

Ingegneria del Software

Introduzione

Software: Prodotto e Processo

Ingegneria del Software: Terminologia

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

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 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 approccio formale allo sviluppo

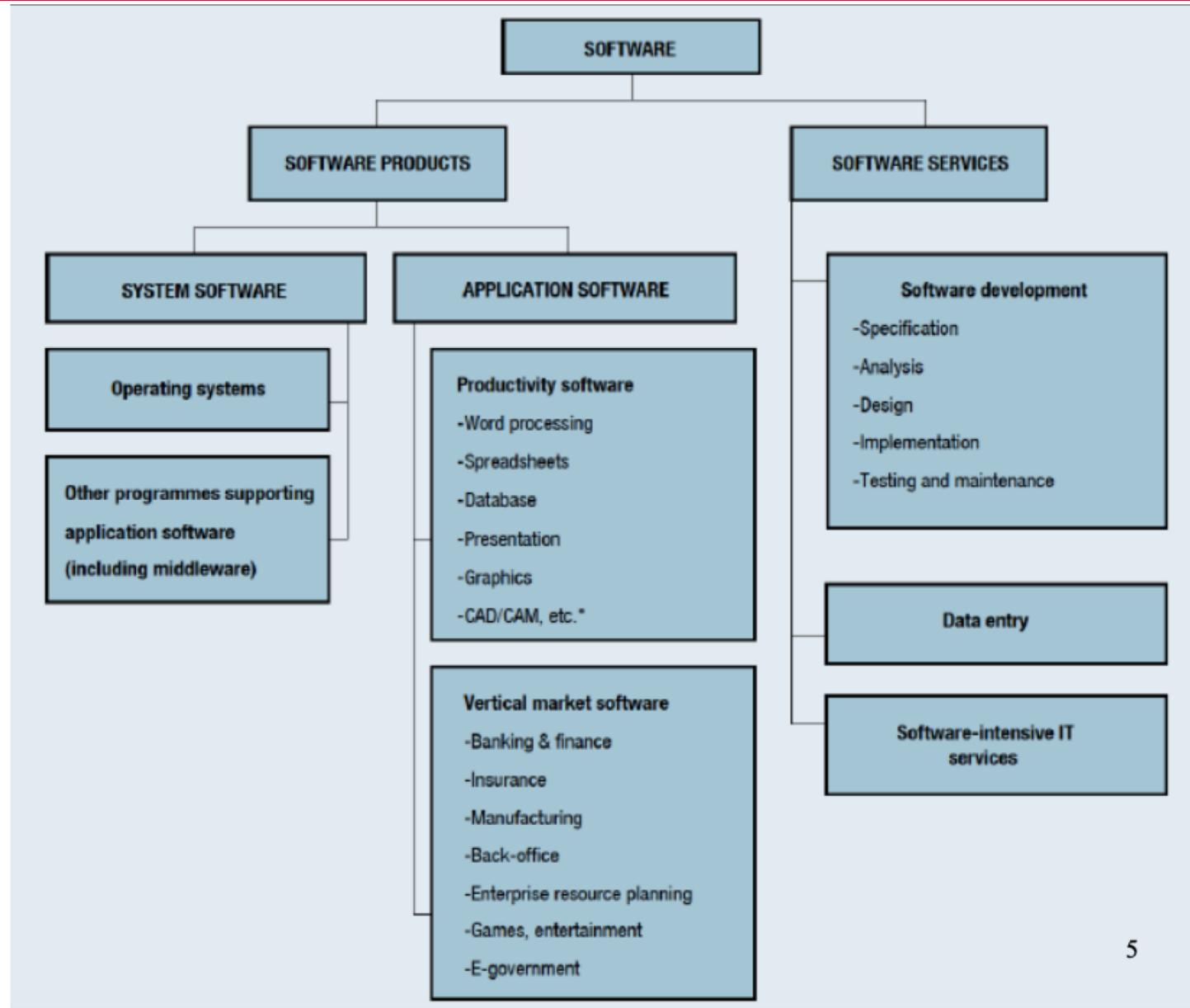
Software: prodotto (2)

- ◆ 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

Tipi di Software



Software

- ◆ **Prodotto** invisibile, intangibile, facilmente duplicabile ma costosissimo: opera dell'ingegno protetta dalle leggi
- ◆ **Servizio**; ha un'interfaccia e si basa su una infrastruttura

