

DOI: 10.18721/JCSTCS.12203  
УДК 004.05

## МЕТОДИКА ВЫБОРА ШАБЛОНА ПРОГРАММНОЙ АРХИТЕКТУРЫ НА ОСНОВЕ АНАЛИЗА ПАРАМЕТРОВ КАЧЕСТВА СИСТЕМЫ

*Ю.В. Ядгарова<sup>1</sup>, В.В. Таратухин<sup>2</sup>*

<sup>1</sup> Московский государственный технический университет имени Н.Э. Баумана,  
Москва, Российская Федерация;

<sup>2</sup> Национальный исследовательский университет «Высшая школа экономики»,  
Европейский исследовательский центр в области информационных систем (ERCIS),  
Москва, Российская Федерация;

Предложен подход к выбору базового шаблона программной архитектуры и сопутствующих тактик проектирования, исходя из требуемых параметров качества систем. Проанализированы основные концепты и понятия предметной области, предложена предметная онтология. Рассмотрены связи между основными понятиями, приведены основные шаблоны программных архитектур. Изучено понятие тактик проектирования, а также их связь с шаблонами архитектур. Сформулирована задача построения программной архитектуры с заданными параметрами качества. В качестве решения предложена аналитическая модель выбора на основе минимизации трудоемкости проекта. Приведен подход к выбору базового шаблона архитектуры и тактик проектирования на примере систем класса превентивного обслуживания.

**Ключевые слова:** архитектура программного обеспечения, качество программного обеспечения, проектирование архитектуры ПО, параметры качества программных систем, тактики проектирования, шаблоны архитектуры, онтология проектирования ПО.

**Ссылка при цитировании:** Ядгарова Ю.В., Таратухин В.В. Методика выбора шаблона программной архитектуры на основе анализа параметров качества системы // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2019. Т. 12. № 2. С. 28–38. DOI: 10.18721/JCSTCS.12203

## GENERAL APPROACH FOR SELECTION SOFTWARE ARCHITECTURE PATTERN BASED ON SYSTEM QUALITY ATTRIBUTES

*Yu.V. Yadgarova<sup>1</sup>, V.V. Taratukhin<sup>2</sup>*

<sup>1</sup> Bauman Moscow State Technical University,  
Moscow, Russian Federation;

<sup>2</sup> National Research University Higher School of Economics,  
European Research Center for Information Systems (ERCIS),  
Moscow, Russian Federation

The paper presents the approach of selection general architecture pattern and design tactics for the software project. Proposed method is based on the satisfying required software quality attributes. The domain ontology is defined along with the main concepts and structure. Base software architectural patterns and main design tactics were analyzed and as a result the problem of building software architecture with required quality attributes has stated. As a solution, analytical model of selecting necessary tactics has presented. The general approach to select the best software architecture pattern and tactics is defined and described with the example of preventive maintenance systems.

**Keywords:** software architecture, software quality, software architecture design, software systems quality attributes, design tactics, software architectural patterns, ontology of the software architecture design.

**Citation:** Yadgarova Yu.V., Taratukhin V.V. General approach for selection software architecture pattern based on system quality attributes. St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems, 2019, Vol. 12, No. 2, Pp. 28–38. DOI: 10.18721/JCSTCS.12203

### Введение

Термин «Программная архитектура» относится к процессу принятия проектных решений при разработке информационных систем для удовлетворения специфических требований. Принимаемые на этапе проектирования архитектуры решения являются компромиссом между параметрами качества разрабатываемой системы и основаны на функциональных требованиях к системе. Решения влияют на качество системы в целом, которое она будет демонстрировать.

Организация Institute of Electrical and Electronics Engineers (IEEE) определяет программную архитектуру как «фундаментальную организацию систем, воплощенную в их компонентах, их связях друг с другом и средой и принципах, руководящих проектированием и эволюцией» [1].

При обсуждении архитектуры специфических систем проектировщики абстрагируются от деталей реализации и фокусируются на решениях, которые необходимо сделать для достижения определенных показателей системы. Эти решения в конечном итоге влияют на то, как система будет декомпозирована на компоненты, и как они взаимодействуют друг с другом. Если подобные решения начинают применяться ко всем системам определенного типа для достижения похожих целей, данные архитектурные шаблоны становятся видимыми. В некоторых источниках их называют *архитектурными типами*, или *стилями*. Детальным исследованиям шаблонов архитектуры ПО посвящены работы [2, 3]. Один из методов оценки применимости архитектурных стилей и шаблонов рассмотрен в [4], а сравнение различных методов оценки стилей приведено в [5]

Соответственно, *тип архитектуры* можно определить как абстракцию программной архитектуры. Другими словами, программ-

ная архитектура является конкретизацией одного или множества архитектурных типов.

Тип архитектуры определяется четырьмя основными свойствами.

1. Множество архитектурных элементов, из которых состоит решение. В качестве примера приведем клиент-серверную программную архитектуру, где элементами соответственно являются машины и ПО клиента и сервера.

2. Структура системы – множество правил и топографических констант, которые определяют допустимые композиции элементов в системе.

