

МЕТОДОЛОГИИ ЗА РАЗРАБОТКА И ВЪЗМОЖНОСТИ ЗА ПРИЛАГАНЕ НА ШАБЛОНИ

Мария Р. Армянова

РЕЗЮМЕ — Шаблоните имат място в процеса на разработката, тъй като техните предимства са от съществено значение за подобряване качеството, разширяемостта и адаптивността на създаваните системи. За да се определи, кои са популярните методологии е проведено изследване за използването на шаблоните в българските компании, разработващи софтуер. В статията тези методологии са разгледани и са сравнени възможностите им за включване на подход за прилагане на шаблони. Шаблоните могат да подобрят някои аспекти на методологиите. Най-подходящи за включване на подход за използването им са RUP и гъвкавите методологии, които се основават на подхода UML, поддържат MDD и са достатъчно гъвкави.

Ключови думи: гъвкави методологии, екстремното програмиране, шаблони за проектиране, Kanban, RUP, Scrum

[Допълнителни материали / Supplementary materials](#)

DEVELOPMENT METHODOLOGIES AND POSSIBILITIES FOR PATTERNS USAGE

Mariya R. Armyanova

ABSTRACT— The patterns have their place in the development process, as their advantages are essential to improve the quality, extensibility, and adaptability of the designing systems. The popular methodologies are determined by a research for the patterns use in Bulgarian companies developing software. In the article these methodologies are review and are compare their ability to include a patterns approach. It has been found that the patterns can improve some aspects of methodologies. RUP and flexible methodologies are the most appropriate to include an approach to patterns use. These methodologies are based on the UML approach, support the MDD and are flexible enough.

Keywords: Agile Modeling, XP (Extreme) Programming, design patterns, Kanban, RUP, Scrum

1. ВЪВЕДЕНИЕ

Разработката на приложения много пъти е еднотипна работа. Програмистите пишат код и го тестват многократно, за да са сигурни в правилната работа на софтуера. Проектирането на сложна система е трудно и изисква опит и професионализъм. За да се улесни разработката се използват шаблони. Те са изключително популярни, заради многобройните си предимства, което поражда и непрекъснато нарастване на средите, за които се разработват и прилагат.

Шаблоните имат място в процеса на разработката, тъй като техните предимства са от съществено значение за подобряване качеството, разширяемостта и адаптивността на създаваните системи, а недостатъците им са преодолими.

В литературата са познати редица шаблони, които подпомагат разработчиците при изграждането на софтуерните системи. За да се реализират предимствата на шаблоните, е необходимо да се въведе повтораемост и систематизираност в процеса на създаването и прилагането им при разработката на информационни системи. За целта е подходящо да се определят съвременните методологии, които позволяват и подпомагат използването на шаблони.

2. МАТЕРИАЛИ И МЕТОДИ

За да се определи, кои са популярните методологии е проведено изследване за използването на шаблоните в българските компании, разработващи софтуер. За целта е използван методът на отзовалите се. Проучването е анонимно и е осъществено чрез пряко допитване до повече от 200 фирми от отрасъла, открити при търсене на „софтуерни фирми” от Варна и София, или известни на автора. Създадена е анкета чрез Google, съдържаща 8 въпроса от затворен тип, но участниците могат да добавят и собствени отговори. Последният въпрос им позволява свободно да изложат идеите си за шаблоните и методологиите за разработка, но те почти не са се възползвали от възможността. Основните въпроси са свързани с използваните във фирмата шаблони и методологии за разработка. Част от въпросите са насочени към определяне на типа на отзовалите се – дейност, размер на фирмата и позиция на анкетирания. Въпросите са описани в Приложение 1.

3. РЕЗУЛТАТИ

Проучването е проведено в периода 31.07.2017г. – 07.08.2017г. Представители на 61 фирми са попълнили анкетата, като 9 са от Варна, останалите са от София. Обобщените резултати са представени в Приложение 2.

С проучването се цели да се провери популярността на шаблоните за проектиране, както и да се определят използваните софтуерни технологии и методологии във фирми от българската софтуерна индустрия. Стремежът е чрез изследването да се определи технологията и средата, за която все още няма разработени достатъчно шаблони. Чрез сравнение на отговорите на двата въпроса за типа на използваните шаблони и за разработването на софтуер в облачна среда установяваме, че над 50% от проектите, обхванати в проучването, са предназначени за облачна среда, а в същото време едва в 18% от тях се използват облачни шаблони. Това разминаване ни дава основание да твърдим, че има нужда от още добре описани шаблони, например според изследването за облачна среда. Целта на определянето на популярността на софтуерни технологии и методологии е да насочим усилията си към подобряване на използването на шаблоните за тези от тях, които са актуални.

Изследването показва, че над 90% от проучените фирми прилагат шаблони при разработката на софтуерните си продукти, което показва актуалността и нуждата от шаблони и методологии, позволяващи системното им прилагане.