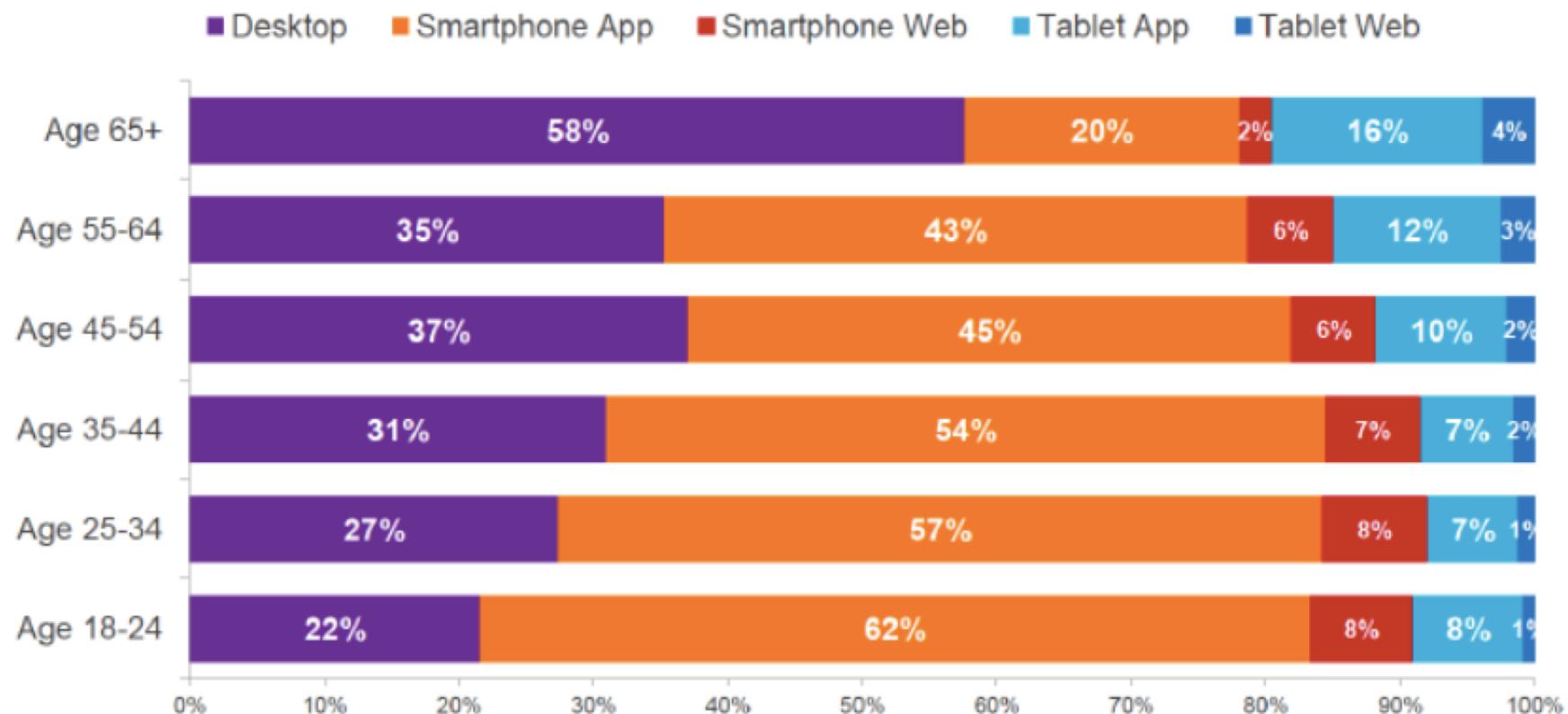
(nuove?) categorie di software

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

Distribuzione del tempo digitale per età

Share of Platform Time Spent by Age

Source: comScore Media Metrix Multi-Platform & Mobile Metrix, U.S., Age 18+, June 2016



Le app più popolari

Most Popular Apps 2020 (Global)

App	Downloads 2020
TikTok	850 million
WhatsApp	600 million
Facebook	540 million
Instagram	503 million
Zoom	477 million
Messenger	404 million
Snapchat	281 million
Telegram	256 million
Google Meet	254 million
Netflix	223 million

Source: [Apptopia](#)

Le app più popolari

Most Popular Shopping Apps 2020

App	Downloads 2020
Amazon	169 million
Shopee	139 million
Wish	138 million
Shein	122 million
Pinduoduo	90 million
AliExpress	79 million
Flipkart	72 million
Lazada	69 million
Mercado Libre	56 million
eBay	51 million

Source: [Apptopia](#)

Le app più popolari

Most Popular Education Apps 2020

App	Downloads 2020
Google Classroom	128 million
Duolingo	61 million
Photomath	48 million
Brainly	42 million
Cake	36 million
Kahoot!	25 million
U-Dictionary	23 million
Simply Piano	23 million
PictureThis	22 million
Qanda	16 million

Source: [Apptopia](#)

