

УДК 004.4^2: 658.5

**Е.А. Бахман, А.В. Кизим**

**РАЗРАБОТКА ЕДИНОЙ ПЛАТФОРМЫ УПРАВЛЕНИЯ СОЗДАНИЕМ  
И ОБСЛУЖИВАНИЕМ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ  
НЕБОЛЬШИМИ КОМАНДАМИ РАЗРАБОТЧИКОВ**

*Актуальность управления жизненным циклом программных приложений связана с необходимостью повышения эффективности совместной работы разработчиков и взаимодействия с пользователями, снижения издержек при создании и поддержке программных систем, а также стоимости конечных программных продуктов. В данной работе в качестве средства решения подобных задач предлагается разработать единую платформу управления созданием и обслуживанием программных приложений, ориентированную на небольшие команды разработчиков.*

*Разрабатываемая система позволит интегрировать существующие программные продукты в области управления жизненным циклом приложений, а также обеспечить недостающие функциональные возможности (в частности, управление требованиями), что представляет научно-техническую новизну предлагаемого решения.*

*Апробация платформы проводится на базе процессов разработки и сопровождения приложений для программно-информационной поддержки задач технического обслуживания и ремонта.*

*Разработка и поддержка программных систем; сопровождение; техническое обслуживание систем; управление жизненным циклом приложений; этапы жизненного цикла программного обеспечения.*

**E.A. Bakhman, A.V. Kizim**

**DEVELOPMENT OF A SINGLE PLATFORM FOR MANAGEMENT OF  
SOFTWARE APPLICATIONS CREATION AND MAINTENANCE BY SMALL  
DEVELOPERS TEAM**

*The urgency of application lifecycle management is assigned to the need to increase the efficiency of developer's collaborative work and user's interaction, decrease software systems creation and support costs and final software product's price. In this paper as a means of solving such problems we suggest to develop a single platform for management of software applications creation and maintenance, which is oriented on small developers teams.*

*The developed system will make it possible to integrate existing software products in the application lifecycle management field and provide the missing functionality (such as requirements management), which provides scientific and technical novelty of the proposed solution.*

*Platform's approbation is performed on the basis of the development and maintenance processes and information support of system maintenance and repair tasks.*

*Development and support of software systems; system maintenance; application lifecycle management; software lifecycle phases.*

Применение программных средств для автоматизации различных задач является важным инструментом повышения эффективности деятельности. Для того, чтобы предоставить потребителю возможность использования программной системы, необходимо обеспечить выполнение целого комплекса задач управления разработкой и обслуживания жизненного цикла программных приложений, который охватывает все этапы с момента появления потребности в программном продукте до вывода его из эксплуатации.

Поддержка разработки и сопровождения программных систем может быть осуществлена с использованием специальных методологий (Agile, MSF, RUP, ITSM и др.), международных и государственных стандартов (ГОСТ 19, ГОСТ 34,

ГОСТ Р ИСО/МЭК 12207-99, ISO/IEC 12207:1995), моделей зрелости процесса разработки (СММ, СММИ). Они охватывают большинство этапов жизненного цикла программного продукта, однако зачастую не удовлетворяют требованиям пользователей из-за сложности применения и степени формализма при описании процесса разработки. Для небольших коллективов, занимающихся созданием и сопровождением программных продуктов, наиболее перспективным и комплексным подходом в области разработки и сопровождения программных систем является ALM. Согласно [1], управление жизненным циклом приложений (Application Lifecycle Management, ALM) представляет собой объединение управления бизнес-процессами и разработки программного обеспечения (ПО), возможное благодаря инструментам, которые упрощают и интегрируют управление требованиями, проектирование, кодирование, тестирование, отслеживание и управление релизами.

Разработка ПО – довольно сложный процесс: создание программного продукта с достаточно четко определенными характеристиками, выполненное с приемлемым качеством, в рамках отведенного бюджета и в требуемый срок требует постоянной координации большого количества действий между многочисленными специалистами. Основная цель ALM – добиться предсказуемого и управляемого процесса создания ПО [2].

