

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт**  информационных технологий | **Кафедра**  информационных систем |

**Реферат № 8**

по дисциплине «**Основы программной инженерии**»

на тему: «Сопровождение программных систем»

|  |  |  |
| --- | --- | --- |
| **Студент** группа ИДБ–22–06 |  | **Мустафаева П.М.** |
|  | подпись |  |
| **Руководитель**  к.э.н., доцент |  | **Ахмедов Э.Р.** |
|  | подпись |  |

**1. Что такое сопровождение ПО?**

Сопровождение программного обеспечения – это комплексный процесс поддержки, модификации и оптимизации ПО после его передачи в эксплуатацию. Он направлен на обеспечение стабильной работы системы, устранение дефектов, адаптацию к изменяющимся условиям и добавление нового функциональности.

Основные аспекты:

* Исправление ошибок – устранение программных дефектов, выявленных в процессе эксплуатации.
* Адаптация – настройка ПО под изменения внешней среды (например, обновление ОС или аппаратной конфигурации).
* Модернизация – расширение функциональных возможностей в соответствии с новыми требованиями пользователей.
* Стандартизация – регулируется международными и российскими стандартами (ISO/IEC 12207, ГОСТ Р ИСО/МЭК 14764), которые определяют процессы жизненного цикла ПО.
* Структура поддержки – включает многоуровневые системы обслуживания (от консультаций до системной интеграции), обеспечивающие бесперебойную работу.

Процесс сопровождения охватывает мониторинг, обновления, техническую поддержку и адаптацию ПО к бизнес-потребностям, что делает его критически важным для долгосрочной эффективности программных систем.

**2. Какие виды работ выполняются при сопровождении?**

Виды работ, выполняемых при сопровождении программного обеспечения, определяются международными стандартами, такими как IEEE 1219. Согласно этому стандарту, ключевые виды работ включают:

* Корректировка (исправление ошибок)

Внесение изменений в программное обеспечение для устранения обнаруженных дефектов и сбоев, чтобы обеспечить его стабильную работу.

* Адаптация

Модификация программного продукта для работы в новых или изменённых условиях эксплуатации, например, при смене аппаратного обеспечения, операционной системы или других внешних факторов, без расширения функциональности.

* Совершенствование (модернизация)

Добавление новых функций, улучшение производительности, повышение удобства использования и расширение возможностей программы по запросу пользователей или в соответствии с новыми требованиями.

Стандарт IEEE 1219 выделяет семь стадий процесса сопровождения, которые соответствуют этапам жизненного цикла разработки ПО:

1. Определение задачи.

2. Анализ.

3. Проектирование.

4. Реализация.

5. Системное тестирование.

6. Приёмка.

7. Поставка.

Каждая из этих стадий сопровождается работами по обработке входных данных, контролю изменений, тестированию, обновлению документации и оценке качества.

Таким образом, при сопровождении выполняются работы по исправлению ошибок, адаптации под новые условия и совершенствованию программного обеспечения, что охватывает все этапы жизненного цикла ПО после его передачи в эксплуатацию.

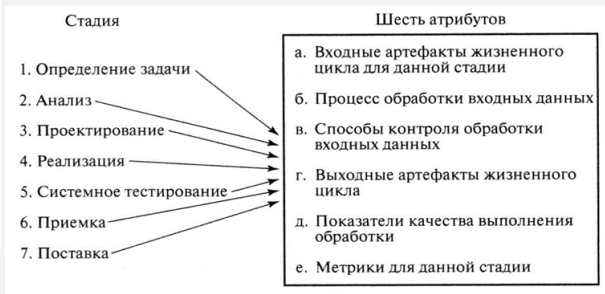


Рис. 1 Работы по сопровождению по стандарту IEEE 1219

**3. Какие основные стандарты используются при организации сопровождения?**

При организации сопровождения программного обеспечения используются международные и национальные стандарты, которые регламентируют процессы, виды работ и требования к сопровождению. Основные стандарты:

* ISO/IEC 12207

Международный стандарт, определяющий процессы жизненного цикла программного обеспечения, включая сопровождение как одну из ключевых фаз. Российский эквивалент – ГОСТ Р ИСО/МЭК 12207-2010.

* ISO/IEC 14764

Стандарт, полностью посвящённый вопросам сопровождения программных средств: определяет виды сопровождения, процессы управления, требования к планированию, выполнению, контролю и завершению работ. Российский аналог – ГОСТ Р ИСО/МЭК 14764-2002.

