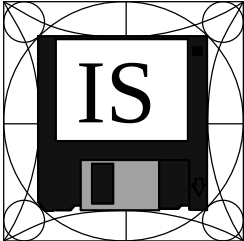




Amministrazione di progetto

IS 2001-9
Corso di Ingegneria del Software
V. Ambriola, G.A. Cignoni,
C. Montanero, L. Semini
Con aggiornamenti di: T. Vardanega (UniPD)



Dipartimento di Informatica, Università di Pisa

1 / 27



Amministrazione di progetto

Contenuti

- ❑ Amministrazione di progetto
- ❑ Documentazione di progetto
- ❑ Ambiente e strumenti
- ❑ Norme di progetto e di codifica
- ❑ Seminario: leggibilità del codice

Dipartimento di Informatica, Università di Pisa

2 / 27



Amministrazione di progetto

Amministrare un progetto

- ❑ Amministrare non è dirigere
 - L'amministratore non compie scelte tecnologiche
 - L'amministratore non compie scelte gestionali
- ❑ Attività
 - Redazione e manutenzione delle regole
 - Ma non la loro approvazione, che spetta al responsabile di progetto
 - Accertamento di applicazione delle regole
 - Responsabilità su reperimento e disponibilità delle risorse
 - Per tutte le risorse di un progetto tranne che per il personale
 - Ambiente, infrastruttura, strumenti, prodotti, documenti

Dipartimento di Informatica, Università di Pisa

3 / 27



Amministrazione di progetto

Documentazione di progetto

- ❑ Tutto ciò che documenta le attività
 - Riguardo al prodotto
 - Riguardo al processo
- ❑ Documenti di sviluppo
 - Documentazione fornita dal cliente
 - Diagrammi di progettazione
 - Codice
 - Piani di qualifica e risultati delle prove
 - Documentazione di accompagnamento del prodotto
- ❑ Documenti di gestione del progetto
 - Documenti contrattuali
 - Piani e consuntivi delle attività
 - Piani di qualità

Dipartimento di Informatica, Università di Pisa

4 / 27



Amministrazione di progetto

Disponibilità e diffusione

- ❑ I documenti sono utili solo se sono sempre e subito disponibili
 - Chiaramente identificati
 - Corretti nei contenuti
 - Verificati e approvati
 - Aggiornati, datati e dotati di versione
- ❑ La loro diffusione deve essere controllata
 - I destinatari devono essere chiaramente identificati
 - Ogni documento ha una sua lista di distribuzione
 - L'amministratore gestisce le liste di distribuzione e ne assicura il rispetto

Dipartimento di Informatica, Università di Pisa

5 / 27



Amministrazione di progetto

Ambiente di lavoro

- ❑ Quanto necessario ai processi coinvolti nella produzione
- ❑ La qualità dell'ambiente determina la produttività
 - Influisce sulla qualità del processo
 - Influisce sulla qualità del prodotto
- ❑ Caratteristiche di qualità dell'ambiente di lavoro
 - Completo
 - Deve offrire tutto il necessario per svolgere le attività previste
 - Ordinato
 - Deve essere facile trovarvi ciò che si cerca
 - Aggiornato
 - Il materiale obsoleto non deve causare intralcio

Dipartimento di Informatica, Università di Pisa

6 / 27



Amministrazione di progetto

Infrastruttura

- ❑ **Risorse hardware**
 - **Server**
 - Archivi logici centralizzati, sempre aggiornati e a disposizione
 - **Infrastruttura di rete**
 - Sempre operativa, protetta, accessibile anche da remoto
 - **Postazioni di lavoro e dispositivi di utilità**
 - Per assicurare la massima produttività individuale
 - **Archivi fisici**
- ❑ **Risorse software**
 - **Ambienti di sviluppo, di prova e di studio e documentazione**
 - Strumenti di sviluppo
 - Prodotti del processo di sviluppo (documentazione inclusa)
 - **Archivio logico (repository) di documentazione di progetto**

