

# Софтуерна документация

# Съдържание

- Техническо писане
- Техническо писане: процесът
- Добри практики
- Изисквания
- Дизайн/Архитектура
- Документация на код
- Документация за крайния потребител

# Техническо писане

- Защо техническото писане е важно (дори за разработчици)
  1. Маркетинг: Продукта трябва да се продава
  2. Изкарва информацията от главата на някой (хората спират да питат, защото го има написано)
  3. Спомага екипа да е „на една вълна“
  4. Намалява разговорите със support; Прави потребителите по-щастливи

# Техническо писане

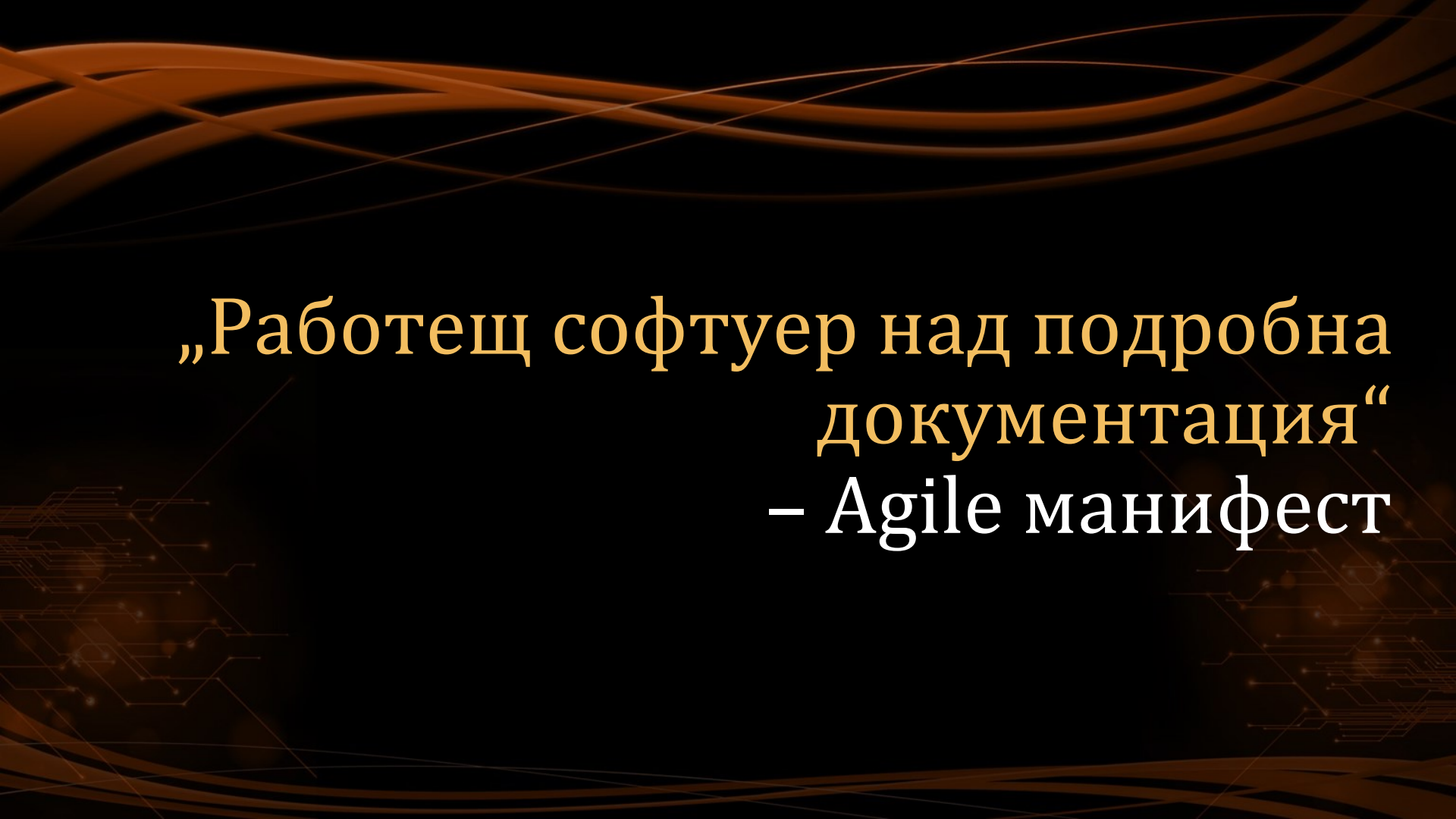
- Защо разработчиците са подходящи за писане на документацията
  - Разбират продукта
  - Интелигентни са
  - Имат добри умения по писане
  - Владееят английски на добро ниво

