**Контрольные вопросы:**

**1. Опишите структуру документации программного продукта.**

* Каждая страница должна иметь одну, четко определенную цель
* Каждая страница должна иметь четко определенную аудиторию
* Страницы должны быть связаны с другими релевантными страницами
* Структурируйте свои страницы как дерево, с обзорными страницами ближе к корню, предоставляя больше деталей по мере продвижения к листьям
* Предоставить несколько страниц содержания, которые отражают роли и интересы читателей
* Обеспечить комплексный поиск
* Поощряйте пользователей страницы вносить свой вклад в содержание

**2. Какую информацию приводят в файлах README, FAQ, CHANGELOG?**

Что должен включать в себя README:

* имена проектов, всех подмодулей и библиотек (иногда они называются по-разному и путают новых пользователей)
* описание каждого проекта, всех подмодулей и библиотек
* 5-строчный сниппет (фрагмент кода) с примером использования (если это библиотека)
* копирайт и информация о лицензии (или ссылка на лицензию)
* инструкция, как найти документацию
* инструкции по установке, настройке и запуску
* инструкция по получению свежих исходников и подробная инструкция по сборке (или быстрый обзор и ссылка на инструкцию по установке)
* список разработчиков или ссылка на файл с описанием авторов
* инструкции по отправке сообщений об ошибках, пожеланий и запросов на изменения, патчей, а также о том как получать анонсы, присоединиться к списку рассылки или к сообществу разработчиков в других формах
* контактная информация (адрес электронной почты, сайт, название компании, адрес и т.д.)
* краткая история изменений, если это обновление или форк другого проекта
* юридическая информация (для криптографического ПО)

**3. Укажите наиболее популярные инструменты для создания установщиков, их достоинства и недостатки.**

**4. Какие работы ведутся на этапе сопровождения программного продукта?**

Принято выделять несколько линий сопровождения (структура приведена на примере внешнего сопровождения ПО):

**0 линия** (call-center, информационный центр, горячая линия) — обработка телефонных обращений от клиентов, передача обращений техническим специалистам (1-я линия сопровождения)

**1 линия** (инженер по сопровождению, инженер технической поддержки, support engineer) — консультация/настройка/устранение ошибок в работе ПО/наполнение базы знаний, составление [мануалов](https://ru.wikipedia.org/wiki/%D0%A0%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F)

**2 линия** (инженер по сопровождению, инженер технической поддержки, support engineer) — функциональное сопровождение/проектная деятельность на этапе запуска ПО на машинах заказчика

**3 линия** (инженер по сопровождению, инженер технической поддержки, support engineer) — системное сопровождение/проектная деятельность на этапе запуска ПО на оборудовании заказчика

**5. Перечислите и кратко опишите стадии готовности продукта.**

Pre-Alpha — начальная разработка

**Начальная стадия разработки** — Период времени со старта разработки до выхода стадии Альфа. Также так называются программы, не вышедшие еще в стадию альфа или бета, но прошедшие стадию разработки, для первичной оценки функциональных возможностей в действии. В отличие от альфа и бета версий, начальный этап может включать в себя не весь спектр функциональных возможностей программы. В этом случае, подразумеваются все действия выполняемые во время проектирования и разработки программы вплоть до тестирования. К таким действиям относятся — разработка дизайна, анализ требований, собственно разработка приложения, а также отладка отдельных модулей.

Alpha — внутренняя разработка

Стадия начала тестирования программы в целом специалистами-тестировщиками, обычно не разработчиками программного продукта, но, как правило, внутри организации или сообществе разрабатывающих продукт. Также это может быть стадия добавления новых функциональных возможностей. Программы на данной стадии могут применяться только для ознакомления с будущими возможностями.

Как правило, альфа-тестирование заканчивается [заморозкой свойств](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BC%D0%BE%D1%80%D0%B0%D0%B6%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) и переходит в бета-тестирование.

Beta — общественная разработка

Стадия активного [бета-тестирования](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D1%82%D0%B0-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [отладки](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B) программы, прошедшей [альфа-тестирование](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D1%8C%D1%84%D0%B0-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) (если таковое было). Программы этого уровня могут быть использованы другими разработчиками программного обеспечения для испытания совместимости. Тем не менее, программы этого этапа могут содержать достаточно большое количество ошибок.

