

## ВЪЗМОЖНОСТИ НА СЕМЕЙСТВОТО ГЪВКАВИ ПОДХОДИ “AGILE” ЗА ПОДОБРЯВАНЕ НА ПРОЕКТНОТО УПРАВЛЕНИЕ В МСП

Кирил Георгиев, Недка Николова, Розалина Димитрова

**Abstract:** Настоящата статия обобщава резултатите от изследванията за идентифициране на възможностите за прилагане на семейството гъвкави подходи AGILE в проектното управление на МСП в България. Анализират се същността, съдържателните характеристики, концептуалния модел и технологията на тяхното реализиране. Особено място е отделено на добрите практики като подхода SCRUM и KANBAN в управлението на проекти при разработването на софтуер. Основната хипотеза на авторите е, че семейството подходи AGILE е ефективен инструмент за подобряване на проектното управление в МСП, когато се налага бърза реакция на чести промени. Със своя методически инструментариум семейството подходи AGILE дава възможност за развитие и усъвършенстване на проектния управление и повишаване на конкурентоспособността на МСП.

**Key words:** AGILE, SCRUM, KANBAN, LEAN, управление на проекти, малки и средни предприятия МСП

### 1. Въведение

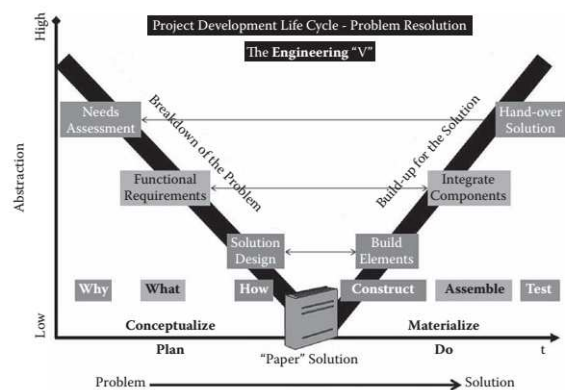
Проектът е начинание, което изисква нов тип организация и мобилизиране на всички ресурси за постигането на предварително зададени и уникални цели и задачи в рамките на ограничен бюджет и срок. От тук се извеждат и основните характеристики на всеки проект: неповторимост, ограничение във времето и ресурсите, осъществяване в условия на динамика и несигурност/риск. Класическият подход за изпълнение на проекти е последователен, „водопаден“ (Waterfall) или „преминаващ през поредица от врати/порти“ (Stage-gate) процес, който е подходящ при стабилна и предсказуема среда [5]. МСП се нуждаят от нови инструменти, повишаващи ефективността на проектното управление, когато се налага бърза реакция на чести промени в динамична рискова среда. За подобряване и ускоряване на проектния процес в последно време се използват семейство нови гъвкави подходи-т.н. AGILE-подходи, вкл. SCRUM, KANBAN и LEAN [3].

### 2. Особенности на проектното управление

Управлението на един проект се различава от класическото фирмено управление/мениджмънт [4],[5].

Всеки проект включва някаква промяна, която предизвиква различни реакции-от еуфория до съпротива, от скептицизъм до засилена мотивация. Това налага проектният мениджмънт да посреща и разрешава множество предизвикателства.

Всеки проект решава някакъв проблем. Правят се изследвания и се търси решение, което се концептуализира, след което се материализира. Това може да се изобрази чрез класическия V-образен жизнен цикъл [ 4 ].



Фиг.1.Проектът като решаване на проблем[4]

Всеки проект е ограничен във времето. Сроковете притискат изпълнителите да приключват работите в зададените срокове, което се отразява негативно на обхвата и качеството.

Обикновено проектът съдържа нещо уникално/неповторимо и води до иновации-въвеждане на нови машини и/или технологии, разработване на нови продукти/софтуер/услуги, излизане на нови пазари и т.н. Това затруднява изпълнението и изисква добиване на ново знание.

