



# Introduzione



Anno accademico 2019/20  
Ingegneria del Software

Tullio Vardanega, [tullio.vardanega@math.unipd.it](mailto:tullio.vardanega@math.unipd.it)

Corso di Laurea in Informatica, Università di Padova **1/42**



## Ingegneria del Software

Corso di Laurea in Informatica  
Università di Padova, a.a. 2019/20  
Docente: Tullio Vardanega

Ultimo aggiornamento: 02 settembre 2019, ore 14:00  
[pubblicata prima bozza di calendario didattico]  
documento collaborativo per la formazione dei gruppi  
Scadenza tassativa di registrazione studenti nel documento collaborativo:  
I lotto: 3 novembre 2018, ore 18  
II lotto: 29 febbraio 2020, ore 18

### Calendario delle lezioni

Settimana	Data	Lezione	Docente	Contenuto	SWEBOK	Sommerville
2	7 ottobre	T1	Vardanega	<b>Introduzione</b> Per approfondire #1: No Silver Bullet, Essence and Accidents of Software Engineering		
	10 ottobre	T2	Vardanega	Per approfondire #2: ACM Code of Ethics		
	11 ottobre	T3	Vardanega	<b>Processi software</b> Per approfondire #3: Origine, formazione ed evoluzione degli standard di processo	\$8	\$2
	14 ottobre	T4	Vardanega	Per approfondire #4: Lo standard ISO/IEC 12207:1995		
	17 ottobre	T5	Vardanega	Continuazione lezione 3 <b>Il ciclo di vita del software</b> Per approfondire #5: I modelli agli Per approfondire #6: Le schede SEMAT (1, 2)		\$3
				<b>Gestione di progetto</b>		



## Introduzione

### Cosa facciamo in questo corso – 1

- **Apprendiamo metodi e pratiche di lavoro alla base della professione informatica**
  - **Gestire il tempo**
    - Disponibilità, scadenze, conflitti, priorità
  - **Collaborare**
    - Fissare obiettivi, dividersi compiti, verificare progressi, riportare difficoltà
  - **Assumersi responsabilità**
  - **Auto-apprendere**
    - “Imparare a imparare” è una essenziale competenza trasversale
- **Integriamo la teoria con la pratica**
  - Integrazione essenziale all'apprendimento



Common soft skills

- Strong work ethic
- Positive attitude
- Good communication skills
- Time management abilities
- **Problem-solving skills**
- Acting as a team player
- Self-confidence
- Ability to accept and learn from criticism
- Flexibility/adaptability
- Working well under pressure



## Introduzione

### Come vogliamo imparare

- **La conoscenza passa dalla comprensione profonda dei significati**
  - Cominciando dai termini evidenziati in colore nelle diapositive
- **Vogliamo costruirci un glossario**
  - Raccolta di termini/concetti centrali al tema e bisognosi di spiegazione
  - Registrati in modo da facilitarne la localizzazione
  - Corredati dalla specifica del loro significato e ogni altra informazione utile a comprenderli
- **Raffinandone la comprensione legando la teoria con la pratica**



MANUALE  
di GIOVANNI  
VIGNOTTO

 **Introduzione**

## Come lo facciamo – 1

- Tramite una attività annuale**
  - Estesa su due semestri
- Con impegni sia collaborativi che individuali**
  - Un progetto collaborativo ✓
  - Una prova scritta individuale
- Relazionandoci con terze parti** ✓
- Usando tecnologie innovative** ✓

Sfide nuove e impegnative

Corso di Laurea in Informatica, Università di Padova 5/42

 **Introduzione**

## Glossario

- Progetto**
  - Insieme di attività e compiti che**
    - Devono raggiungere determinati obiettivi, a partire da specifiche fissate
    - Hanno data d'inizio e data di fine fissate
    - Possono contare su limitate disponibilità di risorse (persone, tempo, denaro, strumenti)
    - Consumano tali risorse nel loro svolgersi

Fonte: Harold Kerzner (1940-), uno dei maggiori esperti mondiali di *project management*