Dipartimento di Informatica, Università di Pisa

7 / 27



Amministrazione di progetto

Strumenti essenziali

- ❑ **Gestione di progetto**
 - Pianificazione, stima e controllo dei costi
 - **Allocazione e gestione delle risorse**
 - Redazione e consultazione di diagrammi di Gantt e PERT (p.es., <http://www.ganttproject.biz/>)
- ❑ **Analisi e progettazione**
 - **Supporto alle metodologie**
 - UML (p.es., <http://www.gaprusuml.org/>)
 - **Tracciamento dei requisiti**
 - Come una speciale base dati (testo, XML)
- ❑ **Realizzazione e automazione**
 - **Misurazione e analisi del codice**
 - **Generazione ed esecuzione automatica delle prove**
 - **Versionamento e configurazione**
 - Ambienti integrati (p.es., <http://www.eclipse.org/>)

Dipartimento di Informatica, Università di Pisa

8 / 27



Amministrazione di progetto

Gestione delle modifiche – 1

- ❑ **Le richieste di modifiche hanno origine da**
 - Utenti (difetti o mancanze)
 - Sviluppatori (idem)
 - Competizione (valore aggiunto)
- ❑ **Le richieste di modifica vanno sottoposte a un rigoroso processo di analisi, decisione, realizzazione e verifica**
 - Sempre tenendo traccia dello stato precedente

Dipartimento di Informatica, Università di Pisa

9 / 27



Amministrazione di progetto

Gestione delle modifiche – 2

- ❑ **Ogni richiesta di modifica va inoltrata in modo formale**
 - **Change request form**
 - Autore, motivo, urgenza
 - Stima di fattibilità, valutazione di impatto, stima di costo
 - Decisione del responsabile
- ❑ **Di ogni richiesta di modifica bisogna tenere traccia**
 - **Tracking elo ticketing**
 - Per esempio con Bugzilla
 - **Stato corrente ed eventuale esito chiusura**

Dipartimento di Informatica, Università di Pisa

10 / 27



Amministrazione di progetto

Strumenti di sviluppo

- ❑ **Non solo compilatore e debugger !**
- ❑ **Produrre codice secondo regole**
 - Editore integrato con verificatore di regole e stile
- ❑ **Verificare il codice a partire dalle unità più piccole**
 - Strumenti di automazione delle verifiche
- ❑ **Versionamento per tener traccia della “storia” dei prodotti**
 - CVS (concurrent versions system, <http://www.nongnu.org/cvs/>)
 - SVN (subversion, <http://subversion.tigris.org/>)
- ❑ **Configurazione per tenere insieme le parti del prodotto finale**
 - Più di “make”
 - Ant (<http://ant.apache.org/>), Maven (<http://maven.apache.org/>)

Dipartimento di Informatica, Università di Pisa

11 / 27



Amministrazione di progetto

Versionamento – 1

- ❑ **Versione**
 - Istanza di prodotto funzionalmente distinta dalle altre
- ❑ **Variante**
 - Istanza di prodotto funzionalmente identica ad altre ma diversa per caratteristiche non funzionali
- ❑ **Rilascio (release)**
 - Istanza di prodotto resa disponibile a utenti esterni
- ❑ **Tutte vanno identificate, pianificate e gestite**
 - Identificazione per numero, caratteristiche, modifiche

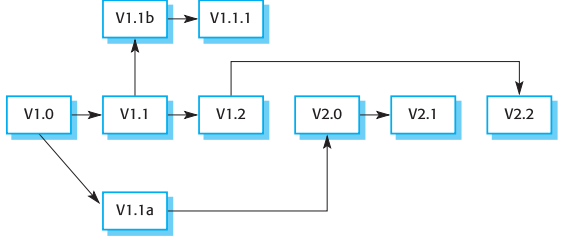
Dipartimento di Informatica, Università di Pisa

12 / 27



Amministrazione di progetto

Versionamento – 2



Tratto da: Ian Sommerville, Software Engineering, 8ª ed.

