

Бондаренко С.П., Акинфина М.А.

СИСТЕМА АНАЛИЗА И ВЫБОРА МЕТОДОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Выбор проектных методологий, моделей жизненного цикла разработки программного обеспечения, метрик проектов и других инструментальных средств управления программным обеспечением представляет собой сложную задачу для компаний-разработчиков программного обеспечения и проектных команд. На практике чаще всего рассматривается ограниченный набор проектных методологий и к решению проблемы выбора подходят без учета специфики компании и проекта.

Применение определенной методологии разработки программного обеспечения позволяет в типичном цикле разработки программ устранять разногласия и отсутствие связей между членами команды разработчиков и конечными пользователями, подключать новых людей к проекту на любой стадии разработки, создать пакет документации, позволяющий в дальнейшем упростить неизбежные сопровождение и дополнения программного продукта, свести к минимуму затраты на доработку, модификацию и сопровождение продукта

В силу вышесказанного в настоящий момент актуальной задачей является разработка методики выбора определенной методологии, которая будет наиболее лучшим образом удовлетворять требованиям различных организационных структур, специализирующихся в разработке программного обеспечения, и реализация информационной системы, автоматизирующей данный процесс. Описываемая в данной статье информационная система анализа и выбора методологии разработки программных продуктов обеспечивает следующий основной функционал.

1. Предоставляет возможность конечному пользователю на основе различных факторов, влияющих на специфику методологии разработки программного обеспечения – размер организации, команды, размер разрабатываемого программного продукта и т. д., получить рекомендации по применению различных методик и методологий.

2. Предоставляет расширенный и удобный информационный ресурс, который предназначен для повышения квалификации конечного пользователя, а так же для аккумулялирования знаний, привносящихся пользователями-специалистами.

3. Предоставляет конечному пользователю возможность конфигурации настроек находящихся в базе методологий на основе личного опыта и в зависимости от различных факторов (особенности команды разработчиков).

4. Предоставляет пользователю возможность добавления новых существующих методологий разработки программного обеспечения, а также создания своих экспериментальных методик.

Подходы к выбору методологий, моделей и стандартов зависят от предполагаемой цели использования. Обычно методологии используются для разных задач, в частности: для обоснования существующей практики управления проектами, для обоснования правильности принятых решений, как руководство, содержащее конкретные рекомендации, как набор требований, предъявляемых клиентами, партнерами, собственниками, государственными организациями.

В результате анализа методологий управления разработкой программного обеспечения были выявлены две группы методологий, которые отличаются целью их использования, историей создания и назначением. Первая группа представляет собой методологии, целью которых является успешное выполнение отдельного проекта. К этой группе относится большинство проектных методологий, и практически все адаптивные методологии. Вторая группа включает в себя методологии, обеспечивающие устойчивое функционирование компании-разработчика программного обеспечения, и нацелены на последовательное развитие компетенций. Логика успешного функционирования предполагает создание, контроль и непрерывное улучшение способностей организации к выполнению проектов, и, как следствие, успешное выполнение проектов.

Две выделенные группы отличаются не только целью использования, но и историей создания и развития, а также практикой использования. Проектные методологии представляют собой ядро теории управления разработкой программного обеспечения. К существующей классификации в зависимости от используемой в ней модели жизненного цикла (водопадные (каскадные) и итерационные методологии) добавилась более общая классификация на прогнозируемые и адаптивные методологии.

Методология включает в себя следующие предметы и темы: роли, навыки, команды разработчиков, инструментарий, техники, виды деятельности, стандарты, рабочие продукты, меры качества и система ценностей, принятых в команде разработчиков. Приведенный список не является фиксированным и может быть расширен.

Определим понятие размера, плотности и веса методологии. Под размером методологии будем понимать число элементов управления в ней,

к которым относятся поставляемые артефакты, стандарты, виды деятельности, меры качества и т.д. Под плотностью методологии будем понимать уровень детализации и связности, необходимых для ее осуществления. Более высокая плотность соответствует жесткому контролю или сильному формализму. Вес методологии определим как численную величину равную размеру, помноженному на плотность. Так же введем понятие размера проекта. Под этим термином будем понимать число людей, работающих над проектом, деятельность которых необходимо координировать. Определим объем методологии, как протяженность жизненного цикла проекта, разнообразие ролей и видов их деятельности, которые и пытается покрыть собой методология.

