

Examen DS

1. Teoria (test albed)
2. Problema model de domini
3. Exercici SOLID (campus virtual)

• Esquema general



Quin sistema cal construir?

Què ha de fer el sistema?

Com ho fa el sistema?

• Especificació:

↳ Interacció amb l'usuari

* Casos d'ús - descripcions de les requiriments d'interaccions entre actors externs i el sistema. (Satisfereix objectius actors)

* Històries d'usuari - descripció d'un tros de funcionalitat que pot ser valuós per l'usuari final (més concret, test d'acceptació)

↳ Especificació de dades

* Model de domini - representació visual de claus conceptuals o objectes del món real en un domini d'interès.

↳ Especificació de processos

* Esquema del comportament

• Disseny

↳ Descripció dels subsistemes i components del sistema software

↳ Relació entre components

↳ Diagrama de classes

↳ Diagrama d'activitats

Tema 2

- **Requisits** - condicions i capacitats necessàries per l'usuari per solucionar un problema o assolir un objectiu
- **FURPS+** (captura de requisits)
 1. **Funcional** - capacitat, reusabilitat i requerit
 2. **Usabilitat** - factors humans, ajuda, documentació
 3. **Reliability (Fiabilitat)** - freqüència fallades, recuperació, prevenció
 4. **Performance (Rendiment)** - temps resposta, disponibilitat, prevenció
 5. **Suportabilitat** - adaptabilitat, manteniment, internacionalització
 6. **+** - implementació, proves internes, implemetat
- **Especificació** - Què ha de fer el sistema
 - ↳ **Comportament requerit** - conjunt complet de casos d'ús, especifica totes les formes d'usar el sistema
 - ↳ **Exercici** - seqüència específica d'accions i interaccions entre actors i el sistema. (instància de cas d'ús)
 - ↳ **Relació entre casos d'ús**
 1. **Inclusió** - cas d'ús dins d'un altre cas d'ús base. El cas base (origen) no té sentit sense el cas inclòs (destí)
 2. **Extensió** - cas d'ús que té la possibilitat d'executar-se en un punt concret d'un altre cas d'ús (destí). Sense el cas extès, el cas base (destí) també té sentit.
 3. **Herència** - cas d'ús gen (origen) té la mateixa especificació i alguna addició del cas d'ús pare (destí).
 - ↳ **Actor** - algú o entitat externa que presenta un cert comportament en un exercici d'un cas d'ús.
 - ↳ **Com trobar casos d'ús**
 1. Definir límits sistema
 2. Identificar actors
 3. Definir objectius actor
 4. Definir casos d'ús que satisfacin els objectius

↳ Parts d'entrevista usuària

1. Card - visió general de la situació i l'objectiu
2. Comença - preguntes concretes per conèixer la situació amb detall
3. Confirmació - permet definir els criteris o tests d'acceptació

↳ Criteris d'acceptació - condicions que el producte de desenvolupament ha de satisfer per a ser acceptat pel client

↳ Invert (característiques dels criteris d'acceptació)

1. Independents
2. Negociables
3. Valuable (pels clients)
4. Estimables (es poden planificar)
5. Small (poc temps)
6. Testables

↳ Contingut del model de domini

1. Classes conceptuals o objectes del domini
2. Relacions entre les classes conceptuals
3. Atributs de les classes conceptuals

↳ Com construir el model de Domini

1. Llista de les classes conceptuals candidates
 - * Llista de categories de classes conceptuals
 - * Identificar sintagmes nominals en els requisits
2. Definir-les al model de domini
3. Afegir-li les associacions necessàries
4. Afegir els atributs necessaris per satisfer els requisits d'informació

↳ Classes conceptuals - entitats que representen elements físics, conceptuals i estats del problema

↳ Concepte / classe - si no és un número o un text en el món real, després sempre és una classe conceptual

↳ Clases d'especificació de X - derivem a X

* S'han d'agregar quan:

1. Necessitem la descripció d'un ítem amb independència de l'existència d'un exemplar.

2. Eliminar instàncies d'un cert tipus resulta una pèrdua d'informació que ha de ser mantinguda.

3. Reduïm la informació duplicada o redundant

↳ Tipus de relacions

1. Associacions - té un nom (paràgraf verbal), esquerra a dreta

* Cada extrem pot tenir un rol

* Multiplicitat indica nombre d'instàncies d'una classe relacionades amb l'altra.

2. Agregació - un tot es relaciona amb les seves parts (Tot ~~es~~ parts)

* Composició - les parts es creen i es destrueixen amb el tot (tot ~~es~~ parts)

* Rellevança - relació d'elements amb els mateixos

3. Generalització - jerarquia de classes

* Regla 100% - la definició d'una superclasse ha de ser aplicable a les subclasses.

* Regla és-un - tots els membres d'una subclasse han de ser membres de la superclasse.

↳ Atribut - valor de dades lògic d'un concepte o classe conceptual

* Han de ser simples

* Si té: operacions, restriccions, atributs, és una unitat → és una classe

* Atribut derivat: pot ser inferit a partir d'una altra informació del model. Exemple:



En aquest cas, quantitat es pot obtenir gràcies a la multiplicitat

Exemples de requistes

1. Registre de tots els alumnes - Funcional
2. Navegador web en diferents idiomes - Suportabilitat
3. 95% transaccions tarden menys 1 segon - Rendiment
4. Els nous usuaris es familiaritzen ràpid - Usabilitat
5. Interfície orientada a finestres i manipulada amb teclat - Usabilitat
6. Sistema disponible 24/7 - Fiabilitat (o rendiment)
7. Ha de complir la llei de protecció de dades - Altres

Diagrama de casos d'ús

- Engloba tots els casos d'ús del sistema i els seus actors.
- Comportament requerit - tota les formes possibles d'usar el sistema (conjunt complet de casos d'ús)
- Cas d'ús - descripció de la seqüència d'interaccions entre actors externs i el sistema.
- Parts cas d'ús:
 1. Pròleg - ^{precondició i postcondició} resum inicial del procediment, designació actors
 2. Fluxe bàsic - procés detallat del cas ideal ← (requeriments funcionals)
 3. Fluxe alternatiu - excepcions, procés diferent al bàsic
 4. Finalitzacions - requistes finals, qüestions pendents
- Inclusió - cas d'ús dins d'un altre cas d'ús bàsic.
- Extensió - cas d'ús que té la possibilitat d'executar-se en un punt concret.
- Herència - cas d'ús que copia les interaccions d'un altre cas d'ús (pare) i pot afegir-ne més o modificar-ne.

- Com treballar els casos d'ús

1. Definir els límits del sistema
2. Identificar els actors principals
3. Definir els objectius per a cada actor.
4. Definir cases d'ús que ratifiquin els objectius d'usuari.

- Exercice d'exemple:

1. No cal initializer emissions

2. Actors: Userii nu registrat ³ → registrar-se
 inclusiv tipuri de
 Soci ³ → se log-in, newt etc., cercor etc., apunere etc.
 ↑ (Hereda)
 Monitor ³
 Temp ³