Ingegneria del Software
1. Introduzione

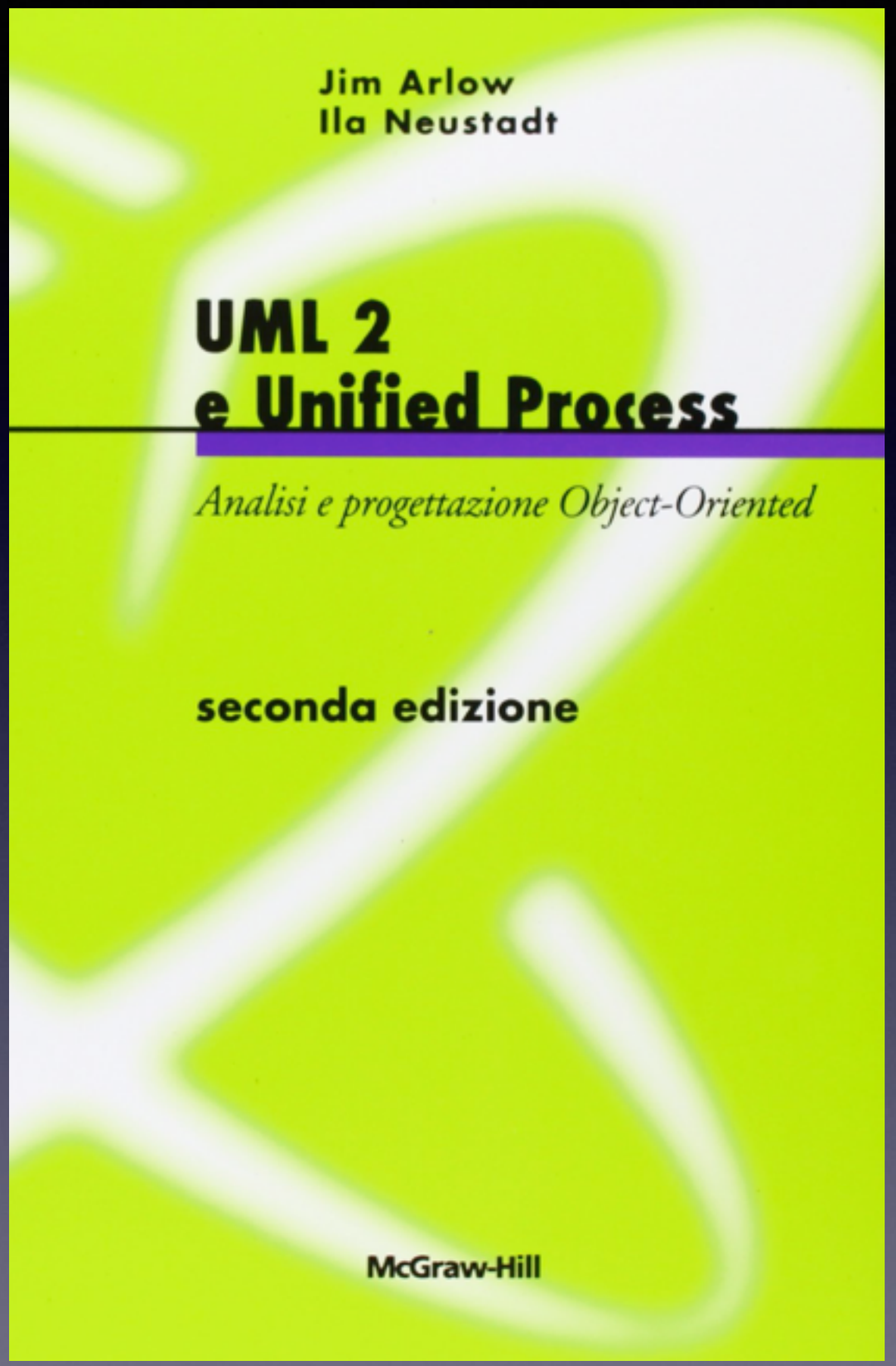
Dipartimento di Informatica
Università di Pisa
A.A. 2014/15

first step: orario etc.

