

# Гиперус. Инфраструктура

Платформа управления ИТ-инфраструктурой

Поддержание жизненного цикла  
платформы Гиперус

---

v1.0

ООО «Гиперус» (Hyperus LLC)  
ОГРН 1233900000850, ИНН 3900006522  
[www.hyperus.ru](http://www.hyperus.ru)

© 2023-2025 ООО «Гиперус» Все авторские права защищены.

Гиперус, Hyperus и логотип Hyperus являются товарными знаками или зарегистрированными товарными знаками ООО Гиперус в России и (или) других странах.

Воспроизведение любой части данного руководства в любой форме без письменного разрешения фирмы ООО Гиперус запрещено. В настоящее руководство могут вноситься изменения без предварительного уведомления. На момент издания были внесены все изменения. Однако если будут найдены какие-либо ошибки, фирма ООО Гиперус убедительно просит сообщить ей об этом. За ошибки, обнаруженные в руководстве, фирма ООО Гиперус ответственности не несет. Не предполагается никакое обязательство за повреждения, обусловленные использованием содержащейся здесь информации.

Все наименования компаний, продуктов и услуг, логотипы, бренды и зарегистрированные или незарегистрированные товарные знаки используются только в целях идентификации, и права собственности на них принадлежат исключительно соответствующим владельцам. Использование каких-либо брендов, наименований, логотипов или любых других сведений, изображений или материалов, принадлежащих третьим лицам, не подразумевает их поддержку. Мы не являемся владельцами подобных сведений, изображений, материалов, знаков и наименований третьих лиц.

# Оглавление

<b>Глоссарий</b>	<b>4</b>
<b>Общие сведения</b>	<b>5</b>
Назначение Программы . . . . .	6
Состав подсистем продукта . . . . .	6
Виды деятельности, функции . . . . .	6
<b>Основные сведения о жизненном цикле</b>	<b>7</b>
Основные принципы разработки . . . . .	7
Жизненный цикл ПО . . . . .	7
Стадии Жизненного цикла . . . . .	9
Проектирование . . . . .	9
Разработка . . . . .	10
Тестирование . . . . .	12
Приобретение и поставка . . . . .	13
Эксплуатация . . . . .	14
Документирование . . . . .	14
Обучение и квалификация персонала . . . . .	14
Поддержка версий и доработка . . . . .	15
Устранение сбойных ситуаций . . . . .	15
Прекращение применения . . . . .	16
<b>Список используемых стандартов</b>	<b>17</b>

# Глоссарий

---

<b>Термины</b>	<b>Определение</b>
Жизненный цикл	Развитие проекта, начиная с разработки концепции и заканчивая прекращением применения
Платформа	Гиперус, Платформа, Система управления инфраструктурой предприятий
Пользователь	Сотрудник, получивший доступ к Системе на основании правил разграничения доступа
Проект	Продукт Гиперус.Инфраструктура
Стадия	Период в пределах жизненного цикла, который относится к основному развитию Платформы и достижению контрольных точек
Компания	Общество с ограниченной ответственностью «Гиперус» (обладатель исключительного права на ПО)
Продукт	программное обеспечение, сопроводительные материалы, обновления, правообладателем которых является Общество с ограниченной ответственностью «Гиперус»
Заказчик	физическое или юридическое лицо, которое правомерно устанавливает и (или) использует Продукт. Если Продукт был загружен или приобретён от имени юридического лица, то под термином «Заказчик» далее подразумевается это юридическое лицо, для которого Продукт был загружен или приобретён, и которое поручило отдельному физическому лицу (лицам) совершать действия по установке и использованию данного Продукта

---

# Общие сведения

Под программным обеспечением подразумевается программа для электронных вычислительных машин «Гиперус.Инфраструктура» (далее - ПО), а также программная продукция, реализуемая с её использованием, включая программные компоненты, работы и услуги. При этом программная продукция рассматривается как одна из ключевых составляющих, обеспечивающих функционирование автоматизированной информационной системы, которая может быть интегрирована в систему или выделена из неё.

Процессы жизненного цикла программного обеспечения классифицируются согласно их назначению на два основных класса: основные и вспомогательные. На основе этой классификации была разработана модель жизненного цикла программы Гиперус.Инфраструктура.