За да се открие мястото на шаблоните, което позволява пълноценно им използване, е необходимо да се изследват методологиите за разработка. Както посочихме, за прилагането на шаблоните за изграждане на информационни системи не е дефинирана специализирана методология, но в редица методологии има възможност да се включат дейности по употребата им. На база направен анализ на съдържанието на основните методологии за разработка (Филипова и др., 2013, с. 30-39) и (Филипова, Парушева и Александрова, 2017, сс. 96-131) може да се направи заключението, че шаблоните е подходящо да бъдат интегрирани основно в итеративните методологии, в които се използват модели. Към тях се отнасят RUP (Rational Unified Process) и гъвкавите методологии.

RUP е известна, сложна и формализирана методология, осигурена със среди и средства за разработка. Според нас тя е подходяща за прилагане на шаблоните, защото е базирана на обектноориентирания метод UML. Шаблоните се документират често с UML, което премахва нуждата от усвояването на нови специфични методи за описанието им и позволява на разработчиците да се съсредоточат върху заложените в тях решения.

RUP (Филипова и др., 2013, с.31-36; Agile Modeling, 2012; Rational Software, 1998) е методология за разработване на системи, която поддържа реализирането на така наречения унифициран процес¹ на моделиране. Основната ѝ характеристика е, че дефинира процес на разработка, базиран на модели, т.е. поддържа подхода Model Driven Development (MDD), и архитектурата Model Driven Architecture (MDA). Подходът MDD и MDA се използват при изграждането на най-модерните и сложни интегрирани системи. Шаблоните поддържат MDD подхода, като различните им видове подпомагат разработването на моделите от различните слоеве на архитектурата.

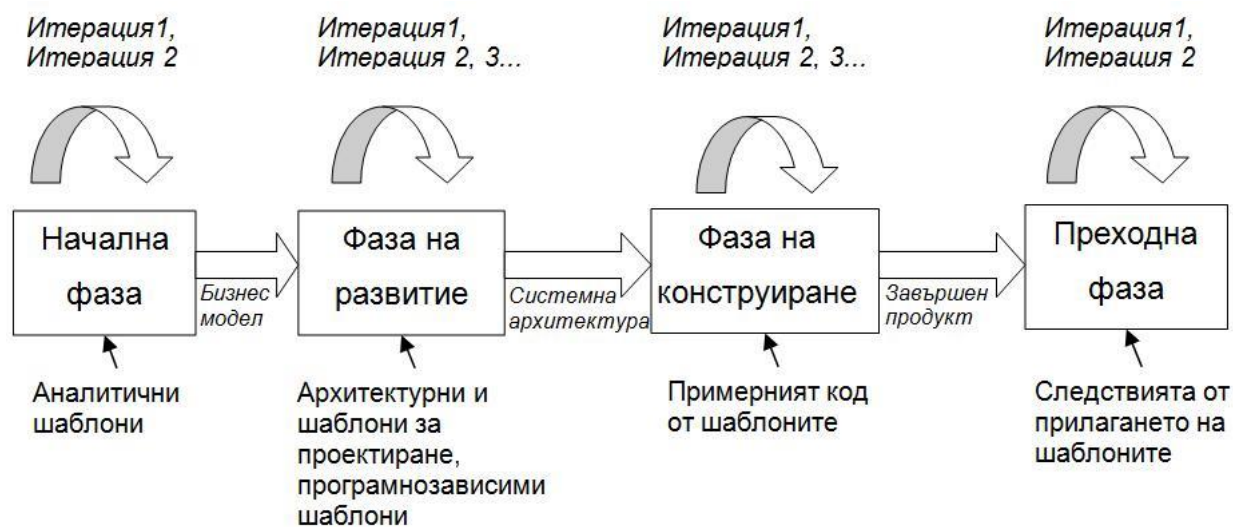
Методологията RUP включва редица практики и е по-скоро софтуерна рамка от процеси, а не строго определен монолитен процес. Така екипът от разработчици избира измежду множеството процеси най-подходящия за конкретния проект. Затова и първата стъпка при внедряване на RUP, е да се определи индивидуален процес за разработка, който по желание на разработчиците да включва и подход за комплексното прилагане на шаблоните.

Още на етап анализ, когато се разработват бизнес модели, отразяващи изискванията, могат да се открият шаблони, например предложените от Фаулър (Fowler, Rice, 2002; Fowler, 1996). Те са база за разработването на бизнес модел и концептуалната архитектура на системата и добре се вписват с RUP методологията. Първо се дефинира архитектурата, а после започва работа по проекта.