- Lezione: Lunedì dalle 11 alle 13
- Lezione: Mercoledì dalle 11 alle 13
- Ricevimento: Mercoledì dalle 14 alle 16
- Pagina web: didawiki.cli.di.unipi.it/doku.php/informatica/is-a/

materiale didattico, I

J. Arlow, I. Neustadt
UML 2 e Unified Process
McGraw-Hill (2007)



materiale didattico, I

J. Arlow, I. Neustadt
UML 2 e Unified Process
McGraw-Hill (2007)

A. Binato, A. Fuggetta, L. Sfardini
Ingegneria del Software
Addison Wesley (2006)

materiale didattico, II

C. Montangero, L. Semini
Architetture software e progettazione di dettaglio **(2014)**

V. Ambriola, C. Montangero, L. Semini
Esercizi di Ingegneria del Software **(2009)**

G. Cignoni, C. Montangero, L. Semini
Il controllo del software: verifica e validazione **(2014)**

Laurea in Informatica e in Informatica applicata

Dipartimento di Informatica

Università di Pisa

Il controllo del software:

verifica e validazione

Giovanni A. Cignoni

Carlo Montangero

Laura Semini

Dispensa per il Corso di Ingegneria del Software

2014

modalità d’esame

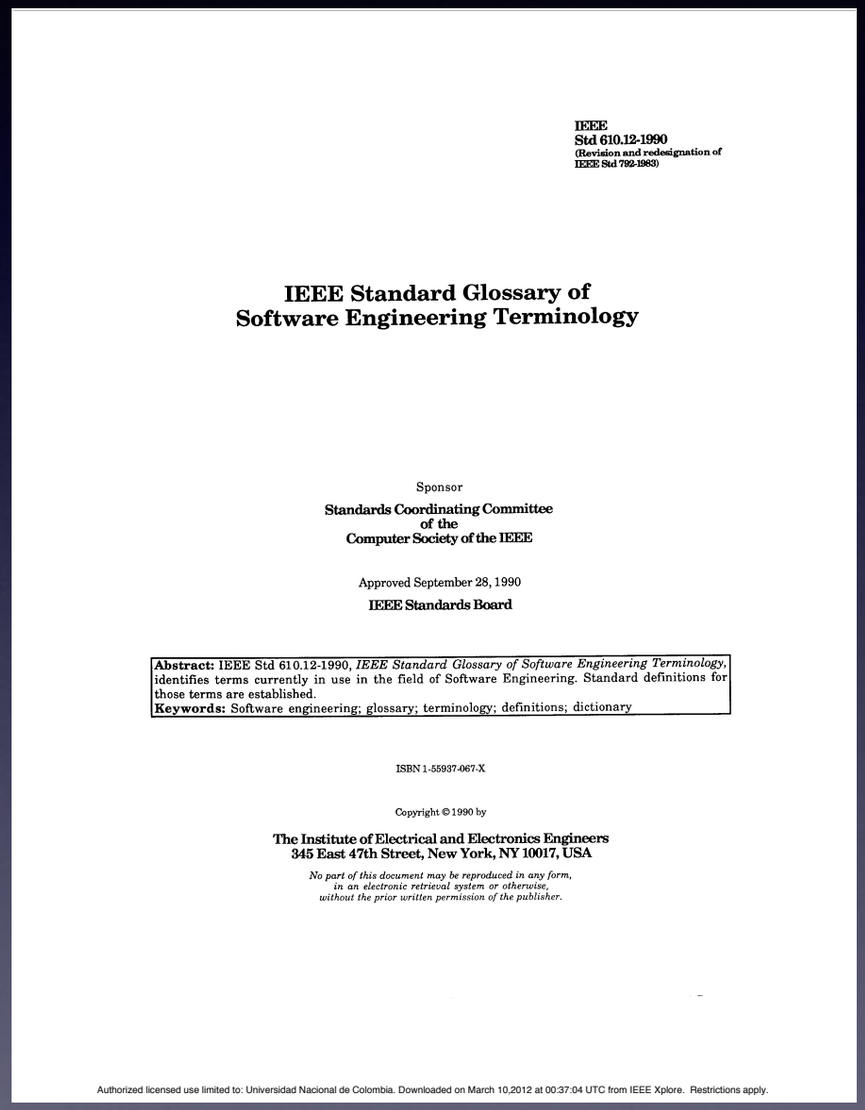
- Scritto: appello o prova in itinere
  - Appello: voto minimo 16
  - Prova in itinere: voto medio 18, voto minimo 15
- Senza appunti, lucidi o testi
- Validità
  - Appello: solo per quell’appello
  - Prova in itinere: per tutti i cinque appelli

