Eastern Academic Journal

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

ДЕФИНИРАНЕ НА ОПРОСТЕН И ОБОБЩЕН ФОРМАТ ЗА ДОКУМЕНТИРАНЕ НА ШАБЛОНИ

Мария Р. Армянова

ИУ-Варна
гр. Варна, България

РЕЗЮМЕ — *Шаблоните обобщават най-добрата практика в областта на софтуерните системи. Те предлагат гъвкави решения на често срещани софтуерни проблеми. Една от основните им задачи е да подпомогнат документирането. Това е особено актуално при използването на съвременните гъвкави методологии за разработка, които изискват бърз процес на документиране. В статията е дефиниран формат за документиране на шаблони. Той е опростен и обобщен, за да е приложим към различните видове шаблони. Стандартизирането на документирането им подпомага разработчика при усвояването им и бързото определяне на необходимия шаблон за конкретен софтуерен проблем.*

Ключови думи: шаблони, документиране, стандартизиране, облачна среда, UML

S [Допълнителни материали](#) / [Supplementary materials](#)

DEFINITION OF A SIMPLIFIED AND GENERALIZED PATTERNS' DOCUMENTATION FORMAT

Mariya R. Armyanova

University of Economics, Varna, Bulgaria

ABSTRACT— *The patterns summarize the best practice in software systems. They offer flexible solutions to common software issues. One of their main tasks is to support documentation. This is particularly relevant when using modern, agile development methodologies that require a quick documentation process. A pattern documentation format is defined in the article. It is simplified and generalized to apply for different types of patterns. Standardization of their documentation assists the developers in assimilating them and quickly identifying the required pattern for a specific software problem.*

Keywords: patterns, documentation, standardization, cloud computing, UML

*Corresponding author at: University of Economics, Varna, Bulgaria, e-mail: **armianova@abv.bg**

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

1. ВЪВЕДЕНИЕ

Бързото откриване на подходящите за прилагане шаблони, зависи от документацията им. Документацията трябва да е ясна, за да се ускори и улесни процесът по изграждането на информационната система.

Според Одентал (Odenthal, 1997) за компаниите от софтуерната индустрия е важно да намерят решение на два проблема свързани с разработката на софтуер и текучеството на персонал:

- Как да запазят знанията и опита на екипа;
- Как да се интегрират ново назначените в екипа, т.е. как да се скъси процесът на обучение.

Тези проблеми са пряко свързани с некачествената софтуерна документация. На този фон е изключително актуално прилагането на шаблоните при разработката на системната документация. Считаме, че от гл.т. на разработчиците най-голяма нужда от документиране има средното нивото на архитектурата, което описва компонентите. Това е именно нивото на шаблоните за проектиране. Компонентите, определени на него, отговарят на изискванията за повторна употреба на проекта: достатъчно големи са, за да направят разработката икономична и са достатъчно малки, за да решават често срещани проблеми. В такъв случай документирането на шаблоните и основните компоненти на проекта на системата често се застъпват.

Подходът за документиране на разработките с помощта на шаблоните се състои в идеята за използване на текста на шаблоните, за което е добре те да са стандартизирани или описанието им да е формализирано. Примерните модели в шаблоните, частично покриват проектните решения и са в основата на документацията за проекта.

2. МАТЕРИАЛИ И МЕТОДИ

За да се дефинира формат за документирането на шаблоните са направени проучвания на съществуващите. Използвани са методи на системен, сравнителен и исторически анализ. Сравнени са подходи за моделиране, между които OMT и UML.

3. РЕЗУЛТАТИ

Шаблоните съдържат описание на проблема и решението му. Независимо от това дали са разработени, като самостоятелни или са интегрирани в система или език, е необходимо да се представят под определена форма, за да са лесни за разбиране, използване и обсъждане. Пълното и стандартизирано описание осигурява нужните детайли за прилагане на шаблона и описва възможните последици от приложението му. Съществуват различни форми за описание на шаблоните (Buschmann & Henney, 2005), като почти всеки автор избира подходящ според него начин за това.

Затова най-голямо противоречие при шаблоните има по отношение на начина на описанието им. Още с възникване на идеите за шаблоните започва дискусията за начина на представянето им. Най-често се представят в строго определен формат. Обаче няма единен формат за запознаване с шаблона, около който да се обединят публикациите на PloP (Buschmann и др., 2001), (Gamma и др., 1993). Някои предпочитат стила на Александър

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

(Alexander и др., 1977), Джонсън и Чилингам (Johnson & Cunningham, 1995) се придържат към формата използван от голямата четворка. Според описанието на GoF (Гама и др., 2004) шаблоните включват няколко основни компонента: име, описващо шаблона; проблем, решаван от шаблона; контекст или област на възникване на проблема; фактори, влияещи на проблема или решението; предложение за решение на проблема; контекст на решението; обосновка на решението, т.е. примери за успешни приложения на шаблона; известни приложения или свързани шаблони; автор и други данни за шаблона; ключови данни за търсене; примерен код за решението. Разделени са в 13 секции, показани на фиг. 1.

1. Pattern Name and Classification
2. Intent
3. Also Known As
4. Motivation
5. Applicability
6. Structure
7. Participants
8. Collaborations
9. Consequences
10. Implementation
11. Sample Code
12. Known Uses
13. Related Patterns

Фиг. 1. Формат за представяне на шаблоните определен от GoF.