Комплексный подход к управлению жизненным циклом программных приложений обеспечивает решение следующих задач (рис. 1):

<b>Управление жизненным циклом приложений</b>				
Управление требованиями	Управление проектом	Управление коммуникациями	Управление конфигурацией	Управление изменениями
Тестирование	Управление ошибками	Управление сборками	Управление релизами	Формирование отчетов

*Рис. 1. Основные задачи управления жизненным циклом приложений*

При анализе существующих программных решений были рассмотрены следующие ALM-системы: Agile ALM Platform; Rally Agile ALM platform; Microsoft Visual Studio Team System; Caliber, StarTeam, Silk (Borland); Rational Requirements Composer, Rational Team Concert, Rational Quality Manager (IBM); Mingle, Go, Twist (ThoughtWorks); Seapine ALM; AccuRev SCM; DevSuite; Polarian ALM. В результате анализа выявлено, что большинство существующих программных систем являются дорогостоящими и не предназначены для использования небольшими командами разработчиков. Кроме того, с точки зрения наличия функциональных возможностей, это достаточно громоздкие системы, большинство которых востребовано лишь при разработке крупных программных проектов. Также было выявлено, что в существующих ALM-системах недостаточное внимание уделено управлению программными проектами: слабо реализованы функции управления требованиями и рисками, которые являются одними из основных, так как конечное приложение должно отвечать потребностям заказчика (бизнеса) и быть реализовано с учетом всех требований к качеству, срокам и бюджету проекта.

Актуальность проблемы влечёт за собой необходимость разработки механизмов, средств и систем управления жизненным циклом приложений, позволяющих обеспечить эффективное взаимодействие между разработчиками и заказчиком, снизить издержки при создании и поддержке программных систем, а также стоимость предлагаемого решения, что является важным аспектом для небольших команд разработчиков.

В качестве средства решения данной проблемы предлагается разработать единую программную платформу управления жизненным циклом приложений, способную объединить (интегрировать) существующие программные решения, а также обеспечить недостающие функциональные возможности (в частности, управление требованиями).

В качестве ядра разрабатываемой программной платформы выбрана система Agile ALM Platform (VersionOne) [3], которая предлагает возможности планирования и управления задачами, отслеживания ошибок, управления коммуникациями, формирования отчетов, предоставления открытого API и Java & .NET SDK. Главное преимущество выбранной системы – возможность использования бесплатных коннекторов с открытым исходным кодом, реализующих интеграцию с системами непрерывной интеграции, отслеживания ошибок (баг-трекинга), тестирования, контроля версий кода и средами разработки. Также VersionOne предлагает бесплатные расширения для управления задачами, общения внутри команды и поиска информации. В системе реализована поддержка методологии Agile для управления жизненным циклом программных приложений. Команды с численностью не более 10 пользователей могут использовать бесплатную версию Agile ALM Platform.

Так как разрабатываемая программная система ориентирована на небольшие команды разработчиков, были выделены следующие роли: руководитель, менеджер проекта, аналитик, архитектор, разработчик (инженер-программист), тестировщик, специалист по внедрению и сопровождению, пользователь, заказчик. В системе предусмотрено разграничение прав доступа для каждой из выделенных ролей пользователей.

Основные функции системы управления разработкой и сопровождения программных приложений:

- ◆ управление проектами (задачами, ресурсами);
- ◆ управление требованиями;
- ◆ управление рисками;
- ◆ управление коммуникациями;
- ◆ управление изменениями;
- ◆ управление непрерывной интеграцией (интеграция с системой Jenkins);
- ◆ контроль версий (интеграция с системой Subversion);
- ◆ предоставление среды разработки (интеграция с системой Eclipse);
- ◆ управление тестированием (интеграция с системой FitNesse);
- ◆ управление дефектами (интеграция с системой Bugzilla);
- ◆ формирование отчетов;
- ◆ управление пользователями (разграничение прав доступа и т.д.);
- ◆ предоставление интерфейсов (наборов функций) для межкомпонентного взаимодействия;
- ◆ предоставление web-интерфейса для доступа пользователей к системе.

Выбрана трехуровневая (трехзвенная) архитектура клиент-сервер: клиент, сервер приложений и сервер БД.

Модульность предлагаемого решения (рис. 2) позволит обеспечить масштабирование программной системы в будущем.

Одним из важных процессов, определяющим свойства программной системы и субъективную удовлетворенность им пользователя, является процесс управления требованиями, который инициируется в начале проекта и продолжается на протяжении всего жизненного цикла вплоть до завершения проекта.

