

Software Engineering

Лекция 12: Жизнь после релиза

Тимофей Брыксин

timofey.bryksin@gmail.com

“Законы Лемана”

1. Непрерывность модернизации
 - для эксплуатируемых программ модернизация -- необходимость
2. Возрастающая сложность
 - с каждой модификацией системы становятся всё более сложными внутри
3. Эволюция больших систем
 - большие системы имеют собственную динамику изменений
4. Организационная стабильность
 - крупные проекты действуют в режиме “насыщения”
5. Стабильность количества изменений
 - количество изменений в каждой версии примерно одинаково
6. Ухудшение качества
 - качество не поддерживаемых программ воспринимается как ухудшающееся
7. Система обратной связи
 - процесс разработки должны постоянно анализироваться и улучшаться

Унаследованные (legacy) системы

- Давно эксплуатируемые системы, всё ещё приносящие пользу
- Проблемы повторной реализации
 - ТЗ неполно или утеряно
 - сильная привязка к бизнес-процессам
 - недокументированные особенности
 - новый проект -- дорого и рискованно
 - перебучение пользователей
- Особенности поддержки
 - разные команды разработки
 - устаревшие языки и технологии
 - устаревшее аппаратное обеспечение
 - неактуальная (отсутствующая) документация и/или код
 - неоднородная архитектура
 - неконсистентность данных
 - ...

Перспективы legacy-систем



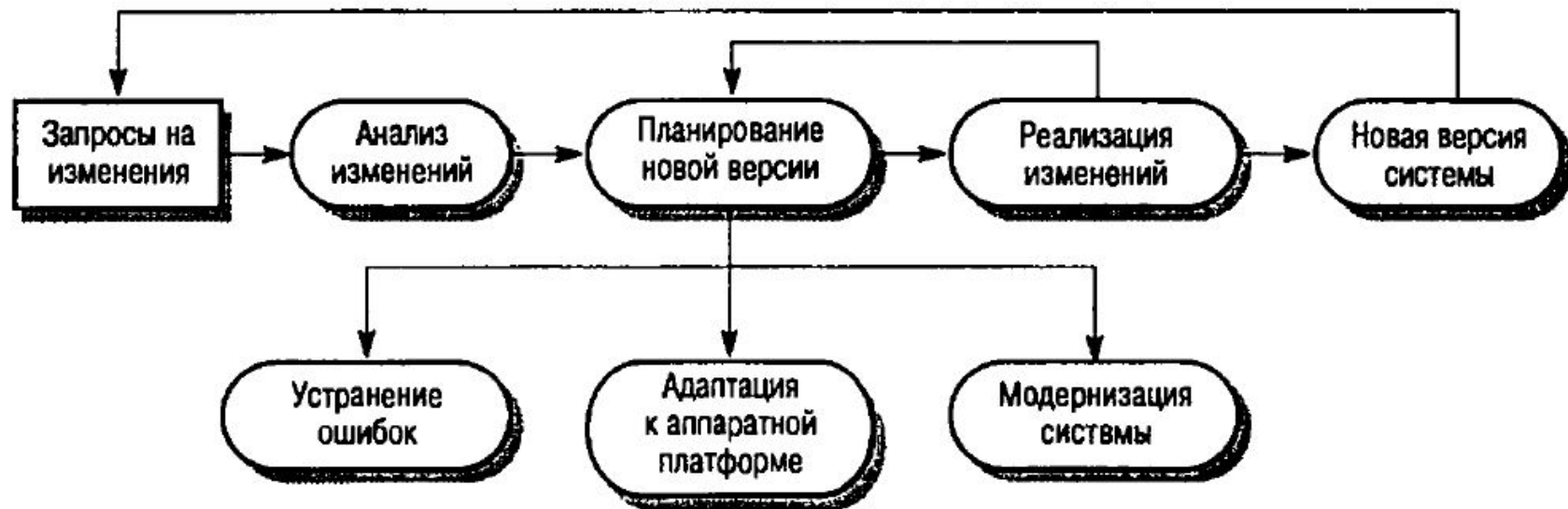
Сопровождение

- Процесс изменения системы после поставки её заказчику
- Виды сопровождения
 - корректирующее (~15-20%)
 - адаптирующее (~15-20%)
 - совершенствующее (~60%)
 - профилактическое (~5-10%)

