



Premesse al corso



Anno accademico 2012/13
Ingegneria del Software mod. A
Tullio Vardanega, tullio.vardanega@math.unipd.it

Corso di Laurea in Informatica, Università di Padova 1/30



Premesse al Corso

Cominciamo da lontano ...

- **Engineering** (fonte: American Heritage Dictionary)
 - *Application of scientific and mathematical principles to practical ends*
 - Questi "practical ends" sono civili e sociali (infrastrutture, servizi) e non solo di consumo ⇒ responsabilità
- **Cosa succede quando questa definizione si applica al software?**
 - Per rispondere occorre capire bene a quali discipline e principi fare riferimento

Corso di Laurea in Informatica, Università di Padova 2/30



Premesse al Corso

Relazione con altre discipline – 1

- ***The scientific disciplines of reference to Software Engineering include not only computer science but also certain areas of discrete mathematics and operation research, statistics, psychology and economics***
- ***Software Engineering isn't a branch of computer science; it's an engineering discipline relying in part on computer science, in the same way that mechanical engineering relies on physics***

Fonte: Lionel Briand, IEEE Software 49(4), 93-95

Corso di Laurea in Informatica, Università di Padova 3/30



Premesse al Corso

Relazione con altre discipline – 2

- **L'ingegneria del software ha strette relazioni con**
 - **Informatica**
 - Programmazione e Linguaggi di Programmazione
 - Architettura degli Elaboratori, Sistemi Operativi, Basi di Dati, Reti e Sicurezza
 - **Matematica**
 - Calcolo Numerico, Matematica Discreta, Ricerca Operativa, Statistica
 - **Scienze gestionali ed Economia**
 - Del tempo, delle risorse, delle persone
 - **Ingegneria**
 - Dei sistemi, elettronica
 - **Psicologia**
 - Il fattore umano nella comprensione e nell'apprendimento

Corso di Laurea in Informatica, Università di Padova 4/30



Ingegneria del software – 1

- **Disciplina per la realizzazione di sistemi SW così impegnativi da richiedere lavoro di gruppo**
 - Capacità di produrre "in grande" oltre che "in piccolo"
 - Assicurare la qualità dei prodotti (efficacia)
 - Contenere i costi e i tempi di produzione (efficienza)
 - Lungo l'intero ciclo di vita del prodotto
- **Esistono svariate tipologie di prodotti SW**
 - **Su commessa:** forma, contenuto e funzione fissate dal cliente
 - **Pacchetto:** forma, contenuto e funzione idonee alla replicazione
 - **Componente:** forma, contenuto e funzione adatte alla composizione
 - **Servizio:** forma, contenuto e funzione fissate dal problema

Parole chiave – 1

- **Ciclo di vita del SW**
 - Gli stati che il prodotto assume dal concepimento al ritiro
- **Efficienza**
 - Inversamente proporzionale alla quantità di risorse impiegate nell'esecuzione delle attività richieste
- **Efficacia**
 - Determinata dal grado di conformità del prodotto rispetto alle norme vigenti e agli obiettivi prefissati



Ingegneria del software – 2

- **Durante il proprio ciclo di vita_{1g1} molti sistemi vengono sottoposti a svariate forme di manutenzione**
 - **Correttiva:** per correggere difetti eventualmente rilevati
 - **Adattativa:** per adattare il sistema alla variazione dei requisiti
 - **Evolutiva:** per aggiungere funzionalità al sistema
- **La manutenibilità è una qualità essenziale del SW**
 - Come ottenerla, come garantirla?
 - *In the large, in the small*



Ingegneria del software – 3

- **I progetti SW_{1g1} hanno spesso esito insoddisfacente**
 - Difficoltà nelle fasi iniziali: ritardi e maggior costo
 - Cambi di piattaforma e tecnologia non pianificati
 - Difetti residui nel prodotto finale
- **A volte falliscono clamorosamente**
 - Per obsolescenza prematura
 - Per incapacità o impossibilità tecnica di completare
 - Per esaurimento dei tempi o dei finanziamenti