Обикновено проектът е доста сложно, комплексно мероприятие - изисква съгласуване, сътрудничество и коопериране

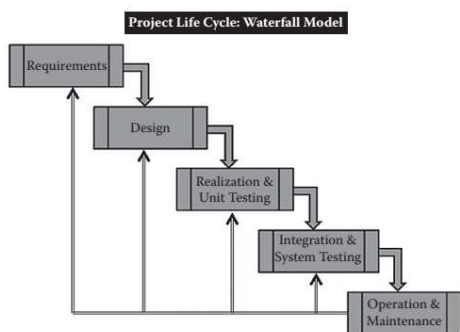
между различно мислещи хора и организации с различни цели (дори различни държави). Поради необходимостта от пълно мобилизиране на разнородни и обемни ресурси за постигане на уникални цели в ограничено време, всеки проект съдържа значителен риск, който трябва да се избегне, прехвърли, намали или да се приеме. Тези особености на проектното управление се проявяват с особена сила при МСП в България.

**3. Системен подход при управление на проектите**

Той е заложен при разработване на общите изисквания за управление на проекти в БДС-ISO 21500: 2012. Неговото приложение предполага използването на т.н. процедури “отгоре-надолу” и/или „отдолу-нагоре“ (top-down-top), изследването на алтернативи за избор на най-добрата, подреждане на дейностите във фази и напредване/постигане на целите като решаване на проблеми [ 5 ].

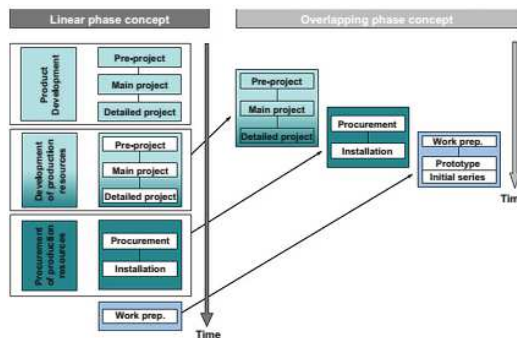
**4. Жизнен цикъл на проекта ЖЦП**

Подобно на жив организъм проектът се развива във времето, като преминава през различни фази. Типичните фази са: инициране, планиране, изпълнение, контролиране и приключване. Класическият модел на ЖЦП- т.н. “водопаден“ модел предполага изпълнение чрез последователно постигане на предварително определени резултати/цели/задачи (deliverables).



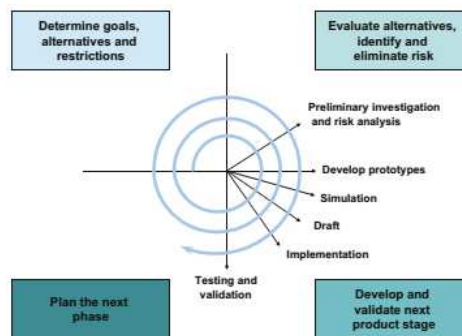
Фиг.2. “Водопаден“ модел на ЖЦП [4]

За ускоряване на изпълнението се прилага паралелен (частично или напълно) подход в различни варианти- например т.н. модел „с припокриване/застъпване“ на фазите [ 5 ]:

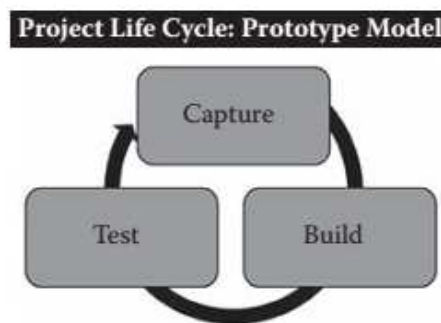


Фиг.3 Модел “със застъпване“ на ЖЦП [5]

Възможно е също циклично повторение на фазите с нарастване на резултата – т.н. „спирален“ модел [5] или т.н. “прототипен“ модел [ 4 ]:

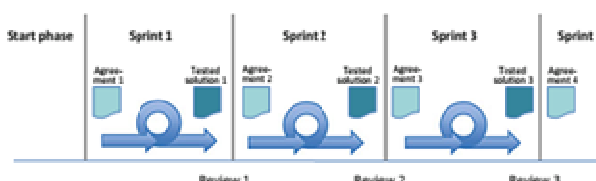


Фиг.4. “Спирален“ модел на ЖЦП [5]



Фиг.5. „Прототипен“ модел на ЖЦП [4]

Един от най-съвременните начини за управление на проектите представлява семейството на т.н.гъвкави (Agile) подходи, което досега се прилага за ИКТ (фиг.6)



Фиг.6. „Гъвкав“ (agile) модел на ЖЦП [5]

Съществуват благоприятни условия гъвкавите подходи да се разпространят/внедрят в практиката на проектното управление на МСП извън сферата на ИКТ.