The background is a dark, textured surface. It features several thick, flowing, wavy lines in a warm orange-brown color that sweep across the top and bottom of the frame. In the lower-left and lower-right corners, there are intricate, glowing patterns resembling electronic circuit boards or data networks, with small points of light and thin lines connecting them.

..., но същевременно...

# Техническо писане

- Разработчиците не са подходящи да пишат документацията
  - Прекалено близо до продукта (не могат да го видят от аспекта на някой, за когото продукта е напълно нов)
  - Пишат от перспективата на разработчик, а не потребител
  - Трудно е да се игнорират предубежденията



„Работещ софтуер над подробна  
документация“  
– Agile манифест

# Техническо писане

- Съществена, стойностна и навременна документация
  - Съществена: документ с точно подбрана детайлност
  - Стойностна: ползите от документацията трябва да оправдават цената за писането и поддържането ѝ
  - Навременна: документацията трябва да е завършена точно навреме, когато е нужно





Техническо писане: Помага на потребителя да  
решава проблеми с технологии и технически  
въпроси

# Техническо писане: Причина

- Изкарва информацията от главата на създателите (разработчиците)...
- ...дава информацията на потребителите във вид, в който е използвана за тях

# Техническо писане: Процесът

1. Планиране
2. Проучване
3. Писане
4. Ревю и редакция
5. Пускане в ход

# Техническо писане: Процесът

- Планиране
- Причини да съществува документът
  - Да се убеди някой да мисли по определен начин
  - Да се спомогне изпълнението на дадена задача
  - Да се помогне нещо да бъде разбрано
  - Да се промени отношението към документираната система

# Техническо писане: Процесът

- Видове софтуерна документация
  - Изисквания: Идентифицира какво ще се създава; Служи да се провери дали ще са удовлетворени очакванията на заинтересованите страни
  - Дизайн/Архитектура: Дефинира как ще е конструирана системата
  - На код/техническа: Позволяват завършването на задачи и разбирането им
  - Планиране на тестове/Тест случаи: Дефинира подхода за тестване
  - За крайния потребител: Позволява изпълнението на дадени задачи, support, отстраняване на неизправности

# Техническо писане: Процесът

- Планиране
- Анализирание на аудиторията
  - Кой ще чете документацията?
  - Какво вече знаят?
  - Защо ще я четат?
  - В каква среда ще я четат?
  - Какво ТРЯБВА да знаят?
  - Какъв тон е подходящ?

# Техническо писане: Процесът

- Стилът на писане се определя спрямо аудиторията
- Може да се наложи да бъдат написани няколко документи за различните аудитории

# Техническо писане: Процесът

- Проучване

- Интервю с експерти в дадена област (Subject Matter Experts – SMEs)
- Ревю на вече съществуваща документация
- Използване на софтуера



# Техническо писане: Процесът

- Подготовка за интервю със SMEs
  - Подгответе въпросите за интервюто
    - Кой, какво, кога, къде, защо, как
    - Всеки въпрос трябва да се фокусира само върху едно нещо
  - Предвидете възможни посоки, в които ще поеме интервюто и помислете за вашата реакция

