

Урок 1. Программное
обеспечение.

Жизненный цикл программного обеспечения

NORDIC IT SCHOOL

ПОДГОТОВИТЕЛЬНЫЕ РАБОТЫ

- Возьмите с собой блокнот и ручку, или откройте заметки
- Запись занятия, материалы и домашнее занятие будут выложены позже (сегодня)
- Создаем группу в телеграмме и добавляем всех обучающихся для оперативной рассылки, общения и вопросов

Немного обо мне

Анастасия Гаврина – старший инженер по разработке в Сбербанк

- Более 5 лет работаю в ИТ, 4 года работаю в тестировании
- Работала в таких компаниях как Люксофт, ВТБ, Сбербанк
- В сбербанке полностью веду процессы тестирования своего продукта – приложение для работы сотрудников кол – центра сбербанка



Давайте познакомимся

- Как Вас зовут, откуда Вы?
- Почему Вы пришли на наш курс?
- Расскажите, что Вас привлекает в тестировании?

О чем мы сегодня поговорим

- Что такое программное обеспечения?
- Типы программного обеспечения
- Жизненный цикл программного обеспечения – этапы, что происходит на каждом этапе цикла
- Практические задания
 - Тестирования формы ввода имени
 - Тестирование элементов дизайна

Программное обеспечение

Программное обеспечение(ПО)- это совокупность программ, позволяющих осуществить на компьютере автоматизированную обработку информации.

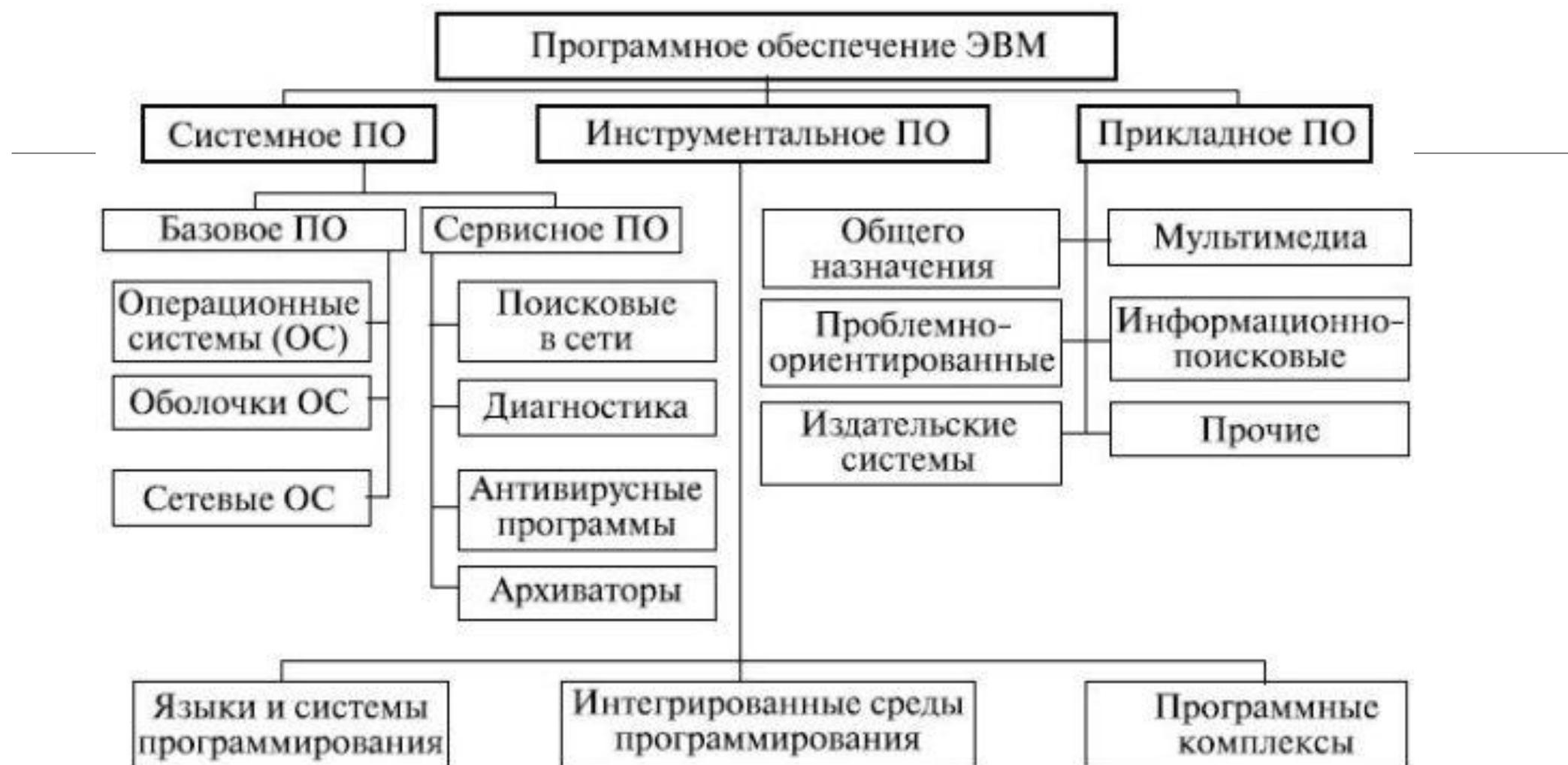
Функции ПО:

- Управлять компьютерными ресурсами организации
- Обеспечивать пользователя всеми инструментами, необходимыми для извлечения пользы из этих ресурсов
- Выполнять роль посредника между организациями и хранимой информацией
- Осуществлять математические, логические, физические и другие действия с набором имеющихся данных, другими словами обработку имеющейся информации для решения определенных задач



Типы программного обеспечения

1. Системное ПО – набор программ, которые управляют компонентами компьютера, такими как процессор, коммуникационные и периферийные устройства;
2. Прикладное ПО – набор программ, которые предназначены для выполнения определенных задач и рассчитаны на непосредственное воздействие с пользователем;
3. Инструментальное ПО – набор программ, предназначенный для разработки программного обеспечения.



Жизненный цикл ПО



Жизненный цикл программного обеспечения(ЖЦПО / SDLC) - период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации

Этапы и подпроцессы ЖЦПО

1. Начало

- Потребность
- Идея
- Задача
- Техническое задание

2. Анализ и оценка

- Анализ и оценка данных
- Анализ возможностей
- Оценка сроков и стоимости
- Оценка рисков

3. Планирование

- Планирование этапов проекта
 - Планирование ресурсов
 - Подготовка проектной документации
-

4. Проектирование

- Создание функциональной спецификации
- Создание архитектурной спецификации
- Дизайн пользовательских интерфейсов

5. Реализация

- Программирование
- Документирование

6. Тестирование

- Тестовая документация
 - Функциональное тестирование
 - Заведение отчетов об ошибке
-

7. Внедрение и сопровождение

- Вывод продукта на рынок
- Обучение пользователя работе с новым продуктом
- Консультация пользователей
- Доработка продукта

