

# Жизнен цикъл на софтуера и мястото на документацията в него

Николай Кънчев  
Валентин Атанасов

# Съдържание

Мотивация

История

Модел и методологии за разработка

Фази от жизнения цикъл на документацията

Гъвкава документация

Видове документация

Версия на документите

# Мотивация

Защо са нужни процеси при разработката на софтуер?

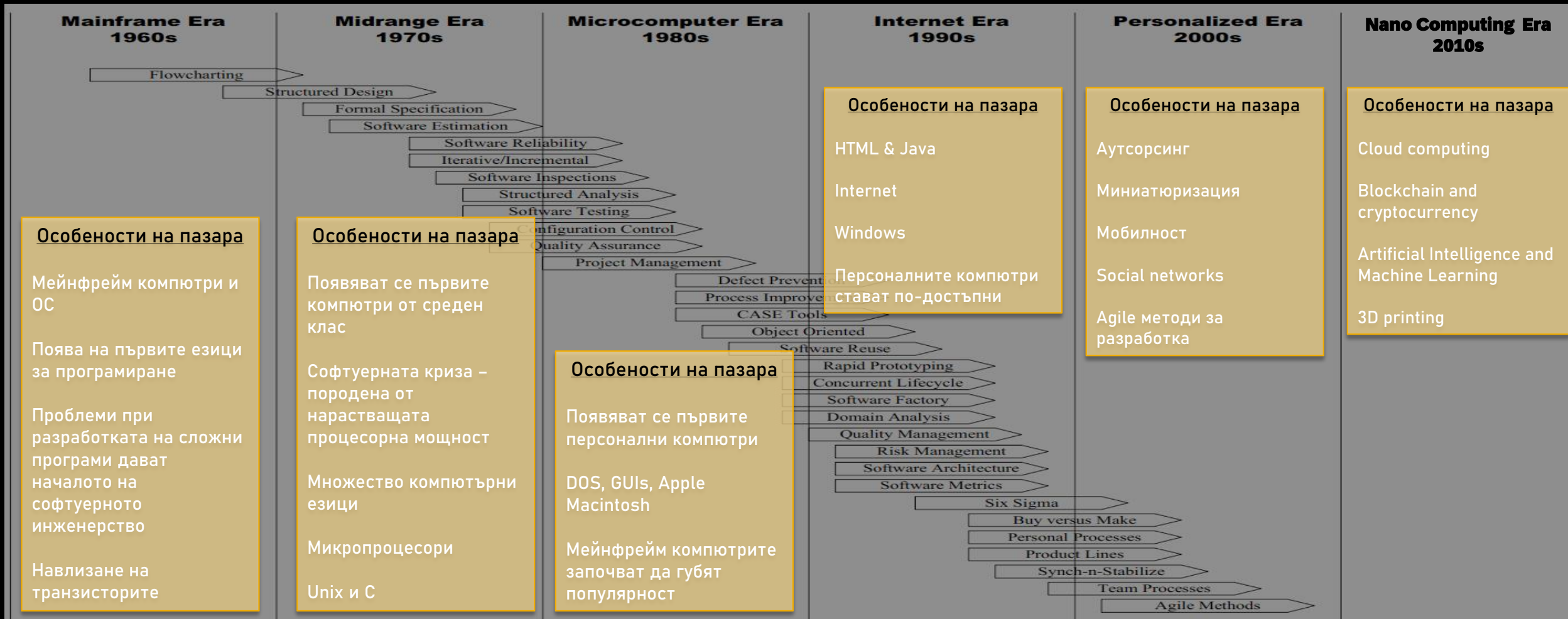
Софтуерът е сложен продукт и не може да се разработи без предварителен план и методология.

