

- **Летняя учебная практика с 30.05 по 25.06**

- ПМ04 "Участие в ревьюировании программных продуктов"(2 недели)
- ПМ05 "Сопровождение программного обеспечения компьютерных систем"(1 неделя)
- ПМ10 "Восстановление требований по программному коду в процессе реинжиниринга"(1 неделя)

- ПМ04 "Участие в ревьюировании программных продуктов"(2 недели)

Задание:

- Написать техническую спецификацию
- Осуществить ревьюирование кода и технической документации (произвести тестирование и оптимизацию созданного программного кода)
- Выполнить процесс измерения характеристик компонент программного продукта.
- **Deadline: 4.06**
- **Hard deadline: 11.06.**

- **Варианты (номер варианта по журналу):**
- Разработка программного обеспечения для автоматизации аренды коньков, роликов, велосипедов, лыж
- Разработка программного обеспечения для автоматизации работы приемной комиссии ВУЗа.
- Разработка программного обеспечения для автоматизации служб такси
- Разработка программного обеспечения для автоматизации лифта
- Разработка программного обеспечения для автоматизации работы парикмахерской
- Разработка программного обеспечения для автоматизации автопроката
- Разработка программного обеспечения для автоматизации работы косметической компании
- Разработка программного обеспечения для автоматизации регистрации больных в поликлинике
- Разработка программного обеспечения «ARM сотрудника отдела кадров»
- Разработка приложения «Клуб обучения танцам»
- Разработка программного обеспечения «Система управления проектом для IT-компании»
- Разработка программного обеспечения для автоматизации работы парикмахерской
- Разработка программного обеспечения для автоматизации транспортной логистики
- Разработка программного обеспечения «Электронная библиотека»
- Разработка программного обеспечения для автоматизации поставки товара на склад
- Разработка программного обеспечения для автоматизации работы

Участие в ревьюировании программных продуктов

- Читать проектную документацию, разработанную с использованием графических языков спецификаций.
- Методы и технологии ревьюирования кода и проектной документации

- Эффективного тестирования классов можно достичь при помощи **ревью** и **тестовых прогонов**.
- **Ревью** представляет собой просмотр исходного кода *ПО* с целью обнаружения ошибок и дефектов, возможно, до того, как это *ПО* заработает.
- **Ревьюирование** предназначено для выявления таких ошибок, как неспособность выполнять то или иное **требование спецификации** или ее неправильное понимание, а также алгоритмических ошибок в реализации.

Кто, что, когда, как и в каком объеме?

- **Кто выполняет тестирование?**

Обычно тестирование классов выполняют их разработчики. В этом случае время на изучение спецификации и реализации сводится к минимуму. Недостатком подхода является то, что если разработчик неправильно понял спецификации, то он для своей неправильной реализации разработает и "ошибочные" тестовые наборы.

- **Что тестировать?**

Необходимо удостовериться, что программный код класса в точности отвечает требованиям, сформулированным в его спецификации, и что он не делает ничего более.

- **Как будет выполняться тестирование?**

Тестирование классов обычно выполняется путем разработки тестового драйвера, который создает экземпляры классов и окружает эти экземпляры соответствующей средой (тестовым окружением), чтобы стал возможен прогон соответствующего тестового случая.

Драйвер посылает сообщения экземпляру класса в соответствии со спецификацией тестового случая, а затем проверяет *исход* этих сообщений. Тестовый *драйвер* должен удалять созданные им экземпляры тестируемого класса. *Статические элементы* данных класса также необходимо тестировать.

Что тестировать?

- Можно выделить два типа классов с точки зрения их взаимодействия с другими классами:

*примитивные классы

Примитивный класс может порождать экземпляры, и эти экземпляры можно использовать без необходимости создания экземпляров каких-либо других классов, в том числе и данного класса. Такие объекты представляют собой простейшие компоненты системы и, несомненно, играют важную роль при выполнении любой программы.

*непримитивные классы.

Общее качество программной системы включает в себя на верхнем уровне ряд составляющих, которые должны быть приняты во внимание при управлении качеством.