obbiettivi

- Introduzione alle tecniche di modellazione
  - *Conoscenze*: Lo studente conoscerà i principali modelli di processo per lo sviluppo software, e le tecniche di modellazione proprie delle varie fasi
  - *Capacità*: Lo studente utilizzerà no(ta)zioni di modellazione standard (i.e. UML) per l'analisi dei requisiti e la progettazione sia architettonica sia di dettaglio di un sistema software

in breve…

- Software Engineering (IEEE 610)
  - A systematic, disciplined, quantifiable approach […]
  - Disciplina sia tecnologica che gestionale
- Strumenti e metodi
  - Dai linguaggi agli ambienti di sviluppo
  - Dai requisiti alle architetture
  - Dai processo di sviluppo al controllo dei fornitori
  - …
- Nella pratica, gestione dei progetti software!



IEEE
Std 610.12-1990
(Revision and redesignation of
IEEE Std 792-1983)

IEEE Standard Glossary of Software Engineering Terminology

Sponsor
Standards Coordinating Committee
of the
Computer Society of the IEEE

Approved September 28, 1990
IEEE Standards Board

Abstract: IEEE Std 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*, identifies terms currently in use in the field of Software Engineering. Standard definitions for those terms are established.
Keywords: Software engineering; glossary; terminology; definitions; dictionary

ISBN 1-55937-067-X

[withdrawn: now check ISO/IEC/IEEE 24765 2010 pascal.computer.org/sev_display/]

“produrre” industrialmente

- Produzione industriale di sw
  - in grande, per dimensione o per volumi
  - garanzia di rispetto dei tempi e dei requisiti
  - efficienza nella produzione (costi, risorse…)
- … e il ruolo dell’Ingegneria del Software
  - metodologia di progettazione
  - “euristiche” di analisi e documentazione [pochi vs. molti]
  - soluzioni pratiche, replicabili, affidabili, sostenibili…

dimensioni del “prodotto”

- Una crescita irresistibile…
  - 1962 Mercury: 1M LdC
  - 1965 Gemini: 3M LdC
  - 1969 Apollo: 11M LdC
  - 1981 Shuttle: 37M LdC
  - 1990 Hubble: 82M LdC

- …e siamo ancora a 25 anni fa!!

una scelta irreversibile…

- Sistemi ormai pervasivi
- Innegabili benefici
- Rivoluzione sociale e culturale
  - Cambiamenti radicali nel modo di lavorare
  - Nuove professionalità/opportunità di impresa
  - Rinnovate concezioni di informazione/conoscenza

contenuti (sort of)

- Processo di sviluppo software
  - Problemi della produzione del software
  - Modelli di ciclo di vita
- Analisi del dominio e dei requisiti
  - modelli statici (classi, casi d'uso)…
  - e dinamici (attività, macchine a stati, …)
- Architetture software
  - Viste strutturali, comportamentali, logistiche
    - Stili architettonici
    - Specifica di componenti e connettori
- Architetture software (ctd.)
  - Progettazione
    - …come realizzazione dei casi d'uso
- Progettazione di dettaglio
  - Progettazione OO di componenti
- Verifica
  - Progettazione e valutazione delle prove: criteri funzionali e strutturali
  - Verifiche statiche
- Standard per la modellazione
  - Unified Modelling Language (UML2)

caratteristiche del software

- Il software è diverso da altri prodotti del lavoro…
  - non è vincolato da materiali, né governato da leggi fisiche o da processi manufatturieri (nessun costo marginale!)
  - si “sviluppa”, non si “fabbrica” nel senso tradizionale
  - non si “consuma”, tuttavia si “deteriora”
  - spesso si “assembla”, ma larga parte ancora si realizza *ad hoc*

miti del software

"In assenza di standard significativi, una nuova industria come quella del software comincia a dipendere dal folclore"
Tom De Marco