Актуальное определение процессов жизненного цикла программного обеспечения основывается на двух ключевых принципах: связности и ответственности. Связность процессов предполагает, что они оптимально соединены в соответствии с практическими и выполнимыми требованиями. Ответственность сторон осуществляется в рамках жизненного цикла программного обеспечения, где организация может осуществлять один или несколько процессов. Один процесс может быть реализован одной или несколькими организациями, при этом одна из организаций (или её структурное подразделение) определяется как ответственная сторона за выполнение данного процесса.

Основные процессы программного обеспечения структурированы в стадии жизненного цикла. Вспомогательные процессы могут быть активированы в ходе выполнения процессов основного класса с целью достижения конкретных целей и, как правило, реализуются в контексте основных процессов на всех этапах жизненного цикла программного обеспечения.

Каждый процесс обладает несколькими ключевыми атрибутами:

- Принадлежность процесса к определённому классу назначения;
- Наименование процесса, которое отражает область его применения;
- Конечные цели, на достижение которых направлено выполнение процесса;
- Начальные условия выполнения, определяющие минимальные требования, необходимые для обеспечения возможности реализации данного процесса.

## Назначение Программы

Гиперус.Инфраструктура представляет собой гиперконвергентную платформу корпоративного класса, предназначенную для виртуализации вычислительных ресурсов. Данный продукт является интегрированной платформой, объединяющей в себе вычислительные ресурсы, системы хранения данных и сетевые технологии в единое решение для управления.

## Состав подсистем продукта

- Подсистема хранения данных:
  - Создание и управление Блочными устройствами;
  - Создание и управление Объектными хранилищами;
  - Создание и управление Файловыми хранилищами.
- Подсистема управления виртуализацией:
  - Управление гипервизором;
  - Управление шаблонами виртуальных машин.
- Подсистема управления сетью:
  - Управление виртуальной сетевой инфраструктурой.
- Подсистема обновлений:
  - Управление обновлениями;
  - Планирование обновлений.
- Подсистема управления инфраструктурой:
  - Визуальное Управление сетью;
  - Визуальное Управление хранилищем;
  - Визуальное Управление виртуализацией;
  - Визуальное Управление обновлениями.

## Виды деятельности, функции

**Система обеспечивает реализацию следующих функциональных возможностей:**

- Управление виртуальными сетями;
- Управление виртуализацией;
- Управление обновлениями системы;
- Управление шаблонами виртуальных машин;
- Управление Гипервизором.

# Основные сведения о жизненном цикле

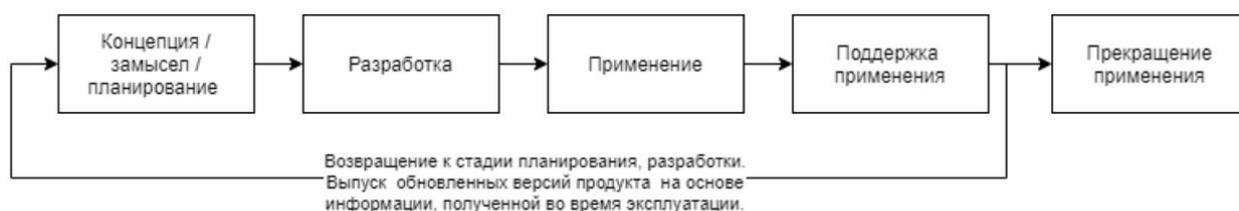
## Основные принципы разработки

При реализации проекта по созданию и сопровождению продукта Гиперус.Инфраструктура используются гибкая методология Agile и SCRUM как способ ее применения. Достижение целей, стоящих перед проектной командой, обеспечивается за счет следующих факторов:

- наивысшим приоритетом признается удовлетворение заказчика за счёт ранней и бесперебойной поставки ценного программного обеспечения;
- выстраивание проекта вокруг заинтересованных людей, которых следует обеспечить нужными условиями работы, поддержкой и доверием;
- работающее программное обеспечение — лучший измеритель прогресса;
- постоянное внимание к техническому совершенству и хорошему проектированию увеличивают гибкость;
- простота, как искусство не делать лишней работы, очень важна;
- команда регулярно обдумывает способы повышения своей эффективности и соответственно корректирует рабочий процесс.

## Жизненный цикл ПО

Для Гиперус.Инфраструктура определена долгосрочная стратегия развития, в рамках которой составляется высокоуровневый план на год. Новые версии Платформы (релизы) выпускаются, в среднем, 3 раза в год. Годовой план детализируется и корректируется для каждой выпускаемой версии. Жизненный цикл ПО рассматривается с учетом основных положений ГОСТ Р ИСО/МЭК 12207-2010 (Таблица 1), и подразумевает последовательность стадий (см. Рисунок 1).