Шкала измерения характеристик(ISO 12207) – введение в метрики

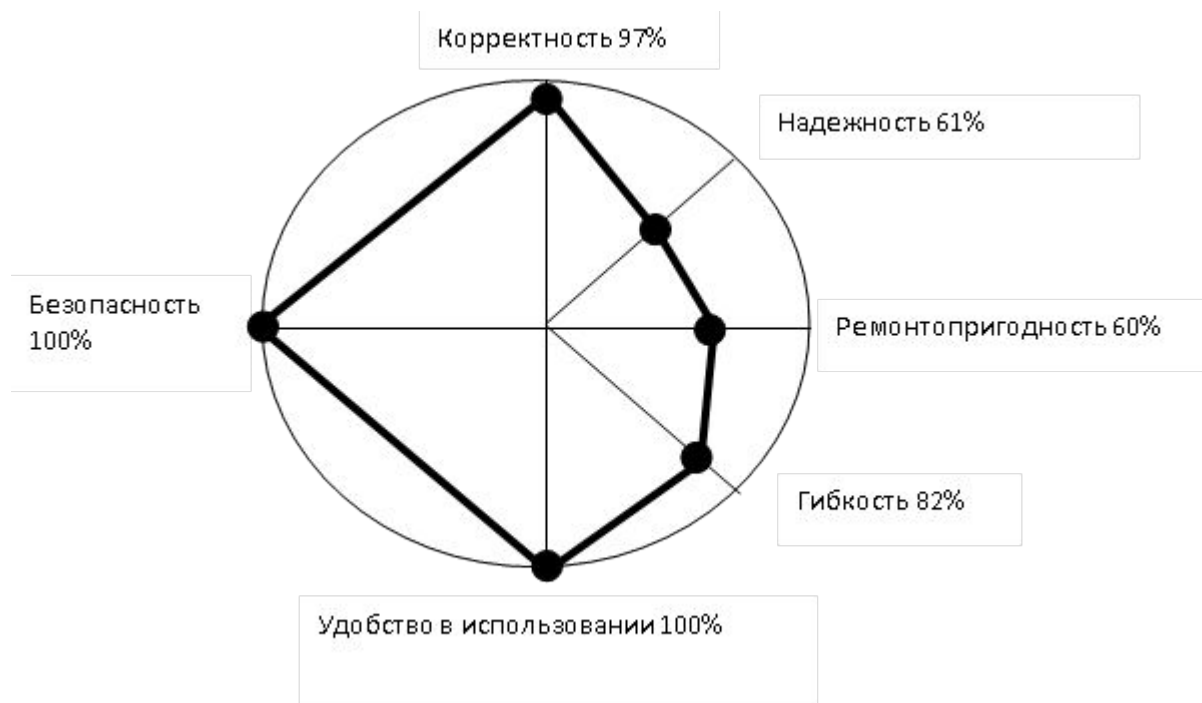
- Для каждой характеристики качества рекомендуется формировать меры и шкалу измерений с выделением требуемых, допустимых и неудовлетворительных значений. Реализация процессов оценки должна коррелировать с этапами жизненного цикла конкретного проекта программного средства в соответствии с применяемой, адаптированной версией стандарта ISO 12207.

- **Функциональная пригодность**
- **Оценка корректности программных средств**
- **Оценка способности к взаимодействию**
- **Оценка защищенности программных средств**
- **Оценка надежности**
- **Потребность в ресурсах памяти и производительности**
- **Оценка практичности**
- **Сопровождаемость**
- **Оценка мобильности**

Пример графического изображения качества

- Для мониторинга метрик качества и подготовки информации для принятия решений собранные метрики должны представляться в наглядном виде, обеспечивающим полноту информации, что особенно важно при отсутствии консолидированных метрик качества *"Пример графического изображения качества"*.

Для конкретного проекта должно быть разработано или дополнено свое множество метрик, которое отражает назначение и особенности окружения разрабатываемого программного продукта.



- ПМ05 "Сопровождение программного обеспечения компьютерных систем"(1 неделя)

Задание:

- Осуществить выбор методов и средств измерения эксплуатационных характеристик объектов профессиональной деятельности.
- Выполнить работы по модификации отдельных компонент программного обеспечения.
- **Deadline: 16.06**
- **Hard deadline: 18.06.**

Сопровождение программного обеспечения компьютерных систем

- **Сопровождéние (поддержка) программного обеспечения** — процесс улучшения, оптимизации и устранения дефектов программного обеспечения (ПО) после передачи в эксплуатацию.
- **Сопровождение ПО** — это одна из фаз жизненного цикла программного обеспечения, следующая за фазой передачи ПО в эксплуатацию.