[check wiki]

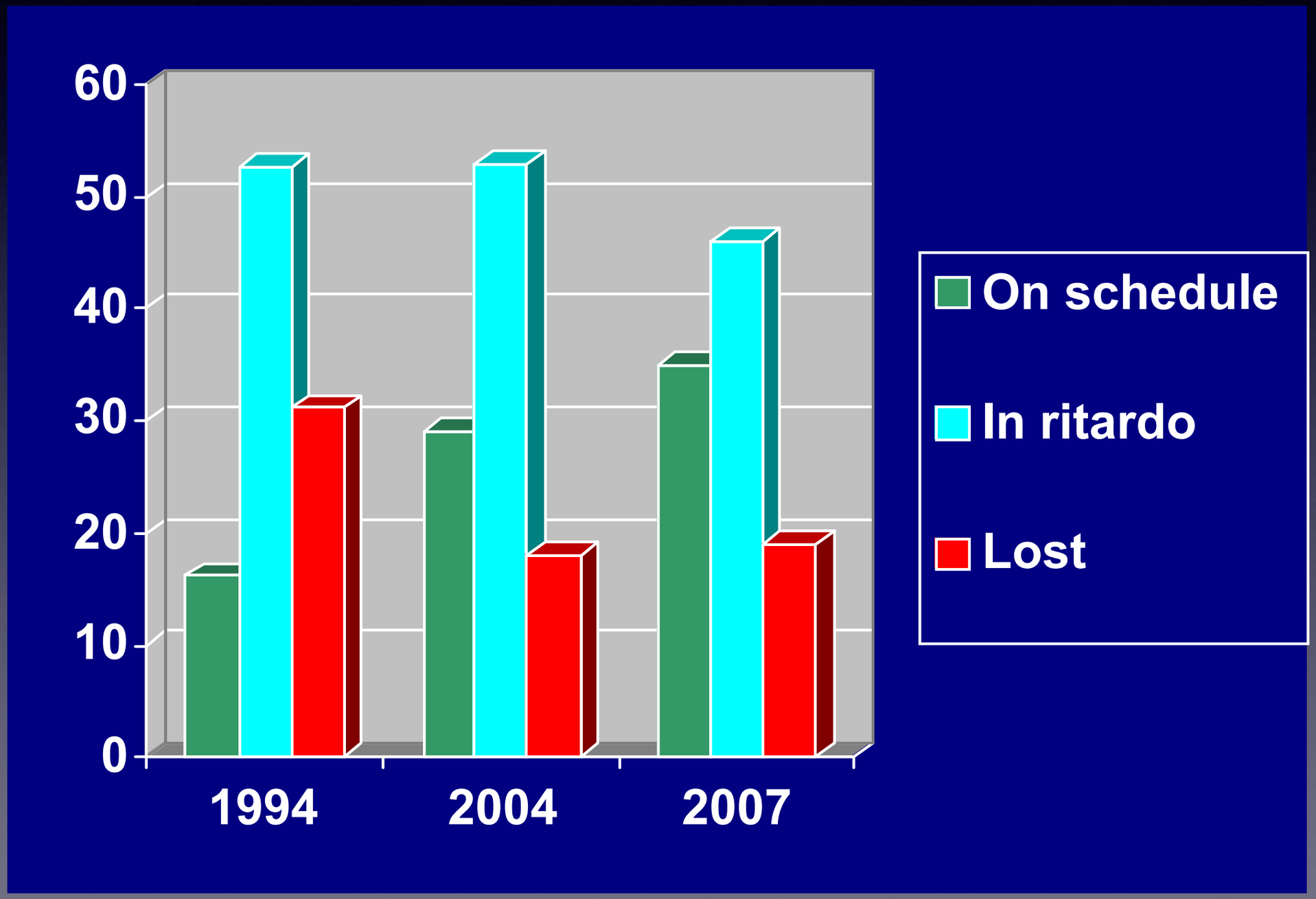
- Abbiamo volumi di standard e procedure da seguire. Non c'è forse già l'indispensabile?
- Abbiamo i più moderni strumenti di sviluppo; acquistiamo sempre i computer più recenti.
- Se siamo in ritardo, possiamo recuperare aumentando il numero di programmatori.
- Un'affermazione generica degli scopi è sufficiente per iniziare a scrivere i programmi.
- Una volta scritto e messo in opera il programma, il nostro lavoro è finito.