Процесите ускоряват разработката

Софтуерът трябва да има документация и тя също е необходимо да бъде част от процеса

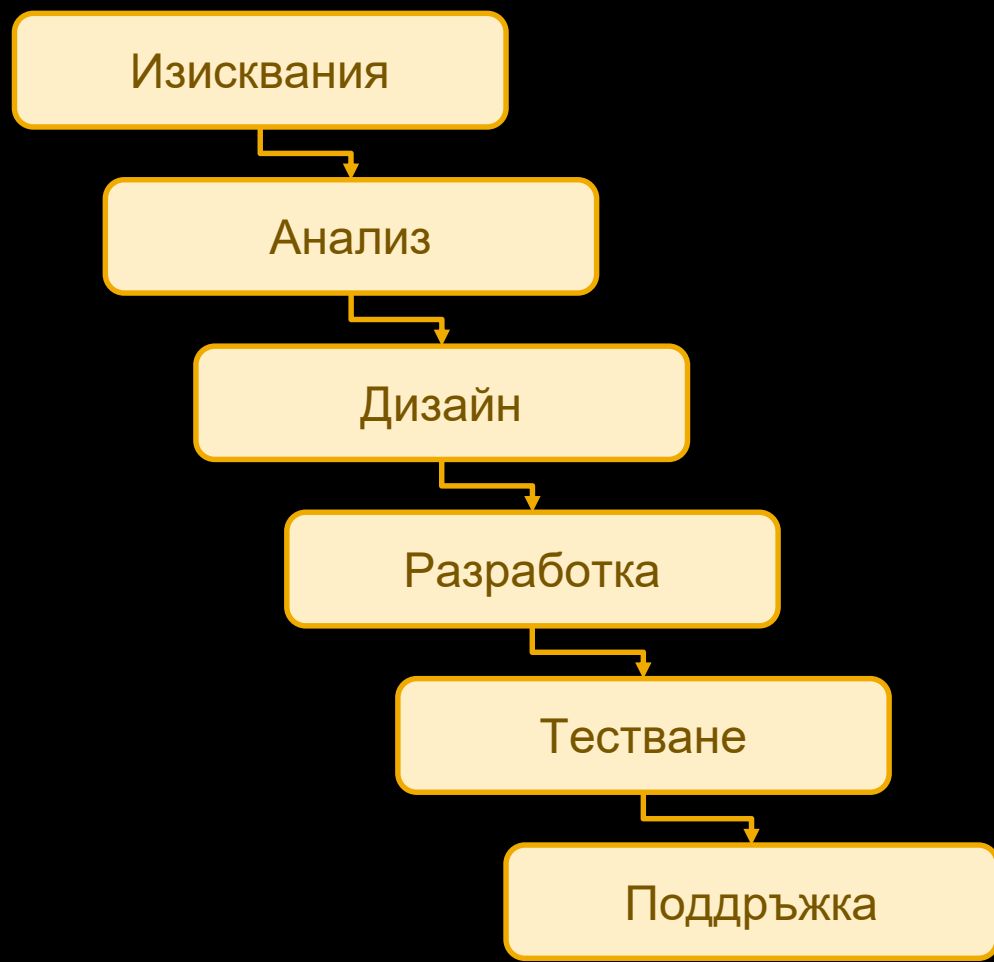
Как се свързват жизнените цикли на софтуера и прилежащата документация?

# История



# Методологии за разработка

## Waterfall



**Waterfall** – Един от най-ранните модели за разработка (1970 г.)

Ясно разграничени и отделени фази

Ефективен само ако 100% от всички изисквания са ясни от самото начало

Липсва „обратна връзка“

Прекалено широко дефинирани цели – разработка на цяла софтуерна система наведнъж

Исторически това е причината на множество големи проекти да излязат извън бюджета си

