

- [13] A Framework for Software Product Line Practice. Version 4.2. <http://www.sei.cmu.edu/productlines/framework.html>.
- [14] *Greenfield J., Short K.* e.a. Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools. John Wiley & Sons, 2004. 666 p.
- [15] *Попова Т. Н., Кознов Д. В., Туинова А. Е., Романовский К. Ю.* Эволюция общих активов в задачах реинжиниринга // Системное программирование. СПб.: 2004. С. 184–199.
- [16] *Гласман К. Ф., Перегудов А. Ф.* Автоматизация телевизионного вещания // Информационно-технический журнал «625». № 6. 1998. № 6. С. 5–8.
- [17] *Перегудов А. Ф.* Сетевые решения в телевизионном производстве. // Информационно-технический журнал «625». № 10. 2004. С. 10–15.
- [18] ITU Recommendation Z.100: Specification and Description Language. 2002. 206 p.
- [19] *Кознов Д. В., Перегудов А. Ф., Романовский К. Ю.* и др. Опыт использования UML при создании технической документации // Системное программирование. Вып. 1. СПб.: Изд-во СПбГУ, 2005. С. 18–35.
- [20] *Selic B., Gullekson G., Ward P. T.* Real-Time Object-Oriented Modeling. John Wiley & Sons. Inc., 1994. 525 p.
- [21] *Павлинов А. А., Кознов Д. В., Перегудов А. Ф.* и др. Комплекс средств для реализации предметно-ориентированных визуальных языков // Наст. сборник. С. 116–141.

Разработка Интернет-приложений в небольшой компании с применением product line-подхода*

Д. В. Кознов О. В. Остапенко**
dkoznov@yandex.ru olgaostap@gmail.com

В работе представлен опыт небольшой Санкт-Петербургской компании «Ленвендо» (20 человек) по наладке product line для массовой разработки Интернет-сайтов. Для решения задачи снижения издержек производства был создан типовой процесс разработки, позволяющий наладить эффективное поточное производство таких систем небольшим коллективом. Процесс был поддержан рядом инструментов автоматизации. Новизна работы заключается в том, что в настоящий момент отсутствуют исследования, суммирующие опыт наладки product line-подхода для разработки Интернет-сайтов, а также вообще нет описаний успешного опыта применения этого подхода в контексте небольших российских компаний.

Введение

В последнее время активно разрабатывается и широко используется product line-подход [10, 11, 7]. Суть подхода заключается в том, что в компании, вокруг определенных бизнес-перспектив (как правило, это некоторый сегмент рынка) создается инфраструктура

* Исследование выполнено при финансовой поддержке РФФИ (грант 05-0951/07).

** Компания «Ленвендо». 197342, ул. Сердобольская, д. 44, Санкт-Петербург, Россия.

© Д. В. Кознов, О. В. Остапенко, 2006.

для повторного использования разных активов разработки. В роли таких активов выступают, прежде всего, программные компоненты, а также различные модели (анализа, проектирования и т. д.), технологии и среды разработки (в том числе программные средства, созданные силами данной компании-разработчика), отдельные процедуры процесса разработки (например, работа с заказчиком, тестирование, работа с субподрядчиком, конфигурационное управление), квалификация персонала компании и пр. Такое повторное использование выгодно отличается от повторного использования «вообще», поскольку, во-первых, ориентировано на особенности и потребности именно этой компании, во-вторых, используется для реализации хорошо определенного класса задач, за решение которых компания берется в рамках данного product line, в-третьих, поддерживается специальными правилами и практиками, налаженными и развернутыми в компании.

Небольшие software-компании (далее — просто небольшие компании) очень разнолики, что существенно затрудняет разработку для них рекомендаций, методов, моделей и пр. Возникают даже трудности в определении того, какую компанию следует считать небольшой [17]. Исследования в этой области часто носят национальный характер — частично из-за того, что маленькие компании в большей степени, чем большие, обладают национальными особенностями, частично из-за того, что малый бизнес во многих странах является предметом особой опеки государства. Относительно product line есть исследования и предложения для небольших китайских [9], а также немецких компаний [14, 8, 20]. Работы по продвижению product line среди последних поддерживаются институтом экспериментальной программной инженерии города Фраунхофа, Германия (Fraunhofer Institute for Experimental Software Engineering, IESE) [16, 14]. В России подобных исследований пока не проводилось.

В данной работе представлен опыт наладки product line в небольшой Санкт-Петербургской компании «Ленвендо» для разработки небольших Интернет-сайтов (до 2 месяцев работы и 360 человеко-часов). Разработка сайтов на основе стандартных (распространяемых на рынке в виде готовых продуктов) CMS-систем позволила решить задачу повторного использования программных компонент. Поэтому мы сосредоточились на создании типового процесса разработки сайта, а также реализовали собственные и адаптировали ряд стандартных инструментов, максимально автоматизирующих

этот процесс. Внедрение product line позволило нам более точно оценивать трудозатраты на разработку сайтов (точность оценок повысилась до 30%), повысить рентабельность разработки сайтов на 21%, а также наладить прибыльное производство очень маленьких сайтов (далее мы называем их шаблонными).

Данный опыт интересен тем, что наша ситуация существенно отличается от тех, которые исследовались в [14, 8, 20]. В последнем случае product line строился для разработки существенно более сложных систем. Соответственно, количество систем в таких product lines было существенно меньше, чем в нашем случае, а решаемые задачи связаны, например, с созданием единой архитектуры (generic architecture) для всех систем семейства, обеспечением повторного использования функциональности, реализованной в прежних системах (с разрешением трудностей интеграции разных платформ, созданием хороших «оберток» (wrappers) и т. д.). Мы же с помощью product line решали в основном задачу наладки эффективного процесса массовой разработки систем.