Но всяка група от учени (Fowler и др., 2002), (Cooper, 1998), (Buschmann и др., 2001), (Coplien, 1991), (Schmidt и др., 2000), (Schumacher и др., 2006), (Hay, 2011), (Fehling и др., 2012) използват различни начини за представяне на шаблоните. Обаче независимо от начина, по който е описан, можем да забележим, че шаблонът има четири основни характеристики – представяне в контекста, на полезност на шаблона; проблем, обхванат от шаблона; роля на шаблона, при формиране на проектното решение и решение, на описания проектен проблем. Тази форма може да включва или пък не, подходящи направления за приложението на шаблона, но се отнася до много от публикуваните шаблони. Важна е тъй като съответства на избраната от нас дефиниция на шаблона, като решение на проблем в зададен контекст. За редица учени използването на фиксирания формат на голямата четворка (фиг. 1) е определящ за това, кое е шаблон. Използването на приетата форма за шаблон, ясно го различава от общия фон на техническата документация.

Обаче фиксираната форма си има и недостатъци. Причината е, че някои шаблони, освен че се явяват решение на конкретен проблем, могат да го решат по различен начин, като се внесат някои промени в тях, т.е. имат вариации. Това разбира се може да се представи и като различни шаблони за всяко решение, но натрупването на няколко близки решения затруднява възприемането и използването им в практиката. Разбира се, традиционният начин за описание на шаблоните има редица предимства, тъй като техническата документация обикновено включва контекст, проблем, роля, фактори и решение. Дали обаче това превръща всяка техническа документация в шаблон също е дискусионен въпрос.

Основен принцип при представянето на шаблона е именуването му. Това е и едно от

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

предимствата на използването на шаблоните. Те обогатяват терминологията на разработчиците. Така въведената терминология позволява с няколко думи да се дадат ефективни указания за бъдещите проекти. Например „използване на прокси (шаблон заместник, Proxy (Гама, 2004))" или „използване на наблюдател (шаблон Observer (Гама, 2004)) за запазване на метриките на продуктите". Разработчиците използват въведената терминология в техническата документация, и това я извежда на ново ниво и улеснява бъдещото откриване на шаблони.

В литературата се предлагат различни методи за прецизно описание на шаблоните, водещи до намаляване на формализма. Формалната спецификация не спомага за разбирането кога и как се използва шаблонът. Формализирането на шаблона прави по-трудно вникването в основната идея. Според Бушман (Buschmann, 2001, с. 19) разработчиците се нуждаят от информация представена в разбираем вид, а не от внушителни формули. Можем да интерпретираме това, като изискване към всеки отделен шаблон да се представи под форма, която го прави максимално разбираем. Строгият формат е нужен за целите на автоматизацията и не е подходящ при представяне на шаблоните от различни нива на архитектурата и решаващи различни типове проблеми. Затова е подходящо да се придържаме към разбирането на общността, че не е нужно разработката на документацията им да се основават на строг формален метод, а за всеки тип може да се използва индивидуален подход.

Според Бушман шаблоните са абстракция, обобщение и затова те предполагат неяснота, двусмислие и неточност. Ако се дефинират в прецизна форма или с математически формули, няма да са шаблони (Buschmann, 2001, с. 19).

Според Коплиен в шаблоните няма фиксирани елементи, за да могат напълно да се променят. С други думи, ако базовата структура се фиксира вече няма да е шаблон (Coplien, 1996). Можем да интерпретираме изказването с оглед на абстрактността на шаблоните, които имат за цел да опишат решения за широк кръг проблеми. Основно предимство от шаблоните е използване на едно решение за няколко, съответстващи си проблеми. Фокусирайки се само върху решенията се загубва това свойство. Шаблоните са концепция с променлива същност и затова са извън обхвата на дословното определяне. Затова приемаме, че формализмът в описанието им може да се търси в известни граници.

Според Бруслер (Brössler и др., 2000) обаче единичното прилагане от разработчиците на шаблоните е дълъг процес, предразположен към грешки. Според Еден (Eden, 2000) прецизното създаване на документацията може да подобри приложението на шаблоните и анализа на взаимовръзките им и инструментите, подпомагащи приложението им. Затова считаме, че е важно да се използва единен формат за представяне на шаблоните, които са от дадена предметна област и едно архитектурно ниво. Още повече това трябва да се спазва, когато шаблоните се обединяват в единен каталог за проблемната област. Единният формат позволява по-бързото запознаване на разработчиците с отделните шаблони и откриване на нужния. Така се засилват някои предимства на шаблоните, а именно да се ускори разработката и да се подпомогне навлизането на разработчиците в нова предметна област или технология.

Според (Eden, 2000) спецификация основана на естествения език е неточна, заради особеностите на езика, който е неформален, неточен и неясен и не може да избегне двусмислието. Считаме, че използването на естествен език има и предимства. Шаблоните се възприемат лесно и бързо от неподготвени разработчици. За да се запознаят с тях не е нужно предварително да изучават методите и начините за формалното им представяне. Така се поддържа ролята на шаблоните, като средство за обучение. Освен това използваните строго формални описания, като кодът на програмен език и диаграмите

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

представят частен пример на шаблона, а не общо правило. Заради липсата на подходящи спецификации разработчиците, използващи шаблоните са принудени да проектират от даден частен пример или да интерпретират насоките от описанията, за да ги приспособят към интересуващия ги контекст. Така, че трябва да използваме и естествен език при документирането им.

Приетите в момента начини на описание не са достатъчно строги и позволяват двусмислена интерпретация, особено заради естествения език. Двусмислените спецификации са пречка за създаването на разбираеми и лесно приложими шаблони. За да се постигне повторяемост на процеса на прилагането им, е нужно строга, формално определена структура на документацията. Считаме, че ясната и строго формалната спецификация на шаблоните е главна цел и постигането ѝ спомага за решаване на въпросите свързани със взаимодействието на шаблоните; валидацията, т.е какво се проверява в конкретната програма, при прилагането на даден шаблон; с начина за съгласуването на шаблона с останалите елементи на програмата и възможните нови проблеми от използването му; използването на инструменти при прилагане на шаблоните.