Dipartimento di Informatica, Università di Pisa

13 / 27



Amministrazione di progetto

Configurazione

- ❑ Un prodotto *software* è l'unione di parti distinte unite insieme seguendo regole rigorose
 - Specifiche, progetti, programmi, dati di verifica, manualistica
- ❑ Le regole di configurazione vanno pianificate
 - Le responsabilità di configurazione vanno assegnate
- ❑ Il piano di configurazione va automatizzato
 - Servono strumenti adatti
 - Configuration Management
 - Build

Dipartimento di Informatica, Università di Pisa

14 / 27



Amministrazione di progetto

Norme di progetto

- ❑ Linee guida per le attività di sviluppo
 - Storicamente, hanno preceduto le procedure aziendali
 - Oggi sono lo strumento operativo che le completa
- ❑ Contenuti
 - Norme di codifica
 - Organizzazione e uso delle risorse di sviluppo
 - Convenzioni sull'uso degli strumenti di sviluppo
 - Organizzazione della comunicazione e della cooperazione
 - Gestione dei cambiamenti!

Dipartimento di Informatica, Università di Pisa

15 / 27



Amministrazione di progetto

Organizzazione di una norma

- ❑ Regole
 - Convenzioni di cui si riconosce necessità e convenienza
 - Ne è richiesto e accertato il rispetto
- ❑ Raccomandazioni
 - Prassi desiderabile
 - Inviti e suggerimenti senza verifica di rispetto
- ❑ Il contesto definisce la portata della norma
 - Non tutto può essere regolato
 - Troppe regole sono di difficile attuazione e verifica

Dipartimento di Informatica, Università di Pisa

16 / 27



Amministrazione di progetto

Obiettivi delle norme di codifica

- ❑ Leggibilità come forma di prevenzione
 - Verificabilità
 - Manutenibilità
 - Portabilità
- ❑ Come è “scritto” il codice?
- ❑ È comprensibile a distanza di tempo?
- ❑ È comprensibile a chi non lo ha prodotto?

Dipartimento di Informatica, Università di Pisa

17 / 27



Amministrazione di progetto

Convenzioni sui nomi

- ❑ Nel codice
 - Tipi, costanti, variabili, funzioni, ...
 - P.es., le norme Javadoc (vedi: <http://java.sun.com/j2se/javadoc/>)
- ❑ Nel progetto
 - Strutturazione in moduli, file, directory, ...
- ❑ Aspetti pratici
 - Conflitti logici all'interno o all'esterno del codice
 - Abbreviazioni, per comodità o per necessità
 - Limiti intrinseci del linguaggio
 - P.es., identificazione forte o debole dei tipi (Java ↔ C)
 - Limiti degli strumenti
 - P.es., lunghezza massima dei nomi di file (p.es.: Windows 95-98)

Dipartimento di Informatica, Università di Pisa

18 / 27



Amministrazione di progetto

Indentazione del codice

- ❑ **Obiettivi**
 - Programmazione strutturata
 - Evidenziare visivamente la struttura di un programma
- ❑ **Aspetti da non sottovalutare**
 - Lunghezza delle linee
 - Indentazione
 - Posizione degli fine linea nei blocchi
 - Posizione degli fine linea nelle espressioni
- ❑ **Come evitare guerre ideologiche sugli stili?**

Dipartimento di Informatica, Università di Pisa

19 / 27



Amministrazione di progetto

Intestazione del codice

- ❑ **Obiettivi**
 - Identificazione e collocamento di una unità (modulo, file)
 - Storia e responsabilità delle modifiche
- ❑ **Contenuti**

○ Dati dell'unità	tipo, contenuto, posizione
○ Responsabilità	autore, reparto, organizzazione
○ Copyright / copyleft	licenze, visibilità
○ Avvertenze	limiti di uso e di garanzia
○ Registro modifiche	storia, spiegazione, <u>versione</u>