**Рис. 1:** Стадии жизненного цикла

Переход к следующей стадии возможен после достижения целей, поставленных в предыдущей стадии. Цели и результаты стадий ЖЦ описаны в Таблице 1.

**Таблица 1:** Этапы жизненного цикла ПО Гиперус

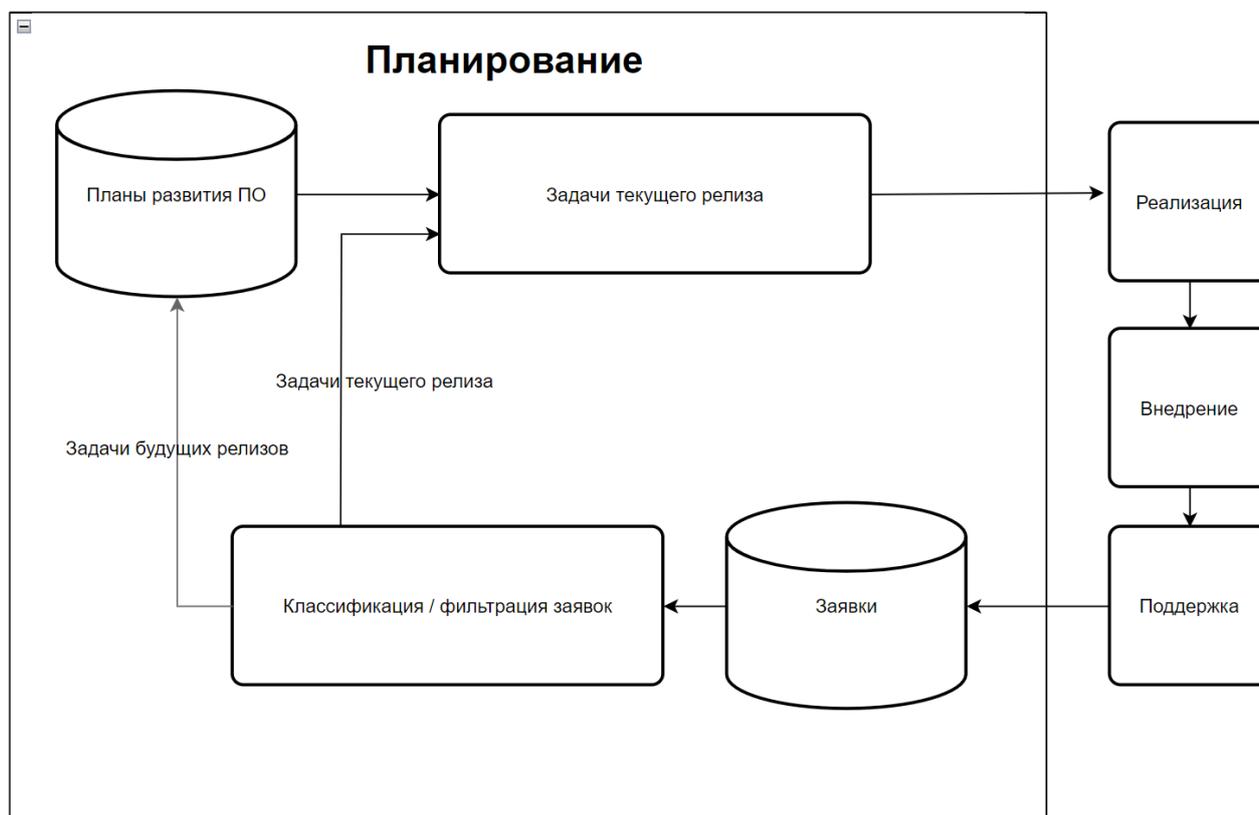
Цель	Результат
Концепция/Замысел/Планирование	
<p>Определить и приоритизировать потребности заинтересованных сторон. Выделить первоочередные задачи. Оценить варианты решений и возможности реализации</p>	<p>Составлено техническое задание на разработку ближайшего релиза с учетом заявок Заказчиков, направленных на развитие функциональности и исправление замечаний</p>
Разработка новой версии	
<p>Создать очередную релизную версию ПО, удовлетворяющую поставленным требованиям и текущим запросам пользователей</p>	<p>Создана релизная версия ПО, отвечающая поставленным требованиям и готовая к применению. Создана актуальная эксплуатационная документация, описывающая программный продукт, включая все обновления, характерные для текущей версии</p>
Применение	
<p>Обеспечить развертывание и обновление ПО на стендах и устройствах пользователей</p>	<p>Программные продукты установлены и функционируют согласно условиям их применения</p>
<p>Осуществить инспекционный контроль в соответствии с требованиями ФСТЭК России</p>	<p>Программные продукты соответствуют требованиям информационной безопасности</p>
Поддержка применения	