3. Семантическая интерпретация композиции элементов, что означает наличие четко прослеживаемого значения у каждой определенной связки элементов.

4. Понимание ограничений и допущений, применимых к данному типу. Каждый тип имеет преимущества и недостатки, которые могут быть критичными или некритичными для определенных видов систем. На основе данных компромиссов чаще всего определяются необходимые параметры качества для реализации в системе.

Элементы процесса проектирования конкретной программной архитектуры с учетом указанных основ в общем случае показаны на рис. 1.

Говоря о структурных элементах типов программных архитектур, выделяют понятие *шаблонов архитектуры*, представляющих собой компоненты и связи между ними (элементы в терминах типа архитектуры). Другими словами, шаблон архитектуры представляет собой абстракцию структуры программной системы, состоящую из компонентов и связей между ними. В литературе нередко используется смешение этих определений, и более широкое понятие типа архитектуры применяют к шаблонам.

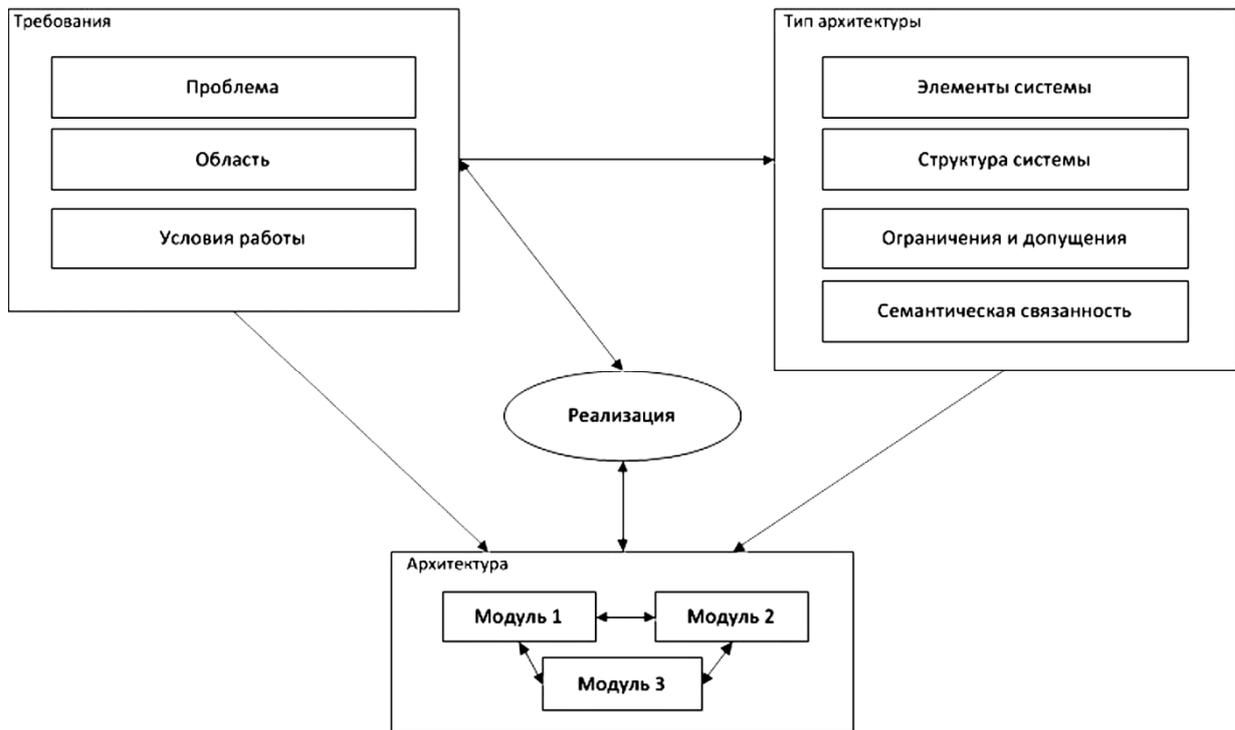


Рис. 1. Элементы процесса проектирования программной архитектуры  
Fig. 1. Elements of the software architecture design process

Приведем некоторые основные архитектурные шаблоны:

- клиент-серверная архитектура;
- архитектура «Точка-точка»;
- архитектура «Каналы и фильтры»;
- событийно-ориентированная архитектура;
- архитектура «Издатель-подписчик»;
- сервисно-ориентированная архитектура (SOA);
- REST-архитектура;
- архитектура слоев;
- облачная архитектура.

Обращаясь к реализации существующих программных архитектур, можно утверждать, что каждый шаблон архитектуры может по построению удовлетворять некоторым параметрам качества (примером может служить хорошая масштабируемость клиент-серверной архитектуры). Для остальных параметров введем понятие *тактики проектирования*. Тактика проектирования (типовой сценарий) – проекционное решение, воздействующее на до-

стижение определенного параметра качества [6]. Реализация тактики всегда меняет ответ системы на конкретное воздействие.

Таким образом, параметры качества программных систем напрямую зависят от исходного шаблона программной архитектуры, а также от реализованных тактик проектирования. Для достижения определенного множества параметров качества программной системы необходима реализация базового шаблона архитектуры и известного количества тактик. В то же время вопрос трудоемкости и достижимости тактик для каждого типа архитектуры остается открытым.