Ecosistemi software

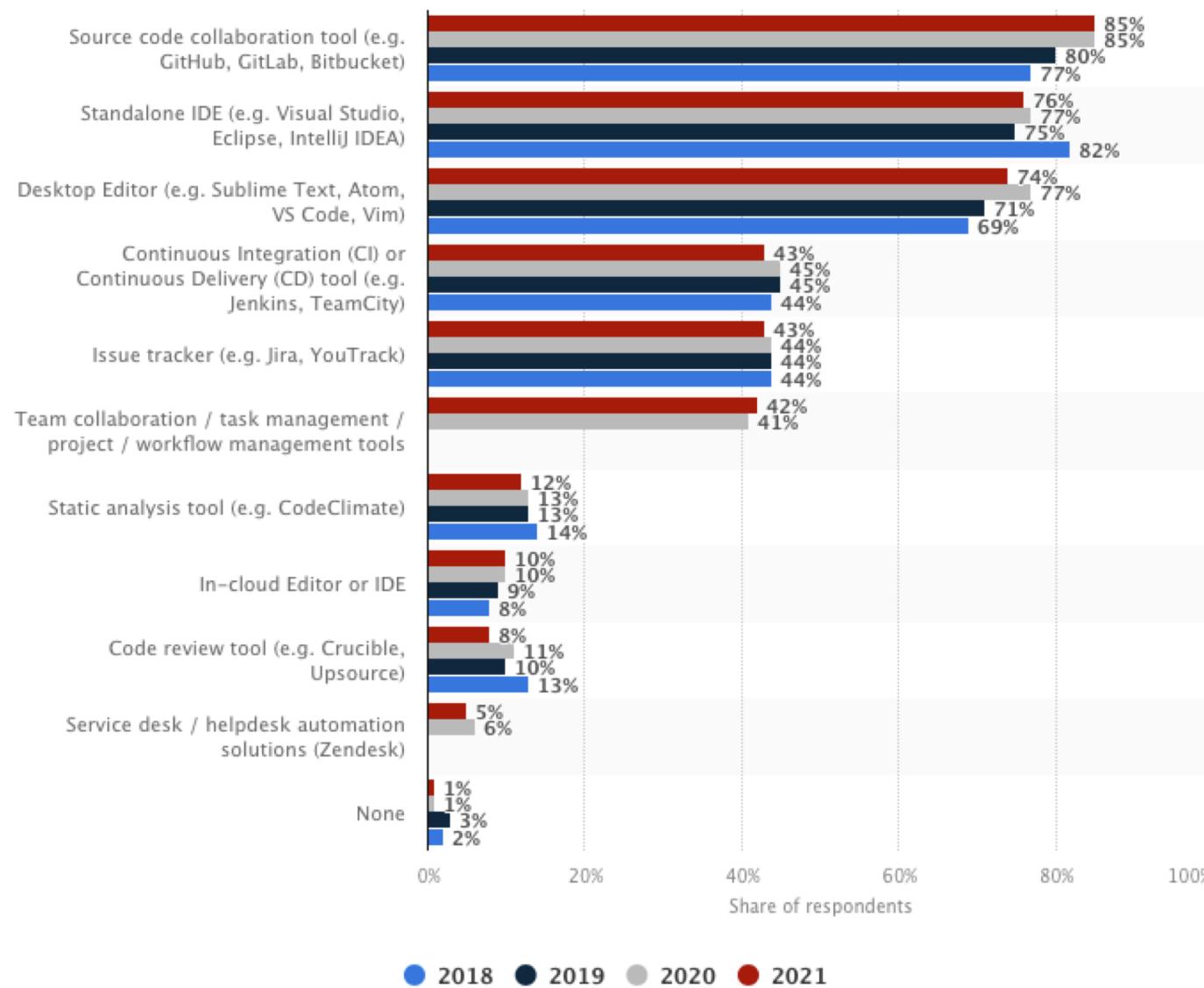
- ◆ Gli ecosistemi software sono mercati, in cui si vendono prodotti (es. AppStore o PlayStore) o componenti e servizi (es. Amazon Elastic Computing)
- ◆ La caratteristica principale è quella di una collezione di prodotti software, su piattaforma definita da un'azienda, che vengono sviluppati ed evolvono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

Prodotti Software

- ◆ Prodotti generici
 - sistemi stand-alone prodotti da un'organizzazione e venduti ad 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

Programming/development tools used by software developers worldwide from 2018 to 2021



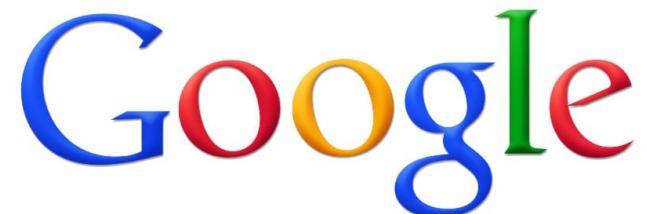
Software as Service

- ◆ Google:

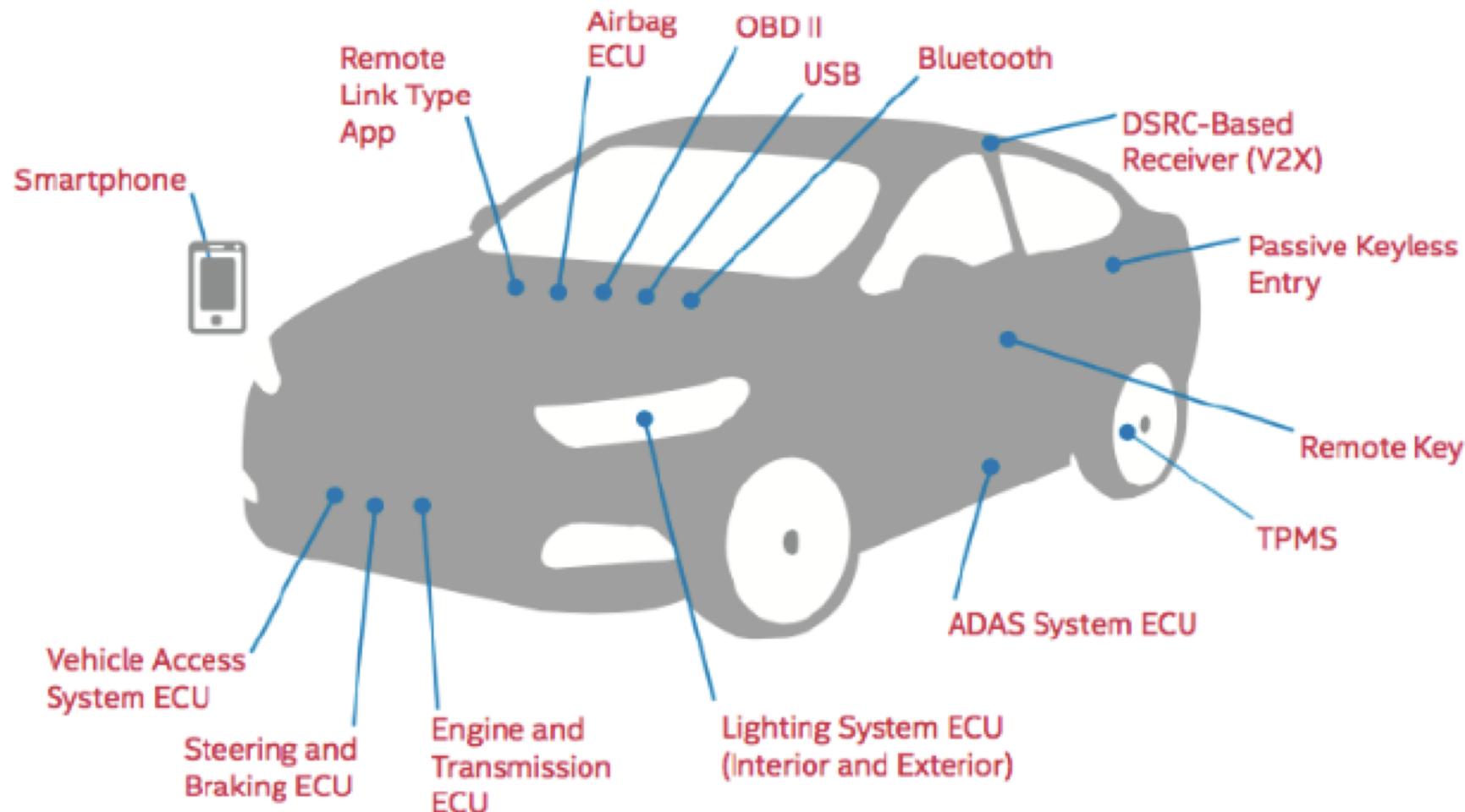
- 2 miliardi di LOC
- 25.000 sviluppatori
- 45.000 commit al giorno