# Техническо писане: Процесът

- Насрочване на интервю със SME
  - Решете дали интервюто ще се записва
    - Поискайте разрешение
  - Предложете 2-3 вариаци за дата и час (продължителност не повече от 60 минути)
  - Запазете конферентна зала, за да ограничите максимално разсейването

# Техническо писане: Процесът

- Провеждане на интервюто със SME
  - Започнете навреме
  - Потвърдете нивото на компетентност от двете страни
  - Бъдете уверени и без напрежение
  - Наблюдавайте езика на тялото
  - Не се разсейвайте
  - Попитайте за допълнително мнение преди да смените темите
  - Благодарете за отделеното време
  - При нужда насрочете нова среща

# Техническо писане: Процесът

- Писане
  1. Организирайте съдържанието и идеите си (хронологически, от сложно към комплексно, от специфично към абстрактно)
  2. Напишете първата чернова
  3. Ревю и оценяване на черновата
    - Въпроси, които да си зададете:
      - Документа изпълнява ли предназначението си?

# Техническо писане: Процесът

- Нещо липсва ли?
- Може ли нещо да се премахне?
- Какви въпроси би имал читателя? Отговорете им.
- Лесно ли е да се разбере написаното?
- Също проверете:
  - Граматиката
  - Избора на думи
  - Правописа

# Техническо писане: Процесът

- Ревю/Редакция
- Причини
  - Реорганизация на съдържанието
  - Силова редакция
  - Редакция за граматика и пунктуация
  - Редакция за включване резултатите от тестове

# Техническо писане: Процесът

- Съвети за редакция и ревю
  - Намерете някой друг, който да го направи
  - Четете на глас
  - Принтирайте документацията
  - Премахвайте набухвателните изречения, думи, изрази