* IEEE 1219

Американский стандарт, описывающий процесс сопровождения ПО, включая стадии (определение задачи, анализ, проектирование, реализация, тестирование, приёмка, поставка) и шесть атрибутов для каждой стадии (входные/выходные артефакты, процессы обработки, контроль, показатели качества, метрики).

* Дополнительные стандарты и методологии

Для управления ИТ-услугами и поддержки могут применяться:

* + ISO 20000 (стандарт по управлению ИТ-услугами)
  + ITIL (библиотека лучших практик по управлению ИТ-услугами)
  + COBIT (стандарт управления ИТ-процессами)
  + SWEBOK (свод знаний по программной инженерии).

Эти стандарты обеспечивают единый подход к организации, выполнению и контролю работ по сопровождению программного обеспечения на всех этапах его жизненного цикла.

**4. Как влияет полнота документации на трудоемкость сопровождения?**

Полнота документации существенно влияет на трудоемкость сопровождения программного обеспечения. Чем полнее и качественнее документация, тем меньше времени требуется на анализ кода, поиск решений и согласование технических вопросов. Это объясняется следующими аспектами:

* Снижение исследовательских затрат

Полная документация уменьшает необходимость в обратном проектировании и анализе исходного кода, так как содержит исчерпывающие сведения о функциональности, архитектуре и особенностях ПО. Это сокращает время на поиск причин ошибок и анализ взаимодействия компонентов.

* Оптимизация коммуникации

Четкие эксплуатационные инструкции и технические спецификации минимизируют недопонимание между разработчиками и сопровождающими, снижая затраты на согласование решений и консультации. Особенно критично это при передаче сопровождения сторонним организациям.

* Фактор качества документации

Корректность и актуальность документации напрямую влияют на трудоемкость: устаревшие или противоречивые сведения увеличивают время проверки данных и риск ошибочных решений. По некоторым данным, затраты на документирование в проектах достигают 20-30%, что подчеркивает его значимость для последующего сопровождения.

* Методический аспект

Неполная документация осложняет нормирование трудозатрат, так как требует дополнительных экспертных оценок и коррективов в процессе сопровождения. Это подтверждается практикой применения поправочных коэффициентов при расчете трудоемкости.

Таким образом, полнота документации выступает ключевым фактором, определяющим эффективность процессов сопровождения ПО.

**5. Как влияет качество управления конфигурациями на трудоемкость процесса сопровождения?**

Качество управления конфигурациями напрямую снижает трудоемкость сопровождения ПО за счет следующих факторов:

* Централизованный контроль изменений

Эффективное управление конфигурациями минимизирует "конфигурационный дрейф" – расхождение настроек между средами, что предотвращает ошибки при развертывании и обновлениях. Автоматизированные инструменты (например, системы управления конфигурацией в DevOps) сокращают ручные правки и время на согласование.

* Снижение ошибок и конфликтов

Строгий учет версий ПО и зависимостей исключает ситуации, когда изменения в одном компоненте нарушают работу других. Это особенно критично в микросервисных архитектурах, где каждая служба использует собственные конфигурационные метаданные.

* Ускорение анализа проблем

Централизованное документирование всех изменений конфигурации с журналами аудита позволяет быстро выявлять причины сбоев, экономя время на диагностику.

* Стандартизация процессов

Управление конфигурациями обеспечивает соблюдение единых стандартов при переносе ПО между платформами, уменьшая трудозатраты на адаптацию.

* Связь с практиками разработки

Интеграция управления конфигурациями в DevOps и Agile-процессы позволяет заранее тестировать изменения, снижая риски критических ошибок на этапе сопровождения.

**6. Какие виды работ выполняются при осуществлении сопровождения?**

При сопровождении программного обеспечения выполняются следующие виды работ:

1. Корректировка ошибок

Исправление дефектов, выявленных в процессе эксплуатации ПО, включая диагностику причин сбоев и выпуск корректировочных версий. На эту работу приходится около 20% общих затрат на сопровождение.

2. Адаптация ПО

Модификация программного обеспечения для совместимости с новыми аппаратными платформами, ОС или внешними системами без изменения функциональности. Занимает примерно 20% ресурсов сопровождения.

3. Модернизация и расширение функциональности

Добавление новых возможностей, улучшение производительности и оптимизация кода в соответствии с обновленными требованиями. Это наиболее затратный вид работ – до 60% бюджета сопровождения.