Факторы стоимости сопровождения

- Стабильность команды разработчиков
- Ответственность согласно контракту
- Квалификация специалистов
- Возраст и структура программы

Процесс сопровождения



Прогнозирование сопровождения

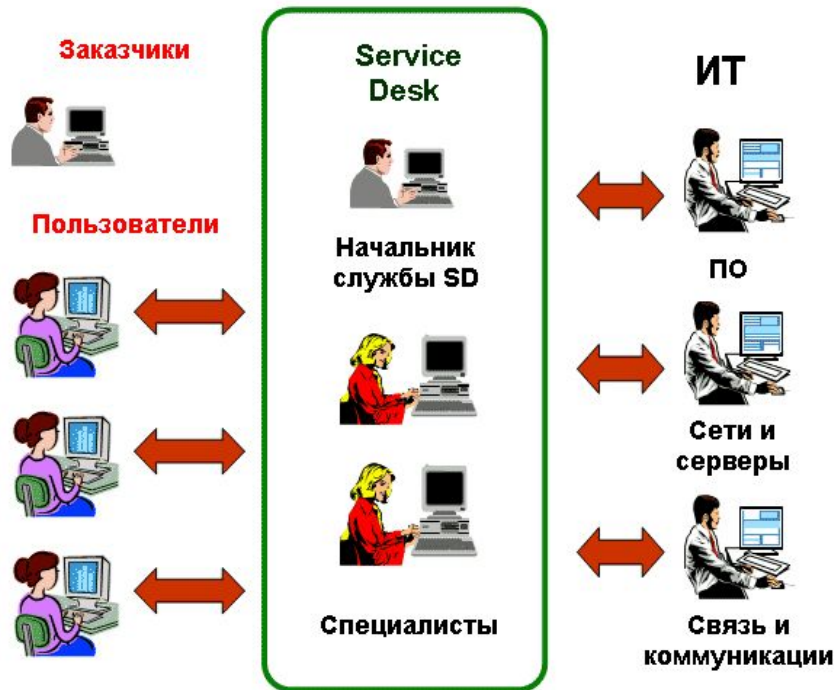
- Критерии оценки
 - количество и сложность системных интерфейсов
 - количество изменяемых системных требований
 - бизнес-процессы, в которых используется данная система
- Полезные метрики
 - количество запросов на корректировку системы
 - количество корректировок, которые затронули каждый модуль
 - среднее время, потраченное на анализ
 - среднее время, потраченное на реализацию изменений
 - количество незавершенных запросов на изменения

Личные качества хорошего мейнтейнера

- Гибкость в работе
- Творческий подход
- Широкий профессиональный кругозор
- Хорошая память
- Терпение
 - работа с заказчиком
 - работа с кодом
- Самостоятельность
- Ответственность и самокритичность
- Психологическая устойчивость

Техническая поддержка

- Виды тех. поддержки
 - по необходимости
 - фиксированный объём работ
 - тех. поддержка на заданный срок
 - сопровождение продукта
- Многоуровневая структура
- Средства коммуникаций
- CRM-система



Как работать с legacy-кодом

- Много-много обратного инжиниринга (reverse engineering)
 - отладчик
 - иерархия вызовов
 - инструментализация
 - сбор и анализ требований
 - “промышленная археология”
- Обязательное документирование полученных знаний

Как работать с legacy-кодом

- Не переписывать код!
- Сохранение имеющегося стиля
- Соблюдение бизнес-интересов
- Дополнительное бонусное время на рефакторинг
- Логирование
- Тестирование, тестирование, тестирование
- Формализованная процедура релиза
- Стратегия версионирования кода
- Ревью кода
- Статические анализаторы
- “Приложение-душитель”
- Выделение (и переписывание) модулей

Реинжиниринг

- Повторная реализация legacy-системы в целях повышения удобства ее эксплуатации и сопровождения
 - повторное документирование системы
 - реорганизация и реструктуризация
 - перевод на более современные технологии
 - модификация и модернизация работы с данными
 - модернизация бизнес-процессов
- Преимущества перед созданием новой системы
 - снижение рисков
 - снижение затрат

Reengineering vs Forward engineering



Варианты реинжиниринга

- Автоматизированное преобразование исходного кода
- Автоматизированная реструктуризация программ
- Автоматизированная реструктуризация с ручными изменениями
- Реструктуризация программ и данных
- Реструктуризация и изменения в архитектуре системы