Dipartimento di Informatica, Università di Pisa

20 / 27



Amministrazione di progetto

Intestazione – esempio 1

```
// File: HAL_kern.H - HAL 9000 KB Data defs -*- C++ -*-
// Module: HAL 9000 KB kernel
// Created: 1997 January 12
// Author: Dr. Chandra - 9000 Proj., HAL Inc., Urbana, ILL
// E-Mail: chandra@p9000.hal.com
//
// Copyright (C) 1996, 1997, Dr. Chandra, HAL Inc.
// All rights reserved.
//
// This software and related documentation are
// distributed under license. No permission is given
// to use, copy, modify or distribute this software
// without explicit authorization of HAL Inc.
// and its licensors, if any.
//
// Software licensed to:
// NO LICENSE - For HAL internal use only.
//
// This software is provided "as is" WITHOUT ANY WARRANTY
// either expressed or implied, including, but not limited
// to, the implied warranties of MERCHANTABILITY or
// FITNESS FOR A PARTICULAR PURPOSE.
```

Dipartimento di Informatica, Università di Pisa

21 / 27



Amministrazione di progetto

Intestazione – esempio 2

```
// BANKSEC project (IST 6087)
//
// BANKSEC-TOOLS/AUTH/RBAC/USER_ROLE
//
// Object: currentRole
// Author: N. Perwaiz
// Creation date: 10th November 2002
//
// © Lancaster University 2002
//
// Modification history
// Version      Modifier Date      Change      Reason
// 1.0          J. Jones      1/12/2002   Add header   Submitted to CM
// 1.1          N. Perwaiz    9/4/2003   New field    Change req. R07/02
```

Tratto da: Ian Sommerville, *Software Engineering*, 8^a ed.

Dipartimento di Informatica, Università di Pisa

22 / 27



Amministrazione di progetto

Uso del linguaggio di programmazione

- ❑ **Serve una strategia forte per costringere i programmatori a lavorare come si conviene**
- ❑ **Prescrizioni tipiche**
 - Compilazione senza errori fatali o potenziali (*warning*)
 - Uso chiaro e coerente dei costrutti del linguaggio
 - **Uso di un sottoinsieme appropriato del linguaggio**
 - I costrutti di maggiore robustezza, verificabilità, leggibilità
 - Non necessariamente quelli di maggiore potenza espressa e velocità

Dipartimento di Informatica, Università di Pisa

23 / 27



Amministrazione di progetto

Leggibilità del codice

- ❑ **Il codice illeggibile è disarmante e irritante**
- ❑ **Modificarlo costa tempo ed è rischioso**
- ❑ **La leggibilità facilita le attività di ispezione**
- ❑ **Il codice è una risorsa**
- ❑ **Il primo (l'ultimo) posto dove guardare**

Dipartimento di Informatica, Università di Pisa

24 / 27

[illegible]



Amministrazione di progetto

Notazione ungherese ☺

- ❑ **lpfnWndProc** (un campo di struttura)
 - **l** *long* (32-bit integer)
 - **p** *pointer* [to a]
 - **fn** *function* [handling messages directed to]
 - **wnd** *[a] window*
 - **Proc** *procedure*

- Un vettore di puntatori a descrittori di finestre, indicizzato sul numero di finestre



❑ **Un'idea di Charles Simonyi @ Microsoft**



Amministrazione di progetto

Riferimenti

- ❑ **V. Ambriola, G.A. Cignoni, “Laboratorio di progettazione”, Jackson Libri, 1996**
- ❑ **“Programming in C++ – Rules and Recommendations”, Ellemtel TSL (Svezia), 1992**
- ❑ **C. Simonyi, M. Heller, “The Hungarian Revolution”, Byte, agosto 1991**

Dipartimento di Informatica, Università di Pisa

27 / 27