|  |  |  |
| --- | --- | --- |
| EAD = (P + 4M +O) / 6  **SDLC**: WATERFALL, AGILE, KANBAN.  **Estimation techniques**: DECOMP, COMP, ANALOG, STORY  **Requirements**: SOFT, FUNC, NONFUNC, BUSIN, USER  **Most Common Requirement Risks**  Недостатнє задієння людей  Мінливі чи двозначні реквайроменти  Додавання девелопером власних функцій  Мінімальна специфікація  Необережне планування  **Characteristics of Excellent Requirements**  Закінчені  Коректні - все є обґрунтованим  Пріоритизовні  Однозначні  Можуть бути перевірені  **Specification**  Завершена  Консистентна  Може мінятись  Може відслідковуватись  **ABSTR** **QUALITIES**: COH, COUP, SUFF, COMPLETNESS  **CREATIONAL**: BUIL, PROT, FACT, ABFACT, SINGL.  **STRUCT**: ADA, DEC, BUI, COMP, FAC, FLY, PROXY  **BEHA**: CoR, COM, ITER, MED, MEM, OBS, STATE, STRA, TEMPLATE, VISITOR.  **UML** **BEH**: USE, ACTIVITY, COMMUNIC, SEQ, STATE  **UML** **ST**: CLS, PACK, OBJ, COMPON, COMPOSIT, DEPL  **SEC** **PRINCIP** CONFIDDENTIALITY, INTEGRITY, AVAILAB  **SEC RISKS**: INJECT, BROKEN IMPL, DATA EXPOSURE,  GRANT CONTROL, CONFIG(EXCEPTIONS), CSRF, DESERIALIZ, MONITORING.  **JAVA SE SEC**: PROVIDERS (CoR), CRYPTO, PKI (K + SER)  **JAVA** **AUTH**: MODULES | **CHAR of APP SEC**: AUTH,AUTH, DATA INTEGR  **EE SEC MECHAN**: APP LAYER SEC, TRANSPORT, MESSAGE(SOAP)  **AUTH: BASIC**(base64), **DIGEST**(401, nonce MD5), **FORM**(plain, sessionId)  **BOOBLE**: O(N) -> O(N2)  **MERGE**: O(N LOG N)  **QUICK**: O(N LOG N)  **BINARY** **SER**: O(LOG N)  **IP** (package), **TCP**+ (trans), **UDP** (trans)  **POP3**(FROM), **IMAP**(H FROM), **SMTP**(SEND)  **IPv4**(32 bit), **IPv6**(128)  **Transaction propagation**:  **MANDATRORY** (curent)  **NESTED**(in nested)  **NOT**\_**SUPPORT**  **REQIRED** (current, create new if no exist)  **REQUIRES**\_**NEW**(new, suspend current)  **SUPPORTS**(current, no transacrion if no exist)  **SOAP:** XML, HTTP, SECURITY, STATEFULL  **+** SCALABILTY (but same request and server)  **-** MANAGE SESSIONS & TRANSACT  **JAX**-**WS**(spec): @WebService, @WebMethod,  @SoapBinfing  **WSDL**(XML) TYPES, OPER, OP-URL BIND, PORT  CLIENT <- WSDL DOC(OPERATIONS, URLS)  **REST**: JSON/XML/HTML, STATELESS  **REST** **PRINCIPLES**: ID, GET/POST/PUT/DEL,  LAZY LOAD, DIFF DATA TYPES.  **+** SCALABILITY, LESS DATA, MORE TYPES  **-** HARD MANAGE SESSION, DEBUG | **JAX-RS** – java API for REST.  Jersey->Controller + (@Path, @Prod, @Cons) + JAXB annot = simple web service  **DDL**: CREATE, ALTER, DROP  **DML**: SELECT, INSERT, UPDATE  **DCL:** GRANT, REVOKE  **WHERE** before grouping  **STORED:** 0-M param, RETURN yes/no, data change  **FUNC**: >0 param, RETURN yes, NO data change  **GROUP BY**(column): count(column), group by (colum)  **UNION –** JOIN SELECTS**, NO** DUPLICATES  **UNION ALL** - JOIN SELECTS ,DUPLCATES  **EXPECT –** rows if not exist in second  **TRUNCATE** – delete table.  **HIB** **interf**: Session, SessionFactory, Transac  **Hibernate** **caches**:  SESSION(postpone updates), SESSION FACT, QUERY (result sets)  **JDBC**: DRVR MANAGER->DRVR->**CON**->STAT->RES  **JPA**: EMFAC->EM->TRANS->PERSIST->COMMIT  **CASCADE**: All, PERSIST, MERGE, DETACH, REFRESH, REMOVE.  **JUNIT**: ASSERT, TESTCASE, RESULT |

SEC:

MessageDigest.getInstance("SHA-256", "ProviderC");

md.update(text.getBytes());

return md.digest();

**SHA-1 160**

**SHA-2 256, 512…**

MD5 128

A (Users soft)

P (OS)

S (client-server session)

T (TCP/UDP, вирішує скільки даних відправляти чи отримувати)

N (IP, address-ip match, відправка пакетів роутером)

D (світчі які роутять конекшени)

P (передача бітів по мережі)

NAT (package info from IP to IP, mask VM)

Ethernet (low level, PC in LAN)

Custom servlet filter -> interface Filter, init(), destroy(), service(). CHoR pattern, doFilter(ServletRequest, ServletResponse, **FilterChain**)

Redirect: 302

JSTL теги:

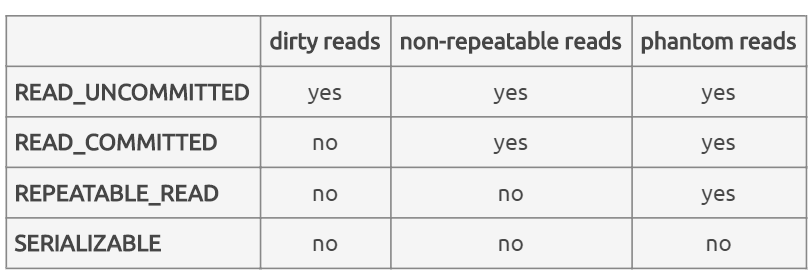
- SQL – теги для роботи з SQL. В них можна прописати dataSource і робити запити. Наприклад: sql:setDataSource, sql:query шоб заекзекютити кверю, sql:param шоб засетати параметр в кверю і так далі.

- XML – теги для роботи з XML. Наприклад: x:parse шоб парсити елемент, ну і ті ж x:choose\x:when\x:otherwise для кондішенів і так далі.

- Core – теги для роботи з змінними, для роботи з лінками, ну і шоб будувати сам флоу. Наприклад: c:if, c:forEach, c:catch щоб ловити ексепшени, c:choose\c:when\c:otherwise для кондішенів і так далі.

- JSTL Functions – набір стандартних функцій. Наприклад: fn:contains шоб перевірити чи стрінга містить сабстрінгу, fn:trim шоб потрімати стрінгу, fn:length шоб чекнути довжину стрінги і так далі.

- Formatting tags – теги для форматування дати і чисел. Наприклад: fmt:parseNumber, fmt:timeZone, fmt:ParseDate і так далі.



Redirect – 302 code