[check more at spmn.com]

problemi anche gestionali

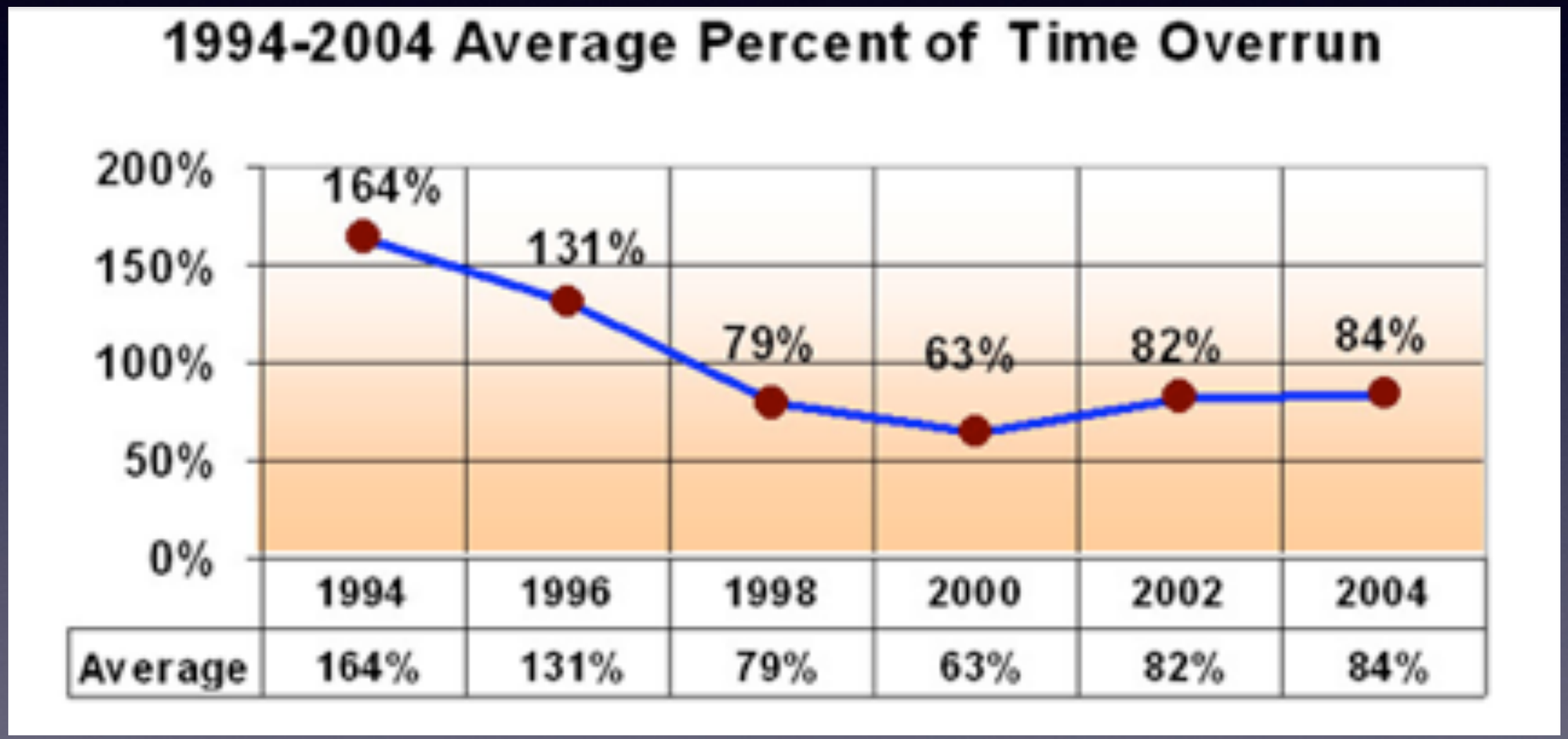
- I progetti software sono spesso in ritardo…
  - Difficoltà nelle fasi iniziali
  - Cambi di piattaforma/tecnologia
  - Carenze nel prodotto finale
- A volte (clamorosamente) falliscono!!
  - Per obsolescenza prematura
  - Per incapacità di raggiungere gli obbiettivi
  - Per esaurimento fondi