<p>Обеспечить устойчивое функционирование ПО, собрать статистику использования, выявить проблемы, с которыми сталкиваются пользователи</p>	<p>В течение всего срока применения собрана аналитика использования, накоплены замечания и пожелания пользователей. Заявки пользователей регистрируются, идентифицируются и классифицируются. Пользователям обеспечено содействие и консультации в соответствии с условиями соглашения. Осуществляется решение проблем пользователей, возможное в рамках текущего цикла. Вопросы, которые не могут быть решены мгновенно, фильтруются, и могут быть включены в планы реализации в одной из следующих итераций ЖЦ</p>
<p>Прекращение применения</p>	
<p>Обеспечить вывод продукта из эксплуатации</p>	<p>Продукт выведен из эксплуатации</p>
<p>Для сертифицированных версий обеспечить прекращение производства и реализации сертифицированных средств защиты информации</p>	<p>Прекращены производство и реализация сертифицированных средств защиты информации</p>

## Стадии Жизненного цикла

### Проектирование

При составлении и детализации плана развития Платформы также учитываются требования по безопасности продукта и требования регулируемые государством и полученная информация в результате взаимодействия с пользователями.

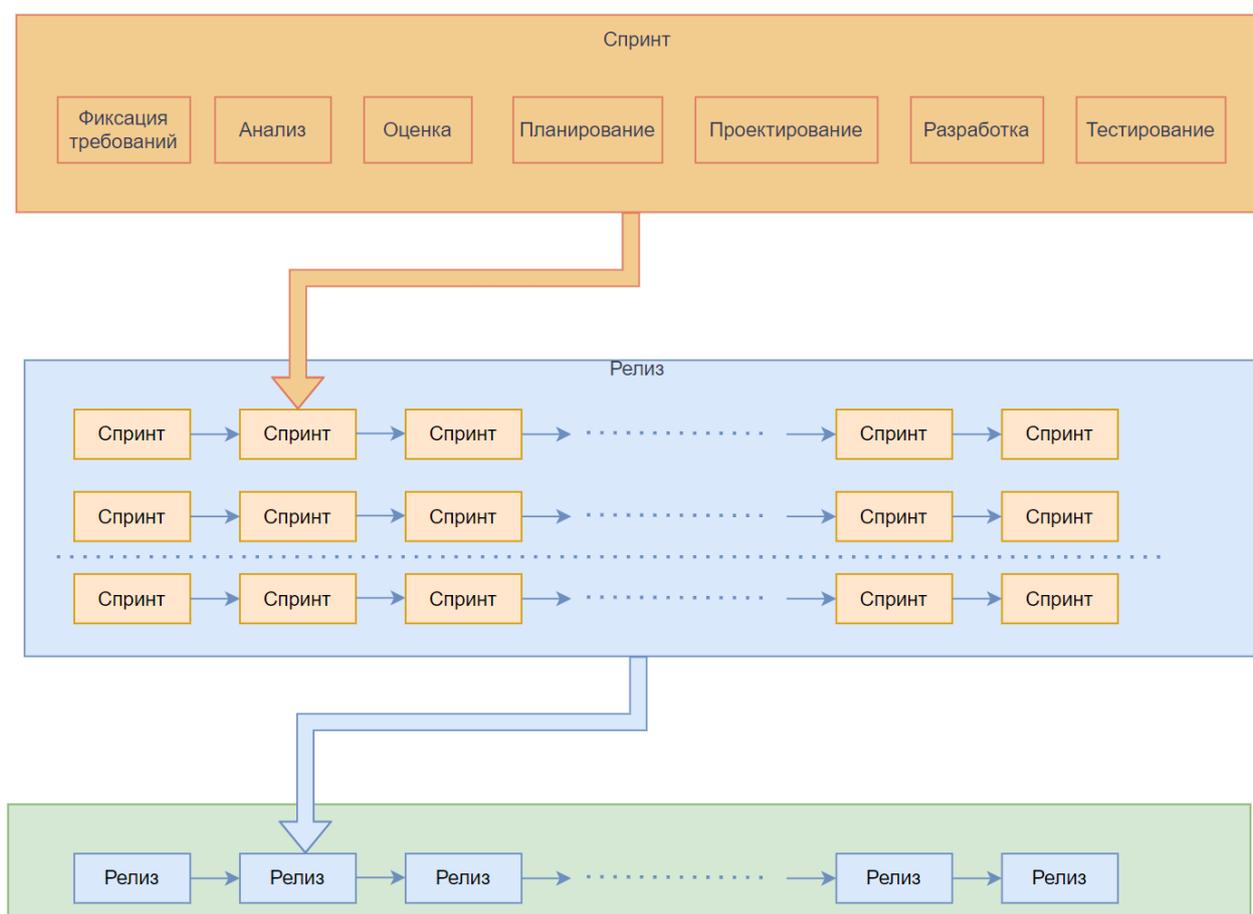


**Рис. 2:** Процесс планирования

На этапе проектирования анализируются два источника данных – набор задач, содержащихся в плане развития Платформы, а также обратная связь от клиентов. Перед каждым релизом из данных источников информации отбираются наиболее важные и актуальные задачи, формируя собой список необходимых задач текущего релиза, которые поступают на этап разработки (см. Рисунок 2).

## Разработка

Разработка новых версий Системы ведется последовательными итерациями (спринтами). Задачи каждого спринта разбиваются таким образом, чтобы обеспечить возможность реализации любой задачи не более, чем в двухнедельный период. В случае, если оценить задачу сложно, планируется время на предварительное исследование. Если задача оказывается большой, она делится на подзадачи. Таким образом, в рамках одного или нескольких последовательных спринтов каждая разрабатываемая задача проходит полный жизненный цикл разработки ПО: анализ, фиксация требований, оценка, планирование, проектирование, разработка, тестирование (см. Рисунок 3). Результат спринта анализируется и является основой для проектирования следующего спринта.