Методологията RUP допуска промени, които да подпомогнат прилагането на шаблоните. Шаблоните в RUP може да се използват в две перспективи: фази и дисциплини. Във фазите ѝ могат да се добавят дейности, които да позволят използването на шаблоните. Например през началната фаза се определят обхватът, целите и изискванията към системата. С помощта на UML диаграми и аналитични шаблони, например предложените от Фаулър (Fowler, Rice, 2002; Fowler, 1996) се създава бизнес модел на системата. Той е база за определяне на системните компоненти и потребностите от шаблони. През фазата на развитие се проектират елементите на системата и се създава архитектурата ѝ на изходно ниво с участието на шаблони при разработката на всички слоеве. През фазата на

¹ Това е единен, итеративен и усъвършенстващ се процес, базиращ се на множество архитектурни модели, които поддържат всички аспекти на системата.

конструирането се разработва и се тества кодът, който се съобразява с примерните реализации на шаблоните. Те се интегрират, съгласно указаните в тях взаимоотношения и изисквания. За определянето на някои условия за валидиране на системата през преходната фаза спомагат следствия от прилагането им. (фиг. 1.).



Фиг 1. Фази при RUP и участие на шаблоните в тях

Шаблоните се вменват и в основните процеси, които дефинират дейности по разработката, известни в RUP като „дисциплини“. Те са: бизнес моделиране; определяне на изискванията; анализ и проектиране; реализация; тестване; внедряване. Също като фазите всяка една дисциплина може да бъде разширена, така, че да позволи широко използване на шаблоните. Например има възможност разработването на бизнес модела да се подпомогне от аналитичните шаблони. По време на анализа и проектирането създаваните модели включват решенията предложени от архитектурните, за проектиране и програмнозависимите шаблоните, а за реализацията спомога включеният в тях примерен. Тъй като в описанието си съдържат предимствата и недостатъците, следващи от прилагането им, те участват и при тестването за откриване на тесните места в системата, които се нуждаят от специално внимание. Използването на шаблоните е подобно на това във фазите (Таблица 1).

Таблица 1. Прилагане на шаблоните във фазите и дисциплините на RUP

Участие на шаблоните	Фази при RUP	Дисциплини на RUP
Аналитични шаблони	Началната фаза	Бизнес моделиране
Определяне на потребностите от шаблони	Началната фаза	Определяне на изискванията
Архитектурни, за проектиране, програмнозависими и платформозависими шаблони	Фаза на развитие	Анализ и проектиране
Примерният код от шаблоните	Фаза на конструиране	Реализация
Следствията от прилагането на шаблоните	Преходна фаза	Тестване, Внедряване

Получаване на определени артефакти² в RUP, е необходимо условие за прилагането и създаването на шаблоните. Съществуват някои препятствия за използването им. При комплексните методологии, към които се отнася RUP, целият процес по разработка е стандартизиран. За да се добави нова дейност, трябва да има строго регламентиран резултат от нея. Шаблоните предлагат решение на често срещани проблеми, а не на всички. Понякога резултатът от опита да се приложат е извода, че няма подходящ шаблон и разработчиците трябва да разработят самостоятелно елемента. Затова е трудно да се управлява и контролира използването им.

Често RUP се прилага заедно с гъвкавите методологии (Манифест за Agile, 2001). Чрез анализа им, можем да открием особеностите им, които допускат използването на шаблоните (таблицата 2).

Таблица 2. Сравнение между RUP и гъвкавите методологии

Критерий	RUP	Гъвкавите методологии
Използване на подходи и стандарти за проектиране	Използва диаграмите на метода UML, който е неразделна част от методологията. Включва указания за начина на работа и изисквания към артефактите.	Не са толкова строги по отношение на използваните подходи при разработката.
Използване на шаблони по време на разработката	Допуска откриването и използването на шаблони, тъй като поддържа средствата на метода UML. Позволява модели от разработените системи или техни части да се запазват, като нови шаблони.	Има възможност разработчиците да въведат собствени, предпочитани от тях подходи, включително и за прилагане на шаблоните.
Получаване на нужните артефакти	Основно предимство е, че съдържа указания, кога какъв тип модели да се разработват, включително и модели, които не са част от метода UML. Например модели на данните и диаграми на потоците от данни.	Повечето дефинират само организационни и социални подходи и дават свобода на избора на управленските и технологичните. Позволяват получаване на предпочитаните артефакти.
Взаимодействието е на екипа	Разчита се на взаимодействието, но то се управлява и документира. Шаблоните позволяват въвеждането на нови термини и подобряват комуникацията.	Разчитат на неформалното взаимодействие на разработчиците от екипа. Шаблоните ускоряват комуникацията чрез споделената лексика.
Повторна употреба на разработени елементи	Позволява повторна употреба на артефактите, а управлението им с цел повторна употреба е част от Enterprise Unified Process (EUP). Могат да се ползват създадени модели, шаблони, компоненти, open source software	Няма въведен подход или ограничения за повторната употреба на елементи.