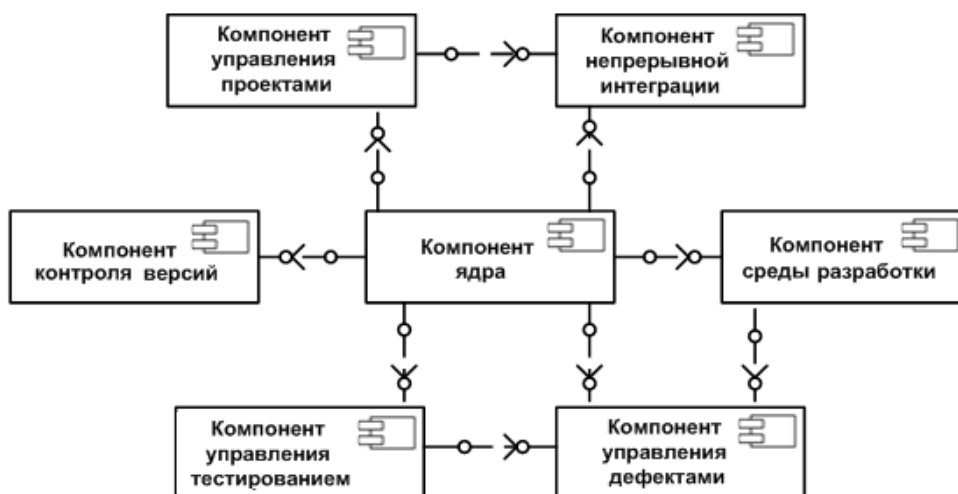


Рис. 2. Диаграмма компонентов системы

Основные участники процессов работы с требованиями: пользователь, заказчик, инженер по программному обеспечению (инженер-программист). Пользователь и заказчик формулируют преимущественно функциональные и нефункциональные требования, в то время как инженер по программному обеспечению описывает системные и программные требования [4].

Общий алгоритм управления требованиями:

- 1) формирование заказчиком и пользователем требований к программному продукту и регистрация их в системе;
- 2) классификация требований инженером-программистом (выделение групп требований: системные, программные, функциональные и нефункциональные);
- 3) определение инженером-программистом приоритетов требований;
- 4) уточнение инженером-программистом требований у заказчика, управление изменениями требований инженером-программистом до окончательного утверждения требований заказчиком;
- 5) трассировка требований инженером-программистом (проверка соответствия получаемого программного продукта требованиям);
- 6) управление изменениями требований инженером-программистом в случае, если во время трассировки были найдены несоответствия.

Основные сущности процесса управления требованиями:

- ◆ требование (requirement);
- ◆ группа требований (requirement\_group);
- ◆ пользователь, последним внесивший изменения в требования (user);
- ◆ роль пользователя, внесившего изменения в требования (user\_role);
- ◆ право доступа пользователя, внесившего изменения в требования (access\_rule).

Были выделены следующие классы, используемые в процессе управления требованиями (рис. 3):

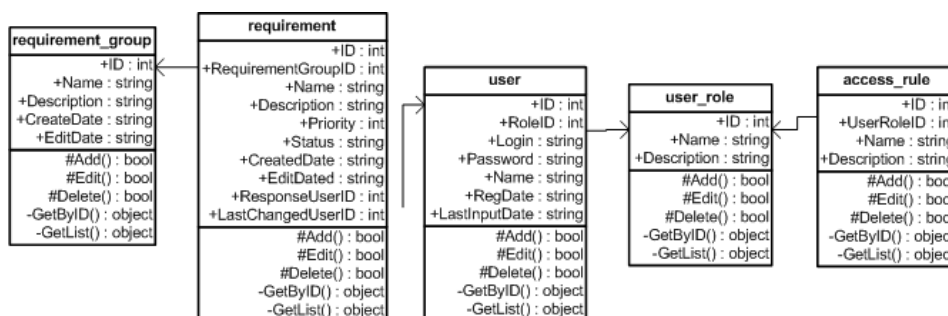


Рис. 3. Диаграмма классов

В результате работы были определены основные процессы управления жизненным циклом программных приложений, выявлены участники этих процессов и описан алгоритм взаимодействия участников процесса управления требованиями к программным приложениям. Определены основные функции и архитектура разрабатываемой системы, рассмотрены основные аспекты проектирования подсистемы управления требованиями: выделены сущности и классы, используемые в подсистеме.

Апробация платформы управления созданием и обслуживанием программных приложений производится на процессе разработки и сопровождения приложений для программно-информационной поддержки задач технического обслуживания и ремонта оборудования предприятий и организаций [5, 6, 7].