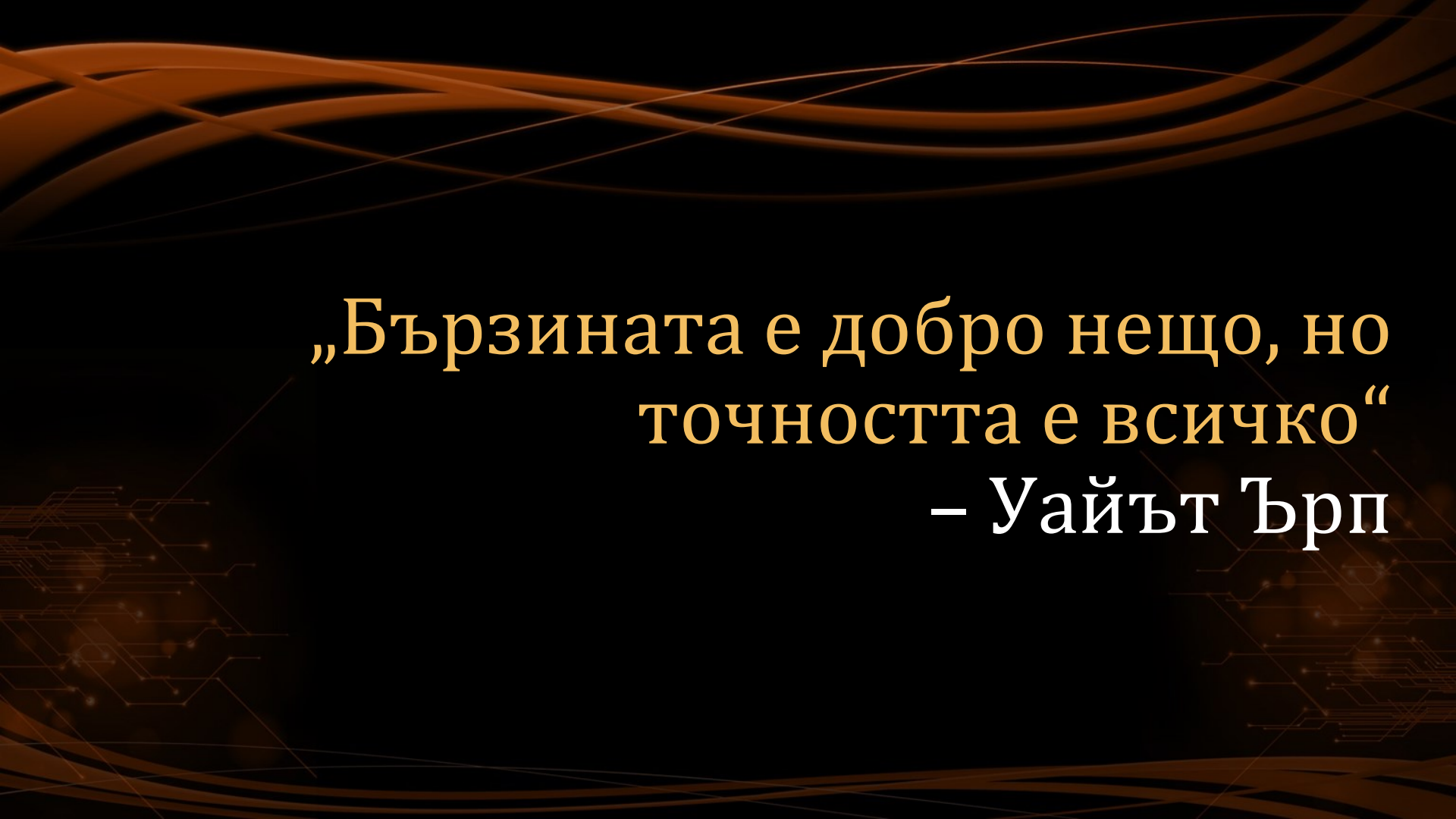
# Техническо писане: Процесът

- Тестване на използваемост: Тестване на документ ред по ред, за да се постигне увереност, че написаното има смисъл и инструкциите работят, както са описани



# Техническо писане: Добри практики

- Точност
- Яснота
- Сбитост
- Тон
- Време
- Граматика, правопис и терминология



„Бързината е добро нещо, но  
точността е всичко“  
– Уайът Ърп

# Техническо писане: Добри практики

- Точност
- Аспекти
  - Точност на документа
  - Стилистична точност
  - Техническа точност

# Техническо писане: Добри практики

- Точност на документа
  - Съдържа подходящи теми, с подходящото ниво на детайлност
  - Фокусира се върху проблем или решение
  - Решава теоретичен или практически проблем

# Техническо писане: Добри практики

- Стилистическа точност
  - Внимателно използване на езика
  - Думите са използвани на място
  - Структурата на параграфите и изреченията описват и анализират темите ефективно

# Техническо писане: Добри практики

- Съвети за подобряване на точността
  - Консултирайте се със SMEs
  - Проведете тестове за използваемостта (usability testing)



„Яснотата позволява фокус“  
– Томас Леонард

# Техническо писане: Добри практики

- Яснота: Леснотата, с която се разбира
- Аспекти
  - Структурна яснота
  - Стилистична яснота
  - Контекстова яснота



