

# Циклични модели

## 1. Итеративен (цикличен) модел

При итеративния модел продуктът се развива чрез повтарящи се цикли на работа, наречени **итерации**. Във всеки цикъл се планират дейности, разработва се или се подобрява част от продукта и се проверява полученият резултат.

Не е необходимо всички подробности на изискванията да бъдат окончателно уточнени още в началото. Проектът обаче трябва да има ясна основна цел и начални изисквания. Моделът позволява:

- постепенно уточняване на изискванията;
- подобряване на вече създаденото;
- добавяне на нови функционалности;
- промени въз основа на проверки и обратна връзка.

**Итеративно развитие** означава подобряване на резултата чрез повторение на дейности.

**Инкрементално изграждане** означава постепенно добавяне на нови части или функционалности. Двата подхода често се съчетават.

**Пример:** Първо се създава основна версия на училищен сайт. След обсъждане с потребителите се подобрява навигацията. В следващ цикъл се добавя календар на събитията.

## 2. Основен принцип на цикличните модели

Работата се организира в последователни цикли. Във всеки цикъл екипът:

- определя какво ще разработи или подобри;
- планира и изпълнява необходимите дейности;
- проверява резултата;
- обсъжда го с възложителя или потребителите;
- използва обратната връзка при следващия цикъл.

Един цикъл може да се разглежда като малък проект със собствена цел, дейности и резултат.

**Важно:** Цикълът не е равнозначен на отделен етап като „проектиране“ или „тестване“. В рамките на един цикъл могат да се извършват и проектиране, и разработване, и тестване.

## 3. Процеси при цикличните модели

**Процесът** е свързан набор от дейности за постигане на определена цел.

За да разберем управлението на работата, можем да разграничим пет групи процеси:

1. **Инициране** — определяне на целта и рамките на проекта или конкретния цикъл.
2. **Планиране** — определяне на задачите, ресурсите, отговорниците и сроковете.
3. **Изпълнение** — извършване на планираната работа.
4. **Наблюдение и контрол** — проследяване на напредъка, сравняване с плана и предприемане на корекции.
5. **Приключване** — обобщаване и документиране на постигнатия резултат.

Тези групи процеси се срещат и при други подходи за управление на проекти. При цикличното разработване планирането, изпълнението и проверката се повтарят за следващите подобрения на продукта.

**Наблюдението и контролът съпътстват работата, а не се извършват само след нейното завършване. Ако се открие проблем, се коригират необходимите дейности. Не е задължително целият цикъл или проект да започва отначало.**

**Задача 1.** Разгледайте схемата. Обяснете как наблюдението и контролът могат да доведат до промяна на плана или до корекция на изпълнението. Дайте пример с разработване на училищен сайт.



### Цикличен модел на управление на проект

Схемата показва връзките между процесите и възможността за корекции чрез наблюдение и контрол. След проверка на резултата екипът определя необходимите подобрения и планира следващата работа.

## 4. Прототип и междинна разработка

**Прототипът** е предварителен модел на продукта, чрез който се изпробват идеи и се уточняват изискванията.

Той може да бъде:

- макет на външния вид;
- интерактивна демонстрация;
- работеща версия с ограничени функции.

Прототипът помага на възложителя и потребителите да разберат предложеното решение и да дадат обратна връзка, преди да е завършен целият продукт.

**Пример:** Преди изграждането на училищен сайт се създава интерактивен макет на началната страница и менюто. Потребителите изпробват навигацията и посочват какво ги затруднява.

Прототипът може да бъде доразвит или заменен с друго решение. Той не е задължително начало на всеки цикличен проект.

**Важно:** Тестването не се ограничава до прототипа. При всеки цикъл се проверява дали новите или променените функции работят правилно и дали не нарушават вече работещите части на продукта.

**Междинна разработка** е резултат, създаден преди окончателното завършване на продукта. Тя може да бъде прототип, отделен разработен компонент или версия с част от предвидените функции. Използва се за проверка на напредъка, обсъждане и последващо доразвиване.