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Chappell, D.* What is application lifecycle management? / D. Chappell. – CHAPPELL & ASSOCIATES, 2008.
2. Открытое управление жизненным циклом приложений (ALM) [Электронный ресурс]. – 2007. – Режим доступа: <http://www.interface.ru/home.asp?artId=8633>.
3. Agile Application Life Cycle Management (ALM) Platform [Электронный ресурс]. – Режим доступа : <http://www.versionone.com/Platform>.
4. Орлик, С. Программные требования (Software Requirements по SWEBOK) [Электронный ресурс] / С. Орлик. – 2010. – Режим доступа: [http://swebok.sorlik.ru/1\\_software\\_requirements.html](http://swebok.sorlik.ru/1_software_requirements.html).
5. Кизим, А.В. Обоснование необходимости автоматизации работ по ремонту и техническому обслуживанию оборудования // Известия ВолгГТУ. Серия "Актуальные проблемы управления, выч. техники и информатики в техн. системах". Вып.6. – Волгоград, 2009. – № 6 (54). – С. 118-121.
6. Кизим, А.В. Постановка и решение задач автоматизации работ по ремонту и техническому обслуживанию оборудования // Доклады Томского государственного университета систем управления и радиоэлектроники (Доклады ТУСУРа). – 2009. – № 2. – С. 131-135.
7. Кизим, А.В. Программно-информационная поддержка технического обслуживания и ремонта оборудования с учетом интересов субъектов процесса / А.В. Кизим, Е.В. Чиков, В.Ю. Мельник, В.А. Камаев // Информатизация и связь. – 2011. – № 3. – С. 57-59.

Статью рекомендовал к опубликованию д.т.н., профессор Е.А. Башков.

**Бахман Екатерина Андреевна** – Волгоградский государственный технический университет; e-mail: [bahman.ekaterina@mail.ru](mailto:bahman.ekaterina@mail.ru); 400005, г. Волгоград, пр. Ленина 28; тел: 88442230076; кафедра «САПР и ПК»; магистрант.

**Кизим Алексей Владимирович** – e-mail: [kizim@mail.ru](mailto:kizim@mail.ru); кафедра «САПР и ПК»; к.т.н.; доцент; докторант.

**Bakhman Ekaterina Andreevna** – Volgograd State Technical University; e-mail: bahman.ekaterina@mail.ru; 28, Lenin avenue, Volgograd, 400131, Russia; phone: 88442230076; postgraduate student.

**Kizim Alexei Vladimirovich** – e-mail: kizim@mail.ru; the department of CAD; cand. of eng. sc.; associate professor; doctoral student.

УДК 65.014.1

**В.А. Петраков**

### **АДАПТАЦИЯ УПРАВЛЕНИЯ ВЫСОКОТЕХНОЛОГИЧНЫМ ПРОЕКТОМ**

*Предложен новый адаптированный подход к управлению плохоструктурированным высокотехнологичным, инновационным проектом, основанный на достижении определенных функций цели (время выполнения проекта, его стоимость, качество), являющихся выходами социотехнической системы, и ресурсов их достижения (активы, инвестиции, профессиональные компетенции исполнителей, инструментарий, в том числе программный продукт и др.), являющихся переменными состояния. Такой подход позволяет осуществить системный синтез профессиональных компетенций исполнителей в структуре социотехнической системы. Формализация цели предлагаемого подхода потребовала рассмотрения особенностей инновационного высокотехнологичного проекта, которые часто не поддаются структурированию, а, значит, не могут быть применены другие известные методы и подходы проектирования и управления таким проектом.*

*Адаптированный подход; социотехническая система; высокотехнологичный инновационный проект.*

**V.A. Petrakov**

### **ADAPTATION OF HIGH-TECH PROJECT MANAGEMENT**

*A new adapted approach to managing of low-structured, high-tech, innovative project, based on the achievement of certain objective function (runtime of the project, its cost, quality), are the yields of the socio-technical systems, and resources to achieve them (assets, investments, professional competence of performers, instruments, including software, etc.), which are state variables, was given. Such approach allows to perform systems synthesis of professional competencies for professional executors in the field of socio-technical system. The purpose formalization of this approach need to analyze features of high-technology innovation project, with low-structured. Beside, that cant apply current techniques and approach project.*

*Adapted approach; socio-technical system; low-structured; high-tech innovative project.*

**Высотехнологичный проект.** Общепринятого определения термина «высотехнологичный проект» сегодня не существует, хотя обычно в это понятие вкладывается смысл наукоемкости, значимости, сложности, повышенной степени риска процесса проектирования, либо самого объекта.

Высотехнологичные проекты реализуются, как правило, в областях: аэрокосмические и ядерные технологии и объекты; оборона; медицина: научные исследования высокотехнологичного лечения, биомедицина; новые материалы, материалы для микро – и нанoeлектроники; мембранные технологии; телекоммуникации, информационные технологии; материаловедение, химия и нанотехнологии; научное приборостроение и наукоемкое оборудование, робототехника; рациональное природопользование и защита окружающей среды; энергосбережение и разработка нетрадиционных источников энергии; экология и переработка отходов и ряде других. Такие проекты не вписываются в хорошо структурированные и принятые в зарубежье методы и методики проектирования и управления проектом в