#### Предметная онтология проектирования программной архитектуры

Для программных систем будем придерживаться определения онтологии, представленного в работе [7]. Согласно этому определению, онтология является эксплицитной (явной) спецификацией концептуализации знаний. Формальную модель

предметной онтологии записывают в виде  $O_d = \langle C, F, R \rangle$ , где  $C$  – конечное множество концептов (понятий) предметной области,  $R$  – конечное множество отношений между концептами,  $F$  – конечное множество функций интерпретации (аксиоматизация), заданных на концептах и (или) отношениях предметной онтологии  $O_d$ .

В зависимости от множеств  $C, F, R$ , онтологии могут вырождаться в частные виды, классификация которых приведена в работе [8]. Например, при  $R = \emptyset$  и  $F = \emptyset$  предметная онтология трансформируется в словарь.

Для описания предметной онтологии проектирования архитектуры будем использовать язык моделирования UML 2 с различными типами отношений. Множество аксиом применительно к данной предметной онтологии будет формироваться далее в модели.

В качестве базовых понятий онтологии рассмотрим ее составные части. Множество концептов онтологии проектирования архитектуры включают в себя:

- архитектуру системы;
- особенность проектирования;
- тактику проектирования;
- параметр качества;
- шаблоны архитектуры;
- элементы шаблона архитектуры;
- реализацию шаблона архитектуры;
- особенность архитектуры системы;
- тактику проектирования;
- параметр качества;

- тактики проектирования;
- особенности архитектуры системы;
- параметры качества;
- шаблоны архитектуры;
- элементы шаблона архитектуры;
- компоненты;
- связи.

Отношения между указанными выше концептами можно определить через нотацию UML 2. Представление онтологии в терминах диаграммы классов приведено на рис. 2.

Как видно из диаграммы, при моделировании предметной области используются отношения реализации, зависимости, композиции и агрегации.

Архитектура системы связана отношением агрегации с реализацией шаблона архитектуры (т. к. реализация может существовать вне контекста конкретной программной архитектуры).

Реализация шаблона архитектуры в свою очередь связана с интерфейсом шаблона архитектуры, который зависит от особенностей архитектуры системы и содержит (отношение композиции) элементы шаблона архитектуры.

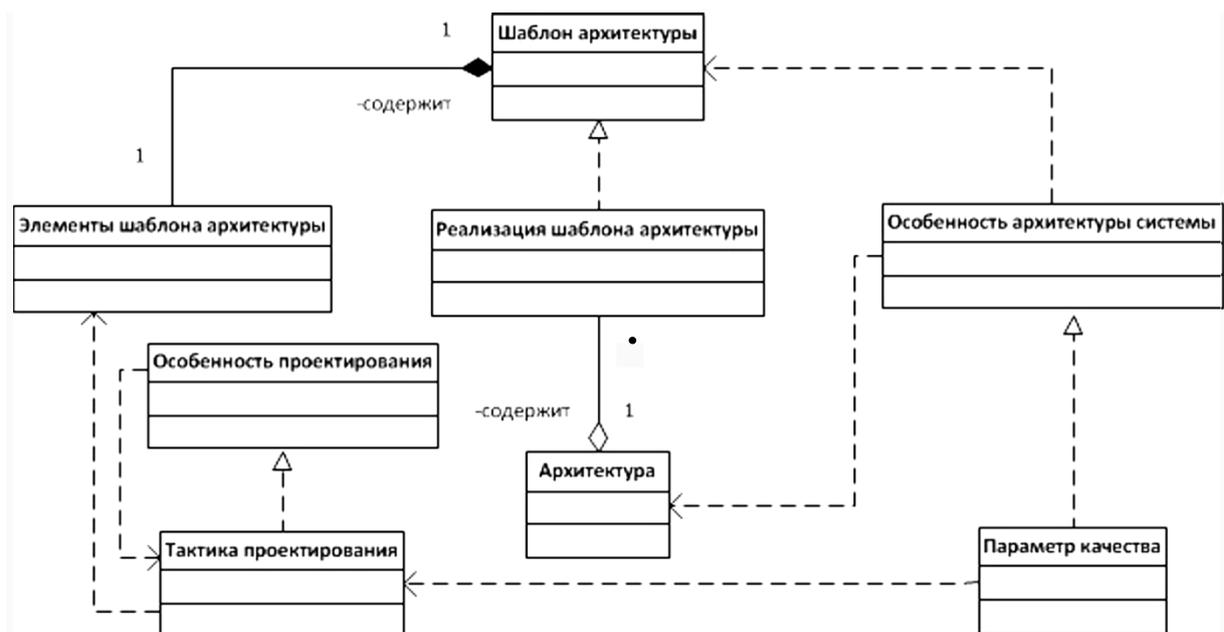


Рис. 2. Структура понятий (диаграмма классов)

Fig. 2. Concept structure (class diagram)