##### **5. Концепция и базови понятия на семейството гъвкави подходи AGILE**

Семейството гъвкави (agile) подходи за разработка е неформален сбор от методи и техники за управление на проекти за разработка на продукти- основно на софтуер.

В основата на гъвкавата методология стои идеята, че разработката на продуктите е динамичен процес, в който дългосрочното планиране има ограничена ефективност.

Гъвкавата методология намира особено широко приложение в разработката на продукти, където чрез бързото и многократно създаване на прототипи, производителите имат възможност да получат обратна връзка от клиентите и да адаптират разработката към непрекъснато променящите се изисквания.

Към този момент най-голямо приложение гъвкавата методология е намерила в ИКТ.

Гъвкавата методология за разработка на софтуер се състои от група методи при създаването на софтуер, базирани на повтарящо се/ итеративно и постепенно разработване. Софтуерните спецификации и решения се развиват поетапно чрез сътрудничество между самоорганизиращи се екипи, съставени от хора с различни функции.

Гъвкавата методология насърчава адаптивно планиране, еволюираща разработка и доставяне на софтуера, времево-разпределен итеративен подход и поощрява бързото и гъвкаво реагиране на промени. Това е концептуална рамка, която насърчава взаимодействията между разработчици и клиенти през цялото времетраене на разработката на софтуерните проекти.

Така наречените леки, гъвкави методи за разработка на софтуер се развиват в средата на 90-те години като реакция срещу тежките традиционни „методи на водопада”, които са критикувани за употребата на силен контрол, правила, микро-управление и прекалено последователен подход на разработване.

През февруари 2001 г. 17 разработчици на софтуер се срещат в курорта Snowbird,

Юта,САЩ, за да обсъдят гъвкавите методи за разработка. Те публикуват Манифест на гъвкавите методологии The Agile Manifesto [3], който дефинира подходите, сега известни като гъвкави (agile) методи за разработка на софтуер. Част от авторите на манифеста създават The Agile Alliance - неправителствена организация, която насърчава разработката на софтуер в съответствие с принципите на манифеста. The Agile Manifesto се основава на дванадесет принципа [3]:

1. Удовлетворение на клиентите чрез бърза доставка на полезен софтуер;
2. Промяна в спецификациите е възможна, дори и в късните фази на проекта;
3. Често предоставяне на работещ софтуер (в периоди от седмици, а не месеци);
4. Работещият софтуер е основната мярка за напредък;
5. Устойчиво развитие, което успява да поддържа постоянно темпо;
6. Тясно, ежедневно сътрудничество между бизнес служители и разработчици;
7. Разговорите лице в лице са най-добрата форма на комуникация (съвместна локация);
8. Проектите се изграждат около мотивирани хора, на които се има доверие;
9. Непрекъснато внимание към техническо съвършенство и добър дизайн;
10. Простота - изкуството максимално количество работа да бъде пропусната без ущърб на качеството, е от съществено значение;
11. Самоорганизиращи се екипи;
12. Редовна адаптация към променящи се обстоятелства;

Между 2001 г. и 2010 г. редица индивидуални разработчици и групи правят допълнения и подобрения към този Манифест.

Гъвкавите методи за разработване разделят задачите на малки стъпки с минимално планиране, без да засягат дългосрочното планиране на проекта. Итерациите (етапите) стават в кратки срокове (timeboxes), които обикновено траят от една до четири седмици. През всяка итерация екипът, съставен от хора с различни функции, работи по всяка една от функциите: планиране, анализ на изискванията, проектиране, разработка,

тестване и проверка при приемане. В края на итерацията работещият продукт/софтуер се представя пред заинтересованите страни, вкл. клиентите. Така се намалява цялостния риск и проектът може да бъде адаптиран бързо към промени. Една итерация може да не добавя достатъчно функционалност, за да обоснове пускане на пазара, но целта е да получим работещо решение (с минимални грешки) в края на всяка итерация. За да се завърши даден софтуер или функционалност, могат да бъдат нужни множество итерации.

