

Enginyeria del Programari

PAC 1: Introducció a l'Enginyeria del Programari i l'Orientació a Objectes

Preguntes del Mòdul 1

Pregunta 1 (15% puntuació)

Enunciat

Compara els mètodes de desenvolupament de programari de cicle de vida clàssic i de cicle vida iteratiu en els següents punts:

- Temps per obtenir el primer producte tangible.
- Flexibilitat en assumir canvis.
- Dimensió del projecte.

Indica quin mètode s'ajusta millor a cada concepte. Justifiqueu la resposta.

Solució

- En el cicle de vida iteratiu, en dividir el projecte en parts més petites podem obtenir un primer producte visible més ràpidament que en el cicle de vida en clàssic.
- En el cicle de vida iteratiu, a cada iteració ens podem plantejar canvis per satisfer necessitats que s'han descobert en la fase anterior o que cal incorporar per canvis en l'especificació inicial. En el cicle de vida clàssic cada fase està definida d'inici i es tanca en finalitzar-la, de manera que és molt poc tolerant als canvis.

- c. El cicle de vida clàssic s'ajusta millor a projectes de dimensió petita o que almenys tinguin uns requisits coneguts i fixats. El cicle de vida iteratiu s'ajusta millor a projectes més llargs per la seva flexibilitat.

Pregunta 2 (20% puntuació)

Enunciat

En els apunts us hem presentat alguns dels rols que participen en un projecte d'enginyeria de programari. Nosaltres ens centrarem en els següents: **cap de projecte, analista funcional, arquitecte, analista tècnic, programador, expert de qualitat i responsable de desplegament**.

Analitza les següents tasques que es duen a terme en un projecte, i indica quin rol desenvoluparia el **responsable** de cada una d'elles:

1. Recollir tots els requeriments del projecte.
2. Definir l'estil del codi font i la nomenclatura a usar.
3. Identificar i documentar els errors identificats en el programari.
4. Determinar les entregues alfa i beta del programari.
5. Prioritzar els requeriments.
6. Instal·lar les versions diàries del desenvolupament en un servidor de proves.
7. Definir el model de dades.
8. Definir les tecnologies de desenvolupament del projecte.
9. Implementar cada requeriment.
10. Calcular la desviació del projecte.

Solució

1. Recollir tots els requeriments del projecte: analista funcional.
2. Definir l'estil del codi font i la nomenclatura a usar: arquitecte.
3. Identificar i documentar els errors identificats en el programari: expert de qualitat.
4. Determinar les entregues alfa i beta del programari: cap de projecte.

5. Prioritzar els requeriments: analista funcional.
6. Instal·lar les versions diàries del desenvolupament en un servidor de proves: responsable de desplegament.
7. Definir el model de dades: analista tècnic.
8. Definir les tecnologies de desenvolupament del projecte: arquitecte.
9. Implementar cada requeriment: desenvolupador.
10. Calcular la desviació del projecte: cap de projecte.

Pregunta 3 (15% puntuació)

Enunciat

En el món de l'Enginyeria del programari, hi ha moltes frases cèlebres. Aquestes dues fan referència al control de la qualitat en el programari:

"Testing leads to failure, and failure leads to understanding"

"Truth can only be found in one place: the code"

Un element important en el desenvolupament de programari és la fase de proves, que en molts àmbits es denomina la fase de QA (*Quality Assessment*). Aquesta fase, com ja sabeu, consisteix en identificar errors en el programari i documentar-los per tal que els desenvolupadors els puguin resoldre i de nou iniciar el procés de proves. Identifica algunes eines de suport - programaris ja existents, comercials o de codi obert (open source) - que es puguin usar per documentar aquests errors identificats i indicar-ne el seu estat.

Escull tres eines de suport a les proves de programari, indicant-ne alguna característica, i si és una solució comercial o no. Finalment, indica breument quina opció de les tres escolliries i justifica'n el per què.

Nota: Indiqueu l'enllaç a cadascuna de les eines.

Solució

Hi ha múltiples opcions d'eines per la gestió d'incidències. En alguns contextos podeu sentir parlar d'eines de *ticketing*, que és una de les nomenclatures que existeixen per denominar-les. Aquesta nomenclatura prové del fet de que el diàleg entre els provadors del programari i els desenvolupadors s'efectua a través d'un *ticket*, que és on es resumeix la incidència detectada i on se'n pot fer el seguiment (responsable en cada moment, estat de la incidència, etc.).

- Jira, de la companyia Atlassian.

<https://www.atlassian.com/software/jira>

És una eina potent, que s'integra molt bé amb molts altres programaris de la pròpia companyia Atlassian, de manera que permeten cobrir tot el cicle de vida d'un projecte. Dóna suport al desenvolupament àgil. És una solució comercial.

- Youtrack, de la companyia JetBrains.

<https://www.jetbrains.com/youtrack/>

És una eina que forma part d'un conjunt d'eines de desenvolupament de programari de les més interessants del mercat (intellij, phpstorm, etc.). Dóna suport al desenvolupament àgil. És una solució de codi obert amb limitacions i comercial.

- MantisBT, de la mateixa companyia.

<http://www.mantisbt.org/>

És una eina de codi obert (tot i que la versió SaaS és de pagament). Disposa de diverses ampliacions que fan que s'ajustin a les necessitats de cada projecte, però a nivell d'integracions és més limitada.

L'elecció d'un d'aquests programaris pot dependre molt del grau de maduresa de l'empresa i del pressupost disponible. La solució de codi obert pot encaixar molt bé per projectes petits i dirigits des d'una òptica de *Get Things Done*. Les solucions comercials o intermitges ja dependran del grau de maduresa en què es trobi el projecte, el número de persones implicades i les eines de suport de tot el cicle de projecte.

