



Amministrazione di progetto

IS 2001-5
Corso di Ingegneria del Software
V. Ambriola, G.A. Cignoni,
C. Montangero, L. Semini
Con aggiornamenti di: T. Vardanega (UniPD)



Dipartimento di Informatica, Università di Pisa 1/22




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 2/22




Amministrazione di progetto

Amministrare un progetto

- **Amministrare non è dirigere**
 - L'amministrazione non compie scelte tecnologiche
 - L'amministrazione non compie scelte gestionali
- **Attività**
 - **Redazione e manutenzione delle regole**
 - 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 3/22




Amministrazione di progetto

Documentazione di progetto

- **Tutto ciò che documenta le attività**
 - Prodotto e 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 4/22



Amministrazione di progetto

Disponibilità e diffusione

- **I documenti devono essere disponibili**
 - Chiaramente identificati
 - Corretti nei contenuti
 - Verificati e approvati
 - Aggiornati, datati e dotati di versione
- **La loro diffusione deve essere controllata**
 - Tutti e soli gli interessati devono averne presa visione
 - Ogni documento ha una specifica lista di distribuzione
 - L'amministratore gestisce le liste di distribuzione e ne assicura il rispetto

Dipartimento di Informatica, Università di Pisa 5/22




Amministrazione di progetto

Ambiente di lavoro

- **Necessario al processo di produzione**
- **Qualità dell'ambiente (produttività)**
 - Influisce sulla qualità del processo
 - Influisce sulla qualità del prodotto
- **Caratteristiche di qualità di ambiente**
 - **Completo** deve offrire tutto il necessario
 - **Ordinato** deve essere facile trovarvi ciò che si cerca
 - **Aggiornato** il materiale obsoleto non deve intralciare


Dipartimento di Informatica, Università di Pisa 6/22



Amministrazione di progetto
Infrastruttura

- **Risorse hardware**
 - *Server, mainframe*
 - Rete locale, connettività
 - Stazioni di lavoro, dispositivi di utilità
 - Archivi logici (dischi)
 - Archivi fisici (schedari, scaffali, armadi)
- **Risorse software**
 - Ambienti di sviluppo, di prova e di lavoro
 - Strumenti di sviluppo
 - Prodotti del processo di sviluppo (documentazione inclusa)
 - **Server interni per la diffusione di informazione (intranet)**


Dipartimento di Informatica, Università di Pisa7/22



Amministrazione di progetto
Strumenti di sviluppo - 1

- **Non occorrono solo compilatori**
 - Editori di testo *vi, (x)emacs, [IDE]*
 - Verificatori *lint, prof, gprof, .*
 - (g)prof: *program execution profiler*; *lint* (verificatore per C/C++);
 - Debugger *gdb, ddd, ...*
 - Versionamento *rcs, sccs, cvs, ...*
 - *sccs*: *source code control system* (1972)
 - *rscs*: *revision control system* (UNIX → FSF)
 - *cvs*: *concurrent versions system*, basato sul modello *client-server* (<http://www.gnu.org/software/cvs>)
 - Configurazione *make, imake, autoconf, ...*
 - *imake*: *makefile generator* (<http://www.x.org>)
 - *autoconf*: generatore di macro per la configurazione automatica (<http://www.gnu.org/software/autoconf/>)

Dipartimento di Informatica, Università di Pisa8/22



Amministrazione di progetto
Strumenti di sviluppo - 2

- **Ambienti di supporto integrati**
 - CASE *Computer Aided Software Engineering*
 - Di concezione originata nei primi anni '80
 - RAD *Rapid Application Development*
 - Uso di "wizard", meno controllo da parte del programmatore e più vincoli di ambiente e di piattaforma (p.es. Delphi, Visual BASIC)
 - CAST *Computer Aided Software Test*
 - Strumenti ed ambienti per la generazione e l'applicazione (semi-)automatica di *harness*, *script*, *stub*, *driver* per prove
 - IDE *Integrated Development Environment*
 - Visual Studio, Visual ONE, GPS, ...


Dipartimento di Informatica, Università di Pisa9/22



Amministrazione di progetto
Strumenti di processo

- **Gestione di progetto**
 - Pianificazione e stima dei costi
 - Allocazione e gestione delle risorse
- **Analisi e progettazione**
 - Ambienti di supporto alle metodologie
 - Tracciamento dei requisiti
- **Supporto alla realizzazione**
 - Misurazione ed analisi del codice
 - Generazione ed esecuzione automatica delle prove

Dipartimento di Informatica, Università di Pisa10/22



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 ed uso delle risorse di sviluppo
 - Convenzioni sull'uso degli strumenti di sviluppo
 - Organizzazione della comunicazione e della cooperazione


Dipartimento di Informatica, Università di Pisa11/22



Amministrazione di progetto
Organizzazione di una norma

- **Regole**
 - Convenzioni di cui si riconosce necessità e convenienza
 - Ne è richiesto ed accertato il rispetto
- **Raccomandazioni**
 - Prassi desiderabile di buona programmazione
 - Inviti e suggerimenti, ma 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 Pisa12/22

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 Pisa13/22

Amministrazione di progetto

Convenzioni sui nomi

- Nel codice e nel progetto**
 - Tipi, costanti, variabili, funzioni, ...
 - Esempio: le norme Javadoc (vedi: <http://java.sun.com/2se/javadoc/>)
 - Strutturazione in moduli, *directory*, *file*, ...
- Aspetti pratici**
 - Conflitti, all'interno o all'esterno del codice
 - Abbreviazioni, per comodità o per necessità
 - Limiti del linguaggio
 - P.es.: identificazione forte o debole dei tipi (Java ↔ C)
 - Limiti degli strumenti
 - P.es.: lunghezza massima degli identificatori (p.es.: Windows 95-98)


Dipartimento di Informatica, Università di Pisa14/22

Amministrazione di progetto

Indentazione

- 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 Pisa15/22


Amministrazione di progetto

Intestazione

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

<input type="radio"/> Dati dell'unità	tipo, contenuto, posizione
<input type="radio"/> Responsabilità	autore, reparto, organizzazione
<input type="radio"/> <i>Copyright</i> o <i>copyleft</i>	licenze, visibilità
<input type="radio"/> Avvertenze	limiti di uso e di garanzia
<input type="radio"/> Registro modifiche	storia, spiegazione, versioni


Dipartimento di Informatica, Università di Pisa16/22

Amministrazione di progetto

Uso del linguaggio

- Una strategia 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 Pisa17/22

Amministrazione di progetto

Leggibilità del codice

- Il codice illeggibile è disarmante ed 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 Pisa18/22

