СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ”
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

Автореферат

на дисертация за присъждане на образователна и научна степен „Доктор”

по научна специалност 01.01.12. „Информатика”

на тема:

Гъвкава методология за разработка на софтуерни приложения

Докторант: Ива Кръстева

Научен ръководител: доц. д-р Силвия Илиева

София, 2011 г.

Съдържание

1. Въведение

С нарастващото развитие на софтуерната индустрия през 80-те години на миналия век е била идентифицирана нуждата от прилагане на систематизиран и повторяем подход за разработването на софтуер [1]. Използването на методи при разработката на софтуер е идентифицирано като начин за подобряване на процеса на разработка, повишаване на качеството на продукта и намаляване на средствата за поддръжката му [2]. В последните години гъвкавата софтуерна разработка (ГСР) бележи покачващ се интерес както в академичните среди, така и в индустрията [3] [4]. Въпреки широкото приложение на методологиите за ГСР, е идентифициран проблемът за ниската степен на повторяемост при внедряването [5]. Съобразяването на прилагането на методологията с характерните особености на проектите е основна стъпка за повишаването на степента на повторяемост при внедряването на методологиите за разработване на софтуер. Нуждата от структуриране на знанието, използвано при внедряване на гъвкавите методологии, и появата на методологии, дефиниращи повторяем подход за създаване на методи за разработка на софтуер, е идентифицирано като предизвикателство пред настоящите изследвания в областта на ГСР и насока за бъдещо развитие [5] [6] [7].

1.1 *Цел и задачи на дисертацията*

Настоящата работа има за цел да предостави решение на представените предизвикателства и проблеми пред ГСР. *Целта на научния труд* е да предложи, опише, реализира и валидира гъвкава методология за разработване на софтуерни приложения, която дефинира повторяем подход за създаване на методи за разработка, основава се на предходен опит и е реализирана върху отворени стандарти и метамодели с индустриално приложение

Дефинирани са следните *задачи* за постигане на целта:

Задача 1: Да се изследва прилагането на гъвкавите методологии и практики в софтуерните компании и да се идентифицират проблемите и потенциалните възможности за разширяването му. Да се направи обзор на съществуващите изследванията с практическа насоченост в областта на ситуационното конструиране на методи и тяхното прилагане в областта на гъвкавите методологии.

Задача 2: Да се създаде нова гъвкава методология за разработка на софтуерни приложения, която използва предходен опит при конструирането на методи за разработка, съобразени с характеристиките на конкретния проект, в който се използват. За целите на методологията :

- да се предложи подход за управление на знанието, придобито при предходно внедряване на гъвкави методологии
- да се дефинира набор от фактори за описание на характеристиките на софтуерните проекти
- да се идентифицира подходящ метамодел за представяне на метода на разработка и да се разшири за нуждите на новата гъвкава методология.

Задача 3: Да се специфицира автоматизирана среда за реализиране на методологията, базирана на отворени стандарти и метамодели с индустриално приложение.

Задача 4: Да се валидира предложената гъвкава методология чрез пример за използването й.

1.2 *Структура на дисертацията*

Дисертацията е структурирана в четири глави, от които първата уводна, и заключение. Към работата са добавени 6 приложения, списък с 11 публикации във връзка с дисертацията, 5 други публикации с участието на докторанта, декларация за оригиналност и цитирана литература от 118 източника. В текста на дисертацията са използвани 13 таблици и 53 фигури.

Във втора глава на дисертационния труд е направен обзор на проблемната област. Представено е приложението на гъвкавото разработване на софтуер, гъвкавите методи и практики в работата на софтуерните компании. Идентифицирани са предизвикателствата пред приложението на гъвкавите методологии. Ситуационното конструиране на методи е залегнало в основата на предложената методология за разработка на софтуерни приложения. В главата е включено представяне на дисциплината. Изследвани са съществуващите научни разработки в областта на ситуационното конструиране на методи с практическо приложение.

Трета глава на дисертацията представя предложената гъвкава методология и описва ситуационния подход за конструиране на методи, който е използван в методологията. Разгледани са основните дейности в процеса за конструиране на методи, предложен в методологията. Описани са концепциите и понятията, които се използват в методологията. В главата е представена структурата и използването на базата от знания, която участва в първата основна дейност в процеса за конструиране на методи, използван в методологията - избора на стандартни гъвкави практики подходящи за използване в конкретен проект. Описани са данните и информацията, която се съхранява в базата и анализите, на чиято база се прави избора на практиките. Втората и третата основни дейности- конструирането и адаптирането на създадения метод за разработка на софтуер също са представени в главата. Описана е спецификация на метамодела, който се използва при конструирането и адаптирането на метода за разработка. Представени са правила за непротиворечивост и цялостност, които са част от оценяването на метода. Предложена е също и оценка на степента на гъвкавост на метода на разработка.

В четвъртата глава е представена реализацията на методологията чрез подходяща среда. Описана е функционалната спецификация на два инструмента, които участват в средата. В главата е включена и валидация на представената гъвкава методология. Използвани са два подхода за валидация – оценка чрез съпоставяне с описателен модел и пример за използване. Направени са изводи от резултатите, получени от използването на методологията в примерния проект.

В заключение е направено обобщение на работата и са набелязани насоки за бъдещо развитие и подобрения на предложената гъвкава методология.

1.3 *Използвани означения и инструменти*

Английските понятия и наименования са въведени в кръгли скоби след първото срещане на понятието на български език. По същия начин са въведени и абревиатурите.

Наклонен шрифт е използван за да се акцентира върху част от текст, която е важна за разбирането на текста или въвежда важно понятие.

С **удебелен и подчертан шрифт** е означено началото на секция в текста, която прекъсва повествованието на текста и въвежда важно понятие.

При създаването на част от диаграмите е използван инструмента MagicDraw UML с академичен лиценз, поради което на фигурите има печат.

Номерацията на фигурите е както в дисертацията.

2. Резюме на съдържанието на дисертацията

Глава 2. Обзор на проблемната област

В глава Глава 2 е направен обзор на изследванията в областите, които са застъпени в най-голяма степен в научния труд – гъвкавата софтуерна разработка и ситуационното конструиране на методи, и е представена мотивацията за създаване на нова гъвкава методология.

2.1. Терминология и основни понятия

ДЕФИНИЦИЯ: *Процес на разработване на софтуер* е набор от дейности за разработване, управление и организация на един софтуерен проект, както и тяхната последователност и взаимовръзки

ДЕФИНИЦИЯ: *Метод за разработване на софтуер* е интегриран набор от процедури и техники, които предоставят ефективен, ефикасен и консистентен подход за разработване на софтуерни системи, както и използваните инструменти, резултантните продукти и артефакти, участващите субекти и процеса на разработка.

ДЕФИНИЦИЯ: *Конструиране на методи* е софтуерна дисциплина за проектиране, реализиране и оценяване на методи за разработване на софтуер, както и на свързаните с тях техники и инструменти.

ДЕФИНИЦИЯ: *Методология за разработване на софтуер* е описаният систематизиран и повторяем подход за създаване, представяне и оценяване на методи за разработване на софтуер.

ДЕФИНИЦИЯ: *Ситуация* на даден проект (също и *проектна ситуация*) е комбинацията от типа на проекта и контекста на проекта.

ДЕФИНИЦИЯ: *Контекст на проект* е външната среда, в която се развива проекта, и оказва влияние върху него.

ДЕФИНИЦИЯ: *Контекстни фактори* за даден проект е набор от характеристики на средата, в която се развива проекта.

ДЕФИНИЦИЯ: *Ситуационни фактори* за даден проект е набор от характеристиките на проекта и контекстни фактори.

Следват свързаните дефиниции на ситуационен метод и ситуационно конструиране на методи:

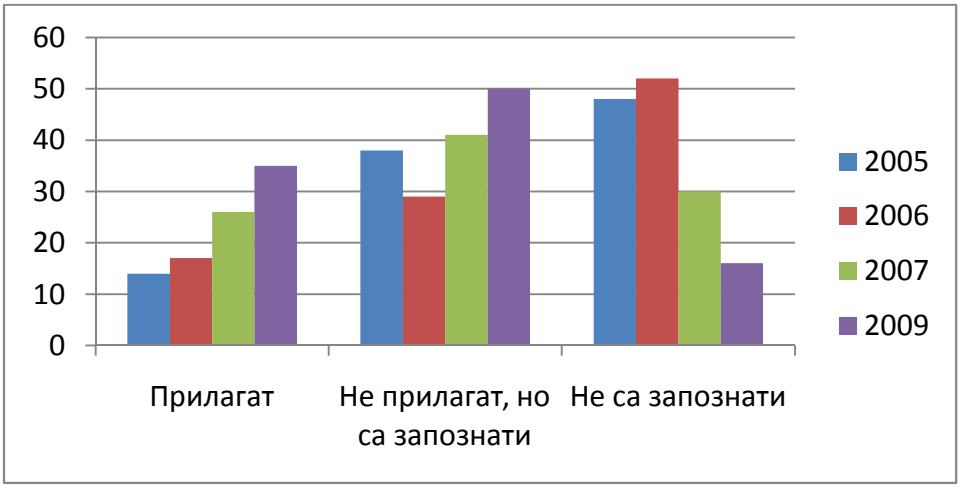
ДЕФИНИЦИЯ: *Ситуационен метод* е метод за разработване на софтуер, който е пригоден за определена проектна ситуация.

ДЕФИНИЦИЯ: *Ситуационно конструиране на методи* е подразделение на дисциплината Конструирането на методи, което се занимава с конструирането на ситуационни методи.

2.2. Гъвкава софтуерна разработка

Гъвкавата софтуерна разработка се появява като алтернатива на традиционните план-базирани методологии за разработване на софтуер в съвременната високо динамична и конкурентна бизнес среда. Хайсмит [8] определя гъвкавостта като способността да се създават промени и да се реагира на тях, за да се извлече полза в турбулентна бизнес среда. Ценностите и принципите, които са залегнали в ГСР, са фокусирани към разработването на софтуер в среда с бързо променящи се изисквания и

намалено време за разработване и внедряване на продукта. Приложението на гъвкавата разработка в практиката бележи постоянен ръст през последните години. По данни на Forrester Research цитирани от Rally Software [4] компаниите, които прилагат гъвкав подход за разработка на софтуер са се увеличили с повече от 20% в периода от 2005 до 2009 година. Компаниите, които не прилагат, но са запознати с гъвкавото разработване на софтуер, са се увеличили до 50% през 2009, а тези, които не са запознати са намалели с повече от 30%. Данните от проучването са показани на Фигура 1.1.



Фигура 1.1 Приложение на гъвкавата разработка в софтуерните компании

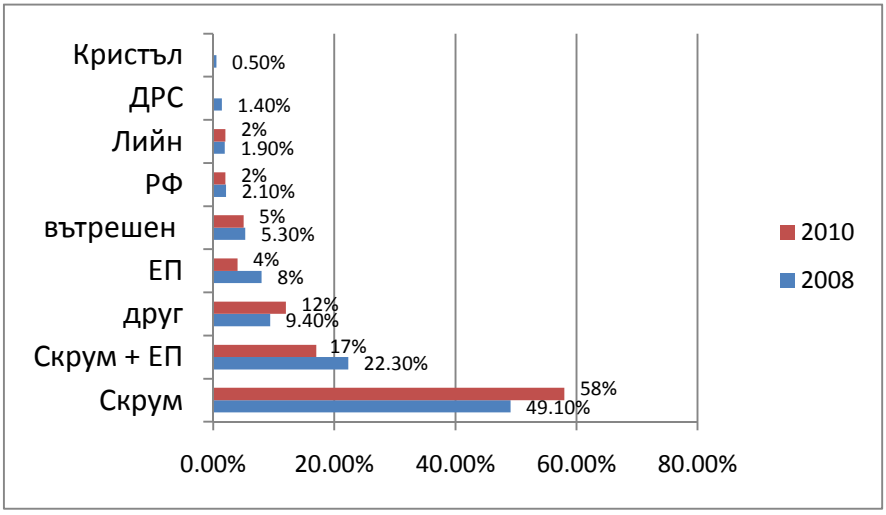
2.2.1. Принципи и ценности на гъвкавото разработване на софтуер

През 2001 година 17 практикуващи софтуерни инженери се събират и публикуват Манифест за Гъвкаво Разработване на Софтуер (Manifesto for Agile Software Development) [9] (наричан още Гъвкав Манифест). В манифеста са описани основните принципи и ценности на гъвкавата разработка. Заглавната страница на Гъвкавия Манифест е представена на Фигура 1.2.

Фигура 1.2 Манифест за Гъвкаво Разработване на Софтуер

2.2.2. Гъвкави методологии за разработване на софтуер

Методологиите за разработване, които следват принципите и стойностите, дефинирани в Гъвкавия Манифест, са определени като гъвкави методологии за разработване на софтуер. Голяма част от гъвкавите методологии се използват от част от създателите на Гъвкавия Манифест още преди официалното публикуване на манифеста през 2001 година. Приложението на отделните гъвкави методологии в практиката може да бъде проследено на Фигура 1.3, която представят данни от проведени от компанията VersionOne изследвания през 2008 и 2010 година [10] [11]. Изследванията представят данни от интернет въпросници с няколко хиляди участници (3061 участници през 2008 година и 4770 през 2010 година) от повече от 80 страни. Скрум (Scrum) е най-широко прилаганата методология с около 50 процента приложение с тенденция към разширяване на приложението - приблизително 10 процента от 2008 до 2010 година. Хибридът между Скрум и Екстремното програмиране (Extreme programming) заема второто място по приложение. След тях с проценти между 10 и 2 се нареждат Екстремното програмиране (ЕП), Разработването по функционалности (Feature-driven development) и Лийн (Lean). Динамичното разработване на софтуер (Dynamic systems development method) и фамилията Кристъл (Crystal family) намират приложение в по-малко от 2% от проектите през 2008 година. Данни за последните две методологии за 2010 година липсват.