**Задача 2.** Разгледайте примера с построяването на къща. Сравнете **водопадния** и **итеративния** подход. Обяснете кога собственикът получава възможност да даде обратна връзка и кои строителни дейности трябва да следват определен ред.

#### **Водопаден модел**

Прави се подробен проект (от строителя заедно със собственика) каква трябва да е къщата и как трябва да изглеждат стаите. Строителят прави план как това ще се осъществи.

1. Построява се къщата до груб строеж.
2. Прокарват се инсталациите.
3. Поставят се врати и прозорци.
4. Извършва се фина замазка.
5. Боядисват се стени и тавани.
6. Слага се подова настилка – плочки, паркет.
7. Монтира се обзавеждане.

След като приключи всичко, готовата къща се показва на собственика.

#### **Итеративен модел**

Фаза 1 – груб строеж. Прави се проект (от строителя заедно със собственика) как ще изглежда къщата в груб строеж. Строителят прави план как ще се осъществи строежът. След като работата приключи, се показва на собственика. Ако той го одобри, се преминава на следващата фаза.

Фаза 2 – изграждане на баня и тоалетна. Прави се план как ще изглеждат банята и тоалетната, и как ще се направят. След като се направят, се показват на собственика. Ако всичко е наред, се преминава към следващата фаза.

Фаза 3 – изграждане на кухня. Прави се план ...

Фаза 4 – изграждане на хол. Прави се план ...

Фаза 5 – изграждане на спалня. Прави се план ...

### **Сравнение между водопаден и итеративен модел**

При водопадния подход работата следва предварително определен план и последователност. При итеративния подход междинните резултати се обсъждат и използват за уточняване на следващите дейности.

**Примерът** с къщата съчетава постепенно добавяне на нови части — инкрементално изграждане — с повторно планиране и обсъждане на резултатите. Самото добавяне на баня, кухня или друга част не показва подобряване на вече създаденото. Строителните дейности имат зависимости, които трябва да се спазват независимо от избора на подход. При софтуерен продукт междинният резултат може да бъде работеща версия с ограничени функции. Например училищен сайт първоначално представя новини, а в следващ цикъл към него се добавя календар на събитията.

**Извод:** Итеративният подход позволява по-ранна обратна връзка и постепенно уточняване на решението. Той изисква редовно планиране, проверка и съгласуване на промените.

## 5. Гъвкави подходи - Agile

**Agile** е общо название на гъвкави подходи, които насърчават сътрудничеството, честата обратна връзка и адаптирането към промени. Работещите резултати се предоставят постепенно, а следващата работа се уточнява според потребностите и постигнатото. Гъвкавостта не означава работа без план — планът се актуализира при необходимост.

## 6. Scrum модел

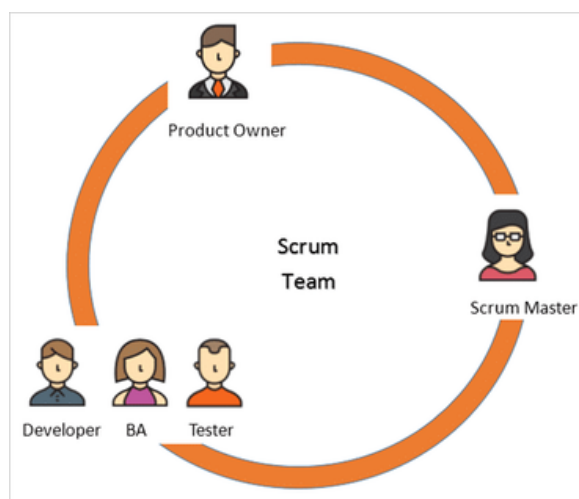
### а) Същност на Scrum

**Scrum** е рамка за организация на екипната работа, при която продуктът се развива чрез кратки цикли, наречени **спринтове**.

Спринтът има определена цел и постоянна продължителност **до един месец**, обичайно от 1 до 4 седмици. След завършването му започва следващият спринт.