- ◆ Chrome

- 17.4 milioni di LOC
- 3.700 sviluppatori
- 380 commit al giorno



Software embedded ("nascosto")



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, ...

Software libero (non gratis!)

- ◆ There are four essential freedoms of free software:
 - 0.The freedom to run the program as they want to, for any purpose.
 - 1.The freedom to study how the program works, and change it so it does what you want. Must have access to the source code.
 - 2.The freedom to redistribute – to help your neighbor.
 - 3.The freedom to redistribute copies of modified versions. By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.
- ◆ Free Software Foundation (Richard Stalmann)

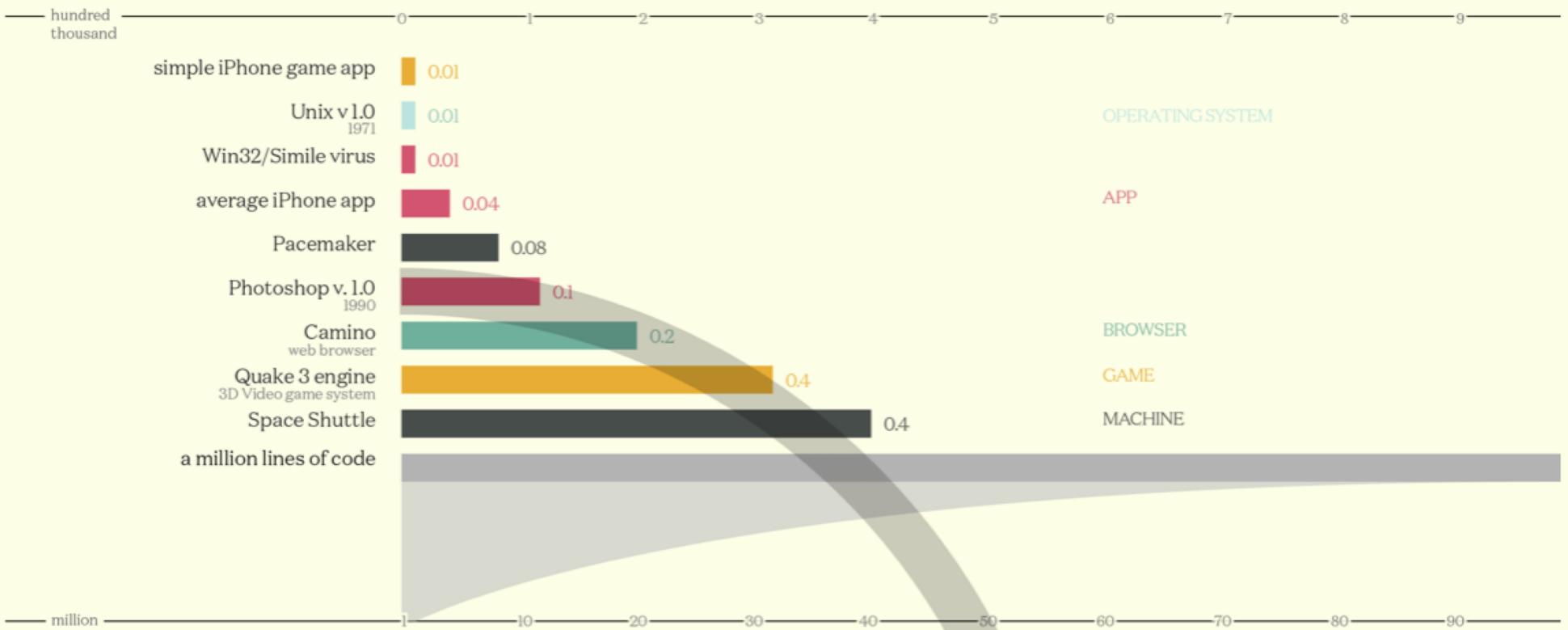
I sw sono sempre più grandi e costosi

- ◆ *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)*

Alcune cifre

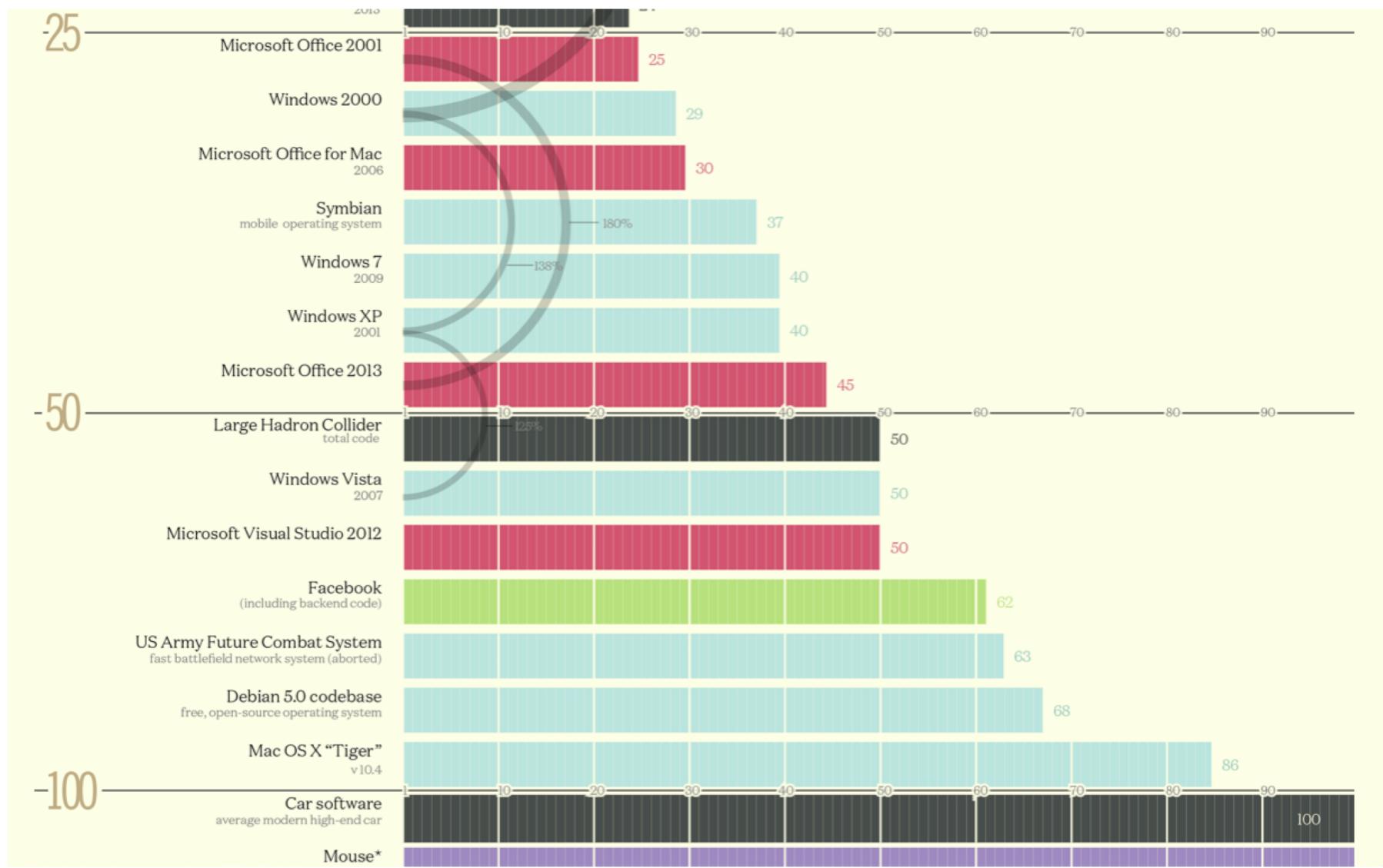
Codebases

Millions of lines of code



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

Alcune cifre



I grandi produttori software

<https://www.news.srl/principali-aziende-software-al-mondo/>

N	Organizzazione	Vendite
1	microsoft	\$ 97.58
2	Oracle	\$ 38.8
3	LINFA	\$ 23.3
4	Symantec	\$ 6.6
5	VMware	\$6
6	HCL Technologies	\$ 5.2
7	Fiserv	\$ 5.1
8	Intuit	\$ 4.6
9	Amadeus IT Group	\$ 4.5
10	CA Technologies	\$ 4.3

N	Organizzazione	Valore di mercato
1	microsoft	1.359 miliardi di dollari (\$)
2	Adobe Inc. –	170,4 miliardi di dollari (\$)
3	Oracle	167 miliardi di dollari (\$)
4	SAP	142,4 miliardi di dollari (\$)
5	Salesforce	144,9 miliardi di dollari (\$)
6	Intuit	70,3 miliardi di dollari (\$)
7	Fiserv	69,7 miliardi di dollari (\$)
8	ServiceNow	67 miliardi di dollari (\$)
9	VMware	55 miliardi di dollari (\$)
10	Dassault Systèmes	37,8 miliardi di dollari (\$)