Независимо какви средства (езици, системи) за програмиране се използват, всеки екип включва представител на клиента. Той се назначава от заинтересованите страни и действа от тяхно име, като има личен ангажимент да бъде на разположение на разработчиците за отговори на въпроси, възникнали по време на дадена итерация. В края на всяка итерация, заинтересованите страни и представителят на клиента преглеждат заедно напредъка на проекта и оценяват приоритетите с оглед оптимизиране на възвращаемостта на инвестициите (ROI) и съобразяване с нуждите на клиента и целите на производителя.

Обединяващи за всички гъвкави методи за разработка са ежедневните срещи относно напредъка на проекта или т.н. стенд-ъп (stand-ups) срещи. По време на тези срещи, членовете на екипа докладват пред всички какво са направили предишния ден, какво възнамеряват да правят днес и има ли нещо, което ги възпрепятства да завършат задачите си.

Специфични инструменти и техники, като например непрекъснатата интеграция, автоматизирани xUnit тестове, програмиране по двойки, разработка чрез тестове (test-driven development), шаблони за дизайн (design patterns), дизайн според областта (domain-driven design), преработка на код (code refactoring) и други техники, често се използват за подобряване на качеството и повишаване на гъвкавостта на проекта.

В гъвкавите методологии за разработка на софтуер се използват т.н. информационни радиатори (напр. дъска с листчета) – те онагледяват физически прогреса на проекта на видно място в офиса, където всички могат

да го видят. Могат да бъдат използвани светлинни и други индикатори, информиращи екипа за текущото състояние на проекта.

## **6. Сравнение с други методологии**

Съществуващите методи за разработване на проекти варират от „адаптивни“ през „формални“ до „планиращи“ методи [4], [5].

Гъвкавите (Agile) методи/подходи попадат в „адаптивната“ част на този спектър. Всички адаптивни методи залагат на бързо приспособяване към променящата се реалност без детайлно анализиране и планиране на подалечното бъдеще. Когато нуждите на един проект се променят, адаптивният екип също се настройва съответно на новите реалности. За разлика от тях планиращите методи за разработване се фокусират върху детайлното анализиране и планиране на бъдещето и справяне с предвидимите рискове. В идеалния случай, един планиращ екип може да даде точен списък на функционалностите и задачите, планирани в целия ход на процеса на разработка. Планиращите методи разчитат на пълен ефективен предварителен анализ и ако той се окаже погрешен, проектът може да изпита съществени затруднения и да промени посоката си на развитие. За разлика от адаптивните и планиращите методи, формалните методи прилагат информационната теория и широк спектър от подходи за формална проверка за коректност. Един формален метод цели доказването на коректност с определена степен на детерминизъм. Някои формални методи са базирани на проверка за непротиворечивост на модела и намират контра-примери в случаите, когато непротиворечивостта не може да бъде доказана. Гъвкавите методи могат да прилагат формални методи с висока степен на строгост [5].

Естествено, че адаптивните и планиращите методи могат да се комбинират при разработване на проекти.

## **7. Кратък преглед на основните гъвкави подходи**

### **7.1. Скръм (Scrum)**

Scrum е подход, използван при изготвяне и управление на големи проекти. Той е разработен с цел дългосрочното планиране на

изработването на даден продукт да бъде значимо улеснено. За разлика от типичния мениджмънт чрез контрол и командване, при Scrum се набляга на обратната връзка и се дава повече власт на хората, вършещи „черната работа“.

#### 7.2. Канбан (Kanban)

Канбан е подход за управление на интелектуални дейности с наблягане на доставката/завършването точно на време, без същевременно членовете на екипа да са пренатоварени. Името произлиза от японски и буквално преведено означава „сигнална карта“. Подходът, както е формулиран от Дейвид Андерсън [1], е преход към постепенен, еволюционен процес. Използва система за ограничаване на текущата работа като основен механизъм, за да разкрива системно операционни проблеми и да стимулира взаимопомощта, с цел непрекъснато подобряване на системата.