- Сопровождение программного обеспечения стандартизовано, имеются национальные стандарты Российской Федерации, идентичные международным (ISO/IEC 12207:2008 System and software engineering — Software life cycle processes, ГОСТ Р ИСО/МЭК 12207-2010 «Национальный стандарт Российской Федерации. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»; ISO/IEC 14764:99 Information technology — Software maintenance, ГОСТ Р ИСО/МЭК 14764-2002 «Государственный стандарт Российской Федерации. Информационная технология. Сопровождение программных средств»; IEEE 1219).

В процессе сопровождения в программное обеспечение вносятся следующие изменения, значительно различающиеся причинами и характеристиками

- **исправление ошибок**
- **модернизация** - расширение функциональных возможностей или улучшение характеристик решения отдельных задач в соответствии с новым или дополнительным техническим заданием на программное изделие.
- регламентированная документами **адаптация программного** обеспечения к условиям конкретного использования, с учетом характеристик внешней среды или конфигурации аппаратуры, на которой предстоит функционировать программам. Адаптация занимает около 20% общих затрат на сопровождение.

В общем случае процесс сопровождения состоит из следующих задач:

- устранение сбоев;
- улучшение дизайна;
- расширение функциональных возможностей;
- создание интерфейсов взаимодействия с другими (внешними) системами;
- адаптация (например, портирование) для возможности работы на другой (или обновленной) аппаратной платформе, применение новых системных возможностей, функционирование в среде обновленной телекоммуникационной инфраструктуры и т.п.;
- вывод программного обеспечения из эксплуатации.

Этапы процесса сопровождения



- Этапы процесса сопровождения основаны на цикле Деминга PDCA (Plan — Do — Check — Analyze) или «планируй — делай — проверяй — анализируй»

Формирование процесса сопровождения начинается с разработки концепции сопровождения. Такой документ, например, по стандарту ISO/IEC 14764 (Standard for Software Engineering— Software Maintenance), должен содержать следующие разделы:

1. Область сопровождения программного средства.

1.1. Типы выполняемого сопровождения.

1.2. Сопровождаемый уровень документов.

Анализ проблем и изменений

Проверка и приемка при сопровождении

Внесение изменений

Перенос

Снятие с эксплуатации

Подготовка

1.3. Реакция (чувствительность) на сопровождение (определение ожиданий к сопровождению заказчика).

1.4. Обеспечиваемый уровень обучения персонала.

1.5. Обеспечение поставки продукта.

1.6. Организация справочной службы («горячей линии»).

2. Практическое применение (адаптация) данного процесса.

3. Определение организаций (лиц), ответственных за сопровождение.

4. Оценка стоимости сопровождения:

4.1. Проезд до места расположения пользователя.

4.2. Обучение как сопроводителей, так и пользователей.

4.3. СПИ (среда программной инженерии) и СТПС (среда тестирования программного средства) и их ежегодное сопровождение.

4.4. Персонал (зарплата и премии).

- Должен быть сформирован соответствующий план сопровождения. Этот план должен подготавливаться одновременно с разработкой программной системы. План должен определять, как пользователи будут размещать свои запросы на модификацию (изменения) или сообщать об ошибках, сбоях и проблемах.

Восстановление требований по
программному коду в процессе
реинжиниринга

- ПМ10 "Восстановление требований по программному коду в процессе реинжиниринга"(1 неделя)

Задание:

- Применить технологий реинжиниринга
- Применить методы абстрагирования спецификаций до уровня требований.
- Провести объектно-ориентированный анализ
- Проектировать программное обеспечение с использованием специализированных программных пакетов.
- **Deadline: 23.06**
- **Hard deadline: 25.06.**

Реинжиниринг программного обеспечения — процесс создания новой функциональности или устранения ошибок, путём революционного изменения, но используя уже имеющееся в эксплуатации программное обеспечение.

Сложность реинжиниринга

- Как правило, утверждается, что «легче разработать новый программный продукт». Это связано со следующими проблемами:
- обычному программисту сложно разобраться в чужом исходном коде;
- реинжиниринг, чаще всего, дороже разработки нового программного обеспечения, так как требуется убрать ограничения предыдущих версий, но при этом оставить совместимость с предыдущими версиями;
- реинжиниринг не может сделать программист низкой и средней квалификации — даже профессионалы, часто не могут качественно реализовать его, поэтому требуется работа программистов с большим опытом переделки программ и знанием различных технологий.
- В то же время, если изначально программа обладала строгой и ясной архитектурой, то провести реинжиниринг будет на порядок проще. Поэтому при проектировании, как правило, анализируется, что выгоднее — провести реинжиниринг или разработать программный продукт «с нуля».