Фигура 1.3 Приложение на гъвкавите методи през 2008 и 2010 година

2.2.3. Практики и техники на гъвкавите методологии

Секцията представя кратък обзор на практиките и техниките описани в спецификациите на шестте гъвкави методологии, които намират приложение в индустрията. В десертацията се използват:

- Тринайсетте практики на методологията ЕП
- Седем практики на Скрум методологията
- Шестнайсет практики от двете методологии от фамилията Кристъл – Кристалния кристал (Crystal Clear) и Оранжевия кристал (Crystal Orange)
- Осем практики на методологията Разработването по функционалности
- Осем практики на методологията Динамичното разработване на софтуер
- Двадесет и две практики на Лийн методологията за разработване на софтуер

Данни за приложението на най-често използваните гъвкави практики през 2007, 2008 и 2010 година са представени в изследванията, описани в [10] [11]. Най-

използвани са практиките на Скрум и ЕП, като в последната година се вижда увеличаване на използването на техники от Лийн методологията. Като цяло се забелязва тенденция към разширяване на използването на практиките като само някои практики от ЕП бележат лек спад в използването.

2.2.4. Предизвикателства пред прилагането и направления за развитие на гъвкавите методологии

В сравнителен анализ на разпространените гъвкави методологии Абрахамсон [5] поставя въпроса за нуждата от адаптиране на метода за разработка за всеки отделен проект и извежда необходимостта от дефинирането на ясни насоки за извършването на тази адаптация. Освен при адаптирането на методите за разработка, съществуват предизвикателства пред разширяване на областта на приложение на гъвкавите методологии.

Използването на структурирано знание в явен вид, с възможност за повторно използване, е идентифицирано като възможна насока за преодоляване на предизвикателствата пред приложението на гъвкавите методологии и разширяването на областта на тяхното приложение. Втората насока за преодоляване на идентифицираните предизвикателства е съобразяването с конкретните характеристики на проектите и изготвянето на насоки за внедряване на методите за разработка, базирани на тези характеристики. В [5] е предложено използването на дисциплината ситуационно конструиране на методи като обещаващ подход за адаптирането на методите към конкретните характеристики на проектите.

Като общо заключение и глобална насока за бъдещо развитие на изследванията в областта на ГСР в [5] е *определена необходимостта от създаването на малко и качествени методологии, които удовлетворяват множество ситуации на приложение*, а не много методологии с ограничено приложение.

2.3. Конструиране на методи

Нуждата от адаптиране на метода на разработка за всеки отделен проект е идентифицирана не само в областта на ГСР [12]. Дисциплината *Конструиране на методи* систематизира изследвания в посока проектиране, създаване и адаптиране на методи за разработване на софтуер. Един от най-важните изследователски въпроси при конструирането на методи е създаването метод за разработка, който е пригоден за нуждите на конкретна проектна ситуация, т.нар. ситуационен метод [13]. За изследванията, свързани със ситуационните методи, се оформя под-дисциплина, наречена ситуационно конструиране на методи (СКМ). Именно СКМ е обект на разглеждане в настоящия труд.

2.3.1. Основни подходи за ситуационно конструиране на методи

Съществуват няколко основни подхода за конструиране на ситуационни методи. *Подходът, базиран на асемблиране* (assembly-based approach) [14] се появява още с първоначалните изследвания в областта на конструирането на методи. Същността на подхода е изграждането на метода за разработка чрез асемблиране на подходящи части от съществуващи методи, които могат да бъдат адаптирани за конкретния метод. Подходът за конструиране на методи на база на асемблиране се развива от повече от 15 години и е намерил частично приложение в практиката поради факта, че формализираното дефиниране и поддържането на голям брой части на методите, както и тяхното настройване и асемблиране, не са тривиална задача и изискват специфични познания в областта. Разработките в тази област са реализирани на практика в близко

сътрудничество с научните институти, където преимуществено се развиват тези изследвания [15].

Вторият подход е *подходът, базиран на разширение,* (extension-based approach). Този подход е предложен и изследван в работите на Денекере [16]. Подходът се базира на използването на подходящи предварително дефинирани шаблони, чрез които да бъде разширен даден метод за конкретна ситуация. Няма сведения за приложението на този подход в чист вид индустрията.

Подходът, базиран на парадигма, (paradigm-based approach) е най-обхватния от всички подходи, тъй като в основата му е залегнало мета-моделирането като основна техника. Идеята на подхода е създаването на методи чрез създаване на инстанция на метамодел или чрез създаване на абстракция на съществуващ такъв. Подходът, базиран на парадигма, се използва като базов в множество подходи за ситуационно конструиране на методи. Сведенията за приложението му в чист вид са ограничени.

Метамоделирането е основна техника в подходите за ситуационно конструиране на методи. При подхода, базиран на асемблиране, частите на метода се представят като инстанции на определен метамодел, който позволява модулно представяне на моделите. Подходът, базиран на разширяване, използва метамоделите за да описва както базовия метод, така и шаблоните за разширяване. Подходът, базиран на парадигма, по дефиниция е базиран на метамоделирането.

2.3.2. Метамодели за описание на методи за разработване на софтуер

Метамоделите на методи, служат за описване както на модели на продуктите, така и на процесите и хората, участващи в метода на разработка [17]. Метамоделите на продуктите дефинират основните концепции и понятия от областта на разработвания продукт, както и основните артефакти, които се създават по време на разработката - документи, пакети, диаграми и др.

Процесните метамодели могат да бъдат разделени в следните категории в зависимост от различните гледни точки за представяне на процеса на разработка - дейности, продукти, решения, контекст и стратегия [18] [19]:

- Метамодели, ориентирани към дейности

Метамоделите, ориентирани към дейности, се използват за описание на модели, представящи дейности и задачи, и тяхната последователност. Най-широко разпространените метамодели, ориентирани към дейности, са SPEM 2.0 [20], The Open Process Framework [21] [22], ISO 24744 [23]. И трите метамодели са стандартизирани.

- Метамодели, ориентирани към продукти

Ориентираните към продукти метамодели се използват за модели, които представят състоянието на продукт в определен момент от процеса на разработка и дейностите, в резултат на които се стига до това състояние. Метамоделите, базирани на автомати (State Machines) и на диаграми на състоянията (Statecharts) са примери за метамодели, ориентирани към продукти.

- Метамодели, ориентирани към решения

Метамоделите, ориентирани към решения, се използват за представянето на успешна трансформация на продукт на база на решения. Метамодел, ориентиран към решения, е представен още през 1970 година, но тези метамодели не са получили голямо разпространение.

- Метамодели, ориентирани към контекст

Метамоделите, ориентирани към контекст, позволяват създаването на модели, които описват ситуация по време на изпълнението на проекта и намеренията на определен *участник* (actor) в този момент за постигане на дадена цел. Метамодел,

ориентиран към контекст, който се използва и в някои от разработките, свързани с конструирането на ситуационни методи, е NATURE [19].

- Метамодели, ориентирани към стратегии

Метамоделите, ориентирани към стратегии, се използват за описание на модели, които представят различни възможни начини за постигане на определена цел, свързана с разработването на дадена система. Пример за метамодел, ориентиран към стратегии, е MAP [24].

2.3.3. Инструменти и среди

Инструментите и средите, които подпомагат конструирането на методи, са известни под името CAME (Computer Aided Method Engineering) [25]. Изискванията към тях са различни от изискванията към тези, използвани да осигуряват компютъризирана поддръжка на софтуерното инженерство – т.нар. CASE инструменти (Computer Aided Software Engineering tools). CAME инструментите се използват за дефиниране на методи, което обхваща асемблирането на части на метода и адаптирането на съществуващ метод. Тези дейности се извършват от инженерите по конструиране на методи (method engineers). В CAME средите могат да бъдат интегрирани и съществуващи CASE инструменти, които осигуряват поддръжка за създадения метод.

Никнафс [15] прави обзор на седем CAME среди – Decameron, MENTOR, MERET, MERU, MetaEdit+, MethodBase, MethodEditor. Всички среди са създадени и се използват като научни разработки като единствено MetaEdit+ има и търговско приложение. От сравнителния анализ, направен в обзора, се вижда също така, че четири от седемте среди поддържат подхода, базиран на асемблиране. Само една от средите (MENTOR) осигурява пълна поддръжка за моделиране на процесната част на метода чрез специално разработен семантичен език. Останалите среди са базирани върху продуктови метамодели. Задоволително ниво на наличната документация за средите е осигурена само от MetaEdit+.

Търговски разработки на инструменти, които поддържат конструирането на методи за разработване на софтуер и имат характеристики на CAME среди, са разработеният от IBM търговски продукт - Rational Method Composer [26], и инструментът с отворен код EPF Composer [27].

2.3.4. Съществуващи научни разработки с практическа насоченост

В обзора на съществуващите научни разработки с практическа насоченост са включени следните изследвания:

- Методът за конфигуриране на методи [28]
- Подходът за конфигуриране на процеси [12]
- Ситуационните подходи, интегриращи знание [29] [30]
- Подходи, базирани на SPEM 2.0 метамодела [31] [32]

Резултатите от обзора показват, че има решения, базирани както на специално разработени метамодели, така и на отворени стандартизирани метамодели. Съществуват разработки, които интегрират знание при създаването на метода на разработка. Изследванията в областта на гъвкавите методологии и СКМ все още са малко.Основен недостатък на всички подходи е, че реализацията е извършена чрез използването на вътрешни CAME среди, което ограничава тяхното разпространение. Няма данни за тяхното по-широко използване в софтуерните компании.

2.4. Заключение

Изводите от разглежданията в настоящата обзорна глава са обобщени в текущия параграф. От направения преглед на ГСР беше установена нуждата от създаване на нова гъвкава методология, която да използва предходен опит и да бъде приложима в широк кръг от софтуерни проекти. Ситуационните подходи за конструиране на методи са подходящи за създаване на методи за разработка адаптирани към конкретна проектна ситуация. Разпространението и прилагането на съществуващите подходи за СКМ е ограничено поради използването на тежки формални подходи или вътрешни разработки на метамодели и CAME среди. Предложената гъвкава методология трябва да дефинира нов ситуационен подход за конструиране на методи на разработка, който е приложим в работата на софтуерните компании и е базиран на отворени стандарти и метамодели с индустриално приложение.

В глава Глава 2 е *представено решението на първата от задачите* на работата:

Задача 1: Да се изследва прилагането на гъвкавите методологии и практики в софтуерните компании и да се идентифицират проблемите и потенциалните възможности за разширяването му. Да се направи обзор на съществуващите изследванията с практическа насоченост в областта на ситуационното конструиране на методи и тяхното прилагане в областта на гъвкавите методологии

Глава 3. Гъвкава методология за разработване на софтуер

Трета глава описва най-съществената част от научния труд, в която се представя нова гъвкава методология за разработване на софтуер. Характеристиките на новата методология отразяват направените изводи при обзора на проблемната област и могат да бъдат обобщени по следния начин:

- Използва систематичен и повторяем подход за създаване на метод за разработка на софтуер
- Базира се на предходен опит в прилагането на гъвкави практики в софтуерни проекти
- Дефинира метамодел за описание на методи за разработка на софтуерни приложения, базиран на стандартизиран метамодел
- Включва правила за оценяване на разработения метод
- Реализирана е чрез инструменти с отворен код и индустриално приложение и е осигурена съвместимост с други инструменти

Основната цел на предложената методология е да бъде използвана от инженерите по конструиране на методи в софтуерните компании при създаването и адаптирането на най-подходящия за конкретния проект метод за разработка. Затова при създаването на методологията са следвани принципите за лекота, изпълнимост и висока степен на автоматизация.

3.1. Подход

Подходът за създаване на метод за разработка, предложен в настоящата работа, е основан на принципите на ситуационното конструиране на методи и основния подход за конструиране на методи чрез асемблиране. Като ситуационен подход за конструиране на методи се цели създаването на най-подходящия за конкретен проект метод на разработка. Чрез автоматизиране на задачите за избор на части на метода и повишаването на нивото на грануларност на използваните части е понижена сложността на основния подход. Подходът има три основни дейности. Първата е избор на подходящи гъвкави практики на база на данните за приложимост на практиките в

подобни външни проекти. Създаването на метод за разработка и неговото адаптиране са съответно втората и трета основни дейности в подхода.

В подхода, предложен в дисертацията, се събира, структурира, запазва, анализира и поддържа знание за внедряването на гъвкави методологии както от външни компании, така и от самата компания. Информацията за приложението на гъвкавите практики в различни външни проекти се съхранява в специално хранилище. Данните се събират от различни източници като публикувани статии или разпространени по друг начин материали, описващи прилагането на гъвкави практики в софтуерни проекти. Проектите се описват чрез предварително дефиниран набор от ситуационни фактори, които еднозначно определят ситуацията на проект. В хранилището се съхранява информация и за теоретичната съвместимост на практиките една с друга и теоретични ограничения за приложението на дадена практика при наличието на определени фактори на проекти. Хранилището на проектите е реализирано като база от знание, в която анализът на данните е направен на база на качествени и количествени подходи.

Внедряването на гъвкави практики в метода за разработка в дадена организация трябва да се извършва поетапно като практиките постоянно се подобряват и адаптират за нуждите на конкретния проект. Предложеният подход използва итеративен и инкрементален процес за постоянно адаптиране и подобряване на метода на разработка към конкретния проект като дава възможност на конструктора на методи да променя и настройва метода за всяка итерация в проекта. Адаптацията към характеристиките на проекта се извършва чрез възможността за избор на практики, прилагани в предходни проекти на организацията с подобни характеристики. Методът за разработка се описва чрез създадено разширение на стандартната инстанция на SPEM 2.0 метамодела - SPEM 2.0 Base Plug-in. Чрез подходящо надграждане на функционалността на инструмента с отворен код EPF Composer, който осигурява до голяма степен поддръжка за стандартната инстанция, се осигурява изпълнимостта на предложения подход.

Предложеният в настоящата работа ситуационен подход за конструиране на методи ще бъде реферирan като подход за създаване на методи за разработка, базиран на предходен опит, или EXPERIMENtER (EXPERIence-based Method ENgenERing approach).