Поскольку бета-продукт не является финальной версией, и публичное тестирование производится на страх и риск пользователя, производитель не несёт никакой ответственности за ущерб, причинённый в результате использования бета-версии. Таким образом, многие производители уходят от ответственности, предоставляя пользователям только бета-версии продукта.

Release candidate - предварительная версия

Стадия-кандидат на то, чтобы стать стабильной. Программы этой стадии прошли комплексное [тестирование](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), благодаря чему были исправлены все найденные критические ошибки. Но в то же время существует вероятность выявления ещё некоторого числа ошибок, не замеченных при тестировании. Если в течение установленного времени не будет найдено крупных недоработок — становится RTM-версией. Пример: [Windows 7 RC 7100](https://ru.wikipedia.org/wiki/Windows_7" \o "Windows 7).

**6. Перечислите и кратко опишите схемы нумераций версий программного продукта.**

**Вариант 1. Нумерация целым числом**

Обычно программы нумеруются целыми числами 1,2,3,4,5,6,7 и т.д. когда новая версия программы сложна, долго пишется и появляется только раз в год или несколько лет. После того, как такая программа будет протестирована, она помечается целым номером и выпускается для использования. Какие-либо мелкие изменения, добавляемые в процессе обслуживания программы, не учитываются в нумерации. Например, целым числом нумеруется Corel Draw (Corel Draw 10, Corel Draw 11)

**Вариант 2. Десятичная дробь**

Другой способ, который позволяет указать в версии программы серьезные и не большие изменения - это нумерация десятичной дробью. Например, как правило первая версия программы получает номер 1.0. При небольшом изменении увеличивается вторая цифра - 1.1. А при добавлении новой функции, изменяется вновь первая цифра, а вторая, следующая за ней, обнуляется - 2.0.

**Вариант 3. Последовательные числа**

Нумерация версий программы последовательными числами выглядит следующим образом.Версия программы состоит из трех или четырех чисел, разделенных точкой: например, 2.7.5.

**Первое число**- старшая версия (major), изменяется при кардинальных изменениях программы

**Второе число** - младшая версия (minor), изменяется при значительных изменениях функциональности

**Третье число (или буква)** - стадия разработки

**Альфа версия** - стадия тестирования приложения, число 0 или символ a

**Бета версия** - стадия публичного тестирования приложения, число 1 или символ b

**RC (Release candidate)** - релиз-кандидат - стадия-кандидат на то, чтобы стать стабильной версией, число 2 или символы rc

**RTM (Release To Manufacturing)**- релиз - стабильная версия приложения, число 3 или символы rtm

**GA (General availability)**- общедоступный релиз

Он может отсутствовать, и тогда вместо него ставится следующее число.

Четвертое число - небольшие изменения (micro, maintenance), изменяется при любом, даже незначительной правке программы

Когда одно из чисел увеличивается, то все следующие за ним сбрасываются до нуля: 1.0.0, 1.0.1, 1.0.2, 1.1.0 и т.д. Часто, последний ноль может отбрасываться из версии, например: 1.0.0 = 1.0

Например, последовательные числа используют в Adobe Photoshop (Adobe Photoshop 7.0)

**Вариант 4. Нумерация годом**

Обычно, год используют в качестве нумерации для программных продуктов, которые долго разрабатываются и новые версии которых выходят не очень часто. Например, продукты того же Microsof, взять хотя бы их операционную систему или пакеты офисных утилит Word, Excel, PowerPoint и т.п.

**Вариант 5. Нумерация текстом**

Кроме чисел, в нумерации программы могут участвовать и различные буквы. Например, как это сделано в интегрированной среде разработки Delphi (Delphi XE)

Выбор, как именно нумеровать программу, выбирается по следующим причинам:

в зависимости от использования системы контроля версии или иных инструментов разработки

частота изменений в программе

в маркетинговых целях, когда чтобы не отставать от конкурентов, разработчики проекта перескакивают на новый номер версии