Chaos Report



[check at standishgroup.com]

Chaos Report



Chaos Report

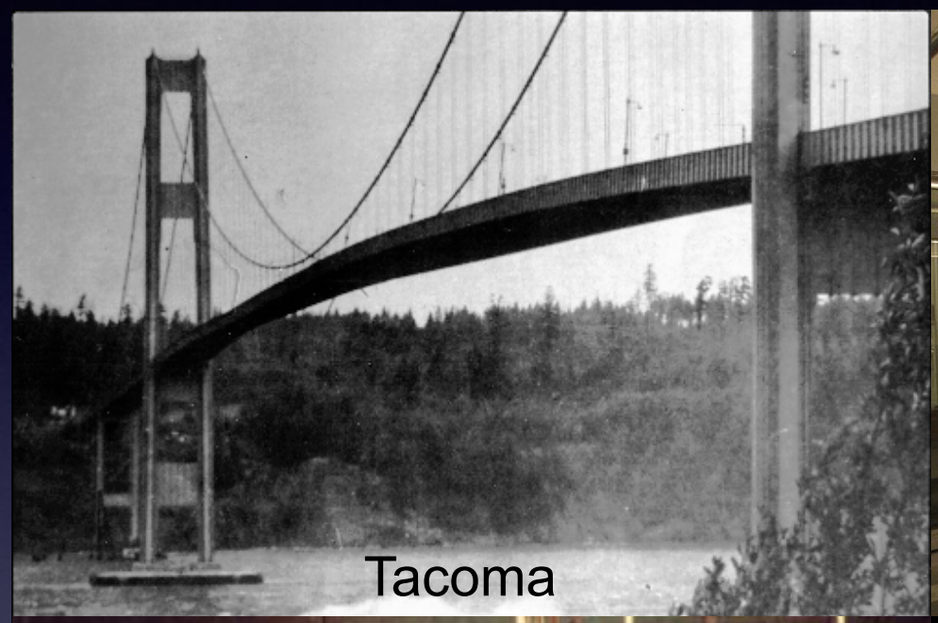
cause di abbandono

1. Requisiti incompleti
2. Scarso coinvolgimento degli utenti
3. Mancanza di risorse
4. Attese irrealistiche
5. Modifiche a specifiche e requisiti

[...]

10. Ignoranza tecnologica

chi è l’intruso?


Tacoma


Paris

Denver


Stockholm

aeroporto di Denver (1995)

- Sistema di smistamento dei bagagli
  - 35 Km di rete, 4000 carrelli, 5000 “occhi”, 56 lettori
  - 193M $ di investimento
- Risultati
  - inaugurazione dell’aeroporto ritardata di 16 mesi (1.1M $ al giorno di costi aggiuntivi)
  - sforamento di 3,2G $ rispetto ai preventivi
- Progettazione difettosa
  - occorrenza di jam (mancanza di sincronizzazione)
- No fault tolerance
  - si perdeva traccia di quali carrelli fossero pieni e quali vuoti dopo un riavvio dovuto a jam

metro di Parigi (1998)

- Driverless CBTC System per la linea 14
  - La linea “Météor” (Metro Est-Ouest Rapid), completamente automatizzata e senza macchinista, è la prima nuova linea cittadina (eccettuata Orlyval) dal 1934
  - Aperta nel 1998, ed estesa nel 2003 e nel 2007 (e ancora nel 2017)
  - Moderno sistema di segnalazione per la gestione del traffico e il controllo della infrastruttura realizzato in collaborazione con Siemens
  - CBTC (Commmunication-based Train Control) è un “continuous, automatic train control system utilizing high-resolution train location determination, independent of track circuits”, come specificato nello standard IEEE 1474
  - Parte del controllo gestito da OS OpenVMS, verificato utilizzando il B-method di Jean-Raymond Abrial (macchine astratte e generazione di codice ADA, C, C++)

e ancora…

- Sistema antimissile Patriot (1991)
  - Problema al clock interno per uso prolungato (100h vs. 14h)
- London Ambulance Service (1992)
  - Ottimizzazione percorsi e guida vocale
  - Problema con interfacce utente, fault tolerance, backup, …
- Arianne 5 (1996)
  - Problema al giroscopio per *legacy code* da Arianne 4