#### 7.3. Леан (Lean) или пестеливо (оскъдно) разработване

Терминът „Lean- софтуер разработване“ произлиза от едноименната книга, написана от Мери и Том Попендийк [ 3 ]. Подходът използва традиционните Lean принципи в модифициран формат, както и набор от 22 инструмента и ги сравнява с гъвкавите практики.

#### 7.4. Екстремно XP програмиране

Екстремното програмиране (Extreme Programming - XP) е друг вид методология за създаване на софтуер. Основната цел на XP е да редуцира цената на проект, ако се наложи дадена промяна. Оттук се прави извода, че XP е методология, подходяща да се използва при проекти, които имат често променящи се изисквания и при които по-стандартни методологии (като Waterfall модела например) не са оптимални за постигане на голяма производителност; подходяща е при проекти, които включват нови технологии или непредвидими проблеми, свързани с имплементацията; използва се също така при по-малки и по-лесни за реализация проекти с неофициални методи; добра технология за проекти, изискващи/включващи изследване.

### 8. Подходът SCRUM по-подробно

SCRUM представлява итеративна гъвкава рамка за управление на проекти. Той е създаден през 1987г., като името произлиза от думата scrummage - “ спорна топка” (англ.). Идеята е взимствана от играта „Ръгби“, в която след кратки прекъсвания за нарушения играта се рестартира отначало. Макар че подходът SCRUM е първоначално предложен за управление на проектите за разработване на продукти, той става все по-приложим при управлението на софтуерни проекти и може да бъде използван за да задвижва екипи за софтуерна поддръжка или като общ подход за проектен / програмен мениджмънт. Тази методология е променила възприятията за типичното управление на проекти, като ясно показва предимствата на гъвкавите пред последователните водопадни или „твърди“ методологии.

SCRUM-процесът се състои от отделни итерации, наречени „спринтове“ с продължителност от една до четири седмици. В края на всеки спринт, екипът създава работеща версия на прототип на продукта, която включва всички готови задачи от backlog-a.

SCRUM е разработен за тези фирми, чиято верига на стойността (изобразяваща процеса от началото до края на продукта и всяка стъпка, на която се добавя стойност) прави дългосрочното планиране на продукта доста трудно, когато изискванията към продукта се променят много бързо и често. За разлика от типичния мениджмънт чрез централизирано ръководене и контрол, тук се набляга на обратната връзка и се дава повече власт/права в ръцете на хората, които извършват операциите по процесите. Елементите, изграждащи подхода, се състоят от собственик на продукта (product owner) или гласът на клиента, SCRUM – ръководител, срещан още като “ господар“ (master) и разработващия екип (team). Дейностите на този екип са в неразривна връзка и зависещи една спрямо друга.

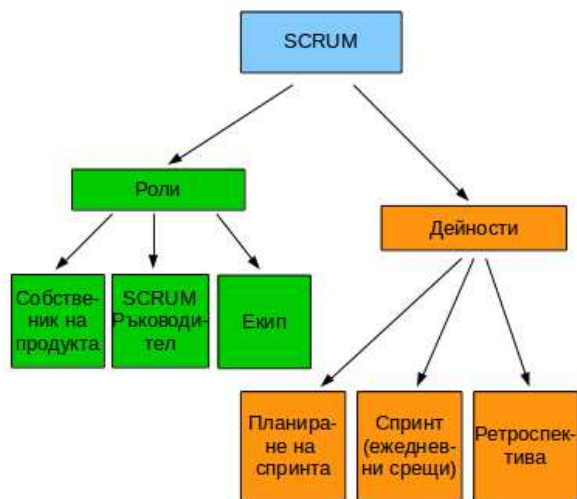
Собственикът на продукта е лицето, което организира приоритизиране на задачите, на база изискванията на клиента. Тъй като за екипа е важно, кои задачи са спешни за изпълнение и кои търпят отлагане във

времето, приоритизирането е базова дейност при изграждане на един (софтуерен или друг) проект.

SCRUM – ръководителят е “гръбнака” на екипа. Той е отговорен за това, хората в екипа да изпълняват своите дейности спрямо поставените цели и в съответствие с методологията SCRUM.

