

INGEGNERIA DEL SOFTWARE

Fritz Bauer:

"L'ingegneria del software è l'istituzione e l'impiego di principi ingegneristici fondati, allo scopo di ottenere in modo economico software affidabile ed efficiente su macchine vere"

Software Engineering: Conference sponsored by the NATO Science Committee, NATO Press, 1969

"Software Engineering: The application of a systematic, disciplined, quantifiable approach to development, operation, and maintenance of software; that is, the application of engineering to software"

IEEE Standard Computer Dictionary, 610, 1990

INGEGNERIA DEL SOFTWARE

Arte, Scienza, Ingegneria?

Steve McConnell, *The Art, Science, and Engineering of Software Development*, in IEEE Software, Vol. 15, No. 1, January/February 1998

Arte?

Da un certo punto di vista, lo sviluppo del software è un'arte. Come dicevano i greci, una *Téchne*. I greci antichi non distinguevano tra arte e tecnica: per loro erano esattamente la stessa cosa. Falegnami, vasai, tintori, pittori, sarti, medici, architetti: tutti artisti / tecnici, nel senso che per fare il loro mestiere era (è) necessaria la padronanza di tecniche specifiche. E la tecnologia, nel significato originario, è il discorso – *lògos* - sulla *téchne*.

Certo, utilizzando una terminologia moderna, più che un artista, lo sviluppatore software può essere considerato un artigiano. E porre in risalto l'aspetto "artigianale" del lavoro evidenzia una caratteristica importante dello sviluppo software, cioè il ruolo decisivo giocato dalle capacità individuali: numerosi studi empirici hanno dimostrato come esistano notevoli differenze tra i singoli sviluppatori in termini di produttività (e di qualità del lavoro), anche a parità di livello di esperienza. Gli studi più cauti parlano di differenze nell'ordine di 10 a 1.

INGEGNERIA DEL SOFTWARE

Scienza?

Purtroppo no, se intendiamo per scienza le cosiddette "scienze esatte". Di "scientifico", nel settore dello sviluppo software (a differenza dalla computer science) esiste poco, a parte alcune tecniche a base matematica, come ad esempio la normalizzazione dei dati.

Se invece utilizziamo una accezione più comprensiva di scienza, che comprenda anche le cosiddette "scienze umane", il discorso cambia.

Ingegneria

Dal punto di vista dei committenti e dei responsabili di progetto, la visione dello sviluppo software come un'attività artigianale, i cui risultati dipendono fortemente dall'estro e dalle capacità individuali dei singoli progettisti, è preoccupante e spesso intollerabile.

Dall'Organizzazione Internazionale per la Standardizzazione (ISO), al Software Engineering Institute (SEI), sponsorizzato dal governo americano e dalle maggiori aziende statunitensi, dai progetti finanziati dalla Comunità Europea all'attività delle diverse associazioni professionali e di categoria, sia internazionali che locali, è in atto da decenni un lavoro enorme di definizione di standard e di

INGEGNERIA DEL SOFTWARE

strategie per far raggiungere all'industria del software un livello di maturità accettabile.

Parlare di ingegneria del software non significa necessariamente negare il ruolo giocato dalle capacità individuali, perseguire un obiettivo di "massificazione" degli sviluppatori software: ci sono stati tentativi in questa direzione, ma sono falliti, come dimostrano gli sforzi delle maggiori società di software mondiale per reclutare i progettisti migliori, a qualunque costo.

L'ingegneria del software, piuttosto, affronta le problematiche di tipo manageriale, organizzativo e metodologico, per permettere che il lavoro degli sviluppatori possa essere condotto con la maggiore efficacia, avvalendosi di tecniche e modi di procedere sperimentati in contesti eterogenei, risolvendo i problemi (soprattutto di comunicazione) legati al fatto che lo sviluppo di sistemi software è un'attività intellettuale complessa, e che coinvolge una pluralità di soggetti diversi tra loro.

INGEGNERIA DEL SOFTWARE

L'Ingegneria del Software è una disciplina *metodologica*, cioè studia i *metodi* di produzione e gli *strumenti* fondamentali di sviluppo e misura della *qualità* di sistemi software complessi.

I problemi principali che affronta l'Ingegneria del Software riguardano la gestione delle risorse durante il processo di sviluppo e l'anticipazione del cambiamento dei requisiti d'uso.

