

УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите приведенную лекцию, законспектируйте основные сведения о способах сопровождения программ.

Ответы на вопросы, фотоотчет, предоставить преподавателю на e-mail: r.bigangel@gmail.com **до 13.03.2023.**

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: <https://vk.com/daykini>

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать *ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).*

Лекция № 38 (продолжение)

Тема: «Сопровождение программ»

Цель: Изучить основные этапы сопровождения программ.

Сопровождение — это обычный процесс изменения системы после ее поставки заказчику. Эти изменения могут быть как элементарно простыми (исправление ошибок программирования), так и более серьезными, связанными с корректировкой отдельных недоработок либо приведением в соответствие с новыми требованиями. Сопровождение не связано со значительным изменением архитектуры системы. При сопровождении тактика простая: изменение существующих компонентов системы либо добавление новых.

Существует три вида сопровождения системы:

1. Сопровождение с целью исправления ошибок. Обычно ошибки в программировании достаточно легко устранимы, однако ошибки проектирования стоят дорого и требуют корректировки или перепрограммирования некоторых компонентов. Самые дорогие исправления связаны с ошибками в системных требованиях, так как здесь может понадобиться перепроектирование системы.

2. Сопровождение с целью адаптации ПО к специфическим условиям эксплуатации. Это может потребоваться при изменении определенных составляющих рабочего окружения системы, например аппаратных средств, операционной системы или программных средств поддержки. Чтобы адаптироваться к этим изменениям, система должна быть подвергнута определенным модификациям.

3. Сопровождение с целью изменения функциональных возможностей системы. В ответ на организационные или деловые изменения в организации могут измениться требования к программным средствам. В таких случаях применяется данный тип сопровождения. Наиболее существенные изменения при этом претерпевает именно программное обеспечение.

На практике однозначно четкое разграничение между различными видами сопровождения провести достаточно сложно. Ошибки в системе могут быть выявлены в том случае, если, например, система использовалась непредсказуемым способом. Поэтому наилучший способ исправления ошибок — расширение функциональных возможностей программы с тем, чтобы сделать работу с ней как можно проще. При адаптации программного обеспечения к новому рабочему окружению расширение функциональных возможностей системы будет способствовать улучшению ее работы. Также добавление определенных функций в программу может оказаться полезным, если в случае ошибок был изменен шаблон использования системы и побочным действием при расширении функциональных возможностей будет удаление ошибок.

Перечисленные типы сопровождения широко используются, хотя им подчас дают разные названия. Сопровождение с целью исправления ошибок обычно называют корректирующим. Название "адаптивное сопровождение" может относиться как к адаптации к новому рабочему окружению, так и к новым требованиям. Усовершенствование программного обеспечения может означать улучшение путем соответствия новым требованиям, а также

усовершенствование структуры и производительности с сохранением функциональных возможностей.

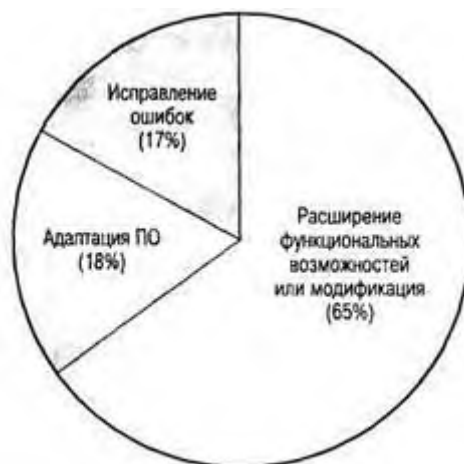


Рисунок 1. Распределение типов сопровождения

Согласно исследованиям, 65% сопровождения связано с выполнением новых требований, 18% отводится на изменения системы с целью адаптации к новому окружению и 17% связано с исправлением ошибок (рис. 1.).

Из этого можно определить, что исправление ошибок не является самым распространенным видом сопровождения. Модернизация системы в соответствии с новым рабочим окружением либо в соответствии с новыми требованиями более эффективна. Поэтому сопровождение само по себе является естественным процессом продолжения разработки системы со своими процессами проектирования, реализации и тестирования. Таким образом, спиральная модель, показанная на рис. 2, лучше представляет процесс развития ПО, чем каскадная модель, где сопровождение рассматривается как отдельный процесс.



Рисунок 2. Спиральная модель развития ПО

Значительная часть бюджета большинства организаций уходит на сопровождение ПО, а не на само использование программных систем. В 1980-х годах было обнаружено, что во многих организациях по меньшей мере 50% всех средств, потраченных на программирование, идет на развитие уже существующих систем. При этом от 65 до 75% средств общего бюджета расходуется на сопровождение. Так как предприятия заменяют старые системы коммерческим ПО, например программами планирования ресурсов, эти цифры никак не будут уменьшаться. Поэтому можно утверждать, что изменение ПО все еще остается доминирующим в статье затрат организаций на программное обеспечение.

Соотношение между величинами средств на сопровождение и на разработку может быть разным в зависимости от предметной области, где эксплуатируется система. Для прикладных систем, работающих в деловой сфере, соотношение затрат на сопровождение в основном сравнимо со средствами, потраченными на разработку. Для встроенных систем реального времени затраты на сопровождение могут в четыре раза превышать стоимость самой разработки. Высокие требования в отношении производительности и надежности таких систем предполагают их жесткую структуру, которая труднее поддается модификации.