Факторы стоимости реинжиниринга

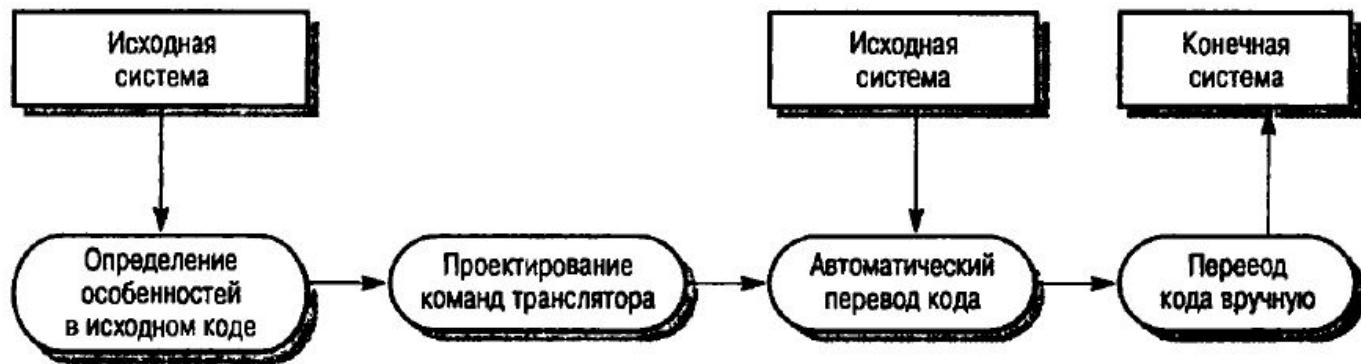
- Качество программного обеспечения
- Размер системы и объём необходимого преобразования данных
- Наличие средств поддержки процесса реинжиниринга
- Наличие необходимых специалистов
- “Похожесть” исходного и целевого стека технологий

Этапы реинжиниринга



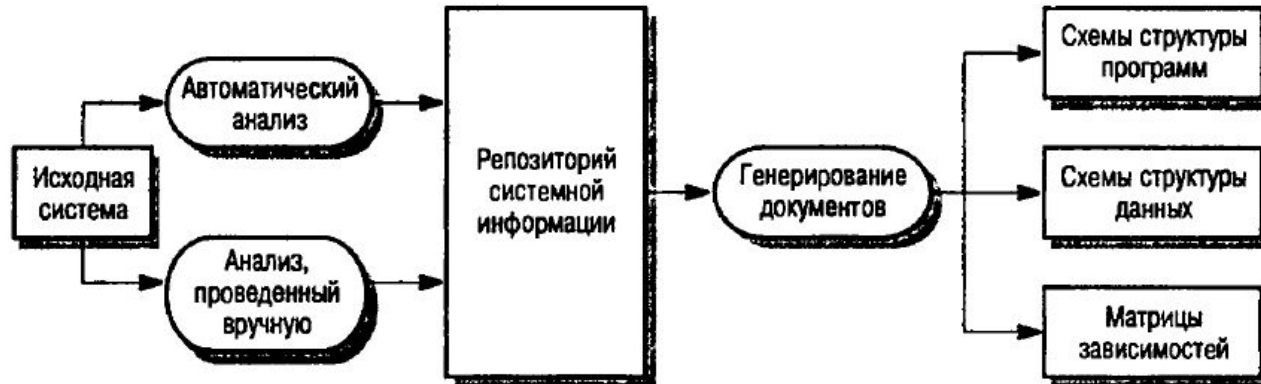
Перевод исходного кода

- Автоматическое
- Ручное



Анализ программ

- Определение архитектуры
- Спецификация логики работы системы
- Определение зависимостей между кодом и данными
- Определение внешних зависимостей



Модификация структуры программ

- Автоматические и ручные модификации
 - реструктуризация архитектуры
 - перенос на новые технологии
 - оптимизация кода
 - ...



Разбиение на модули

- Абстракции данных
- Аппаратные модули
- Функциональные модули
- Модули поддержки отдельных процессов

Модернизация работы с данными

- Методы хранения и доступа
- Структура и формат данных
- Неконсистентность данных
- Технические ограничения кода
- Конвертация имеющихся данных