- **Основная идея объектно-ориентированного анализа и проектирования** (object-oriented analysis and design) состоит в рассмотрении предметной области и логического решения задачи с точки зрения объектов (понятий и сущностей). В процессе объектно-ориентированного анализа основное внимание уделяется определению и описанию объектов (или понятий) в терминах предметной области. В процессе объектно-ориентированного проектирования определяются логические программные объекты, которые будут реализованы средствами объектно-ориентированного языка программирования. Эти программные объекты включают в себя атрибуты и методы. И, наконец, в процессе конструирования (construction) или объектно-ориентированного программирования (object-oriented programming) обеспечивается реализация разработанных компонентов и классов.
- Рассмотрим вкратце некоторые основные принципы объектно-ориентированного анализа и проектирования, в дальнейшем мы дадим более точные определения и более полную расшифровку всех рассмотренных терминов.

Рассмотрим вкратце некоторые основные принципы объектно-ориентированного анализа и проектирования

- С начала производится так называемый анализ требований (requirements analysis) во время которого выделяются основные процессы, происходящие в моделируемой системе и их формулировка в виде прецедентов.
- Шаг второй. Объектно-ориентированный анализ предметной области (object-oriented domain analysis). Задача этого шага в определении видов деятельности участников процесса и составлении концептуальной модели (conceptual model), которая отражает различные категории элементов предметной области. Причем не только виды деятельности участников, но и все относящиеся к делу понятия.
- Шаг третий. Разбираемся, кто, чем занимается. Эта деятельность и называется объектно-ориентированным проектированием (object-oriented design), при котором основное внимание сосредоточено на распределении обязанностей. Распределение обязанностей (responsibility assignment) означает выделение задач и обязанностей различных программных объектов в приложении.

- Наиболее важным моментом объектно-ориентированного анализа и проектирования является квалифицированное распределение обязанностей между компонентами программной системы. К тому же он оказывает определяющее влияние на робастность, масштабируемость, расширяемость и возможность повторного использования компонентов.
- Обязанности объектов и их взаимодействия изображаются с использованием диаграмм классов (design class diagram) и диаграмм взаимодействий (collaboration diagram).

- принцип абстрагирования - заключается в выделении существенных аспектов системы и отвлечения от несущественных;

Case-технология проектирования программного обеспечения информационных систем

- CASE-технология (Computer-Aided Software / System Engineering) представляет собой совокупность методологий анализа, проектирования, разработки и сопровождения сложных систем программного обеспечения (ПО), поддерживаемую комплексом взаимосвязанных средств автоматизации. CASE предоставляет системным аналитикам, проектировщикам и программистам инструментарий для автоматизации проектирования и разработки ПО.

- ПМ04 "Участие в ревьюировании программных продуктов"(2 недели)

Задание:

- Написать техническую спецификацию / **use case diagram**
- Осуществить ревьюирование кода и технической документации (произвести тестирование и оптимизацию созданного программного кода) / **Написать тестовый сценарий**
- Выполнить процесс измерения характеристик компонент программного продукта. / **определить характеристики ПО (doc “Conspect”)**
- **Deadline: 4.06**
- **Hard deadline: 11.06.**

- ПМ05 "Сопровождение программного обеспечения компьютерных систем"(1 неделя)

Задание:

- Осуществить выбор методов и средств измерения эксплуатационных характеристик объектов профессиональной деятельности. / **Должен быть сформирован соответствующий план сопровождения.**
- Выполнить работы по модификации отдельных компонент программного обеспечения.
- **Deadline: 16.06**
- **Hard deadline: 18.06.**

- ПМ10 "Восстановление требований по программному коду в процессе реинжиниринга"(1 неделя)

Задание:

- Применить технологий реинжиниринга
- Применить методы абстрагирования спецификаций до уровня требований.
- Провести объектно-ориентированный анализ /**Методологии**
- Проектировать программное обеспечение с использованием специализированных программных пакетов. /**UML**
- **Deadline: 23.06**
- **Hard deadline: 25.06.**