# Техническо писане: Добри практики

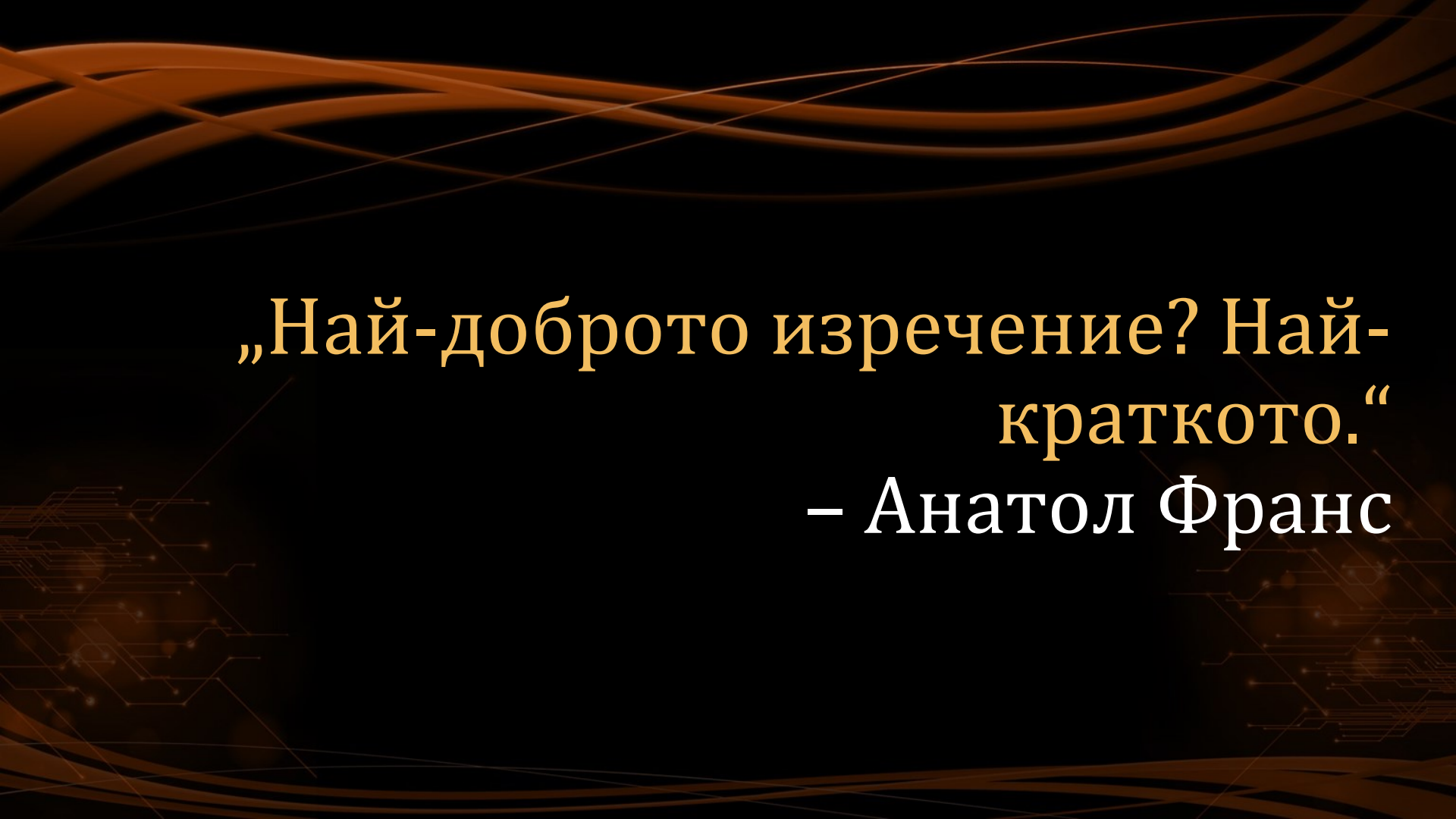
- Съвети за подобряване структурната яснота
  - Абстракции
  - Въведения
  - Съдържание
  - Графи
  - Таблици
  - Описателни заглавия

# Техническо писане: Добри практики

- Съвети за подобряване на стилистичната яснота
  - Бъдете специфични
  - Избягвайте нееднозначност
  - Не бъдете прекалено лаконични
  - Избягвайте страдателен залог

# Техническо писане: Добри практики

- Съвети за подобряване на контекстовата яснота
  - Посочвайте причината за нещата
  - Какво предхожда документа?
  - Как този документ е обвързан с другите?



