

## Проектирование с повторным использованием компонентов

В большинстве инженерных разработок процесс проектирования основан на повторном использовании уже имеющихся компонентов. В таких сферах, как механика или электротехника, инженеры никогда не разрабатывают проект "с нуля". Их проекты базируются на компонентах, уже проверенных и протестированных в других системах. Как правило, это не только малые компоненты, например фланцы и клапаны, но также целые подсистемы, например двигатели, компрессоры или турбины.

В настоящее время не вызывает сомнений тот факт, что необходимо сравнивать различные подходы к разработке программного обеспечения. Если программное обеспечение рассматривать как актив, то повторное использование этих активов позволит существенно сократить расходы на его разработку. Только с помощью систематического повторного использования ПО можно уменьшить расходы на его создание и обслуживание, сократить сроки разработки систем и повысить качество программных продуктов.

Чтобы повторное использование ПО было эффективным, его необходимо учитывать на всех этапах процесса проектирования ПО или процесса разработки требований. Во время программирования возможно повторное использование на этапе подбора компонентов, соответствующих требованиям. Однако для систематического повторного использования необходим такой процесс проектирования, в ходе которого постоянно рассматривалась бы возможность повторного использования уже существующих архитектур, где система была бы явно организована из доступных имеющихся компонентов ПО.

Метод проектирования ПО, основанный на повторном использовании, предполагает максимальное использование уже имеющихся программных объектов. Такие объекты могут радикально различаться размерами.

1. *Повторно используемые приложения.* Можно повторно использовать целые приложения либо путем включения их в систему без изменения других подсистем (например, коммерческие готовые продукты, см. раздел 14.1.2), либо с помощью разработки семейств приложений, работающих на разных платформах и адаптированных к требованиям конкретных заказчиков.

2. *Повторно используемые компоненты.* Можно повторно использовать компоненты приложений — от подсистем до отдельных объектов. Например, система распознавания текста, разработанная как часть системы обработки текстов, может повторно использоваться в системах управления базами данных.

3. *Повторно используемые функции.* Можно повторно использовать программные компоненты, которые реализуют отдельные функции, например математические. Основанный на стандартных библиотеках метод повторного использования компонентов применяется в программировании последние 40 лет.

Повторное использование целых приложений практикуется довольно широко; при этом компании, занимающиеся разработкой ПО, адаптируют свои

системы для разных платформ и для работы в различных условиях. Также хорошо известно повторное использование функций через стандартные библиотеки, например графические и математические. Интерес к повторному использованию компонентов возник еще в начале 1980-х годов, однако на практике такой подход к разработке систем ПО применяется лишь последние несколько лет.

Очевидным преимуществом повторного использования ПО является снижение общей стоимости проекта, так как в целом требуется специфицировать, спроектировать, реализовать и проверить меньшее количество системных компонентов. Но снижение стоимости проекта — это только потенциальное преимущество повторного использования. Как видно из таблицы, повторное использование ПО имеет ряд других преимуществ.

Таблица – Преимущества повторного использования ПО

<b>Преимущество</b>	<b>Описание</b>
Повышение надежности	Компоненты, повторно используемые в других системах, оказываются значительно надежнее новых компонентов. Они протестированы и проверены в разных условиях работы. Ошибки, допущенные при их проектировании и реализации, обнаружены и устранены еще при первом их применении. Поэтому повторное использование компонентов сокращает общее количество ошибок в системе
Уменьшение проектных рисков	Для уже существующих компонентов можно более точно прогнозировать расходы, связанные с их повторным использованием, чем расходы, необходимые на их разработку. Такой прогноз — важный фактор администрирования проекта, так как позволяет уменьшить неточности при предварительной оценке сметы проекта
Эффективное использование специалистов	Часть специалистов, выполняющих одинаковую работу в разных проектах, может заниматься разработкой компонентов для их дальнейшего повторного использования, эффективно применяя накопленные ранее знания
Соблюдение стандартов	Некоторые стандарты, такие как стандарты интерфейса пользователя, можно реализовать в виде набора стандартных компонентов. Например, можно разработать повторно используемые компоненты для реализации различных меню пользовательского интерфейса. Все приложения предоставляют меню пользователям в одном формате. Использование стандартного пользовательского

	интерфейса повышает надежность систем, так как, работая со знакомым интерфейсом, пользователи совершают меньше ошибок
Ускорение разработки	Часто для успешного продвижения системы на рынке необходимо как можно более раннее ее появление, причем независимо от полной стоимости ее создания. Повторное использование компонентов ускоряет создание систем, так как сокращается время на их разработку и тестирование

Для успешного проектирования и разработки ПО с повторным использованием компонентов должны выполняться **три основных условия**.

1. Возможность поиска необходимых системных компонентов. В организациях должен быть каталог документированных компонентов, предназначенных для повторного использования, который обеспечивал бы быстрый поиск нужных компонентов.

2. При повторном использовании необходимо удостовериться, что поведение компонентов предсказуемо и надежно. В идеале все компоненты, представленные в каталоге, должны быть сертифицированы, чтобы подтвердить соответствие определенным стандартам качества.

3. На каждый компонент должна быть соответствующая документация, цель которой — помочь разработчику получить нужную информацию о компоненте и адаптировать его к новому приложению. В документации должна содержаться информация о том, где используется данный компонент, и другие вопросы, которые могут возникнуть при повторном использовании компонента.

Таблица – Проблемы повторного использования компонентов

Проблема	Описание
Повышение стоимости сопровождения системы	Недоступность исходного кода компонента может привести к увеличению расходов на сопровождение системы, так как повторно используемые системные элементы могут со временем оказаться не совместимыми с изменениями, производимыми в системе
Недостаточная инструментальная поддержка	CASE-средства не поддерживают разработку ПО с повторным использованием компонентов. Интегрирование этих средств с системой библиотек компонентов затруднительно или даже невозможно. Если процесс разработки ПО осуществляется с помощью CASE-средств, повторное использование компонентов можно полностью исключить

Синдром "изобретения велосипеда"	Некоторые разработчики ПО предпочитают переписать компоненты, так как полагают, что смогут при этом их усовершенствовать. Кроме того, многие считают, что создание программ "с нуля" перспективнее и "благороднее" повторного использования написанных другими программ
Содержание библиотеки компонентов	Заполнение библиотеки компонентов и ее сопровождение может стоить дорого. В настоящее время еще недостаточно хорошо продуманы методы классификации, каталогизации и извлечения информации о программных компонентах
Поиск и адаптация компонентов	Компоненты ПО нужно найти в библиотеке, изучить и адаптировать к работе в новых условиях, что "не укладывается" в обычный процесс разработки ПО

Из перечисленного выше следует, что повторное использование компонентов должно быть систематическим, плановым и включенным во все организационные программы организации-разработчика.