Corso di Laurea in Informatica, Università di Padova 6/42

 **Introduzione**

## Costituenti di un progetto

- Quattro attività complementari (e complesse)**
  - Pianificazione**
    - Gestire risorse (persone, tempo, fondi, strumenti) e responsabilità
  - Analisi dei requisiti**
    - Definire cosa bisogna fare
  - Progettazione (*design* ≠ *project*)**
    - Definire come farlo
  - Realizzazione**
    - Farlo quel che serve, perseguendo qualità
    - Verificando che quanto prodotto non contenga errori
    - Validando i risultati rispetto alle attese



Corso di Laurea in Informatica, Università di Padova 7/42

 **Introduzione**

## Cosa non è un progetto?

- Aiutatemi a dirlo ...**

Corso di Laurea in Informatica, Università di Padova 8/42



Introduzione

## Cosa non è un progetto – 2

- ❑ *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*

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

Corso di Laurea in Informatica, Università di Padova

9/42



Introduzione

## Ingredienti

- ❑ Per svolgere un progetto con confidenza nel suo successo serve un metodo
- ❑ Questo metodo si chiama ingegneria

Engineering: application of scientific and mathematical principles to practical ends  
Fonte: American Heritage Dictionary

- **Applicazione** (non creazione!) di principi noti e autorevoli: *best practice*
- **Practical ends** che spesso sono civili e sociali, associati a responsabilità etiche e professionali

Corso di Laurea in Informatica, Università di Padova

10/42



Introduzione

## Glossario

- ❑ **Software engineering [SWE]**
  - Disciplina per la realizzazione di prodotti SW così impegnativi da richiedere il dispiego di attività collaborative
  - Capacità di produrre "in grande" e "in piccolo"
  - Garantendo qualità: efficacia
  - Contenendo costi e tempi di produzione: efficienza
  - Lungo l'intero ciclo di vita dello sviluppo e dell'uso del prodotto richiesto

Corso di Laurea in Informatica, Università di Padova

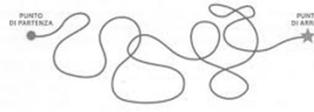
11/42



Introduzione

## Glossario

AZIONE EFFICACE MA NON EFFICIENTE



AZIONE EFFICACE ED EFFICIENTE



- ❑ **Efficacia**
  - Misura della capacità di raggiungere l'obiettivo prefissato
- ❑ **Efficienza**
  - Misura dell'abilità di raggiungere l'obiettivo impiegando le risorse minime indispensabili

Corso di Laurea in Informatica, Università di Padova

12/42



Introduzione

**Glossario**

**Prodotto SW**

- Commessa**
  - Con forma, contenuto e funzione fissate dal committente
- Pacchetto**
  - Con forma, contenuto e funzione idonee alla replicazione
- Componente**
  - Con forma, contenuto e funzione adatte alla composizione
- Servizio**
  - Con forma, contenuto e funzione fissate dal problema

Corso di Laurea in Informatica, Università di Padova

13/42



Introduzione

**Glossario**

**Ciclo di vita [dello sviluppo SW]**

- Gli stati che lo sviluppo del prodotto SW richiesto, assume dal concepimento al ritiro**



**Best practice**

- Modo di fare (*way of working*) noto, che abbia mostrato di garantire i migliori risultati in circostanze note e specifiche**

Corso di Laurea in Informatica, Università di Padova

14/42



Introduzione

**SWE rispetto alle altre discipline**

***The scientific disciplines of reference to SWE include not only computer science but also certain areas of discrete mathematics and operation research, statistics, psychology and economics***

***SWE is not a branch of computer science; it is 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

15/42



Introduzione

**SWE rispetto a se stessa**

**Un sistema SW è tanto più utile quanto più è usato**

- Metrica: integrale degli usi (o #utenti) nel tempo**

**Prodotti con lunga vita operativa hanno costi dominati dalla manutenzione**

- Manutenzione: insieme di attività necessarie a garantire l'uso del prodotto**
- Quali attività servono dipende dal tipo di manutenzione richiesta**

**I costi della manutenzione derivano da**

- Mancato guadagno (e perdita di reputazione), recupero o reclutamento esperti, sottrazione di risorse ad altre attività**