„Най-доброто изречение? Най-  
краткото.“  
– Анатол Франс

# Техническо писане: Добри практики

- Съвети за подобряване на сбитостта
  - Имайте ясен фокус от самото начало
  - Елиминирайте материали и думи, ненужни за поддържане на твърдението си
  - Използвайте визуализация
  - Кратки изречения от 15-20 думи (принт), 10-15 думи (онлайн)

# Техническо писане: Добри практики

- Тон: Отношението на писателя към аудиторията
- Тонът трябва да се определя спрямо аудиторията
- Тонът трябва да се определя от този, който подписва платежния чек

# Техническо писане: Добри практики

- Време
- Най-често използвано време в техническото писане
  - Сегашно
  - Бъдеще

„Вашата граматика е отражение на вашия образ. Добро или лошо, направихте впечатление. И като всички впечатления, вие сте в пълен контрол.“

– Джефри Гитомер



# Техническо писане: Изисквания

- Документиране на изисквания
- Добрите изисквания са
  - Унитарни: Адресират само едно нещо
  - Пълни: Напълно зададени на едно място, като не липсва никаква информация
  - Консистентни: Не противоречат на някое друго изискване

# Техническо писане: Изисквания

- Самостоятелни: Не съдържат зависимост от друго изискване
- Проследими
- Текущи: Не са станали остарели или нерелевантни с времето
- Недвусмислени: Без жаргон, акроними, не са езотерични
- Могат да се тестват:
- Уточняват важност: Уточняват нивото на важност, определено от заинтересованите страни, време или бюджет

# Техническо писане: Изисквания

- Стиллове на писане при изисквания
  - User Story
  - Traditional (Text-Based)
  - Use cases

# Техническо писане: Изисквания

- Типове изисквания
  - Бизнес
  - Потребителски
  - Функционални
  - Нефункционални
  - Интерфейс

The background is a dark, textured surface. It features several thick, flowing, wavy lines in shades of orange and brown that sweep across the top and bottom of the frame. In the lower-left and lower-right corners, there are intricate, glowing patterns resembling circuit boards or digital networks, with small points of light and thin lines connecting them.

Без значение от стила и типа, целта е да се  
комуникира ясно и ефективно.

# Техническо писане: Изисквания

- Съвети за писане на ясни изисквания
  - Дефинирайте термините в речник
  - Използвайте деятелен залог
  - Не използвайте неясни изрази (efficient, high-performance, several)
  - Не спекулирайте (usually, often, typically)
  - Не изразявайте възможности (could, ought to, probably)

# Техническо писане: Изисквания

- User Stories
  - Популярни в Agile разработката на софтуер
- User Story: Едно или повече изречения във всекидневния или бизнес езика, което обхваща какво трябва да направи потребителя

# Техническо писане: Изисквания

- User Stories формат
  - As a <type of user>, I want <goal> so that <reason>
- Пример
  - As a <patient>, I want <to receive an SMS message when my prescription is ready to pick up> so that <I can avoid unnecessary waiting at the pharmacy>



# Техническо писане: Изисквания

- User Stories ползи
  - Кратки, разбираеми от потребителите и разработчиците, не се нуждаят от поддръжка
- User Stories недостатъци
  - Може да са неясни, отворени за интерпретация, незавършени

# Техническо писане: Изисквания

- Traditional (Text-Based): Едно или повече изречения, които определят функционалността на високо ниво
- Формат: <Subject doing the action> <auxiliary verb> <capability or functionality to be provided> <criterion that limits or further explain requirement(optional)>
- Пример: <The Company> <shall> <develop an SMS notification system> enabling patients to <receive alerts when their prescriptions are available to pick up>

# Техническо писане: Изисквания

- Ползи: Може да се използва, за да се обхванат пълни изисквания рано в проекта
- Недостатъци: Може да липсват достатъчно детайли за имплементация

# Техническо писане: Изисквания

- Use Cases: Списък с действия или събитийни стъпки, обикновено дефиниращи интеракции между лице и система, за да се постигне цел