1. Предпосылки

1.1. Product line-подход

Семейство программных продуктов — это не просто ряд систем, разрабатываемых в одной компании и как-то использующих общие активы разработки. Повторное использование в рамках семейства хорошо организовано, существует единый метод разработки целевых систем. Кроме того, создаваемые системы еще совместно продвигаются на рынке — т. е. компания предлагает целый спектр систем определенного вида¹. Именно в таком случае часто оказывается, что невыгодный бизнес по созданию отдельных, единичных систем вдруг при совместной их разработке и продвижении начинает приносить прибыль.

Теперь рассмотрим подробнее, что может выступать в роли повторно используемых активов. Согласно [3, 7], ими могут быть:

¹Например, в [5] представлена линейка продуктов для реинжиниринга устаревших приложений, созданных с помощью языков Cobol, PL/1, Adabas/Natural. Поддерживаются различные диалекты этих языков. Предлагаются сервисы как по изучению устаревших систем, так и по их трансформации на новые платформы. Эта линейка продуктов ориентирована на крупные организации в Европе и Америке, которые имеют большие информационные системы, созданные 20–40 лет назад с использованием этих языков.

1. *Требования.* Существенная часть требований к продуктам семейства является общей, что упрощает процесс разработки требований (requirement engineering) очередной системы семейства.
2. *Архитектура.* В рамках семейства продуктов создается архитектура типовой системы. Архитектура конкретной системы создается на ее основе и тем самым существенно экономятся ресурсы на проектирование.
3. *Компоненты.* Общая для всех представителей семейства функциональность реализуется в виде повторно используемых программных компонент, что значительно упрощает разработку отдельных систем.
4. *Различные модели* — анализа, проектирования, результаты анализа производительности и т. д. также повторно используются при создании продуктов данного семейства.
5. *Средства разработки.* В рамках семейства продуктов формируется общая технологическая среда разработки ПО — среды разработки (IDEs — Integrated Development Environments), системы управления базами данных (СУБД), средства поддержки планирования, тестирования, конфигурационного управления и т. д. Кроме того, могут создаваться специальные программные средства, специализированные для данного семейства (например, кодогенераторы по визуальным моделям).
6. *Процессы.* Здесь речь идет о повторном использовании процедур процесса, таких как конфигурационное управление, проектный менеджмент, планирование, тестирование.
7. *Квалификация специалистов.* Поскольку системы, разрабатываемые в рамках семейства, похожи, а также существует общая инфраструктура разработки (технологии и средства автоматизации), то в такой ситуации легко «передвигать» работников из одного проекта в другой (например, при необходимости усилить какой-то проект).

В данной работе в качестве повторно используемых активов мы рассматриваем процесс и средства разработки.

1.2. Небольшие компании и product line-подход

Согласно [17], основная проблема, препятствующая внедрению различных передовых практик программной инженерии в небольшие компании, заключается в том, что их главной бизнес-задачей является выживание. Это, в частности, сильно сужает технологический кругозор таких компаний и не дает развить постоянное и непрерывное усилие, без которого невозможно внедрение никакого системного метода. Рассмотрим, какие трудности обнаруживаются и возникают при внедрении product line-подхода в небольшие компании.

1. Трудность ассигнования значительных инвестиционных ресурсов на разработку product line. Как правило, небольшие компании не имеют достаточно денежных средств, квалифицированного и свободного персонала для анализа, проектирования и реализации product line, для закупки, наладки и доработки необходимого программного обеспечения, для дополнительного обучения, для исследований, организации и выполнения пилотных проектов и т. д.
2. Большая зависимость небольших компаний от рынка: например, возможна радикальная смена приоритетов при получении прибыльного заказа (вплоть до отказа заниматься прежним направлением). Но даже в рамках выбранных направлений небольшим компаниям трудно придерживаться формальных рамок, которые неизбежно создает product line. Желания клиентов — закон для небольших компаний, даже если эти желания существенно выходят за пределы предшествующего опыта, формализованного в повторно используемых активах product line. Жесткие требования на время разработки часто не дают возможности улучшить product line (создать новые активы и правила работы с ними), и продукт разрабатывается «рядом» с product line. А гибкий учет потребностей клиента и короткое время разработки — это одно из главных конкурентных преимуществ небольших компаний перед крупными [16].
3. Огромная занятость персонала текущими задачами, что сильно препятствует компании сосредоточиться на своих технологических перспективах. Создание моделей общей архитектуры будущего продукта, анализ прошлого опыта, создание

качественных документов (а не лаконичных, с потерей контекста и смысла, обрывочных комментариев по электронной почте) — все это часто оказывается запредельным для работников небольших компаний [14]. Более того, часто дело доходит до того, что они даже не понимают, зачем это нужно.

4. Отсутствие хорошо организованных процессов. Как правило, в небольших компаниях царит «героический» хаос: все занимаются всем, и большинство проектов, следуя классификации Йордана, являются «безнадежными» [2], т. е. их успешное завершение почти непредсказуемо и зависит от удачи, требует сверхусилий команды (работа по ночам, в выходные и т. д.) А между тем организация повторного использования в рамках product line основывается на хорошо определенных рабочих процессах.

Посмотрим, что же все-таки заставляет говорить о перспективности product line-подхода для небольших компаний.

