

11. Гъвкав подход (Agile) за създаване на софтуерни системи

Манифест на подхода Agile 2001– основни ценности на подхода

- **Индивиди и взаимодействия** е над процеси и инструменти
- **Работещ софтуер** е над изчерпателната документация
- **Постоянното сътрудничеството с клиента** е над договарянето
- **Приемане и отговаряне на промените** е над формалното следване на дефиниран план

Тоест, докато има стойност в елементите вдясно, ние ценим артикулите вляво повече.

Agile (гъвкав) подход- е технически подход за това как да се изпълняват проекти. Този подход не посочва какакви и кои точно инструменти и процеси, кога и как трябва да се използват при управлението и реализацията на проекти. Agile е общия подход обединяващ много други методи и методологии за управление на проекти, например PMI Agile, Scrum, Lean, Kanban, Extreme Programming (XP), Scaled Agile и др.

- **Някои основни сертификации в Agile са:**
 - PMI-ACP: PMI Agile Certified Practitioner – PMI.org
 - Professional Scrum Master (PSM I and II) – Scrum.org
 - Certified Scrum Master (CSM) certification - Scrum Alliance
 - Lean Six Sigma certifications and Kanban framework certifications
- Agile се съсредоточава около **постепенни и итеративни стъпки** за реализация и завършване на проекти. Тези стъпки се реализират чрез краткосрочни цикли на развитие с времетраене от две до няколко седмици.

- Подходът дава приоритет на **бърза доставка, бързо адаптиране и приемане на промените и постоянно засилено сътрудничество**, а не само на стриктно и формално управление отгоре надолу и следване на строго дефиниран план.
- Този подход контрастира с традиционните форми на управление на проекти, които обикновено напредват линейно във времето и преминаването на проекта през добре формулирани етапи и завършени на планиране, проектиране, изпълнение и затваряне
- В Agile процесите за реализация на проект поддържат **постоянна обратна връзка с клиента**, която позволява на членовете на екипа да се адаптират към промените и новите предизвикателства, ако и когато те възникнат, а на заинтересованите страни дава възможност да общуват постоянно, по често и последователно.
- С най висок приоритет е **удовлетворения клиент чрез ранна и непрекъсната доставка на работещ софтуер**
- Трябва да се приемат променящите се изисквания, дори ако са дадени късно в реализацията на проекта. Този процес осигурява конкурентно предимство на клиента и му дава възможност да отговори адекватно на бързо променящите се изисквания на бизнеса
- Необходими са доставки по-често на **инкременти на документиран работещ софтуер**, напр. от две седмици до два месеца, е за предпочитане пред цялостното доставяне на софтуера в края на проекта с финална изчерпателна документация
- Бизнес потребителите и разработчиците трябва да осъществяват постоянно **взаимодействие и комуникация на ежедневна база** и това е за предпочитане пред създаване на процеси и инструменти. Необходимо е да се направи всичко възможно за да

се осигури **тясно и постоянно сътрудничество с клиентите** при договаряне на проекта

- Реализацията на проекта трябва да се извърши от **мотивирани и осигурени хора**, които да имат поддръжката от организацията и необходимата за проекта екипна среда. Доверете им се, че те могат и ще извършат съответната работа

- Най добрия и ефективен начин аз комуникация е **комуникацията лице в лице**

- **Работещият софтуер е основния критерий за прогрес на П.**

- Непрекъснатото внимание към **постоянно техническо развитие на разработчиците и тяхното осъвършенстване**, както и фокус върху **адекватната софтуерна архитектура и добрия потребителски интерфейс и дизайн** на системата осигуряват основа за постигане на **успеха на проекта**

- Гъвкавите процеси насърчават устойчивото развитие на проекта във времето. Бизнес спонсорите, разработчиците и потребителите трябва да бъдат в състояние да поддържат постоянно темпо през цялото време на проекта

- **Простотата на разработката** е от съществено значение за доброто възприемане на системата от страна на потребителите и максимизиране на ефекта от реализираната работа

- Най-добрите архитектури, изисквания и дизайни се създават и реализират от **самоорганизиращи се екипи**

- На **редовни интервали екипът трябва да анализира свършената работа**, да обмисли и реши за насоки как да стане по-ефективен.