Липсата на подходящ, общоприет (въпреки че има редица езици за описание и допълнения на UML, те не са общо приети) и насочен специално към шаблоните език за спецификацията им, не позволява шаблоните пълно да се класифицират и взаимодействията им не могат да се определят напълно. Например според Еден (Eden, 1998) отношения като „този шаблон използва онзи” или „тези два шаблона често се използват едновременно” не могат да се дефинират за всички шаблони, особено ако са разработени от различни автори. Но все пак трябва да приемем, че най-подходящият възможен вариант в съвременните условия е да се използва UML.

Затова можем да обобщим направените изводи за документацията на шаблоните. Добре е да има близък формат за представяне на шаблоните, които са от дадена предметна област и едно архитектурно ниво, особено ако са от един каталог. Документацията на шаблоните от един тип би трябва да има строго дефинирана структура. За описанието им е подходящо да се използва естествен език. Нужно е обаче да се съчетае с програмен код и диаграми, като се използва спецификацията на UML. Целта е улесняването на усвояването и съвместното използване на шаблоните. За да се постигне е важно ясно да се поставят изискванията към шаблоните.

Изисквания към документирането на шаблоните

Още с първата конференция OOPSLA’91 се открива ползата от използване на шаблоните за създаването на документацията по проекта (Gamma,1991). Затова и Голямата четворка (Гама, 2004) концентрира усилията си в развитие и документиране на известни софтуерни рамки. Шаблоните обединяват в едно техника за проектиране и документират микро-архитектура, която лесно се открива и може да се ползва многократно. Чрез въвеждане на абстракцията в шаблона става възможно да се открият общи черти в системите и да се намали прекалената сложност в някои от разгледаните системи. Документацията трябва да се разработва успоредно с итерациите и развитието на системата. Точно двете възможности на шаблона да подпомага разработката и описанието на системата, съответстват на принципа за документиране при моделирането и проектирането ѝ.

Въз основа на двете основни функции на шаблоните, можем да определим няколко критерия, на които е важно да отговаря документацията на шаблоните. Те се обуславят от следните функции изпълнявани от шаблоните:

Шаблоните трябва да са така документирани, че ясно да излагат идеите си и да позволяват лесно усвояване;

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

Шаблоните трябва да са документирани така, че да са съвместими с другите шаблони от съответната предметна област, тъй като те трябва да могат да си взаимодействат с тях, да се прилагат заедно и да могат да се съпоставят;

Те се ползват и за документирането на разработката. Така, че трябва тяхната документация да може да се включи в документацията на проекта, което улеснява и ускорява разработката;

Те трябва да са така документирани, че да се откриват лесно, т.е. трябва да са в съответствие с правилата за бизнес моделиране, например на метода UML. По този начин се запазва гъвкавостта на система, използваща шаблони, лесно се внасят промени и се усъвършенства.

За да се отговорят на поставените изисквания, шаблоните се документират с различни подходи, например с текстово описание. То обикновено се използва в книги и колекции от шаблони. Друг подход е структурният, който се базира на мета-модели и общи принципи. Може да се използва и визуален подход, който използва UML модели, диаграми, графики и програмен код. Някои подходи са известни, като езици за шаблони и идеята им е да дефинират начин за представяне на шаблона, който да позволи да се разработи пълен и сложен проект на системата с използване на шаблона. Докато текстовото описание осигурява изчерпателно, но прекалено дълго описание на шаблона, мета-моделите капсулират същата информация в структурата си. Визуалният подход представя използването на шаблона чрез модели или програмен код. Мнозинството от подходите покриват само част от аспектите за представяне на шаблон, като документиране или илюстриране с UML диаграми. Учени предлагат различни подходи, например AM3D (Architectural Modelling with Design Decision Documentation) (Durdik, 2014). Този подход комбинира няколко аспекта, като формализиране на документацията за описание и прилагане на шаблоните и моделиране чрез шаблони в архитектурни диаграми.

От изброените характеристики на подходите се вижда, че всеки един си има предимства и недостатъци. Например методът на текстово описание реализира функцията за лесно усвояване на шаблоните от разработчиците. Най-голямото предимство на този метод е, че не е нужна предварителна подготовка на читателя, за да навлезе в същността на шаблона. Затова този подход е подходящ за първите секции във формата за описание на шаблона, които имат за цел да представят шаблона. Така разработчиците се запознават лесно с шаблоните и могат бързо да открият подходящия.

Визуалните подходи пък са подходящи при описание на структурата на шаблоните. Създадените модели покриват функцията за използване на документацията на шаблоните в документацията на проекта. Структурният подход пък е подходящ за представяне на инстанции на шаблона.

Приемаме подхода на Дурдик, че при документирането на шаблоните трябва да се съчетаят предимствата и на трите подхода, като в различните секции от се използват различни форми за документиране.

Дефиниране на общ формат за представяне на шалоните

Шаблоните се представят в колекции и каталози, съдържащи като цяло независими шаблони, но обединени според някаква категоризация. Освен началната категоризация според областта на приложението им, на по-късен етап не се прави ново групиране, за да се обединят в единна класификация шаблоните разработени от различни автори. Основно усилията на общността на разработчиците са насочени към подобряване на взаимодействието на шаблоните от голямото множество от съществуващи шаблони. За да се постигне тази цел е важно при създаването си шаблоните да използват съвместими

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

формати за описание.

Както заключихме в предната подточка не може да се избере единен формат за представяне на всички видове шаблони, но можем да открием някои общи елементи. Анализирайки различни видове шаблони можем да открием най-подходящите секции в структурата им.

С най-голямо влияние са шаблоните разработени от голямата четворка. Това е каталог от 23 шаблона, като всеки е форматиран по строго определен начин и включва 13 секции, показани на фиг. 1 и описани в таблица 1.