4. Техническая поддержка пользователей

Консультации, помощь в настройке, обучение и решение проблем, связанных с эксплуатацией ПО.

5. Мониторинг и анализ производительности

Постоянное наблюдение за работой системы, выявление узких мест и прогнозирование потенциальных сбоев.

6. Управление конфигурациями и версиями

Контроль изменений в кодовой базе, ведение журналов модификаций и обеспечение согласованности версий.

7. Документирование изменений

Обновление технической документации в соответствии с внесенными правками и новыми функциональными возможностями.

8. Профилактическое сопровождение

Выявление и устранение скрытых дефектов до их проявления в продуктивной среде.

9. Перенос ПО в новую среду

Миграция системы на другие платформы или интеграция с новым оборудованием.

10. Вывод ПО из эксплуатации

Организация процесса прекращения использования устаревших систем.

Работы выполняются по стандартизированным процессам (ISO/IEC 12207, ГОСТ Р ИСО/МЭК 14764-2002), часто с использованием многоуровневой структуры поддержки.

**7. Какие ресурсы необходимы для сопровождения?**

Ресурсы, необходимые для сопровождения ПО, включают три ключевых компонента:

1. Трудовые ресурсы

Квалифицированные специалисты (разработчики, тестировщики, технические писатели), чьи затраты зависят от:

* Типа приложения (например, legacy-системы требуют больше экспертизы);
* Новизны ПО (устаревшие технологии увеличивают сложность поддержки);
* Масштаба изменений (чем больше модифицируемых компонентов, тем выше трудозатраты).

2. Временные ресурсы

Включают сроки на:

* Анализ проблем (диагностика ошибок и их воспроизведение);
* Реализацию модификаций (внесение изменений в код и документацию);
* Тестирование (проверка совместимости и отсутствия регрессий).

3. Финансовые ресурсы

Формируются из:

* Заработной платы специалистов;
* Стоимости инструментов (системы управления конфигурациями, CI/CD-платформы);
* Инфраструктурных расходов (тестовые среды, лицензии ПО).

Факторы, осложняющие прогнозирование:

* Незапланированные работы (устранение критических дефектов требует экстренного выделения ресурсов);
* Сложность оценки масштаба изменений (например, модернизация архитектуры может вызвать каскадные правки);
* Зависимость от внешних компонентов (обновления сторонних библиотек/API).
* Структура затрат:
* Корректирующие изменения (20% бюджета – исправление ошибок);
* Адаптивные изменения (20% – адаптация к новым платформам);
* Совершенствующие изменения (до 60% – добавление функционала и оптимизация).

Дополнительно требуются организационные ресурсы: стандартизированные процессы (ISO/IEC 14764), многоуровневая поддержка, системы управления версиями.

**8. Возможно ли осуществлять сопровождение ПО силами сторонних организаций, не принимавших участия в его создании?**

Сопровождение программного обеспечения может осуществляться сторонними организациями, даже если они не участвовали в его разработке. Это подтверждается распространенной практикой ИТ-аутсорсинга, когда компании делегируют поддержку ПО специализированным подрядчикам. Такой подход позволяет снизить затраты на содержание штатных специалистов и получить доступ к узкопрофильной экспертизе, особенно при работе с устаревшими технологиями или сложными системами.

Одним из ключевых механизмов передачи сопровождения является модель ИТ-аутсорсинга (ITO), при которой внешние провайдеры берут на себя обновление ПО, устранение ошибок и мониторинг производительности. Например, поддержка корпоративных CRM-систем или веб-приложений часто передается сторонним сервисам, предлагающим гибкие условия масштабирования ресурсов. Отдельное направление - облачные решения (SaaS), где провайдер автоматически обеспечивает безопасность, резервное копирование и обновления без прямого участия заказчика.

Однако передача сопровождения сопряжена с рисками. Качество документации становится критическим фактором: неполные спецификации или устаревшие схемы архитектуры значительно увеличивают сроки адаптации подрядчиков. Вопросы безопасности требуют тщательного контроля доступа к исходному коду и конфиденциальным данным, что регламентируется NDA и специализированными лицензиями. Юридические аспекты включают прописывание SLA в договорах, где четко определяются зоны ответственности, сроки устранения инцидентов и порядок передачи знаний.