Практически неприложим в чистата си форма за по-сложни проекти

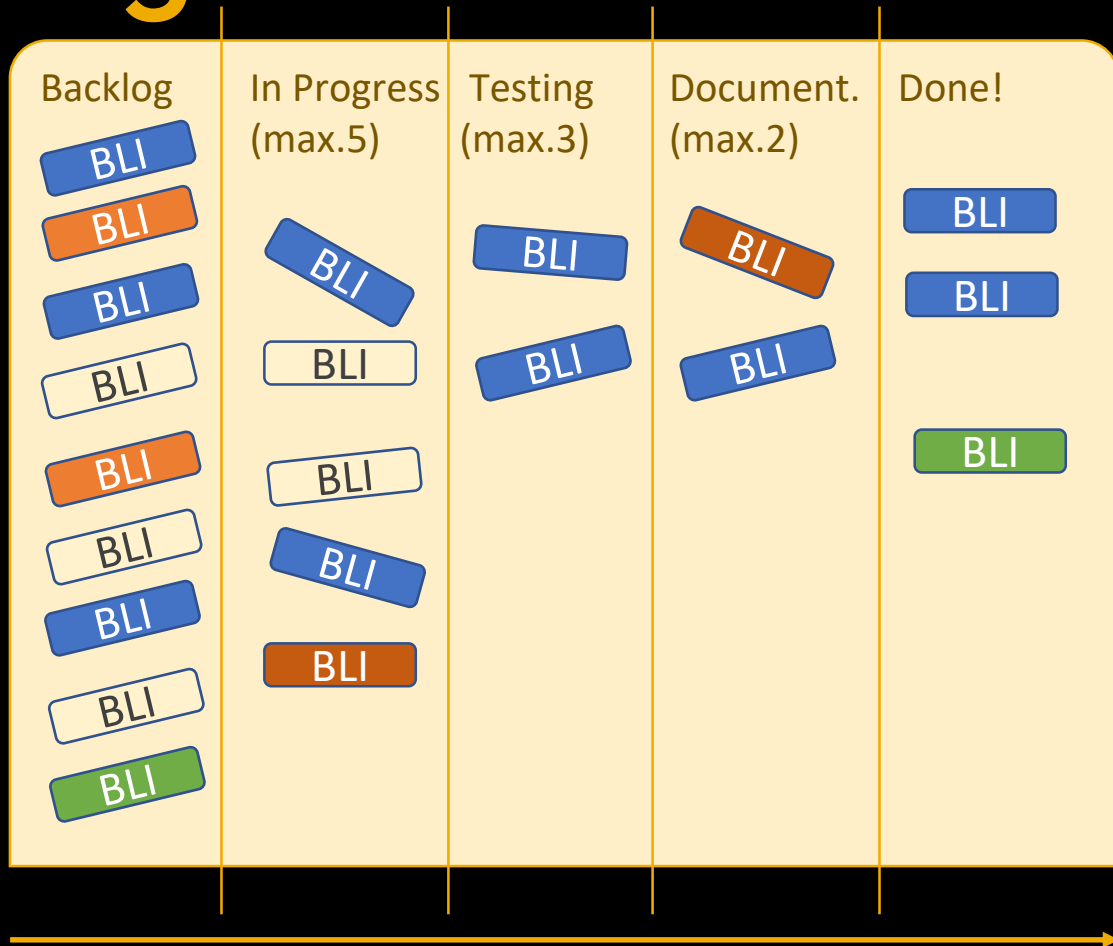
# Методологии за разработка Agile

**Lean Software Development –**  
Елиминиране на излишното

Разработен от Toyota за целите на  
производството в автомобилната  
индустрия

Адаптиран за софтуерната индустрия  
Елиминиране на всичко излишно  
Доставяне на продукта бързо  
Упълномощаване на екипа