Можно получить значительную общую экономию средств, если заранее потратить финансы и усилия на создание системы, не требующей дорогостоящего сопровождения. Весьма затратно проводить изменения в системе после ее поставки заказчику, поскольку для этого требуется хорошо знать систему и провести анализ реализации этих изменений. Поэтому усилия, потраченные во время разработки программы на снижение стоимости такого анализа, автоматически снизят и затраты на сопровождение. Такие технологии разработки ПО, как формирование четких требований, объектно-ориентированное программирование и управление конфигурацией, способствуют снижению стоимости сопровождения.

На рис. 3 показано, как снижается стоимость сопровождения хорошо разработанных систем. Здесь при разработке системы 1 выделено дополнительно \$25 000 для облегчения процесса сопровождения. В результате за время эксплуатации системы это помогло сэкономить около \$100 000. Из сказанного следует, что увеличение средств на разработку системы пропорционально снизит затраты на ее сопровождение.

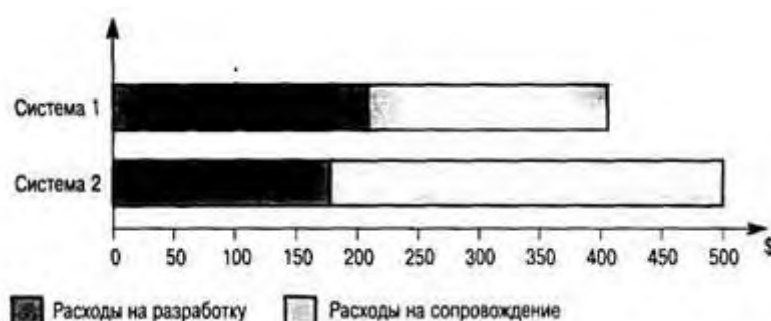


Рисунок 3. Расходы на разработку и сопровождение систем

Причиной высоких затрат на сопровождение является сложность модернизации системы после ее внедрения, поскольку расширить функциональные возможности намного легче в процессе создания системы. Ниже приведены ключевые факторы, которые определяют стоимость разработки и сопровождения и могут привести к подорожанию сопровождения.

1. *Стабильность команды разработчиков.* Вполне естественно, что после внедрения системы команда разработчиков распадается, специалисты будут работать над другими проектами. Новым членам команды или же отдельным специалистам, которые возьмут на себя дальнейшее сопровождение системы, будет трудно понять все ее особенности. Поэтому на понимание системы перед внесением в нее изменений уходит много времени и средств.

2. *Ответственность согласно контракту.* Контракт на сопровождение обычно заключается отдельно от договора на разработку программы. Более того, часто контракт на сопровождение может получить фирма, сама не занимающаяся разработкой. Вместе с фактором нестабильности команды это может стать причиной отсутствия в команде стимула создать легко изменяемую, удобную в сопровождении систему. Если членам команды выгодно пойти кратчайшим путем с минимальными затратами усилий, то вряд ли они откажутся от этого даже с риском повышения последующих затрат на сопровождение.

3. *Квалификация специалистов.* Специалисты, занимающиеся сопровождением, часто не знакомы с предметной областью, где эксплуатируется система. Сопровождение не пользуется популярностью среди разработчиков. Это считается менее квалифицированной разработкой и часто поручается младшему персоналу. Более того, старые системы могут быть написаны на устаревших языках программирования, не знакомых молодым специалистам и требующих дополнительного изучения.

4. *Возраст и структура программы.* С возрастом структура программ нарушается вследствие частых изменений, поэтому их становится сложнее понимать и изменять. Кроме того, многие наследуемые системы были созданы без использования современных технологий. Они никогда не отличались хорошей качественной структурой; изменения, сделанные в них, были направлены скорее на повышение эффективности функционирования,

чем на повышение удобства сопровождения. Документация на старые системы часто бывает неполной либо вообще отсутствует.

Первые три проблемы объясняются тем, что многие организации все еще делают различие между разработкой системы и ее сопровождением. Сопровождение считается делом второстепенным, поэтому нет никакого желания инвестировать средства для снижения затрат на будущее сопровождение. Руководству организаций необходимо помнить, что у систем редко бывает четко определенный срок функционирования, наоборот, они могут находиться в эксплуатации в той либо иной форме неограниченное время.

Дилемма заключается в следующем: или создавать системы и поддерживать их до тех пор, пока это возможно, и затем заменять их новыми, или разрабатывать постоянно эволюционирующие системы, которые могут изменяться в соответствии с новыми требованиями. Их можно создавать на основе наследуемых систем, улучшая структуру последних с помощью реинжиниринга, либо путем изменения архитектуры этих систем.

Последняя проблема, а именно нарушение структуры, является самой простой из них. Технология реинжиниринга поможет усовершенствовать структуру и повысить понимаемость системы. В подходящих случаях адаптировать систему к новым аппаратным средствам может и преобразование архитектуры. Профилактические меры при сопровождении будут полезны, если возникнет необходимость усовершенствовать систему и сделать ее более удобной для изменений.