² Artifacts е термин въведен в методологията RUP за обозначаване на крайни или междинни резултати, които се създават и използват по време на разработката на даден проект. Те се използват за капсулиране и предаване на информация в проекта. Те са документи, модели или елементи в модел, като клас, подсистема.

Критерий	RUP	Гъвквите методологии
	(OSS) и дори инструменти.	
Обновяване на средствата, инструментите и технологиите	Разчита се на използване на актуалните средства и технологии. Строго се проследява всеки елемент от разработката, промените се внасят трудно, но регулярно. Има силна връзка между използваните подходи и артефактите им. Все пак е допустимо да се използва подход за използване на шаблоните.	Разчита се на актуалните средства и инструменти, но промените се внасят, когато е необходимо. Допускат лесно прилагане на подход за използване на шаблоните.
Използване на лесно усвоими инструменти за разработка	Разчита на инструменти, поддържащи UML метода (като Enterprise Architect), но могат да се използват и други.	Позволява използване на предпочитани от разработчиците инструменти.

Въпреки, че UML е естествена предпоставка за използването на шаблоните, все пак има някои препятствия за това в RUP. При комплексните методологии, към които се отнася RUP, целият процес по разработка е строго регламентиран. Сложно е да се внасят промени, поради силната обвързаност на дейностите и артефактите на разработката. За да се добави нова дейност, трябва да има строго регламентиран резултат от нея. Шаблоните не са в състояние да разрешават абсолютно всеки проблем в проекта. Те решават често срещани проблеми и е възможно за определен проблем да няма дефиниран подходящ шаблон. Така, че понякога резултатът от опита да се приложат е извода, че няма шаблон и разработчикът сам трябва да разработи елемента. Затова е трудно да се управлява и контролира използването им.

При гъвквите методологии се преодоляват част от трудностите на добавяне на подход за използване на шаблоните. Те се отличават с по-голяма свобода по отношение на управлението и документирането на дейностите по разработката, като целта им е да се получи успешен краен резултат (Филипова и др., 2013; Филипова, Парушева и Александрова, 2017). Целта им е да се получи успешен краен резултат и в процеса на разработка е възможно да се използват различни подходи и механизми, включително и за прилагане на шаблоните.

Целта на прилагането на гъвквите методологии и шаблоните е да се ускори и улесни процеса на разработка, тъй като дългосрочните проекти са с по-голям риск от провал, особено за малките фирмите, разработващи софтуер (Griffiths, 2016; Griffiths, 2017). Чрез гъвквите методологии разработчиците се опитват да постигнат баланс между бързо достигане до краен резултат и спазването на определени стъпки и процедури, които гарантират качество на разработката (Eeles, 2009). Основната разлика с останалите методологии е, че при тях се отделя по-малко време и усилия за разработка на документация, като основната цел е създаване на работеща система. Това определя мястото на шаблоните в тях.

Шаблоните ускоряват създаването на документацията, като позволяват в проекта лесно да се включат схемите, диаграмите и моделите им. Считаме, че така те поддържат принципите на гъвквите методологии. За гъвквите методологии е рационална документацията, която не ограничава обучението и се създава в сътрудничество за вътрешните нужди на екипа. Един от стълбовете на гъвквите методологии е

взаимодействието в екипа (Griffiths, 2017). Шаблоните улесняват комуникацията с другите участници в проекта, като увеличават професионалната лексика. Те въвеждат нови постановки, които не е нужно да се доказват и обясняват, например „MVC модел”. Шаблоните показват начините да се приложат в проекто-програмните решения новите технологии, а иновативността е принцип на гъвкавите методологии.

Считаме, че втората важна особеност на гъвкавите методологии от гл.т. на шаблоните е, че са ориентирани към разработчиците, а не към процеса на разработка. Техен принцип е съобразяване на разработката с индивидуалните качества на всеки разработчик от екипа, което позволява всеки да използва индивидуален подход, подобряващ качеството на работата му. Използването на шаблоните е начин да се включи опита на експертите при решаването на проблемите.

Гъвкавите методологии, също като RUP се базират на итеративния модел на жизнения цикъл на софтуера. Техен принцип е, че след всяка итерация се натрупва опит в разработката и екипът подобрява уменията си. Дефинирането на предишни решения, като шаблони е начин да се запази натрупания опит. Това е практика във 31% от проучените фирми, разработващи софтуер с гъвкави методологии [Приложение 1, 2].

Сред най-популярните представители на гъвкавите методологии е екстремното програмиране XP (Extreme Programming) (Beck, 2004). При него се използва итеративен модел на жизнения цикъл, като итерациите са екстремно кратки. Считаме, че прилагането на шаблоните е по-лесно при итеративния процес на разработка. На всяка итерация се разработва компонент на системата и за него се откриват подходящите шаблони. Шаблоните ускоряват разработката, като предлагат решение на често срещаните проблеми и а при XP итерациите продължават по една, две седмици, но е възможно и всеки ден да се правят по няколко итерации и да се предоставят по няколко версии на системата.