1. Как указывалось в [16], типичным сценарием существования небольшой компании является разработка какой-то одной идеи, воплощенной в одном продукте. Разрабатывать больше продуктов у компании не хватает ресурсов, тем более что распространение этого продукта требует, как правило, постоянного выполнения различных модификаций в соответствии с особенностями очередного заказчика. У небольшой компании не хватает «веса» распространять «коробочку», продавая лишь копии. Таким образом, product line в этом случае возникает естественно².
2. Наладка процесса разработки в какой-то момент оказывается абсолютно необходимым шагом для развития компании. Так, в [12] указывается, что этот момент наступает, когда количество сотрудников достигает 10–15 человек. Создание product

²Приведенный выше сценарий существования небольшой компании, представленный немецкими авторами, не столь распространен в России. У нас существует большое количество маленьких компаний, которые создают и распространяют именно «коробочки», не составляющие product line. А сценарий, представленный в [16], справедлив для тех российских небольших компаний, которые распространяют продукт, «выросший» из большого заказного проекта. Примерами являются компании, продающие CMS-системы, которые, как правило, рождаются на основе какого-либо большого Интернет-проекта.

line в небольшой компании может оказаться хорошим способом для ее технологического развития, поскольку ориентировано на реальные бизнес-перспективы и соединяет улучшение управления и решение технических задач. Как показывает опыт, такое объединение может давать хорошие результаты, воодушевляя разработчиков³. Так, например, в [6, 13] описана наладка тестирования и конфигурационного управления в департаменте реинжиниринга ЗАО «ЛАНИТ–ТЕРКОМ», сопровождавшаяся проектированием, созданием и поддержкой интересных и нетривиальных программных продуктов. В результате, к наладке процесса были привлечены лучшие специалисты департамента, и неформальный рейтинг этой деятельности среди работников департамента был очень высоким.

3. При выполнении небольшими компаниями потока небольших заказов (например, в области разработки Интернет-сайтов) возникает реальная потребность в минимизации издержек, более точной оценке себестоимости проектов и т. д. Организация обработки большого количества заказов небольшим количеством разработчиков оказывается заманчивой перспективой для небольших компаний, которые часто испытывают трудности с расширением своего состава (недостаток финансов, коллектив энтузиастов, трудно принимающий к себе чужих, и т. д.).

Для нас основной мотивацией внедрения product line были причины, изложенные в пункте 3.

1.3. CMS-системы в разработке сайтов

Было замечено, что большинство Интернет-сайтов достаточно похоже, т. е. содержат одинаковые разделы, которые должны поддерживать стандартную функциональность, например, новостная лента, форум, заказ услуги или товара. Кроме того, современные сайты весьма динамичны: частая смена содержания многих страниц (например, цены на услуги компаний, меню ресторанов и т. д.),

³В противоположность этому, например, в СММ только на третьем уровне зрелости процесса предполагается введение software engineering-практик, а до этого считается, что необходимо совершенствовать только систему менеджмента [19].

необходимость быстро добавлять типовые разделы — например, новости в раздел новостей. Наконец, для Интернет-сайтов остро стоят вопросы производительности, безопасности, гибкость разделения прав доступа и пр.

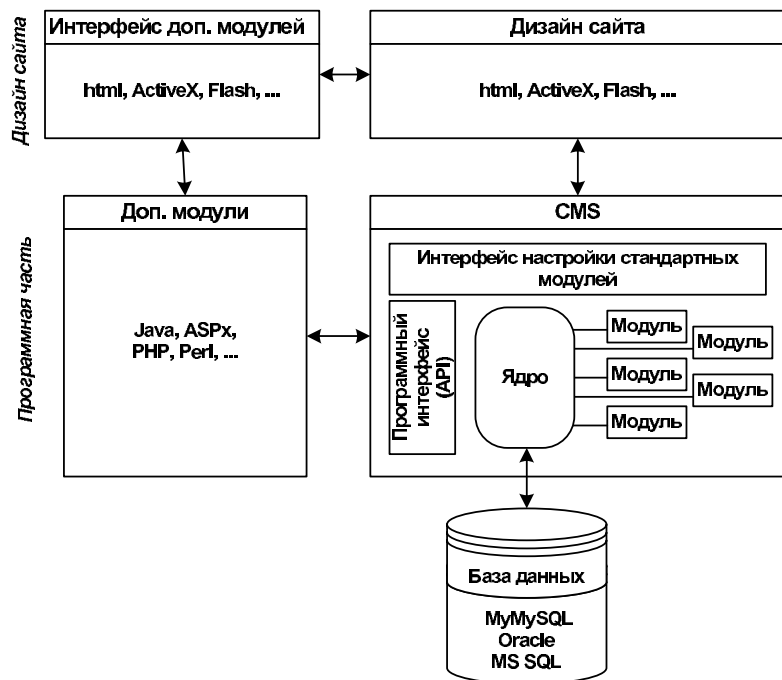


Рис. 1. Типовая архитектура Интернет-сайта на основе CMS-системы

Для решения всех этих, ставших уже классическими, задач, существуют CMS-системы (Content Management Systems). Такие системы являются (а) средой разработки, позволяя создателям сайтов пользоваться готовой, реализованной функциональностью, лишь дополняя и настраивая ее; (б) «движком» сайта, поддерживая такие функции как обеспечение безопасности, распределение прав доступа, решение вопросов производительности; (в) средством предоставления пользователям сайта ряда важных услуг, таких как легкость обновления и сопровождения сайта, сбор статистических

данных о его посещаемости и т. д. Типовая архитектура Интернет-сайта, созданного на основе CMS-системы, представлена на рис. 4⁴.

Мы использовали одну из самых зрелых отечественных CMS-систем под названием «Битрикс»⁵.