Таблица 1. *Стандарт за форматиране на документацията на шаблона според GoF*

Секция	**Съдържание**
Име на шаблона и класификация (Pattern Name and Classification)	Името има за цел да представи на кратко съдържанието на шаблона. Изборът на подходящо име е важен, тъй като става част от речника на разработчиците. Класификацията отразява мястото и ролята на шаблона и подпомага разработчиците при откриването на подходящ шаблон за разработвания софтуер.
Цел (Intent)	Накратко се представя работата на шаблона, обосновава се накратко проблема, специфичните въпроси за разработката и целите на шаблона.
Известен като (Also Known As)	Ако има се изброяват и другите имена, под които е известен шаблона.
Мотивация (Motivation)	Описва се сценария, илюстриращ възникналия при проектирането проблем и начина, по който структурите от класове и обекти в шаблона го решават.
Приложимостта (Applicability)	Изясняват се ситуациите, при които могат да се прилагат шаблоните. Определят се некачествените проекти, при които възниква проблемът решаван от шаблона. Описват се характеристиките на подобни ситуации, за да се разпознават по-лесно.
Структура (Structure)	Представя се чрез диаграми от метода UML. Диаграмата на класовете се базира на OMT(Rumbaugh и др., 1991), а интерактивната диаграма на диаграмите на Якобсон (Jacobson и др., 1992) и Буш (Booch, 1994), които са в основата на метода UML. Интерактивната диаграма се ползва за илюстриране на последователността на исканията, отговорите и взаимодействията, разменяни между обектите.
Участници (Participants)	Описват се участващите в шаблона класове и обекти и техните отговорности.
Сътрудничество (Collaborations)	Поясняват се начините на взаимодействие между класовете и обектите, за да изпълнят задълженията си.
Последствията (Consequences)	Описва начините, по които шаблонът постига целите си, посочват се недостатъците или слабите места от използването му. Уточнява се кои елементи от системната архитектура, остават независими от шаблона и могат да се променят, без да го засегнат.
Приложение (Implementation)	Чрез кратки съвети се уточняват технологиите нужни за реализацията на шаблона, възможните проблеми при прилагането му и особеностите на използвания програмен език. Специално за шаблоните на Голямата четворка това са C++ or Smalltalk.
Примерен код (Sample Code)	Представя кратки фрагменти от кода, които илюстрират как се прилага шаблона за съответния програмен език.
Известни употреби (Known Uses)	Показва примери за използване на шаблона в реални системи. Включват се примери от различни предметни области.
Свързани шаблони (Related Patterns)	Описват се други шаблони, които са тясно свързани с разглеждания шаблон. Показва се каква е разликата между шаблоните и кои други шаблони използва за реализацията си.

Всички спецификации на шаблоните с изключение на секциите Структура и Примерен код са написани на естествен език. Структурата е представена с диаграми на метода OMT

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

(Blaha и др., 1991), а примерният код е написан на C++ или Smalltalk.

За да направим съпоставка с предложената от GoF структура на шаблона се спираме на другия вид шаблони, които са насочени към прилагане в информационните системи. Това са обектноориентираните архитектурни шаблони на Фаулър (Fowler и др., 2002) и са създадени, като специализиране на шаблоните за проектиране за конкретен тип корпоративна система. За описанието им Фаулър (таблица 2) е избрал формата, използван за ориентирани към шаблони софтуерни архитектури (Buschmann, 2001).

Таблица 2.

Стандарт за форматиране на документацията на шаблона, използван от Фаулер

Секция	Съдържание
Име на шаблона (Name)	Освен, че именува шаблона, обобщава и предназначението му.
Известен като (Also Known As)	В нея се представя друго име на шаблона, ако е известен и по друг начин.
Пример (Example)	Избран е реален пример, срещащ се при някои разработки, за да се демонстрира същността на проблема и нуждата от шаблона. При описанието на проблема се посочват примери за илюстриране на решението и се представят приложни аспекти, когато това решение е нужно или полезно.
Контекст (Context)	Това са ситуации, в които шаблонът може да се използва.
Проблем (Problem)	Проблемът засегнат в шаблона, заедно с допълнителните фактори, които по някакъв начин му влияят.
Решение (Solution)	Представят се основните принципи за решението, на които се основава шаблона.
Структурата (Structure) на шаблона	По-точно се прави детайлна спецификация на структурните аспекти на шаблона, чрез подходящи означения. Използват се диаграми от метода UML.
Динамика (Dymanics)	Чрез типичен сценарий се описва поведението на шаблона по време на изпълнение.
Приложението (Implementation)	Представят се насоки и указания за прилагане на шаблона. Това са само предложения, а не строги формални правила. Представеното приложение трябва да се адаптира, за да се приспособи към различните нужди на разработваната система. Понякога се налага да се добавят елементи, да се детайлизират отделни стъпки или да се променя редът на изпълнението им. Понякога се илюстрират възможни приложения на шаблона с помощта на метода UML. Могат да се представят и отделни важни фрагменти от решението на примерния проблем.
Примерно решение (Example Resolved)	Въвежда се дискусия за някои важни аспекти на решението, които не са обхванати в предните четири секции – Решение, Структура, Динамика и Приложение.
Варианти (Variants)	Представя специализации или конфигурации на шаблона.
Известни употреби (Known Uses)	Включени са примери с използване на шаблона, взети от реални системи.
Последствия (Consequences)	Описва ползите от използването на шаблона, заедно с възможните обвързаности или ограничения, наложени от използването на шаблона.
Свързани шаблони (See Also)	Посочват се шаблоните, които решават сходни проблеми или подпомагащи дейността на описания шаблон.