Тази методология улеснява използването и откриването на подходящи шаблони, защото на всяка от „екстремно“ кратките итерации се разработва отделна част от системата, която решава проблем, съответстващ на обхвата на функционалните задачи, описани в шаблоните (например проблемът за съответствието на данните при едновременната им визуализация в няколко елемента се решава от шаблона „наблюдател”).

Обикновено екстремното програмиране се използва за системи, които имат малък по обем програмен код, но жизнено важен за качеството на системата, т.е. за които бързодействието, надеждността и устойчивостта на сривове са от голямо значение, например за системи, работещи в реално време или многонишковы процеси, за които има разработени много шаблони (Hanmer, 2007; Schumacher, 2006; Sandén, 2003). Използването им осигурява въвеждането на желаното свойство, например шаблоните на Ханмер подпомагат разработката на устойчив на сривове софтуер.

Според направеното от нас проучване за 60% от проектите се използват гъвкавите методологии, като най-популярни са Scrum и Kanban. Това обуславя разглеждането на особеностите им.

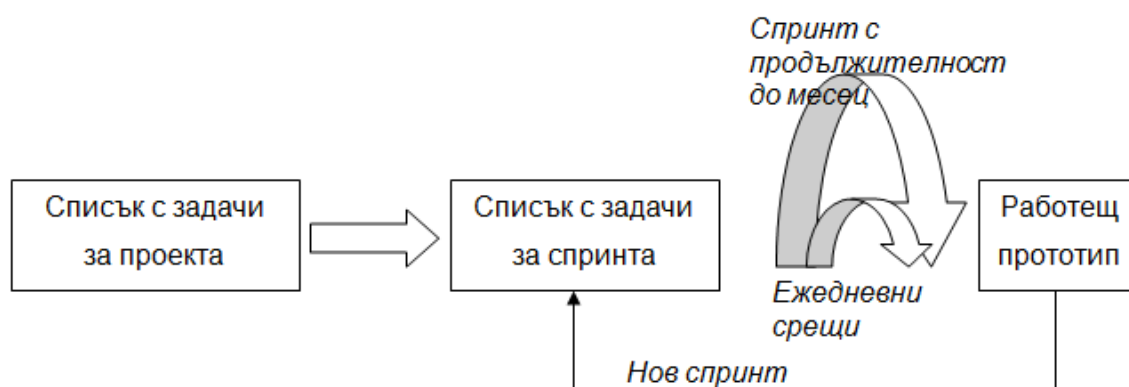
Използването на шаблоните не оказва влияние върху процесите на управление на проектите и затова твърдим, че не подобрява самите методологии за управление, като Scrum и Канбан. Но шаблоните могат да поддържат принципите им, т.е. имат косвен ефект. Те подпомагат комуникацията, като въвеждат нови термини, а това повишава качеството на взаимодействието в екипа. Шаблоните представят начини за практическо приложение на новите технологии, предимствата и недостатъците им, възможните следствия от използването им, а поддържането на иновациите също е принцип на Scrum и Канбан. От друга страна считаме, че гъвкавите методологии подпомагат управлението на процесите по

прилагането на шаблоните. В тях могат да се добавят контролни точки, да се следи за получаването на определени артефакти, които да подобрят управлението и организацията на подхода за прилагане на шаблоните.

Scrum (Schwaber, 2004) е методология за разработка, която се базира на обектноориентираната парадигма, както и повечето шаблони. Според направеното проучване в момента е най-популярната методология в изследваните фирми [Приложение 1, 2], като и в световен мащаб повече от половината от разработчиците, прилагащи гъвкави методологии, са се ориентирали именно към нея (Versionone.com, 2013).

Методологията Scrum осигурява работна рамка за управлението на проекти, като цели да повиши производителността на малки екипи, за които са трудни и ограничаващи ориентиранията към процеси методологии. Целта ѝ е управлението на проекти, а не предлагане на подход за осъществяването на самата разработка, което позволява използването на предпочитаната техника от разработчиците. Затова тя може да се прилага съвместно с други софтуерни методологии – с XP и с подход за комплексно прилагане на шаблони.