# Техническо писане: Изисквания

## ■ Формат

**Use Case Number:** Уникален идентификатор

**Title:** Наименува целта или основния деятел

**Description:** Кратко описание и причина за use case-a

**Actors:** Всички извършители, които имат общо с use case-a

**Scope:** Име на система или подсистема, дефинирана от use case-a

**Priority:** Колко важно е изискването?

**Assumptions:** Всички условия, за които се предполага, че са верни

**Preconditions:** Състояние, в което системата трябва да се намира, за да продължи use case-ът

**Post conditions:** Промени в средата, в резултат на use case-a

**Trigger:** Какво ще задейства use case-a

**Main Success Scenario:** Описание на use case-a стъпка по стъпка

**Extensions:** Алтернативни начини за постигане на целта

# Техническо писане: Изисквания

**Use Case Number: UC-2.1.5**

**Title: Receive and acknowledge notification to opt in to the program**

**Description: The patient receives an SMS on their cell phone, indicating they have been successfully enrolled. Upon receipt, the patient replies with a message indicating they accept.**

**Actors: Patient**

**Scope: X patient SMS**

**Priority: Essential**

**Assumptions: Patient is able to receive SMS messages**

**Preconditions: A valid phone number for the patient is stored in the system**

**Post conditions: Patient has fully enrolled in the program with a double opt-in**

**Trigger: Pharmacist enrolls the patient in the notification program**

**Main Success Scenario: 1. When the pharmacist enrolls the patient in the program, the system sends an opt-in SMS message to the patient's cell phone 2. The patient opens the notification on their phone 3. The patient acknowledges receipt of the message and opts in by responding with "Accept"**

**Alternative Scenario: If the SMS message is not received after one attempt on the patient's device (due to system outage, wrong phone number, etc.), the system will: 1. Log a failure message 2.**

**Discontinue delivery attempts for the current message**

# Техническо писане: Изисквания

- Типовете изисквания ще варираят според проекта

# Техническо писане: Изисквания

- Бизнес изисквания
  - Функционалност на системата от високо ниво
  - Описват какво ще се постигне за бизнеса (не как)
  - Traditional (Text-Based) стил



# Техническо писане: Изисквания

- Потребителски изисквания/изисквания на заинтересованите страни
  - Описват какво ще бъде постигнато за потребителите/заинтересованите страни
  - Traditional (Text-Based), User stories стилове

# Техническо писане: Изисквания

- Функционални изисквания
  - Дефинират функцията на системата или нейните компоненти
    - Поведения, вход, изход
  - User Story, Use Cases

# Техническо писане: Изисквания

- Нефункционални изисквания
  - Критерий, който дефинира операционната среда, в която съществуват функционалните изисквания
    - Качесво, зависимости
  - User Stories, Traditional (Text-Based), Use Cases

# Техническо писане: Изисквания

- Интерфейс
  - Описва как системата ще взаимодейства с друга система
    - Хардуер, софтуер, комуникация и потребителски интерфейс
  - Traditional (Text-Based), Use Cases

# Техническо писане: Изисквания

- Software Requirement Specification (SRS)
  - Описание на софтуерна система, която да се разработи
  - Подпомага ревюта
  - Снабдява софтуерните дизайнери с референция
  - Описва полето на работа

# Техническо писане: Изисквания

SRS структура:

- Table of Contents
- Introduction
  - Причина
  - Конвенции в документа
  - Аудитория
  - Референции

# Техническо писане: Изисквания

- Revision History
- Overall Description
  - Перспектива на продукта
  - Функции на продукта
  - User класове и характеристики
  - Ограничения за дизайна и имплементацията
  - Потребителска документация
  - Предположения и зависимости

# Техническо писане: Изисквания

- External Interface Requirements
  - Потребителски интерфейси
  - Хардуер интерфейси
  - Софтуер интерфейси
  - Комуникационни интерфейси
- Functional Requirements
  - System feature 1
  - System feature 2
  - ...