3.2. Процес за конструиране на методи

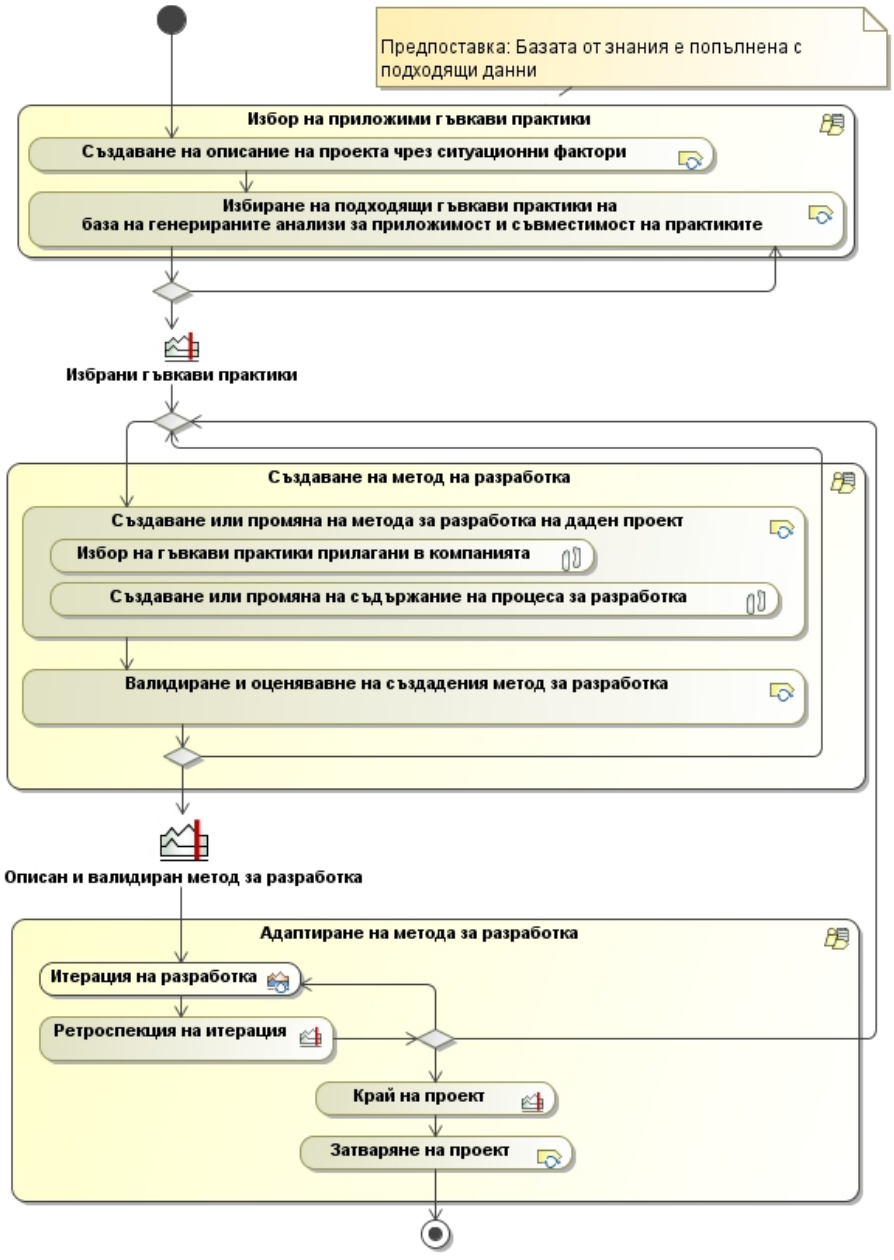
Процесът за конструиране на метод за разработка е описан чрез модела, представен на Фигура 1.4. Три основни дейности се извършват за изграждане на метода на разработка. Първата дейност - *избор на гъвкави практики за разработка*, съдържа две основни задачи:

- описание на ситуацията на проекта чрез предефинирани ситуационни фактори
- избор на гъвкави практики от изградената база от знания, които са най-подходящи за дадения проект.

Втората основна дейност е *създаването на метода за разработка за конкретен проект*. Тъй като процесът за конструиране на методи е итеративен, то задачите в тази дейност се отнасят както за създаване на началното съдържание, така и за промяна и адаптиране на това съдържание по време на изпълнението на проекта. Една основна стъпка при създаването на метода е изборът на подходящи гъвкави практики, използвани в организацията в други проекти. След като е описан метода за разработка се прави валидация чрез оценка на непротиворечивост и цялостност. Освен това се измерва степента на гъвкавост на база на участващите практики в проекта. Методът на

разработка е описан формално чрез метамодел, който е разширение на стандартната инстанция на SPEM 2.0 метамодела за софтуерни проекти - SPEM 2.0 Base Plug-in [20].

Създаденият метод за разработка се променя и настройва по време на изпълнението на проекта. *Подобряване на метода на разработка и неговото адаптиране* за конкретния проект е третата основна дейност. Подобряването се извършва на база на проведените ретроспективни срещи в края на итерациите на проекта. След края на проекта се прави въпросник, който се попълва от всички участници в проекта.

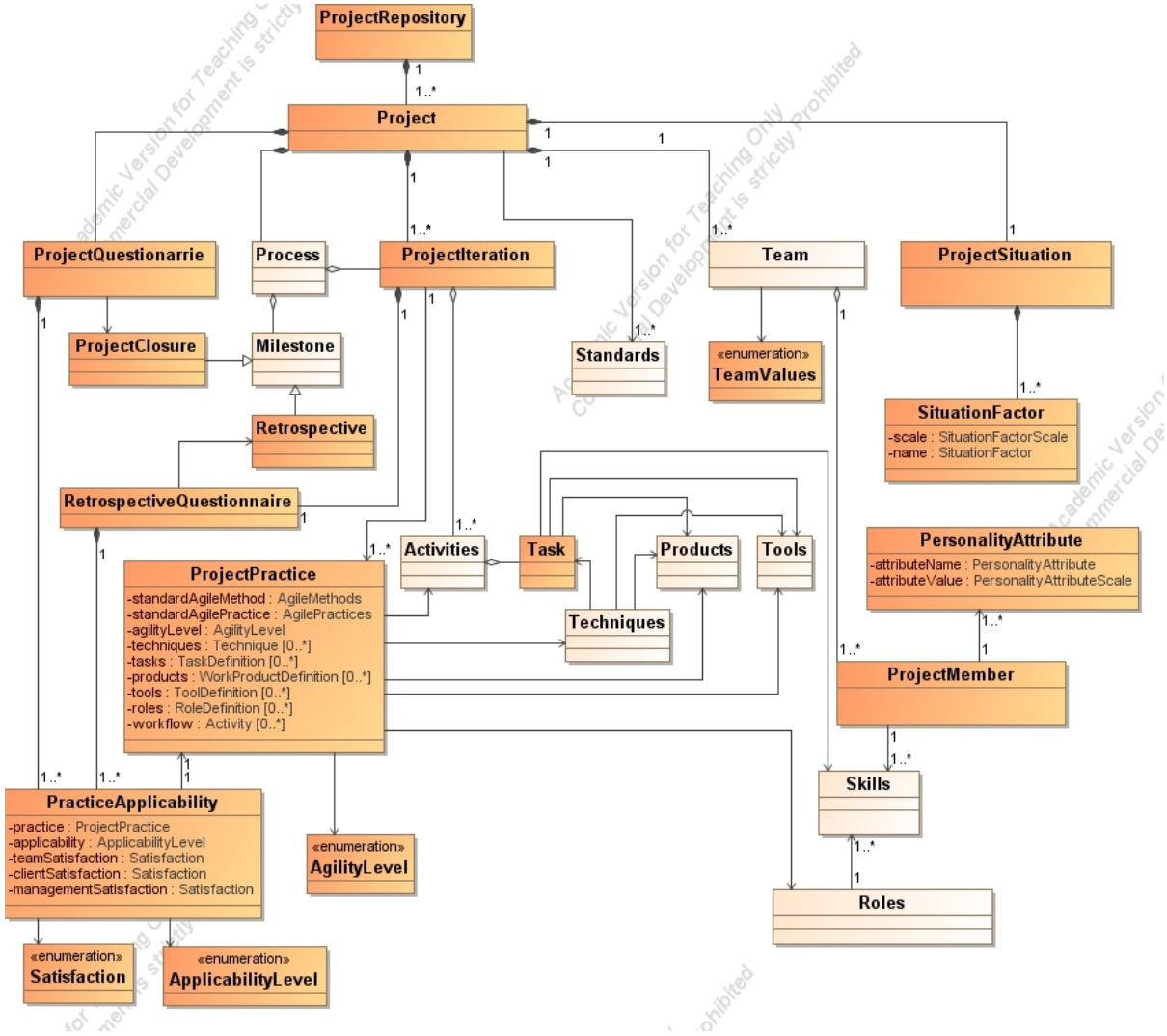


Фигура 1.4 Процес на подхода за конструиране на методи EXPERIMENtER

3.3. Основни концепции

Вторият основен артефакт, използван при описанието на подходите за конструиране на методи, е моделът на основните концепции и понятия, които се използват от методологията и които са входящи или изходящи елементи от дейностите и задачите в процеса за конструиране на метода. Концепциите, които са представени в

модела, са съвкупност от стандартни елементи за описване на методологии, дефинирани от Кобърн [33], основни понятия за ситуационното конструиране на методи, както и концепции, използвани в конкретния подход за конструиране на методи. Основните концепции, връзките между тях и между стандартните елементи на методологиите са описани на Фигура 1.5. Добавените концепции са в по-тъмен цвят. В по-светъл цвят са представени стандартните елементите на методологиите.



Фигура 1.5 Допълнителни концепции за реализиране на подхода за създаване на методи на разработка, базиран на предходен опит

3.4. Изграждане на база от знания

Първата основна стъпка в процеса за конструиране на метода за разработка на софтуер на методологията, представена в дисертацията, е изборът на приложими в проекта гъвкави практики. В специално изградена за целите на методологията база от знание се събира, структурира, запазва, анализира и поддържа знание за приложимостта на гъвкавите практики в различни проекти ситуации.

Основният анализ, който се прави на основа на информацията в базата от знания, е да се намерят онези множества от гъвкави практики, които са най-приложими в дадена проектна ситуация. Тези множества се подлагат на три допълнителни анализа. Първият анализ проверява теоретична неприложимост на практиките в множествата. Вторият анализ проверява за дадено множество от практики какво е покритието на

отделните дейности, присъстващи във всеки процеса за разработка на софтуер [34], както и за съпътстващите дисциплини като управление на проекта, управление на конфигурациите и управление на качеството. Множествата от практики се анализират и за поддръжката, която осигуряват за стойностите и принципите на гъвкавите методи, описани в Гъвкавия Манифест.

3.4.1. Ситуационни фактори

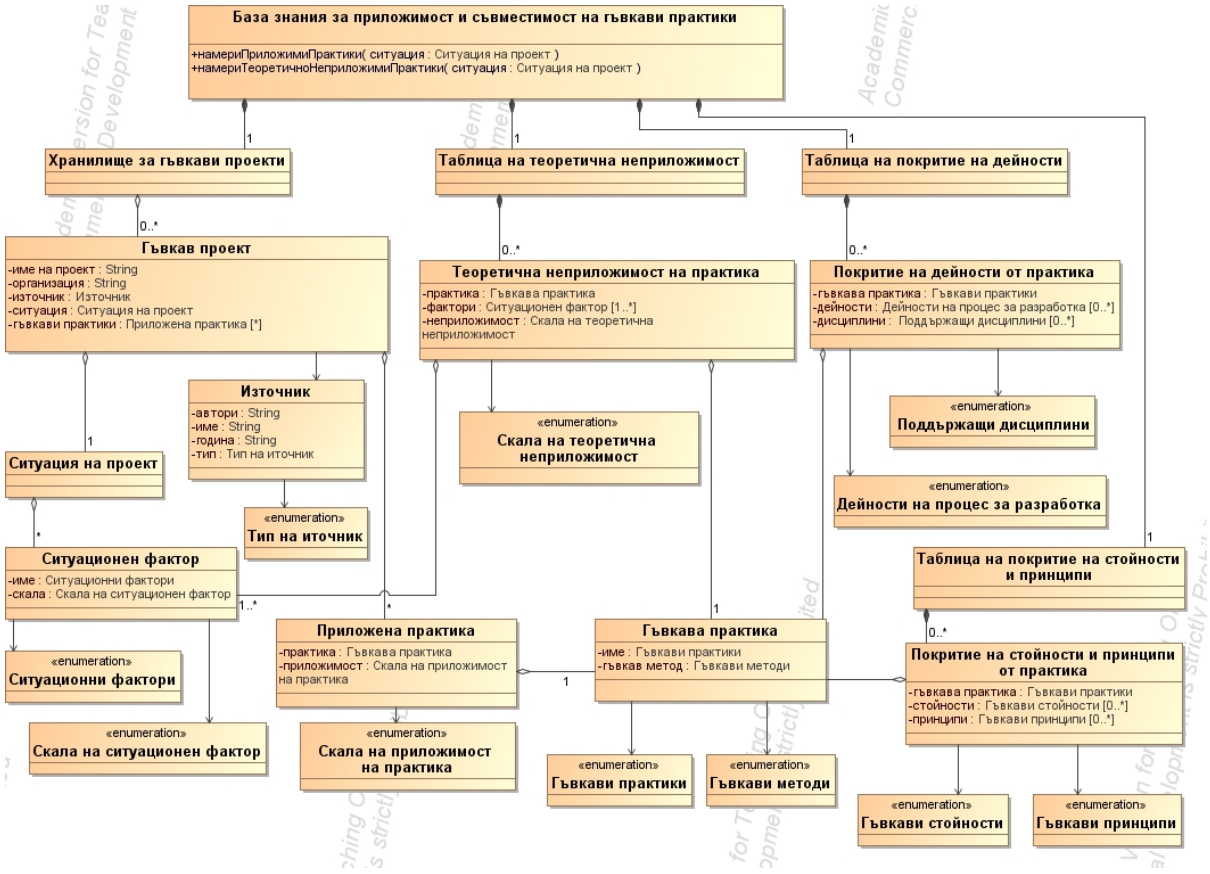
Секцията описва процедурата, използвана за да се идентифицират ситуационните фактори, които служат за описание и категоризиране на софтуерни проекти и които оказват влияние на използването на определени гъвкави практики. До момента не е известна стандартизирана таксономия за описване и категоризиране на софтуерни проекти по техни общи характеристики. В резултат на предложената в работата процедурата са идентифицирани 47 ситуационни фактора, от които осем попадат в категория Контекст, а останалите 39 – в категория Тип на проекта. Дефинирани са и осем подкатегории. Процедурата за идентифициране на ситуационните фактори, както и скалите за оценката им, са описани в детайли в [35].