Разработващият екип извършва така наречената “черна работа” – например анализира, разработва, тества и внедрява дадено приложение. Състои се от 3 до 9 души, които имат различни, но сходни умения и отговорности.

SCRUM процесът се състои от отделни итерации, наречени спринтове, които могат да имат продължителност от една до четири седмици. За това време, екипът създава завършени части от даден продукт. В края на всеки спринт, екипът разполага с работеща версия на продукта, която включва всички готови задачи, които са били предварително приоритизирани.



Фиг.7 Характерни роли и дейности при SCRUM

Характерни за всеки спринт са ежедневно повтарящите се срещи, които са в рамките от 5 до 15 минути. На тези срещи се правят анализи на изминалия ден и планират действия за текущия.

Ключов принцип на SCRUM е това, че още в самото начало на проекта се приема, че изискванията няма как да са пълни, ясни и

напълно разбрани. Промените от страна на клиента са очаквани. SCRUM се фокусира върху способността на екипа да доставя бързи решения и да е готов да отговори бързо на неочакваните промени. Това е положителна черта за SCRUM, защото резките промени не могат да бъдат добре осъществени с традиционните методи.

## 9. Състояние и перспективи за приложение на подхода SCRUM в българските фирми

В България приложението на подхода SCRUM е ограничено, но набира все по-голяма популярност. Може да бъде дадена аналогия с жизнения цикъл на продукта, където подходът заема място в началото на етапа “Растеж” от своето развитие. Голяма част от фирмите извън сектора на ИКТ не го познават, в някои организации (предимно в ИКТ) подхода е в процес на обмисляне, в други на изпробване, а в трети на използване от дълго време.

Филиалите на чуждестранните (ИКТ и др.) фирми у нас отдавна използват SCRUM в своята дейност. Пример за българска фирма, прилагаща подхода SCRUM е Webventure.Ltd, занимаваща се с ИТ – консултинг повече от десет години. Фокусирайки силите си в задоволяване на вкусовете и предпочитанията на своите клиенти, организацията базира своята дейност на “гъвките” методи за управление. Използвайки стандартни подходи, не може да бъде осъзната нуждата на отделния клиент. Фирмата възприема себе си, не като доставчик на услуги, а като партньор на своите клиенти. За повече от десет години опит, в изработка на големи и малки проекти, фирмата прилага принципите на гъвкавото разработване, като това ѝ дава възможност да реагира адекватно на външната среда.

Осъществено от нас пилотно/непредставително анкетно проучване показва, че МСП, разработващи нови продукти или внедряващи нови технологии например в хранителната, мебелната, облекло и обувки и др. индустрии, са склонни да изпробват/използват подхода SCRUM, като по този начин отчетат по –добре изискванията на клиентите.



## 10. Заключение

(1)Класическите подходи за управление на проекти (водопаден и др.) не работят добре в условията на бързи промени в средата, промени в изискванията на клиентите.

(2)Семейството гъвкави AGILE подходи като SCRUM, KANBAN и др. апробирани предимно в областта на софтуерни проекти, създават добри условия за проектно управление в динамични условия и риск.

(3)Съществуват благоприятни възможности българските МСП да използват в своята практика гъвкавите подходи за проектно управление (особено SCRUM), с което полесно ще преодолеят затрудненията поради бързите промени в средата и ще повишат своята конкурентоспособност.

### За контакти:

9010 Варна, ул. “Студентска”1  
Технически университет - Варна

доц. Кирил Георгиев  
e-mail: kirilvg@gmail.com

Розалина Димитрова  
e-mail: joanica@abv.bg

### Литература:

- [1] Хенрик Книберг & Матиас Скарин, Канбан и Скръм - как да вземем най-доброто и от двете, С4Media Inc. (издателство на InfoQ.com.), 2010
- [2] Alan Moran, Managing Agile Strategy-Implementation, Organisation and People, Springer, 2015
- [3] Charles G. Cobb The Project manager’s guide to mastering agile- Principles and Practices for an Adaptive Approach, Wiley, 2015
- [4] Claude H. Maley, Project Management Concepts, Methods, and Techniques, CRC Press, 2012
- [5] Juorg Kuster etall, Project Management Handbook, Springer, 2015 .