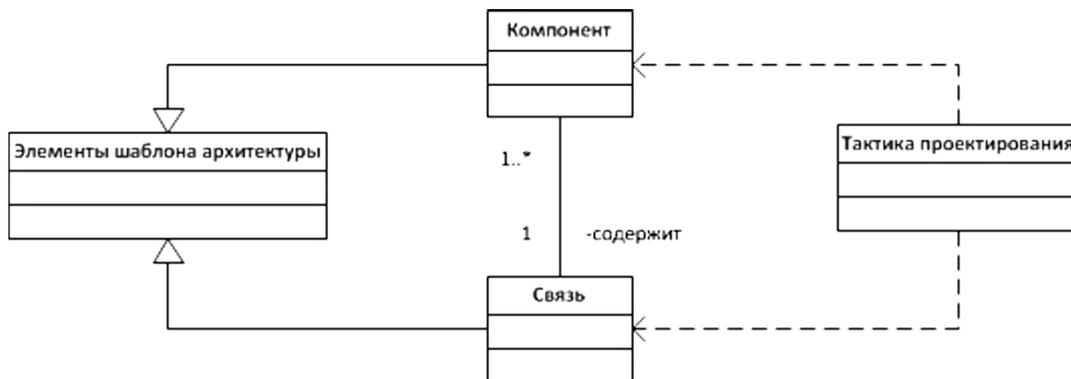


Рис. 3. Элементы шаблона архитектуры (диаграмма классов)  
Fig. 3. Elements of the architecture template (class diagram)

На элементы шаблона архитектуры влияют выбранные тактики проектирования, реализующие особенности проектирования и зависящие от них.

Тактики проектирования зависят от необходимых параметров качества системы, которые реализуют особенность архитектуры системы.

Говоря о структуре понятий в области проектирования архитектуры применительно к данной работе, необходимо обратить особое внимание на связь элементов шаблона архитектуры и тактик проектирования. Структурная декомпозиция указанных понятий приведена на рис. 3.

Компоненты архитектуры и связи между ними являются элементами шаблона

архитектуры (отношение обобщения). Тактика проектирования влияет на архитектуру путем влияния на компоненты шаблона и связи между ними.

Здесь следует отдельно указать, что часть данной работы – конкретизация отношения зависимости компонентов и связей между ними для различных тактик проектирования. Для этого было введено понятие *типа изменения шаблона тактикой*, отражающее сущность модификации шаблона архитектуры и, следовательно, влияющее на трудоемкость. Каждая тактика может вносить от одного до нескольких типов изменений. Различные типы изменений приведены в табл. 1.

Типы изменения шаблона тактикой

Таблица 1

Table 1

Types of pattern change tactics

Тип изменения	Трудоемкость	Описание
Добавление компонента в шаблон Доб <sub>кш</sub>	Низкая	Добавление нового совместимого компонента в шаблон, связь с другими. Структура остается прежней
Модификация компонента Мод <sub>к</sub>	Средняя - Высокая	Меняется структура компонентов шаблона. Влияет на остальные компоненты
Реализация внутри компонента Вн <sub>к</sub>	Низкая	Меняется один из компонентов шаблона
Дублирование компонента Дуб <sub>к</sub>	Низкая	Дублируется один из компонентов, добавляются связи
Добавление компонента вне шаблона Доб <sub>квш</sub>	Высокая	Добавляется компонент, по структуре не подходящий шаблону

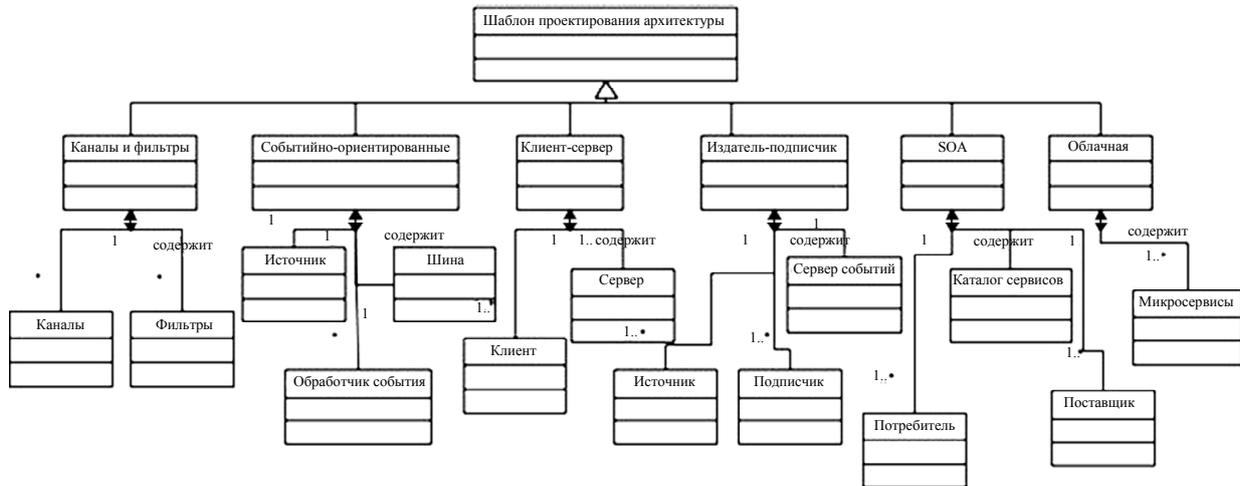


Рис. 4. Метасистема понятий шаблонов архитектуры

Fig. 4. Metasystem of concepts of architecture patterns

В зависимости от типа изменения шаблона может существенно изменяться трудоемкость реализации тактики проектирования, что будет показано далее.