Ingegneria del software – 4

- Riuscire a soddisfare obiettivi prefissati entro limiti certi di tempo e di sforzo**
 - Massimizzare efficacia ed efficienza
- Vogliamo applicare principi ingegneristici alla produzione del SW**
 - Mancano base matematica solida e parametri tecnici certi
 - Ciclo virtuoso "esperienza ↔ sistematizzazione"
 - Ciò che si chiama "*best practice*"
- Conferenza NATO a Garmisch (D)**
 - 7-11 ottobre 1968: nasce la disciplina *Software Engineering*
 - Quindi ancora estremamente "giovane"



Parole chiave – 2

- Best practice**
 - Prassi (modo di fare) che per esperienza e per studio abbia mostrato di garantire i migliori risultati in circostanze note e specifiche
- Stakeholder (portatore/i di interesse)**
 - L'insieme di persone a vario titolo coinvolte nel ciclo di vita del SW con influenza sul prodotto



Definizione secondo IEEE

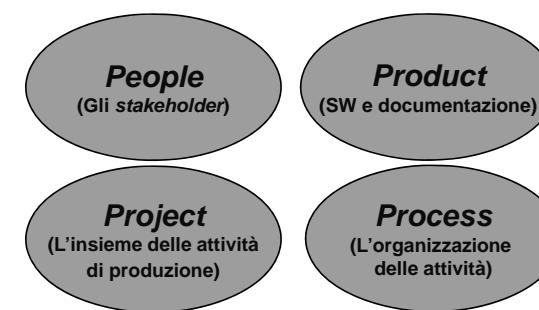
L'approccio sistematico, disciplinato e quantificabile allo sviluppo, l'uso, la manutenzione e il ritiro del SW

(*Glossario IEEE*)

- L'applicazione di principi ingegneristici al SW**
- Il SW è un prodotto con un proprio ciclo di vita**
 - La sua manutenzione costa spesso molto più della sua produzione
 - I costi di produzione sono spesso dominati dai costi di verifica
- Il controllo di efficienza ed efficacia ha bisogno di un approccio sistematico**



Le 4 P del Software Engineering





Premesse al Corso

People

- Business management**
 - Chi fissa gli obiettivi in termini di **costi, profitto, priorità strategiche**
 - Competizioni sul mercato, soddisfazione del cliente, efficienza, efficacia
- Project management**
 - Chi gestisce le risorse di progetto e riferisce all'organizzazione e al cliente
- Development team**
 - Chi realizza il prodotto: il luogo di appartenenza dei *software engineer*
- Customers**
 - Chi compra il prodotto SW
- End users**
 - Chi usa il prodotto SW

Corso di Laurea in Informatica, Università di Padova **13/30**



Premesse al Corso

Figure professionali – 1

- Software engineer ≠ programmatore**
- Il programmatore**
 - Figura professionale dominante negli anni pionieristici dell'informatica ('50-'70)
 - Scrive programmi per se stesso, da solo, sotto la propria responsabilità tecnica
 - Svolge un'attività creativa fortemente personalizzata

Corso di Laurea in Informatica, Università di Padova **14/30**



Premesse al Corso

Figure professionali – 2

- Il software engineer**
 - Realizza parte di un sistema complesso con la consapevolezza che potrà essere usato, completato e modificato da altri
 - Deve guardare e comprendere il quadro generale nel quale il sistema cui contribuisce si colloca
 - La dimensione "sistema" include ma non si limita al SW
 - Deve operare compromessi intelligenti e lungimiranti tra visioni e spinte contrapposte
 - Costi – qualità
 - Risorse (tempo, mezzi e competenze) – disponibilità
 - ...