3.4.2. Структура на базата от знания

Изборът на приложими за даден проект гъвкави практики се извършва от база от знания за приложимостта на гъвкави практики. Базата съдържа няколко елемента чрез които е организирано съхранението на информацията в нея. Първият елемент е хранилище на проекти, което съдържа описания на проекти. Всеки проект в хранилището е представен чрез името си, организацията, в която е приложен, източника, от където е взета информацията, ситуацията на проекта и приложените в проекта гъвкави практики. Източниците на информация могат да бъдат два типа - публикувани репорти или опит от реална експлоатация. Всяка гъвкава практика се описва чрез името си и гъвкавият метод, в който е специфицирана. Ситуация на проект е представена чрез предефинирани набор от ситуационни фактори и техните стойности за съответния проект. Приложението на дадена гъвкава практика в проекта е описана със скала, която определя до каква степен е била приложена практиката в проекта.

В базата знания се съхранява информация за теоретичната неприложимост на гъвкава практика при наличието на определени фактори на проекти. За измерване на теоретичната неприложимост на практика е дефинирана скала, която задава степента в която една практика е неприложима при определен ситуационен фактор. Измерването на теоретичната неприложимост на гъвкавите практики е извършено на базата на съществуващи теоретични изследвания в областта.

Таблицата на покритие на дейности съхранява информация за основните дейности на процеса на разработка и поддържащите дисциплини, за които всяка практика предоставя поддръжка. Тази информация се използва за да се проследи за избран набор от подходящи за прилагане практики какво е покритието му върху основните дейности на процеса на разработка и съпътстващите го дисциплини. Подобен анализ се извършва и за покритието на множество от практики върху гъвкавите стойности и принципи, дефинирани в Гъвкавия Манифест. Таблицата за покритие на стойности и принципи съдържа информация за покритието, осигурено от всяка една от гъвкавите практики, на гъвкавите стойности и принципи. Структурата на базата знания е представен на Фигура 1.6.



Фигура 1.6 Структура на базата знания за приложимост на гъвкави практики

3.4.3. Избор на гъвкави практики

Как да се изберат гъвкави практики, които са най-подходящи за приложение в конкретен проект, е важен научноизследователски въпрос. За решаването му в настоящата работа са приложени няколко различни анализа на данните в базата от знания. При изборът на гъвкави практики са използвани качествените подходите за извличане на данни на база на подобие, класифициране и представяне чрез метрики за сумарни количества. Количествен анализ на база на статистически подход е добавен с цел да се увеличи прозрачността на анализа за подобие.

Като първа стъпка от избора на гъвкави практики, подходящи за прилагане в конкретен проект, се прави оценка на подобност на проектни ситуации. Оценката на подобност на ситуации съобразява видовете скали за описание на ситуационните фактори и наличието на липсващи стойности. Въведена е функция на подобност на ситуационен фактор (ω(x,y)), която се използва при оценка на подобност на две проектни ситуации. Подобността на две проектни ситуации ($\Omega(X,Y)$) представлява средно-аритметична сума от подобността на всеки един от ситуационните фактори.

Качественият анализ има за цел да представи в систематизиран вид информацията за приложението на гъвкавите практики в проекти с подобни проектни ситуации. Предоставената информация се анализира допълнително от инженера по конструиране на методи за да се изберат най-подходящите за прилагане в конкретния проект набори от практики. В синтезиран вид е представена информация за прилагането на всяка една гъвкава практика в подобните на целевия проект ситуации. Анализът съдържа:

- общ брой от базови проекти, в които определена практика е приложена в най-голяма степен

- общ брой от базови проекти, в които определена практика е приложена в средна степен
- общ брой от базови проекти, в които определена практика е приложена в най-ниска степен
- общ брой от базови проекти, в които определена практика не е приложена или е приложена неуспешно
- общ брой от базови проекти, в които няма информация за приложението на определена практика.

Предложеният в дисертацията метод за *количествен анализ* е клъстерния анализ. Анализът се извършва върху подобните на базовия проект проекти чрез класификация на клъстери от практики. Този анализ може да се извърши с подходящ инструмент, в който да бъдат въведени данните за подобните проекти и използваните в тях практики. Могат да бъдат извършени и други количествени анализи, които да помогнат за определяне на най-подходящите за прилагане практики - например за определяне на ниво на сходство между няколко набора от практики и определяне на този, който е най-близо до предложен от инструмента.

3.4.4. Теоретична неприложимост на гъвкавите практики

Теоретичната неприложимост на гъвкави практики при наличието на определени ситуационни фактори се използва за допълнителен анализ на наборите от практики, идентифицирани при избора на подходящи гъвкави практики за проекта. Прилага се само качествен анализ чрез визуално представяне на включените в набора практики, за които има теоретична неприложимост. Анализът помага на инженера по конструиране на методи да съобрази възможните последствия от използването на теоретично несъвместима практика.

Теоретичната неприложимост на гъвкавите практики е базирана на характеристиките на практиките, на чиято основа са направени предположения за приложимостта на дадена практика при наличието на определени ситуационни фактори. Направените заключения са обосновани чрез преглед и анализ на различни литературни източници напр. [36] [37] [38]. Пълният анализ е описан в [39]. За измерване на теоретичната неприложимост на гъвкавите практики е въведена 4-степенна скала {0,1,3,5}, където със стойност ‘5’ се измерва най-високото ниво на неприложимост на практика. Ако определен фактор не пречи или способства за прилагането на дадена практика се обозначава с нулева стойност.

3.4.5. Покритие на фазите на процеса на разработка

Вторият допълнителен анализ на избраните множества от подходящи за прилагане гъвкави практики е покритието на дейностите на процеса на разработка и съпътстващите процеса дисциплини. Този анализ дава информация за целостта на избраното множество от практики и може да идентифицира нуждата от добавяне на нови практики към него с цел по-пълно покритие на дейностите и дисциплините в софтуерния процес. Използват се идентифицираните от Съмървил [34] седем дейности, които присъстват във всеки процес за разработка.

Освен основните дейности, които присъстват във всеки процес за разработка на софтуер, дисциплините управление на проекта, управление на конфигурациите и подобрение на процеса са основна част от софтуерната разработка и съпътстват процеса във всяка една от неговите дейности.

3.4.6. Покритие на гъвкави ценности и принципи

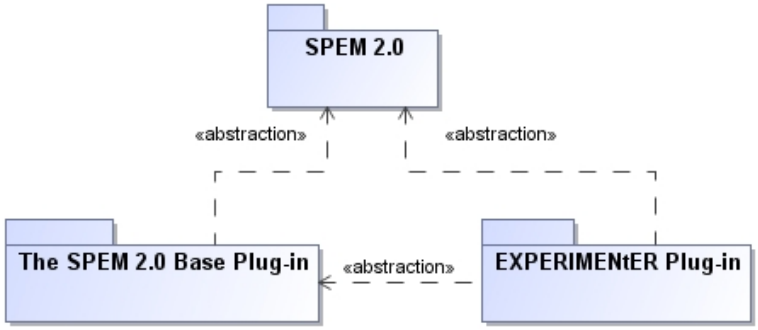
Третият допълнителен анализ върху избрано множество от подходящи гъвкави практики е покритието на гъвкавите ценности и принципи, дефинирани в Гъвкавия Манифест [9]. Този допълнителен анализ предоставя на инженера на метода бърз поглед върху степента на гъвкавост на дадено множество от практики. Покритието на всички гъвкави ценности и принципи би осигурило цялостно покритие на важните за гъвкавата разработка аспекти.

3.5. Конструиране на метод за разработка на софтуер

Втората и третата основни дейности от процеса на подхода за конструиране на методи, използван в предложената гъвкава методология са конструирането и адаптирането на метода на разработка. В целта на предложената в настоящата работа методологията е поставено изискването да бъде изградена върху отворени стандарти и метамодели за конструиране на методи с индустриално приложение. Поради тази причина предложеният в методологията метамодел за описание на конструираните методи е базиран на SPEM 2.0 метамодела [40].

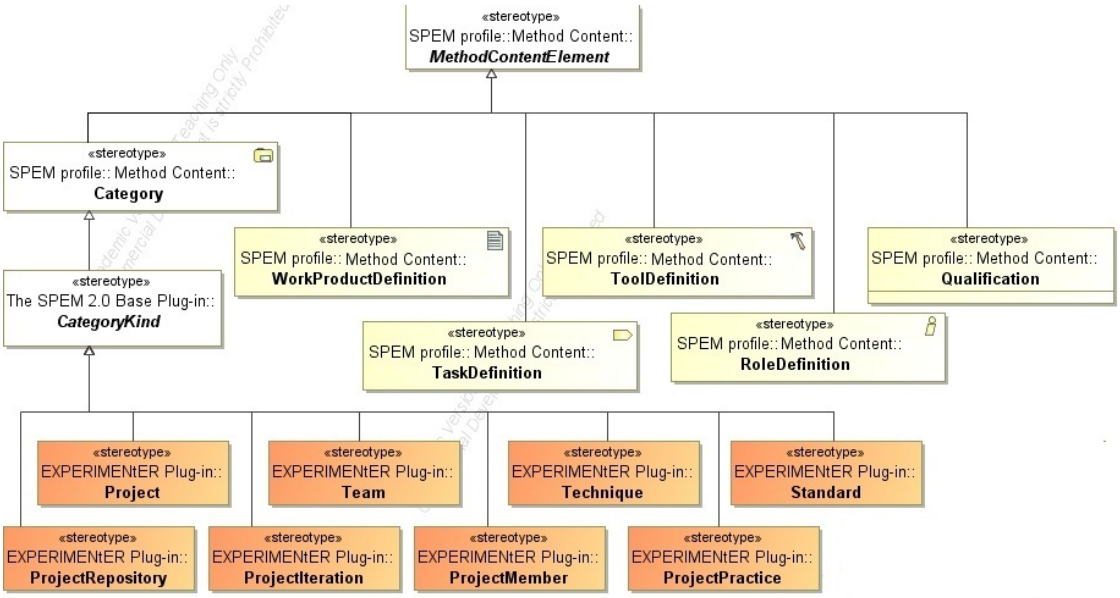
3.5.1. SPEM-базиран метамодел

Метамоделът, използван за описание на метода за разработка на софтуер, дефинира нови елементи, които са специализации на елементи от SPEM 2.0 метамодела и на стандартната инстанция за софтуерни проекти SPEM 2.0 Base Plug-in. Новите елементи са добавени в пакет наречен EXPERIMENtER Plug-in. Връзките между елементите в SPEM 2.0 метамодела, в стандартната инстанция SPEM 2.0 Base Plug-in и добавените елементи в пакета EXPERIMENtER Plug-in са показани на Фигура 1.7. Добавянето на елементи, които са специализации на елементите в от SPEM 2.0 метамодела и SPEM 2.0 Base Plug-in, позволява преносимост на моделите, между различни SPEM-базирани инструменти. Другото предимство е, че са наследени важни свойства на елементите в метамодела.



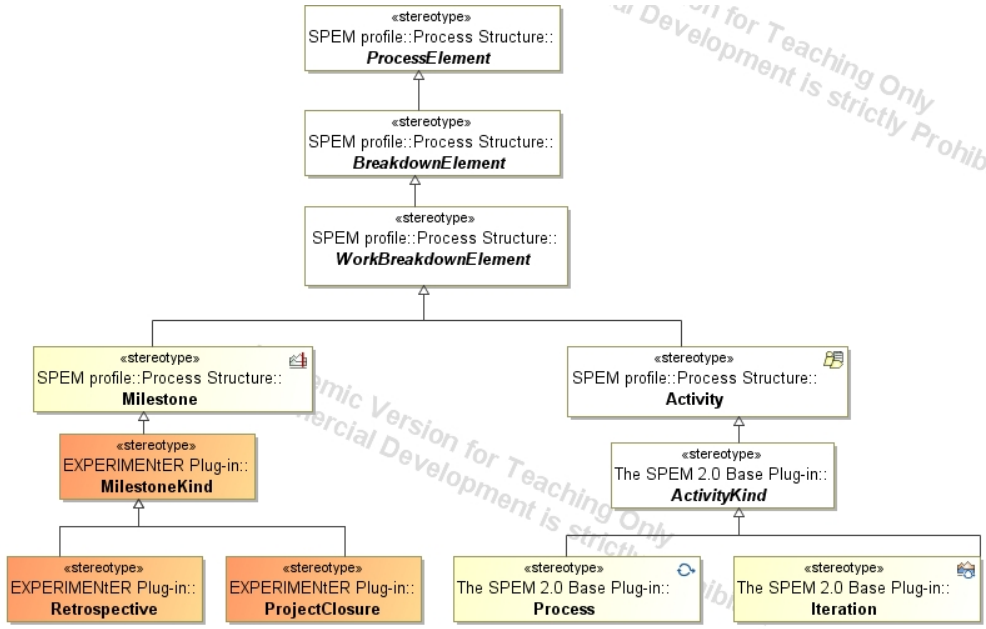
Фигура 1.7 Диаграма на връзките между елементите в SPEM 2.0, стандартната инстанция и елементите в EXPERIMENtER Plug-in пакета

На Фигура 1.8 са представени основните концепции на методологията, които са описани чрез подходящи инстанции на SPEM 2.0 елементи в пакетите MethodContent и ManagedContent. На диаграмата са представени дефинираните елементи чрез подходящи стереотипи и техните връзки. В по-светъл цвят са показани концепциите, представени чрез стандартни елементи от SPEM 2.0 профила и SPEM 2.0 Base Plug-in профила. Базовите елементи от SPEM 2.0, от които са наследени елементите в подхода, са маркирани с бяло. В по-тъмен свят са представени елементите добавени за конкретния подход. Те са дефинирани в UML профил EXPERIMENtER Plug-in.



