

Ефективно документиране при производство на софтуер

„Един самолет излита, когато документацията му започне да тежи повече от него“

Изявление на един от дизайнерите на Боинг 747 Джъмбо Джет

(Боинг 747 тежи повече от 300 т., а документацията му – повече от 400 т.)

Документацията играе важна роля в софтуерното инженерство. Всеки етап от жизнения цикъл на софтуера изисква създаване и поддържане на разнообразни документи: изисквания към системите и софтуера; програмен код, удобен за четене и поддържане; тестови сценарии и резултати от тестове; ръководства за крайните потребители и администраторите; описания на архитектури и бази данни; модели на бизнес процеси; документи по управлението на waterfall или agile проекти и др.п.

Ако софтуерната организация е внедрила ISO система за управление или добра практика (напр. Система за управление на ИТ услуги по ISO 20000-1 или ITIL), от нея се очаква да създаде и поддържа документация и записи, както за процесите за създаване и доставка на ИТ услуги, така и за ИТ услугите и реализиращия ги софтуер. Тази документация трябва да отговаря на ред задължителни изисквания.

Курсът демонстрира организацията на създаване и поддържане на типични категории документи, както за вътрешните цели на софтуерната организация, така и за нуждите на клиентите и потребителите. Дава насоки за структурата и съдържанието на важни документи като спецификацията на изискванията към софтуера, програмния код, тестовите сценарии, потребителската документация.

Обсъждат се начини за мотивиране на екипа от разработчици да произвежда качествена и използвана техническа документация. Разглеждат се примери за документиране на софтуерни разработки при изпълнението на класически waterfall или гъвкави Agile проекти. Посочват се популярни платформи за управление на документацията при разработването на софтуер.

За кого е предназначен този курс?

- Софтуерни инженери и мениджъри;
- Разработчици на техническа документация;

Език на обучението

Български. Курсът може да бъде поръчан и на английски и немски език.

Продължителност и форма

Обучението продължава 2 дни, 16 учебни часа. Обучението е Zoom/ Webex и обхваща лекции, упражнения, изследване на бизнес казуси, уъркшоп.

Цена

508 лв. Цената е за един участник и е без включен ДДС.

Цената за вътрешнофирмено обучение се договаря.



Сертификат

Сертификат, издаден от Училище за бизнес компетенции.

За преподавателя



Огнян Дренски, MSc, е директор в Училище за бизнес компетенции. Работи повече от 30 години в ИТ като системен и приложен програмист, научен сътрудник по системно програмиране, тийм лидер в софтуерна къща, технически директор, ИТ директор в банка и голяма фармацевтична компания, изпълнителен директор на технологичен холдинг. Участвал е в разработването на софтуер за надеждни комуникации, медицински системи, строително проектиране, обучение, управление на предприятие. Има опит в експлоатацията на големи системи. Огнян е сертифициран проектен мениджър, ITIL, водещ одитор по системи за управление на ИТ услуги, информационна сигурност и непрекъснатост на бизнеса. Разработва учебно съдържание, преподава и консултира в областта на ИТ, реинженеринга на бизнес процесите, управлението на проекти и на бизнеса.

Акредитации

Качеството на учебните програми на Училището се гарантира с международни и национални акредитации.

Училище за бизнес компетенции е акредитирано от Националната Агенция за Професионално Образование и Обучение (НАПОО), притежава Лиценз за професионално образование и обучение по ключови компетенции и статут на Център за Професионално Образование с право да обучава и провежда изпити.

Училището е акредитирано от EBC*L International – The European Business Competences Licence и е първият в България обучителен и изпитен център за сертификата за бизнес компетенции EBC*L. Училището има статут на EBC*L Академия.



Програма на курса

Ден 1

08.30 – 09.00 Регистрация

09.00 – 10.30 Блок 1

За курса

- Администриране
- Представяне на участниците

Въведение

- Основни причини за документиране при производството на софтуер.
- Документи и записи. Софтуерна документация.
- Типични видове софтуерна документация. Документация ориентирана навътре към софтуерната организация и навън от нея.

Бизнес-казусът, върху който ще демонстрираме идеите на курса

- Проектът „Лаборатория за ДНК диагностика“ (DNA Lab).
- Документиране на разработката на софтуера за ДНК-лабораторията.
- Среда за демонстриране – Jira, Confluence, ARIS Express.