Preguntes del Mòdul 2

Pregunta 4 (10% puntuació)

Enunciat

Un dels conceptes que hem vist al mòdul 2 és el d'associació polimòrfica. Poseu un exemple del món real, de coneixement comú, d'un concepte a partir del qual es pugui aplicar aquesta pràctica. Indica els passos que has seguit en aquest procés. Cal que sigui un exemple nou, que no aparegui als materials. Justifiqueu la vostra resposta.

Solució

Una empresa tecnològica té diverses persones contractades. Cada contractat té assignat un departament, que és qui n'assumeix el cost. Aquesta mateixa empresa disposa també de col·laboradors que no són contractats sinó que realitzen tasques puntuals com advocats, assessors o gestors. Aquests col·laboradors també tenen assignat un departament pels mateixos motius que els contractats.

Tant els contractats com els col·laboradors, doncs, tenen un departament assignat; i tant els contractats com els col·laboradors tenen un conjunt de dades que pot ser comú. Podem aplicar la generalització i creem una classe *treballador* amb els atributs comuns i de la qual heretaran tant els *contractats* com els *col·laboradors*. La classe *treballadors* és la que tindrà una associació amb la classe *departament* i aquesta associació és l'associació polimòrfica, heretada doncs per part de la classe *contractats* i la classe *col·laboradors*.

Pregunta 5 (10% puntuació)

Enunciat

Suposeu un sistema orientat a objectes amb una classe abstracta *Persona* que té una subclasse concreta *Agent*. Suposeu que la classe *Persona* té un atribut privat *dataAlta*, un atribut protegit *identificador* i un atribut públic *nom*; i que la classe *Agent* té un atribut privat *adreça* i un atribut públic *email*.

Volem estudiar la visibilitat dels atributs mencionats.

- Suposeu que una operació *op1* de la classe *Agent* rep una instància *a0* de tipus *Agent*. Fes una llista dels atributs que té la instància *a0*. Per cada un d'ells, indica si *op1* hi tindrà accés i justifica breument per què.

Solució

La instància d'*a0* té els següents atributs:

- *dataAlta*: no accessible perquè és privat i sols accessible des de la classe *Persona*.
- *identificador*: accessible perquè és protegit i *Agent* és una subclasse de *Persona*.
- *nom*: accessible per a tothom.
- *adreça*: accessible perquè és privat per la classe *Agent* i *a0* és una instància d'*Agent*.
- *email*: accessible per a tothom.

Pregunta 6 (30% puntuació)

Enunciat

Suposeu que volem modelar, utilitzant orientació a objectes, un sistema de gestió d'e-mails molt bàsic.

Un *e-mail* té un assumpte, un destinatari (o varis) i un text. A més, els e-mails poden tenir cap o varis documents adjunts. Per cada document adjunt volem registrar el nom, el *content-type* i la mida en bytes.

Els documents adjunts poden ser del tipus *embedded*, dels quals se n'haurà de saber el *contentId* (un identificador intern); o del tipus *attachment*, amb la qual cosa cal saber-ne el nom original.

A més, els e-mails tenen una o diverses etiquetes, les quals tenen un nom i un color. Les etiquetes es poden organitzar de manera jeràrquica entre elles, de manera que una etiqueta sols sigui accessible sota una altra etiqueta.

- a. (15%) Fes una llista de totes les classes que puguis identificar. Per a cada classe, indica si és abstracta o no i, en cas que sigui una subclasse, de quina altra classe ho és. A més, per a cada classe, fes una llista dels seus atributs. Per a aquells que siguin opcionals i/o multivaluats, indica-ho al costat del seu nom.
- b. (15%) Fes una llista de totes les associacions i/o classes associatives que identifiquis. Per a cada associació, utilitza la següent plantilla substituint-ne les paraules entre parèntesi pel que correspongui (i traient els parèntesis):
 - (nomAssociació): Associa cada instància de (Classe 1) amb (cap o una / una i només una / qualsevol nombre d') instància/es de (Classe 2) i cada instància de (Classe 2) amb (cap o una / una i només una / qualsevol nombre d') instància/es de (Classe 1).

Solució

a)

- Email (concreta): assumpte, text, destinataris (multivaluat).
- Etiqueta (concreta): nom, color.
- Adjunt (abstracta): nom, content-type, mida.
- Embedded (concreta, subclasse d'Adjunt): content-id.
- Attachment (concreta, subclasse d'Adjunt): nomOriginal.

Una solució alternativa és considerar els destinataris com una classe:

- Destinatari (concreta): adreça.

En aquest cas, la classe Email no tindria l'atribut multiavaluat destinataris.

b)

- *té*: Associa una instància d'Etiqueta amb qualsevol nombre d'instàncies d'Email, i cada instància d'Email amb una o més d'una instància d'Etiqueta.
- *conté*: Associa cada instància d'Email amb qualsevol nombre d'instàncies d'Adjunt, i cada instància d'Adjunt amb cap o una instància de Email.
- *pertany (associació reflexiva)*: Associa una instància d'Etiqueta amb cap o una instància d'Etiqueta, i cada instància d'Etiqueta amb qualsevol nombre d'instàncies d'Etiqueta.

En el cas d'haver desenvolupat la solució alternativa comentada a l'apartat a), cal definir una nova associació:

- *s'adreça*: Associa una instància d'Email amb qualsevol nombre d'instàncies de Destinatari, i cada instància de Destinatari amb qualsevol nombre d'instàncies d'Email.