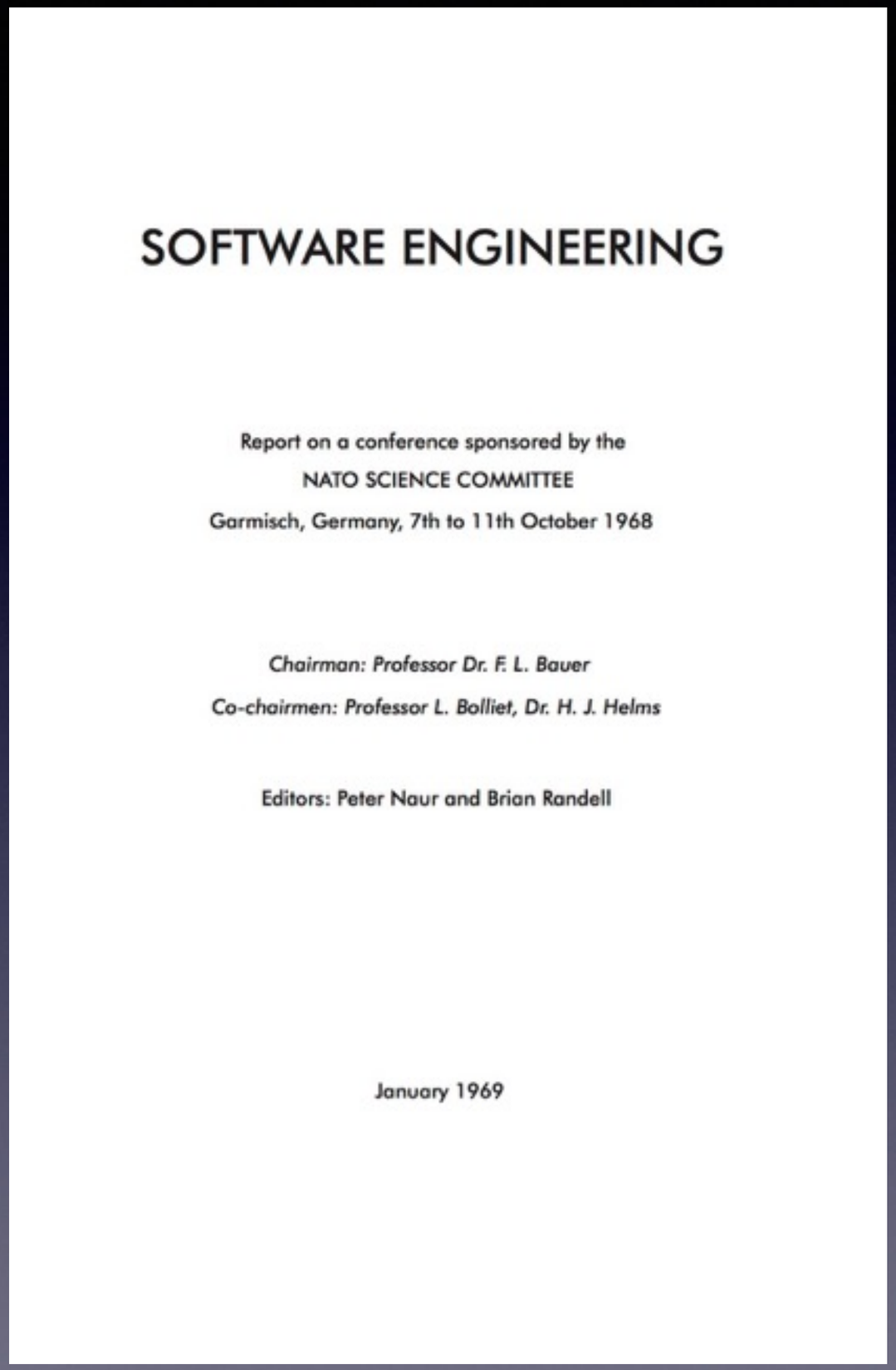
da dove veniamo

- Due concetti nati negli anni Sessanta
  - *crisi del software:* problemi incontrati nello sviluppo di sistemi software complessi
  - *ingegneria del software*: soluzione alla crisi del software
- Prima: software sviluppati informalmente
  - Per risolvere sistemi di equazioni…
- Poi: grandi sistemi commerciali
  - OS per CEP (migliaia di LdC)
  - OS 360 per IBM 360 (milioni di LdC)
  - Sistemi informativi aziendali (gestione centralizzata di tutte le informazioni sulle funzioni)
- Fact: la programmazione strutturata (et similia) non basta!

un evento chiave

Conclusione: l’industria del software ha bisogno di metodologie…

…per controllare prodotto e produzione!!



SOFTWARE ENGINEERING

Report on a conference sponsored by the
NATO SCIENCE COMMITTEE
Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer

Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

January 1969

crisi temporanea, o…

- I grandi sistemi software moderni sono tra le strutture più complesse costruite dagli uomini
  - milioni di linee di codice, migliaia di tabelle nelle basi di dati
  - esecuzione simultanea/interattiva su un numero altissimo di dispositivi
- È migliorata l'abilità generale nella produzione, ma è anche aumentata la complessità dei sistemi
  - Le prestazioni dell'HW crescono…
  - …e crescono le aspettative (e le esigenze) dai sistemi!!

"[T]here is no single development, in either technology or management technique, which by itself promises even one order-of-magnitude improvement within a decade in productivity, in reliability, in simplicity."

Fred Brooks [*No silver bullet*, 1986]

ancora sulla definizione

- “[T]he application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to sw” [ISO 24765]
  - Il software è un prodotto con il suo ciclo vitale
  - La parola chiave è “sistematico”
- “La disciplina tecnologica e gestionale per la produzione sistematica e la manutenzione di prodotti software sviluppati e modificati con tempi e costi preventivati” [Richard Fairley 1985]
  - Disciplina gestionale: costi, tempi, risorse