2. Классификация Интернет-сайтов

Мы предлагаем следующую классификацию Интернет-сайтов, основанную на задачах, которые выполняют сайты, и затратах на их разработку.

1. *Шаблонный сайт* предназначен для предоставления через Интернет общей информации о деятельности компании или частного лица. Сайт имеет шаблонный (уже готовый заранее) дизайн, разрабатывается за 1–2 дня (20–30 человеко-часов). Этот вид сайтов содержит лишь простые варианты стандартных разделов, поддерживаемых базовой CMS-системой.
2. *Сайт-визитка* — усложненный вариант шаблонного сайта, по функциональности, обычно, не сильно отличается от последнего, но имеет индивидуальный дизайн. Функциональная часть разрабатывается за 3–7 дней (около 50 человеко-часов), разработка дизайна может занять до 20 дней (это же справедливо для всех остальных сайтов, имеющих индивидуальный дизайн). Дополнительная по сравнению с шаблонным сайтом функциональность обычно включает в себя средства навигации по сайту, каталог товаров или фотогалерею, простую форму заказа услуги или товара.
3. *Корпоративный сайт* существенно сложнее предыдущих сайтов, может иметь много индивидуальных особенностей, не укладывающихся в CMS-систему и требующих «ручного» программирования. Разрабатывается в среднем за 1–2 месяца (около 360 человеко-часов). Помимо функций сайта-визитки может поддерживать: форумы, блоги, многосайтовость, многоязыковость, многодоменность, возможности подписки на новости и рассылки сообщений, техническую поддержку, продажу рекламы, онлайн-регистрацию на семинары и пр.

⁴ Дальнейшую информацию о свойствах и характеристиках различных CMS-систем можно получить здесь: <http://www.cmsobzor.ru>, <http://www.cmslist.ru>.

⁵ Компания «Битрикс», <http://www.bitrix.ru>.

4. *Интернет-магазин* обеспечивает продажу товаров и услуг через Интернет: заключение сделок, оплату, обновление данных о наличии товаров на складе, интеграцию со стандартными и нестандартными складскими и платежными системами (1С–торговля и склад и др.). Разработка типового Интернет-магазина в среднем занимает 1–2 месяца (около 520 человеко-часов).
5. *Информационный портал* — большой сайт, включающий в себя множество различных тематических разделов и/или некоторое количество самостоятельных сайтов. Разрабатывается в среднем 2–4 недели (около 360 человеко-часов).
6. *Самостоятельный форум/блог* представляет собой отдельный сайт для общения различных специальных сообществ, обычно имеет большую посещаемость, требует постоянной доработки, оптимизации, поддержки и развития⁶. Разработка такого сайта может продолжаться все время, пока он существует.
7. *Web-интерфейс информационной системы какой либо компании* — это сайт, который обеспечивает дистанционный доступ сотрудников компании к ее информационным ресурсам. Может быть большим проектом — от нескольких месяцев до одного года и больше.

Мы остановимся на первых трех видах сайтов, называя их *небольшими Интернет-решениями*. Несмотря на малую стоимость каждого из них в отдельности, они составляют значительную долю рынка Интернет-сайтов. Для того, чтобы их производство стало прибыльным бизнесом для компании, требуется наладить поток, конвейер по их выпуску. Это, в свою очередь, требует создания отлаженного производственного процесса, поддержанного средствами автоматизации.

⁶Примерами форумов являются сайты <http://www.sql.ru>, <http://www.phpclub.ru>, <http://www.forum.ixbt.com>, блогов — <http://www.livejournal.com>, <http://www.myspace.com>.

3. Процесс разработки небольших Интернет-решений

При разработке небольших Интернет-решений главной проблемой оказываются большие транзакционные издержки⁷, особенно на этапе постановки задачи, например, затраты на уточнение требований, согласование договора, формирование технического задания, получение необходимых материалов от заказчика, оплата почтовых расходов при обмене документами, оценка трудозатрат, поиски соисполнителей и т. д. Эти издержки на небольших Интернет-проектах составляют существенно больший процент от общих работ, чем для обычных IT-проектов. Созданный нами типовой процесс нацелен, в первую очередь, на снижение транзакционных издержек. Его общая схема представлена на рис. 2.

Теперь кратко опишем типовой процесс.

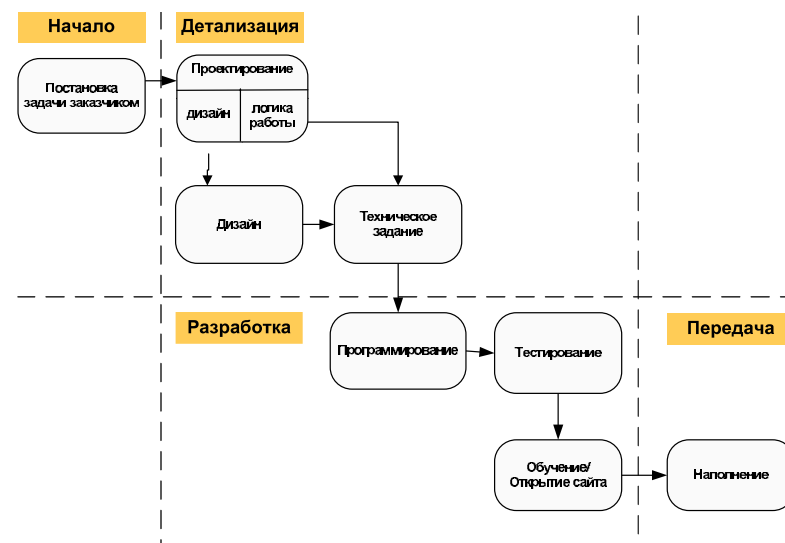


Рис. 2. Модель типowego процесса разработки небольшого Интернет-приложения

⁷Издержки по проекту, связанные с затратами компании, которые не оплачиваются заказчиком [1].