<https://it.ripleybelieves.com/which-are-biggest-software-companies-operating-in-world-today-951>

Aziende software e informatiche (Italia)

RAGIONE SOCIALE	FATTURATO 2019 (000 eur)	2019/2018 (%)	MOL 2019 (%)
UNICREDIT SERVICES S.C.P.A.	1.973.236	-13,6	28,62
ACCENTURE S.P.A.	1.826.008	11,9	9,45
IBM ITALIA S.P.A.	1.749.588	-1,5	15,48
TECH DATA ITALIA SRL	1.163.366	0,1	0,66
NEXI PAYMENTS S.P.A.	1.119.827	5,8	n.d.
ENGINEERING S.P.A.	878.760	2,8	9,93
S.A.P. ITALIA S.P.A.	582.697	3,5	8,57
SOGEI S.P.A.	576.002	9,1	13,20
REPLY S.P.A.	534.007	-5,6	2,19
MICROSOFT S.R.L.	507.333	77,0	9,83
DELL S.P.A.	500.840	10,2	2,53
LOTTOITALIA S.R.L.	488.968	1,6	81,01
ALMAVIVA S.P.A.	483.077	17,8	12,86
GSS GENERALI SHARED SERVICES S.C.A.R.L.	444.717	-1,3	15,25
SIA S.P.A.	443.836	7,7	32,29
NTT DATA ITALIA S.P.A.	424.609	17,1	-2,04

I salari di 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

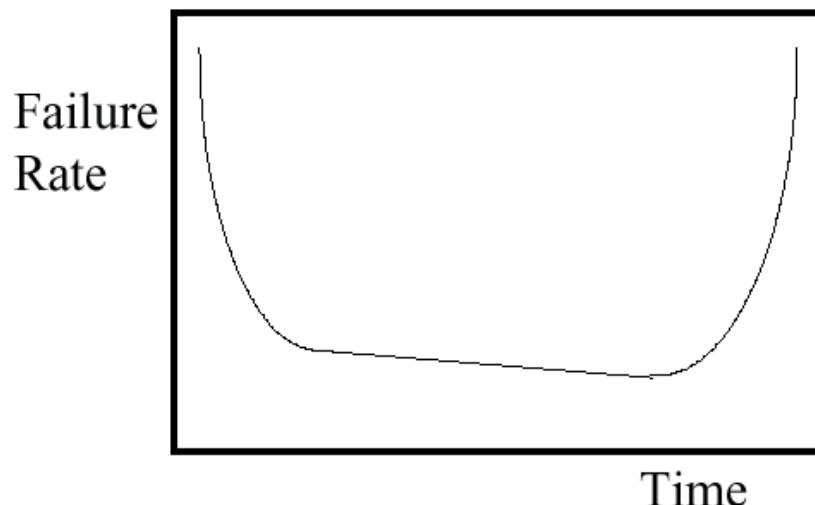
Il sw è un prodotto industriale

- ◆ 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

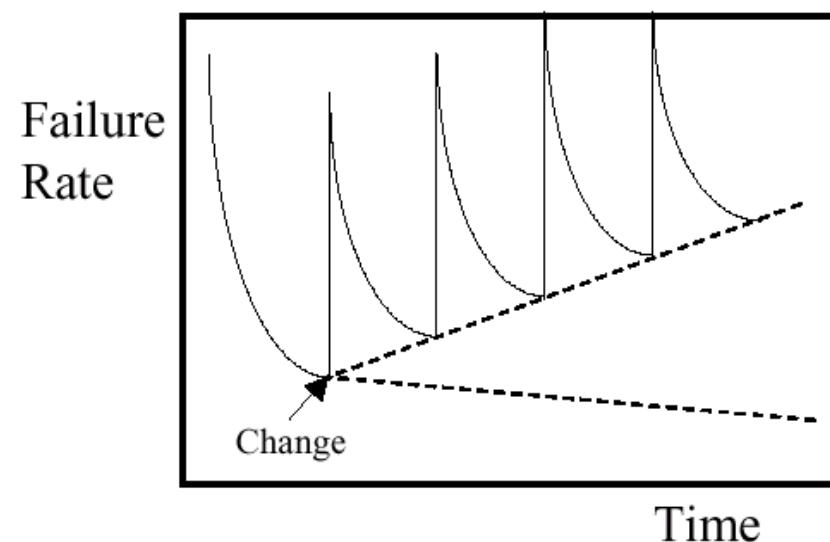
Natura del Software

◆ 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

Alcuni miti del Software (1)

◆ Miti del management

- Abbiamo i più moderni computer ...
- Se siamo in ritardo possiamo recuperare aumentando il numero di programmatore

◆ 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

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

Problemi della produzione software: Ritardi

- ◆ Ritardi nelle consegne e aumenti dei costi stimati
 - Esempio: US Air Force sistema di comando e controllo (da un report 1981):
 - » stima iniziale della azienda vincitrice per la fornitura: 400.000\$,
 - » costo successivamente rinegoziato:
 - ◆ a 700.000\$,
 - ◆ poi a 2.500.000 \$
 - ◆ costo finale 3.200.000\$.
 - » ... costo finale maggiore di circa 10 volte la stima iniziale !!
 - » ... e con notevole ritardo rispetto alla stima iniziale