**I principi SWE puntano ad abbassare tali costi**

- Sviluppando SW facilmente manutenibile**

Corso di Laurea in Informatica, Università di Padova

16/42

Introduzione

## Le varianti della manutenzione

**The Types of Maintenance**

Maintenance

- Preventive Maintenance
  - Time Based
  - Failure Finding
  - Condition Based
  - Predictive
  - Risk Based
- Corrective Maintenance
  - Deferred
  - Emergency

ROAD TO RELIABILITY™

Corso di Laurea in Informatica, Università di Padova 17/42

Introduzione

## Cos'è l'ingegneria del software – 1

- ❑ Nasce nel 1968 → ha da poco compiuto 50 anni
  - Conferenza NATO 7-11/10/1968 @ Garmisch (D)
- ❑ Raccogliere, organizzare e consolidare conoscenza (*body of knowledge*) necessaria a realizzare progetti SW con massima efficienza ed efficacia
  - Acquisizione, utilizzo e manutenzione migliorativa di *best practice*
- ❑ L'applicazione di principi ingegneristici alla produzione del SW

Corso di Laurea in Informatica, Università di Padova 18/42

Introduzione

## Cos'è l'ingegneria del software – 2

*L'approccio sistematico, disciplinato e quantificabile allo sviluppo, l'uso, la manutenzione e il ritiro del SW*  
Fonte: Glossario IEEE

- ❑ **Approccio sistematico**
  - Modo di lavorare metodico e rigoroso
  - Che studia, usa ed evolve le *best practice* di dominio
- ❑ **Disciplinato**
  - Che segue le regole che si è dato
- ❑ **Quantificabile**
  - Che permette di misurare l'efficienza e l'efficacia del suo agire

Corso di Laurea in Informatica, Università di Padova 19/42

Introduzione

## Cosa facciamo in questo corso – 2

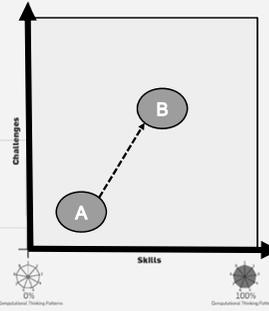
- ❑ **Studiamo tutte le attività di progetto**
  - Prevalentemente nel I semestre
- ❑ **Proviamo a metterle in pratica**
  - Nel progetto didattico, da fine novembre al collaudo di consegna
- ❑ **Verifichiamo il grado di apprendimento**
  - In itinere: tramite revisioni di avanzamento
  - In fine: tramite una prova scritta individuale

Corso di Laurea in Informatica, Università di Padova 20/42



Introduzione

## Perché lo facciamo così – 1



- ❑ *Student acquisition of computational thinking skills advances in response to challenges*
- ❑ *Pedagogical approaches can be described as instructional trajectories connecting a skill/challenge starting point (A) with a destination point (B)*

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

Corso di Laurea in Informatica, Università di Padova

21/42



Introduzione

## Pensiero computazionale

- ❑ **L'essenza dell'informatica, fatto di**
  - **Pensiero algoritmico**
    - Specifica di una sequenza chiara e ordinata di passaggi / azioni che puntano a un determinato fine
  - **Pensiero logico**
    - Elaborazione di un ragionamento chiaro e preciso per convincere/convincersi, stabilire, controllare
  - **Astrazione**
    - Concentrarsi sulle informazioni rilevanti
  - **Riconoscimenti di elementi ricorrenti**
    - Nel problema, nella possibile soluzione
  - **Generalizzazione**
    - Applicazione di quanto sopra a situazioni e nuove

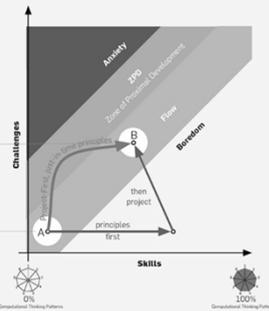
Corso di Laurea in Informatica, Università di Padova

22/42



Introduzione

## Perché lo facciamo così – 2



- ❑ *The project-first just-in-time-principles approach is in the Zone of Proximal Flow (ZPF)*
- ❑ *The Flow is the ideal condition for learning*
- ❑ *The ZPF orchestrates students' take in best practices with assistance and tool use*