- Lo studio dei modelli di processo di sviluppo del sw
- Lo sviluppo degli strumenti di produzione del sw
- Gli aspetti economici dei prodotti e dei processi

Parliamo di Ingegneria del Software per identificare gli studi di processi della produzione di software, principi di sviluppo, tecniche e notazioni con lo scopo ultimo di ottenere un prodotto di qualità, sviluppato nei tempi ed entro il budget previsti e che soddisfi il cliente.

INGEGNERIA DEL SOFTWARE

La disciplina ben si adatta in tutti quei progetti che per la loro complessità:

- coinvolgono un team di persone ciascuna con il proprio ruolo specifico
- in cui il prodotto o la famiglia di prodotti risultanti abbia un ciclo di vita lungo
- in cui il riutilizzo sia auspicabile per una riduzione di costi
- in cui la manutenzione e gli ulteriori sviluppi abbiano un peso significativo.

Possiamo quindi identificare alcuni principi chiave:

- rigore e formalismo
- separazione dei concetti (modularità ed astrazione)
- anticipazione dei cambiamenti
- generalizzazione
- scalabilità
- produzione di componenti

Questi ed altri principi rendono l'ingegneria del software una disciplina particolarmente indicata nel caso di sviluppo software object-oriented.

INGEGNERIA DEL SOFTWARE

Alcuni dati rilevanti

Lo sviluppo di sistemi software è una disciplina giovane, con poco più di 50 anni di storia alle spalle.

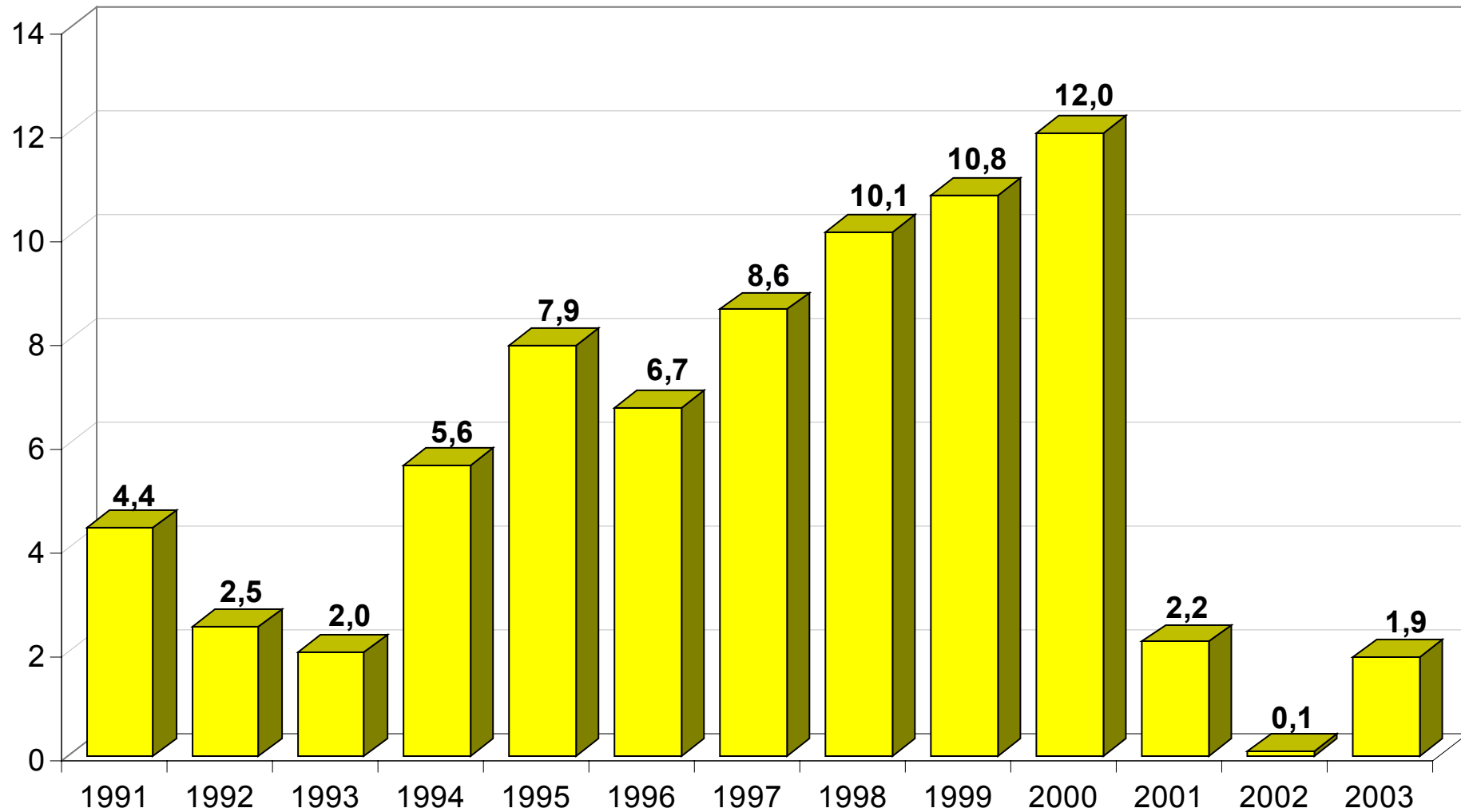
E, se si guarda alle statistiche, si tratta di un'industria profondamente immatura.