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

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

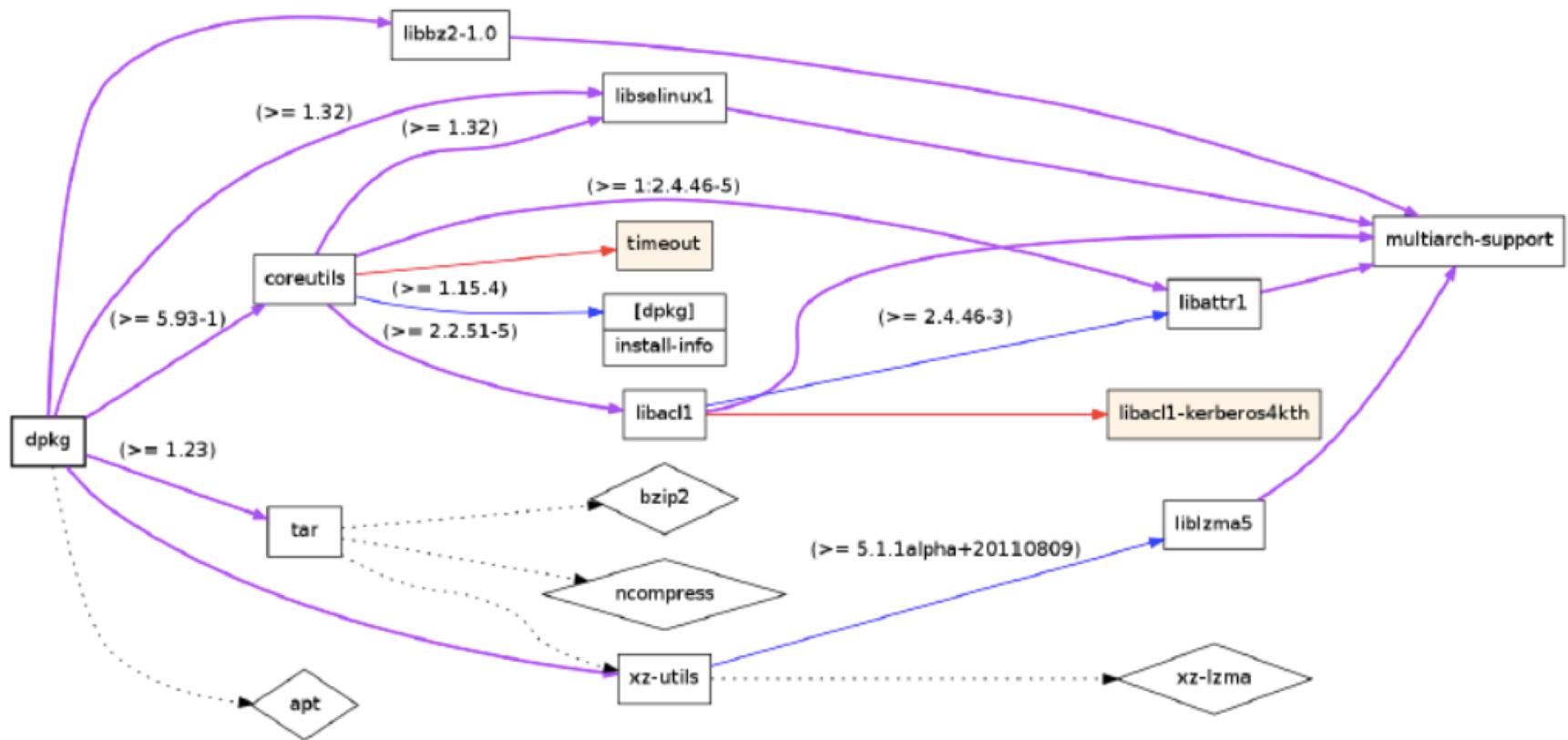
Esempio

- ◆ Feature: carrello per negozio elettronico
- ◆ Requisiti di un servizio di commercio elettronico: l'utente deve poter registrarsi, aggiungere o togliere elementi al carrello, specificare indirizzi alternativi, pagare

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

Esempio



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

Discussione

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

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 o open-source software community.
- ◆ Dependencies once removed
 - These are the code dependencies our third- party code dependencies depend upon.

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 acquistato su licenza
 - Proprietaria (normale, shareware, freeware)
 - Public domain
 - Open source

TASK

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

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>

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

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:

Io dice il contratto che si visualizza
quando 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.

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*”)

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

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

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)