**Рис. 3:** Процесс разработки

Каждый компонент Системы разрабатывается силами отдельных кросс - функциональных команд. Такие команды укомплектованы специалистами, которые гарантируют полный цикл реализации требований к продукту:

- Менеджер продукта;
- Разработчики;
- DevOps;
- AppSec;
- Тестировщики;

Постоянный мониторинг требований на предмет соответствия критериям качества:

**Таблица 2:** Критерии качества

Критерий качества	Описание
Единичность	Требование описывает одну и только одну реализуемую сущность

<b>Критерий качества</b>	<b>Описание</b>
Завершенность	Требование полностью определено в одном месте, вся необходимая информация присутствует
Последовательность / Непротиворечивость	Требование не противоречит другим требованиям и полностью соответствует внешней документации
Атомарность	Требование «атомарно». Не может быть разбито на ряд более детальных требований без потери завершенности
Отслеживаемость	Требование полностью или частично соответствует деловым нуждам, как заявлено заинтересованными лицами, и задокументировано
Актуальность	Требование не стало устаревшим с течением времени
Однозначность	Требование определено без использования технического жаргона, акронимов и других скрытых формулировок. Оно выражает объективные факты, не субъективные мнения. Возможна одна и только одна интерпретация. Определение не содержит нечетких фраз. Использование отрицательных утверждений и составных утверждений запрещено
Обязательность	Требование представляет определенную заинтересованным лицом характеристику, отсутствие которой приведет к неполноценности решения, которая не может быть проигнорирована. Необязательное требование – противоречие самому понятию требования
Измеримость	Требование может быть измерено количественными либо качественными показателями

Фактический адрес размещения разработчиков и инфраструктуры разработки: 123001, Россия, Москва, ул. Большая Садовая, д. 5 корп.1, весь 3 (третий) этаж.

## Тестирование

Проводится как функциональное, так и нефункциональное тестирование. Функциональное тестирование включает в себя ручное тестирование пользовательского интерфейса и автоматизированное тестирование. Нефункциональное тестирование охватывает производительность и безопасность приложения, для чего применяются специализированные инструменты.