Corso di Laurea in Informatica, Università di Padova

23/42



Introduzione

## Con quale quantità di impegno

- ❑ **13 crediti → 325 ore di lavoro individuale**
- ❑ **75 ore impegnate in lezioni, esercitazioni e seminari**
  - 64 di esse nel I semestre
- ❑ **150 le impegniamo nel progetto didattico**
  - Fino a 105 ore di impegno in attività obbligatorie
  - Altre 45 per auto-formazione su strumenti e metodi di lavoro rilevanti per il progetto
- ❑ **100 per studio personale in preparazione alla prova scritta individuale**
  - Studiando con *zero-latency* e non *zero-laxity*

Corso di Laurea in Informatica, Università di Padova

24/42

**Introduzione**

## Regole e vincoli di progetto – 1

- ❑ **Svolgere attività strettamente collaborative**
  - ~7 persone per gruppo → condividere, ripartire, coordinare, verificare
- ❑ **Cercare soluzioni sostenibili a problemi complessi**
  - Tipologia utenti, dominio d'uso, risorse disponibili, prospettive
  - Auto-apprendimento di tecnologie e metodi di lavoro
- ❑ **Adottare un approccio ingegneristico**
  - Lavorare in modo disciplinato, sistematico, quantificabile
  - [85..105] ore di impegno individuale → costo esterno rendicontato per attività obbligatorie
  - ≈ 45 ore di esplorazione tecnologica → costo interno per attività integrative (da condividere, ripartire e contenere)

Corso di Laurea in Informatica, Università di Padova

**25/42**

**Introduzione**

## Regole e vincoli di progetto – 2

- ❑ **Partecipa solo chi ha soddisfatto le propedeuticità**
  - Basi di Dati (superamento completo)
  - Programmazione a oggetti (almeno prova scritta)
- ❑ **Chi ha altri "arretrati", li sani prima di cimentarsi con il progetto**
- ❑ **I gruppi sono formati in sessione pubblica dopo la pubblicazione dei capitoli**
  - Gli aventi diritto si registrano in un tabellone @ Google Docs
  - Gli altri studenti si registrano in altro tabellone, specificando le propedeuticità non soddisfatte
    - Per costoro varranno ulteriori regole e scadenze di ingresso

Corso di Laurea in Informatica, Università di Padova

**26/42**

**Introduzione**

## Regole e vincoli di progetto – 3

- ❑ **L'impegno necessario per raggiungere gli obiettivi di progetto deve essere superiormente limitato**
  - E deve integrarsi con tutti gli altri obblighi personali
  - Questo sconsiglia la partecipazione con "arretrati"
- ❑ **Gli obiettivi di progetto devono essere fissati in modo elastico**
  - Tra un minimo accettabile e un massimo ambizioso, negoziati dinamicamente con il proponente

Corso di Laurea in Informatica, Università di Padova

**27/42**

**Introduzione**

## Regole e vincoli di progetto – 4

Nominativo	Ore per ruolo						Ore totali
	PM	AM	AN	PT	PR	VE	
Rendicontate	8	0	6	41	24	26	105
Totali	8	4	16	41	32	40	141
Rendicontate	4	3	6	32	16	44	105
Totali	9	3	12	32	24	61	141
Rendicontate	4	8	0	36	23	34	105
Totali	12	14	11	36	31	36	140
Rendicontate	3	5	7	42	14	34	105
Totali	3	6	15	42	22	53	141
Rendicontate	3	0	6	18	30	48	105
Totali	10	2	15	18	38	57	140
Rendicontate	5	5	5	25	23	42	105
Totali	5	5	17	25	31	58	140
Rendicontate	10	6	0	38	20	31	105
Totali	10	12	19	38	28	33	140

**7 persone con impegno totale complessivo individuale ≤ 150h**

**Dati 2016-2017**

	Re	Am	An	Pi	Pr	Ve	Tot
7	7	14	30(+2)	24(+1)	64(+1)		146(+4)
7	12	18(+1)	41	22(-2)	41		146(-1)
12	13	14	18(-2)	28(+1)	61(+5)		146(+4)
16	9	12	42(+2)	31(+1)	36(+1)		146(+4)
15	12	19	49(+2)	19(+1)	32(+1)		146(+4)
9	12	24	33(+2)	26(-3)	42(+5)		146(+4)
13(+1)	13	16	24(+6)	39(+5)	41(+8)		146(+4)
<b>Tot in ore</b>							
	79(+8)	78	107(+12)	237(+12)	389(+4)	317(+5)	1071(+23)