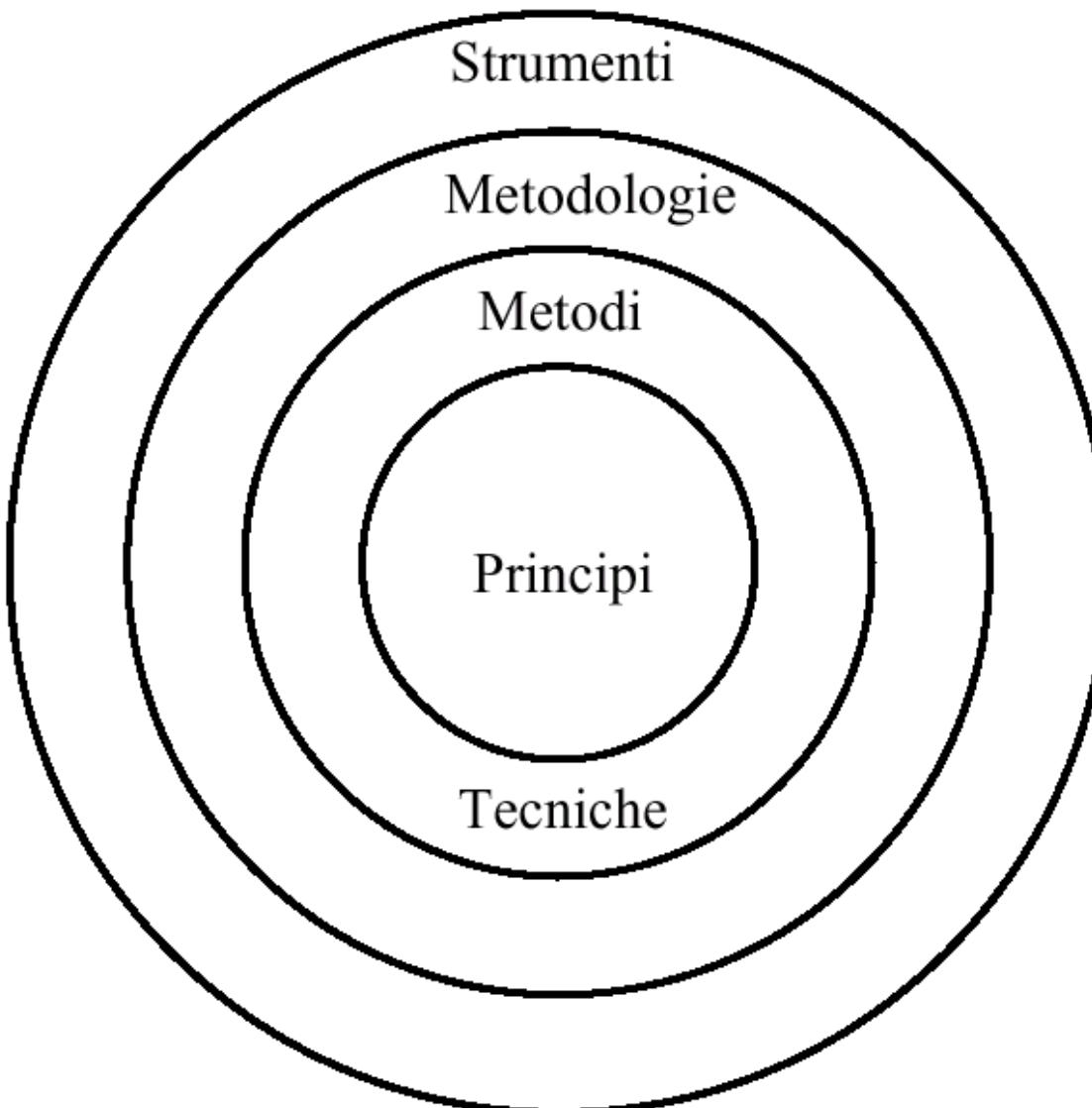
Ingegneria del Software: Contesto

- ◆ Software Engineering ⊂ System Engineering
 - La maggior parte del SW è collocata all'interno di un “sistema” misto HW/ SW
 - L'obiettivo finale di chi produce è creare tale sistema che soddisfa globalmente i requisiti dell'utente
 - 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 es. sistema di controllo di un aeroplano

Ingegneria del Software: Definizioni

- ◆ Applicazione di una strategia sistematica, disciplinata e misurabile allo sviluppo, esercizio e manutenzione del software (programmi, procedure, regole e documentazione associata, 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

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

Ingegneria del Software: Principi

◆ Rigore e formalità

- Rigore: concetto primitivo (precisione, accuratezza)
- Formalità: oltre il rigore. Fondamento matematico.
 - » In ogni fase del processo di produzione del SW bisogna definire il livello di rigore/ formalità necessario

◆ Separazione di aspetti diversi: affrontare separatamente i vari lati di un problema complesso, per esempio:

- rispetto al tempo
- rispetto a diverse proprietà del sistema
- rispetto a diverse viste del sistema
- rispetto a diverse componenti del sistema
- ...

Ingegneria del Software: Principi

- ◆ **Modularità:** suddividere un sistema complesso in parti più semplici
 - » Decomporre il problema (*divide et impera*)
 - » Comporre le soluzioni ai sotto-problemi
 - » Comprendere meglio il sistema globale
 - high cohesion: alta coesione all'interno del modulo
 - low coupling: poca interdipendenza fra moduli
- ◆ **Astrazione:** si identificano gli aspetti cruciali in un certo istante ignorando gli altri
- ◆ **Anticipazione del cambiamento:** la progettazione deve favorire l'evoluzione del SW

Ingegneria del Software: Principi

- ◆ **Generalità:** tentare di risolvere il problema nella sua accezione più generale
 - più semplice o più complesso del problema originario
 - trade-offs con:
 - » fattibilità
 - » costo
 - » beneficio
- ◆ **Incrementalità:** lavorare per passi successivi
 - un sottoinsieme iniziale, una consegna iniziale, feedback iniziali
 - molte feature possono essere aggiunte in un modo incrementale

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 lessso, ...
- ◆ **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, ...

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

Processo software: Standard IEEE 610.12-1990

- ◆ *Software 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 software. 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

Fonti

- ◆ 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.jelonsoftware.com
 - www.vWorker.com (www.rentacoder.com)
 - best-practice-software-engineering.blogspot.com
- ◆ Gruppi linkedin
 - Software testing and quality assurance
 - Software as a service