При този формат на описание на шаблоните, не е задължително да се използват абсолютно всички секции. Например шаблонът може да няма алтернативни имена или вариации. За други шаблони в секция Решение се представя достатъчно ясно и пълно същността им и може да се пропуснат секции Структура, Динамика или Приложение. Ако за шаблон се представя само едно примерно приложение, не е необходимо то да се описва в

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

отделни секции. Да се избере, кои секции да останат или отпаднат, е трудоемък и продължителен процес, обикновено преминава през няколко редакции, от различни експерти в съответната предметна област. Целта е пълното и ясно описанието на шаблона.

Съпоставянето на двата формата сме представили в таблица 3. Въпреки, че някои секции имат еднакви имена, не винаги имат едно и също съдържание. Може съдържанието им да се представят от различен ъгъл или с различни средства (диаграми). Например в секцията Последствията (Consequences) на GoF се набляга на начина на работа и възможните недостатъци. Докато при Фаулър в Последствия (Consequences) се описват ползите от използването на шаблона, заедно с възможните обвързаности или ограничения, наложени от използването му.

Таблица 3. *Сравнение между формата на класическите шаблони и архитектурните*

Формат на GOF	**Архитектурни шаблони Фаулър**	**Сходства**	**Несъответствия**
Pattern Name and Classification *Intent* *Also Known As* *Motivation* *Applicability* *Structure* *Participants Collaborations* *Consequences Implementation* *Sample Code* *Known Uses* *Related Patterns*	*Name* *Also Known As* *Example* *Context* *Problem* *Solution* *Structure* *Dymanics* *Implementation* *Example Resolved* *Variants* *Known Uses* *Consequences* *See Also*	Името на шаблона и другите имена се срещат и в двата формата. Секцията Intent на GOF шаблона донякъде съответства на Problem Motivation накратко изпълнява функциите на Solution и Problem Consequences разглежда от различен ъгъл последствията. В GOF се представят начин на работа, недостатъците срещу ползите и ограничения от използването на шаблона при архитектурните. Applicability се покрива със Context Collaborations не съвпадат с Dymanics, но имат подобни роли. Implementation, Structure имат еднаква роля във форматите Known Uses и Related Patterns съответстват на Known Uses и See Also	Participants няма съответствие, но може да се проучи от структурата Sample Code няма съответната секция, но в редица други се представя код Example Resolved и Variants нямат съответствие в GOF.

От сравнението в таблица можем да направим извода, че до голяма степен секциите в двата формата за представяне на шаблоните се припокриват. Основните различия са в начина, по който е представен кодът. При корпоративните архитектурни шаблони, които описват голяма структура, той не е от съществено значение и на места в някои други секции се представят фрагменти. Например в Example Resolved. За сметка на това в архитектурните шаблони се набляга на различните вариации на шаблона, според конкретния контекст. Това отново се обуславя от характера на шаблоните и от това, че те са от различни нива на декомпозиция на системата. Докато GOF шаблоните предлагат общо приложими обектно-ориентирани шаблони, то архитектурните шаблони, зависят от спецификата на конкретната област и трябва да предложат възможности за по-голямо приспособяване.

За документиране на шаблоните е важно да се предвиди използването на шаблона при описанието на разработката. Идеята е документацията да се сведе до нуждите на

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

разработчиците. Така, че ако се пропуснат различията идващи от спецификата на всеки вид шаблони, секциите могат да се обобщят по-малко на брой, но важни секции. Като резултат се получава структура на шаблоните, която има общия вид, показан на таблица 4.

Таблица 4. *Секции в структурата на общата документация за шаблон*

Секция	Съдържание
Име (Name)	Име на шаблона
Кратко описание (Overview) или Цел (Intent)	Стреми се да запознае с проблема, решаван от шаблона
Контекст (Context) или Мотивация (Motivation)	Описва подробно контекстът на проблема. Представят се компонентите на проекта. Полезно е да се разгледат и последователността от дейности, които навеждат разработчиците на решението да използват точно този шаблон. Показва се връзката между проблемната област и шаблона.
Начин на работа	Описва действието на шаблона и резултата от използването му. Може да се използват различни диаграми.
Предизвикателства или Примерното решение (Example Resolved)	Представят насоки и указания за прилагане на шаблона. Посочват се начини за адаптирането му. Определят се важните въпроси за всяка реализация.
Приложение (Implementation)	Посочва характеристики на прилагането на шаблона и се пояснява със съответните секции в програмния код. Програмният код може да се отдели и в отделна секция.
Въпроси или Последствия (Consequences)	Обсъждат въпроси за възможностите и ограниченията, т.е. възможностите за прилагането и развитието им, съпоставени с други алтернативни проектни решения.

За секцията Кратко описание може да се използва илюстрация, подходяща за ориентир в контекста на проблема или диаграма на класовете за обектноориентираните шаблони.

Ако шаблонът е предназначен за описание на принципите на обектноориентираната парадигма, като шаблоните на GoF, Начин на работа може се замени със секции, които описват класове и обекти. Например може да се включи и секция Роли (Roles) или участници (Participants). В нея се именуват класовете за конкретната инстанция на шаблона като “Pattern: Role”. По този начин читателят бързо се ориентира за спецификите на ролите. Посочва се всяка роли и какво е значението ѝ за шаблона. Може да се добави и секция Сътрудничество (Collaborations) се описва взаимодействието между клиента и екземпляра на шаблона.

Ако е необходимо освен секция Предизвикателства може да се добави и секция Варианти (Variants). Също могат да се добавят и секции Известни употреби и Свързани шаблони.

Въпреки, че този формат е сравнително общ, той не е универсален. Съгласно извода, че единен формат е възможен само за представяне на шаблоните, които са от дадена предметна област и едно архитектурно ниво, се налага форматът да се специализира при конкретната си употреба. Целта е да се постигне съвместимост с останалите шаблони от същата предметна област, като се отразят особеностите на средата.