Corso di Laurea in Informatica, Università di Padova

**28/42**



Introduzione

## Gli argomenti che tratteremo

<ul style="list-style-type: none"> <li><input type="checkbox"/> Processi, ciclo di vita e modelli di sviluppo del SW</li> <li><input type="checkbox"/> Gestione di progetto</li> <li><input type="checkbox"/> Amministrazione IT</li> <li><input type="checkbox"/> Analisi dei requisiti</li> <li><input type="checkbox"/> Progettazione</li> <li><input type="checkbox"/> Documentazione</li> <li><input type="checkbox"/> Qualità</li> <li><input type="checkbox"/> Verifica e validazione</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> UML: diagrammi dei casi d'uso</li> <li><input type="checkbox"/> UML: diagrammi delle classi e dei <i>package</i></li> <li><input type="checkbox"/> UML: diagrammi di sequenza e di attività</li> <li><input type="checkbox"/> <i>Design pattern</i>: creazionali, comportamentali, architetturali</li> <li><input type="checkbox"/> Stili architetturali</li> <li><input type="checkbox"/> Principi SOLID</li> </ul>
---	--

Corso di Laurea in Informatica, Università di Padova

29/42



Introduzione

## Come lo facciamo – 2

- Tre lezioni in modalità *flipped***

Flipping the classroom means that students gain first exposure to new material **outside** of class (studying documents and online resources) and then use class time to do the **harder** work of assimilating that knowledge through problem-solving, discussion, debates

Corso di Laurea in Informatica, Università di Padova

30/42



Introduzione

## Come lo facciamo – 3

- Strumenti di lavoro collaborativo (21 ottobre)**
  - Gestione progetto (calendari, canali, *ticket*)
  - In concerto con l'insegnamento TOS
    - Versionamento (p.es., git, SVN)
    - Configurazione e *build* (p.es, Maven, Ant, Gradle, ...)
- Lessons learned (22 novembre)**
  - Ricercare ed elaborare "consigli dei veterani" su come organizzare la gestione collaborativa delle attività di documentazione
- Ways of working (5 dicembre)**
  - SEMAT, sviluppo *agile* (p.es. Scrum)

Corso di Laurea in Informatica, Università di Padova

31/42



Introduzione

## Fonti e risorse – 1

- I libri di SWE si dividono in due categorie**
  - Teorici** : trattano la materia in modo privo di riflessi di **esperienza concreta**
    - Principi esposti, ma spesso non vissuti
  - Esperienziali** : **espongono l'esperienza dell'autore, senza relazionarla bene alla visione generale del problema e della disciplina**
    - Eccessiva enfasi sugli aspetti accidentali
- Noi useremo**
  - Software Engineering**, 10th ed., 2014, di Ian Sommerville, edito da Addison Wesley (Pearson Education)

Corso di Laurea in Informatica, Università di Padova

32/42



Introduzione

**Fonti e risorse – 2**

- ❑ Faremo anche riferimento a
  - Guide to the Software Engineering Body of Knowledge (SWEBOK v3)  
IEEE Computer Society  
Software Engineering Coordinating Committee
  - <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
- ❑ Che ci aiuta a familiarizzarci con le aree di conoscenza della disciplina SWE

Corso di Laurea in Informatica, Università di Padova

33/42



Introduzione

**Fonti e risorse – 3**

Table I.1. The 15 SWEBOK KAs	
Software Requirements	←
Software Design	←
Software Construction	←
Software Testing	←
Software Maintenance	←
Software Configuration Management	←
Software Engineering Management	←
Software Engineering Process	←
Software Engineering Models and Methods	←
Software Quality	←
Software Engineering Professional Practice	←
Software Engineering Economics	←
Computing Foundations	
Mathematical Foundations	
Engineering Foundations	

Noi ci occupiamo di queste 10

Corso di Laurea in Informatica, Università di Padova

34/42



Introduzione

**Fonti e risorse – 4**

- ❑ Come altri testi di consultazione useremo
  - E. Gamma, R. Helm, R. Johnson, J. Vlissides  
*Design Patterns*, 2002  
Addison-Wesley (Pearson Education Italia)
    - E le moltissime risorse digitali che li approfondiscono
  - 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