Тип изменения шаблона архитектуры является понятием, зависящим от конкретного шаблона архитектуры. Определение типа полностью зависит от выбранного базового шаблона. Таким образом, реализация тактик проектирования для разных шаблонов будет иметь различные типы изменения и, следовательно, различную трудоемкость.

Говоря о таксономии объектов, необходимо выделить отдельно видовые отношения шаблонов архитектур. На рис. 4 показана метасистема понятий некоторых шаблонов архитектуры, включающая различные виды и элементы указанных шаблонов.

Каждый из специфических шаблонов архитектуры содержит элементы, зависящие от вида шаблона. Так, элементами в шаблоне SOA являются Потребитель сервиса, Каталог сервисов и Поставщик.

#### Модель выбора базового шаблона архитектуры ПО и тактик проектирования

Задача выбора базового шаблона архитектуры и тактик для программных систем состоит в нахождении такой совокупности компонентов и связей между ними для ба-

зового шаблона и множества тактик, реализация которых приводит к достижению требуемых функциональных параметров и параметров качества системы при минимизации трудоемкости.

Полагаем, что критерием достижения параметра качества программной системы является реализация в выбранной архитектуре множества тактик проектирования  $T = t_i, i = 1, 2, \dots, n$ , отвечающих за упомянутый параметр качества. При этом в литературе подчеркивается, что некоторые шаблоны программных архитектур по своему исходному построению удовлетворяют определенным параметрам качества. Данное условие выразим с помощью лингвистической переменной  $K_{ij}$ , определяющей соответствие базового шаблона архитектуры  $i$  требуемому параметру качества  $j$ :

$$K_{ij} = \begin{cases} BC & \left| \begin{array}{l} \text{тип архитектуры удовлетворяет} \\ \text{по параметру качества,} \end{array} \right. \\ CP & \left| \begin{array}{l} \text{тип архитектуры нейтрален} \\ \text{к параметру качества,} \end{array} \right. \\ HЗ & \left| \begin{array}{l} \text{тип архитектуры не удовлетворяет} \\ \text{по параметру качества.} \end{array} \right. \end{cases}$$

В случае если базовый шаблон архитектуры содержит высокую оценку по требуемому параметру качества, реализация тактик проектирования для данного парамет-

ра не требуется, что в свою очередь представим булевой переменной  $x_{ij}$ , значение которой равно:

$$x_{ij} = \begin{cases} 1, & \text{реализация тактик для параметра качества не требуется,} \\ 0, & \text{реализация тактик для параметра качества необходима.} \end{cases}$$

Информация о соответствии базовых типов архитектур некоторым параметрам качества приведена в табл. 2.

В качестве оценки трудоемкости среди множества аналитических методик оценки выбрана наиболее применимая к начальной стадии проекта: СОСОМО II (СОnstructive СОst МОdel – Конструктивная модель стоимости) [9, 10].

Для решения задачи выбора шаблона архитектуры и тактик проектирования, введем понятие *сети параметров качества* системы. Им будем называть совокупность

параметров качества, зависящую от типа системы и требований  $S_t = S\{i, t\}$ , где  $i = 1..k$  – параметр качества,  $t = 1..n$  – тип системы. Каждый параметр качества характеризуется либо реализацией его в базовом шаблоне архитектуры, либо суммой тактик проектирования, необходимых для его достижения на базовом шаблоне:

$$S_t \begin{cases} x_{ij}, \\ \sum_{k=1}^m T_k, \end{cases}$$

где  $T_k$  – тактика проектирования;  $m$  – количество тактик, необходимое для реализации параметра;  $x_{ij} = 1$ , если реализация тактики проектирования для достижения не требуется;  $x_{ij} = 0$  в обратном случае.

Тактика проектирования может быть представлена как модификация элементов архитектурного шаблона (компонентов и связей между ними).

Таблица 2

Соответствие типов архитектур некоторым параметрам качества

Table 2

Correspondence of types of architectures to some quality parameters

Типы архитектур	Интероперабельность	Модифицируемость	Производительность	Масштабируемость	Надежность	Информационная безопасность
Клиент-сервер	1	1	0	0	0	0
Точка-точка	0	0	0	1	0	0
Каналы и фильтры	0	0	1	0	1	0
Событийно-ориентированная	0	1	0	0	0	0
Издатель-подписчик	0	1	0	0	0	0
Сервисно-ориентированная	1	1	0	0	0	0
Брокер	1	1	0	0	0	0
Архитектура слоев	0	0	0	0	0	0
Облачная	1	1	1	1	0	0

Однако реализация одних и тех же тактик проектирования для различных базовых шаблонов архитектуры может существенно отличаться, а, следовательно, отличаться будет и трудоемкость реализации. Для того чтобы учесть данное ограничение, введем понятие типа изменения шаблона архитектуры тактикой. Подсчитаем с помощью методики СОСОМО II относительную трудоемкость реализации добавления компонента в базовый шаблон архитектуры, состоящий из  $n$  компонентов.

