

Corso di Laurea in Informatica - Ingegneria del Software 2 1



6. Documentazione

Docente: Tullio Vardanega
tullio.vardanega@math.unipd.it

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 2



Contenuto della lezione

- ◆ Perché documentare
 - ◆ Processo di *supporto* secondo ISO/IEC 12207
- ◆ Cosa documentare
 - ◆ Attività e prodotti da *pianificare, eseguire, verificare, correggere* (ciclo PDCA)
 - ◆ Secondo gli standard di processo applicabili o richiesti
- ◆ Come documentare
 - ◆ Contenuti attesi
 - ◆ Ai fini di revisione
 - ◆ Contenuti rilevanti
 - ◆ Ai fini di pianificazione ed esecuzione

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 3



Documentazione come processo di supporto: Perché - 1

- ◆ Ingegneria del software
 - ◆ Applicazione di principi ingegneristici allo sviluppo, l'uso e la manutenzione del software
 - ◆ Processi primari
 - ◆ Adozione di un approccio *sistematico, disciplinato, quantificabile*
 - ◆ Comporta esecuzione di processo/i di gestione
 - ◆ Processo organizzativo secondo ISO/IEC 12207
 - ◆ Pianificazione, coordinamento, monitoraggio e misurazione, controllo e rendiconto

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 4



Documentazione come processo di supporto: Perché - 2

- ◆ Complessità inerente dei processi produttivi
 - ◆ Volatilità dei requisiti
 - ◆ Processi internamente iterativi più spesso che rigidamente sequenziali
 - ◆ Delicato bilanciamento tra creatività e disciplina
 - ◆ Mancanza di una teoria matematica o fisica di riferimento
 - ◆ Rapida evoluzione della tecnologia di supporto

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 5



Documentazione come processo di supporto: Perché - 3

- ◆ Il processo gestionale richiede elementi di misurazione
 - ◆ Sia quantitativa che qualitativa
- ◆ La gestione della comunicazione elemento essenziale dei processi organizzativi
- ◆ Le attività di processi che seguono lo schema PDCA debbono essere ripetibili e misurabili

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 6



Documentazione come processo di supporto: Perché - 4

- ◆ Cosa misurare?
 - ◆ Non tutto indistintamente
 - ◆ Solo ciò che consente di focalizzare il processo (organizzativo) di miglioramento
 - ◆ Secondo obiettivi strutturali
 - ◆ Con effetto permanente
 - ◆ Secondo priorità assegnate dall'organizzazione
 - ◆ Obiettivi che vanno al di là del progetto (o prodotto)
 - ◆ Misurazione per obiettivi (*ad hoc*)
 - ◆ *Processi, prodotti e risorse* posseggono componenti misurabili

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 7



Documentazione come processo di supporto: Perché - 5

- ◆ Misurare è difficile
 - ◆ Alcune misure (metriche) sono essenziali
 - ◆ Dimensione del prodotto software
 - ◆ ISO/IEC 14143 Software engineering – Software measurement – Functional size measurement (1998)
 - ◆ ISO/IEC 14598 Software product evaluation (1998)
 - ◆ Struttura del prodotto software
 - ◆ Flusso di controllo, flusso dei dati, annidamento, modularità ed interazione
 - ◆ Uso delle risorse
 - ◆ Strumenti, hardware (p.es.: memoria), personale
 - ◆ Qualità del prodotto
 - ◆ ISO/IEC 9126 Software product quality (1999-2001)

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 8



Documentazione come processo di supporto: Perché - 6

- ◆ Trattamento dei dati di misurazione
 - ◆ Selezionare l'insieme *ottimale* di misure
 - ◆ Quelle di *maggior uso potenziale* (a scopo di previsione) secondo gli obiettivi fissati
 - ◆ A costo *contenuto* di determinazione e proporzionato ai benefici attesi
 - ◆ Occorrono modelli d'uso
 - ◆ Dei dati di misurazione e della conoscenza ad essi associata
 - ◆ Ai fini di *analisi, classificazione e previsione*
 - ◆ I dati vanno valutati
 - ◆ I modelli vanno calibrati
 - ◆ Durante e dopo il progetto

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 9



Contenuti della documentazione: Cosa - 1

- ◆ Modello software
 - ◆ Descrizione *semplificata* del sistema
 - ◆ *Gerarchico*, secondo criteri congruenti di decomposizione
 - ◆ Composto di simboli organizzati secondo una *data convenzione*
 - ◆ Costruito mediante metodi e strumenti riconosciuti (*standard*)
 - ◆ Usato per *ragionare* sul software da sviluppare (sviluppato)

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 10



Contenuti della documentazione: Cosa - 2

- ◆ Modello logico → ST
 - ◆ Prodotto nella fase di ingegneria dei requisiti
 - ◆ Indipendente dall'implementazione
 - ◆ Mostra ciò che il sistema deve fare
 - ◆ È organizzato gerarchicamente attraverso livelli di astrazione successivi
 - ◆ Consente di stabilire relazioni tra cause ed effetti
 - ◆ Consente di comprendere i requisiti utente nel loro insieme (prima che individualmente)

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 11



Contenuti della documentazione: Cosa - 3

- ◆ Decomposizione funzionale
 - ◆ Primo passo (*top-down*) per la produzione del modello logico
 - ◆ Funzioni/entità con *un solo* obiettivo e criticità definita
 - ◆ Elevata coesione
 - ◆ Congruenti al livello di astrazione al quale appaiono
 - ◆ Con il minimo numero possibile di interfacce
 - ◆ Basso grado di accoppiamento
 - ◆ Misurabile in termini di
 - ◆ Servizi esportati (a quante entità distinte)
 - ◆ Servizi importati (da quante entità distinte)
 - ◆ A grado di decomposizione limitato

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 12



Contenuti della documentazione: Cosa - 4

- ◆ Decomposizione ad oggetti
 - ◆ Procede dai requisiti utente e produce
 - ◆ Modello ad oggetti (struttura *statica* del sistema)
 - ◆ Oggetti e classi con attributi ed associazioni
 - ◆ Aggregazione (i.e.: è una parte di)
 - ◆ Generalizzazione / specializzazione (i.e.: è un tipo di)
 - ◆ Ereditarietà strumento di organizzazione e semplificazione della struttura delle classi
 - ◆ Modello dinamico
 - ◆ Comportamento del sistema e sequenza delle interazioni tra i suoi componenti
 - ◆ Modello funzionale
 - ◆ Identifica i valori in ingresso ed in uscita
 - ◆ Mostra il flusso dei dati (attraverso gli oggetti) che trasforma gli ingressi in uscite

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 13

**Contenuti della documentazione:
Cosa - 5**

◆ Modello fisico → DP **DCI** Omesso nel progetto didattico 2003

- ◆ Proceda dal modello logico
- ◆ Consente lo sviluppo *parallelo ed indipendente* dei componenti terminali (di basso livello)
- ◆ Consente di stimare lo sforzo (costo, tempi) di realizzazione
- ◆ È valutabile mediante metriche
 - ◆ Coesione, accoppiamento, utilità (*fan-in*), dipendenza (*fan-out*), complessità

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 14

**Contenuti della documentazione:
Cosa - 6**

◆ Disegno architetturale

- ◆ Identifica, per ogni componente del sistema
 - ◆ Funzione svolta
 - ◆ Strutture dati e flussi di controllo
 - ◆ Dati in ingresso
 - ◆ Dati in uscita
 - ◆ Risorse utilizzate

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 15

**Contenuti della documentazione:
Cosa - 7**

◆ Disegno di dettaglio

- ◆ Proceda dal disegno architetturale
 - ◆ Decompone in moduli (determinati dal linguaggio di implementazione) fino a quando
 - ◆ Ogni modulo ha lunghezza, complessità, coesione ed accoppiamento accettabili
 - ◆ È influenzato da esigenze ed opportunità di riuso

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 16

**Contenuti della documentazione:
Cosa - 8**

◆ Per ogni modulo

- ◆ Intestazione
 - ◆ Titolo
 - ◆ Identificatore del corrispondente elemento di configurazione
 - ◆ Autore
 - ◆ Data di creazione
 - ◆ Registro delle modifiche
- ◆ Comprensibilità del codice
 - ◆ Variabili dichiarate e con nomi espressivi
 - ◆ Evitare variabili temporanee ed ambiguità espressive e logiche
 - ◆ Formato e commenti per leggibilità

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 17

**Struttura della documentazione:
Come - 1**

- ◆ Caratteristiche generali
 - ◆ Chiarezza, consistenza, modificabilità
- ◆ Caratteristiche specifiche (DP)
 - ◆ Tipo: caratteristiche logiche e fisiche del modulo
 - ◆ Obiettivo: in relazione ai requisiti software
 - ◆ Funzione: ciò che il modulo fa
 - ◆ Relazioni d'uso (in uscita ed in entrata): flussi di controllo e flusso dei dati
 - ◆ Meccanismi e modalità di invocazione
 - ◆ Attività svolte
 - ◆ Dati trattati: per ogni struttura dati
 - ◆ Descrizione di ciascun elemento (nome, tipo, dimensione, rango), relazione tra elementi, valore iniziale

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 18

**Struttura della documentazione:
Come - 2**

- ◆ Caratteristiche generali
 - ◆ Frasi brevi, paragrafi brevi e focalizzati, forma attiva, correttezza grammaticale
 - ◆ Adatto alle caratteristiche dell'utente
 - ◆ Adatto alle caratteristiche dell'interfaccia utente
- ◆ Caratteristiche specifiche (MU)
 - ◆ Evoluzione: nasce presto e cresce con il prodotto
 - ◆ Forma
 - ◆ Documento cartaceo tradizionale
 - ◆ Documento ipertestuale
 - ◆ Documento in linea al prodotto

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 19

Struttura della documentazione: Come - 3

- ◆ Tracciamento
 - ◆ Documenta la relazione tra due o più prodotti del processo di sviluppo
 - ◆ In avanti (*forward*) → completezza
 - ◆ Ciascun ingresso ad una fase deve essere messo in relazione con una uscita di quella fase
 - ◆ Mediante matrici di tracciabilità (base dati)
 - ◆ Evidenziano incompletezza e duplicazione
 - ◆ All'indietro (*backward*) → necessità
 - ◆ Ciascuna uscita di una fase deve essere messa in relazione con ingresso a quella fase
 - ◆ Mediante matrici di tracciabilità
 - ◆ Componenti non tracciate/tracciabili sono superflue (a meno di omissioni all'ingresso)

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 20

Struttura della documentazione: Come - 4

- ◆ Tracciamenti necessari
 - ◆ Requisiti utente ↔ requisiti software
 - ◆ Requisiti software ↔ descrizione di componenti
 - ◆ Test di unità ↔ moduli di disegno di dettaglio
 - ◆ Test di integrazione ↔ componenti architetturali
 - ◆ Test di sistema ↔ requisiti software
 - ◆ Test di accettazione ↔ requisiti utente

Documentazione - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 21

Struttura della documentazione: Come - 5

Requisito utente	Requisito software	Componente/Modulo	Test d'unità
RU-1.1.1.1.1	RS-2.4.6.3	DA-7.3.2.1	TU-1.1.1.1
	RS-2.4.6.4		TU-1.1.1.2
	RS-5.1.9.7		

Livello 1
Livello 2
Livello 3

Documentazione - Tullio Vardanega - 2003

