, эксплуатации, сопровождения или других процессов)

Процессы жизненного цикла ПО (3)

- Организационные
 - Управление (действия и задачи, которые могут выполняться любой стороной, управляющей своими процессами)
 - Создание инфраструктуры (выбор и сопровождение технологии, стандартов и инструментальных средств, выбор и установка аппаратных и программных средств, используемых для разработки, эксплуатации или сопровождения ПО)
 - Усовершенствование (оценка, измерение, контроль и усовершенствование процессов ЖЦ)
 - Обучение (первоначальное обучение и последующее постоянное повышение квалификации персонала)

Типичная схема разработки программного обеспечения

- Понять природу и сферу применения предлагаемого продукта
- Выбрать процесс разработки и создать план
- Собрать требования
- Спроектировать и собрать продукт
- Выполнить тестирование продукта
- Выпустить продукт и обеспечить его сопровождение

Стадии разработки программного обеспечения

СТАДИЯ	ДОКУМЕНТАЦИЯ
Постановка задачи	Техническое задание ГОСТ 19.201-78
Анализ требований <i>формулируют содержательную постановку задачи, выбирают математический аппарат формализации, строят модель предметной области, определяют подзадачи и выбирают или разрабатывают методы их решения</i>	Спецификация <i>точное и формализованное описание функций</i> Стандарт IEEE 830-1993
Проектирование	Детальный проект <i>проектирование программ (подпрограмм) и определение взаимосвязей между ними, так и проектирование данных, с которыми взаимодействуют эти программы или подпрограммы</i>
Реализация проекта, отладка и тестирование	Программа и методика испытаний (ПМИ); эксплуатационная документация

Сопровождение

Сопровождение – процесс создания и внедрения новых версий программного продукта.

Причины выпуска новых версий:

- необходимость исправления ошибок, выявленных в процессе эксплуатации предыдущих версий;
- необходимость совершенствования предыдущих версий
- изменение среды функционирования, например, появление новых технических средств и/или программных продуктов, с которыми взаимодействует сопровождаемое программное обеспечение.

РАЗНОВИДНОСТИ ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММ



- Водопадная (каскадная, последовательная) модель – классическая модель процесса разработки программ
 - *процесс представляется последовательностью фаз анализа требований, проектирования, реализации, интеграции и тестирования*
- Спиральная модель – модификация классической (водопадной) модели
 - *фазы водопадной модели повторяются неоднократно. На каждой итерации строится очередная версия программы на основе ее предыдущей версии*
- Инкрементальная модель – эволюционная модель разработки
 - *разработка проходит в непрерывном процессе обновления компонентов программы. Такая модель особенно полезна на этапах сопровождения с целью «безударного» перехода к новым версиям программы.*

Необходимость возвратов на предыдущие стадии

Данная модель позволяет:

- сократить время до появления первых версий программного продукта;
- заинтересовать большое количество пользователей, обеспечивая быстрое продвижение следующих версий продукта на рынке;
- ускорить формирование и уточнение спецификаций за счет появления практики использования продукта;
- уменьшить вероятность морального устаревания системы за время разработки.

Сравнение процессов разработки

Фактор	водопадный процесс	Итеративные процессы	
		Спиральный	Инкрементальный
Легкость контроля документации	Легче	Тяжелее	Тяжелее/Средне (пояснение 1)
Возможность взаимодействия с заказчиком	Тяжелее	Легче	Легче
Поддержание хорошего проектирования	Средне/легче	Легче (пояснение 2)	Тяжелее
Сбор метрических данных, собранных в ходе проекта	Тяжелее	Средне/легче	Средне/легче

Пояснение 1.

Пояснение 2.

Быстрая разработка приложений. Технология RAD

Rapid Application Development – Быстрая разработка приложений) - технология, ориентированная на максимизацию эффективности разработки, что достигается за счет:

- проведения разработки небольшими группами разработчиков (3-7 человек), каждая из которых проектирует и реализует отдельные подсистемы проекта – позволяет улучшить управляемость проекта;
- использования итерационного подхода;
- наличия четко проработанного графика цикла, рассчитанного не более чем на три месяца.

Технология RAD (1)

- На этапе *анализа и планирования требований* формулируют наиболее приоритетные требования, что ограничивает масштаб проекта
- На этапе *проектирования*, используя имеющиеся CASE-средства, детально описывают процессы системы
- На этапе *реализации* выполняют итеративное построение реальной системы, при этом для контроля над выполнением требований к создаваемой системе привлекаются будущие пользователи

Технология RAD (2)

- эта технология не применима для построения сложных расчетных программ, операционных систем или программ управления сложными объектами в реальном масштабе времени, т.е. программ с большим процентом уникального кода.
- Не годится она и в случае создания приложений, от которых зависит безопасность людей так как технология предполагает, что первые несколько версий не будут полностью работоспособны

Экстремальное программирование

- Разновидность RAD программирования
- В основе методологии командная работа (обычно всего несколько разработчиков, от 2-х), эффективная коммуникация между заказчиком и исполнителем в течение всего проекта по разработке ИС.
- Разработка ведется с использованием последовательно дорабатываемых прототипов.