Фигура 1.8 Основни концепции в подхода, представени чрез MethodContentElementa елемента от пакета MethodContent

Фигура 1.9 представя концепции на методологията, които се описват чрез елементи, които наследяват WorkBreakdownElement от пакета ProcessStructure от SPEM 2.0 метамодела.



Фигура 1.9 Основни концепции в подхода, представени чрез елемента ProcessElement от пакета ProcessStructure

3.5.2. Спецификация на EXPERIMENtER Plug-in

Спецификацията на добавените елементи в пакета EXPERIMENtER Plug-in е направена по следния шаблон:

- Име
- Наследен клас
- Семантика

- Описание
- Атрибути
- Нотация - описание на добавените в профила стеротипи, техните наследени и мета класове, ключови думи и икона за представяне.

В UML профила EXPERIMENtER Plug-in са включени спецификациите на следните елементи:

- Хранилище на проекти (ProjectRepository)
- Проект (Project)
- Ситуация на проект (ProjectSituation)
- Екип (Team)
- Член на проект (ProjectMember)
- Итерация на проект(ProjectIteration)
- Практика (ProjectPractice)
- Техника (Technique)
- Ретроспективна среща (Retrospective)
- Въпросник на итерация (RetrospectiveQuestionnaire)
- Стандарт (Standard)
- Въпросник на проект (ProjectQuestionnaire)
- Затваряне на проект (ProjectClosure)

3.5.3. Избор на гъвкави практики

Изборът на гъвкави практики от хранилищата за проекти на компанията за конкретна итерация се извършва по следната процедура:

1. Избират се хранилищата на проекти, в които да се търсят подходящи практики
2. Специфицира се набора от стандартни гъвкави практики, които да бъдат включени в проекта. Този набор може да е генериран в резултат на първата стъпка в подхода EXPERIMENtER (подход за създаване на метод за разработка, базиран на предходен опит).
3. Специфицират се:
   3.1. (опционално) продукти, които са идентифицирани да бъдат създадени при изпълнение на практиките на проекта
   3.2. (опционално) инструменти, които са идентифицирани да се използват в проекта
   3.3. (опционално) задачи, които са идентифицирани да се включат в проекта
   3.4. (опционално) минимално ниво на гъвкавост
4. Предлагат се най-подходящите множества от практики, използвани в предходни проекти на компанията, и които представляват реализации на стандартните практики, зададени на стъпка 2, на базата на:
   4.1. ситуацията на проекта
   4.2. данните за удовлетвореността от приложението на практиките в хранилищата на проектите
   4.3. ограниченията, зададени на стъпка 3 и
   4.4. данните от въпросници на ретроспективните срещи от предходни итерации (ако има такива)

Процедурата описана по-горе може да бъде изпълнявана при стартирането на всяка итерация, като на последната стъпка се взимат под внимание и данните от ретроспективните срещи. Описаната процедура може да бъде автоматизирана и изборът

на гъвкави практики да се извършва автоматично на базата на данните от въпросниците на проекти и итерации, съхранявани в хранилищата на проектите.

3.6. Оценяване

Предложената в настоящата работа гъвкава методология за разработване на софтуер включва оценяване на метода за разработка в две направления - оценяване на непротиворечивост и цялостност, и оценяване на степен на гъвкавост. Двете направления са разгледани в следващите секции.

3.6.1. Оценяване на непротиворечивост и цялостност

Правилата за непротиворечивост и цялостност, дефинирани на ниво метамодел, са описани чрез езика OCL (Object Constraint Language) [41]. Правилата за цялостност определят елементи, които трябва задължително да присъстват в спецификацията на елемент от съдържанието на метода. Например, една техника трябва задължително да е свързана с поне една задача, а една задача трябва задължително да определя стъпки за извършването ѝ и умения, които са необходими за изпълнението и. От друга страна, ако в дефиницията на определена техника присъства дадена роля, а квалификациите, свързани с тази задача не присъстват в ролята, се нарушава непротиворечивостта на метода. Правилата, които оценяват непротиворечивост на метода, проверяват дали свързаните с даден елемент елементи присъстват и в елементите, които използват продукта. Също така, правилата за непротиворечивост могат да се приложат и към конкретен тип елемент от съдържанието на метода. Например, продуктите описани в дефиницията на един стандарт, който се използва в даден проект, трябва да бъдат включени в дефинициите на някои от практиките, използвани в проекта.

Оценката за цялостност и непротиворечивост на метода за разработка се извършва чрез броя на правилата, които са изпълнени за конкретния метод. Петнайсет правила за цялостност и единайсет правила за непротиворечивост са дефинирани върху елементи от съдържанието на метода за разработка.

3.6.2. Степен на гъвкавост

Степента на гъвкавост се измерва на база на използваните гъвкави практики и поддръжката, която осигуряват те за 5 основни атрибута за гъвкавост. Петте атрибута за гъвкавост са:

- Гъвкавост - практиката поддържа ли адаптиране към закъснели и непредвидени промени
- Скорост – практиката поддържа ли създаването на резултати бързо
- Лекота – практиката поддържа ли разработка в кратки срокове, използване на неформални процедури и ефективни инструменти
- Адаптивност – практиката поддържа ли използването на предходно знание и опит за подобряване и развитие
- Отзивчивост – практиката поддържа ли адекватна и навременна реакция на промени

Дефинирана е формула за определяне на степента на гъвкавост на метод на база на степента на гъвкавост на всяка от гъвкавите практики, използвани в метода.Степента на гъвкавост се изменя в границите от 0 до 1, като най-голяма степен на гъвкавост имат методи на разработка, чиято стойност на гъвкавост е близка или равна на 1.

3.7. Заключение

Глава Глава 3 представя предложената в научния труд нова гъвкава методология за разработка на софтуерни приложения. Описаната методология е представена в две научни статии [42] и [43].

В глава Глава 3 е *решена втората задача* на докторантския труд:

Задача 2: Да се създаде нова гъвкава методология за разработка на софтуерни приложения, която използва предходен опит при конструирането на методи за разработка, съобразени с характеристиките на конкретния проект, в който се използват. За целите на методологията:

- да се предложи подход за управление на знанието, придобито при предходно внедряване на гъвкави методологии
- да се дефинира набор от фактори за описание на характеристиките на софтуерните проекти
- да се идентифицира подходящ метамодел за представяне на метода на разработка и да се разшири за нуждите на новата гъвкава методология

Глава 4. Реализация и валидация на методологията

Реализацията на предложената в дисертацията гъвкава методология за разработване на софтуер и нейното валидиране е обект на разглеждане в глава 4.

4.1. Реализация

За осигуряване на автоматизирана поддръжка на предложената гъвкава методология за разработване на софтуер е изготвена спецификация на подходяща CAME среда. Предложената в настоящата работа CAME среда има за цел да запази лекотата и изпълнимостта на подхода и неговата индустриална приложимост. Чрез използването на индустриални инструменти с отворен код, базирани на стандартизирани метамодели, приложимостта и разширяемостта на предложения подход се повишава в значителна степен.

Средата, осигуряваща реализация на методологията, се състои от два инструмента, които поддържат трите основни дейности в EXPERIMENtER подхода - избор на приложими гъвкави практики и създаване и адаптиране на метода на разработка. Реализацията на базата от знания за външните проекти се разработва като отделно интернет-базирано приложение, тъй като данните и анализите върху тях не са обвързани с конкретен проект. Също така свободният достъп до базата позволява тя да се обогатява с информация от различни източници. Инструментът, който предоставя реализация на базата от знания, е наречен APR Tool (Agile Projects Repository tool). Вторият инструмент, който осигурява поддръжка за конструиране и адаптиране на метода на разработка, е именуван EPFExtSME (EPF Extension for Situational Method Engineering). Инструментът е разработен като разширение на инструмента с отворен код EPF Composer [27]. EPF Composer има някои характеристики на CAME средите като поддържане на хранилище от части на методи и генерация на CASE инструменти като редактор на диаграми. Основното ограничение на инструмента е, че не предоставя възможности за създаване на ситуационни методи. Това се дължи на факта, че EPF Composer реализира SPEM 2.0 метамодела и стандартната инстанция за софтуерни проекти, в които не са включени понятия, необходими за създаването на ситуационни методи. Инструментът EPFExtSME предоставя поддръжка за конструирането и адаптирането на ситуационните методи, които са създадени при внедряването на новата гъвкава методология, чрез разширяване на EPF Composer с понятията, дефинирани в новата методология.

4.1.1. Инструмент APR Tool

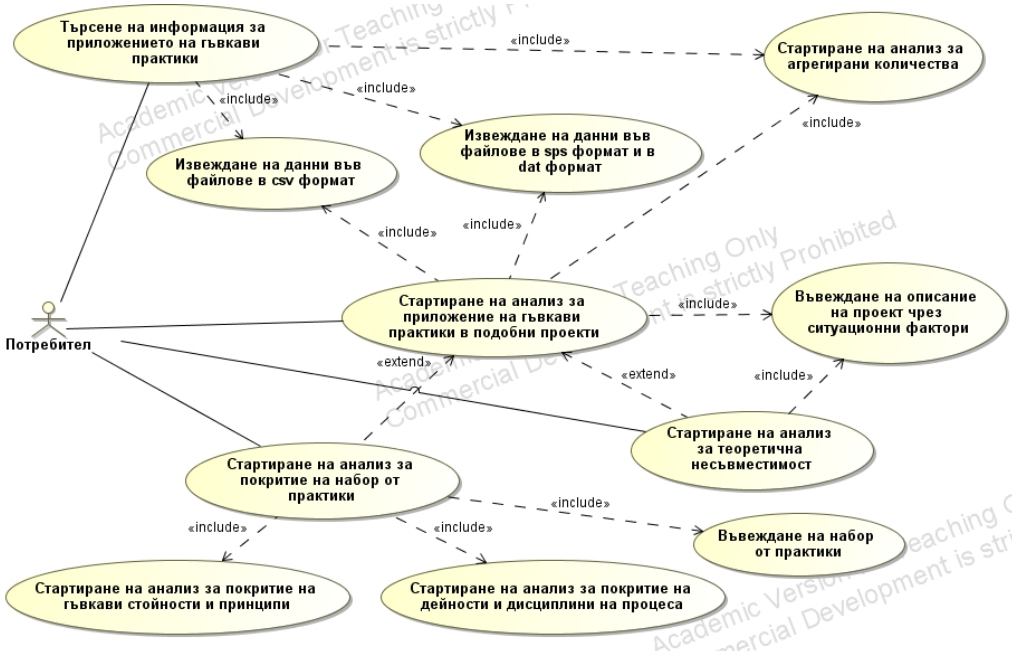
Инструментът APR Tool реализира базата от знание за приложението на гъвкавите практики. Инструментът се използва за съхраняване на информация за:

- Степента на приложение на гъвкави практики в различни проекти, описани чрез ситуационни фактори
- Степента на неприложимост на гъвкави практики при наличието на определени ситуационни фактори
- Покритието на гъвкави практики върху дейностите на процеса на разработка и поддържащите дисциплини
- Покритието на гъвкави практики върху гъвкавите ценности и принципи

В приложението са дефинирани три роли потребители: администратор на приложението, анонимен потребител и регистриран потребител.

Администраторът управлява потребителите на приложението, които получават достъп до системата чрез регистрация. Ролята дава права да се въвеждат, редактират и изтриват номенклатурите за гъвкави практики и гъвкави методи, ситуационни фактори и техните скали, скалите за приложимост, теоретична неприложимост, дейности и дисциплини на софтуерния проект и гъвкави ценности и гъвкави принципи. Администраторът въвежда и редактира данните за покритие на дейности и дисциплини, както и покритие на гъвкави ценности и принципи. Потребител с административни права може да изтрива въведената от други потребители информация за приложението на гъвкави практики в проекти.

Анонимният потребител може да достъпи системата без регистрация. Потребителите без регистрация могат да виждат информацията за всички въведени проекти и използването на гъвкави практики в тях. Те могат да стартират различни анализи за избор на набор от подходящи за прилагане практики. Потребители без регистрация могат да извеждат информацията за приложението на гъвкавите практики в проектите във файлове в csv формат и файлове, описващи структурата и данните необходими за въвеждане в инструмента за статически анализи IBM SPSS Statistics [44]. Фигура 1.10 представя случаите на употреба за приложението за потребители без регистрация.



Фигура 1.10 Диаграма на случаите на употреба на APR Tool за потребителска роля

Регистрираният потребител може да добавя информация за използването на гъвкави практики в проекти. Всеки регистриран потребител може да редактира и изтрива въведената от него информация. Регистрираният потребител може да извършва всички операции, които може и анонимния потребител.

4.1.2. Инструмент EPFExtSME

Инструментът EPFExtSME предоставя функционалност за конструиране и адаптиране на метода за разработка, използван в проект на организация. Изискване към реализацията на инструмента EPFExtSME е той да е разширение на EPF Composer инструмента. EPF Composer осигурява поддръжка за по-голяма част от SPEM 2.0 метамодела [45] и инстанцията за софтуерни проекти SPEM 2.0 Base Plug-in. В EPFExtSME са добавени елементите от спецификацията на EXPERIMENtER Plug-in и допълнителна функционалност, която подпомага инженера по конструиране на методи при избора на подходящи гъвкави практики за определен проект. Дефинирането на елементите от EXPERIMENtER Plug-in като специализации на елементите в SPEM 2.0 метамодела и инстанцията SPEM 2.0 Base Plug-in осигурява преносимост на моделите за разработка, създадени в EPFExtSME, както към EPF Composer, така и към други продукти, базирани на SPEM 2.0 метамодела. Преносът се осъществява чрез сериализация на моделите в стандартизирания от OMG XMI (XML Metadata Interchange) формат [46].

