# Вступление

## Тестирование

Тестирование-процесс, подтверждающий правильность программы демонстрирующий, что ошибок в программе нет.

Тестирование призвано улучшить через обнаружение багов;

QA призвано улучшить ПО через улучшение процесса разработки ПО.

# Типы тестирования

**Познаниювнутренностейсистемы:**

* *Черныйящик*(blackboxtesting)–областьнезнаниятестируемой частибэк-энда(e.g.кодпрограммиста);
* *Белыйящик*(whiteboxtesting)–знаниеустройстваилогики тестируемойчастибэк-энда;
* *Серыйящик*(greyboxtesting)–видтестирования,сочетающий элементычерногоибелогоящиков.

**Пообъектутестирования:**

* + *Функциональноетестирование*(Functionaltesting)–проверка функционалаПО;
  + *Тестированиескоростиинадежности*(load/performance/stress testing)–проверкаповеденияпродуктаприодновременном наплывемножествапользователей;
  + *Тестированиеинтерфейсапользователя*(UItesting)–проверка элементовинтерфейсапользователя;
  + *Тестированиелокализации*(Localizationtesting)–проверка множествааспектов,связанныхсадаптациейпродуктадля пользователейиздругихстран.
* *Тестированиебезопасности*(Securitytesting)–проверка множествааспектов,сутькоторыхзаключаетсявусложнении условийдлякражи(e.g.денег,информации);
* *Тестированиеопытапользователя*(Usabilitytesting)–вид тестирования,позволяющийобъективнооценитьопыт пользователя(userexperience);
* *Тестированиесовместимости*(Compatibilitytesting)– позволяетпроверитьвзаимодействиемеждуe.g.Веб-сайтоми ПО.

**Посубъектутестирования:**

* *Альфа-тестировщик*(Alphatester)–сотрудник, которыйпрофессионально/непрофессионально проводяттестирование;
* *Бета-тестировщик*(Betatester)–тестировщик, неявляющийсясотрудникомкомпании,проводит тестированиедотого,какПОстановится доступнымвсемостальным.

**Повременипроведениятестирования:**

* *Альфа-тестирование*(Alphatesting)–**до**передачиПО пользователю;
* Бета-тестирование(Betatesting)–**после**передачиПО пользователю
* тестприемки(smoketest);
* тестированияновыхфункциональстей(newfeature testing);
* регрессивноетестирование(regressiontesting);
* тестсдачи(acceptancetest).

**По критерию “позитивности”сценариев:**

* *Позитивноетестирование*(Positivetesting)– тестирование,предпологающее“правильное”,нормальное использованиеиработусистемы;
* *Негативноетестирование*(Negativetesting)–сценарий, проверяющийситуацию,связаннуюспотенциальной ошибкой(error)ипотенциальнымдефектом(failure).

**Постепениизолированноститестируемыхкомпонентов:**

* *Компонентноетестирование*(Componenttesting)– тестированиелогическогокомпонента;
* *Интеграционноетестирование*(Integrationtesting)– тестированиевзаимодействия2-хилиболее компонентов;
* *Системноетестирование*(System/end-to-endtesting)– проверкавсейсистемыотначаладоконца.

**Постепениавтоматизированноститестирования:**

* *Ручноетестирование*(Manualtesting)–исполнение тестированиябезпомощипрограмм;
* *Автоматизированноетестирование*(Automatedtesting)– исполнениетестированияприпомощипрограмм;
* *Полуавтоматизированноетестирование*(Semiautomated testing)–сочетаниеручногоподходасавтоматизированным.

**Постепениподготовкиктестированию:**

* *Тестированиепотест-кейсам*(Documentedtesting)– исполнениетестированияподокументации;
* *Интуитивноетестирование*(Adhoctesting)– интуитивноеисполнениетестирования.