Стандарти за представяне на облачни шаблони

Дефинираният общ формат за прилагане на шаблона, трябва да може лесно да се специализира за шаблоните от конкретна предметна област. Като най-бурно развиващо се направление в компютърната индустрия е подходящо да се спрем на облачните изчисления. В момента в България повече от половината от създавания софтуер е предназначен за облачна среда, подари което шаблоните за облачната среда са актуални и подходящи за

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

прилагане на определения формат за документиране (таблица 4).

Доставчиците на облачни услуги се опитват да подпомогнат разработката на софтуер за облачните си платформи и затова голяма част от тях предлагат различни видове шаблони. Някои отразяват специфични възможности на конкретната облачна среда, други са насочени към общи характеристики на облачните приложения. Те са от различно архитектурно ниво. Някои дори са определени, като механизми. Цялото това многообразие се отразява и на документирането им. За всяка група се определя индивидуален формат.

Както коментирахме използването на единен формат позволява лесното разбиране на шаблоните и съпоставянето им (Petre, 1995, с. 33-44). Затова е важно да се определи общ формат, който да може да представи особеностите на различните групи и да позволи съвместното им прилагане при създаването на единната документация на моделите на системата.

NIST (National Institute of Standards and Technology) описва общите характеристики на облаците и не включва специфичните особености на модели за доставка на услуги от различните облачни доставчици. (Fehling и др., 2014) NIST предлага формат за представяне на шаблоните, подробно описан в таблица 5.

Таблица 5. Стандарт на NIST за форматиране на документацията на облачните шаблони

Секция	Съдържание
Име на шаблона (Pattern Name)	Използва се за идентификация на отделния шаблон. Може да се посочи и за коя облачна среда е разработен.
Цел (Intent)	За всеки шаблон най-напред се обявяват целите му, за да може разработчикът лесно да се ориентира, дали това е нужният му шаблон. Освен това така се пояснява решението.
Картинка (Icon)	Използва се за графична идентификация на шаблона. Така се дава възможност на разработчика лесно да включи в диаграма на архитектурата на облачното приложение шаблона.
Контекст (Context)	Описва средата и факторите, влияещи на решавания от шаблона проблем.Представя различни, подходящи на пръв поглед, решения, които обаче са неуспешни или не са оптимални и ефективни. Могат да се споменават и други шаблони. Шаблоните може да представят модели за разширяване на облака, за предлаганите облачни услуги или за базовите функции на облака въпреки, че тези шаблони не се използват от разработчиците на приложения, работещи в облачна среда. Но подобни шаблони поясняват поведението на облачната среда и предлаганите облачни услуги. По този начин разработчикът може да избере нужните за приложението си среда и услуги. Освен това тези шаблони създават контекста, в който ще работи разработваното приложение. Представят се и взаимоотношенията с шаблоните, използвани от разработчика в приложението му. Така разработчикът може оптимално да използва предлаганите от облачния доставчик услуги.
Решение (Solution)	Показва се мястото на шаблона в структурата на системата. Накратко се описва начинът на решаване на проблема от шаблона чрез наблягане на основните въпроси. Целта е описанието да е кратко, за да могат потребителите на бързо да прочетат всички секции до тази и да придобият представа какво точно прави шаблонът. Обикновено това се постига с кратка скица на архитектурата на решението.
Резултат (Result)	Решението се представя пълно и подробно. Скицираната в решението архитектура се описва и се обсъжда поведението на шаблона, след приложението му. Представят се и нови проблеми и заплахи, с които се сблъсква шаблонът. Разглеждат се и други шаблони, които са обвързани с него, както и шаблони, с чиято помощ могат да се преодолеят новите проблеми.
Вариации (Variations)	Представя различни вариации на шаблона. Ако разликите при прилагане на шаблона са незначителни, не е нужно да се разглеждат различни вариации.

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

Секция	Съдържание
	Но ако вариациите са прекалено големи може да се наложи да се създаде отделен шаблон за определени случаи.
Свързаните шаблони (Related Patterns)	Разглежда шаблоните, които решават сходни проблеми или без които е невъзможна пълноценната работа на шаблона. Възможно е приложението на един шаблон да е несъвместимо с приложението на друг. Затова се дискутират взаимоотношенията на шаблоните. По този начин се създава схема на взаимоотношенията на шаблоните в облачната среда.
Известни употреби (Known Uses)	Описва съществуващи примерни приложения на шаблона, продукти предлагащи шаблона или поддържащи приложенията му.

Препоръчително е да разгледаме стандартите на шаблоните за най-популярните облачни доставчици. В момента в България това са облачните платформи MS Azure и AWS. Те имат свой формат, представен в таблица 6 и 7.

Таблица 6. Стандарт за форматиране на документацията за шаблоните на Azure

Секция	Съдържание
Име (Pattern Name)	Името и общата характеристика на шаблона.
Контекст и проблем (Context and problem)	Описва на кратко проблема и контекста, за който е приложим шаблона. Това позволява на читателя да се запознае с шаблона и да открие дали шаблона има нещо общо с решавания от него проблем.
Решение (Solution)	Описва се начинът на работа на шаблона, представя се кратка схема и се отбелязват някои особености.
Въпроси и мнения (Issues and considerations)	Поставят се важните за реализацията на шаблона въпроси и ограниченията при прилагането му.
Кога да се използва шаблонът (When to use this pattern)	В нея се изброяват подходящи примерни приложения на шаблона и неподходящи такива.
Пример (Example)	Посочва примерна реализация (инстанция) на шаблона и се представят важни фрагменти от програмния код.
Допълнителни насоки (Related guidance)	Изброяват се други шаблони, които могат да се използват съвместно с шаблона при определен контекст.

