

26,27

Category	Empty
Files	Empty
Created	May 25, 2024 1:48 PM
Reminder	Empty
Status	Open
URL	Empty
Updated	June 11, 2024 12:43 PM

Въпроси:

- **Управление на процеса на тестване.** в тема 26 - какво се очаква да напишем
- Участници в софтуерния процес - 26

27:

- Качествени атрибути (КА) - дизайн на архитектурата. - за качествени изисквания ли се пита?
 - attribute driven design или да опишем качествени изисквания и основни компоненти на сценариите за качество
- Разпределени, мрежови, децентрализирани архитектури? - какво да се пише
- **Критерии за избор на подходящи конектори**
- Типове конектори и техните **променливи характеристики?**
- Архитектурни стилове - достатъчно ли е да споменем някко стила накратко
- List

Бих се радвала да коментираме на високо ниво какво се очаква да разпишем по подточките от конспекта. По-специфично има няколко точки, за които не съм убедена какво точно се иска:

Тема 26:

- Управление на процеса на тестване
- Участници в софтуерния процес
- Верификация и валидация на софтуера. Тестване на софтуера. - достатъчно ли е да дефинираме софтуерно тестване и да опишем накратко основните видове тестване на софтуера: функционално тестване, потребителско тестване, тестване на натовареността и производителността и тестване на сигурността?

Тема 27:

- Качествени атрибути (КА) - дизайн на архитектурата
 - за качествени изисквания ли се пита?
 - attribute driven design или да опишем какво са качествени изисквания, основни компоненти на сценариите за качество
- Дизайн за постигане на ефективност, сложност, скалируемост и хетерогенност, адаптируемост на архитектурата - достатъчно ли е да посочим, че архитектурният стил, ориентиран към услуги (микроуслуги) може да послужи за постигане на посочените изисквания? Какво друго би се очаквало?
- Разпределени, мрежови, децентрализирани архитектури - какво се очаква да се пише
- Критерии за избор на подходящи контролери - не намерих да сме споменавали такива, окей ли е да кажем, че:
"конекторите представляват механизма за взаимодействие между компонентите,
то те трябва да бъдат подбрани по такъв начин, че да удовлетворяват качествените изисквания към системата. За тази цел е нужно добро познаване на възможните конектори и техният ефект върху системата."
- Типове конектори и техните променливи характеристики - под типове конектори предполагам се има в предвид различните роли, но какви променливи характеристики можем да споменем?

- Архитектурни стилове - достатъчно ли е да опишем накратко няколко стила (като pipe-and-filter, споделени данни, неявно извикване, MVC, обвивка)
-

Консултация

Тема 26

1. Софтуерен продукт и процес

- трябва да има нещо, което да задава рамки как да действат и какво да правят екипите...
- процес - набор от дейности, които да осигурят разработване на софтуерния продукт по придвидим начин с придвидими характеристики

2. Модел на софтуерния процес

Водопад, прототипен модел, personal software process

- хубаво е да споменем 2-3, да напишем кои са, кога се използват, предимства и недостатъци
- по принцип се пада цялата тема
- силно ще препоръча, ако темата се пада да се паднат само части от нея

3. СТ - надзаглавие за предните две точки

4. Управление на софтуерен проект и ресурси

- накратко - управление на проекти е дейност, която предполага няколко основни фази: планиране, проектиране, анализ на риска, оценка на риска, планиране на дейностите при възникване на някои от рисковете

5. Участници в софтуерния процес

- Включва всички заинтересовани лица(разработващите продукта също), различните модели на софтуерни процеси предполагат различно участие на заинтересованите лица
 - Може да има или няма представител на клиента
 - т.е. зависят от модела на процеса
- заинтересовани лица - краен потребител, клиент,

6. Функ. и нефунк изисквания

- Функ - отговарят на въпроса какво
- нефунк - отговарят на въпроса как
 - Може да дадем пример за нефунк х-ки

7. Анализ и проектиране на софтуерните изисквания

- Включва набор от дейности, които предполагат точно семантично определяне и извличане на изискванията към системата от всички заинтересовани лица

Резултат: добре формулирани изисквания по някакъв стандарт

Проектиране:

- Има няколко варианта за описание на изисквания, функц със описание на потока на изпълнение или сценарии на употреба / **варианти на употреба (use case)**
- Ако има време - да сложим и примерен use-case
- + табличка за описание на изискванията: предусловие, поток, постусловие, включени актьори,