# Баг

*Баг(bug)—*этоотклонениефактическогорезультата(actual result)отожидаемогорезультата(expectedresult)

* *Фактическийрезультат(actualresult)*-наблюдаемоеили генерируемоеповедениекомпонентаилисистемывовремя тестирования.
* *Ожидаемыйрезультат(expectedresult)*-поведение компонентаилисистемыприустановленныхусловиях,которое определенноспецификациейилидругимиисточниками

# Цикл тестирования

**Тестированиепрограммногообеспечения**—процесс исследованияпрограммногообеспечения(ПО)с цельюполученияинформацииокачествепродукта

ЦиклтестированияПО состоитизтрехэтапов:

* Изучениеи анализпредметатестирования;
* Планированиетестирования;
* Исполнениетестирования;

Налюбомизэтапов*можетбытьнайденбаг*(каквПО,такив документации),багдолженбытьотремонтирован ответственнымлицом(например,программистомилиПМ),и качестворемонтадолжнобытьсертифицировано Тестировщиком

Изучение и анализ предмета тестирования

* Функциональность*(functionality,feature)*—этосредстводля решениянекойзадачи.
* Проверкаработыфункциональностейназывается **функциональнымтестированием***(functionaltesting)*

Планирование тестирования

Этастадиятребуетот*Тестировщика*наибольшеготворчестваи профессионализма,таккакименнонанейрешаетсямножество головоломок,отвечающихнаодинпростойвопрос:"Какбудем тестировать?",причемкачествопродуктанапрямуюзависитот мудростинайденныхрешений:

* Путипроверкифункциональностейдолжныбытькраткимии простыми;
* Нахождениякомпромиссамеждуобъемомтестирования, которыйвозможенв*теории*иобъемомтестирования, которыйвозможенна*практике*

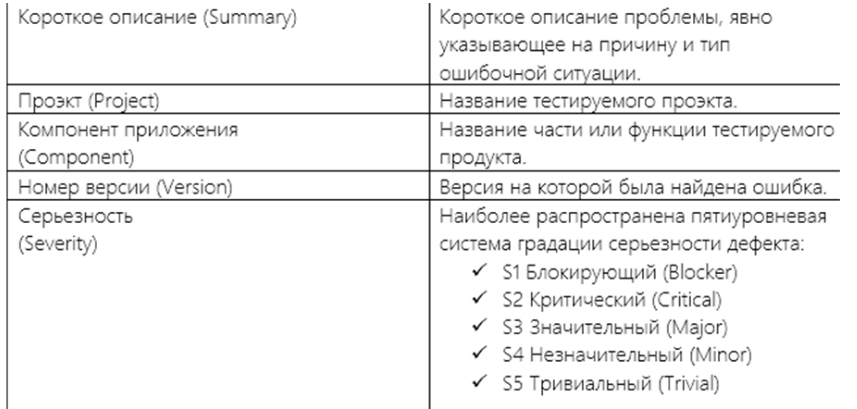
Исполнение тестирования

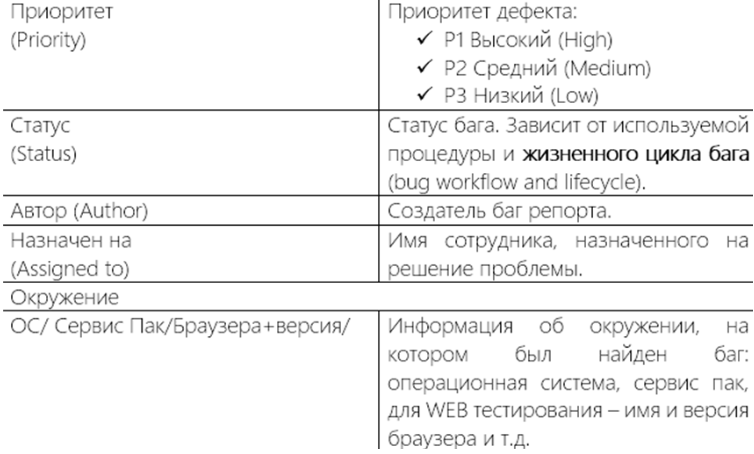
* *Сутьисполнениятестирования—*этопрактическийпоиск баговвнаписанномкодесиспользоваиемтест-кейсов, созданныхранее.
* Сначалаидетпроверка*новыхфункциональностей*поновым тест-кейсам*.*Вомногихслучаяхновыетест-кейсыредактируются, проходяобработкупервымисполнением;
* Затемпроверка*старыхфункциональностей*постарымтест-кейсам.Тоесть,происходиттестированиеновых функциональностей*(newfeaturetesting)*исоответственно регрессивноетестирование*(regressiontesting)*

