1 вопрос

Перед началом разработки или тестирования новой функции рекомендуется выполнить следующие подготовительные действия:

1. Анализ и понимание требований: Важно полностью понять требования к новой функциональности. Это позволяет определить, какие функции должны быть реализованы и какая функциональность уже существует.

2. Создание плана разработки или тестирования: Необходимо разработать детальный план, включающий этапы, сроки и ответственных лиц. План помогает организовать работу и обеспечить ее успешное выполнение.

3. Анализ рисков: Оценка и анализ возможных рисков связанных с новой функциональностью. Это поможет идентифицировать потенциальные проблемы заранее и разработать стратегии по их устранению.

4. Создание тестовой среды: Разработать или подготовить окружение, в котором будет проводиться тестирование новой функциональности. Это может включать создание тестовых баз данных, настройку среды разработки и так далее.

5. Участие заинтересованных сторон: Связаться с заинтересованными сторонами, такими как менеджеры продукта, дизайнеры и другие разработчики, чтобы уточнить требования и согласовать план работы.

6. Подготовка документации: Разработать необходимую документацию, такую как технические спецификации, пользовательские руководства или тестовые сценарии. Это помогает обеспечить единое понимание новой функциональности и упрощает последующее тестирование и документирование.

7. Проверка наличия всех необходимых ресурсов: Убедиться, что у вас есть все необходимые инструменты, ресурсы и доступы, чтобы успешно начать разработку или тестирование новой функциональности.

8. Планирование тестовых случаев: Разработка тестовых случаев, которые покрывают основные сценарии использования новой функциональности. Это помогает обнаружить потенциальные проблемы и убедиться, что новая функциональность работает должным образом.

Эти подготовительные действия помогут обеспечить успешное разработку или тестирование новой функциональности, а также предотвратить возможные проблемы и задержки.

2 вопрос

После выкатки фичи в продукцию следует выполнить следующие действия:

1. Проверить, что фича работает правильно и не вызывает проблем для пользователей. Протестировать ее функциональность и убедиться, что она соответствует требованиям и ожиданиям.

2. Мониторинг производительности. Необходимо следить за производительностью системы после внедрения новой фичи. Если производительность системы снижается, возможно, потребуется оптимизация или доработка фичи.

3. Следить за обратной связью пользователей. Пользователи могут сообщить об ошибках или проблемах, которые возникают после выкатки фичи. Важно внимательно отслеживать и реагировать на обратную связь, чтобы быстро исправить проблемы.

4. Провести анализ использования фичи. Мониторить, как пользователи активно используют новую функциональность, и анализировать полученные данные. Это поможет понять, насколько полезна новая фича и позволит сделать выводы о ее эффективности.

5. Планирование последующих шагов. Важно определить, какие будут следующие шаги по развитию фичи и планы по ее улучшению. Это может включать в себя доработку, оптимизацию или добавление новых возможностей.

6. Коммуникация с командой и заинтересованными сторонами. Рассказать о выкатке фичи и ее результаты команде и другим заинтересованным сторонам. Обсудить обратную связь пользователей и планы на будущее.

7. Выпустить исправление, если будут обнаружены какие-либо ошибки или проблемы. Если появятся ошибки или проблемы в работе фичи, важно быстро реагировать и выпустить исправление, чтобы минимизировать воздействие на пользователей.

8. Следить за метриками и результатами. Оценивать результаты фичи с помощью ключевых показателей эффективности и метрик. Это поможет определить, достигли ли вы поставленных целей и принять решение о дальнейшем развитии фичи или корректировке стратегии.

В целом, после выкатки фичи в продукцию необходимо продолжать активно отслеживать и улучшать ее работу, реагировать на обратную связь пользователей и стремиться к постоянному развитию продукта.

3 вопрос

При тестировании мобильных приложений есть несколько особенностей, которые следует учитывать:

1. Различные платформы: Мобильные приложения могут быть созданы как для iOS, так и для Android. Поэтому важно проводить тестирование на обеих платформах, чтобы убедиться в правильной работе приложения.

2. Различные версии устройств: Существует множество различных моделей и версий устройств, на которых могут работать мобильные приложения. Тестировщику необходимо проверить, работает ли приложение на разных устройствах и версиях операционных систем.

3. Разрешение экрана и ориентация: Мобильные устройства имеют различные разрешения экрана и возможность разных ориентаций. Приложение должно правильно отображаться и работать в любых условиях.

4. Различные сенсоры: Мобильные устройства имеют различные сенсоры, такие как акселерометр, гироскоп, GPS и т. д. Приложение должно быть протестировано на взаимодействие с этими сенсорами и должно правильно реагировать на изменения данных сенсоров.

5. Ограниченные ресурсы: Мобильные устройства обычно имеют ограниченные ресурсы, такие как процессор, память и батарея. Тестировщик должен проверить, что приложение не затрачивает слишком много ресурсов и не влияет на производительность устройства.

6. Различные сетевые условия: Мобильные приложения часто работают в различных сетевых условиях, таких как 3G, 4G или Wi-Fi. Тестировщик должен проверить, как приложение ведет себя при низкой скорости интернета или при потере связи.

7. Мобильные браузеры: Если приложение работает в браузере, следует протестировать его на различных мобильных браузерах, таких как Safari, Chrome или Firefox, чтобы убедиться в его правильной работе и совместимости.

В целом, тестирование мобильных приложений требует более широкого охвата и учета различных факторов, чем при тестировании обычных веб-приложений.