8. Проектиране на софтуера

- трябва да се избере/създаде архитектура
- асоциира се с проектирането на т.нар. софтуерна архитектура
- Резултат: софтуерна архитектура, която удовлетворява изискванията
- Може да дадем **дефиниция на софтуерна архитектура**
 - структура - преведено като изглед
 - деф: (структура) конкретно представяне на някаква гледна точка към софтуерната система
 - Целта е да опрости нещата и да ги направи недвусмислени за всички заинтересовани лица
 - За да е достатъчно проста, трябва да е разделена на различни гледни точки, които представят определени детайли и предполагат включване на различни елементи
 - Важно да кажем различните гледни точки, може да дадем примери за изгледи/структури (декомпозиция на модулите, изглед на процесите да напишем накратко какво са)
 - външно видимите свойства кои са?
 - Цели се абстракция, опростяване,
 - Както в ООП дизайн разглеждаме различни гледни точки, които включват елементи, разглеждани като черни кутии. Казваме какви са отговорностите в отделните компоненти без да задълбочаваме как са реализирани.
 - За да разберем как да работим с тези черни кутии използваме интерфейсите им, т.е. външно видимите им свойства

- Практическият смисъл е **комуникация** между заинтересованите лица. За да можем да разказваме на другите хора за софтуера/архитектурата ни трябва език, на който да комуникираме. Искаме този език да е на по-абстрактно ниво, за да дискутираме аспекти на процеса и управлението на процеса. Може да споменем ADL езиците. Да покажем с коя диаграма коя структура може да се опише
 - Клас и package диаграми - модулни структури
 - sequence и state - структури на процесите
 - deployment diagram - за физически изглед

Модулни структури, структури на процесите, структури на разположението (deployment)

9. Обектно-ориентиран дизайн

- List

10. Верификация и валидация

- верификация - дали софтуера работи правилно; да установим дали разработчиците са си свършили работата правилно
- валидация - дали изискванията са окей; да установим дали разработеното е това, което ни е трябвало

alpha-beta тестване - валидация

За верификация - има формална верификация, code reviews, ...

11. Тестване на софтуера

12. Управление на процеса на тестване

- Може да се каже, че важат всичките неща за процес
- 1. да разберем какво искаме да тестваме - анализ на изискванията за тестване
- 2. планиране на тестването (какви данни ще ни трябват, какво ще тестваме, как ще тестваме)
- 3. Изпълнение на тестването
- 4. Анализ на резултатите
- 5. Заключение
- List

13. Управление на качеството на процеса на създаване на софтуер

- нещата за валидация и верификация важат

14. Съвременни софтуерни технологии

- List

15. Гъвкави (agile) софтуерни технологии

16. Feature driven development

- List

Процес: - 5 дейности (последователности в waterfall процеса) ги има в разработването на всеки комерсиален продукт. Хубаво е да ги кажем

- анализ на изискванията
- проектиране на системата
- разработване
- тестване
- поддръжка

1. Софтуерни технологии центрирани около софтуерната архитектура
СТ предполагат приложението на някакъв структуриран, добре дефиниран процес..
Важна част от дейностите в този процес е проектирането на софтуера с резултат софтуерната архитектура
От какво зависи най-много СА - от изискванията за качество
2. Качествени атрибути са качествени изисквания
3. Дизайн ...
 - Сложност - структурна сложност?
 - Какви методи да дадем? Трябва да се обосновем защо, а не просто да кажем микроуслуги
 - ефективност - колко ресурси използваме
 - Да опишем за всяко едно какъв дизайн може да ползваме
 - многослоен - хетерогенност и адаптивност
 - различните видове wrapper-и увеличават сложността
4. Надеждност и сигурност
 - и само дефиниции е ок
5. Компоненти и конектори
 - компоненти - осигуряват функционалността
 - конектори - връзките между тях
6. Типове конектори и техните променливи характеристики
 - типове конектори, техните променливи характеристики и критерии за избор - да го обединим с архитектурните стилове
7. Критерии за избор на подходящи конектори
те са качествените изисквания, които да удовлетворим.

8. Архитектурни стилове

Най-често избираме комбинация между тях.

За него точки от Дизайн до архитектурни стилове са преплетени имат много връзки и могат да се опишат като едно цяло.

- Взимаме (3) архитектурни стила, казваме плюсове и минуси, къде се използват
- Многослен архитектурен стил, микроуслуги?

9. Разпределени, мрежови, децентрализирани архитектури. Надзаглавие на следващите две

- разпределена - работи на много машини, RMI,
- децентрализирани - peer-to-peer?
- мрежови -
- но може директно по следните точки

10. Архитектури, ориентирани към услуги и уеб услуги

- Трябва да знаем какъв е интерфейсът към услугите и как да намерим..
- Архитектура, ориентирана към услуги. Компоненти:
 - потребител на услуга, доставчик на услугата, регистър на услугите user - server - register - user