




## Amministrazione di progetto

IS 2001-8  
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'amministrazione
  - Non compie scelte tecnologiche
  - 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 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
  - 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 5/27




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 dell'ambiente
  - 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 intralci

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
  - **Postazioni di lavoro e dispositivi di utilità**
    - Per assicurare la massima produttività
  - **Archivi fisici**
    - Documentazione cartacea
- ❑ **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 Pisa7/27




Amministrazione di progetto

## Strumenti di processo

- ❑ **Gestione di progetto**
  - **Pianificazione e stima dei costi**
    - Diagrammi Gantt e PERT (<http://gantproject.biz/>)
  - **Allocazione e gestione delle risorse**
- ❑ **Analisi e progettazione**
  - **Ambienti di supporto alle metodologie**
    - UML (<http://topcased-mm.gforge.enseeiht.fr/website/modeling/uml/index.html>)
  - **Tracciamento dei requisiti**
    - Organizzato come una piccola base dati
- ❑ **Supporto alla realizzazione e all'automazione**
  - **Misurazione e analisi del codice**
  - **Generazione ed esecuzione automatica delle prove**
  - **Versionamento e configurazione**
    - Più facile se tutto integrato in un solo ambiente (<http://www.eclipse.org/>)

Dipartimento di Informatica, Università di Pisa8/27



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 Pisa9/27



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 accurata**
  - **Tracking e/o ticketing**
    - Per esempio con Bugzilla
  - **Stato corrente ed eventuale esito chiusura**

Dipartimento di Informatica, Università di Pisa10/27



Amministrazione di progetto

## Strumenti di sviluppo

- ❑ **Non solo compilatore e *debugger!***
- ❑ **Produrre codice secondo regole**
  - Editore integrato con verificatore di 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 Pisa11/27



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 Pisa12/27

Amministratozione di progetto

## Versionamento – 2

```
graph LR; V1.0 --> V1.1a; V1.0 --> V1.1; V1.1a --> V2.0; V1.1 --> V1.1b; V1.1 --> V1.1.1; V1.1b --> V1.1.1; V1.1.1 --> V2.0; V2.0 --> V2.1; V2.0 --> V2.2;
```

Tratto da: Ian Sommerville, *Software Engineering*, 8<sup>a</sup> ed.

Dipartimento di Informatica, Università di Pisa 13/27

Amministratozione di progetto

## Configurazione

- ❑ **Un prodotto software è l'unione di parti 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

Amministratozione 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

Amministratozione 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, 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 Pisa 16/27

Amministratozione 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

Amministratozione di progetto

## Convenzioni sui nomi

- ❑ **Nel codice e nel progetto**
  - Tipi, costanti, variabili, funzioni, ...
    - Esempio: le norme Javadoc (vedi: <http://java.sun.com/j2se/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 Pisa 18/27

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 Pisa19/27


Amministrazione di progetto

## Intestazione

- 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, versione

Dipartimento di Informatica, Università di Pisa20/27

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 Pisa21/27


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<sup>a</sup> ed.

Dipartimento di Informatica, Università di Pisa22/27

Amministrazione di progetto

## Uso del linguaggio

- 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 Pisa23/27

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 Pisa24/27