Във всеки спринт екипът създава проверено, използваемо подобрене или допълнение към продукта. То се нарича **инкремент**. Това може да бъде нова функционалност, която работи съвместно с вече създадените части.

**Пример:** При разработване на училищен сайт целта на един спринт е посетителите да могат да разглеждат предстоящите събития. Резултатът е работещ календар, добавен към сайта.



### Scrum екип и основни роли

Диаграмата показва итеративния характер на Scrum и мястото на основните роли в процеса на разработване.

**BA** означава бизнес анализатор. Програмистът, бизнес анализаторът и тестерът са примери за специалисти, които могат да участват в групата **Developers**.

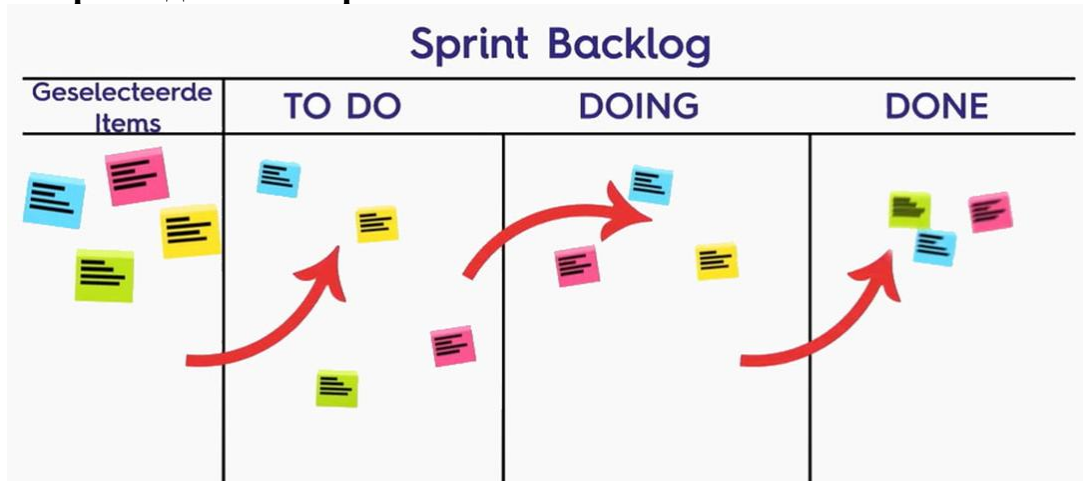
### б) Scrum екип

Scrum екипът включва:

- **Product Owner** — отговорник за продукта, който определя приоритетите според потребностите и стойността на резултата.
- **Scrum Master** — подпомага прилагането на Scrum, съвместната работа и отстраняването на пречки.
- **Developers** — специалистите, които създават продукта; сред тях могат да бъдат програмисти, дизайнери и тестери.

Екипът сам организира изпълнението на работата.

## в) Табло за проследяване на работата



Примерно табло за работата в спринта

**Sprint Backlog** е планът за работа в текущия спринт. Той включва целта на спринта, избраната работа и начина за нейното изпълнение. Показаното табло помага да се проследява състоянието на задачите.

В лявата колона са избраните задачи. **TO DO** — „За изпълнение“, **DOING** — „В процес“, а **DONE** — „Завършено“.

Всяка задача се представя с карта, която се премества според състоянието ѝ. Така екипът вижда напредъка и затрудненията.

## 7. KANBAN

Kanban е подход за организиране и подобряване на потока на работа. Задачите се представят чрез карти на табло с колони, например „За изпълнение“, „В процес“ и „Завършено“. Ограничава се броят на започнатите, но незавършени задачи, за да се избегне претоварване.

**Пример:** При поддръжка на училищен сайт екипът допуска най-много три задачи в колоната „В процес“. След приключването на една задача може да започне следваща. За разлика от Scrum, Kanban не изисква фиксирани спринтове. Таблото помага да се проследяват напредъкът и затрудненията.