  - Pianificazione e controllo della qualità: costi e risultati definiti

Software engineering concepts

sulle tematiche

- Una disciplina articolata…
  - Strette connessioni con la tecnologia
  - Rilevanti agganci con temi economici e gestionali
- Una forte reattività ai problemi concreti
- Tre aree tematiche principali
  - Realizzazione di sistemi software
  - Processo software
  - Controllo del prodotto

[check more at computer.org/web/swebok/]

realizzazione di sistemi sw

- Strategie di analisi e progettazione
  - Tecniche per la comprensione e la soluzione di un problema
  - Top-down/bottom-up, progettazione modulare, a componenti, OO…
- Linguaggi di specifica e progettazione
  - Strumenti formali per la definizione di sistemi software
  - Reti di Petri, Z, OMT, UML, SysML…
- Ambienti di sviluppo
  - Strumenti per analisi, progettazione e realizzazione
  - Strumenti tradizionali, CASE, CAST, IDE, RAD…

processo software

- Organizzazione e gestione dei progetti
  - Metodi di composizione dei gruppi di lavoro
  - Strumenti di stima, pianificazione, analisi, controllo
- Cicli di vita del software
  - Definizione e correlazione delle attività
  - Modelli ideali di processo di sviluppo
- Modellazione del processo di sviluppo
  - Norme per la definizione dei processi
  - Definizione e gestione di processo

controllo del prodotto

- Modelli di qualità
  - Definizione di caratteristiche della qualità
- Metriche software
  - Unità di misura, scale di riferimento, strumenti
  - Indicatori di qualità
- Metodi di verifica e validazione
  - Metodi di verifica, criteri di progettazione delle prove
  - Controllo della qualità, valutazione del processo di sviluppo

altre specificità del software

- Poca manifattura e tanta progettazione
  - Il costo della riproduzione è irrilevante…
  - gli errori nella fase di produzione facilmente rimediabili…
    - ma lo sviluppo e la modifica sono solo apparentemente semplici!!
- Difficile da automatizzare
- Difficile stabilire produttività
  - Il sw è intangibile, e dipende dalle caratteristiche dell'applicazione
- Complessità esponenziale sulle dimensioni