На практике крупные интеграторы, такие как IBM или HP, десятилетиями предоставляют услуги сопровождения корпоративных систем. В России популярен аутсорсинг поддержки 1С, веб-платформ и мобильных приложений. Успех такого сотрудничества зависит от прозрачности процессов, наличия стандартизированных регламентов (например, ITIL) и компетентности подрядной организации. Таким образом, передача сопровождения сторонним исполнителям не только возможна, но и экономически оправдана при соблюдении организационных и технических условий.

**9. Чем реинжиниринг отличается от обычного процесса разработки?**

Реинжиниринг программного обеспечения принципиально отличается от обычной разработки по целям, методам и масштабу изменений. Вместо создания системы «с нуля» он предполагает глубокую трансформацию существующего ПО с сохранением его функционального ядра.

Ключевые отличия:

* Цель. Обычная разработка направлена на создание нового продукта, тогда как реинжиниринг фокусируется на модернизации устаревших систем для соответствия современным требованиям. Это включает переход на новые архитектуры (например, с монолитной на микросервисы), интеграцию актуальных технологий или устранение накопленного технического долга.
* Процесс. Разработка начинается с анализа предметной области и проектирования, а реинжиниринг требует обратного проектирования (reverse engineering) - декомпозиции существующего кода для понимания его логики. Это осложняется необходимостью работы с плохо документированными компонентами и устаревшими технологиями.
* Сложность. Реинжиниринг зачастую дороже разработки нового ПО из-за необходимости сохранять совместимость с предыдущими версиями и преодолевать архитектурные ограничения. Он требует участия высококвалифицированных специалистов, способных анализировать чужой код и принимать решения о целесообразности изменений.
* Риски. В отличие от разработки, где требования формируются постепенно, реинжиниринг сталкивается с «наследственными» проблемами: неочевидные зависимости между модулями, недокументированные особенности поведения системы. Это увеличивает вероятность ошибок при модификации кода.
* Результат. Обычная разработка создает систему с прогнозируемыми характеристиками, а реинжиниринг направлен на эволюционное улучшение существующего ПО. При этом ключевым преимуществом реинжиниринга считается снижение затрат по сравнению с полным переписыванием кода – в среднем в 4 раза.
* Методологии. Реинжиниринг часто сочетает элементы сопровождения (например, исправление ошибок) и разработки (перепроектирование архитектуры). В отличие от классического жизненного цикла ПО, он требует адаптивных моделей управления, таких как итеративный подход с постоянным тестированием изменений.

Таким образом, реинжиниринг служит мостом между устаревшими системами и современными технологиями, но требует уникальных компетенций и тщательного планирования.

**10. Как можно оценить трудозатраты на сопровождение.**

Оценка трудозатрат на сопровождение ПО требует комплексного подхода, сочетающего количественные методы и учет специфики проекта. Основные методы и практики включают:

* Декомпозиция задач

Разделение процесса сопровождения на атомарные операции (корректировка ошибок, обновление документации, мониторинг производительности) с последующей оценкой времени на каждую задачу методом «снизу-вверх». Например, исправление типовой ошибки может занимать 2-4 часа с учетом анализа, правки кода и тестирования.

* Использование исторических данных

Анализ фактических трудозатрат по аналогичным проектам. Если сопровождение предыдущей версии системы требовало 100 часов в месяц, эта цифра становится базой для прогнозирования с поправкой на сложность изменений.

* Параметрические модели

Применение формул типа «время на обновление конфигурации = 1 час на модуль + 0,5 часа на интеграцию». Для CMDB сопровождения выделяют отдельные нормы: 30 минут на регистрацию конфигурационной единицы, 2 часа на аудит.

* Экспертная оценка

Привлечение специалистов, знакомых с системой, для прогнозирования времени на реализацию изменений. Метод уязвим к субъективности, поэтому требует перекрестной проверки через другие подходы.

* Учет рисковых факторов

Добавление резерва (15-25% от общего времени) на непредвиденные работы: экстренные исправления критических ошибок или доработки из-за неполной документации.

* Автоматизация учета

Использование ITSM-систем для фиксации фактических трудозатрат по категориям (инциденты, запросы, изменения). Инструменты вроде ITSM 365 автоматически агрегируют данные по времени, затраченному на обработку заявок.

* Распределение по фазам

Следование структуре, выявленной в веб-проектах: 50-60% на непосредственную разработку/исправления, 15% на тестирование, 22% на управление. Для сопровождения пропорции корректируются в сторону увеличения операционной поддержки.

* Метод трех точек

Оценка по оптимистичному, пессимистичному и реалистичному сценариям с расчетом средневзвешенного значения.