Трудоемкость проекта задается формально как функция зависимости от количества тысяч строк кода:

$$Q = \alpha \cdot (KLOC)^b \cdot EAF, \quad (1)$$

где  $Q$  – трудозатраты, выраженные в человеко-месяцах;  $EAF$  – фактор корректировки трудозатрат в зависимости от среды;  $\alpha$  – фактор, зависящий от детальности оценки (постоянная величина, для предварительной оценки равная 2,94);  $b = B + 0,01 \cdot \sum_{j=1}^5 SF_j$ ,

где  $SF_j$  – факторы масштаба. Тогда исходную трудоемкость реализации можно представить как  $Q = \alpha \cdot (KLOC_0)^b \cdot EAF$ . Так как детальность оценки и среда остаются неизменными, реализация дополнительного компонента зависит от количества тысяч строк кода  $KLOC$  реализации и фактора масштаба  $b$ . Трудоемкость готового проекта, выраженная через сумму трудоемкостей компонентов и связей между ними:

$$\begin{aligned} Q &= \sum_{i=1}^n Q_i + \sum_{j=1}^m Q_j = \\ &= \sum_{i=1}^n \alpha \cdot KLOC_i^b \cdot EAF + \sum_{j=1}^m \alpha \cdot KLOC_j^b \cdot EAF = \quad (2) \\ &= \alpha \cdot EAHF \cdot \left( \sum_{i=1}^n \alpha \cdot KLOC_i^b + \sum_{j=1}^m \alpha \cdot KLOC_j^b \right). \end{aligned}$$

Следует отметить, что существенным допущением при данном преобразовании является то, что при реализации каждого последующего компонента фактор масштаба остается неизменным.

Выведем трудоемкость разработки одного модуля через трудоемкость всего проекта (считаем количество модулей):

$Q_i = \alpha \cdot (KLOC_M / n)^b \cdot EAF$ , где  $n$  – количество модулей в проекте,  $KLOC_M$  – количество тысяч строк кода на реализацию всех модулей. Тогда, принимаем  $t = KLOC_M / n$ , где  $t$  – усредненное количество строк кода на один модуль. При этом аналогично,  $Q_j = \alpha \cdot (KLOC_C / m)^b \cdot EAF$ ,  $t_{cb} = KLOC_C / m$ , где  $KLOC_C$  – количество тысяч строк кода на связь между модулями,  $m$  – количество таких связей. Тогда, переписывая выражение (1), получаем трудоемкость проекта как

$$Q = \alpha \cdot EAF \cdot (n \cdot t^b + m \cdot t_{cb}^b). \quad (3)$$

Возвращаясь к типам изменения шаблона архитектуры, в ходе исследования было выделено пять основных способов модификации. Данные способы приведены в табл. 2 и отражают характер изменений. Приведенные значения трудоемкости выражены как функция от величин  $t$  и  $t_{cb}$ . Например, расчет величины  $\Delta_{доб}$  (Добавление нового совместимого компонента в шаблон) заключается в следующем. При наличии базового шаблона архитектуры с известными величинами  $n$  и  $t$  и  $t_{cb}$  добавление еще одного компонента меняет известную трудоемкость (3) как минимум (с наличием хотя бы одной связи) на величину равную:

$$\Delta_{доб} = \alpha \cdot EAF \cdot (t^b + t_{cb}^b).$$

При типе изменений  $\text{Мод}_k$  (Модификация компонента) известная трудоемкость (2) изменяется как

$$\Delta_{\text{мод}} = \alpha \cdot EAF \cdot n \cdot (t_{\text{мод}})^b,$$

где  $t_{\text{мод}}$  – изменение количества кода на модуль. Учитывая, что количество модулей и изменения в них могут быть существенными, трудоемкость модификации компонентов значительна. На основе расчета для каждого типа изменения построен пример графика зависимости величины трудоемкости изменения от  $t$ , принимая  $t_{cb} = 0,2t$  (рис. 5). При количестве модулей равном 10, а  $t_{\text{мод}} = 0,5t$  график принимает вид, показанный на рисунке.