Corso di Laurea in Informatica, Università di Padova **15/30**



Premesse al Corso

Principi etici di SW engineering

- Considerare la qualità come il primo obiettivo
- Produrre SW di alta qualità è possibile
- Aiutare il cliente a comprendere i suoi veri bisogni
- Adottare i processi più adatti al progetto
- Ridurre la distanza intellettuale tra il SW e il problema da risolvere
- Essere proattivi nel cercare e rimuovere gli errori
- Motivare, formare, far crescere le persone

Corso di Laurea in Informatica, Università di Padova **16/30**



Un principio chiave – 1

- **Frederick P. Brooks, Jr.**
Computer Magazine, aprile 1987
No Silver Bullet. Essence and Accidents of Software Engineering

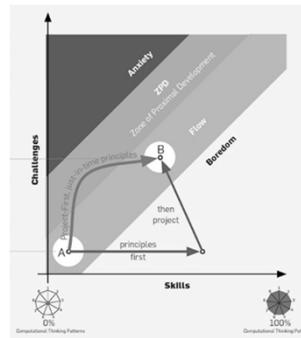
- **Distingue tra problematiche essenziali ...**
 - Specifica, realizzazione, verifica, manutenzione di prodotti SW
- **... e problematiche accidentali**
 - Gli strumenti e le tecniche per la rappresentazione e la verifica di accuratezza di rappresentazione delle problematiche essenziali

Un principio chiave – 2

- **L'evoluzione tecnica e tecnologica può rendere più agevole affrontare le problematiche accidentali**
- **Nessuna soluzione tecnica o tecnologica può esoneraci dall'impegno concettuale, di astrazione, di analisi, di rigore (ecc.!) necessario per affrontare le problematiche essenziali**



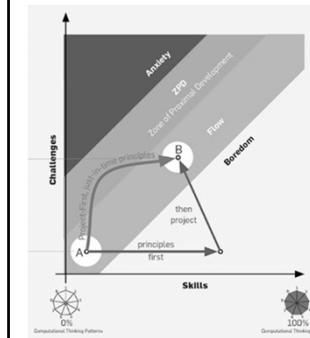
Progetti: approccio didattico – 1



- **X axis: the students' computational thinking skills**
 - **Y axis: the level of design challenge in STEM simulation**
 - **The project-first approach is in Zones of Proximal Flow (ZPF)**
 - **Where Flow is an ideal condition for learning**
 - **ZPF orchestrates students' take in best practices using forms of assistance and tool use**
- STEM = science, technology, engineering, and mathematics



Progetti: approccio didattico – 2



- **Student acquisition of skills advances in response to the challenges**
 - **An ideal path in the Flow region would progress from the origin to top right**
- **Pedagogical approaches can be described as instructional trajectories connecting a skill/challenge starting point (A) with destination point (B)**

D.C. Webb, A. Repenning, K.H. Koh, "Toward an emergent theory of broadening participation in computer science education", Proc. 43rd ACM Computer Science Education symposium, 173-178 (SIGCSE '12)



Premesse al Corso

Che cosa non è un progetto

- One is blinded to the fundamental uselessness of their products by the sense of achievement one feels in getting them to work at all***
- In other words, their fundamental design flaws are completely hidden by their superficial design flaws***

Douglas Adams, "The Hitchhikers Guide to the Galaxy", 1979

Corso di Laurea in Informatica, Università di Padova 21/30



Premesse al Corso

Che cosa è un progetto

- Pianificazione**
 - Organizzare e controllare tempo, risorse e risultati
- Analisi dei requisiti**
 - Definire cosa bisogna fare
- Progettazione**
 - Definire come bisogna farlo
- Realizzazione**
 - Farlo con la massima efficienza e la massima efficacia
- Verifica e validazione**
 - Assicurare che quanto fatto soddisfi i requisiti e non contenga errori
- Manutenzione**
 - Assicurare piena utilizzabilità del SW fino al momento del suo ritiro

Corso di Laurea in Informatica, Università di Padova 22/30



Premesse al Corso

La terza P: il processo

- Il quadro metodologico, normativo e strategico delle attività di progetto**
 - L'insieme delle attività raggruppate per obiettivi
 - Come tali attività sono correlate le une alle altre
 - Come attuarle e in quale ordine
 - I loro gradi di libertà
- Per alimentare svariati modelli di ciclo di vita del SW**
 - Per organizzare al meglio le attività necessarie all'interno di vincoli dati di tempo, di risorse e di obiettivi