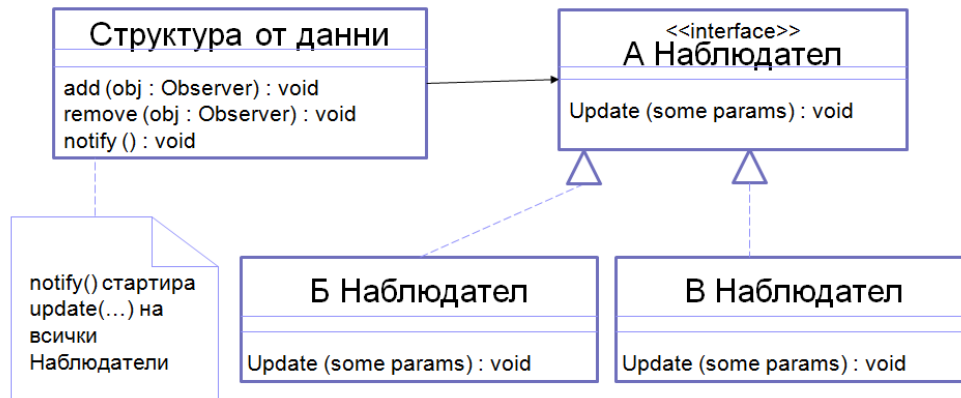
Scrum разделя разработката на итерационни цикли с продължителност между седмица и месец, наречени спринтове и изисква тясно проследяване и контрол чрез ежедневни scrum срещи. Всеки спринт е пълен итеративен цикъл, включващ всички стадии на жизнения цикъл на разработката на информационна система. При него се разработва завършен прототип на част от системата. След всеки спринт се демонстрира разработената функционалност, проверяват се постигнатите резултати и се набелязват новите задачи за следващия спринт, като се съобразяват с промените в изискванията, технологиите и средата. Процесът на работа е показан на фиг. 2. Считаме, че списъкът от задачи на методологията Scrum може да включи действия по прилагане на шаблоните, например откриване на подходящ шаблон, проследяване на взаимовръзките му, проучване на възможностите за интегрирането му в системата и др.



Фиг. 2. Последователност на стъпките при Scrum

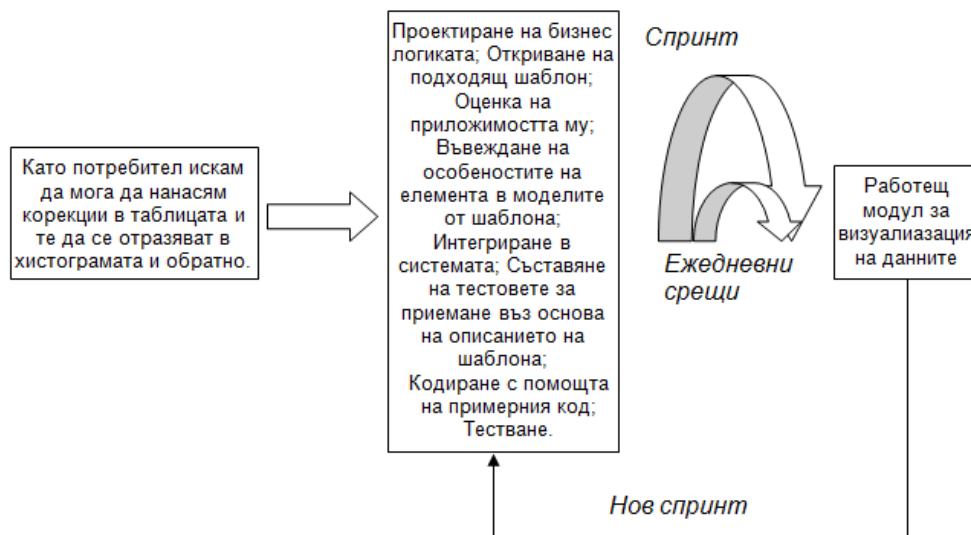
За илюстрация на начина, по който се вписва подходът за прилагане на шаблони в методологията Scrum, разглеждаме една история от продуктивния списък. За изпълнението ѝ се дефинират задачи в списъка на проекта, които избираме да участват в спринтовия

списък на представения спринт³. Примерът е опростен, като са показани задачите само по една история, от изпълняваните в спринта. Задачите са съставени така, че да позволят използването на шаблон за реализацията на функционалността, описана от историята. В случая при реализацията ѝ е подходящо да се използва шаблона за проектиране „наблюдател” на GoF (Гама, 2004) (фиг. 3).



Фиг. 3 Диаграма на класовете на шаблона Наблюдател

Той разглежда таблицата и хистограмата, като наследници на абстрактния клас „наблюдател”, а данните, които се публикуват в тях са наследници на абстрактния клас „обект” (наблюдавания обект). При промяна в наблюдавания обект, наблюдателите се уведомяват. (В случая при промяна в данните, таблицата и хистограмата получават съобщение и обновяват визуализираните данни.) Методологията Scrum допуска въвеждане на задачите, които да позволят използване на шаблон и контролира тяхното изпълнение (фиг. 4).