Существенную роль в поисках нужной методологии играет определение принципов, на основании которых она была разработана. Можно выделить четыре основных принципа.

Принцип 1. Большая по размерам методология нужна тогда, когда в проекте занято большое число разработчиков [1].

Принцип 2. Большая корректность методологии или, другими словами, большая плотность нужна в тех случаях, когда скрытые ошибки в программном продукте могут повлечь за собой значительный ущерб. Ссылаясь на [2] можно классифицировать программные системы по категориям возможного ущерба: потеря комфорта в работе, потеря несущественной суммы, потеря невозполнимой суммы, потеря жизни.

Принцип 3. Незначительное увеличение размеров или плотности методологии ведет к существенному увеличению стоимости проекта.

Приостановка работы одной команды программистов для координации с другой командой требует не только дополнительного времени, но и дополнительной концентрации ([3],[4]). Для обновления документации, относящейся к требованиям, дизайну системы и тестированию, понадобится немало времени. Этот принцип справедлив для любого проекта.

Принцип 4. Наиболее эффективная форма коммуникации – непосредственное взаимодействие, лицом к лицу.

Для проведения анализа было выбрано 20 методологий: начальные (Shlaer-Mellor, Coad-Yourdon, RDD, Booch, OMT, OSA, OOSE, Fusion); тяжеловесные (OPM, Catalysis, OPEN, RUP/USDP, EUP, FOOM), гибкие (DSDM, Scrum, XP, ASD, dX, Crystal).

Оценка методологий реализовалась в два этапа. На первом этапе начальный набор критериев применялся к имеющемуся набору методологий. На втором осуществлялась выработка окончательного набора критериев. Среди критериев можно отметить такие как поддержка основных этапов жизненного цикла разработки программного обеспечения (анализ,

проектирование, имплементация, тестирование), управление рисками, управление проектом, контроль качества, масштабируемость, конфигурируемость, гибкость.

Информационная система анализа и выбора методологии разработки программного обеспечения содержит ряд модулей, каждый из которых имеет специальное назначение, реализуя определенный процесс. К основным модулям относятся модули обеспечивающие:

- процесс получения рекомендаций по применению различных методик и методологий;
- процесс изменения характеристик методологии;
- процесс добавления и создания новых методологий;
- процесс просмотра и обновления информационного ресурса.

В соответствии с представленной структурой системы в нее могут быть добавлено любое количество новых критериев со своими собственными классами оценок. Для оптимизации работы с базой данных в системе был реализован специальный механизм (cache). Пользователи системы разбиты на группы, причем каждая группа обладает набором определенных прав. Имеются следующие группы: администраторы, эксперты, обычные пользователи. Администраторы обладают такими возможностями, как перезапуск приложения, включение/выключение функционала приложения, изменения различных графических настроек. Эксперты обладают всеми правами, которыми обладает группа администраторов, а так же, возможностями изменять характеристики методологий из базового набора, добавлять и удалять методологии. Пользователи обладают возможностью просматривать информационный ресурс, получать рекомендации в отношении использования определенной методологии, создавать конфигурационные файлы для поддерживаемого инструментария.

Литература:

1. Harrison, N. Patterns of productive software organizations / N. Harrison, J. Coplein // Bell Labs Technical Journal. – 1996. – V.2. – P. 138-145.
2. Коберн, А. Каждому проекту своя методология / А. Коберн // Humans and Technology Technical Report. – 1999. TR 99.05.
3. Cocburn, A. Surviving Object-Oriented Projects/ A. Cocburn. – NY: Addison-Wesley, 1998. – 346 p.
4. Cocburn, A. Crystal/Clear: A Human-Powered Methodology for Small Teams / A. Cocburn. – NY: Addison-Wesley, 2000. – 270 p.