3.1. Начало

Постановка задачи заказчиком. Здесь происходит первичный контакт с заказчиком, изучение его запроса, оценка сроков и стоимости работ по детализации проекта сайта. В результате этой деятельности происходит заключение контракта. Позднее, после этапа детализации, заключается дополнительное соглашение к контракту, в котором прописывается цена и сроки на разработку самого сайта.

3.2. Детализация

Проектирование. Здесь происходит перевод (трансляция) требований заказчика в технические задачи проекта. Выделяются стандартные функции (они требуют минимальных затрат при реализации), нестандартные функции (каждая из них строго согласуется с коммерческим отделом с точки зрения затрат на их реализацию), а также определяются требования к дизайну сайта. Часть этой деятельности удается автоматизировать за счет использования стандартной анкеты на предыдущем этапе. Этот этап может опускаться, если, например, заказчик предоставил четкую постановку задачи, из которой явно следует, что решение содержит только стандартные компоненты.

Дизайн. В рамках этого вида деятельности происходит разработка внешнего вида сайта. Данный вид деятельности не выполняется в компании «Ленвендо» — компания пользуется услугами сторонних дизайн-студий, заказывая для одного проекта несколько вариантов дизайна. Заказчик выбирает понравившийся вариант, который после этого дорабатывается. Довольно часто, однако, и этот этап опускается, так как заказчик может предоставить готовый дизайн сайта, или же создается шаблонный сайт.

Техническое задание. Здесь окончательно определяются объем и содержание дальнейших работ, а также определяются сроки и цена. Для унификации процесса создания технического задания используются специальные шаблоны, которые различаются по типу создаваемого Интернет-решения. Все это согласовывается с заказчиком и коммерческим отделом компании. В результате этой деятельности создается дополнительное соглашение к контракту.

3.3. Разработка

Программирование. Здесь происходит разработка программной части Интернет-решения. При этом важными аспектами является: свобода коммуникаций между разработчиками, контроль хода проекта и качества создаваемого решения, возможность для менеджеров переключать программистов с одной задачи на другую.

Тестирование небольших Интернет-сайтов имеет следующие особенности: сложность проверки динамических компонент, а также необходимость кропотливого тестирования совместимости сайта с разными видами Web-браузеров.

Обучение/открытие сайта. Здесь происходит размещение созданного сайта на хостинге заказчика, обучение контент-менеджеров заказчика сопровождению сайта, подписание акта приемки/сдачи работ.

3.4. Передача

Наполнение. Здесь происходит передача сайта клиенту и наполнение сайта информацией (как правило, последнее делают контент-менеджеры заказчика).

3.5. Особенности модели процесса

Модель процесса имеет два уровня декомпозиции: каждый вид деятельности, представленный на рис. 2, подробно описан в виде отдельной диаграммы с обозначением всех ролей, средств автоматизации и документов (создаваемых и используемых). Применение визуального моделирования для формализации типового процесса позволило создавать точные спецификации и облегчило его обсуждение со специалистами компании, а также исправление и доводку.

Представленная модель имеет следующие особенности:

1. *Водопадность.* С одной стороны, итерации, возвраты и уточнения существенно повышают стоимость разработки [18] (тем не менее в отдельных местах нашего процесса итерации допускаются — например, при согласовании с заказчиком дизайна, но и там тщательно отслеживается, чтобы итераций не было слишком много). С другой стороны, в нашем случае разные проекты очень похожи друг на друга, так что неточно-

сти, индивидуальные особенности и связанные с этим риски разработки можно свести к минимуму за счет хорошей организации процесса. Кроме того, линейная последовательность действий позволяет конвейеризовать разработку — т. е. параллельно выполнять несколько проектов в строго установленном порядке.

2. *Масштабируемость.* В конкретных проектах некоторые виды деятельности могут отсутствовать. В частности, в разработке шаблонных сайтов отсутствует фаза *детализация*, а фаза *начало* сведена к минимуму. В других случаях часть работ заказчик может выполнять самостоятельно, например, дизайн, составление технического задания и пр. Модель обеспечивает «бесповное» удаление ненужных работ, требуя только наличия результатов их выполнения в виде соответствующих документов.

4. Средства автоматизации

На рис. 3 представлена схема взаимодействия всех инструментов, используемых нами при управлении разработкой небольших Интернет-решений.

Остановимся подробнее на этих инструментах.

4.1. Интернет-анкета

Инструмент представляет собой многоуровневую анкету, расположенную на сайте компании «Ленвендо», которую заполняет потенциальный клиент. На основе этой анкеты формируется заказ Интернет-решения. Анкета используется для первичной постановки задачи на дизайн и разработку бизнес-логики, для оценки сроков проектирования, разработки дизайна, и главное — в качестве входа транслятора стандартных требований. Как показала практика, почти все требования к небольшим Интернет-решениям одинаковы для большинства проектов и, кроме того, заказчик их хорошо понимает (то есть не нужно их из него «вытягивать», применяя различные специальные техники выделения и формализации требований, обзор которых можно найти, например, в [20]). Мы создали максимально подробный список практически всех возможных функций, которыми может обладать небольшое Интернет-решение, а также

