



unidade **didáctica** 2

O PROCESO DE CONSTRUCCIÓN DO SOFTWARE

José A. Taboada González
Departamento de Electrónica e Computación
Escola Técnica Superior de Enxeñaría



© Universidade de Santiago de Compostela, 2008

Deseño
Unidixital

Edita
Vicerreitoría de Cultura
da Universidade de Santiago de Compostela
Servizo de Publicacións
da Universidade de Santiago de Compostela

Imprime
Unidixital
Servizo de Edición Dixital da
Universidade de Santiago de Compostela

Dep. Legal: C 864-2008
ISBN 978-84-9750-930-5

ADVERTENCIA LEGAL: reservados todos os dereitos.
Queda prohibida a duplicación parcial ou total desta
obra, en calquera forma ou por calquera medio (elec-
trónico, mecánico, gravación, fotocopia ou outros) sen
consentimento expreso por escrito dos editores.

BIBLIOGRAFÍA BÁSICA

- 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. 2001; "Ingeniería del Software. Un enfoque práctico". Ed. Mc Graw Hill, España. ISBN: 84-481-3214-9.
- SOMMERVILLE, Ian. 2002; "Ingeniería de Software". Ed. Addison Wesley, México. ISBN: 970-26-0206-8.
- COLMENAR, Antonio et al. 2001; "Gestión de proyectos con Microsoft Project 2000", Ed. RA-MA, España. ISBN: 84-7897-443-1.

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 CONSTRUCCIÓN 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. A ANÁLISE

Análises de requisitos

- Análise de requisitos do sistema
- Análise de requisitos do software

Análise estruturado

- Técnicas de especificación e modelado
- Consistencia entre modelos
- Metodoloxía da análise estruturada

UNIDADE IV. PROBAS DO SOFTWARE

O proceso de proba

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

ANEXO: TEXTOS PARA LECTURA COMPLEMENTARIA

Recoméndase a lectura dos seguintes capítulos da bibliografía proposta:

Tema 3 de Piattini 2003: neste capítulo trátanse con detalle as normas de modelos de procesos que se discutirán nesta unidade. Neste capítulo tamén se describen en detalle os ciclos de vida clásicos. No capítulo 3 de Piattini 1996 podedes obter unha visión detallada da norma 12207 que en Piattini 2003 se obvia debido ao seu aliñamento coa norma ISO/IEC 15504 que é a que se describe en detalle.

Tema 4 de Pressman: este capítulo, no que se introduce o desenvolvemento áxil, foi tomado de referencia para a presentación na unidade deste aspecto.

Tema 4 de Sommersville: neste capítulo trátanse os ciclos de vida clásicos de forma máis breve que en Piattini. Os procesos discútense segundo a perspectiva e o criterio do autor.

Tema 17 de Sommersville: neste tema discútense, con máis extensión da tratada na unidade, os métodos de desenvolvementos áxiles, facendo especial fincapé na metodoloxía XP que tamén se trata na unidade.

Tema 3 de Pressman: neste capítulo discútense os modelos do ciclo de vida aínda que o autor introduce na explicación propostas propias que modifican o proceso orixinal. Aínda que as achegas do autor son interesantes é o menos adecuado para ter unha visión clara do proceso orixinal.

Lecturas avanzadas:

BOEHM, B. W., 1998; "A Spiral Model of Software Development and Enhancement. Computer", pp 88-72. Neste artigo Boehm fai un repaso dos modelos clásicos do momento e presenta o seu modelo de desenvolvemento en espiral, que se viu nesta unidade didáctica, como un dos modelos clásicos.

ACTIVIDADES PROPOSTAS

- Presentación da ferramenta Project para a planificación de proxectos.
- Traballo individual dos alumnos para identificar, nunha secuencia de actividades dun proxecto de software, as que pertencen a cada proceso de enxeñaría. Posteriormente, deberanse engadir aquelas da norma 12207 que non estean presentes.
- Debater cos alumnos unha serie de afirmacións sobre cada un dos ciclos de vida.
- A partir dunhas descricións ambiguas de proxectos software, os alumnos divididos en grupos de traballo deberán defender a súa opción de ciclo de vida e atacar as opcións dos outros equipos.

AVALIACIÓN DAS UNIDADES DIDÁCTICAS

- Os traballos realizados co Project permitirannos avaliar a asimilación dos obxectivos perseguidos nos procesos presentados.
- Valorarase, mediante as discusións en grupo, a capacidade do alumno para defender razoadamente un ciclo de vida na realización dun determinado proxecto.

A POSTA EN PRÁCTICA DESTA UNIDADE NO TRABALLO FINAL DO CURSO

Os alumnos deberán realizar a planificación dun proxecto de construción de software no que, coa guía de exercicios previos, seleccionen o ciclo de vida axeitado para o traballo e ponderen o esforzo a realizar en cada proceso.

1. Deberán razoar a súa proposta de ciclo de vida e as ponderacións.
2. Deberán facer un seguimento da planificación realizada.

ÍNDICE

Presentación	7
Os obxectivos	8
Os principios metodolóxicos	8
1. Definicións e normas.....	8
1.1. Definicións de enxeñaría do software.....	9
1.2. Os procesos de construción do software.....	9
2. Os ciclos de vida.....	10
2.1. Modelos do ciclo de vida do software	10
2.2. Desenvolvemento áxil.....	11
Actividades propostas	12
Avaliación das unidades didácticas	12
A posta en práctica desta unidade no traballo final do curso	12
Anexo: Textos para lectura complementaria	13
Bibliografía básica	14

A principal limitación de todos estes modelos é que, sen pretendelo, resultan tremendamente pesados. Quere isto dicir que, para garantir a calidade do proceso, xeran gran cantidade de documentación que tende a enlentece, ás veces ata de forma inaceptable, o proceso de xeración de software. Para evitar esta limitación xorde o concepto de desenvolvemento áxil.

2.2. Desenvolvemento áxil

O desenvolvemento áxil ou lixeiro xorde como contraposición ás metodoloxías tradicionais que centran o éxito do proceso na sistemática e na disciplina. Como resultado xérase unha gran cantidade de información que debe ser tamén mantida con grandes esforzos por parte do equipo de desenvolvemento.

Pero o peso da documentación non é a única limitación que se aprecia nestas metodoloxías xa que a esixencia de disciplina, para a conservación do método, xera defectos asociados ás limitacións humanas. Dado que o persoal que traballa no proxecto non son robots, e este é o papel que lles conceden estas metodoloxías, calquera erro humano na disciplina xera un defecto no proceso.

Para enfrontarse a estas limitacións, as metodoloxías áxiles propoñen adaptar os métodos aos grupos de traballo. Obviamente isto reducirá o rendemento do proceso, e asúmese que será necesario repetir algunhas tarefas, desta forma a disciplina das anteriores propostas substitúese agora pola tolerancia. O éxito depende agora da calidade do equipo de desenvolvemento xa que debe ser este o que decida a calidade e a cantidade dos modelos, o esforzo e a organización do traballo.

Aínda que son bastantes as metodoloxías que seguen esta formulación centrámonos en "Extreme Programming XP", por ser unha das que alcanzou maior difusión e renome. Nela os únicos modelos que se producen son o código da aplicación, que se xera en entregas o máis frecuentes posibles, e uns cantos artefactos de deseño como as tarxetas CRC e as solucións de pico. O control e a planificación dos esforzos levaráanse a cabo a través das historias de usuario que establecen o tamaño e a frecuencia dos incrementos entregados ao usuario. Estes entréganse en períodos de días ou semanas e constitúen aplicacións usables. Para iso reconstrúese, en parellas, o código e o deseño constantemente nun proceso de refactorización que se avalía constantemente mediante a automatización das probas.

temente, da IS, é imprescindible adquirir unha capacidade autoformativa que lle permita manter a súa capacitación profesional.

2. Os ciclos de vida

O tema anterior proporcionounos unha visión estática da construción do software. Nel centrámonos en describir as pezas necesarias para levar a cabo esta tarefa pero obviámos o problema de que estas deben desenvolverse no tempo de acordo a algunha secuencia.

A continuación, trataremos o problema da secuencialización dos procesos. Partiremos dunha visión histórica que nos aproxima aos modelos preceptivos clásicos do desenvolvemento do software e finalizaremos discutindo as súas limitacións e as últimas propostas baseadas no que se deu en chamar desenvolvemento áxil.

2.1. Modelos do ciclo de vida do software

Os modelos dos ciclos de vida describen as distintas vinculacións dos procesos de enxeñaría discutidos anteriormente. Deles, o máis antigo é o modelo en cadoiro que propón unha conexión secuencial de todas as actividades. O modelo controla o paso dun proceso ao seguinte pola entrega dunha documentación que garante a conclusión da etapa anterior e serve de base para o comezo da seguinte. A principal achega do modelo é que permite sistematizar a construción do software facendo o proceso repetible, permite estimar os seus custos e seguir o seu progreso.

O principal defecto do ciclo de vida en cadoiro é a súa secuencialidade. Aínda que este modelo ten éxito noutras enxeñarías, as peculiaridades do software fáno un mal candidato a unha construción monolítica. En xeral, son máis exitosas as estratexias que permitan a construción mediante iteracións, cada unha das cales xera unha mellor aproximación ao entregable final. Con este obxectivo expóñense outros ciclos de vida como o incremental ou o de prototipos. Estes modelos, denominados evolutivos, xeran entregables intermedios que poden ser analizados e criticados polo cliente. A principal diferenza céntrase en que no modelo incremental cada entregable é un produto acabado ao que unicamente lle falta algunha funcionalidade. No modelo de prototipos, cada entregable é un prototipo co que se quere mostrar ou comprobar algo para o cliente pero todos eles deben ser refugados para construír o entregable final.

O último modelo clásico estudado é o modelo en espiral, creado por Boehm en 1988, e que é o primeiro que está dirixido por riscos. Neste modelo, os entregables seguen un proceso en espiral no que, en cada revolución, créase un documento, no sentido amplo da palabra. Na proposta inicial, os devanditos documentos coincidían cos propostos no modelo en cadoiro. Neste caso, a diferenza consiste en que con cada revolución, se analiza o risco de realizar a etapa e se seguen estratexias para minimizar o risco, o que habitualmente supón o desenvolvemento dalgún prototipo.

PRESENTACIÓN

A identificación do software como un elemento de enxeñaría cambiou radicalmente a formulación proposta no seu proceso de construción. Desde unha concepción inicial no que na construción do software, como unha composición artística, se prima a creatividade do programador, trátase de pasar a un proceso de construción sistemático e repetible, onde a codificación é unha das fases presentes, e non precisamente, a máis creativa.

Nace deste xeito unha nova especialidade denominada Enxeñaría de Software que ten por obxectivo aplicar os principios da enxeñaría ao software. Nesta formulación xorde como primeiro problema a identificación dos procesos implicados na construción do software. Partindo dos procesos presentes nas enxeñarías tradicionais o SEI (Software Engineering Instituto) e outras organizacións internacionais como a ISO (International Standard Organization) xeraron normas de modelos nas que se definen con precisión os distintos procesos que se foron identificando.

A identificación dos devanditos procesos non supón, con todo, o final do traballo, xa que, a construción do software esixe o seu encadeamento temporal. Nos ciclos de vida do software discútese os procesos e o seu desenvolvemento dende que a aplicación xorde coma unha idea ata que, unha vez construído e usado por un período de tempo, se abandona para ser substituído por unha nova aplicación.

Con frecuencia o sistematismo e a disciplina introducida na construción do software contrólanse mediante a xeración dunha extensa documentación. Esta documentación, que ten por obxecto garantir a calidade do produto e do proceso, reduce a velocidade na construción e dificulta a adaptación ao cambio das necesidades do cliente. Como resposta a estes problemas xorde o desenvolvemento áxil, que trata de alixeirar o peso da documentación aínda á conta de sacrificar a eficiencia do proceso.

OS OBXECTIVOS

- Expor a necesidade de aplicar os principios da enxeñaría tradicional á construción do software.
- Presentar os procesos esenciais na construción do software e xustificar a súa importancia.
- Enumerar os procesos de construción de software e a súa evolución.
- Describir as estratexias para o encadeamento dos procesos.
- Discutir o problema das metodoloxías pesadas e que solucións achegan as metodoloxías áxiles.
- Facer consciente o alumno da necesidade da autoformación.

OS PRINCIPIOS METODOLÓXICOS

- Exporemos os contidos teóricos partindo de coñecementos previos e nocións intuitivas que posteriormente se formalizarán a través de normas e, cando estas non existan, a través dos paradigmas máis usados.
- Para facilitar a memorización e o uso dos conceptos e procesos, no seu contexto axeitado, trataremos de asocialos cos levados a cabo noutras enxeñarías, propondo exemplos que poidan resultar próximos ao alumno.
- Realización de traballos prácticos en grupo para fomentar o espírito crítico e a colaboración.

OS CONTIDOS BÁSICOS

1. Definicións e normas

No tema anterior establecemos que o software debe ser considerado un elemento de enxeñaría. Polo tanto, a súa construción debe seguir os principios que permitiron a estas crear elementos repetibles e de calidade a un custo controlado. Esta formulación implica abandonar a idea da construción do software como a creación intuitiva de código que satisfai unha función para considerar que supón o seguimento sistemático e disciplinado dunha serie de procesos que nos garantan o éxito de tal actividade.

No primeiro apartado deste tema preséntanse unha serie de definicións dos conceptos aplicables doutras enxeñarías. De entre estes conceptos, o resto do tema centrarase na definición dos procesos implicados na construción do software.

1.1. Definicións de enxeñaría do software

As enxeñarías propoñen unha serie de procesos, que habitualmente se desenvolven de forma secuencial, para a construción dos seus produtos. Entre tales procesos aparecen: o proceso de análise, centrado na determinación do produto que é necesario construír; o proceso de deseño, centrado na descrición de como se construírá o elemento; o proceso de construción, que dará lugar ao elemento de enxeñaría final ou a algún prototipo ou maqueta intermedios; o proceso de probas, centrado na comprobación da calidade do produto que se construíu, e os procesos de uso e mantemento, que tratan de garantir o perfecto funcionamento do produto durante a maior cantidade de tempo posible.

Estes procesos foron seleccionados porque todos eles aparecen, ou deben aparecer, na construción do software. Autores como Pressman identifican estes procesos como fases xenéricas que, de feito, están sempre presentes en todos os modelos propostos para a construción do software.

Neste punto podería parecer todo dito respecto dos procesos que utilizaremos para a construción do software pero, isto dista moito de ser certo. O instituto de enxeñaría do software e algúns organismos internacionais como a ISO, conscientes da importancia de estandarizar ao máximo estes procesos, realizaron constantes esforzos por establecer con precisión os procesos implicados na construción do software, definíndoos e detallando as actividades que estes implican. Tales definicións fóronse plasmando en normas que ano tras ano sofren modificacións para adaptar os novos conceptos e tendencias que xorden nunha especialidade tan nova como de feito é a Enxeñaría do Software.

1.2. Os procesos de construción do software

Na actualidade existen numerosas normas que definen os procesos de construción do software. Naturalmente, non é viable presentalas todas; de feito, nin sequera é posible unha visión detallada dunha delas. En lugar desta opción describiremos varias delas co obxectivo de destacar, en primeiro lugar, que todas teñen aspectos comúns pois todas tratan a construción dun elemento de enxeñaría e, en segundo lugar, que todas elas continúan evolucionando como a enxeñaría do software en si mesma.

O primeiro modelo estudado, a norma IEEE 1074, é a única claramente diferente ás outras estudadas, xa que a ISO 12207 e a ISO/IEC TR 15504-2 aínda non tratándose da mesma norma, están fortemente aliñadas e, de feito, a segunda pode considerarse unha extensión da primeira. Finalmente, presentaremos a evolución que a última das normas sofre entre as súas versións do 1998 e do 2003.

A revisión das normas debe servir para achegar ao alumno as definicións formais dos procesos. Ademais, deben facelo consciente que alén dos procesos de enxeñaría, cuxa necesidade resulta evidente, existen numerosos procesos que afectan á calidade do resultado e que, en consecuencia, deben ser igualmente comprendidos. Finalmente, é necesario que o alumno comprenda que, dada a evolución destes estándares co tempo e, consecuen-