

MATERIA
Enxeñaría do Software

TITULACIÓN
Grao en Enxeñaría Informática

unidade
didáctica
3

Enxeñaría de requisitos

José A. Taboada González

Linguaxes e Sistemas Informáticos
Departamento de Electrónica e Computación
Escola Técnica Superior de Enxeñaría

unidadesdidácticas
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

DESCATALOGADO

© Universidade de Santiago de Compostela, 2013



Esta obra atópase baixo unha licenza Creative Commons BY-NC-SA 3.0. Calquera forma de reprodución, distribución, comunicación pública ou transformación desta obra non incluída na licenza Creative Commons BY-NC-SA 3.0 só pode ser realizada coa autorización expresa dos titulares, salvo excepción prevista pola lei. Pode acceder Vde. ao texto completo da licenza nesta ligazón:

<http://creativecommons.org/licenses/by-nc-sa/3.0/es/legalcode.gl>

Deseño e maquetación

J. M. Gairí

Edita

Vicerreitoría de Estudantes,
Cultura e Formación Continua
da Universidade de Santiago de Compostela
Servizo de Publicacións
da Universidade de Santiago de Compostela

ISBN

978-84-9887-527-0

MATERIA: Enxeñaría do software

TITULACIÓN: Enxeñaría Técnica en Informática de Sistemas

PROGRAMA XERAL DO CURSO

Localización da presente unidade didáctica

Unidade I. O software. Un produto de enxeñaría

- Evolución da industria do software
- Características do software
- Atributos
- Clasificación das aplicacións
- O software herdado

Unidade II. O proceso de construción do software

- Definicións e normas
- Definicións de enxeñaría do software
- Os procesos de construción do software
- Os ciclos de vida
- Modelos do ciclo de vida do software
- Desenvolvemento áxil

Unidade III. Enxeñaría de requisitos

- Fundamentos
- Actividades iniciais
- Especificación
- Análise, modelado
- V&V, e xestión de requisitos

Unidade IV. Probas do software.

- Filosofía das probas do software.
- Probas estruturais
- Proba funcional
- Execución das probas
- Estratexia de aplicación das probas
- Probas en desenvolvementos orientados a obxectos

ÍNDICE

Presentación

Os obxectivos

Os principios metodolóxicos

Os contidos básicos

1. Enxeñaría de requisitos. Fundamentos
2. Actividades iniciais
3. Especificación
4. Análise, modelado
5. V&V, e xestión de requisitos

Actividades propostas

Avaliación da unidade didáctica

Anexos

- Anexo 1. Textos para lectura complementaria

Bibliografía

PRESENTACIÓN

Nos temas anteriores estableceuse o software como un elemento de Enxeñaría e propuxéronse unha serie de normas e paradigmas que definen os procesos necesarios para a súa construción, o primeiro dos cales é a análise do sistema e dos requisitos do software.

Nesta fase o obxectivo fundamental é definir de xeito inequívoco que problema estamos tratando de resolver. A problemática que se xera é tal que, en realidade, o que acabamos de mencionar como un proceso, se divide en varios que mesmo se agrupan de forma separada debido á diferente natureza das actividades implicadas: Identificación de necesidades, comprensión e modelado, especificación, verificación e validación dos requisitos e xestión dos mesmos.

A importancia deste proceso pode apreciarse no controvertido Chaos Report que sinala, entre as 10 causas máis significativas de fracaso dos proxectos, 6 que están directamente relacionadas co traballo realizado nas actividades de análise. A dificultade na realización deste proceso compréndese pola necesidade de obter a descrición dun elemento lóxico, o software xerado en contornas moi diversas, a partir da descrición en linguaxe natural, inherentemente ambigua, e procedente de distintos implicados cuxa cultura e formación lle fai que expresen ideas semellantes de forma moi distinta.

Para abordar esta problemática propóñense técnicas para a extracción da información e para representar o sistema a construír mediante modelos que permitan demostrar que se comprendeu correctamente a información transmitida e que o sistema que representan é o que realmente se necesita construír.

OS OBXECTIVOS

- **obxectivo 1.** Mostrar ao alumno o concepto de requisito e a dificultade da súa formulación sobre obxectos abstractos como o software.
- **obxectivo 2.** Facer que o alumno distinga as actividades vinculados aos requisitos.
- **obxectivo 3.** Proporcionar ao alumno técnicas e métodos que lle permitan a realización das actividades identificadas.
- **obxectivo 4.** Propoñer ao alumno estratexias iterativas que secuencien as actividades descritas.
- **obxectivo 5.** Que o alumno comprenda a enorme importancia que ten a enxeñaría de requisitos para o resultado final dun proxecto e, polo tanto, o risco que supón.
- **obxectivo 6.** Que o alumno comprenda a relación da enxeñaría de requisitos coa realización dun proxecto e en particular coa fase final de aceptación do cliente do produto entregado a través das probas de aceptación.
- **obxectivo 7.** Facer consciente ao alumno da necesidade da autoformación.