# Agile



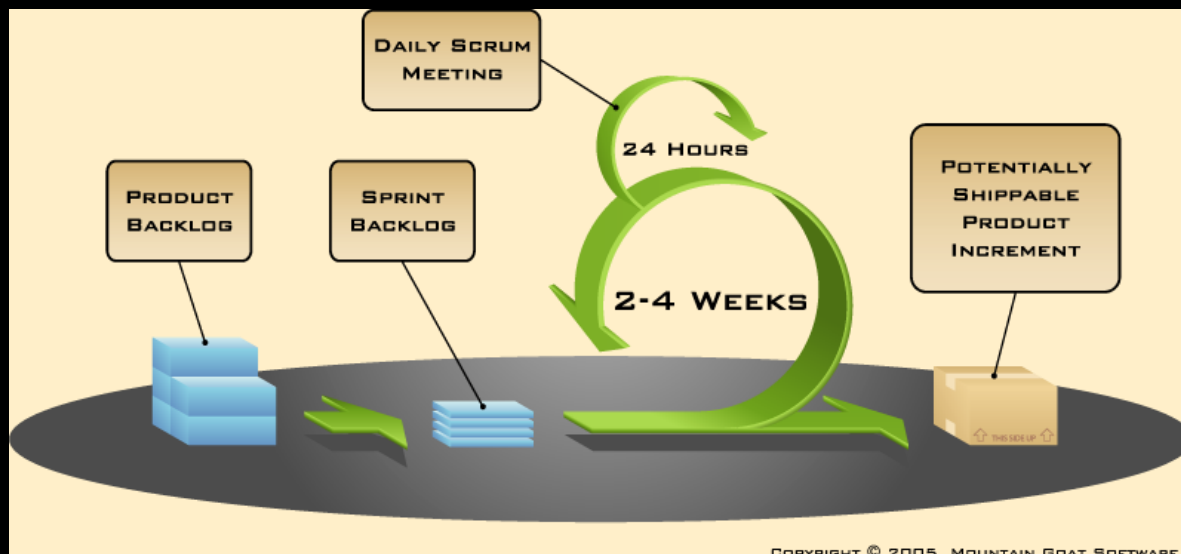
# Kanban – Метод за управление и онагледяване на дейностите

## Ясно визуализиране на работния процес

# Ограничение на задачите в процес на изпълнение

## Подобрено управление и оценка на работните процеси

# Методологии за разработка Agile



**Scrum** – Въвежда итеративна и инкрементална рамка за разработката на даден продукт

Софтуерът за разработка е разделен на малки компоненти, подредени по приоритет

Много кратък цикъл на разработка (2-4 седмици)

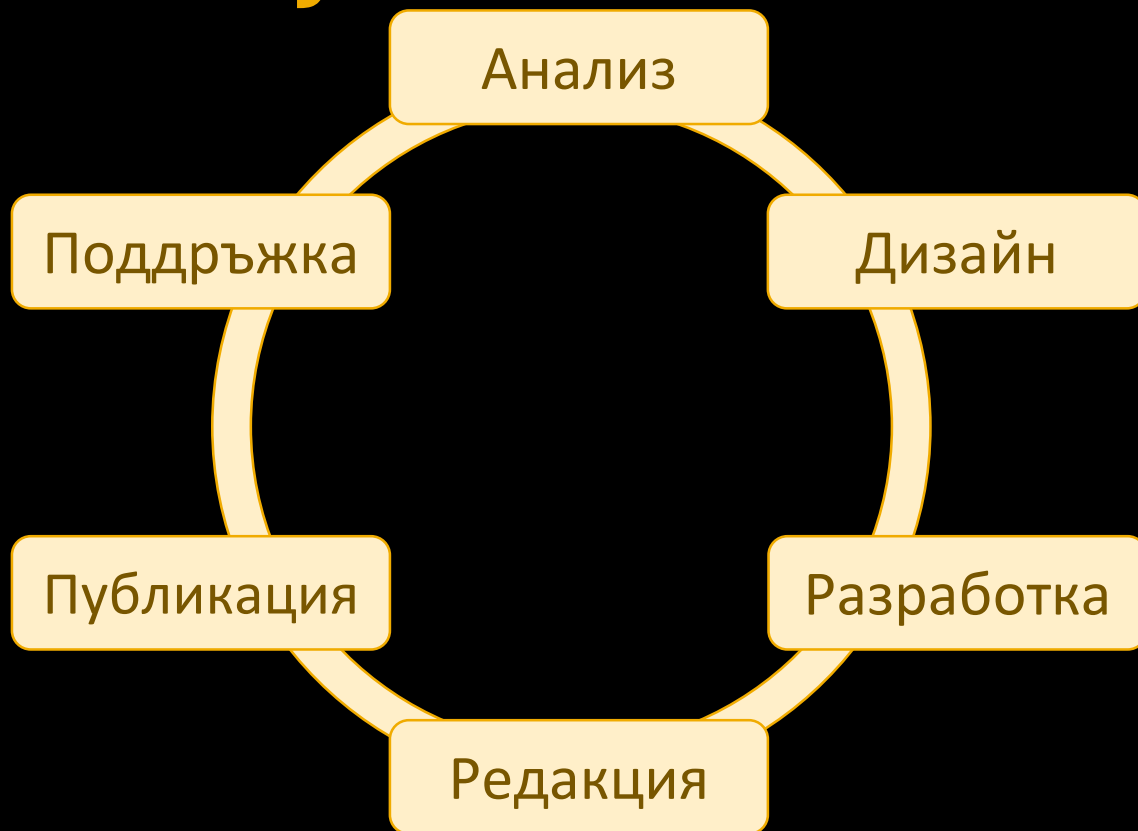
Самоорганизиране и регулиране в рамките на SCRUM екипите