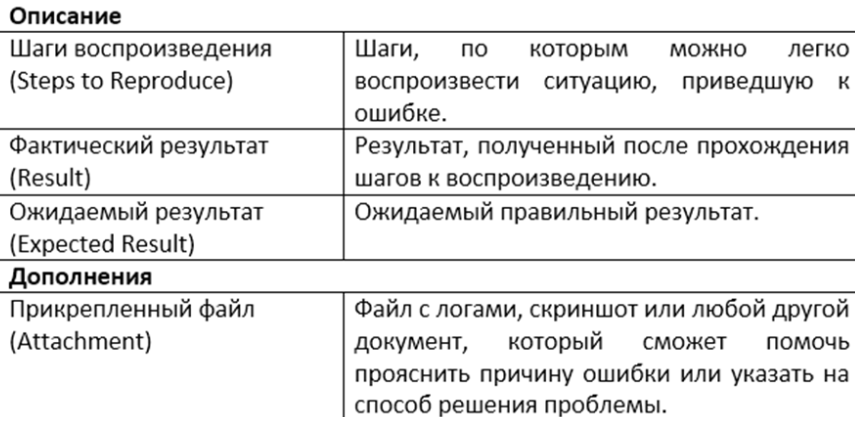
# Баг репорт

*Багрепорт*-этодокумент,описывающийситуациюили последовательностьдействийприведшуюк некорректнойработеобъектатестирования,суказанием причиниожидаемогорезультата

Структура Баг репорта

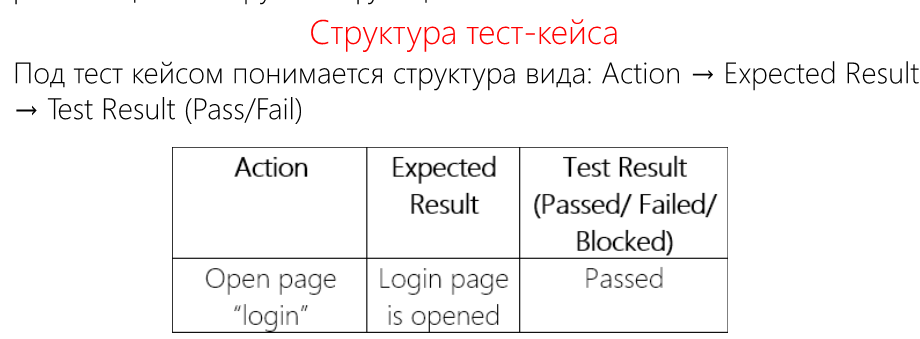






# Тест кейс

**TestCase**-этодокумент,описывающийсовокупностьшагов, конкретныхусловийипараметров,необходимыхдляпроверки реализациитестируемойфункцииилиеёчасти



Структура тест-кейса

* шаги —это инструкция по вводу;
* исполнение шагов —это ввод;
* ожидаемый результат —это ожидаемый вывод;
* фактический результат —это фактический вывод.
* Каждыйтест-кейс,исполнениекоторогозавершено,даетнамодно издвух:
* 1.Положительныйисход(PASS),еслифактическийрезультат*равен* ожидаемомурезультату,либо
* 2.Отрицательныйисход(FAIL),еслифактическийрезультат*неравен* ожидаемомурезультату;