После выполнения тестов анализируются результаты. Все обнаруженные дефекты фиксируются в JIRA, где разработчики получают уведомления о новых проблемах. Регулярные встречи по статусу позволяют обсудить прогресс в исправлении ошибок и планировать повторное тестирование.

В конце каждого спринта собираются метрики качества: количество найденных дефектов, процент успешных тестов и время на исправление ошибок. Эти данные помогают улучшать процесс тестирования и повышать качество конечного продукта. Регулярные ретроспективы позволяют выявлять области для улучшения и внедрять новые практики в процесс разработки и тестирования

## Приобретение и поставка

Приобретение и поставка происходит в 4 этапа:

### 1. Инициация:

1. Квалифицированный пресейл;
2. Бюджетная оценка проекта внедрения;
3. Формирование технико-коммерческого предложения;
4. Расчет первичной конфигурации;
5. Расчет TCO;
6. Организация доступа к демо-стенду;
7. Подготовка тендерной документации.

### 2. Планирование:

1. Экспертная оценка инфраструктуры Заказчика;
2. Уточнение конфигурации;
3. Разработка архитектурной схемы и пояснительной записки к ней;
4. Уточняющие сессии;
5. Открытие проектного офиса у Заказчика;
6. Актуализация TCO.

### 3. Поставка:

1. Пилотная инсталляция;
2. Техническая поддержка на период пилота;
3. Приемо-сдаточные испытания;
4. Масштабирование и переход к эксплуатации;
5. Мероприятия по миграции и интеграции в соответствии с экспертной оценкой инфраструктуры Заказчика.

### 4. Сопровождение эксплуатации:

1. Техническая поддержка;
2. Работы по обновлению или сопровождению обновления;
3. Технические консультации.

## Эксплуатация

В процессе внедрения системы интегрируется с информационными системами, используемыми Заказчиком, настраивается на действующие у Заказчика бизнес-процессы.

Процесс эксплуатации системы осуществляется силами Заказчика.

## Документирование

Документирование – составляющая процесса разработки, целью которой является обеспечение разработки и внедрения документов, требуемых для исполнения бизнес-процессов.

Документы разрабатываются и поставляются в соответствии с планом выпуска продукта. При производстве и распределении документов могут использоваться бумажные, электронные или другие носители. Важные материалы хранятся в соответствии с требованиями по содержанию записей, защищенности, сопровождению и резервированию. Изменения в документацию вносятся при разработке программного обеспечения на постоянной основе.

## Обучение и квалификация персонала

Одной из важных задач компании Гиперус является повышение квалификации своих сотрудников. Необходимый уровень знаний, умений и навыков, как требование к каждой должности, определяется моделью профессиональных и личных компетенций, принятой для соответствующей позиции в Компании. Цель повышения квалификации персонала – поддержание и повышение уровня квалификации сотрудников с учетом требований Компании к определенной должности, роли.

Техническую поддержку ПО осуществляет Компания Гиперус. Пользователи могут обратиться в службу поддержки в следующих случаях: если возникли проблемы с установкой или настройкой программы, если столкнулись с ошибками в работе или чтобы получить разъяснения по функционалу. Наша команда технической поддержки состоит из высококвалифицированных специалистов с богатым опытом в области информационных технологий и глубоким пониманием нашего программного обеспечения. Каждый член команды прошел тщательное обучение и регулярно повышает свою квалификацию, чтобы оставаться в курсе последних тенденций и обновлений в отрасли.

Электронная почта технической поддержки компании Гиперус: [info@hyperus.team](mailto:info@hyperus.team). Контактная информация доступна на сайте <https://hyperus.ru/#contact>. Техническая поддержка доступна в рабочие дни с 10.00 до 19.00 по московскому времени.

Техническая поддержка расположена по адресу: 123001, Россия, Москва, ул. Большая Садовая, д. 5 корп.1, весь 3 (третий) этаж.

## Поддержка версий и доработка

Поддержание ЖЦ Продукта осуществляется за счет сопровождения каждого из компонентов и включает в себя:

- проведение регулярных модернизаций Продукта в соответствии с собственными планами развития;
- обновления Продукта, связанные с обращениями Заказчиков.

При выявлении критичной уязвимости в версиях ПО предприятие-изготовитель незамедлительно информирует об этом потребителей продукции и предоставляет описание проводимых организационно-технических процедур, направленных на устранение выявленной уязвимости. Предприятие-изготовитель предоставляют потребителю обновления, закрывающие выявленную уязвимость.