Важни роли – SCRUM Master, Product Owner, Developer

Важни срещи – Daily Stand-up, Planning, Review, Retrospective



# Какъв процес следва документацията?



## Основни Фази

**Анализ** – Събира се полезна информация за проекта, за крайните потребители на документа, сложността и т.н.

**Дизайн** – Определят се основни параметри като крайни формати и видове документи

**Разработка на съдържание** – Създаване/писане на документите

**Редакция** – Редактиране за изглаждане на граматически, смислови и правописни грешки

**Публикация** – Публикуване на документите в крайния им формат – дигитален или печатен

**Поддръжка** – Актуализиране на документацията спрямо въвеждания и корекции. Създаване на различни версии.

# Документация в Agile Development

Документацията е важен елемент от всеки софтуерен продукт и нейното създаване и поддръжка е неизменно. Както софтуерния, така и документационния процес могат да са Agile.

Признаци, по които да познаем един Agile документ:

Ползата от документа трябва да надхвърля инвестицията за създаването му.

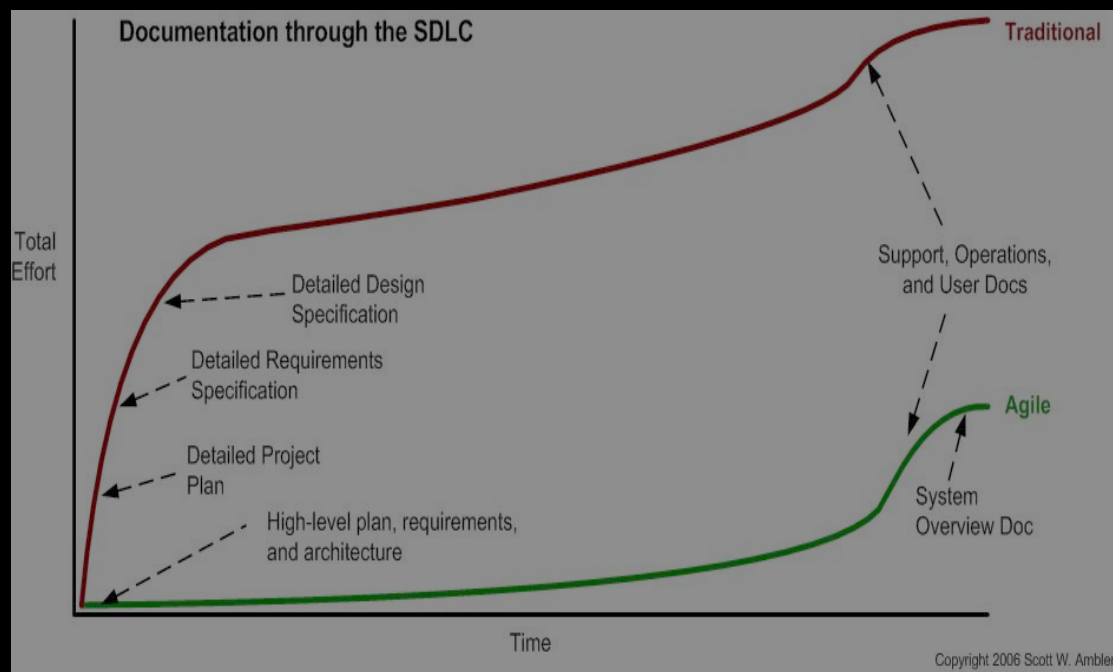
Заинтересованите страни знаят TCO (Total Cost of Ownership) на документа.