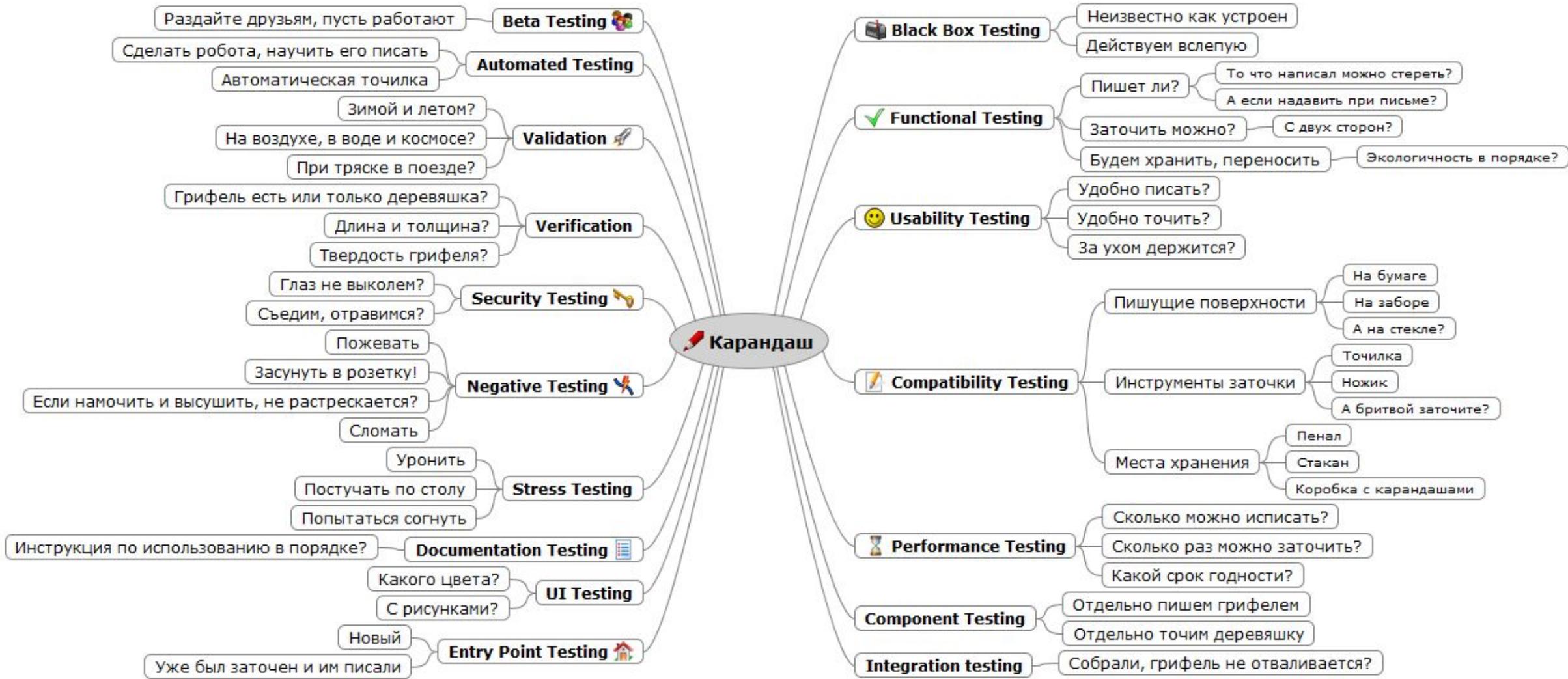
8. Вывод из эксплуатации

Роли в ИТ

- Менеджер проекта
- Менеджер продукта
- Аналитик
- Дизайнер
- Разработчик
- Специалист по тестированию
 - Тестировщик
 - QC (Quality Control)
 - QA (Quality Assurance)
- Scrum master (опционально)

Тренировочные задачи

- 1. Давайте протестируем карандаш!



2. Проверка поля для формы регистрации

<http://testingchallenges.thetestingmap.org/>

Основные используемые кейсы:

Базовые кейсы

1. Обычное значение (Average value)

Форма должна работать корректно, если пользователь ввёл в него обычное имя, например, *Kate*.

2. Граничные значения (Minimum value/Maximum values/More than maximum values)

Одной из самых распространённых техник тест-дизайна является анализ граничных значений. В спецификации указано, что максимальное количество символов в поле [30]. Проверим значения на границе (30, 31), а также минимально возможное [1] символ.

3. Пустое значение (Empty value)

В данной форме поле имени отмечено, как обязательное. Первый из негативных тестов, который стоит проверить - это отправка пустого значения

Кейсы с "небуквами"

4. Не ASCII символы (Non ASCII)

Форма представлена на английском языке, и она подразумевает ввод имени в английской раскладке. Но что будет, если ввести, например, кириллические символы (Иван/Пётр...)?

5. Небуквенные символы (Other chars then alphabetic)

В имени может содержаться дефис. Поэтому кейс *Masha-Sasha* можно считать позитивным. Но также пользователь может попытаться ввести в поле знаки препинания, числовые значения и тому подобные символы.

Кейсы с пробелами

6. Пробел внутри (Space in the middle)

Имя может быть составным, поэтому нужно проверить, что корректно обрабатывается случай, при котором введено несколько слов через пробел (*Ина бин Хатаб*).

7. Тримминг пробелов (обрезание пробелов) (Space values at the end/Space values at the begining)

Пользователь может скопировать своё имя из документа или другой страницы и, вместе с тем, пробелы тоже могут попасть случайно в буфер обмена. Поэтому важна проверка обрезания пробелов перед и после имени.

8. Пробел (Space)

Что если пользователь будет достаточно ленив, чтобы вводить своё имя, и поставит в поле только пробел? Это тоже важный кейс, потому что пробелы по-разному обрабатываются в БД, и такое значение может повлечь за собой непредвиденные ошибки.

Напоследок – игра на проверку
дизайна

<https://cantunsee.space/>

Что мы сегодня узнали?

- С чем Вы выходите с занятия?
- Остались ли вопросы по занятию?
- Пожалуйста оцените наше сегодняшнее занятие

Спасибо за внимание!