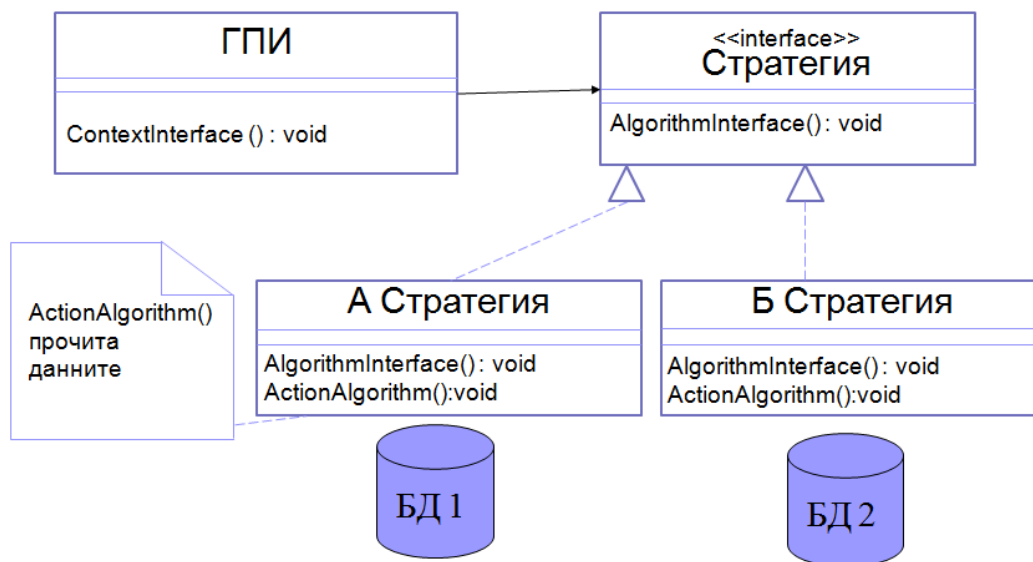


Фиг. 4. Примерни задачи в списъка на спринт, които да позволят използването на шаблон в методологията Scrum

³ Всеки спринт е пълен итеративен цикъл, включващ всички етапи на жизнения цикъл на разработката на информационна система. При него се разработва завършен прототип на част от системата.

В методология Scrum се дава повече власт на разработчиците. Именно затова смятаме, че за да се реализира пълноценно методологията, екипът трябва да има нужните познания, т.е. да е в крак с новостите и да познава възможностите за решаването на един или друг проектен проблем, което се способства от шаблоните. Шаблоните представят възможностите на новите технологии и начините за преодоляване на проблемите им, с което улесняват въвеждането им в проекто-програмното решение.

Канбан (Kanban – сигнална карта) (Atlassian, 2017) е втората много популярна методология за организация и управление в момента според направеното проучване. Тя е не само метод за разработка, но е и насочена към подобряване на дейността в организацията разработчик. Целта ѝ е да се осигури разработката на софтуерния продукт в срок без да се претоварват разработчиците и шаблоните отново могат да подпомогнат процеса с указания и документация. Нейна задача е да се стимулира комуникацията и помощта в екипа и оттам развитието на разработчиците, което се подпомага от шаблоните, чрез дискуссионни въпроси за последствията от прилагането им. Капсулирането на откритите модели в шаблони, прави достойни опита на експертите и за останалите членове на екипа. Принцип на методологията, както и на шаблоните, е да се използват натрупаните знания на участниците в екипа при решаването на задачите. Често разработчиците срещат проблеми с използването на определена технология, като в обектноориентираната чест проблем е установяването на определено поведение в обектите. Например проблем е начинът на реализация на много на брой свързани класове, които се различават само по поведението си. В тази проектна ситуация е подходящо да се използва шаблонът „стратегия” (фиг. 5) на GoF (Гама, 2004).



Фиг.5. Модел на шаблона Стратегия (Strategy)

В Канбан се разчита се на инициативата на всеки един член от екипа, както в поемането на нови задачи така и при подпомагането на колегите. Считаме, че шаблоните отново могат да се много полезни, тъй като капсулират решенията на проблемите. Например типичен проблем е за уведомяването и автоматичното обновяване на зависимите обекти при промяната на главния обект. Решението му е капсулирано в разгледания шаблон

Наблюдател.

Ако един член на екипа вече е решил даден проблем, той може да опише решението, като шаблон. За да бъдат шаблоните лесни за възприемане, считаме че е подходящо да определим общ формат за документирането им. Когато останалите членове на екипа се изправят пред подобен проблем, могат да използват шаблона, за да въведат вече намереното решение.

Има само едно условие при използването на шаблоните, в която и да е методология – да са средство, а не цел. Според Дейнау (Daigneau, 2011) цел на всяка разработка е търсене на оптимално решение на проблема, а не непременно начин да се използва шаблон. Считаме за нужно да се използва подходящ подход, който да гарантира някаква степен на умереност и да противодейства на трудностите при прилагането им. Подходът не е методология и не съдържа средства за управление, планиране и контрол на дейностите си.