### Трудоёмкость добавления модуля

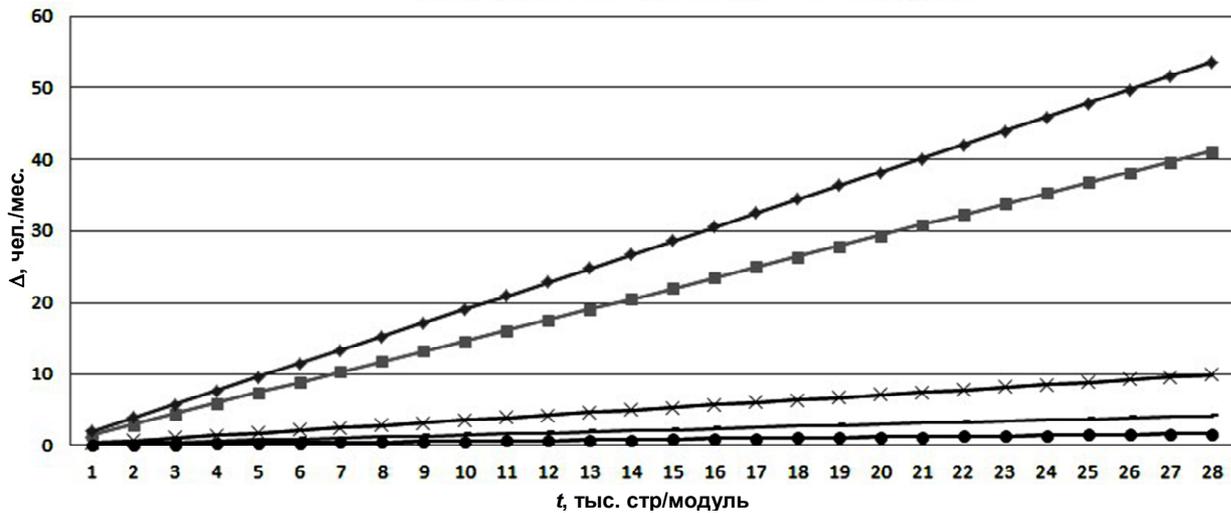


Рис. 5. Трудоёмкость изменения шаблона архитектуры  
 (—x—) — Δдоб\_ш; (—■—) — Δмод; (—) — Δвн; (—●—) — Δдуб; (—▲—) — Δдоб\_вн\_ш

Fig. 5. The complexity of changing the architecture template

Таким образом, общая трудоёмкость проекта для выбранного базового типа состоит из базовой относительной трудоёмкости реализации шаблона (выраженной через величины  $t$  и  $t_{cb}$ ) и суммы относительных трудоёмкостей  $n$  тактик проектирования:

$$Q = Q_0(t, t_{cb}) + \sum_{i=1}^n Q_i(t, t_{cb}, m) = \alpha \cdot EAF \cdot \left( n \cdot t + m \cdot t_{cb} + \sum_{i=1}^n Q_i(t, t_{cb}, m) \right), \quad (4)$$

где  $r$  — тип изменения по табл. 2. Величины  $t$  и  $t_{cb}$  являются зависящими как от шаблона архитектуры, так и от реализации компонента, и рассчитываются с помощью метода функциональных точек для каждого конкретного проекта. Следовательно, задача сводится к нахождению такой комбинации базового шаблона архитектуры и тактик, при которой относительная величина трудоёмкости является минимальной:  $Q \rightarrow \min$ .

В табл. 3 представлен пример относительной оценки шаблона архитектуры «Брокер», выполненный для систем превентивного обслуживания.

В число необходимых достигаемых параметров качества для систем Интернета

Вещей указанного типа входят надёжность, производительность, модифицируемость и информационная безопасность как первичные параметры. Особенностью выбранного шаблона архитектуры является высокий уровень модифицируемости и интероперабельности системы (см. табл. 2). При этом для достижения параметров качества необходимо реализовать дополнительно тактики, адресующие надёжность, производительность и безопасность. Часть списка данных тактик с относительной стоимостью реализации ( $n = 5$ ,  $t_{mod} = 0,2t$ ,  $t_{cb} = 0,1t$ ) указан в табл. 3.

В столбце «Тип изменения шаблона» указаны соответствующие типы изменений, необходимые для реализации тактики проектирования. Относительная стоимость соответственно подсчитана как стоимость изменения трудоёмкости в зависимости от типа.

Подставляя данные значения в выражение (3), получаем выражение для относительной оценки трудоёмкости  $Q = \alpha \cdot EAF(n \cdot t + m \cdot 0,2 \cdot t + 33,863 \cdot t)$ , которое можно использовать в оценке и выборе результирующей архитектуры.

Таблица 3

## Оценка шаблона архитектуры «Брокер»

Table 3

## Evaluation of the Broker architecture pattern

Тактика	Тип изменения шаблона	Относительная стоимость
Увеличение вычислительной эффективности (П)	Мод <sub>к</sub>	1,19t
Уменьшение накладных расходов (П)	Мод <sub>к</sub>	1,19t
Управление темпом событий (П)	Доб <sub>кш</sub>	3,857t
Распараллеливание (П)	Мод <sub>к</sub> , Дуб <sub>к</sub>	1,428t
Копии данных (П)	Дуб <sub>к</sub>	0,238t
Улучшение доступных ресурсов (П)	Мод <sub>к</sub>	1,19t
Политика планирования (П)	Доб <sub>кш</sub> , Мод <sub>к</sub>	5,047t
Пинг-эхо (Н)	Мод <sub>к</sub>	1,19t
Обработка исключений (Н)	Мод <sub>к</sub>	1,19t
Голосование (Н)	Дуб <sub>к</sub> , Доб <sub>кш</sub>	4,095t
Активная избыточность (Н)	Дуб <sub>к</sub>	0,238t
Пассивная избыточность (Н)	Дуб <sub>к</sub> , Мод <sub>к</sub>	1,458t
Повторное включение компонента: затемнение (Н)	Дуб <sub>к</sub> , Мод <sub>к</sub>	1,458t
Аутентификация пользователей (ИБ)	Доб <sub>кш</sub>	3,857t
Авторизация пользователей (ИБ)	Доб <sub>кш</sub>	3,857t
Сохранение конфиденциальности данных (ИБ)	Мод <sub>к</sub>	1,19t
Сохранение целостности (ИБ)	Мод <sub>к</sub>	1,19t