OS PRINCIPIOS METODOLÓXICOS

Os contidos teóricos partirán do concepto de requisito e a súa importancia nas aplicacións software para posteriormente describir os procesos que os involucran no desenvolvemento dun proxecto e as técnicas e métodos precisos para a súa realización.

Para facilitar a comprensión do concepto e os procesos recorrerase á descrición de exemplos prácticos que non sempre se corresponderán ao campo do software. Nestes casos tratarase de darlle aos alumnos propostas máis visuais dos problemas e mostrarlles como a natureza abstracta do software só agrava os problemas.

A realización de traballos prácticos en grupo e o seu intercambio será esencial para proporcionarlle ao alumno a visión axeitada da dificultade de discutir e transmitir información de obxectos abstractos.

OS CONTIDOS BÁSICOS

1. Enxeñaría de requisitos. Fundamentos

No primeiro tema desta unidade comezarase discutindo como se aborda e que importancia se lle dá á análise de requisitos en diferentes estándares e normas que describen procesos de Enxeñaría de Software e como este proceso encaixa nos distintos ciclos de vida. Preténdese que o alumno adquira conciencia do valor que se lle dá na enxeñaría a este proceso cuxa importancia tamén se destacará na parte final deste tema a través da discusión do impacto que os erros nel cometidos teñen para o éxito ou fracaso dos proxectos software.

Nesta revisión inicial sinálanse as actividades vinculadas ao proceso, destacando a variabilidade da súa natureza, que supoñen técnicas e procesos mentais totalmente diferentes e que inciden nas habilidades requeridas no responsable do proceso. No tema tamén se abordan cuestión relevantes como o custo ou os participantes no proceso.

O resto do tema dedícase esencialmente á comprensión do concepto de requisito, partindo das definicións formais e propoñendo distintas clasificacións que darán idea da enorme dimensionalidade deste concepto, que é, en si mesmo, causa da complexidade do proceso e da maioría dos erros que poden cometerse nel.

2. Actividades iniciais

Neste tema centrarémonos na contextualización das actividades en distintos modelos que nos propoñerán aproximacións iterativas na súa realización e detallaranse aquelas vinculadas ao inicio do ciclo. Cabe sinalar que entre estas actividades se inclúe o estudo da viabilidade do proxecto. Aínda que tal estudo non forma parte do proceso de obtención de requisitos e formalización das necesidades inclúese aquí debido á súa proximidade coa enxeñaría de requisitos, ademais o estudo da

viabilidade utiliza o documento de requisitos e debe realizarse nas mesmas etapas temperás que o resto de actividades que se describen.

O resto do tema dedícase ás actividades iniciais centradas no descubrimento ou elicitación de requisitos que pasan por identificar ás fontes e obter delas a información que é necesaria. Con tal obxectivo descríbense unha serie de técnicas como: observación, estudo de documentación, cuestionarios, remuíño de ideas, ferramentas JAD, entrevistas, prototipado e puntos de vista.

3. Especificación

Este tema está centrado no documento de especificación. Comeza coa definición do documento que fai o IEEE e coa descrición xenérica da información que debe conter. Posteriormente estúdanse varias propostas de estrutura para o documento entrando a describir, con maior nivel de detalle, a proposta no estándar IEEE/ANSI 830, onde, ademais de detallar as seccións se inclúe a desagregación de como a información do documento se distribúe en cada unha delas. Debido á propia flexibilidade do estándar sinálase que a súa estrutura pode variar ao recoller os requisitos funcionais e se mostran varias propostas recollidas no estándar

O tema tamén recolle as propiedades que deben cumprir os documentos de especificación e que están descritas no propio estándar. Estas propiedades como: non ser ambigua, ser doado de verificar ou ser consistente entre outras están detalladamente explicadas no tema e teñen unha importancia fundamental xa que, dado que dan validez a estes documentos, serán os criterios que se utilizarán para a valoración das prácticas entregadas polo alumno.

4. Análise, modelado

Comezamos agora coa discusión sobre os obxectivos que persegue a análise e a necesidade e utilidade dos modelos para representar o software de forma simple e comprensible. Tamén se discute a necesidade de combinar modelos de distintos tipos debido a que o problema presenta distintas dimensións e debe mostrarse con distintos graos de abstracción. Para abordar os distintos niveis de abstracción propónse unha estrutura Top-down que parte do nivel máis abstracto e vaia afondando para aumentar o nivel de detalle. Esta formulación é o que se seguirá no tema para ordenar os distintos modelos presentados comezando por aqueles que deben usarse para describir o sistema completo e deixando para o final os que dan maior nivel de detalle.