Таблица 7. Стандарт за форматиране на документацията за шаблоните на AWS

Секция	Съдържание
Име (Pattern Name/Summary)	Съдържа име и кратко описание на шаблона.
Решавани Въпроси (Solving Issues)	Посочват се причините довели до създаване на шаблона и предизвикателствата, които се решават чрез прилагането му.
Обяснение на шаблона (Explanation of pattern/Resolution in the cloud)	Описва начина за решение на проблема в облака или конфигурацията на решението на шаблона.
Приложение (Implementation)	Обяснява се начинът на прилагане на шаблона в AWS платформата.
Ползи (Benefits)	Описва ползите от използването на шаблона за приложението.
Забележки (Notes)	Разглежда тесните места, предимствата, недостатъците и особеностите при прилагането на шаблона.
Други (Other)	Прави сравнение с други шаблони, подходящи ситуации за прилагане и други особености.

В облачната платформа на Google са описани 17 собствени програмнозависими шаблони. Те обаче на са документирани с определена структура, а с няколко думи е описан метод за решаване на конкретен проблем. Така, че можем да приемем, че GCP няма собствен формат.

В таблица 8 са представени отделните секции във форматите за описание на шаблоните. Анализирайки ги можем да определим някои важни прилики и разлики в съдържанието на документациите.

Таблица 8. Сравнение между форматите на шаблоните за облачни платформи

Формат на NIST	**Формат за MS Azure**	**Формат за AWS**	**Сходства**	**Различия**
Pattern Name *Intent* *Icon* *Context* *Solution* *Result* *Variations* *Related Patterns* *Known Uses*	*Pattern Name* *Context and problem* *Solution* *Issues and considerations* *When to use this pattern* *Example* *Related guidance*	*Pattern Name/ Summary* *Solving Issues* *Explanation of pattern/ Resolution in the cloud* *Implementation* *Benefits* *Notes* *Other*	И в трите формата се представя и името и контекста на шаблона В Solution се представя накратко работата на шаблона. Related Patterns, Related guidance и Other имат едно и също предназначение.	NIST използват графично означение на шаблона, което позволява по-лесното му включване в проектната документация В Result на NIST се представя пълно и подробно решение, докато във формата на Azure се дават насоки и фрагменти от кода. NIST представя вариации на шаблона, докато при Azure се скицира един пример.

Представеното сравнение в таблицата показва, че при облачните платформи до голяма степен има прилики във форматите. Различията се получават от добавянето на някои елементи, като графични означения. Освен това в платформата MS Azure се набляга на по-лесното усвояване и откриване на подходящ шаблон, за сметка на пълната документация. Най-формална е структурата предложена от NIST, като основните характеристики се описват в отделни секции.

Сравнявайки форматите за представяне на шаблоните за облачните платформи с определения от нас общ формат (таблица 4) се вижда, че има общи елементи, като имена, кратко описание, контекст на прилагане, т.е. като цяло секциите са подобни. Но се забелязват и различия. Облачните шаблони са представени с по-кратко описание, тъй като целта на документация им е да позволи на разработчиците бързо да се запознаят с тях и да започнат да ги прилагат. Освен това облачните шаблони могат да имат и графични означения (икони), което позволява лесното включване на шаблона в UML моделите на системата. Тъй като всички разгледани шаблони са обединени в каталози, форматът им включва секция за описание на взаимоотношенията им.

Независимо от различията си, повечето формати за представяне на шаблоните, включват няколко характеристики. Според нас това са:

- Име (Name) – позволява разпознаване на шаблона и откриването му в структурата на системата;
- Описание (Description) – кратко описание на предназначението на шаблона;
- Контекст (Context of use) – показва кога и къде е най-подходящо да се използва шаблона;
- Примери (Where used) – представя успешни употреби на шаблона;
- Начин на работа (How it works) Описва какво е действието на шаблона и какъв е

резултатът от използването му.

Можем да направим извода, че определения формат за шаблон (таблица 4) има всички характеристики и следователно лесно може да се впише в различни предметни области с малки промени. Например за да се документират облачни шаблони с него могат да се добавят секции Икона и Свързани шаблони и да се специализира съдържанието им.

Все пак в момента според направено авторово проучване по метода на отзовалите се повечето български фирми използват гъвкави методологии при разработката на софтуер. А те се характеризират с бърза разработка, за сметка на качеството на документацията. Тази тенденция е отразена във формата на шаблоните за облачните платформи. Освен това по-голямата част от софтуерните специалисти разработват софтуер за облачни платформи, което прави изключително актуално прилагането на точно този тип шаблони.

4. ЗАКЛЮЧЕНИЕ

След анализиране на форматите за представяне на шаблони от различни автори е определен общ вариант за документацията им, с цел постигане на предимства им, изведени в първа глава. Тя е стандартизирана и опростена, за да е приложима за различните видове шаблони. Актуална технология в момента е облачната, затова са проучени съществуващите шаблони за нея.

Общият им формат позволява на разработчика бързо да се ориентира и определи, подходящите за дадения проблем. Така се подпомага получаването на качествен резултат от процеса на прилагането им. Качествената документация на шаблоните намалява разходите, времето и трудността при разработката на системите. Създадените с тях приложения са с по-лесни за съпровождане, т.е. с по-дълъг жизнен цикъл, гъвкави и преносими.

4. СЪТРУДНИЧЕСТВО

Използвани са начините за документиране на шаблоните на различни учени, но те са цитирани в текста.

5. ДОПЪЛНИТЕЛНИ МАТЕРИАЛИ

Представено е документирането на шаблонът Наблюдател (Observer) на GoF .

6. ЛИТЕРАТУРА

Статия

Brössler, P. & Prechelt, L. & Tichy, W. F. & Unger, B. & Votta, L. G., (2000). *A Controlled Experiment in Maintenance Comparing Design Patterns to Simpler Solutions.* н.м.: IEEE Transactions on Software Engineering.

Eden, A. H. (1998). *Giving 'The Quality' a Name.* н.м.: New York: SIGS Publications: Journal of Object-Oriented Programming, Vol. 11, No. 3, pp. 5-11.

