

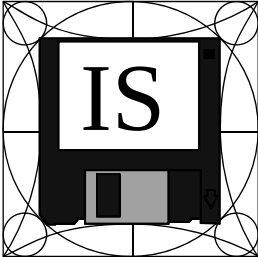


# Amministrazione di progetto

Ingegneria del Software

V. Ambriola, G.A. Cignoni,  
C. Montangero, L. Semini

Aggiornamenti: T. Vardanega (UniPD)



Dipartimento di Informatica, Università di Pisa1 / 32



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 Pisa2 / 32



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
    - La loro approvazione spetta al responsabile di progetto
  - Accertamento di applicazione delle regole
  - 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 Pisa3 / 32



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 Pisa4 / 32



Amministrazione di progetto

Disponibilità e diffusione

- I documenti sono utili solo se sono sempre 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 Pisa5 / 32



Amministrazione di progetto

Ambiente di lavoro

- Quanto serve ai processi di 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 Pisa6 / 32



Amministrazione di progetto

**Infrastruttura**

- ❑ **Risorse HW**
  - **Server**
    - Archivi logici centralizzati di prodotti («*repository*») sempre aggiornati e accessibili
  - **Infrastruttura di rete**
    - Sempre operativa, protetta, accessibile agli autorizzati anche da remoto
  - **Postazioni di lavoro e dispositivi di utilità**
    - Per assicurare la massima produttività individuale
  - **Archivi fisici**
- ❑ **Risorse SW**
  - Ambienti di sviluppo, prova, studio e documentazione
- ❑ **La gestione dei servizi dell'infrastruttura è spesso delegata all'amministrazione di sistema**
  - Funzione aziendale

Dipartimento di Informatica, Università di Pisa

7 / 32



Amministrazione di progetto

**Supporto di processi – 1**

- ❑ **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/>)
  - **Strumenti collaborativi di controllo gestionale e di qualità e di coordinamento attività**
    - Assembla (<http://www.assembla.com/>)
    - Maven (<http://maven.apache.org/>)
    - Jira (<http://www.atlassian.com/software/jira>)
- ❑ **Gestione documentale**
  - TWiki (<http://www.twiki.org/>)
  - Google Docs (<http://docs.google.com/>)
  - Versionamento e configurazione

Dipartimento di Informatica, Università di Pisa

8 / 32



Amministrazione di progetto

**Supporto di processi – 2**

- ❑ **Analisi e progettazione**
  - **Analisi, gestione e tracciamento dei requisiti**
    - eRequirements (<http://erequirements.com/app/>)
  - **Supporto alle metodologie**
    - UML (p.es., <http://www.eclipse.org/modeling/mdt/papyrus/>)
- ❑ **Codifica e integrazione**
  - Ambienti integrati di sviluppo (p.es., <http://www.eclipse.org/>)
  - **Strumenti di integrazione continua (*continuous integration*)**
    - Hudson (<http://hudson.dev.java.net/>)
    - CruiseControl (<http://cruisecontrol.sourceforge.net/index.html>)
    - Merlin ToolChain (<http://merlintoolchain.sourceforge.net/>)
  - **Misurazione e analisi statica del codice prima dell'integrazione**
  - **Generazione ed esecuzione automatica delle prove prima dell'integrazione**

Dipartimento di Informatica, Università di Pisa

9 / 32



Amministrazione di progetto

**Configurazione**

- ❑ **Un prodotto SW 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
- ❑ **La gestione di configurazione va automatizzata**
  - **Servono strumenti adatti**
    - Configuration Management
    - Build

Dipartimento di Informatica, Università di Pisa

10 / 32



Amministrazione di progetto

**Gestione di configurazione – 1**

- ❑ **Obiettivi**
  - Mettere in sicurezza le *baseline*
  - Prevenire sovrascritture accidentali
  - Consentire ritorno a configurazioni precedenti
  - Permettere il recupero da perdite accidentali
- ❑ **Attività**
  - Identificazione di configurazione
  - Controllo di *baseline*
  - Gestione delle modifiche
  - Controllo di versione

Dipartimento di Informatica, Università di Pisa

11 / 32



Amministrazione di progetto

**Gestione di configurazione – 2**

- ❑ **Identificazione di configurazione**
  - Decidere quali sono le parti (*configuration item*, CI) che compongono il prodotto
  - **Attribuire una identità unica a ciascun CI**
    - ID, nome, data, autore, registro delle modifiche, stato corrente
- ❑ **Controllo di *baseline***
  - Un insieme di CI a una specifica *milestone*
  - Base verificata, approvata e garantita per la prosecuzione dello sviluppo
  - L'esistenza di *baseline* ben identificate permette
    - Riproducibilità
    - Tracciabilità
    - Analisi e confronto

Dipartimento di Informatica, Università di Pisa

12 / 32



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

13/32



Amministrazione di progetto

Gestione delle modifiche – 2

- Ogni richiesta/proposta di modifica va inoltrata in modo formale
  - Change request
    - 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

14/32



Amministrazione di progetto

Controllo di versione – 1

- Si appoggia su un repository
  - DB centralizzato nel quale risiedono - individualmente - tutti i CI di ogni baseline nella loro storia completa
- Permette a ciascuno di lavorare su vecchi e nuovi CI senza rischio di sovrascritture accidentali
  - Checkout
- E di condividere il lavorato nello spazio comune
  - Checkin
- Verifica la bontà di ogni modifica di baseline
  - Build

Dipartimento di Informatica, Università di Pisa

15/32



Amministrazione di progetto

Controllo di versione – 2

- 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

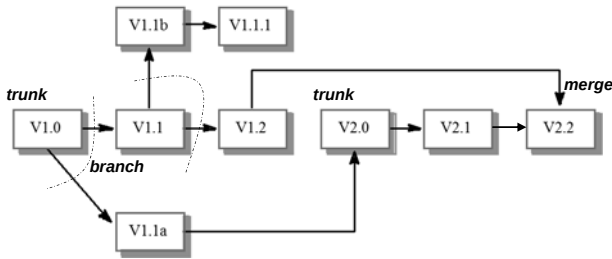
Dipartimento di Informatica, Università di Pisa

16/32



Amministrazione di progetto

Controllo di versione – 3



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

Dipartimento di Informatica, Università di Pisa

17/32



Amministrazione di progetto

Strumenti di lavoro – 1

- 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
  - SVN (subversion, <http://subversion.tigris.org/>)
  - Git (<http://git-scm.com/>)
- Configurazione (build) per integrare le parti del prodotto finale
  - Ant (<http://ant.apache.org/>)
  - Maven (<http://maven.apache.org/>)

Dipartimento di Informatica, Università di Pisa

18/32



Amministrazione di progetto

Strumenti di lavoro – 2

Amministrazione di progetto (parte 2)  
Per approfondire #12: [Uno strumento di gestione progetto](#)  
Per approfondire #13: [Strumenti di collaborazione](#)  
Per approfondire #14: [Strumenti di gestione del ciclo di vita del software](#) (Ambiente Java, [Guida](#))  
Per approfondire #15: [Strumenti di gestione del ciclo di vita del software](#) [8MB] (Ambiente Microsoft)

Dipartimento di Informatica, Università di Pisa

19/32



Amministrazione di progetto

Norme di progetto

Linee guida per le attività di sviluppo

- In origine precedevano processi e procedure aziendali
- Oggi sono uno strumento operativo di complemento alle procedure

Contenuti

- Organizzazione e uso delle risorse di sviluppo
- Convenzioni sull'uso degli strumenti di sviluppo
- Organizzazione della comunicazione e della cooperazione
- Norme di codifica
- Gestione dei cambiamenti!

Dipartimento di Informatica, Università di Pisa

20/32



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

21/32



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

22/32



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

23/32



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

Evitare guerre ideologiche sugli stili

Dipartimento di Informatica, Università di Pisa

24/32





Amministrazione di progetto

Notazione ungherese ☺

❑ 1pfnWndProc (un campo di struttura)

○ 1long (32-bit integer)

○ ppointer [to a]

○ fnfunction [handling messages directed to]

○ Wnd[a] window

○ Procprocedure

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



❑ Un'idea di Charles Simonyi @ Microsoft

Dipartimento di Informatica, Università di Pisa

31/32



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

❑ F. Lanubile et al., "Collaboration Tools for Global Software Engineering", IEEE Software, 27:2, 2010, 52-55

❑ J. Portillo Rodriguez et al., "Technologies and Tools for Distributed Teams", IEEE Software, 27:5, 2010, 10-14

Dipartimento di Informatica, Università di Pisa

32/32