Ademais no tema discútese que non se está a tratar cun problema unidimensional senón que, igual que para representar unha peza de enxeñaría temos que facer unha descrición nos tres eixes do espazo, na descrición do software tamén temos que representar varias dimensións. En particular as dimensións da información, da función e do comportamento, cada unha das cales precisa de modelos específicos para representalas. Desta forma verase como chegados a un

punto de detalle a descrición dun único problema implica a construción de varios modelos que deben ser consistentes entre si.

A estrutura Top-down comentada nos parágrafos anteriores para a descrición dos modelos implica comezar polo modelo do sistema. Este non sempre é necesario xa que unicamente se incluírá cando se trate de sistemas mixtos de software e hardware. Neste modelo propónse utilizar diagramas de bloques no que se identifiquen os subsistemas nos que poidamos dividir o problema asignando responsabilidades a cada un deles.

Cando o problema se aborda unicamente a nivel software recoméndase comezar a través de diagramas de arquitectura nos que se reflicte, de novo, unha posible división de responsabilidades. Este diagrama de arquitectura, igual que no do sistema, só trata de dividir o problema para facelo máis alcanzable polo que se debe facer consciente ao alumno que pode ser distinto do proposto finalmente para a arquitectura xerada na etapa de deseño, onde xa se está a describir unha solución concreta. Se o problema é o suficientemente complexo, Pressman, proponnos transformar estes diagramas nunha xerarquía no que cada subsistema explota nun novo diagrama de bloques que o describe con maior nivel de detalle. Esta estratexia trátase con detalle na construción do modelo de diagramas de fluxo polo que aquí só se comenta brevemente.

Unha vez que a arquitectura nos simplificou os subproblemas a un nivel aceptable comecemos a usar modelos máis refinados cos que representar todas as dimensións do problema, e dado que ningún modelo pode representar todas as dimensións, será preciso crear varios complementarios cos que construír unha representación completa. As dimensións que se deben abordar son: a dimensión da Información, que describe con que datos traballa o sistema; a dimensión da función, que describe as funcionalidades que se abordan ou como a información é transformada polo sistema; e a dimensión do comportamento, que describe cando as funcionalidades do sistema se levan a cabo ou, o que é o mesmo, por que estados de funcionamento pasa o sistema e que condiciona a transición entre eles.

A elección dos modelos concretos que poden usarse en cada caso dependen de diversos factores como a experiencia do noso interlocutor cliente ou o paradigma de desenvolvemento que vai utilizarse. En calquera caso o analista estará tanto máis preparado para realizar esta función cantos máis modelos coñeza polo que no tema non nos restrinximos aos modelos orientados a obxectos e inclúense tamén os habitualmente usados nun enfoque estruturado

Desta forma, para un enfoque estruturado preséntanse os modelos de diagrama Entidade-Relación, os diagramas de fluxo de datos e diagramas de fluxo de control. Para o enfoque orientado a obxectos propóñense os modelos de casos de uso, diagramas de clase e diagramas de actividades aínda que tamén se sinalan como posibles os diagramas de estado que tamén se usan neste paradigma.

5. V&V, e xestión de requisitos

Ao principio da unidade coméntase que os procesos de análise de requisitos seguen modelos iterativos, sendo moi excepcional que isto non sexa así. Os procesos descritos neste tema permiten pechar cada iteración confirmando se se fixo de forma correcta e se o problema que describen os requisitos é realmente o que ten o cliente. Estes dous obxectivos son abordados pola verificación, centrada en probar que a análise se fixo correctamente, e a validación, centrada en probar que se identificou correctamente o problema. Se ben no resto das procesos de enxeñaría estes dous procesos se encontran claramente diferenciados na análise confúndense con facilidade porque o proceso está moi preto do cliente.

En calquera caso as iteracións do proceso deben realizarse de forma controlada xa que a especificación propón un documento que compromete tanto o equipo de desenvolvemento coma o cliente. Por conseguinte calquera cambio neste documento pode producir conflitos e debe realizarse de forma consensuada por ambas as dúas partes.

Para abordar esta problemática descríbese o proceso de xestión de requisitos que, como mínimo incluírá o repositorio documental para os requisitos e un proceso de control de cambio que, á súa vez, xerará demandas no documento como a necesidade da trazabilidade dos requisitos esta trazabilidade farase cara atrás , para permitir saber quen solicitou un requisito ou que obxectivo se persegue con el, e cara adiante , para permitir determinar os elementos de deseño que xera. Ambas as dúas trazabilidades son imprescindibles para poder analizar a importancia, o custo e o risco dos cambios.