Документите съдържат цялата нужна информация и нищо извън нея.

Документите имат ясно определена цел.

Документите са предназначени за определени групи от потребители – администратори, поддръжка, маркетинг.

# Кога и колко да документираме?



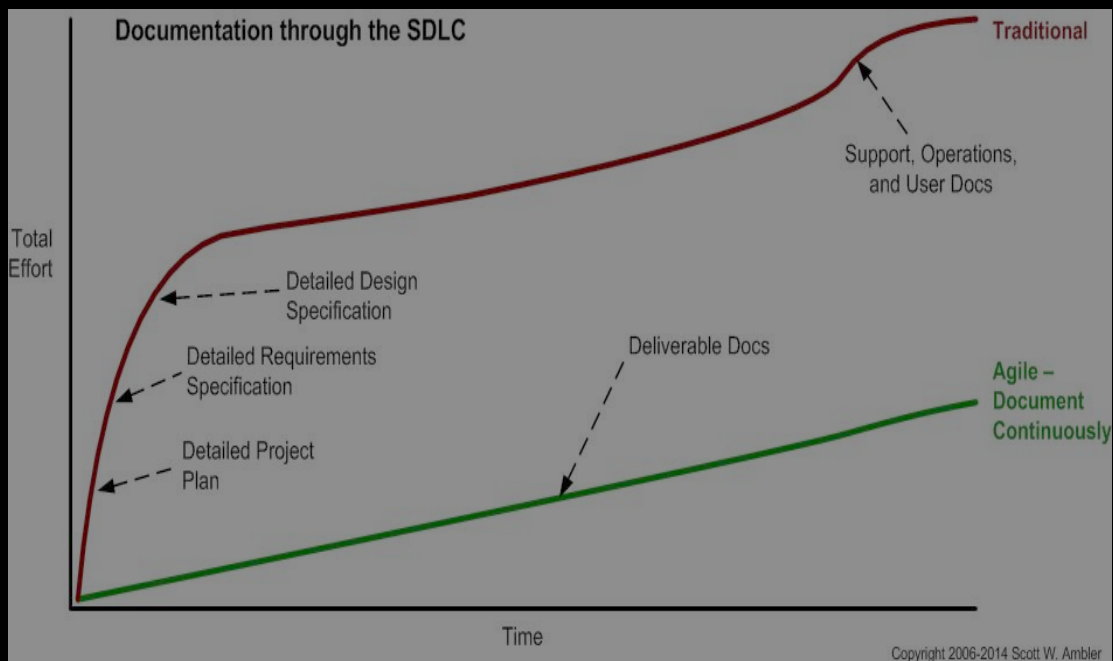
Документиране възможно най-късно

В началото на проекта фокусът е върху общи планове, а не върху детайли.

Подробностите се документират, когато са нужни, а не предварително.

Наръчници и подобни документи се пишат при стабилна система.

# Кога и колко да документираме?



## Постоянно документирање

Ако е нужно често да се доставят крайни версии, документацията трябва да е готова навреме.

Документират се само стабилизирани функции.