Разработка технического задания (1)

основные факторы, определяющие характеристики разрабатываемого программного обеспечения:

- исходные данные и требуемые результаты, которые определяют *функции* программы или системы;
- среда функционирования (программная и аппаратная) – может быть задана, а может выбираться для обеспечения параметров, указанных в техническом задании;
- возможное взаимодействие с другим программным обеспечением и/или специальными техническими средствами – также может быть определено, а может выбираться исходя из набора выполняемых функций.

Разработка технического задания (2)

Последовательность действий:

- устанавливают набор выполняемых функций
- устанавливают перечень и характеристики исходных данных
- определяют перечень результатов, их характеристики и способы представления
- уточняют среду функционирования программного обеспечения: конкретную комплектацию и параметры технических средств, версию используемой операционной системы и, возможно, версии и параметры другого установленного программного обеспечения, с которым предстоит взаимодействовать будущему программному продукту
- необходимо также четко регламентировать действия программы в случае сбоев оборудования и энергоснабжения

Гос. стандарт на разработку ТЗ

стандарт ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению». В соответствии с этим стандартом техническое задание должно содержать следующие разделы:

- введение;
- основания для разработки;
- назначение разработки;
- требования к программе или программному изделию;
- требования к программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки.

Анализ требований и разработка спецификации

Два подхода к разработке ПО:

- Структурный подход
 - Функциональные диаграммы
 - Диаграммы потоков данных
 - Диаграммы переходов состояний
- Объектно-ориентированный подход
 - UML диаграммы

Unified Modeling Language —
унифицированный язык моделирования)
- язык графического описания.

UML диаграммы (1)

UML содержит стандартный набор диаграмм и нотаций

- **диаграммы вариантов использования (use case diagrams)** – для моделирования бизнес-процессов организации и требований к создаваемой системе);
- **диаграммы классов (class diagrams)** – для моделирования статической структуры классов системы и связей между ними;
- **диаграммы поведения системы (behavior diagrams):**
 - 1) **диаграммы взаимодействия (interaction diagrams):**
 - **диаграммы последовательности (sequence diagrams)** и
 - **кооперативные диаграммы (collaboration diagrams)** – для моделирования процесса обмена сообщениями между объектами;

UML диаграммы (2)

- 2) **диаграммы состояний (statechart diagrams)** – для моделирования поведения объектов системы при переходе из одного состояния в другое;
- 3) **диаграммы деятельности (activity diagrams)** – для моделирования поведения системы в рамках различных вариантов использования, или моделирования деятельности;
- **диаграммы реализации (implementation diagrams):**
 - **диаграммы компонентов (component diagrams)** – для моделирования иерархии компонентов (подсистем) системы;
 - **диаграммы размещения (deployment diagrams)** – для моделирования физической архитектуры системы.

Литература

- Г.Майерс. Надежность программного обеспечения. - М.: Мир, 1980.
- Э. Дейкстра. Заметки по структурному программированию / У. Дал, Э. Дейкстра, К. Хоор. Структурное программирование. - М.: Мир, 1975. - С. 7-97.
- В. Турский. Методология программирования. - М.: Мир, 1981.
- Г. Буч. Объектно-ориентированное проектирование с примерами применения. - М.: Конкорд, 1992.
- Ф.П. Брукс, мл. Как проектируются и создаются программные комплексы. - М.: Наука, 1979.
- Дж. Хьюз, Дж. Мичтом. Структурный подход к программированию. - М.: Мир, 1980.
- Б. Боэм, Дж. Браун, Х. Каспар и др. Характеристики качества программного обеспечения. - М.: Мир, 1981.
- В.В. Липаев. Качество программного обеспечения. - М.: Финансы и статистика, 1983.
- Требования и спецификации в разработке программ. - М.: Мир, 1984.

Литература (2)

- С.А.Орлов. Технологии разработки программного обеспечения. - С.-П.:Питер, 2002.
- Дж.Арлоу, А.Нейштадт. UML 2 и унифицированный процесс. – С.-П.:Символ, 2008.
- В.В.Мухортов, В.Ю.Рылов. Объектно-ориентированное программирование, анализ и дизайн. – Новосибирск: Новософт, 2002.