- След този анализ и решения, екипът трябва да се **самонастрои** за да работи по качествено и по ефективно според приетите насоки

Agile – управление на създаването на софтуер

Софтуерът се създава чрез итерации:

- **Итерацията** е препоръчително да бъде **не повече от 4 седмици**, но това зависи от естеството на проекта.

Допустимо е да се започне с по-голяма продължителност (напр. 8 седмици или 2 месеца)

- **Фокус на проекта** е да се **произведе и достави полезен софтуер** за бизнеса и потребителите

- Трябва да се направят **толкова итерации, колкото са необходими**, за да се достави **полезния софтуер**

- **Оценката на работата** се извършва на **малки части от разработката**, адекватни на **съответните итерации**, а не на по-големи части или на целия проект

- **Оценката на работата** се извършва от **хората, които ще я реализират и внедрят**

- Добре е оценката да се извърши или да помага някой, който е вършил същата или подобна работа преди

- Оценката да се представи като диапазон.

- Колкото по-далече е доставката на съответната порция софтуер, толкова по-голям е диапазонът на съответната оценка.

Например, ако оценяваме цената на шестмесечен проект в началото на проекта, може да има диапазон от +/- 20%. Ако има един месец до съответната доставка, диапазонът може да бъде +/- 5%.

- **Актуализиране на оценката** по време на целия проект. Когато постъпват актуализирани и действителни изисквания и промени за разработката в проекта, тогава има и по-добра представа за това какъв ще **обхвата и бюджета** и какво ще е **времетраенето** на проекта
- Трябва да се избере техническа среда, архитектура и дизайн за реализация и на тази база да се изберете съответния agile подход, който отразява тази среда.
- Може точно да се планира в дълбочина и в детайли само за най-близките задачи
- Хората, които вършат работата, трябва да участват активно в планирането
- Ако е възможно, разработчиците в екипа да могат да избират техническата работа, особено ако са я работили и преди. Така те ще могат да оценят и извършат работата по най-ефективен начин
- Препоръчително е да се организира **разработката в кратки итерации (от 2 до 4 седмици)**
- Да се включат и планират задачи за комуникация и презентация пред външни за проекта бизнес групи и потребители.
- Когато трябва да се **демонстрира разработената софтуерна система на други заинтересовани страни вътрешни на проекта или да се създаде специална демонстрационна или тестова версия**, тогава задачата трябва да се **включи в графика** на проекта
- Не забравяйте да включите в задачите и **обучение**
- Избраният подход, трябва да е съобразен с изискванията, комуникацията с клиента и да отразява избраната среда

- **Скорост на изгаряне (time to value).** Колко пари/време/ресурси са необходими за реализация итерациите в П и как се оценява прогреса на разработката на качествен софтуер и най-важното неговата навременна доставка и внедряване според итерациите по П
- **Честота и скорост на доставките.** Доставената и внедрена бизнес функционалност, адаптирана и отговаряща на реалните изисквания, включително и промените, ако и когато ги има
- **Доставката на работещ софтуер** е единствената истинска мярка за прогрес и напредък
- **Точност** на доставките. Колко функционалност е било планирано и доставено по време на итерациите
- **Намаление на дефектите в софтуера.** Чрез анализ на дефектите да се определи какви са дефектите и да се оцени тежестта на тези дефекти. Като резултат трябва да се дефинират решения за намаляване на дефектите и техния негативен ефект върху реализацията на проекта

Предимства и предизвикателства

Предимства	Предизвикателства
Възможност за управление при промяна бизнес приоритетите	Ораганизациите и бизнеса не са съгласни и не подкрепят промяната на модела и даже се противопоставят на него
Увеличена прозрачност на работата в проекта поради малките стъпки за създаване на софтуер	Екипите не могат да се самоорганизират или използват множество различни методи, технически средства и практики

Подобрена връзка и комуникация между разработчици и бизнес	Изисква много сериозна подкрепа от ръководството за промяна на модела на работа, както и сериозен бюджет за мотивация на екипа
Повишена възможност за бързи промени и по добра конкурентноспособност в зависимост от промените в бизнеса	Необходим е сериозен контрол и управление на реализацията на проекта за да се спази бюджета и отговори на очакванията
Намален риск и увеличена предсказуемост поради честата и бърза доставка на резултатите	Реализацията на ежедневна и ефективна комуникация изисква сериозна ежедневна ангажираност и отделено време от страна на бизнеса и потребителите. Това представлява допълнително сериозно натоварване като ресурси и бюджет както и нова комуникационна култура