ACTIVIDADES PROPOSTAS

- Desenvolvemento dunha sesión TFEA para a elicitación de requisitos dun produto software. Nesta reunión cada alumno participará con propostas individuais.
- Construción dun documento de especificación de requisitos do produto. Esta práctica realizarase en grupo e incluírá: obxectivos do software, os seus requisitos clasificados e descritos seguindo un modelo determinado, e o modelado do software incluíndo a súa división en subsistemas.

AVALIACIÓN DA UNIDADE DIDÁCTICA

- Valorarase o traballo individual do alumno, realizado de forma previa e durante a sesión TFEA, para determinar a comprensión do alumno do

concepto de requisito e a súa capacidade para xeralos correctamente libres dos problemas descritos nos contidos teóricos.

- Valorarase o traballo conxunto realizado coa especificación de requisitos de acordo ás propiedades que o estándar IEEE/ANSI 830 sinala que debe ter unha especificación.

ANEXOS

Anexo 1. Textos para lectura complementaria

Recoméndase a lectura dos seguintes capítulos da bibliografía proposta

Os capítulos 6, 7 e 8 do libro de Piattini 2003 que no seu capítulo 6 trata as actividades de recollida de información e de análise de viabilidade do proxecto en función dos seus requisitos. Este autor é o que fai un tratamento máis extenso da viabilidade e tamén trata técnicas de obtención de información como as entrevistas e as JAD que se propoñen como práctica. No capítulo 7 céntrase nas características que debe unha boa especificación para que cumpra a súa función axeitadamente tal e como se recollen na materia. Nel tamén se tratan as técnicas de modelado orientadas a obxectos que poden ser utilizadas nas devanditas especificacións das que recollemos sobre todo as de modelado de funcións.

Os capítulos 4 e 5 do libro de Sommerville 2011 que presenta, no seu capítulo 6, a forma na que poden clasificarse os requisitos e as características que presentan cada un dos devanditos tipos. Tamén se presenta o estándar IEEE/ANSI 830-1998 para definir a estrutura e as características que debe ter a especificación. Finalmente presenta algunhas técnicas de obtención de requisitos, a validación e as necesidades de xestión que formulan os requisitos e, por ende, dos documentos de especificación. As técnicas de modelado, sobre todo orientadas a obxectos, inclúense neste libro no seu capítulo 5.

Os capítulos 7 e 8 do libro de Pressman 2006 no que o 7 é un capítulo introdutorio con todos os procesos vinculados á enxeñaría de requisitos incluídos os de xestión e inclúense aspectos como a negociación que non é abordada na materia. As técnicas de modelado discútnense extensamente no capítulo 8.

BIBLIOGRAFÍA

- PIATTINI, MARIO ET AL. (2003): *Análisis y diseño detallado de aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería de Software*. Ed. RA-MA, España. ISBN: 84-7897-587-X.
- PIATTINI, MARIO ET AL. (1996): *Análisis y diseño detallado de aplicaciones Informáticas de Gestión*. Ed. RA-MA, ESPAÑA. ISBN: 84-7897-233-1
- PRESSMAN, ROGER S. (2006): *Ingeniería del Software. Un enfoque práctico*. 6ª Ed. Mc Graw Hill, España. ISBN: 970-10-5473-3.
- SOMMERVILLE, IAN. 2011; *Ingeniería de Software*. 9ª Ed. Addison Wesley, México. ISBN: 978-607-32-0603-7

Estándares relacionados

- IEEE (1998): *IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998, Revision of IEEE Std 830-1993*. Nueva York (EEUU). IEEE Computer Society Press.
- IEEE (1986): *IEEE Std. 1008-1986. Standard for Software Unit Testing*, IEEE.
- IEEE (1986): *IEEE 1012/1986, Standard for Software Verification and Validation Plans*, Nueva York (EE.UU.). IEEE.
- IEEE (1990): *Computer Dictionary. Compilation of IEEE Standard Computer Glossaries*, IEEE Std. 610-1990, Nueva York (EE.UU.), IEEE Computer Society.
- IEEE (1991): *IEEE. Standard for Developing Software Life Cycle Processes. IEEE Std. 1074-1991*. Nueva York (EE.UU.), IEEE Computer Society.
- IEEE (1999): *IEEE Software Engineering Standards Collection 1999 Edition. Volume 1: Customer and Terminology Standards*. Nueva York (EEUU). IEEE Computer Society Press.
- ISO (2008): *ISO/IEC Systems and software engineering – Software life cycle processes, ISO/IEC 12207:2008*. Ginebra (Suiza).



Unha colección orientada a editar materiais docentes de calidade e pensada para apoiar o traballo do profesorado e do alumnado de todas as materias e titulacións da universidade

unidadesdidácticas
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA