Ingegneria del Software

Introduzione

Software: Prodotto e Processo

Ingegneria del Software: Terminologia

1

1

Produzione Software: Evoluzione

- ***Arte***: applicazioni sviluppate da singole persone e utilizzate dagli stessi sviluppatori
- ***Artigianato***: applicazioni sviluppate da piccoli gruppi specializzati per un cliente
- ***Industria***: diffusione del software in diversi settori; crescita di dimensioni, complessità e criticità delle applicazioni; mercato e concorrenza; necessità di migliorare la produttività e la qualità; gestione dei progetti; evoluzione del software

2

2

Programmi vs Prodotti (Sommerville)

- Programma: **l'autore è anche l'utente** (e.g., non è documentato, quasi mai è testato, non c'è progetto)
  - non ci serve un approccio "formale"
- Prodotto software: **usato da persone diverse da chi lo ha sviluppato**
  - è *software industriale* il cui costo è circa 10 volte il costo del corrispondente programma
  - dobbiamo avere un approccio ingegneristico allo sviluppo

3

3

Software: prodotto

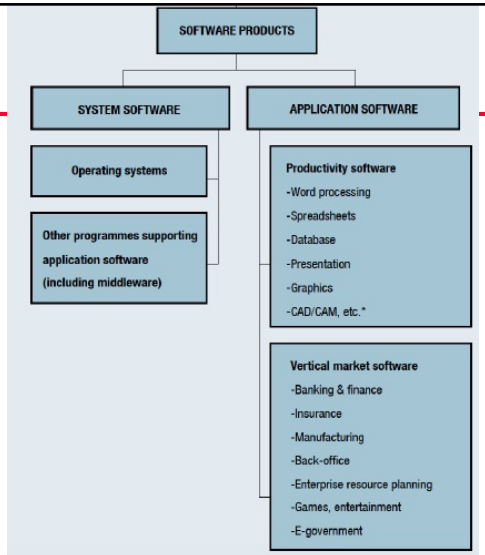
- Standard IEEE (Institute of Electrical and Electronics Engineers) 610.12-1990
  - *software*: the set of computer programs, procedures and possibly associate documentation and data pertaining to the operation of a computer system;
  - *software product*:
    1. The complete set of computer programs, procedures and possibly associate documentation and data designed for delivery to a user.
    2. Any of the individual items in 1.

... non solo il codice !

*... ma tutti gli 'artefatti' che lo accompagnano e che sono prodotti durante l'intero sviluppo: codice, documentazione, casi di test, specifiche di progetto, procedure di gestione, manuali utente ...*4

4

Tipi di Software



5

5

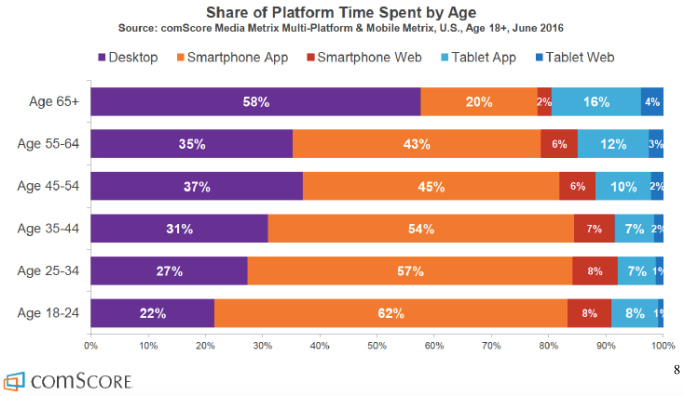
(nuove?) categorie di software

- Apps e software ecosystems
- Servizi software
- Nuovi strumenti di sviluppo
- Social software
- Scraping/mining big data
- Embedded software, IoT
- …

7

7

Distribuzione del tempo digitale per età

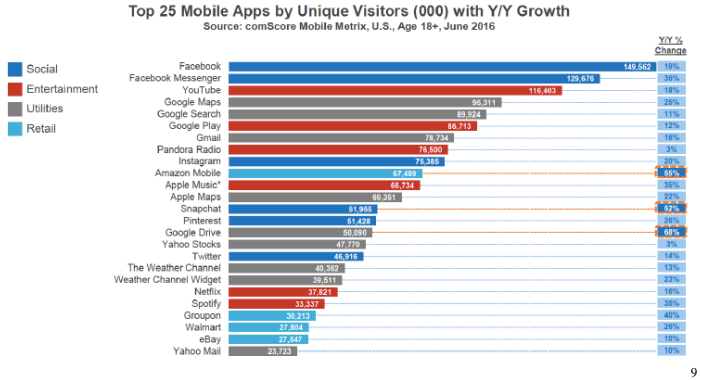


8

8

Le app più popolari

(in USA, 1 sem 2016, fonte ComScore)



9

9

Ecosistemi software

- Gli ecosistemi software sono mercati, in cui si vendono prodotti (es. AppStore o PlayStore) o componenti e servizi (es. Amazon Elastic Computing)
- La caratteristica principale è quella di una collezione di prodotti software, su piattaforma definita da un'azienda, che vengono sviluppati ed evolvono nello stesso ambiente
- Es. Appstore (al 2016): 100 miliardi di download, utili oltre 40miliardi$; 20 "grandi" sviluppatori incassano il 50% degli utili.

Which Country Has the Best Developers?

Ranked by Average Score Across All HackerRank Challenges

Rank	Country	Score Index	Rank	Country	Score Index
1	China	100.0	26	Netherlands	78.9
2	Russia	99.9	27	Chile	78.4
3	Poland	98.0	28	United States	78.0
4	Switzerland	97.9	29	United Kingdom	77.7
5	Hungary	93.9	30	Turkey	77.5
6	Japan	92.1	31	India	76.0
7	Taiwan	91.2	32	Ireland	75.9
8	France	91.2	33	Mexico	75.7
9	Czech Republic	90.7	34	Denmark	75.6
10	Italy	90.2	35	Israel	74.8
11	Ukraine	88.7	36	Norway	74.6
12	Bulgaria	87.2	37	Portugal	74.2
13	Singapore	87.1	38	Brazil	73.4
14	Germany	84.3	39	Argentina	72.1
15	Finland	84.3	40	Indonesia	71.8
16	Belgium	84.1	41	New Zealand	71.6
17	Hong Kong	83.6	42	Egypt	69.3
18	Spain	83.4	43	South Africa	68.3
19	Australia	83.2	44	Bangladesh	67.8
20	Romania	81.9	45	Colombia	66.0
21	Canada	81.7	46	Philippines	63.8
22	South Korea	81.7	47	Malaysia	61.8
23	Vietnam	81.1	48	Nigeria	61.3
24	Greece	80.8	49	Sri Lanka	60.4
25	Sweden	79.9	50	Pakistan	57.4

HackerRank

Prodotti Software

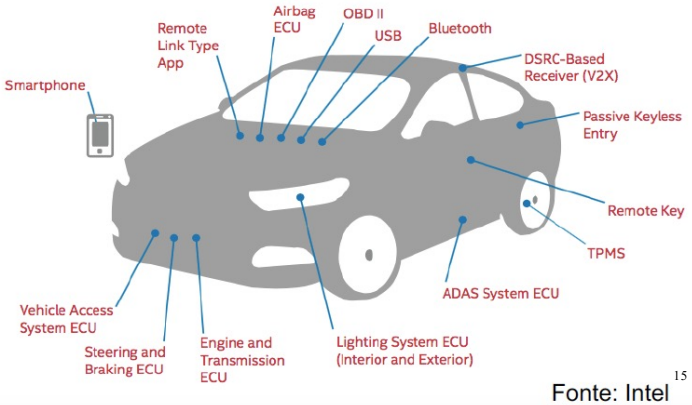
- Prodotti generici
  - sistemi stand-alone prodotti da una organizzazione e venduti a un mercato di massa
- Prodotti specifici
  - sistemi commissionati da uno specifico utente e sviluppati specificatamente per questo da un qualche contraente

La fetta maggiore della spesa è nei prodotti generici ma il maggior sforzo di sviluppo è nei prodotti specifici

Software as a service

Google: 2 miliardi di linee di codice
25.000 sviluppatori
45.000 commit al giorno
Chrome: 17.4 milioni di linee di codice
3.700 sviluppatori
380 commit al giorno

Software embedded ("nascosto")



Fonte: Intel

15

15

Social Software (Web 2.0)

- Software che supporta la "conversazione" di comunità di utenti. Es. Facebook, Twitter, LinkedIn, Instagram, Pinterest, ecc.
- "The term Social software is normally applied to a range of web-enabled software programs. The programs usually allow users to interact, share, and meet other users"
- Wiki, chat, forum, blog, ecc

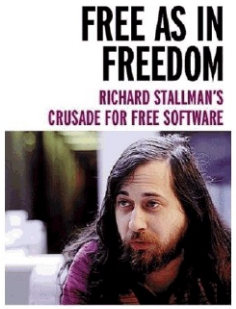
16

16

Software libero (non gratis!)

0. A program can be run for any purpose
1. A program can be studied and changed to adapt it to new needs
2. A program can be freely distributed
3. A program can be freely improved and these improvements can be freely distributed

Free Software Foundation



Richard Stallman
FSF founder

17

17

Videogiochi

- Sforzo tipico: 100÷500 anni/persona
- Team: di solito 50 ÷ 100 persone (Assassin Creeds 2009: 450 persone)
- Vendere un milione di copie è ok ma non eccellente

http://www.informationisbeautiful.net/visualizations/million-lines-of-code/

18

18

I sw sono sempre più grandi e costosi

*It would cost over **$1 billion** to develop REDHat 7.1 GNU/Linux distribution by conventional proprietary means in the U.S. (in year 2000 U.S. dollars).*

*Compare this to the **$600 million** estimate for Red Hat Linux version 6.2 (which had been released about one year earlier).*

*Red Hat Linux 7.1 includes over **30 million** physical source lines of code (SLOC), compared to well over **17 million SLOC** in version 6.2.*

Using the COCOMO cost model, this system is estimated to have required about 8,000 person-years of development time (as compared to 4,500 person-years to develop version 6.2).

*Red Hat Linux 7.1 represents over a **60% increase in size, effort, and traditional development costs** over Red Hat Linux 6.2. This is due to an increased number of mature and maturing open source / free software programs available worldwide (D.Wheeler, 2002)*

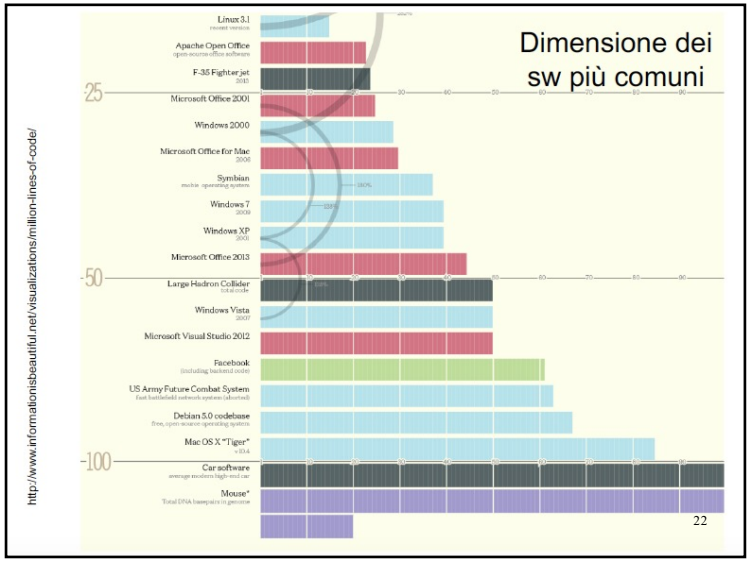
Contare le righe

Questa tabella conta quante linee di codice sorgente in Linux RedHat 7.1 sono scritte in vari linguaggi

Linguaggio	SLOC (%)
C	21461450 (71.18%)
C++	4575907 (15.18%)
ShellBourne	793238 (2.63%)
Lisp	722430 (2.40%)
Assembly	565536 (1.88%)
Perl	562900 (1.87%)
Fortran	493297 (1.64%)
Python	285050 (0.95%)
Tcl	213014 (0.71%)
Java	147285 (0.49%)
yacc/bison	122325 (0.41%)
Expect	103701 (0.34%)
lex/flex	41967 (0.14%)
awk/gawk	17431 (0.06%)
Objective-C	14645 (0.05%)
Ada	13200 (0.04%)
C shell	10753 (0.04%)
Pascal	4045 (0.01%)
sed	3940 (0.01%)

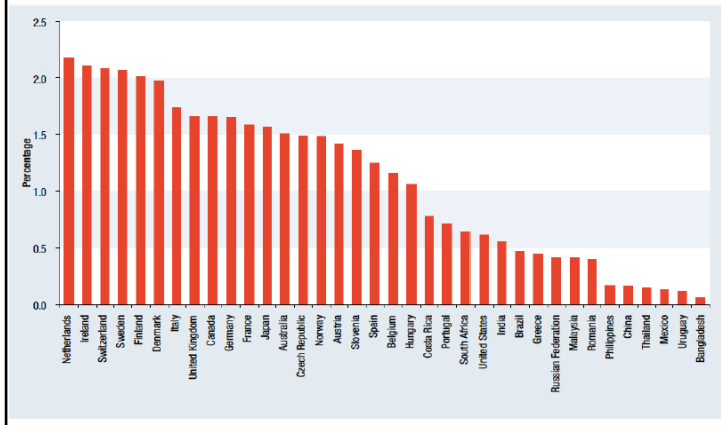
Alcune cifre

Prodotto	**SLOC** (righe di codice sorgente)
NASA Space Shuttle Flight Control	430K(shuttle) + 1.4M (ground)
Sun Solaris 1998-2000	7-8M
Microsoft Windows 3.1 (1992)	3M
Microsoft Windows 95	14M
Microsoft Windows 98	18M
Microsoft Windows NT (1992)	4M
Microsoft Windows NT5.0 (1998)	20M
RedHatLinux 6.2 (2000)	17M
MacOS 10.4 (2005)	86M
Linux kernel 4.2 (2016)	20.2M
Debian 7.0 (2012)	419M



Quanti producono software

(occupati a produrre sw o servizi, in percentuale sul totale degli occupati)



Source: UNCTAD, based on international and national sources. Total employment data sourced from WITSA.

23

23

I grandi produttori di software in Europa

(2015, source Truffle Report)

Rank	Company	Country of HQ location	Public	Software + Services 2014 (m€)	Total revenue 2014 (m€)	R&D employees 2014
1	SAP	DE		17 243.9	17 560.0	18908
2	Dassault Systemes	FR		2 076.6	2 346.7	5562
3	Sage	UK		1 539.5	1 620.5	1169
4	Hexagon	SE		1 442.3	2 622.4	3430
5	Wincor Nixdorf	DE		1 367.0	2 471.0	750
6	Assseco Group	PL		1 103.3	1 487.6	3606
7	Software AG	DE		935.6	957.8	968
8	DATEV	DE		790.7	843.5	1355
9	Wolters Kluwer	NL		740.2	3 660.0	2222
10	Misys	UK		639.5	639.5	1100
11	Micro Focus	UK		627.9	627.9	997
12	SWIFT	BE		596.8	628.0	493
13	Unit4	NL		516.0	516.0	1360
14	Visma	NO		464.7	851.7	730
15	Cegedim	FR		460.6	911.5	959
16	Sopra Steria	FR		445.4	3 370.0	800
17	Olik	SE		416.9	416.9	356
18	Avaloq	CH		411.6	411.6	300
19	Swisslog	CH		411.4	551.3	146
20	Northgate Information Solutions	UK		373.7	826.0	900
21	Fiducia & GAD	DE		369.3	458.4	220
22	Compugroup Holding	DE		366.8	515.1	1426
23	Murex	FR		366.0	368.0	366
24	Temenos	CH		352.6	352.6	441
25	Fidessa	UK		341.0	341.0	437
26	Gemalto	NL		333.9	2 465.0	1105
27	IFS	SE		333.4	333.4	333
28	ESET	SK		326.8	326.8	382
29	Zucchetti	IT		310.3	358.0	900
30	Schneider Electric	FR		304.1	1 713.0	924
31	Reply	IT		267.0	632.2	360

24

24

I salari d'ingresso dei big players (2016)

Azienda	Stipendio annuo medio in $ - junior	Bonus medio $	Totale $
Amazon	109.000	22.000	131.000
Apple	104.000	16.000	120.000
Google	86.000	20.000	106.000
Cisco	67.000	1.000	68.000
Oracle	67.000	-	67.000
Microsoft	58.000	9.000	67.000
Telefonica	45.000	4.000	49.000
Orange	48.000	-	48.000
IBM	48.000	-	48.000
SAP	44.000	4.000	48.000

25

http://money.cnn.com/2016/05/06/pf/jobs/tech-jobs-entry-level-pay-salary/index.html

25

Il sw è un prodotto industriale

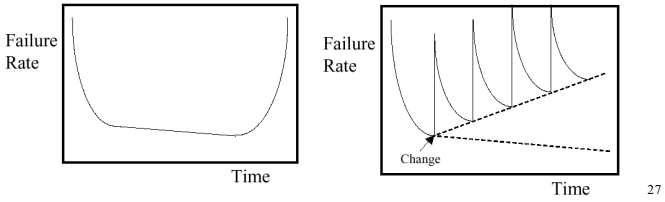
- Il software è sempre il prodotto di un processo di sviluppo, che inizia con un'idea e termina quando il software viene ritirato
- L'industria mondiale del sw cresce a tassi dal 5 al 10% annuo
- Il costo di sviluppo di un prodotto software tende a crescere in proporzione al quadrato delle sue dimensioni

26

26

Natura del Software

- Differenze con prodotti industriali classici
  - Intangibile
  - Malleabile
  - Ad alta intensità di lavoro umano
  - Spesso costruito ad hoc invece che assemblato
  - Manutenzione = cambiamento



Failure Curve for Hardware

Failure Curve for Software

27

27

Alcuni miti del Software

- Miti del management
  - Abbiamo i più moderni computer …
  - Se siamo in ritardo possiamo recuperare aumentando il numero di programmatori
- Miti del cliente
  - Un'affermazione generica degli scopi è sufficiente per cominciare a scrivere programmi
  - I mutamenti nei requisiti di un progetto si gestiscono facilmente grazie alla flessibilità del software
- Miti del programmatore
  - Una volta messo in opera il programma, il lavoro è finito
  - Il solo prodotto di un progetto concluso è il programma

28

28

Problemi della produzione software: Costi

- Il software ha costi elevati
  - Sono i costi delle risorse usate: ore lavoro (manpower), hardware, software e risorse di supporto.
  - Il manpower è dominante !
    - » Il costo è espresso in mesi/uomo
- Il testing impiega fino al 50% dei costi di sviluppo
- La manutenzione costa più dello sviluppo
  - Per sistemi che rimangono a lungo in esercizio i costi di manutenzione possono essere svariate volte il costo di produzione

29

29

Problemi della produzione software: Ritardi

- Molti progetti sono "runaway" (in ritardo o fuori dal budget e lo schedule stimato)
- Esempio: US Air Force sistema di comando e controllo
  - stima iniziale della azienda vincitrice per la fornitura: 400.000$,
  - costo successivamente rinegoziato a 700.000$ e poi a 2.500.000$ …
  - costo finale 3.200.000$, maggiore di circa 10 volte la stima iniziale !
  - … e con notevole ritardo rispetto alla stima iniziale

30

30

Prodotti, sistemi, servizi

- Prodotti generici (OTS: off the shelf)
  - Prodotti creati da qualche produttore di software e venduti sul mercato a più (tanti) clienti
  - Es.: videogioco
- Sistemi commissionati ("customizzati" "specifici")
  - Sistemi commissionati da un cliente specifico e sviluppati apposta da un qualche fornitore
  - Es.: portale del ministero
- Servizi in perpetuo sviluppo ("continuous development")
  - Sistemi che offrono servizi 24/7 in continuo cambiamento
  - Es. Facebook, Amazon

31

31

Requisito e Feature

- Requisito software: funzione o proprietà **controllabile** (testabile) che deve possedere l'implementazione di un prodotto software. E' importante per il cliente
- Feature software: insieme di funzioni che permettono di usare un prodotto software in un servizio o business.

  E' importante per il fornitore

32

32

Esempio

- Feature: carrello per negozio elettronico
- Requisiti (funzionali) di un servizio di commercio elettronico:
  - l'utente deve poter registrarsi,
  - aggiungere o togliere elementi al carrello,
  - specificare indirizzi alternativi,
  - pagare
- Requisiti (non funzionali) di un servizio di commercio elettronico:
  - Per registrarsi l'utente non deve impiegare più di 3 click
  - Il sistema non deve consentire ad altri utenti l'accesso ai dati della carta di credito di un utente

33

33

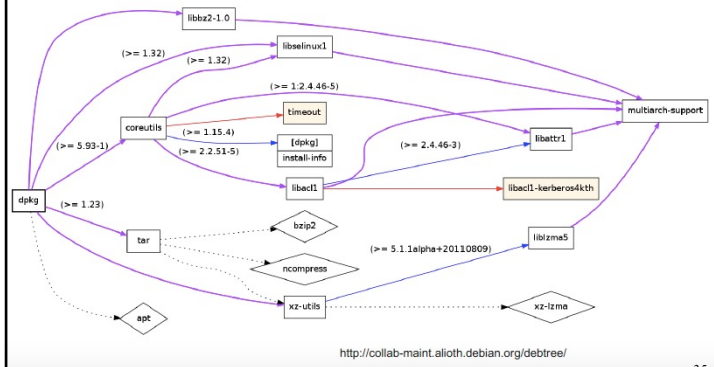
Dipendenze

- Ogni prodotto sw dipende da altri prodotti sw, ...che a loro volta dipendono da altri sw
- Associamo a ciascun prodotto o sistema software un grafo di dipendenze
- I nodi del grafo delle dipendenze sono pacchetti software (es librerie) in diverse versioni

34

34

Esempio



35

https://softvis.wordpress.com/tools/

35

Discussione

Perché è importante conoscere le dipendenze?
Quali tipi di dipendenze esistono?

36

36

«Nulla è permanente tranne che il cambiamento»

«L'unica costante è il cambiamento»

Eraclito

37

37

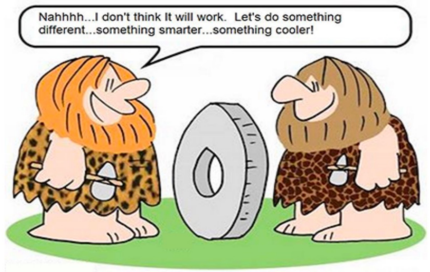
Tipi di dipendenze

- **Dependencies we control**
  - This is code written and owned by us or our organization.
- **Dependencies we DON'T control**
  - This is code written by a third party vendor or open-source software community.
- **Dependencies once removed**
  - These are the code dependencies our third-party code dependencies depend upon.

38

38

Why third-party code dependencies are good



- Third-party code — in the form of libraries — allows you to quickly implement those commoditized features of your app, so you can stay focused on your "secret sauce." 39

39

Why third-party code dependencies are bad

What if one or more of those third-party libraries changes drastically, or disappears, or breaks?

40

40

Balancing the good and the bad

https://www.freecodecamp.org/news/code-dependencies-are-the-devil-35ed28b556d/ 41

41

Il SW è speciale

- **È invisibile e intangibile**
- **Ogni prodotto ha molte dipendenze**
- È facilmente duplicabile e distribuibile su rete
- In Europa non è brevettabile (ma protetto)
- Il software di consumo non è garantito
- Viene acquisito su licenza
  - Proprietaria (normale, shareware, freeware)
  - Public domain
  - Open source

42

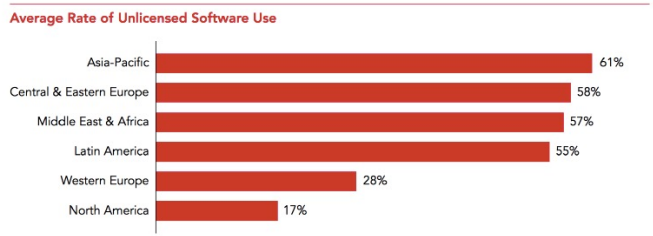
42

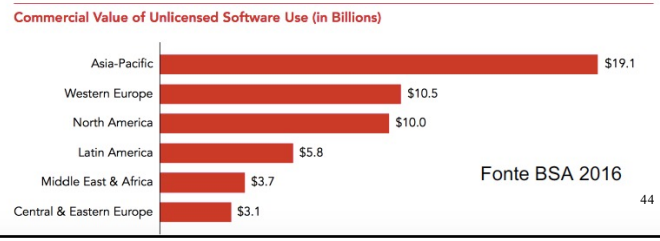
Task

- Analizzare le differenze tra licenza
  - Proprietaria (normale, shareware, freeware)
  - Public domain
  - Open source

43

43





44

44

Protezione legale del sw

- **Protezione dell'autore**: Il software è un'opera dell'ingegno: chi lo produce è un autore che ha diritto ad un compenso

  Copiare software abusivamente è illegale (anche se non lo si fa per profitto) e in Italia costituisce un reato penale:

  La legge italiana 248/2000 punisce col carcere da 6 mesi a 3 anni chi duplica abusivamente software
- Per informazioni sulla brevettabilità del software negli USA: http://www.softwarepatent.com

45

45

SIAE: pubblico registro del SW

- Possono essere registrati i sw che rispettino requisiti di originalità e creatività tali da poter essere identificati come opere dell'ingegno.
- è possibile registrare tutti gli atti che trasferiscono in tutto o in parte diritti di utilizzazione economica relativi a programmi per i quali sia già avvenuta la registrazione
- Per registrare un programma, il richiedente deve trasmettere a SIAE una "dichiarazione" e una "descrizione" oltre, naturalmente, ad un esemplare del programma da depositare registrato su supporto digitale non riscrivibile

46

46

La garanzia del software

Protezione del compratore: Quale protezione ha il compratore da difetti del prodotto?

Nel software di consumo in teoria NON c'è alcuna garanzia.

Il software viene venduto "così com'è", e se ci sono difetti il fabbricante non se ne fa carico:

lo dice il contratto che si visualizza quendo si usa per la prima volta un'applicazione

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1.GRANT OF LICENSE. The SOFTWARE PRODUCT is licensed as follows:
* Installation and Use. Microsoft grants you the right to install and use copies of the SOFTWARE PRODUCT on your computers running validly licensed copies of the operating system for which the SOFTWARE PRODUCT was designed [e.g., Windows(r) 95; Windows NT(r), Windows 3.x, Macintosh, etc.].
* Backup Copies. You may also make copies of the SOFTWARE PRODUCT as may be necessary for backup and archival purposes.
* Components. Certain software components of the SOFTWARE PRODUCT are subject to the following additional provisions:
2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.
* Distribution. You may not distribute copies of the SOFTWARE PRODUCT to third parties.
* Prohibition on Reverse Engineering, Decompilation, and Disassembly.
4. COPYRIGHT. All title, including but not limited to copyrights, in and to the SOFTWARE PRODUCT and any copies thereof are owned by Microsoft or its suppliers. All rights not expressly granted are reserved by Microsoft.
8. NO WARRANTIES. To the maximum extent permitted by applicable law, Microsoft and its suppliers provide the SOFTWARE PRODUCT and any (if any) Support Services related to the SOFTWARE PRODUCT AS IS AND WITH ALL FAULTS, and hereby disclaim all warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties or conditions of merchantability, of fitness for a particular purpose, of lack of viruses, of accuracy or completeness of responses, of results, and of lack of negligence or lack of workmanlike effort, all with regard to the SOFTWARE PRODUCT, and the provision of or failure to provide Support Services.
ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT, WITH REGARD TO THE SOFTWARE PRODUCT.
THE ENTIRE RISK AS TO THE QUALITY OF OR ARISING OUT OF USE OR PERFORMANCE OF THE SOFTWARE PRODUCT AND SUPPORT SERVICES, IF ANY, REMAINS WITH YOU.

47

47

Garanzie sul software

- La verifica garantisce l'aderenza ad una *specifica*
- La validazione garantisce l'accettazione da parte del cliente
- La certificazione garantisce l'aderenza a specifiche definite dalla legge
- NB: il sw commerciale di solito viene venduto senza garanzie ("*as is*")

48

48

Alcuni dati

- Numero di difetti (fault) rilevati durante l'esercizio
  - I peggiori sistemi militari: 55 faults/KLoC
  - I migliori sistemi militari: 5 faults/KLoC
  - Prodotti ottenuti con sviluppo agile (XP): 1.4 faults/KLoC
  - Apache web server (open source): 0.5 faults/KLoC
  - NASA Space shuttle: 0.1 faults/KLoC

www.easterbrook.ca/steve/?p=1366

49

49

Problemi della produzione software: Abbandoni

- Molti progetti vengono abbandonati …
- Una azienda nel settore della grande distribuzione di prodotti aveva richiesto un sistema che, stima iniziale, sarebbe stato sviluppato in 9 mesi al prezzo di 250.000 $
  - Due anni dopo, e dopo una spesa di 2.500.000 $, il lavoro non era stato ancora completato e fu stimato che erano necessari altri 3.600.000 $ (!!)
  - Il progetto fu abbandonato !

50

50

Problemi della produzione software: Affidabilità

- Il software è (spesso) inaffidabile
  - Molti malfunzionamenti sono rilevati durante l'operatività del sistema
  - Da un' analisi del ministero della difesa USA risulta che più del 70 % di tutti i malfunzionamenti sono dovuti al software (in sistemi con complicati e sofisticati apparati meccanici, elettrici, idraulici ….)
- …. l' Arianne 5 …
- …. la sonda MARINER …..

51

51

Necessità di un approccio ingegneristico

- Necessità di applicare principi ingegneristici alla produzione software per sviluppare:
  - il *giusto* prodotto
  - al *giusto* costo
  - nel tempo *giusto*
  - con la *giusta* qualità

52

52

Ingegneria del Software: Scopo

- Riguarda la costruzione di software:
  - di grandi dimensioni
  - di notevole complessità
  - sviluppati tramite lavoro di gruppo
- Progetti software di questo tipo hanno tipicamente:
  - versioni multiple
  - lunga durata
  - frequenti cambiamenti
    - » eliminazione di difetti
    - » adattamento a nuovi ambienti
    - » miglioramenti e nuove funzionalità

Multi-person construction of multi-version software (Parnas)

53

53

Ingegneria del Software: Contesto

- Software Engineering ⊂ System Engineering
  - La maggior parte del SW è collocata all' interno di un "sistema" misto HW/ SW
  - Lobiettivo finale di chi produce è creare tale sistema che soddisfa globalmente i requisiti dellutente
  - Coinvolgimento nella definizione dei requisiti del sistema
- Conoscenza del dominio applicativo
  - È essenziale per un efficace sviluppo del SW
  - Il SW è utile quando riesce a condensare nei suoi algoritmi la conoscenza del dominio applicativo
  - Altrimenti inutile o dannoso
    - » Per esempio, sistema di controllo di un aeroplano

54

54

ll'Ingegneria

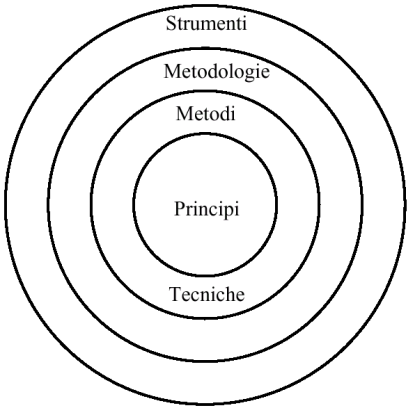
Ingegneria del Software: Definizioni

- Applicazione di una strategia sistematica, disciplinata e misurabile allo *sviluppo, esercizio e manutenzione* del software (programmi, procedure, regole e associata documentazione, dati)
  - IEEE (Institute of Electrical and Electronics Engineers) Standard Glossary of Software Engineering Terminology, 1993
- La disciplina tecnologica e manageriale che riguarda la produzione sistematica e la manutenzione dei prodotti software che vengono sviluppati e modificati entro i tempi e i costi preventivati
  - D. Farley

55

55

Ingegneria del Software: Fondamenti



L'ingegneria del software si occupa dei *metodi*, delle *metodologie*, dei *processi* e degli *strumenti* per la gestione professionale (sviluppo, manutenzione, ritiro) del software

56

56

Ingegneria del Software: Principi

- **Rigore:** concetto primitivo (precisione, accuratezza)
- **Formalità:** oltre il rigore (fondamento matematico)
- **Separazione di aspetti diversi**: affrontare separatamente i vari aspetti di un problema complesso
- **Modularità**: suddividere un sistema complesso in parti più semplici
- **Astrazione**: si identificano gli aspetti cruciali in un certo istante ignorando gli altri
- **Anticipazione del cambiamento**: la progettazione deve favorire l'evoluzione del SW
- **Generalità**: tentare di risolvere il problema nella sua accezione più generale
- **Incrementalità**: lavorare per passi successivi

57

57

Metodi e Metodologie

- **Metodo (o tecnica)**: procedimento generale per risolvere classi di problemi specificati di volta in volta
  - linee guida o regole che governano le attività
  - il metodo dei minimi quadrati, il metodo di Montecarlo, il metodo di Newton, come fare il brodo di carne, come fare il lesso, ...
- **Metodologia**: insieme di principi, di metodi, degli elementi di cui una o più discipline si servono per garantire la correttezza e l'efficacia del proprio procedere
  - e.g. la metodologia della macerazione carbonica permette di ottenere vini novelli, freschi, profumati, …

58

58

ll'Ingegneria

Strumenti, Procedure, Paradigmi

- **Strumento (tool):** un artefatto, un sistema per fare qualcosa in modo **migliore**
  - e.g., il frullatore per fare la maionese, un cavatappi per aprire una bottiglia
  - supporti SW pratici all'applicazione
- **Procedura**: una combinazione di **strumenti** e **metodi** che assieme permettono di produrre un certo prodotto
  - e.g., la ricetta della Saker Torte: montare a neve le chiare di 4 uova ...
- **Paradigma**: un particolare approccio o filosofia per fare qualcosa
  - e.g., è lo stile della cucina, noi riconosciamo la cucina Francese, quella Italiana, quella Cinese …

Processo

Un processo è un particolare *metodo* per *fare qualcosa* costituito da una **sequenza di passi** che coinvolgono **attività, vincoli e risorse** (Pfleeger)

Processo: una particolare metodologia operativa che nella tecnica definisce le singole operazioni fondamentali per ottenere un prodotto industriale (Zingarelli)

Processo software: **un metodo per sviluppare del software** (Sommerville)

Processo software

- Insieme **organizzato** di *attività* che sovrintendono alla costruzione del prodotto da parte del team di sviluppo utilizzando metodi, tecniche, metodologie e strumenti.
- È suddiviso in varie *fasi* secondo uno schema di riferimento (*il ciclo di vita del software*)
- Descritto da un *modello*: informale, semi-formale o formale (*maturità del processo*)

Processo software: Standard IEEE 610.12-1990

- *Sofware development process*: The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use.
- Note: These activities may overlap or be performed iteratively.
- See also: incremental development; rapid prototyping, spiral model, waterfall model.

CASE (Computer-Aided Software Engineering)

- Sistemi Software che intendono fornire un supporto automatico per le attività di un processo software
- Upper-CASE
  - Strumenti che supportano le attività delle fasi di analisi e specifica dei requisiti e progettazione di un processo sosftware. Includono editor grafici per sviluppare modelli di sistema, dizionari dei dati per gestire entità del progetto
- Lower-CASE
  - Strumenti che supportano le attività delle fasi finali del processo, come programming, testing e debugging. Includono generatori di graphical UI per la costruzione di interfacce utente, debuggers per supportare la ricerca di program fault, traduttori automatici per generare nuove versioni di un programma

63

Info

- **Newsgroup**: nntp://comp.software-eng
- **Siti importanti**:
  - www.swebok.org (SWE Body of Knowledge)
  - www.ieee.org (per gli standard IEEE del software)
  - www.w3c.org (per gli standard del software Web)
  - www.omg.org (per gli standard del software a oggetti)
  - www.oasis-open.org (per gli standard del software business)
  - www.softwarehistory.org
- **Blog e altro**
  - www.joelonsoftware.com
  - www.vWorker.com (www.rentacoder.com)
  - best-practice-software-engineering.blogspot.com
- **Gruppi linkedin**
  - Software testing and quality assurance
  - Software as a service

64

Warning!

- Questo corso si supera facilmente
  - avendo i prerequisiti,
  - seguendo le lezioni e
  - facendo quanto richiesto;
- è difficile da superare se
  - non si frequentano le lezioni e/o
  - non si hanno i prerequisiti
- Copiare (i compiti scritti, il progetto, i task…) è vietato e quando si viene scoperti l'esame viene annullato e diventa "molto difficile" da superare

65