Fehling, C. & Ewald, T. & Leymann, F. & Pauly, M. & Schumm, D., (2012). *Capturing Cloud Computing Knowledge and Experience in Patterns.* н.м.: IEEE 5th International conference on cloud computing, pp.726-733, IEEE inc.

Gamma, E. & Helm, R. & Johnson, R. & Vlissides, J., 1993. *Design Patterns: Abstraction and Reuse of Object-Oriented Design.* н.м.: Proceedings of ECOOP '93, pp. 406- 431. [ECOOP93]

Johnson., R & Cunningham, W., 1995, *PLoPD Conference 1, 95, Pattern Languages of Program Design.* н.м.: Addison Wesley, 95, ISBN 0-201-60734-4.

Odenthal, G., & Quibeldey-Cirkel, K., (1997). *Using patterns for design and documentation.* н.м.: ECOOP'97, LNCS#1241, pp 511-529.
Petre, M., **(**1995**).** *Why looking isn't always seeing: : readership skills and graphical programming.* Communications of the ACM, Volume 38 Issue 6, pp. 33–44.

Книга

Alexander, C. & Ishikawa, S. & Silverstein, M. & Jacobson, M. & Fiksdahl-King, I. & Angel, S., (1977). *A Pattern Language: Towns - Buildings – Construction*. н.м.: Oxford University Press.
Blaha, M. & Eddy, F. & Lorensen, W. & Premerlani, W. & Rumbaugh, J., (1991). *Object Oriented Modeling and Design.* н.м.: Prentice Hall.
Booch, G., (1994). *Object Oriented Analysis and Design with Applications,* Addison Wesley.
Buschmann, F. & Meunier, R. & Rohnert, H. & Sommerlad, P. & Stal, M., (2001). *Pattern-Oriented Software Architecture – A System of Patterns.* н.м.: John Wiley & Sons, 2nd Edition.
Buschmann, F. & Henney, K., (2005). *Pattern-Oriented Software Architecture – On Patterns and Pattern Languages.* н.м.: John Wiley & Sons.
Coplien, J. O., (1991). *Advanced C++ – Programming Styles and Idioms.* н.м.: Addison-Wesley.
Cooper, J., (1998). *The Design Patterns, Java Companion.* н.м.: Addision-Wesley.
Durdik, Z., (2014). *Architectural Design Decision Documentation through Reuse of Design Patterns.* н.м.: KIT Scientific Publishing, ISBN 978-3-7315-0292-0.
Eden, A. H. (2000). *Precise Specification of Design Patterns and Tool Support in Their Application. PhD Dissertation.* н.м.: Department of Computer Science - Tel Aviv University.
Fehling, C. & Leymann, F. & Retter, R. & Schupeck, W. & Arbitter, P., (2014). *Cloud computing patterns.* н.м.: Springer-Verlag Wien. ISBN 978-3-7091-1567-1
Fowler, M. & Rice, D. & Foemmel, M. & Hieatt, E. & Mee, R. & Stafford, R., (2002). *Patterns of Enterprise Application Architecture.* н.м.: Addison-Wesley.
Hay, D., (2011). *Data model patterns Conventions of Bought.* н.м.: Dorset House Publishing, New York.
Gamma, E., (1991). *Object-Oriented Software Development based on ET++: Design Patterns, Class Library, Tools., Ph.D. Thesis.* н.м.: University of Zurich.
Jacobson, I., Christerson, M., Jonsson, P., Oevergaard, G., (1992).*Object-Oriented Software Engineering -- A Use Case Driven Approach*, Addison-Wesley.
Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., (1991). *Object-Oriented Modeling and Design*, Prentice-Hall.
Schmidt, D. C. & Stal, M. & Rohnert, H. & Buschmann, F., (2000). *Pattern-Oriented Software Architecture – Patterns for Concurrent and Networked Objects.* н.м.: John Wiley & Sons.
Schumacher, M. & Fernandez, E. B. & Hybertson, D. & Buschmann, F. & Sommerlad, P., (2006). *Security Patterns: Integrating Security and Systems Engineering.* н.м.: John Wiley & Sons.
Гама, Е. & Хелм. Р. & Джонсън, Р. & Влисидес, Дж., (2004). *Шаблони за дизайн.* н.м.: СофтПрес.

7. СЪКРАЩЕНИЯ

1. GoF – Gang of Four, Голямата четворка. Това е групата на Гама, Хелм, Джонсън и Влисидес (Гама и др., 2004).

2. PloP – Pattern Languages of Programs. Група от конференции организирани от Hillside Group обединени от идеята за развитие и подобряване на шаблоните за проектиране.

3. UML – Unified Modeling Language е подход, който се състои от последователност от диаграми за визуализиране, специфициране, конструиране и документиране на елементите на една софтуерна система.

ISSN: 2367–738
Issue 1, pp. 63-78, March, 2018

4. OOPSLA – (Object-Oriented Programming, Systems, Languages & Applications). Серия от конференциираз глеждащи проблемите на обектноориентираната разработка на софтуер.

5. OMT – Object Modeling Technique. Обектноориентиран подход за моделиране и проектиране на софтуер. Създаден е през 91 от Rumbaugh, Blaha, Premerlani, Eddy, Lorensen.

6. MS Azure – Microsoft Azure е облачна платформао сигуряваща услуги за създаване, тестване, развитие и управление на софтуерни приложения. Тя осигурява услуги за трите слоя SaaS (software as a service) PaaS (platform as a service) и IaaS (infrastructure as a service) на облачната архитектура.

7. AWS – Amazon Web Services осигурява облачна платформа за работа на софтуер с индивидуални характеристики, съгласно сключеното споразумение.

8. GCP – Google Cloud Platform Облачна платформа за реализация на софтуер, предоставена от фирма Google.