## Техники сопровождения: реинжиниринг; “обратный” инжиниринг.

Данная секция вводит некоторые общепринятые техники, используемые в процессе сопровождения программных систем.

*4.1 Понимание программных систем (Program Comprehension)*

Для реализации изменений программисты тратят значительную часть времени на чтение и формирование понимания программного продукта. Средства работы с кодом являются ключевым инструментом для решения этой задачи. Четкая, однозначная и лаконичная документация обеспечивает адекватное понимание программных систем.

*4.2 Реинжиниринг\* (Reengineering)*

Реинжиниринг определяется как детальная оценка (examination) и перестройка программного обеспечения для формирования понимания, воссоздания (на уровне модели и, в ряде случаев, требований, *прим. автора*) и дальнейшей реализации его <функций> в новой форме (например, с использованием новых технологий и платформ, при сохранении существующей и расширением и облегчением возможностей добавлений новой функциональности, *прим. автора*). Отмечается, что в индустрии существуют различные позиции в отношении реинжиниринга – одни считают, что реинжиниринг является наиболее радикальной и затратной формой изменений программных систем, другие, что такой подход может применяться и для не столь кардинальных изменений (например, как смена платформы или архитектуры, *прим. автора*). Реинжиниринг, обычно, провидится не столько для улучшения возможностей сопровождения (maintainability), сколько для замены устаревшего программного обеспечения. В принципе, реинжиниринг можно рассматривать как самостоятельный проект (такой позиции придерживается автор), включающий в себя, как отмечает SWEBOK, формирование концепции, применение соответствующих инструментов и техник, анализ и приложения опыта проведения реинжиниринга, а также оценку рисков и преимуществ, связанных с такими работами.

Хочу отметить, что реализация продукта в новом качестве (форме) при сохранении основной функциональности оригинального продукта, является неотъемлемой и определяющей частью реинжиниринга, в отличие от обратного инжиниринга, рассматриваемого ниже и являющегося важной составной частью реинжиниринга.

*4.3 Обратный инжиниринг\* (Reverse engineering)*

“Обратный” инжиниринг (часто путаемый с реинжинирингом, в том числе, в понимании SWEBOK, прим. автора) или это процесс анализа программного обеспечения с целью идентификации программных компонент и связей между ними, а также формирования представления о программном обеспечении, с дальнейшей перестройкой в новой форме (уже, в процессе реинжиниринга). Обратный инжиниринг является пассивным, предполагая отсутствие деятельности по изменению или созданию нового программного обеспечения. Обычно, в результате усилий по обратному инжинирингу создаются модели вызовов (call graphs) и потоков управления (control flow graphs) на основе исходного кода системы. Один из типов обратного инжиниринга – создание новой документации на существующую систему (redocumentation). Другой из распространенных типов – восстановление дизайна системы (design recovery). К вопросам обратного инжиниринга, как и к вопросам реинжиниринга, также относятся работы по рефакторингу (см. работы Мартина Фаулера, впервые систематизировавшего и описавшего рефакторинг, прим. автора). Рефакторинг – трансформация программного обеспечения, в процессе которой программная система реорганизуется (не переписываясь) с целью улучшения структуры, без изменения поведения. Сохранение “формы” (платформы, архитектурных и технологических решений) существующей программной системы позволяет рассматривать рефакторинг как один из вариантов обратного инжиниринга.

### Реинжиниринг или оптимизация: «как есть» или «как надо»

http://www.fox-manager.com.ua/images/1_w2.gif

Существует два метода описания бизнес-процессов предприятия:

1. «**Как есть**» - в этом случае все процессы строятся так, как они функционируют на сегодняшний день, а затем постепенно на основе анализа приводятся к виду «Как надо».
2. «**Как надо**» - в этом случае процессы строятся сразу так, как они должны функционировать, то есть наиболее оптимальным способом.

Если процессы сначала описываются по схеме «как есть» и потом постепенно приводятся к состоянию «как надо», то такой подход получил название «**оптимизация бизнес-процессов**». Если же существующая схема работы отвергается, и все процессы строятся сразу «как надо», то такой подход принято называть «**реинжинирингом бизнес-процессов**»

Оба описанных способа имеют свои недостатки: в первом случае, Вы выполняете лишнюю работу, описывая процессы по несколько раз, во втором случае, построение процессной модели осложняется тем, что очень сложно найти специалиста, который бы одинаково хорошо знал, как правильно должны выполняться абсолютно все функции предприятия.

С помощью программы [Fox Manager ФМ](http://www.fox-manager.com.ua/fox-manager.html) можно воспользоваться любым из описанных выше способов описания бизнес-процессов. На практике более удобно применять комбинированный подход: наиболее грубые ошибки, такие, как, например, дублирование ответственности устраняются сразу, в остальном процессы строятся «как есть» и лишь потом оптимизируются, руководствуясь мнениями специалистов и встроенными в программу инструментами.

Не нужно также забывать о третьем варианте использования программы: реструктуризации деятельности уже существующего предприятия. Причин для реструктуризации может быть несколько: закрытие или открытие новых направлений деятельности, слияние с другой компанией, необходимость кардинальных изменений в деятельности компании. При планировании будущей реструктуризации компании очень важно заранее просчитать загрузку людей, определить перечень новых вакансий и распределить ответственность за новые функции среди уже имеющегося персонала.

Какова бы ни была причина, побудившая Вас однажды прибегнуть к построению процессной модели, необходимо помнить, что наилучших результатов можно достичь лишь постоянно совершенствуя и дорабатывая построенную модель предприятия, оценивая эффективность выполнения текущих процессов.

## Типы обратного инжиниринга: создание новой документации на существующую систему; восстановление дизайна системы

Один из типов обратного инжиниринга – создание новой документации на существующую систему (redocumentation). Другой из распространенных типов – восстановление дизайна системы (design recovery).

К вопросам обратного инжиниринга, как и к вопросам реинжиниринга, также относятся работы по рефакторингу (см. работы Мартина Фаулера, впервые систематизировавшего и описавшего рефакторинг, *прим. автора*). *Рефакторинг* – трансформация программного обеспечения, в процессе которой программная система реорганизуется (не переписываясь) с целью улучшения структуры, без изменения поведения. Сохранение “формы” (платформы, архитектурных и технологических решений) существующей программной системы позволяет рассматривать рефакторинг как один из вариантов обратного инжиниринга.