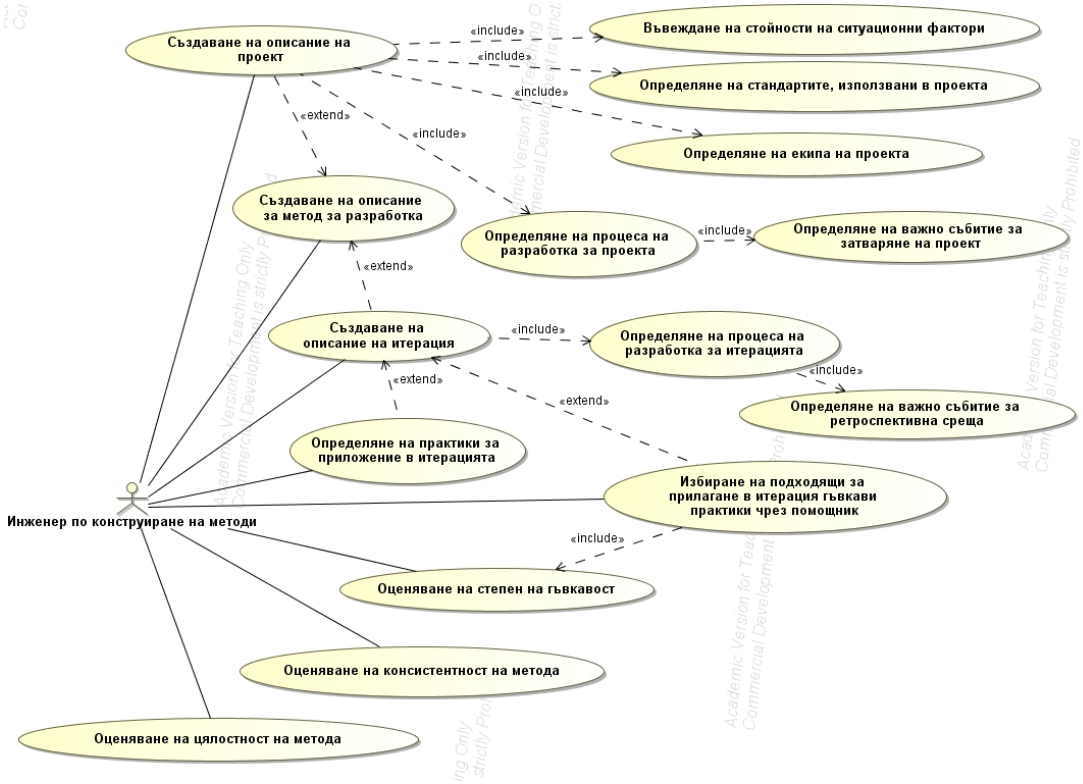
Функционалната спецификация на инструмента EPFExtSME може да бъде разделена в следните три функционални модула:

- Управление на елементите на метода за разработка
- Създаване на описание на метода за разработка
- Адаптиране на метода за разработка и затваряне на проекта

Модулът *Управление на елементите на метода на разработка* включва операциите по създаване и редактиране на елементите, които ще бъдат включени в описанието на метода за разработка. Инженерът по конструиране на методи създава и редактира описания на членове на проектите, стандарти, техники, практики и гъвкави практики. Създаването и редактирането на практики включва определянето на инструментите, продуктите, задачите, техниките и ролите, които участват в описанието на практиката. Дефинирането на гъвкави практиките включва определянето на стандартната гъвкава практика, която е реализирана от практиката, използвана в проекта. За всяка гъвкава практика е предварително дефинирана степента на гъвкавост на практиката, която може да бъде променена от инженера по конструиране на методи. Инженерът по конструиране на методи дефинира и екипите, които участват в проектите, като определя членовете на проектите, които участват в екипите и техните роли в конкретния екип.

Модулът *Създаване на описание на метода за разработка* включва създаване на описание за метода на разработка както и неговото оценяване. Първа стъпка при създаването на метода за разработка е описването на проекта, в който се прилага метода. Ситуацията на проекта се задава чрез стойностите на множество от ситуационни фактори. За всеки проект се определят стандартите, екипа на проекта и процесът на разработка на проекта. В описанието на процеса на разработка се включва и задаването на важното събитие за край на проект. Втората стъпка от създаването на метода е създаването на описанието на итерациите участващи в процеса на разработка. Определянето на практиките, които ще бъдат включени във всяка от итерациите на проекта, може да се извърши както ръчно така и чрез използването на помощник за избор на подходящи за прилагане в итерацията практики. Помощникът реализира процедурата за избор на практики, описана в секция 3.5.3. Описанието на итерацията

включва и определяне на процеса на разработка за конкретната итерация, в който е зададена и ретроспективната среща за итерацията. Инженерът по конструиране на методи за разработка може да оцени създадения метод както чрез подходящи правила за непротиворечивост и цялостност на метода, така и чрез степента му на гъвкавост. Случаите на употреба за модула Създаване на описание на метода за разработка са представени на Фигура 1.11.



Фигура 1.11 Диаграма на случаите на употреба на инструмента EPFExtSME за модул Създаване на описание на метода за разработка

Третият функционален модул на инструмента EPFExtSME е *Адаптиране на метода за разработка и затваряне на проекта*. Той включва функционалностите по промяна на практиките, които се използват в новите итерации на проекта, и създаване на въпросници от ретроспективните срещи и затварянето на проекта. Изборът на практики за приложение в проекта може да бъде извършено ръчно или чрез използване на помощник. Разликата с избора на практики с използването на помощник при създаването на описанието на проекта, е че при адаптирането на метода участва информацията за приложението на практиките в предходните итерации на проекта. Информацията се анализира на база на данните от въпросниците от ретроспективните срещи. Ако дадена практика е прилагана в итерация с ниска степен на удовлетворение, то тя може да бъде сменена с друга в следващите итерации. При направени промени в метода за разработка могат да бъдат изпълнени отново операциите за оценяване на метода. При затварянето на проекта се извършва оценка за цялостност на ниво проект.

4.2. Валидация чрез съпоставяне с описателен модел

За валидация на резултатите в настоящата работа са използвани два подхода, които се използват в областта на софтуерното инженерство [47] - оценка чрез съпоставяне с описателен модел и пример за използване.

За да се оцени, че подходът за конструиране на методи, предложен в настоящата работа, е валиден ситуационен подход се използва обобщения процес за ситуационно разработване на методи, представен в [48]. Чрез дефинирането на подходящи стратегии, използвани в предложения подход, базиран на предходен опит, са достигнати всички намерения в модела на обобщения процес. Представянето на подхода, базиран на предходен опит, чрез обобщения процес за конструиране на методи доказва, че създаденият за целите на методологията подход е валиден ситуационен подход.

4.3. Валидация чрез пример за използване

Примерът за използване проследява изпълнението на трите основни стъпки в процеса за конструиране на методи за разработка, описан в секция 3.2. Примерът за използване е върху проект в областта на компонентно-базираното инженерство. Проектите в областта на компонентно-базираното инженерство крият определени предизвикателства за прилагането на гъвкави методологии поради своите характерни особености [49] [50]. Поради тази причина използването на предходен опит при внедряването на гъвкава методология има множество предимства. Примерът за използване има за цел да потвърди изпълнимостта на предложената методология. Също така той илюстрира използването на знание при внедряването на гъвкави методологии в проекти с идентифицирани ограничения пред прилагането на гъвкави методи в тях.

Прототипите на инструментите, предложени в глава Глава 4, за автоматизирано изпълнение на методологията - APR Tool и EPFExtSME, са разработени като студентски проекти в курс към магистърската програма „Софтуерни технологии” на Факултета по Математика и Информатика към Софийския Университет „Св. Климент Охридски”. Описание на проекта

Примерният проект, в който е внедрена новата методология е среден проект с продължителност две години с големината на екипа 12 човека. Продуктът, който се разработва, е компонент за вградена система с многократно използване в няколко хиляди системи. Проектът е продължение на предходен проект и има за цел да подобри съществуващ компонент и да го развие. Критичността на системата е средно висока – отказът на компонента би довел до загубата на значително количество пари. Очакваните ограничения пред прилагането на гъвкави методологии в проекта произхождат както от типа на продукта – вградена система [51], така и от типа на проекта - разработка на компонент [49] [50]. Също така в проекта се наблюдават и ограничения за прилагането на стандартните гъвкави методи поради големината на екипа и критичността на системата [38] [33]. Поради тези ограничения използването на подход, който не отчита характерните особености на проекта, не би бил приложим.

4.3.1. Избор на стандартни гъвкави практики

Изборът на стандартни гъвкави практики е първата дейност в процеса на ситуационния подход, използван в предложената гъвкава методология. Основните стъпки, които са илюстрирани в настоящия пример са:

1. Събиране на информация
2. Попълване на данните в базата от знание
3. Търсене на подобни проекти
4. Избор на подходящи за прилагане практики

При *събирането на информация* за попълване на базата от знания с данни за приложимостта на гъвкавите практики при разработката на компоненти са използвани два подхода – *изготвяне на въпросник и систематичен анализ на приложни изследвания*. Подробна информация за съдържанието, структурата и резултатите от

изготвения въпросник за приложението на гъвкавите методи и практики в индустрията може да бъде намерена в [52], [53] и [54]. Въпросникът е попълнен от 33 професионалисти в областта на компонентно-базираното инженерство предимно от скандинавските страни. От тях 13 отговора засягат разработването на компоненти, а останалите – разработването на системи чрез асемблирането на компоненти. *Систематичният анализ на приложни изследвания* е извършен чрез филтрирането на подходящи научни изследвания от електронните библиотеки Compendex, Inspec и GEOBASE. Следните критерии са използвани при търсенето на приложни изследвания за прилагането на гъвкави методологии при разработката на компоненти: ((component OR cots) AND agile AND (experience OR case study)). От върнатите 324 резултата на база на резюметата и заглавията на първите 25 статии, бяха избрани 8 статии за допълнителен анализ. От тях 2 са избрани за включване в базата от знания поради наличието на достатъчно данни в тях.

Прототипът на инструмента APR Tool е използван при *попълването на базата от знания* с данните събрани на предходната стъпка.

Търсене на подобни проекти се извърши няколко пъти с различни критерии. Първото търсене с всички ситуационни фактори и праг на подобие 1 не върна резултат. При модифициране на прага на подобие на 0.5 също не бяха избрани проекти. При праг на подобие 0.3 бяха върнати 8 проекта – всички за разработване на компоненти и използващи гъвкави методи. Следващото търсене беше извършено с задължително съвпадащи стойности за Тип на проекта, Метод на разработка, Тип на продукта и Критичност и резултата съдържаше 3 проекта. При последното извършено търсене на задължително съвпадащи стойности за факторите Тип на проекта и Метод на разработка, и праг на подобност 0.3 бяха върнати 8 записа. Осемте проекта от последното търсене са избрани за използвани при анализите на приложението на практиките, поради съвпадението му като резултат от две селекции и по-големия брой проекти в резултата.

Изборът на подходящи за прилагане практики включва извършването на качествените и количествени анализи, описани в секции 3.4.3, 3.4.4, 3.4.5 и 3.4.6. Анализът *чрез сумарни количества* показа, че най-използваните практики са подобряване на качеството на кода и програмиране по двойки. Беше установено също така, че се използва клиент на място със средна степен на прилагане, което отчита модифициране на практиката чрез използването на бизнес експерти вместо клиент. Практиката предварително писане на тестове е прилагана в един проект в най-висока степен, в два проекта в средна степен и в един проект в ниска степен. Повечето проекти използват итеративна разработка, но чести доставки на версии на продукта не се практикуват в половината от проектите. Данните за използване на практики от Скрум са само от един проект, в който практиките са прилагани в средна и висока степен. Извършеният анализ на сумарните количества показа, че част от ограниченията за използване на гъвкави методи в проекта са разрешени в други подобни проекти. Практиката за предварително писане на тестове е прилагана при проекти за вградени системи, което е едно от предизвикателствата за внедряване на гъвкави методологии в такива проекти [51]. Използването на практиката подобряване на качеството на кода в голяма степен е идентифицирано като предизвикателство пред гъвкавата разработка на компоненти поради ограничените възможности за смяна на проекта на системата [50]. Направеният анализ показа използването на практиката предимно в най-високата степен.

Количественият анализ е извършен с помощта на инструмента за статистически анализ IBM SPSS Statistics [44]. Данните за осемте проекта са изведени във файлове, описващи структурата и данните за въвеждане в инструмента IBM SPSS Statistics. В

резултат на клъстерния анализ са генерирани два клъстера от статистически значими данни. Единият клъстер обединява 7 от проектите, а вторият клъстер покрива само един проект. В първия по-значим за изследванията клъстер попадат предимно практики с липсващи стойности, поради високата степен на липсващи данни. Резултатите от количествения анализ показват, че е необходим достатъчно голям набор от данни за проекти за да бъдат намерени клъстери с налични данни.

В резултат на двата основни анализа бяха избрани два набора от практики. Вторият набор има по-добро покритие върху гъвкавите принципи и затова е избран за използване в проекта.

4.3.2. Създаване на метод за разработка

В настоящата секция е описано изпълнението на тези две дейности чрез няколко основни стъпки:

- Създаване на описание на проекта в хранилището на проекти
- Създаване на съдържание на метода на разработка чрез използване на практики от предходни проекти на компанията
- Създаване на въпросник на итерация
- Адаптиране на метода на разработка на база на информацията от въпросниците на итерации
- Оценяване на метода за разработка

Методът за разработка на примерния проекта за разработване на вградено приложение е създаден чрез прототипа на инструмента EPFExtSME. Елементите от EXPERIMENtER Plug-in, които са добавени в стандартните категории на EPF Composer.

Първа стъпка при създаване на метода за разработка е *добавянето на проекта и неговото описание в хранилището с проекти*. Създаден е проект exampleProject, в който е добавен един екип – exampleProjectTeam. Не са задавани стандарти за използване в проекта. В секцията Detail Information са описани ситуационните фактори за проекта. За целите на примера, представен в настоящата глава, в хранилището от проекти е създаден проект embeddedCProject. Проектът embeddedCProject описва проект за разработване на вградена система, в който са използвани практики на Скрум. Въпросникът на проекта е попълнен като използването на практиките е било високо, както и удовлетворението от прилагането на практиките.