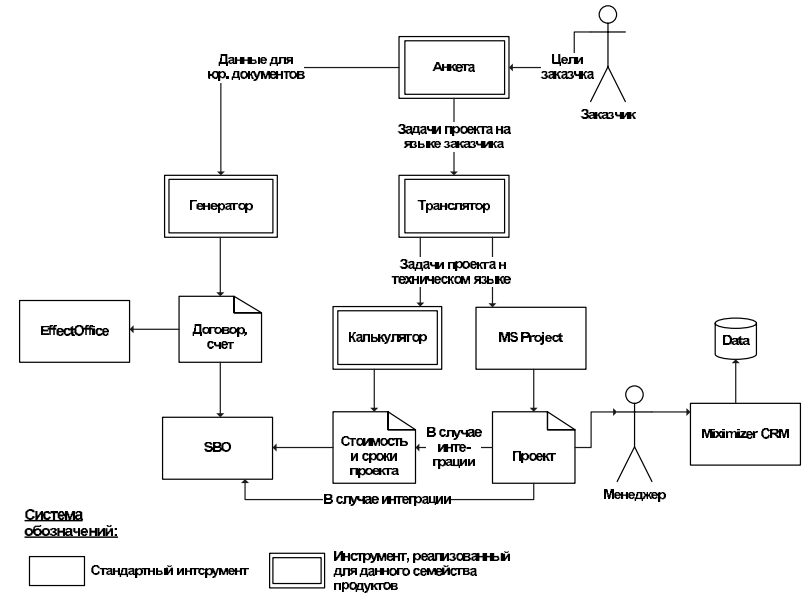


Рис. 3. Схема взаимодействия инструментов управления разработкой продуктов семейства

список соответствующих вопросов к заказчику. Получилось около 40 вопросов и порядка 100 ответов.

4.2. Калькулятор сроков и стоимости проекта

Инструмент предназначен для приблизительной оценки стоимости разработки сайта заказчиком. Внешне он представляет собой просто кнопку «Рассчитать» на сайте компании под анкетой рядом с кнопкой «Отправить заказ менеджеру» и окошко с суммой, которое появляется при нажатии на эту кнопку. Информация для расчета стоимости берется из специальной базы данных, например, «Размещение навигационной цепочки: 0,5 часа». Получившееся время работы умножается на стоимость этой единицы измерения.

4.3. Генератор юридических документов

Этот инструмент формирует документы в соответствии со стандартами, принятыми в компании, но с учетом индивидуальных параметров заказчика (суммы договора, реквизитов, дат начала и окончания работ, контактов и т. д.). Использование генератора существенно сокращает время на составление документов, снижает вероятность ошибок, обеспечивает автоматическую передачу документов заказчику. Если заказчик, увидев стоимость разработки своего проекта, согласен с ней, а также согласен с условиями работы, о которых он может узнать в специальном разделе сайта компании, можно предложить ему заключить договор-оферту⁸ без привлечения менеджеров компании.

4.4. Транслятор ответов заказчика

Задача этого инструмента — автоматизировать перевод требований к сайту, согласованных с заказчиков, в перечень задач программистам. Как мы уже писали, из-за схожести небольших Интернет-решений, их разработка тоже довольно стандартна. При этом постановка задачи может быть самой неожиданной, например: «Сделайте мне небольшой сайт в несколько страничек», «Настройте самую младшую редакцию CMS», «Сделайте так, чтобы информацию о моих услугах можно было найти в Интернете» и т. п. Все эти просьбы для менеджера проекта звучат одинаково: «Проведите интеграцию выбранного мной дизайнера с младшей версией системы управления, включив в нее все стандартные возможности и настроив стандартные группы пользователей». Эту постановку уже можно передать разработчикам, представив ее в пакете Microsoft Project. Менеджер в данном случае является «переводчиком» (транслятором) с языка заказчика на язык программиста. Конечно, некоторые задачи в проекте могут быть индивидуальными (например, создать интерактивную игру для посетителей на время рекламной акции), и для таких задач в модели процесса предусмотрено действие «транслятор индивидуаль-

⁸Офертой признается адресованное одному или нескольким конкретным лицам предложение, которое достаточно определенно и выражает намерение лица, сделавшего предложение, считать себя заключившим договор с адресатом, которым будет принято предложение. Оферта должна содержать существенные условия договора (Гражданский Кодекс Российской Федерации. Статья 435, Оферта).

ных особенностей проекта». Но большинство задач все-таки стандартны.

Вход транслятора обеспечивается следующим образом. Заказчик формулирует свои потребности на собственном языке (А-язык). Потом он находит способ выразить свои требования на языке Интернет-анкеты (В-язык), потом эти требования автоматически переводятся на язык программиста (С-язык). По сути, транслятор — это таблица в базе данных, в которой записаны варианты требований на В-языке и они же — на С-языке. Примеры представлены на рис. 4.

А-язык (язык заказчика)	В-язык (язык Интернет-анкеты)	С-язык (язык программиста)
Хочу вставить свой логотип, когда хочу	Обеспечить возможность смены логотипа	В области шаблона, предназначенном для размещения логотипа, разместить включаемую область такого же размера с возможностью размещения графических и текстовых элементов.
Хочу, чтобы можно было менять логотип		
Хочу вместо логотипа иногда размещать свою фотографию		
Хочу мультик на сайте	Включить анимацию на страницах сайта	Разместить флэш-элемент. В случае, если в браузере отключена опция показа флэш-элементов, показывать на этом же месте статический графический элемент. В случае, если у клиент-сервера не установлен флэш-проигрыватель, вывести предложение установить его.