## Устранение сбойных ситуаций

В случае возникновения сбойных ситуаций рекомендуется обратиться к производителю. Электронная почта технической поддержки компании Гиперус: [info@hyperus.team](mailto:info@hyperus.team).

## Проблема с высокой нагрузкой на кластер, приводящей к медленной работе виртуальных машин

1. Проверка загрузки ресурсов: Войдите в интерфейс управления Гиперус.Инфраструктура и перейдите на дашборд. Здесь вы сможете увидеть загрузку CPU, памяти и дискового пространства. Обратите внимание на любые узкие места.
2. Выравнивание нагрузки: Если один из узлов перегружен, вы можете переместить виртуальные машины на другие узлы. Для этого выберите виртуальную машину, которую хотите переместить, и воспользуйтесь функцией “Migrate VM” в меню действий (Actions).
3. Оптимизация конфигурации VM: Проверьте, достаточно ли ресурсов назначено виртуальным машинам. Возможно, их необходимо пересмотреть и увеличить количество выделенных ресурсов (CPU, RAM).
4. Обновление ПО: Убедитесь, что ваша версия Гиперус.Инфраструктура обновлена до последней. Устаревшее ПО может содержать баги, которые влияют на производительность.
5. Контроль за приложениями: Если проблема сохраняется, проверьте, какие приложения запущены на виртуальных машинах. Возможно, некоторые из них требуют слишком много ресурсов, и их функциональность следует оптимизировать.

## Ошибка “No valid host found” при развертывании виртуальной машины

1. Проверка ресурсов: Убедитесь, что есть доступные ресурсы для создания новой виртуальной машины. Проверьте, достаточно ли свободных CPU, памяти и дискового пространства.
2. Квоты проекта: Убедитесь, что в проекте не достигнут лимит квот на виртуальные машины. При необходимости увеличьте квоты с помощью команды `openstack quota set`.
3. Фильтры и характеристики: Проверьте, что заданные характеристики виртуальной машины (flavor) соответствуют доступным ресурсам на гипервизорах. Если параметр flavor требует больше ресурсов, чем доступно на имеющихся узлах, то следует создать VM с меньшими параметрами.
4. Сетевая конфигурация: Убедитесь, что для создания виртуальной машины выбрана правильная сеть, и все настройки корректны. Проверьте, доступны ли ваши сети и подсети, используя команду `openstack network list`.
5. Логи и ошибки: Проверьте логи сервисов для получения более подробной информации об ошибке. Это можно сделать, зайдя в панель управления.

## Прекращение применения

По мере появления следующих версий ПО происходит обновление Продукта. Предыдущие версии прекращают использоваться, происходит миграция на новые.

**Цель процесса:** обеспечить завершение существования программного обеспечения.

**Начальные условия выполнения:** выпуск, хранение и распространение программного обеспечения прекращены. Результаты процесса: обеспечена утилизация программного обеспечения в порядке, предусмотренном законодательством Российской Федерации.

**Описание деятельности:** Деятельность состоит из следующего ряда задач:

- определение регламента утилизации программного обеспечения с учетом соблюдения законодательства Российской Федерации;
- прекращение выполнения полного или частичного перечня процессов поддержки программного обеспечения через определенный период времени;
- уничтожение программных составных частей ПО и документации связанной с ним;
- определение доступных альтернативных вариантов применения ПО (переработка ПО).

Для достижения цели и решения отдельных задач процесса также выполняются процессы:

- Ревизия и верификация процессов жизненного цикла программного обеспечения;
- Документирование программного обеспечения.

# Список используемых стандартов

Таблица 1: Используемые стандарты

ГОСТ	Описание
ГОСТ Р ИСО 9000-2015	Системы менеджмента качества. Основные положения и словарь.
ГОСТ Р ИСО/МЭК 12207-2010	Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств.
ГОСТ Р ИСО 10005-2007	Менеджмент организации. Руководящие указания по планированию качества.
ГОСТ Р ИСО 19011-2012	Руководящие указания по аудиту систем менеджмента.
ГОСТ Р ИСО 9004-2010	Менеджмент для достижения устойчивого успеха организации
ГОСТ Р ИСО 9001-2015	Системы менеджмента качества. Требования.
ГОСТ 19781-90	Обеспечение систем обработки информации. Термины и определения.