При добавянето на итерация към съдържанието на примерния проект exampleProject се стартира помощник, който се използва за *създаване на съдържание на метода на разработка чрез използване на практики от предходни проекти на компанията*. Помощникът се използва за избор на практики по описаната в секция 3.5.3. На първата стъпка на помощника се избират 11-те стандартните гъвкави практики, които бяха идентифицирани на предходната стъпка. На екрана на помощника за избор на практики от предходни проекти се показват практиките от проекта embeddedCProject, тъй като те реализират част от стандартните гъвкави практики, избрани на предходната стъпка и са прилагани с висока степен на удовлетвореност. След приключване на помощника петте практики на Скрум са добавени автоматично в съдържанието на първата итерация на проекта.

Тъй като няма проекти, които да са използвали практики на ЕП в компанията, практиките на ЕП, необходими за примерния проект, са създадени на ръка и са добавени към практиките на първата итерация.

След първата итерация е *създаден въпросник на итерацията*, на който са отразени приложението и удовлетворението от прилагането на практиките. Зададена е ниската удовлетвореност от прилагането на три от практиките. При създаването на втората итерация в проекта се използва помощник за избор на практики. Задава се

набора от стандартни практики, идентифицирани на първата стъпка, и за този набор се показват реализациите на практики, които са използвани в предходни итерации със средно или високо ниво на удовлетвореност. От дванайсетте практики в първата итерация се показват само девет, тъй като останалите три са с ниско ниво на прилагане или удовлетвореност. Деветте практики са избрани за използване във втората итерация на проекта. На основа на информацията за приложението и удовлетвореността от прилагането на практиките в първата итерация, във втората итерация е добавена нова практика, която е подобрение на практика, използвана в първата итерация.

Оценяването на метода за разработка е илюстрирано чрез правила за цялостност на ниво итерация. Чрез добавен в настройките на средата редактор се въвеждат правила за цялостност на ниво итерация, описани в секция 3.6.1. Въведените две правила се активират за да се включат при валидацията на метода. При изпълнението на валидацията се получава съобщение за грешка поради неизпълнение на правилото за добавените към итерацията работни последователности (workflows). Работните последователности описват частта от процеса за разработка, който е свързан с втората итерация. След добавяне на работна последователност към съдържанието на итерацията валидацията за цялостност преминава успешно.

4.4. Анализ на получените резултати

Получените резултати набелязаха някои необходими предпоставки за внедряването, както и предложения за подобрение. Изводите от направените анализи на резултатите са следните:

- Необходим е достатъчно голям брой записи в базата от знания, за да може да се извършат количествените анализи на данните
- Изборът на подобни проекти да позволява многократно търсене с различни критерии и промяна на прага на подобие
- Покритието на принципи, ценности, дейности и дисциплини да указва броя на практиките от избрания набор практики, които осигуряват това покритие
- Създаването на практики да бъде включено като част от помощника за избор на практики от вътрешните проекти. Това ще даде възможност да се проследи покритието на избраните и добавените реализации спрямо началния набор от стандартни гъвкави практики
- Практиките в хранилището да бъдат групирани по стандартните гъвкави практики, които реализират, за да се улесни търсенето им.

4.5. Заключение

В глава Глава 4 е представена реализацията и валидацията на предложената в работата гъвкава методология. Примерът за използване потвърди изпълнимостта на предложената методология и показа ползите от прилагането на предходен опит при внедряването на гъвкави методологии в проекти с идентифицирани ограничения за прилагането на гъвкави методи. На база на опита от прилагането на методологията бяха предложени насоки за подобряване и развитие на методологията.

В глава Глава 4 *са представени решенията на последните две задачи* на дисертацията:

Задача 3: Да се специфицира автоматизирана среда за реализиране на методологията, базирана на отворени стандарти и метамодели с индустриално приложение.

Задача 4: Да се валидира предложената гъвкава методология чрез пример за използването й.

3. Заключение

В настоящия дисертационен труд беше представена нова гъвкава методология за разработване на софтуерни приложения. Предложената методология е базирана на предходен опит за приложението на гъвкавите практики и използва ситуационен подход за създаване на метод за разработка на софтуер. Подходът, използван в методологията, е наречен подход, базиран на предходен опит. Подходът, базиран на предходен опит, съдържа три основните дейности. Първата основна дейности е избора на подходящ набор от гъвкави практики от база от знания с информация за приложимостта на практиките в подобни проекти. Следвайки процедура за систематизиран анализ на представени в научни разработки фактори, влияещи на внедряването на гъвкави методологии, са идентифицирани 47 ситуационни фактора за описание на гъвкави софтуерни проекти. Информацията за проектите, представени чрез ситуационните фактори и прилаганите в тях гъвкави практики се съхранява в изградената база от знания. Тази информация се използва за извършване на качествени и количествени анализи за приложимостта на практиките при наличието на определени ситуационни фактори. В резултат на анализите се идентифицират множества от практики, подходящи за приложение в конкретен проект. За реализиране на другите две основни дейности в методологията - конструирането и адаптирането на метода на разработка, е предложен метамодел за описание на конструираните методи, базиран на стандартизирания от OMG SPEM 2.0 метамодел. Моделът е избран поради неговата поддръжка от индустриални CAME среди. Основните концепции на методологията са моделирани както чрез елементи на SPEM 2.0 метамодела и неговата инстанция за софтуерни проекти SPEM 2.0 Base Plug-in, така и чрез специално дефинирани за целите на настоящата работа UML профил, наречен EXPERIMENtER Plug-in. Профилът дефинира допълнителните елементи, които се използват за описване на проектни ситуации и приложението на гъвкавите практики в предходни проекти на компанията. На тяхна основа се прави избор на подходящи за прилагане практики в конкретен проект. Подобряването на метода за разработка се извършва и на база на оценки за непротиворечивост, цялостност и степен на гъвкавост.

В дисертационния труд е включена спецификация на CAME среда, която осигурява автоматизирана поддръжка на предложената гъвкава методология за разработване на софтуер. При избора и спецификацията на инструментите в средата е спазен принципа за разширяемост и индустриална приложимост на методологията. Валидацията на предложената методологията е извършена чрез два подхода - оценка чрез съпоставяне с описателен модел и пример за използване. Представянето на подхода, базиран на предходен опит, чрез обобщения процес за конструиране на методи доказва, че създаденият за целите на методологията подход е валиден ситуационен подход. Примерът за използване потвърди изпълнимостта на предложената методология и илюстрира ползата от използването на предходен опит при внедряването на гъвкави методологии в проекти с идентифицирани ограничения за прилагането на ГСР.

Няколко основни посоки за бъдещо развитие на предложената в настоящия дисертационен труд гъвкава методология могат да бъдат набелязани. Първото направление за развитие е внедряване на методологията в повече експериментални и най-вече реални проекти. По този начин ще бъде оценена нейната ефективност и ефикасност. Резултатите от реалната експлоатация ще бъдат използвани за повишаване

на използваемостта на методологията и прецизиране на предложените в нея анализи. Като част от бъдещата работа е усъвършенстването на инструментите в CAME средата, осигуряваща автоматизирана поддръжка на методологията. Основното разширение е при конструирането на метода на разработка и неговото адаптиране в посока на процеса на разработка. Асемблирането на процес за разработка от съществуващи в хранилището елементи на метода би улеснило конструкторите на методи в значителна степен. Това асемблиране би могло да се извърши по автоматизирана процедура на база на входните и изходните продукти на дейностите в процеса. Предложеното в работата разширение на стандартния метамодел SPEM 2.0 и реализацията на прототипа на инструмента EPFExtSME са идентифицирани като възможен принос към международната инициатива SEMAT (Software Engineering Method and Theory) [55]. SEMAT инициативата е създадена през 2010 година с участието на множество софтуерни специалисти и индустриални партньори с цел да бъде предложена теоретично обоснована основа за софтуерната разработка, която да може да се използва широко в индустрията. Към настоящия момент решенията, които се развиват, са именно в създаването на стандартизирано разширение на SPEM 2.0 метамодела, което да позволява конструирането на методи за разработка за различни проектни ситуации.

Приноси на дисертационния труд

Научни приноси:

Принос 1: Дефиниране на набор от фактори за описание и характеризиране на софтуерни проекти, които използват гъвкави методологии за разработка на софтуер

Принос 2: Дефиниране на метамодел за описание на гъвкави методи за разработка на софтуер, който осигурява поддръжка за конструиране на методи в контекст

Научно-приложни приноси:

Принос 3: Създаване на нова гъвкава методология, използваща систематизиран подход за конструиране на методи за разработване на софтуер в областта на гъвкавата софтуерна разработка, област в която изследванията са базирани предимно на приложни подходи

Принос 4: Изграждане на база от знания за прилагането на гъвкавите практики с цел структуриране, документиране и разпространяване на опит от използване на гъвкави методи. Използването на опита е ограничено поради факта, че е базирано предимно на знание в неявен вид

Приложни приноси:

Принос 5: Създаване на спецификация на метамодел за описание на методи за разработка на софтуер, който осигурява поддръжка на предложената методология

Принос 6: Специфициране на автоматизирана среда за реализиране на методологията, базирана върху отворени стандарти и метамодели с индустриално приложение

Публикации във връзка с дисертацията

Резултатите от дисертацията са публикувани в 1 международно списание, 8 международни конференции, на една от които с две статии в две поредни години, и една университетска библиотека с две публикации.

Публикации в международни списания:

1. Sajeev, A.S.M., Land, R., **Krasteva, I.**, Punnekkat, S., Larsson, S.: Software Process Practices and Preferences: A Pilot Study. *Journal of Software Maintenance and Evolution: Research and Practice* (accepted for publication 2010)

Публикации на международни конференции:

2. **Krasteva, I**., Ilieva, S.: Rush into Agile - Analytical Framework for Agile Practices Applicability. In : IET International Conference on Agile Manufacturing (ICAM 2007), Durham, UK, pp.229-237 (2007)

3. **Krasteva, I.**, Branger, P., Land, R.: Challenges for agile development of COTS components and COTS-based systems A theoretical examination. In : International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE'08), Funchal, Madeira, pp.99-106 (2008)

4. **Krasteva, I.**, Ilieva, S.: Adopting an Agile Methodology - Why It Didn't Work. In : The 30th International Conference on Softawre Engineering (ICSE'08) workshop on Scrutinizing agile practices or shoot-out at the agile corral, Leipzig, Germany, pp.33-36 (2008)

5. **Krasteva, I.**, Land, R., Sajeev, A.S.M.: Being Agile when Developing Software Components and Component-Based Systems - Experience from Industry. In : The 16th EuroSPI Conference, Madrid, Spain, vol. Industrial proceedings, pp.8.7-8.17 (2009)

6. Causevic A., **Krasteva, I**.: A Survey on Industrial Software Engineering. In Pekka Abrahamsson, Michele, ed.: The 10th Agile Processes in Software Engineering and Extreme Programming (XP 2009), Sardinia, Italy, pp.240-241 (2009)

7. Land, R., Sundmark, D., Lüders, F., Causevic, A., **Krasteva, I**.: Reuse with Software Components – A Survey of Industrial State of Practice. In : 11th International Conference on Software Reuse (ICSR'2009), Falls Church, VA, pp.150-159 (2009)

8. **Krasteva, I.,** Ilieva, S., Dimov, A.: Experience-Based Approach for Adoption of Agile Practices in Software Development Projects. In : The 22nd International Conference on Advanced Information Systems Engineering (CAiSE'10), Hammamet, Tunisia, pp.266-280 (2010)

9. **Krasteva, I.**, Ilieva, S.: A Systematic Approach for Selection and Adoption of Agile Practices in Component-Based Projects. In Springer-Verlag, ed. : The 11th

International Conference on Agile Software Development, XP2010, Trondheim, Norway, pp.403–404 (2010)

Публикации в университетски библиотеки:

10. **Krasteva, I.**, Branger, P., Land, R.: A Systematic Comparison of Agile Principles and the Fundaments of Component-Based Software Development. MRTC report ISSN 1404-3041 ISRN MDH-MRTC-220/2007-1-SE, Mälardalen Real-Time Research Centre, Mälardalen University, Vasteras (2007)

11. Causevic, A., **Krasteva, I.,** Land, R., Sajeev, A.S.M., Sundmark, D.: An Industrial Survey on Software Process Practices, Preferences and Methods., MRTC report ISSN 1404-3041 ISRN MDH-MRTC-233/2009-1-SE, Mälardalen Real-Time Research Centre, Mälardalen University, Sweden (2009)

Апробации

Публикациите са отчетени като приноси в няколко проекта – както международни, така и български:

- PROGRESS (http://www.mrtc.mdh.se/progress/) е проект на Шведския Университет Malardalen University за намиране на ефективни методи за справяне със сложността на софтуерните продукти. Проектът е финансиран от Шведската фондация за стратегически изследвания (SSF). Публикациите, включени в проекта са: 1, 3, 5, 6, 7, 10 и 11.
- FLEXI (Flexible Integration in Global Product Development) е международен европейски проект, включен в програмата ITEA2 (http://www.itea2.org/about_itea2), номер 06022. Целта на проекта е да подобри стратегическото предимство на Европейската софтуерна индустрия чрез предоставянето на гъвкави и бързи методи за разработване на софтуер. В проекта участват 40 организации от 8 страни. Публикациите включени в проекта са: 2, 5, 6 и 7.
- SISTER (Strengthening the IST Research Capacity of Sofia University) е европейски проект от 7 рамкова програма - FP7-REGPOT-2007-1, номер 205030. Целта на проекта е да повиши капацитета на СУ ”Св. Кл. Охридски” в три направления, едно от които е Софтуер и услуги. Публикациите включени в проекта са 8 и 9
- Проект Гъвкава Методология за Разработване на Софтуер (МУ-ФС-08/2007). Проектът е финансиран от Министерството на Образованието и Науката на РБ по конкурс „Стипендии за млади учени, които подготвят докторантски труд в национална фирмена структура”. В проекта са включени публикациите от 3 до 11.

Резултатите от работата са докладвани на множество семинари:

- докторантски семинари през 2007, 2008, 2009 и 2010 година
- семинари на катедра Софтуерни Технологии през 2008 и 2009 година
- екипен семинар във фирма Рила Солюшънс през 2008 година
- Международен симпозиум „Софтуер и услуги” в рамките на международната конференция ”Автоматика и Информатика”, проведена на през 2008 година в гр.София

- SISTER Симпозиум през 2010 година

Изследванията, направени във връзка с дисертацията, бяха включени в лекционния материал на курсовете по Управление на качеството, през 2008 и 2009 година в бакалавърската програма, специалност Софтуерно инженерство на ФМИ, както и за магистърската програма по Софтуерни Технологии в курса Съвременни софтуерни технологии през 2007, 2008, 2009, 2010 и 2011 година.