Рис. 4. Две фазы перевода с языка заказчика на язык программиста

4.5. Стандартные инструменты

Опишем теперь стандартные инструменты управления разработкой продуктов нашего семейства.

Maximizer CRM⁹. Это CRM-система (Client Relationship Management System), предназначенная для координации работы с заказчиками: хранение данных о компаниях, контактных лицах, а также самих контактов и т. д. В нашем бизнес-процессе инструмент используется в деятельности под названием «Постановка задачи заказчиком».

⁹<http://www.maximizer.com>.

EffectOffice¹⁰. Это система документооборота, используемая в компании «Ленвендо», обеспечивающая хранилище данных, документов, шаблонов документов, любых файлов и рабочих продуктов, к которым необходимо иметь доступ во время реализации проекта (почти полностью покрывает улучшение двух рассматриваемых видов деятельности — документооборот и хранилище данных).

Microsoft Project. Кроме стандартных функций автоматизации планирования, создан шаблон типового проекта, состоящий из действий, определенных общим технологическим процессом. Данный шаблон детализируется информацией, поступающей от транслятора ответов заказчика. Виды деятельности процесса — это укрупненные задачи стандартного проекта, а действия — обычные задачи. Действия можно детализировать и до третьего уровня при необходимости, причем задачи на третьем уровне должны совпадать с результатами работы транслятора, т.е. должны выбираться из словаря *C* — языка задач для каждого конкретного проекта.

Кроме того, все инструментальные средства процесса интегрируются с продуктом **SBO**¹¹, обеспечивающим комплексную автоматизацию работы фирмы, т.е. управление бюджетированием, бухгалтер, управление персоналом, зарплатой и т.д.

5. Что получилось

Как уже указывалось, одной из главных проблем организации product line в компаниях (особенно небольших), является трудность долгосрочного планирования. Не понятно, останется ли рынок достаточное время стабильным в тех рамках, которые зафиксированы в product line, для того, чтобы затраты на создание product line окупились и принесли прибыль. Относительно рынка Интернет-сайтов в Санкт-Петербурге следует сказать следующее. Рынок рекламы в Интернете, по данным компании «Romir Monitoring»¹², растет в среднем на 70% в год, и его рост будет только увеличиваться ближайшие пять лет. Количество пользователей Интернета в Санкт-Петербурге растет на 44% в год, по данным исследова-

¹⁰<http://www.garant.spb.ru>.

¹¹SAP Business One (<http://www.sap.com>) — ERP-система компании SAP, ориентированная на малый и средний бизнес.

¹²<http://www.rmh.ru>.

тельского центра «Гортис»¹³, количество сайтов увеличивается на 49% в год. Спрос на услуги компании «Ленвендо» по изготовлению сайтов неуклонно возрастает. Мы рассчитаем заниматься данной темой не менее чем 4 года.

Product line разрабатывался и внедрялся 6 месяцев. Планируемая окупаемость затрат на его разработку и внедрение — 5,5 месяцев (50 шаблонных сайтов или 30 сайтов-визиток, или 15 корпоративных сайтов). На данный момент в рамках созданного product line уже разработано 8 шаблонных сайтов, 5 сайтов-визиток и 3 корпоративных сайта, что окупило создание product line приблизительно на 50%.

В создании product line участвовало 3 человека — менеджер, технический руководитель компании, разработчик. Из-за сильной вовлеченности этих сотрудников в основную работу компании самой большой трудностью было «оторвать» их от этого и организовать участие в работах по наладке product line. Мешало также скептическое отношение руководства к деятельности, которая не направлена непосредственно на получение прибыли, и недоверчивое отношение коллектива, нежелание учиться чему-то новому. С другой стороны, вся деятельность была организована и контролировалась сотрудником компании (менеджером), который был ее инициатором и сумел преодолеть указанные выше трудности, терпеливо объясняя текущие проблемы и описывая грядущие выгоды, воодушевляя товарищей и т.д. Сыграли свою роль и небольшие размеры компании — фактически, во многом, оказалось достаточно создать в целом позитивную атмосферу вокруг данной инициативы и после этого решать уже отдельные проблемы и трудности.

Product line внедрялся по легковесной схеме (lightweight strategy) [6], т.е. отдельные активы тут же после их создания опробовались на практике. Их успешность сильно повышала рейтинг данной деятельности в компании.

Были достигнуты следующие экономические выгоды от внедрения product line: (а) оказалось, что компания неправильно оценивает себестоимость разработки небольших Интернет-решений — реальная себестоимость оказалась выше на 30%, что стало заметно после формализации процесса; наличие хорошо специфицированного процесса позволило аргументировано объяснить заказчикам

¹³<http://www.gortis.info>.

причину увеличения стоимости проектов; (б) в результате внедрения типового процесса и средств автоматизации удалось уменьшить издержки компании на 21%; (в) product line позволил компании наладить выпуск шаблонных сайтов; вне product line разработка этих сайтов была нерентабельна.

В данный момент в рамках product line работает 7 человек, в ближайшее время не планируется расширять состав этого коллектива, а использовать прибыль от деятельности по разработке Интернет-решений на развитие других направлений компании.

Заключение

Налаживая product line в российской компании, мы столкнулись в общем-то с теми же проблемами, что описаны в работах немецких коллег [7, 22, 3, 4]. Одно из главных отличий от исследований в Германии, как, впрочем, и от исследований этой тематики в Китае [2], является отсутствие методологической, стратегической и организационной поддержки подобных исследований и экспериментов российскими учеными и правительством. Участие специалистов Санкт-Петербургского государственного университета в данном проекте было не столь значительно, как могло бы быть. Пока не планируется дальнейших масштабных исследований и экспериментов в этой области, подобных тем, что проводились в Германии [22] (и которые опирались еще и на общеевропейские программы). Российским компаниям не опереться на какие-то общегосударственные стандарты и последовательную политику, проводимую правительством, как, например, в Китае. Хочется верить, что это временные явления и, в частности, данное исследование не окажется одиноким.

