Documentazione

IS

Anno accademico 2007/8
Ingegneria del Software mod. A

Tullio Vardanega, tullio.vardanega@math.unipd.it

Domande ricorrenti

- **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

Perché documentare – 1

- **Ingegneria del *software***
  - **Applicazione di principi ingegneristici allo sviluppo, l'uso e la manutenzione del *software***
    - Sviluppo, uso operativo e manutenzione sono processi primari di ISO/IEC 12207
  - **Adozione di un approccio sistematico, disciplinato, quantificabile**
  - **Comporta esecuzione di processi di gestione**
    - Istanziati da processi organizzativi secondo ISO/IEC 12207
      - Pianificazione, coordinamento, misurazione, controllo, analisi e correzione
      - Ciclo PDCA

Perché documentare – 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**

Perché documentare – 3

- **Il processo gestionale richiede elementi di misurazione**
  - **Quantitativa**
  - **Qualitativa**
- **La gestione della comunicazione come elemento essenziale dei processi organizzativi**
- **Le attività di processi che seguono lo schema PDCA devono essere ripetibili e misurabili**

Cosa misurare – 1

- **Non serve misurare tutto indistintamente**
- **Focalizzarsi su quanto che serva 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 attributi misurabili**

Università degli Studi di Padova

Documentazione

Cosa misurare – 2

- **Metriche essenziali**
  - **Dimensione del prodotto**
    - ISO/IEC 14143 *Software engineering – Software measurement – Functional size measurement* (1998)
    - ISO/IEC 14598 *Software product evaluation* (1998)
  - **Struttura del prodotto**
    - Flusso di controllo, flusso dei dati, annidamento, modularità e interazione
  - **Uso delle risorse**
    - Risorse tecniche (strumenti), risorse fisiche e logiche (spazio di memoria, tempo d'esecuzione), risorse umane (personale)
  - **Qualità del prodotto**
    - ISO/IEC 9126 *Software product quality* (1999-2001)

Università degli Studi di Padova

Documentazione

Cosa misurare – 3

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

Università degli Studi di Padova

Documentazione

Cosa documentare – 1

- **Modello *software***
  - **Descrizione semplificata del sistema**
  - **Visione gerarchica**
    - Secondo criteri congruenti di decomposizione
  - **Realizzato mediante uso di simboli e notazioni organizzate secondo una convenzione fissata e coerente**
    - P.es.: UML
  - **Costruito mediante metodi e strumenti standard**
    - Standard aziendale, di fatto, internazionale (meglio)
  - **Usato per ragionare sul *software* da sviluppare**
    - Anche sull'esito dello sviluppo

Università degli Studi di Padova

Documentazione

Modelli architetturali

- **Un'architettura *software* ha diverse dimensioni di interesse**
  - **Modello strutturale statico**
    - Identifica le componenti principali
    - Procede per decomposizione gerarchica
  - **Modello dinamico**
    - Illustra la struttura "a processi" del sistema
  - **Modello delle interfacce**
    - Definisce le interfacce fornite / richieste da / tra componenti del sistema
  - **Modello delle relazioni**
    - Identifica il flusso dei dati tra componenti distinti in relazione tra loro
  - **Modello di distribuzione**
    - Mostra l'associazione tra nodi fisici e componenti logiche

Università degli Studi di Padova

Documentazione

Cosa documentare – 2

- **Architettura logica → ST**
  - **Prodotta al termine della fase di ingegneria dei requisiti**
    - Fissa linee e strategie di realizzazione
    - Avvia la fase realizzativa (ingegneria di progetto)
    - **Non fissa** gli aspetti realizzativi concreti
  - **Mostra ciò che il sistema deve fare**
  - **È organizzata gerarchicamente attraverso livelli di astrazione ↑ (o decomposizione ↓) successivi**
  - **Consente di stabilire relazioni tra cause ed effetti**
  - **Offre una visione d'insieme della soluzione proposta al problema complessivo**

Università degli Studi di Padova

Documentazione

Decomposizione funzionale

- **Primo passo (*top-down* ↓) per la produzione dell'architettura logica**
  - **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)
  - **Profondità di decomposizione limitata**

Università degli Studi di Padova

Documentazione

Decomposizione a oggetti

- **In continuità logico-notazionale con l'analisi dei requisiti OO**
- **Modello statico**
  - **Classi e oggetti con attributi e associazioni**
    - Aggregazione (i.e.: A è una parte di B)
    - Generalizzazione / specializzazione (i.e.: C è un tipo di D)
  - **Ereditarietà come strumento di organizzazione, semplificazione e riuso della struttura delle classi**
- **Modello dinamico**
  - **Comportamento del sistema e sequenza delle interazioni tra i suoi componenti**
- **Modello funzionale**
  - **Identifica i valori in ingresso / uscita**
  - **Mostra il flusso dei dati (attraverso gli oggetti) che trasforma gli ingressi in uscite**

Università degli Studi di Padova

Documentazione

Cosa documentare – 3

- **ST → progetto (*design*) architetturale**
  - **Specifica per ogni componente del sistema**
    - Funzione svolta
      - Strutture dati utilizzate
      - Flussi di controllo impiegati
    - Dati in ingresso (tipo)
    - Dati in uscita (tipo)
    - Risorse logiche e fisiche necessarie