Авторска справка

Публикациите, включени в дисертацията по глави, са както следва:

Втора глава:

Публикация номер 1 е приета за публикувана в международно Списание за подобряване и еволюция на софтуер (Journal of Software Maintenance& Evolution: Research& Practice*)*. Публикацията представя резултатите от проучване за методите и практиките на разработка, използвани в индустрията, както и предпочитанията на софтуерните специалисти.

Публикация номер 4 е представена на семинар на 30-тата Международна Конференция по Софтуерно Инженерство (ICSE) в Лайпциг, Германия. Конференцията е най-известния форум в областта на софтуерното инженерство. Публикацията представя предизвикателствата пред прилагането на гъвкави методологии в няколко студентски проекта.

Трета глава:

Публикация номер 2 е представена в основната секция на Международна Конференция за Гъвкаво Производство (ICAM) в Дърам, Англия. Публикацията описва теоретичните предизвикателства пред прилагането на гъвкави практики при наличието на определени проектни фактори.

Публикация номер 8 е изнесена на докторантски семинар на 11-тата Международна Конференция за Гъвкава Софтуерна Разработка (XP) в Трондхейм, Норвегия. Конференцията е най-престижния форум в областта на ГСР в Европа. Публикацията представя описаната гъвкава методология и използваните в докторантския труд научни подходи.

Публикация номер 9 е включена в основната секция на 22-рата Международна Конференция за Съвременно Разработване на Системи(CAiSE) в Хамет, Тунис. CAISE е единствената конференция в Европа, която ежегодно има специализирана част за конструиране на методи. Публикацията представя подхода за ситуационно конструиране на методи, използван в предложената в работата гъвкава методология.

Четвърта глава:

Публикация номер 3 е представена в основната секция на Международната Конференция за Изследване на Нови Подходи в Софтуерното Инженерство (ENASE) в Мадейра, Португалия. Публикацията изследва предизвикателствата пред прилагане на гъвкави методи в проекти в областта на компонентно-базираното софтуерно инженерство.

Публикация номер 5 е изнесена в секцията за индустриални изследвания на Международната Конференция за Подобрение на Софтуерните Процеси (EuroSPI). Публикацията представя резултати от анализ на приложението на гъвкави методологии при разработването на компоненти и компонентно-базирани системи в проекти на софтуерни компании.

Публикация номер 6 е представена в секцията за постери (posters) на 10-тата Международна Конференция за Гъвкава Софтуерна Разработка (XP) в Сардиния, Италия. Публикацията представя структурата и резултатите от въпросник за приложението на гъвкавите методи и практики в проектите на софтуерни компании.

Публикация номер 7 е представена на Международната Конференция за Повторно Използване (ICSR) във Falls Church, САЩ. Публикацията представя данни за използването на методи и практики в проектите на софтуерните компании при разработката на компоненти с висока степен на повторно използване.

Публикация номер 10 е вътрешен доклад публикуван в библиотеката на шведския университет Malardalen University. Докладът представя систематизиран анализ на предизвикателствата пред прилагането на гъвкавите методи в областта на компонентно-базираното софтуерно инженерство.

Публикация номер 11 е вътрешен доклад публикуван в библиотеката на шведския университет Malardalen University. Докладът представя подробно описание на структурата и резултатите от въпросник за приложението на гъвкавите методи и практики в проекти на софтуерни компании.

Други публикации

1. **Krasteva**, I. Manova, S. Ilieva, "Methodology for Automated Functional Testing of Internet Applications", Proceedings of 20th International Conference on Systems for Automation of Engineering and Research (SAER'2006), pp. 89-97, Varna – St. Konstantin resort, 24-26 September, 2006

2. **Krasteva,** I., I. Manova, S. Ilieva, 'Testing Approach for SOA E-Government Application, Proceedings of International Conference on Software & Systems Engineering and their Applications (ICSSEA 2008), Paris, France, 9-11 December 2008

3. Dzhurov Y., **Krasteva I.**, Ilieva S., Personal Extreme Programming – An Agile Process for Autonomous Developers, International Conference on Software, Services & Semantic Technologies (S3T'09), Sofia, Bulgaria, October 28-29, 2009

4. Iliev M., **Krasteva I.**, Ilieva S., A Case Study on the Adoption of Measurable Agile Software Development Process, International Conference on Software, Services & Semantic Technologies (S3T'09), Sofia, Bulgaria, October 28-29, 2009

5. **Krasteva, I.**, Wahid A., Ilieva, S.: Characterizing Agile Projects, Internal Report, Sofia Uniresity 'St. Kliment Ohridski', Sofia (2009)

Библиография

1. Fitzgerald, B.: Formalized Software Development Methods: a critical perspective. Information Systems Journal 6, 3-23 (1996)
2. Hardy, C., Thompson, J., Edwards, H.: The Use, Limitations and Customization of Structured Systems Development Methods in the United Kingdom. Information and Software Technology 37(9), 467-477 (1995)
3. Dingsoyr, T., Dyba, T., Moe, N.B.: Agile Software Development: An Introduction and Overview. In : Agile Softawre Development: Current Research and Future Directions.

Springer-Verlag Berlin Heidelberg (2010)

4. Softwate, Rally: Agile Adoption Trends. In: Rally Softwate. Available at: http://www.rallydev.com/learn_agile/agile_for_executives/
5. Abrahamsson, P., Oza, N., Siponen, M.: Agile Software Development Methods: A Comparative Review. In : Agile Softawre Development: Current Research and Future Directions. Springer-Verlag Berlin Heidelberg (2010)
6. Abrahamsson, P., Conboy, K., Wang, X.: 'Lots Done, More to Do': the Current State of Agile Systems Development Research. European Journal of Information Systems 18, 281–284 (2009)
7. Conboy, K., Morgan, L.: Future Research in Agile Systems Development: Applying Open Innovation Principles Within Agile Organisation. In : Agile Softawre Development: Current Research and Future Directions. Springer-Verlag Berlin Heidelberg (2010)
8. Highsmith, J.: Agile Software Development Ecosystems. Addison-Wesley Professional (2002)
9. Manifesto for Agile Software Development. Available at: http://agilemanifesto.org/
10. VersionOne: 3rd Annual State of Agile Survey. (Accessed 2008) Available at: http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf
11. VersionOne: 2010 State of Agile Development Survey. (Accessed 2010) Available at: http://www.versionone.com/pdf/2010_State_of_Agile_Development_Survey_Results.pdf
12. Bajek, M., Vavpotic, D., Krisper, M.: Practice-driven approach for creating project-specific software developmnet methods. Information and Software technology(49), 345-365 (2007)
13. Brinkkemper, S.: Method engineering: engineering of information systems development. Information and Software Technology 38(7), 275-280 (1996)
14. Brinkkemper, S., Saeki, M., Harmsen, A.: Meta-Modelling Based Assembly Techniques for Situational Method Engineering. lnformation Systrms 24(3), 209-228 (1999)
15. Niknafs, A., Ramsin, R.: Computer-Aided Method Engineering: An Analysis of Existing Environments. In Bellahsène Z., Léonard, ed. : Conference on Advanced Information Systems Engineering (CAiSE), p.525–540 (2008)
16. Deneckere, R., Souveyet, C.: Patterns for extending an OO model with temporal features. In Springer-Verlag, ed. : Object-Oriented Information Systems, Paris (1998)
17. Gonzalez-Perez, C., Henderson-Sellers, B.: Modelling software development methodologies: A conceptual foundation. The Journal of Systems and Software(80), 1778–1796 (2007)
18. Hug, C., Front, A., Rieu, D., Henderson-Sellers, B.: A method to build information systems engineering process metamodels. The Journal of Systems and Software(82), 1730–1742 (2009)
19. Rolland, C.: A Comprehensive View of Process Engineering. In Pernici B., Thanos, ed. : Conference on Advanced Information Systems Engineering (CAiSE), pp.1-24 (1998)
20. Software process engineering metamodel. Version 2.0. formal/2008-04-01, OMG (2008)
21. OPEN Process Framework Repository Organization: Open Process Framework. Available at: http://www.opfro.org/
22. Firesmith, D.G., Henderson-Sellers, B.,: The OPEN Process Framework. Pearson Education, London (2002)
23. International Organization for Standardization/ /International Electrotechnical Commission: ISO/IEC 24744:2007 Software Engineering -- Metamodel for Development

Methodologies.

24. Rolland, C., Prakash, N., Benjamen, A.: A Multi-Model View of Process Modelling. Requirements Engineering(4), 169-187 (1999)
25. Harmsen, A.: Situational Method Engineering: Doctoral dissertation University of Twente. Moret Ernst & Young Management Consultants, Utrecht (1997)
26. IBM: Rational Method Composer. Available at: http://www-01.ibm.com/software/awdtools/rmc/features/index.html
27. EPF Composer. In: Eclipse Process Framework. Available at: http://www.eclipse.org/epf/
28. Karlsson, F., Agerfalk, P.: Method configuration: adapting to situational characteristics while creating reusable assets. Information and Software Technology(46), 619-633 (2004)
29. Klooster, M., Brinkkemper, S., Harmsen, F., Wijers, G.: Intranet facilitated knowledge management: a theory and tool for defining situational methods. In : Conference on Advanced Information Systems Engineering, Barcelona, Spain, pp.303-317 (1997)
30. Qumer, A., Henderson-Sellers, B.: A framework to support the evaluation, adoption and improvement of agile methods in practice. The Journal of Systems and Software(81), 1899–1919 (2008)
31. Seidita, V., Cossentino, M., Gaglio, S.: Using and Extending the SPEM Specifications to Represent Agent Oriented Methodologies. In : International Workshop on Agent Oriented Software Engineering, vol. LNCS, p.46–59 (2008)
32. Cossentino, M., Gaglio, S., Henderson-Sellers, B., Seidita, V.: A Metamodelling-based Approach for Method Fragment Comparison. In : International Workshop on Exploring Modeling Methods in Systems Analysis and Design at CAISE'06, pp.419-432 (2006)
33. Cockburn, A.: Agile Software Development: The Cooperative Game 2nd edn. Addison-Wesley Professional (2006)
34. Sommerville, I.: Software Engineering 8th edn. Addison Wesley (2006)
35. Krasteva, I., Wahid, A., Ilieva, S.: Characterizing Agile Projects. Internal Report, Sofia Uniresity 'St. Kliemnt Ohridski', Sofia (2009)
36. Boehm, B: Get ready for agile methods, with care. Computer, 64-69 (2002)
37. Cockburn, A., Highsmith, J.: Agile software development, the people factor. Computer 34(11), 131-133 (2001)
38. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile software development methods review and analysis. VTT Publications(478), 3-107 (2002)
39. Krasteva, I., Ilieva, S.: Rush into Agile- Analytical Framework for Agile Practices Applicability. In : IET International Conference on Agile Manufacturing (ICAM 2007), Durham, UK, pp.229-237 (2007)
40. OMG: SPEM 2.0. (Accessed May 2011) Available at: http://www.omg.org/spec/SPEM/2.0/
41. In: OCL 2.2 Specification. (Accessed February 23, 2011) Available at: http://www.omg.org/spec/OCL/2.2/PDF/
42. Krasteva, I., Ilieva, S.: A Systematic Approach for Selection and Adoption of Agile Practices in Component-Based Projects. In Springer-Verlag, ed. : 11th International Conference on Agile Software Development, XP2010, Trondheim, Norway, p.403–404 (2010)
43. Krasteva, I., Ilieva, S., Dimov, A.: Experience-Based Approach for Adoption of Agile Practices in Software Development Projects. In : 22nd International Conference on Advanced Information Systems Engineering (CAiSE'10), Hammamet, Tunisia, pp.266-

280 (2010)

44. IBM: IBM SPSS Statistics. Available at: http://www-01.ibm.com/software/analytics/spss/products/statistics/
45. Haumer, Peter: EPF Composer Overview Part 1. (Accessed 2011) Available at: http://eclipse.org/epf/general/EPFComposerOverviewPart1.pdf
46. OMG: XMI. Available at: http://www.omg.org/spec/XMI/
47. Shaw, M.: What Makes Good Research in Software Engineering? International Journal of Software Tools for Technology Transfer 4(1), 1-4 (2002)
48. Ralyté, J., Deneckère, R., Rolland, C.: Towards a Generic Model for Situational Method Engineering. In : CAiSE 2003, vol. LNCS 2681, p.95–110 (2003)
49. Krasteva, I., Branger, P., Land, R.: A Systematic Comparison of Agile Principles and the Fundaments of Component-Based Software Development. MRTC report ISSN 1404-3041 ISRN MDH-MRTC-220/2007-1-SE, Mälardalen Real-Time Research Centre, Mälardalen University, Vasteras (2007)
50. Krasteva, I., Branger, P., Land, R.: Challenges for agile development of COTS components and COTS-based systems A theoretical examination. In : International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), Funchal, Madeira, pp.99-106 (2008)
51. In: The AGILE ITEA Project. Available at: http://www.agile-itea.org/public/info.php
52. Causevic, A., Krasteva, I., Land, R., Sajeev, A.S.M., Sundmark, D.: An Industrial Survey on Software Process Practices, Preferences and Methods., Malardalen University, Sweden (2009)
53. Causevic A., Krasteva: A Survey on Industrial Software Engineering. In Pekka Abrahamsson, Michele, ed. : 10th Agile Processes in Software Engineering and Extreme Programming (XP 2009), Sardinia, Italy, pp.240-241 (2009)
54. Land, R., Sundmark, D., Lüders, F., Causevic, A., Krasteva, I.: Reuse with Software Components – A Survey of Industrial State of Practice. In : 11th International Conference on Software Reuse, Falls Church, VA, pp.150-159 (2009)
55. SEMAT initiative: SEMAT: Software Engineering Method and Theory. Available at: http://www.semat.org/bin/view