Итерациите за документация са зависими от итерациите за разработка.

Дълги периоди (2-4 седмици) – документират се функциите на актуалната версия.

Кратки периоди (до 2 седмици) – документират се функциите на предишната версия.

# Видове документи

## Разработка

- Подробен дизайн документ
- Коментари в кода
- API

## Тестова фаза

- Тестови планове (White-Box & Black-Box)
- Test Cases

## Въвеждане в експлоатация

- Наръчници за инсталация
- Материали за обучение
- Наръчници за администратори
- ...

## Поддръжка

- Информация за ъпдейти и корекции
- Troubleshooting информация

## Фаза на подобрене

- Lessons Learned
- Feedback

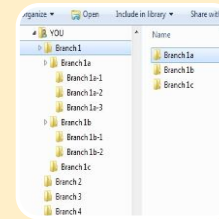


# Версии на документите

Защо?



Как?



## Файлова Структура

- Без нужда от допълнителен софтуер
- Проблеми при работа над един документ
- Използват се стандартните програми за всеки вид документ
- Трудна автоматизация



## Wiki

- Collaborative Authoring
- Без нужда от допълнителен софтуер
- Трудна координация при голям брой документи/автори



## CMS

- Механизми за Lock/Release
- Подобрено мащабиране

**Благодарим за вниманието!**

# Определяне на целева група. Разработка на персона.

Валентин Атанасов  
Николай Кънчев



# За кого пишем документация?

Софтуерна документация за програмисти, администратори, крайни клиенти, поддръжка, маркетинг специалисти

От целевата група зависи как ще изглежда и какво ще съдържа един документ, графика или видео

# Задача 1

Отворете сайта на SAP Cloud Platform:

<http://help.sap.com/viewer/product/CP> и определете коя част от информацията за кого е предназначена.

Пример: **Development > API Documentation** е предназначена за програмисти

# Какво е персона?

Персона е съвкупност от потребители, които имат общ поведенчески модел при използване на един софтуерен продукт или услуга, взимане на решения при онлайн пазаруване, предпочитания за клиентско обслужване и други.

# Как се определя една персона?

Провеждат се интервюта с потребители

Оформят се общи предпочитания и се групират

Информацията се анализира

Изграждат се профили на персони

За повече информация: <https://uxmag.com/articles/personas-the-foundation-of-a-great-user-experience>

# Каква информация включва една персона?

**Име:** създава се по-близка връзка с най-важните потребители

**Длъжност:** какъв тип потребител

**Образование, умения:** изгражда се цялостен профил и лична връзка

**Отговорности в работата:** как протича един работен ден

**Стил на работа, отношение към технологиите:** кои функционалности и функции са важни за потребителя

**Болни места:** какво притеснява потребителя в работата

**Основни цели:** области, в които потребителят има нужда от помощ за подобряване на уменията

# Примери за персона



**Име:** Мария

**Длъжност:** Програмист

**Образование, умения:** ФМИ,  
Компютърни науки, Java

**Отговорности в работата:** създава  
код за облачни приложения,  
участва в код ревюта, има крайни  
срокове на всеки 2 седмици

**Инструменти:** Eclipse, Git, Jenkins,  
Wiki

# Примери за персона



**Болни места:** трудни за работа  
инструменти, сложен продукт,  
зависим от много компоненти

**Основни цели:** писане на  
качествен код, разработване на  
добър продукт, който отговаря  
на изискванията на клиентите,  
признание, повишение

# Задача 2

Дефинирайте 4 различни персони и опишете подробно една от тях. Нарисувайте я.



# Задача 3

За вече определената персона от задача 2, дефинирайте задача, която тази персона да има нужда да изпълни и опишете процедурата за постигането ѝ в документ.

Пример за задача

В сайта на SAP Cloud Platform:

<http://help.sap.com/viewer/product/CP> опишете как един програмист може най-бързо да си направи trial account за SAP Cloud Platform.

**Благодарим за вниманието!**