Università degli Studi di Padova

Documentazione

Cosa documentare – 3

- **Architettura fisica → DP**
  - **Procede dall'architettura logica**
  - **Consente sviluppo *parallelo* e *indipendente* dei componenti terminali (di basso livello)**
  - **Consente di stimare lo sforzo (costo, tempi) di realizzazione**
  - **Ha qualità valutabile mediante precise metriche**
    - Coesione
    - Accoppiamento
    - Utilità (*fan-in*)
    - Dipendenza (*fan-out*)
    - Complessità

Università degli Studi di Padova

Documentazione

Cosa documentare – 4

- **DP → progetto (*design*) di dettaglio**
  - **Procede dal progetto architetturale**
  - **Decompone le componenti architetturali in moduli a grana più fine finché**
    - Ogni modulo ha **dimensione**, **complessità**, **coesione** e **accoppiamento** adeguati
  - **È influenzato da esigenze e opportunità di riuso**
    - La natura dei "moduli" è fissata dal supporto offerto dal linguaggio di programmazione selezionato per la codifica
      - Modulo ≠ *file* !

Università degli Studi di Padova

Documentazione

Cosa documentare – 5

- **Per ogni modulo**
  - **Intestazione**
    - Titolo (nome logico del modulo)
    - Identificatore del corrispondente **elemento di configurazione** e versione
    - Autore
    - Data di creazione della versione corrente
    - Registro delle modifiche
  - **Comprensibilità del codice**
    - Variabili dichiarate e con nomi espressivi
    - Evitare variabili temporanee e ambiguità espressive e logiche
    - Formato e commenti per massima leggibilità

Università degli Studi di Padova

Documentazione

Documento DP

- **Caratteristiche generali**
  - **Chiarezza espressiva, consistenza logica e terminologica, modificabilità**
- **Caratteristiche specifiche**
  - **Tipo: caratteristiche logiche e fisiche del modulo**
    - Con / senza flusso di controllo
    - Con / senza metodi sincronizzati
    - Con condizioni logiche di attesa / risveglio
    - Associato a risorse fisiche (dispositivi)
  - **Obiettivo: in relazione ai requisiti *software***
  - **Funzione: ciò che il modulo fa**
  - **Relazioni d'uso in uscita e 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

Università degli Studi di Padova

Documentazione

Documento MU

- **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**
  - **Evoluzione**
    - Nasce presto e cresce con il prodotto
  - **Forma**
    - Documento cartaceo tradizionale
    - Documento ipertestuale
    - Documento (ipertestuale) in linea al prodotto
    - Aiuto contestuale

Università degli Studi di Padova

Documentazione

Tracciamento dei requisiti – 1

- Fissa la relazione tra i prodotti del processo di sviluppo
- **In avanti (*forward*) → completezza**
  - Ciascun ingresso a una fase deve essere messo in relazione con una specifica uscita di quella fase
  - **Mediante matrici di tracciabilità**
    - Una sorta di base dati
    - Evidenziano incompletezza e duplicazione
- **All'indietro (*backward*) → necessità**
  - Ciascuna uscita di una fase deve essere messa in relazione con uno specifico ingresso a quella fase
  - **Mediante matrici di tracciabilità**
    - Le componenti non tracciate o non tracciabili sono **superflue** e da eliminare (a meno di omissioni all'ingresso)

Università degli Studi di Padova

Documentazione

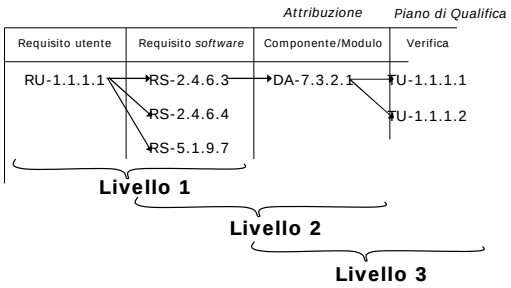
Tracciamento dei requisiti – 2

- **Tracciamenti necessari**
  - Requisiti utente (capitolato) ↔ requisiti *software* (AR)
  - Requisiti *software* (AR) ↔ descrizione di componenti (ST)
  - Test di unità ↔ moduli di disegno di dettaglio (DP)
  - Test di integrazione ↔ componenti architetturali (ST)
  - Test di sistema ↔ requisiti *software* (AR)
  - Test di accettazione ↔ requisiti utente (capitolato)

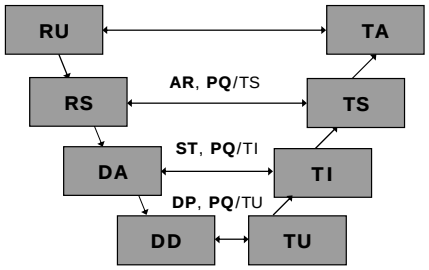
Università degli Studi di Padova

Documentazione

Tracciamento dei requisiti – 3



Università degli Studi di Padova

Documentazione

Architettura della documentazione