Corso di Laurea in Informatica, Università di Padova 23/30



Premesse al Corso

Problematica essenziale

- Qualità**
 - Metodi di verifica e di controllo
 - Come verificare, come progettare le verifiche
 - Come garantire la qualità del prodotto finito
 - Come valutare la qualità dei processi impiegati
 - Modelli di qualità
 - Per valutare le caratteristiche salienti dei prodotti SW
 - Per valutare le capacità dei processi
 - Metriche
 - Unità di misura, scale di riferimento, strumenti di misurazione
 - Indicatori di qualità

Corso di Laurea in Informatica, Università di Padova 24/30



Premesse al Corso

Libro di testo – 1

- I libri esistenti si dividono in due categorie**
 - Teorici : trattano la materia in modo privo di riflessi di esperienza concreta
 - Principi esposti, ma troppo spesso non vissuti
 - Esperienziali : espongono l'esperienza degli autori, ma spesso senza relazionarla in modo convincente alla visione astratta del problema e della disciplina
 - Eccessiva enfasi sugli aspetti accidentali
- Non utilizzeremo direttamente né gli uni né gli altri**

Corso di Laurea in Informatica, Università di Padova

25/30



Premesse al Corso

Libro di testo – 2

- Faremo riferimento esplicito e implicito a**
 - Guide to the Software Engineering Body of Knowledge (SWEBOk)
IEEE Computer Society Software Engineering Coordinating Committee
<http://www.swebok.org>
- Lo SWEBOk ci aiuterà a familiarizzarci con le 10 aree di conoscenza della disciplina**

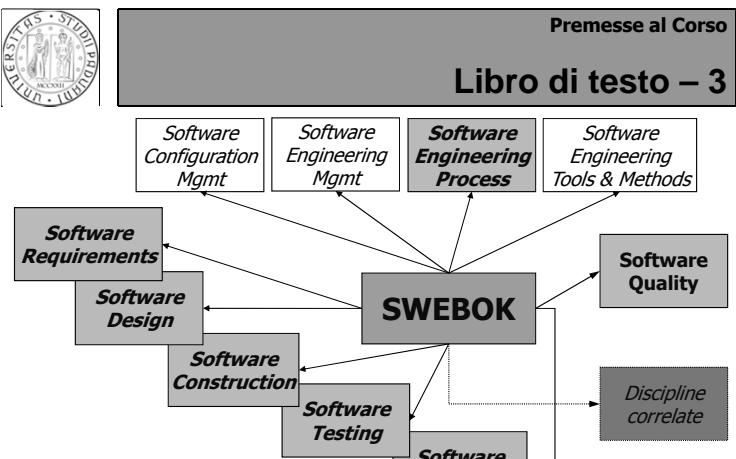
Corso di Laurea in Informatica, Università di Padova

26/30



Premesse al Corso

Libro di testo – 3



Corso di Laurea in Informatica, Università di Padova

27/30



Premesse al Corso

Libro di testo – 4

- Come testi di consultazione useremo**
 - Ian Sommerville
Software Engineering, 9th ed., 2011
Addison Wesley (Pearson Education)
 - E. Gamma, R. Helm, R. Johnson, J. Vlissides (GoF)
Design Patterns, 2002
Addison-Wesley (Pearson Education Italia)
 - Luciano Baresi, Luigi Lavazza, Massimiliano Pianciamore
Dall'idea al codice con UML2, 2006
Addison-Wesley (Pearson Education Italia)

Corso di Laurea in Informatica, Università di Padova

28/30



Progetto didattico – 1

- Un progetto didattico impegnativo da svolgere in gruppo**
 - Riconduce a unità i 2 moduli dell'insegnamento IS
 - Aiuta a esplorare in forma esperienziale le 3 dimensioni fondamentali della disciplina IS
- 6-7 persone per gruppo**
- ~100 ore di impegno individuale**
 - In aggiunta a impegno di esplorazione tecnologica



Progetto didattico – 2

- Dimensione 1: attività di gruppo**
 - Ripartita, coordinata, regolata e controllata
- Dimensione 2: analisi di problema e soluzione**
 - Al di là degli aspetti puramente realizzativi
 - Attenzione alla tipologia degli utenti, all'ambito d'uso, alle risorse disponibili, alle evoluzioni future, ...
- Dimensione 3: approccio ingegneristico**
 - Disciplinato, sistematico, quantificabile