След анализа на възможностите за прилагане на шаблоните в методологиите за разработка, стигаме до извода, че шаблоните имат своето място сред средствата за разработка. Интегрирането им в методология не е тривиално, но в RUP и гъвките методологии е възможно да се включи подход за използването им, който да въведе систематичност и повтаряемост. Той трябва да е съвместим с процеса на разработката, да засили предимствата на шаблоните и да реши някои от проблемите на прилагането им. За целта са изследвани особеностите, предимствата и недостатъците им и следва да се потърси начин да се улеснят отделните стъпки в подхода, както и откриването на подходящ шаблон.

4. ЗАКЛЮЧЕНИЕ

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.

Проучването ни дава основание да твърдим, че шаблоните имат свои място и роля в съвременните методологии за разработка на информационни системи. Те могат да подобрят някои аспекти на методологиите. Най-подходящи за включване на подход за използване на шаблони са методологиите, които се основават на подхода UML, поддържат MDD и са достатъчно гъвкави, за да се въведе подход за прилагане на шаблоните. Това са RUP и гъвките методологии.

4. ДОПЪЛНИТЕЛНИ МАТЕРИАЛИ

Приложение 1 и 2, съдържащи въпросите и обобщените отговори от проучването.

5. ЛИТЕРАТУРА

Статия

Манифест за Agile, (2001). [Онлайн] Available at: <http://agilemanifesto.org>;
<http://agilemanifesto.org/iso/bg>

Agile Modeling, (2012). *Agile Modeling and the Rational Unified Process (RUP)*. [Онлайн] Available at: <http://www.agilemodeling.com/essays/agileModelingRUP.htm> (08.08.2017)

Atlassian, (2017). *How the kanban methodology applies to software development*. [Онлайн] Available at: <https://www.atlassian.com/agile/kanban> (08.09.2017)

Griffiths, M., (2016). *Agile Risk Management*. [Онлайн] Available at: http://leadinganswers.typepad.com/leading_answers/2016/11/agile-risk-management.html (19.09.2017).

Griffiths, M., (2017). *Agile 2017*. [Онлайн] Available at: http://leadinganswers.typepad.com/leading_answers/2017/08/agile-2017.html (19.09.2017).

Griffiths, M., (2017). *Agile Consulting*. [Онлайн] Available at: http://leadinganswers.typepad.com/leading_answers/2017/06/agile-consulting.html (19.09.2017).

Sandén, B. I., (2003). *Entity-life modeling: modeling a thread architecture on the problem environment*. н.м.: IEEE Software.

Versionone.com, (2013). *7th annual state of agile Agile Made Easier development survey*. [Онлайн] Available at: <http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf>

Книга

Гама, Е. & Хелм. Р. & Джонсън, Р. & Влсидес, Дж., (2004). *Шаблони за дизайн*. н.м.: СофтПрес.

Филипова, Н. & Филипов, Ф. & Армянова, М. & Филчева, Д. & Василева, Н., (2013). *Интегрирана система за съвместна работа в обучението по разработване на софтуер*. н.м.: Университетско издателство ИУ-Варна, ISBN 978-954-21-0717-0.

Филипова, Н. & Парушева, С. & Александрова, Я., (2017). *Основи на информационните системи*. н.м.: Наука и икономика, ИУ-Варна.

Beck, K. & Andres, C., (2004). *Extreme Programming Explained: Embrace Change*, 2nd Edition (The XP Series). н.м.: Addison Wesley. ISBN-13: 978-0321278654.

Daigneau, R., (2011). *Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services*. н.м.: Addison-Wesley. ISBN-13: 978-0321544209

Eeles P. & Cripps, P., (2009). *The Process of Software Architecting*. н.м.: Addison-Wesley.

Fowler, M., (1996). *Analysis Patterns*. н.м.: Addison-Wesley Professional.

Fowler, M. & Beck, K. & Brant, J. & Opdyke, W. & Roberts, D., (2002). *Refactoring: Improving the Design of Existing Code*. н.м.: Addison-Wesley.

Hanmer, R., (2007). *Patterns for Fault Tolerant Software*. н.м.: Wiley.

Rational Software, 1998. *Rational Unified Process*, н.м.: Rational Software. Available at: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf (06.09.2017)

Schumacher, M. & Fernandez, E. B. & Hybertson, D. & Buschmann, F. & Sommerlad, P., (2006). *Security Patterns: Integrating Security and Systems Engineering*. н.м.: John Wiley & Sons.
Schwaber, K. (2004). *Agile Project Management with Scrum*. н.м.: Microsoft Press.

6. СЪКРАЩЕНИЯ

1. MDD – Model Driven Development, подход за разработка основан на модели.
2. MDA – Model Driven Architecture, архитектура основана на модели.
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.