## Заключение

В настоящей статье продемонстрирован подход к проектированию программной архитектуры, основанный на достижении требуемых параметров качества систем. После произведенного анализа проблемы построена онтология предметной области, а также приведена модель достижения требуемых параметров с использованием комбинаций архи-

тектурного шаблона и необходимых тактик проектирования. В качестве критерия выбора шаблона и тактик в модели используется трудоемкость проектирования системы, рассчитанная по методике СОСОМО II. На основе предложенной аналитической модели демонстрируется выбор базового шаблона архитектуры и тактик проектирования для систем класса превентивного обслуживания.

## СПИСОК ЛИТЕРАТУРЫ

1. Coplien J.O., Bjørnvig G. Lean architecture: For agile software development. John Wiley & Sons, 2011 // URL: [http://www.spooch.dk/Ebooks/IT%20Various/lean\\_architecture.pdf](http://www.spooch.dk/Ebooks/IT%20Various/lean_architecture.pdf)
2. Midwinter A., et al. Software architectural styles // URL: <https://slideplayer.com/slide/8014475/>
3. Pahl C., Giesecke S., Hasselbring W. An ontology-based approach for modelling architectural styles // ECSA. 2007. Vol. 4758. Pp. 60–75.
4. Kazman R., et al. The architecture tradeoff analysis method // Proc. of the 4th IEEE Internat. Conf. on Engineering of Complex Computer Systems. IEEE, 1998. Pp. 68–78.
5. Maurya L.S. Comparison of software architecture evaluation methods for software quality attributes // J. of Global Research in Computer Science. 2010. Vol. 1. No. 4.
6. Clements P., Kazman R., Bass L. Software architecture in practice. 3rd Ed.
7. Gruber T.R. The role of common ontology in achieving sharable, reusable knowledge bases // Proc. of the 2nd Internat. Conf. on Principles of Knowledge Representation and Reasoning. Morgan Kaufmann, 1991. Pp. 601–602.
8. Евгеньев Г.Б. Интеллектуальные системы проектирования. 2012. С. 416.

9. **Boehm B.W., et al.** Software engineering economics. Englewood Cliffs (NJ): Prentice-hall, 1981. Vol. 197.

10. **Boehm B., et al.** Cost models for future software life cycle processes: COCOMO 2.0 // Annals of software engineering. 1995. Vol. 1. No. 1. Pp. 57–94.

Статья поступила в редакцию 20.03.2019.

#### REFERENCES

1. **Coplien J.O., Bjørnvig G.** *Lean architecture: For agile software development*. John Wiley & Sons, 2011. Available: [http://www.spooch.dk/Ebooks/IT%20Various/lean\\_architecture.pdf](http://www.spooch.dk/Ebooks/IT%20Various/lean_architecture.pdf)

2. **Midwinter A., et al.** *Software architectural styles*. Available: <https://slideplayer.com/slide/8014475/>

3. **Pahl C., Giesecke S., Hasselbring W.** An ontology-based approach for modelling architectural styles. *ECSA*, 2007, Vol. 4758, Pp. 60–75.

4. **Kazman R., et al.** The architecture tradeoff analysis method. *Proceedings of the 4th IEEE International Conference on Engineering of Complex Computer Systems*, IEEE, 1998, Pp. 68–78.

5. **Maurya L.S.** Comparison of software architecture evaluation methods for software quality attributes. *Journal of Global Research in Computer Science*, 2010, Vol. 1, No. 4.

6. **Clements P., Kazman R., Bass L.** *Software architecture in practice*. 3rd Ed.

7. **Gruber T.R.** The role of common ontology in achieving sharable, reusable knowledge bases. *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 1991, Pp. 601–602.

8. **Evgenev G.B.** *Intellektualnyye sistemy proyektirovaniya [Intelligent Design Systems]*. 2012, P. 416. (rus)

9. **Boehm B.W., et al.** *Software engineering economics*. Englewood Cliffs (NJ): Prentice-hall, 1981, Vol. 197.

10. **Boehm B.W., et al.** Cost models for future software life cycle processes: COCOMO 2.0. *Annals of software engineering*, 1995, Vol. 1, No. 1, Pp. 57–94.

Received 20.03.2019.

#### СВЕДЕНИЯ ОБ АВТОРАХ / THE AUTHORS

**ЯДГАРОВА Юлия Владимировна**  
**YADGAROVA Yulia V.**

E-mail: [y.v.yadgarova@gmail.com](mailto:y.v.yadgarova@gmail.com)

**ТАРАТУХИН Виктор Владимирович**  
**TARATUKHIN Victor V.**

E-mail: [vtaratoukhine@hse.ru](mailto:vtaratoukhine@hse.ru)