В заключение хочется выразить признательность Виталию Гаврилову, без чьих советов и профессионального участия данное исследование не могло состояться, а также поблагодарить работников компании «Ленвендо» за участие в проекте и долготерпение. Мы благодарим также Александра Иванова, чьи порой весьма сварливые замечания позволили существенно улучшить данную статью.

Список литературы

- [1] Доналд Э. Дж., Линдсей Д. Макроэкономика (пер. с англ). СПб.: Литера полус, 1994. 405 с.
- [2] Йордон Э. Путь камикадзе. Изд. 2-е, дополненное (пер. с англ). М.: Изд-во Лори, 2005. 304 с.
- [3] Кознов Д. В., Перегудов А. Ф., Бугайченко Д. Ю. и др. Визуальная среда проектирования систем телевизионного вещания // Наст. сборник. С. 142–168.
- [4] Кулямин В. В., Пакулин Н. В., Петренко О. Л. и др. Формализация требований на практике. Препринт ИСИ РАН. 2006. 82 с.
- [5] Попова Т. Н., Кознов Д. В., Туунова А. Е., Романовский К. Ю. Эволюция общих активов в задачах реинжиниринга // Системное программирование. СПб., 2004. С. 184–199.
- [6] Попова Т. Н., Туунова А. Е., Кожарина Л. А. Опыт использования визуального представления информации для контроля качества в процессе создания программного обеспечения // Системное программирование. СПб., 2004. С. 225–249.
- [7] A Framework for Software Product Line Practice. Version 4.2. — <http://www.sei.cmu.edu/productlines/framework.html>.
- [8] Gacek C., Knauber P., Schmid K., Clements P. Successful Software Product Line Development in a Small Organization: A Case Study // IESE-Report No. 013.01/E Version 1.0 March 15, 2001. 41 p.
- [9] Chao Li, Huaizhang Li, Mingshu Li. A Software Factory Model Based on ISO9000 and CMM for Chinese Small Organizations // Proc. 2nd Asia-Pacific Conference. 2001. P. 288–292.
- [10] Czarnecki K., Eisenecker U. W. Generative programming: Methods, Tools, and Applications. Addison-Wesley, 2000. 832 p.
- [11] Greenfield J., Short K. et al. Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools. John Wiley & Sons, 2004. 666 p.
- [12] Horvat R. V., Rozman I., Gyorkos J. Managing the Complexity of SPI in Small Companies // Software process improvement and practice. N 5. 2000. P. 45–54.
- [13] Kiyayev V., Sobolev I., Terekhov A. A., Fedotov B. Formalization and Automation of Global Software Development Processes // Proc. 2nd International Workshop «New Models of Business. Managerial Aspects and Enabling Technology». 2002. P. 150–160.

- [14] *Knauber P., Muthig D., Schmid K., Widen T.* Applying Product Line Concepts in Small and Medium-Sized Companies // IEEE Software. September/October, 2000. P. 88–95.
- [15] *McGregor J. D., Northrop L. M., Jarrad S., Pohl K.* Initiating Software Product Lines // IEEE Software. July/August, 2002. P. 24–27.
- [16] *Muthig D., Bayer J.* Helping Small and Medium-Sized Enterprises In Moving Towards Software Product Lines // Proc. Software Product Lines: Economics, Architectures, and Implications Workshop N 15 at 22nd International Conference on Software Engineering (ICSE). 2000. P. 137–140.
- [17] *Paulk M.* Using the Software CMM in Small Organizations // Proc. 16th Annual Pacific Northwest Software Quality Conference. 1998. P. 350–362.
- [18] *Royce W. W.* Managing the Development of Large Software Systems // Proc. IEEE WESCON. 1970. P. 1–9.
- [19] Software Engineering Institute. Capability Maturity Model—Integrated for System Engineering // Software Engineering. V 1.0. CMU/SEI-00-TR-018. Carnegie Mellon University, 2000.
- [20] *Verlage M., Kiesgen T.* Five years of product line engineering in a small company // Proc. 27th international conference on Software engineering. 2005. P. 534–543.

Метод разработки документации семейств программных продуктов

К. Ю. Романовский
kr@tepkom.ru

В работе представлен новый метод разработки документации семейств программных продуктов. Метод включает в себя процесс разработки документации, язык DRL (Documentation Reuse Language) для проектирования пакета документации и разметки текстов, а также архитектуру инструментальных средств для поддержки метода. Реализован прототип пакета инструментальных средств. Метод апробирован при создании документации в проекте разработки ПО семейства систем управления телевидением.

Введение

Разработка семейств программных продуктов — популярный подход к разработке ПО, позволяющий эффективно реализовать повторное использование различных активов процесса разработки ПО [7, 6, 11]. Семейство программных продуктов — это набор совместно разрабатываемых программных продуктов, относящихся к одной предметной области и разделяющих общую функциональность. Традиционные методы разработки семейств программных продуктов поддерживают повторное использование разнообразных общих активов — программных компонент, архитектуры, тестовых сценариев, требований и даже языков, ориентированных на предметную область (Domain—Specific Languages) [4, 17, 16, 14]. В ряде работ предлагается использовать визуальные модели для облегчения управления общими активами — это диаграммы возможностей

© К. Ю. Романовский, 2006.