Кога да се използва Agile

- **Принципите на Agile** – адаптивност, повторение, непрекъснатата доставка и кратки времеви рамки и др – го правят стил на управление на проекти, който е по-подходящ за текущи проекти и проекти, при които множество детайли не са известни от самото начало на П
- Когато има готов и добре мотивиран финансово и подготвен технически екип, който е вършил същата работа и преди този проект
- Когат П няма ясни изисквания, ограничения, срокове, бюджет
- Когато потребителите и бизнеса са наясно с изискването за ежедневна ефективна комуникация с разработчиците и са готови да участват в нея

- Когато, както се казва нищо не е записано на камък. Agile работи наистина добре, когато визията или функциите на продукта не са добре дефинирани в началото.
- Когато е жизненоважно да се реализират нови изисквания и функции и бизнесът да отговори бързо на новите предизвикателства
- Когато е императивно да се получи обратна връзка много по-рано в проекта
- Когато може постоянно да се работи върху идентификация на грешките и тяхното отстраняване
- Когато няма фиксирана крайна дата и бюджет на проекта и те са зависими от разработките и доставките на софтуер в проекта
- Когато възникват често **нови бизнес изисквания и те трябва да се въведат нови промени.** Agile позволява на собствениците на продукти да коригират изискванията и приоритетите по пътя, за да се възползват от възможностите и в крайна сметка да доставят по-адекватен към времето продукт от проекта
- Когато за да се отговори на промените и за да внедри нова функционалност, разработчиците работят само няколко дни или дори само часове, а не седмици и месеци, за да анализират промените и техическото решение и да я внедрят
- Когато е ограничено времето за предварително планиране за да се започне с проекта. Това означава бърз старт и реализация възможно най бързо на работещ софтуер

Недостатъци на Agile

- Може сериозно да надхвърли предварително планирания бюджет, особено когато се правят недобре обмислени и съобразени с наличния бюджет промени.
- Първоначалните оценки могат да бъдат трудни и даже използването на диапазон на оценките да се окаже невярно (напр. вместо в диапазон от 20%, реализацията да се окаже 75% по висока)
- Изисква голямо взаимодействие с бизнес хората и непрекъснат контрол на изразходваните средства и постигнатите резултати. Това отнема сериозно време от техните непосредствени задължения, увеличава сериозно разходите и води до отрицателна нагласа към П
- В крайна сметка проектът може да се отклони от първоначалния план и да не удовлетвори първоначалните очаквания
- Докато Agile проектите се фокусират върху непрекъснато развиващите се бизнес нужди и изисквания на клиента, ако П има непроменящи се изисквания и статични цели и задачи, по-добре е да се използват традиционни методи за разработка и управление на П
- Традиционните подходи за управление на проекти могат да бъдат по-лесни за планиране и прогресът може по-лесно да се измерва и контролира
- Ако бюджета е предварително фиксиран и не може да се променя по време на проекта
- Традиционните подходи са препоръчителни **когато времето за реализация и ресурсите са основни ограничители и когато разработващият екип няма достатъчно предварителен опит със съответния тип проекти**

- Когато се създават постоянно **нови проекти отначало** в различни бизнес области и **няма изграден конкретен опитен екип** за конкретната реализация
- Когато мотивацията на екипа (особено при екип от вътрешни разработчици и външни консултанти) не е достатъчна и желанието му да извърши работата не е достатъчно
- Когато не може да се осъществи ежедневна комуникация и когато се очаква екипите да работят дълго време независимо от заинтересованите страни

Кога да използвате традиционни методи (обобщение)

- За проекти с **фиксирана цена/бюджет – фиксирана продължителност**, при който спонсорите се нуждаят от **добре направена оценка за пари, време и продължителност** и **ограниченията на П не са много гъвкави**, ако изобщо са гъвкави.
- Когато повечето **изисквания на високо ниво са зададени в началото и не се променят**
- Когато е **задължително да има процес за контрол и управление на промените**, за да се избегне неконтролирано увеличаване на обхвата (scope creep) и да има достатъчно време и ресурс за да се анализират и оценят промените
- Когато е необходимо да се направи **предвариелно техническа спецификация, архитектура и дизайн**, които не се променят във времето
- Когато жизненият цикъл на разработка на софтуер следва предимно последователен (Waterfall) подход