35/42



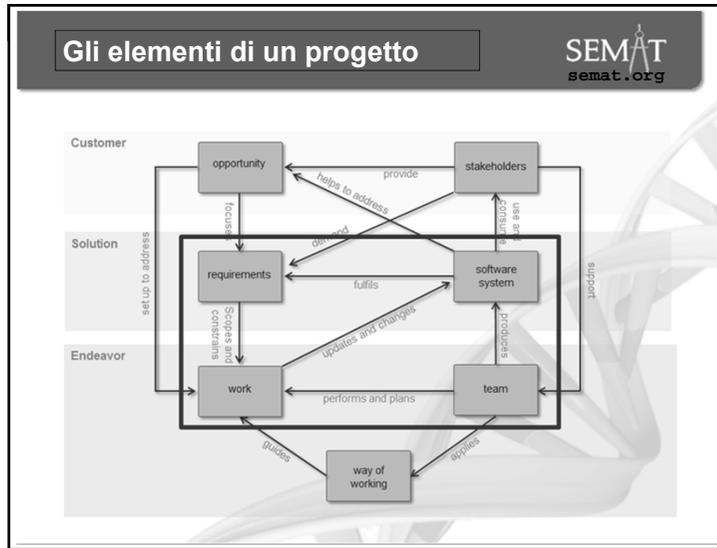
Introduzione

**Come si studia SWE**

- ❑ Costruendo incrementalmente il proprio glossario
  - Basandolo inizialmente sulla teoria
    - Individuazione dei termini, definizione dei significati
  - Consolidandolo con la pratica
    - Applicazione dei significati, confronto critico con l'esperienza
  - Discutendolo con i colleghi
    - Unendo conoscenze parziali, correggendosi reciprocamente
- ❑ Integrando le diapositive con studio personale
  - Ricercando altre fonti e risorse autorevoli

Corso di Laurea in Informatica, Università di Padova

36/42



**Introduzione**  
**Glossario**

**Team[work]**

- Lavoro collaborativo che punta a raggiungere un obiettivo comune in modo efficace ed efficiente
- I membri del *team* sono inter-dipendenti
- La gestione di questa inter-dipendenza richiede il rispetto di regole e di buone pratiche
  - Comunicazioni aperte e trasparenti: risoluzione dei conflitti
  - Costruzione e preservazione delle fiducia reciproca: condivisione e collaborazione
  - Assunzione di responsabilità: coordinamento
  - Condivisione dei rischi
- La sua base è normare il *way of working*

**3** norming ← storming **2**  
 ↓ ↓  
**4** performing ← forming **1**

Corso di Laurea in Informatica, Università di Padova 38/42

**Introduzione**  
**Glossario**

**Stakeholder (portatore di interesse)**

- L'insieme di coloro che a vario titolo influenzano sul prodotto, sul progetto, sui processi
  - La comunità degli utenti (che usa il prodotto)
  - Il committente (che compra il prodotto)
  - Il fornitore (che sostiene i costi di realizzazione)
  - Eventuali regolatori (che verificano l'attuazione di processi)

**Way of working → i processi**

- La maniera di rendere sistematiche, disciplinate e quantificabili le attività di progetto

Corso di Laurea in Informatica, Università di Padova 39/42

**Introduzione**  
**Figure professionali – 1**

**Software engineer ≠ programmatore**

**Il programmatore**

- Figura professionale dominante nei primi decenni dell'informatica
- Scrive programmi da solo, sotto la propria responsabilità tecnica
- Svolge un'attività creativa fortemente personalizzata

Corso di Laurea in Informatica, Università di Padova 40/42



**Introduzione**

## 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 comprendere il quadro generale in cui si colloca il sistema cui contribuisce**
  - 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
  - Usabilità

Corso di Laurea in Informatica, Università di Padova41/42



**Introduzione**

## Principi etici della professione

Association for Computing Machinery

**Software Engineering Code of Ethics and Professional Practice (Short Version)**

Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

1. PUBLIC - Software engineers shall act consistently with the public interest.
2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.
5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.
8. SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Corso di Laurea in Informatica, Università di Padova42/42