Документацията в Software Development Life Cycle (SDLC)

- SDLC – преглед на типични документи по SDLC.
- Документация при класически (waterfall) и гъвкави (Agile) проекти.
- Добри практики за документиране в Agile.

10.30 – 10.45 Почивка

10.45 – 12.15 Блок 2

Документи преди началото на един софтуерен проект

Упражнения 1 и 2

Изследване на документите **Мандат за стартиране на проекта DNA Lab** и **Бизнес казус на проекта DNA Lab**,

SDLC Етап 1 - Инженеринг на изискванията (Requirements engineering)

- Документиране на изискванията:
 - Stakeholders' Requirements Specification (StRS).
 - System's requirements Specification (SyRS).
 - Software Requirements Specifications (SRS).

Упражнение 2

Изследване на структурата на документите StRS, SyRS, SRS.

- Изразяване на изискванията на естествен език и чрез таблица „условие – действие“



Упражнение 3

Документиране на изискванията към софтуера (SRS).

12.15 – 13.15 Обедна почивка

13.15 – 14.45 Блок 3

SDLC Етап 1 – Документиране на изискванията в Agile

- Категории изисквания в Agile.
- Изразяване на изискванията чрез потребителски истории (**User Stories**).
- Йерархия на потребителските истории: Теми, Епоси, Потребителски истории (**Themes, Epics, User Stories**). Подробни потребителски истории.
- Критерии за приемане (**Acceptance Criteria & Definition of Done**).
- MoSCoW и изготвяне на приоритизиран списък на изискванията (**PRL**).
- Полезни темплейти за документиране на изискванията в Agile.
- Персони (**Personas**) и Пътешествие на потребителя (**User Journey**). **Use Cases**.
- Визуализация на изискванията (**Story Map**).
- Документация в Agile-таймбокса/спринта. Пример: Канбан в Jira.

Упражнение 4

Изследване на изискванията към софтуера на ДНК-лабораторията, представени в Agile и с инструментите на Jira и Confluence.

Упражнение 5

Практическа работа по документиране в Agile.

Упражнение 6

Практическа работа по визуализация на изискванията – структуриране и визуализация на изискванията от Упражнение 5 чрез Story Map.

14.45 – 15.00 Почивка

15.00 – 16.30 Блок 4

SDLC Етап 2 – Дизайн

- Документиране на софтуерната архитектура. **Architecture Description (AD), Architecture Description Languages (ADL), Architecture Frameworks (AF)**.

Упражнение 7

Практическа работа по документиране на елементи на архитектура – моделиране на бизнес процеси с Business Process Model and Notation.

- Документиране на бази данни.



Ден 2

09.00 – 10.30 Блок 5

SDLC Етап 3 – Внедряване. Документиране на кода и тестовете

- Документиране на кода. Добри практики и примери **Source Code Document (SCD)**.
- Преглед на кода. **Code Review Document**. Добри практики за преглед на кода и примери.

Упражнение 8

Изготвяне на чеклист за преглед на кода

10.30 – 10.45 Почивка

10.45 – 12.15 Блок 6

SDLC Етап 3 – Внедряване. Клиентска документация

- Правила при писането на клиентска документация. Примери.

12.15 – 13.15 Обедна почивка

13.15 – 14.45 Блок 7

Упражнение 8

Практическа работа върху документация за крайни клиенти.

SDLC Етап 4 – Тестване

- Тестова документация. Документи за подготовка, изпълнение и отчитане на резултатите от тестването на софтуера. Полезни темплейти.

Упражнение 9

Практическа работа по документиране на тест.

14.45 – 15.00 Почивка

15.00 – 16.00 Блок 8

SDLC Етап 5 – Еволюция

- Типични документи по предаването на софтуера за експлоатация и по оперирането, поддържането и еволюцията на системата: Искания за промяна (**Change request**), докладвани инциденти и проблеми, наблюдение на производителността и пр.

Упражнение 10

Темплейти на документи за регистриране на инциденти/проблеми и на искания за промяна.

- Внедряване на документацията като част от ежедневния гъвкав процес. Роли и отговорности по софтуерната документация.



16.00 – 16.30 Приключване на курса

- Дискусия. Въпроси и отговори.
- Обратна връзка.
- Раздаване на сертификатите.

16.30 Край на курса