# Техническо писане: Изисквания

- Non-Functional Requirements
  - Производителност
  - Сигурност
  - Качество на софтуера
  - Бизнес правила
- Other Requirements
- Glossary

# Техническо писане: Дизайн/Архитектура

- Software Design Document (SDD)
  - Разписано описание на софтуерен продукт, което дава на разработчиците обща представа за архитектурата на софтуерния проект

# Техническо писане: Дизайн/Архитектура

## SDD структура

- Introduction
- System Architecture
- User Interface
- Glossary/Apendix

# Техническо писане: Дизайн/Архитектура

- Introduction
  - Важни дефиниции, акроними, абривиатури
- Detailed System Design
  - Шаблони за дизайн и техники
  - Модули, класове, файлове и т.н.
  - Структури от данни
  - Алгоритми
  - Интерфейси

# Документация на код

- Коментари
  - Как да не ги използваме
    - За нещо очевидно
    - За обяснение на зле написан код
    - За да се изтрие код
  - Как да ги използваме
    - Коментари от високо ниво
    - По принцип полезни на ниво клас
    - TODO, HACK, UNDONE

# Документация на код

- Документация на API: Инструкции как ефективно да се използват API-та за хардуер или софтуер



И най-доброто API на света е безполезно  
без документация

# Документация на код

- Документация на API
- Аудитории
  - Начинаещи разработчици
  - Разработчици, които трябва да debug-ват
  - Бизнес лица или разработчици, които трябва да оценят API-то



# Документация на код

- Документация на API
- Какво да включва
  - Бърз преглед и концепции
  - Уроци / трейнинги
  - Инсталация / как да започнем /  
отстраняване на грешки
  - Информация за лиценз
  - Документация на пакет за разработка на  
софтуер (SDK)

# Документация на код

- Пример за добра API документация: TWILLO
- Популярни инструменти за генериране на документация на API:
  - [swagger.io](https://swagger.io)
  - [mashery.com](https://mashery.com)
  - [apiary.io](https://apiary.io)
  - [raml.org](https://raml.org)
  - ASP.NET Web API

# Документация на код

- README файлове: Файлове (обикновено .txt), които помагат на потребители, други разработчици да разберат как да направят нещо със софтуера
- Пишете README файлът преди кода

# Документация на код

- README задължително
  - Дата
  - Име на софтуера и номер на версията
  - Кратко описание на софтуера
  - Изисквания за инсталация и инструкции
  - Информация относно лиценза
  - Информация за контакти

# Документация на код

- README – ако е нужно
  - Списък с файловете
  - Инструкции за конфигурация
  - Известни бъгове
  - Отстраняване на грешки
  - Кредити
  - Новини, ъпдейти

# Документация за крайния потребител

- Три типа документация за крайния потребител
  - User Guide
  - Quick Reference Guide (също познат като Getting Started)
  - Release Notes

# Документация за крайния потребител

- User Guide: помага на потребителя да разбере и използва системата
  - Какво прави?
  - Каква е основната функционалност?
  - Защо да я използвам?
  - Как да я използвам?
  - Как се връзва системата с други системи, които използвам?

# Документация за крайния потребител

## User Guide структура

- Table of Contents
- Overview
- Getting Started
- Troubleshooting
- FAQs
- Tutorials
- Support
- Glossary
- Index



# Документация за крайния потребител

- Quick Reference Guide пример
  1. Направете си акаунт
  2. Влезте в системата
  3. Добавете нов пациент
  4. Създайте нова рецепта
  5. Запишете пациента за SMS известяване

# Документация за крайния потребител

- Quick Reference Guide полезни елементи
  - Диаграми
  - Стъпки
  - Графики, инфографики

# Документация за крайния потребител

- Release Note: документира разликите между две версии на един софтуер
- Коя е аудиторията?
  - Съществуващи потребители
  - Продажби и маркетинг

# Обобщение

- Техническо писане
- Техническо писане: процесът
- Добри практики
- Изисквания
- Дизайн/Архитектура
- Документация на код
- Документация за крайния потребител