- Elevatissimo tasso di crescita
- Skill shortage
- Alta percentuale di fallimento nei progetti software

INGEGNERIA DEL SOFTWARE

Tasso di crescita

Fonte: EITO - European Information Technology Observatory 2002



INGEGNERIA DEL SOFTWARE

Skill shortage

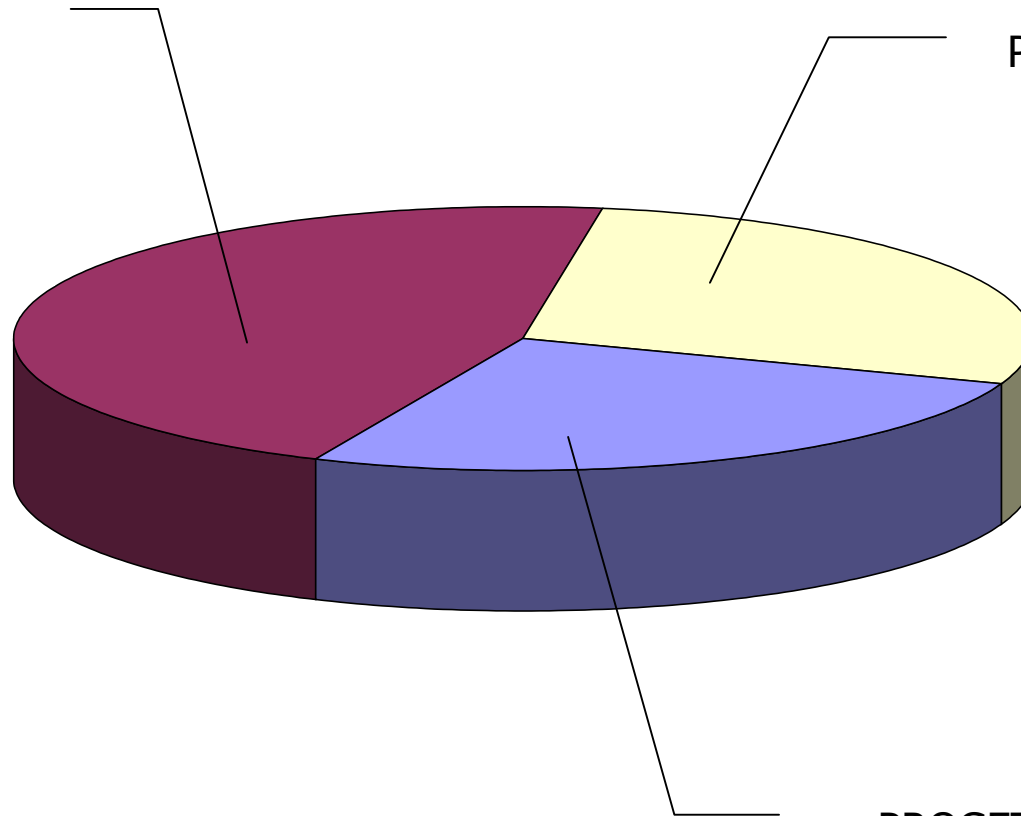
Secondo una ricerca condotta da EITO nel 2000, in Europa ci sono 1 milione e 230 mila posti nell'Information Technology che non riescono ad essere coperti perché mancano le giuste risorse. In Italia la carenza di risorse passerà da 230 mila nel 2000 a 440 mila nel 2003.

Alta percentuale di fallimento

Indagine dello Standish Group, basata su un campione di 28.000 progetti e pubblicata da Computer Weekly il 9 luglio 1998:

INGEGNERIA DEL SOFTWARE

PROGETTI IN RITARDO
CON COSTI IMPREVISTI
CON FUNZIONALITA'
INADEGUATE
46%



PROGETTI FALLITI
28%

PROGETTI RIUSCITI
26%

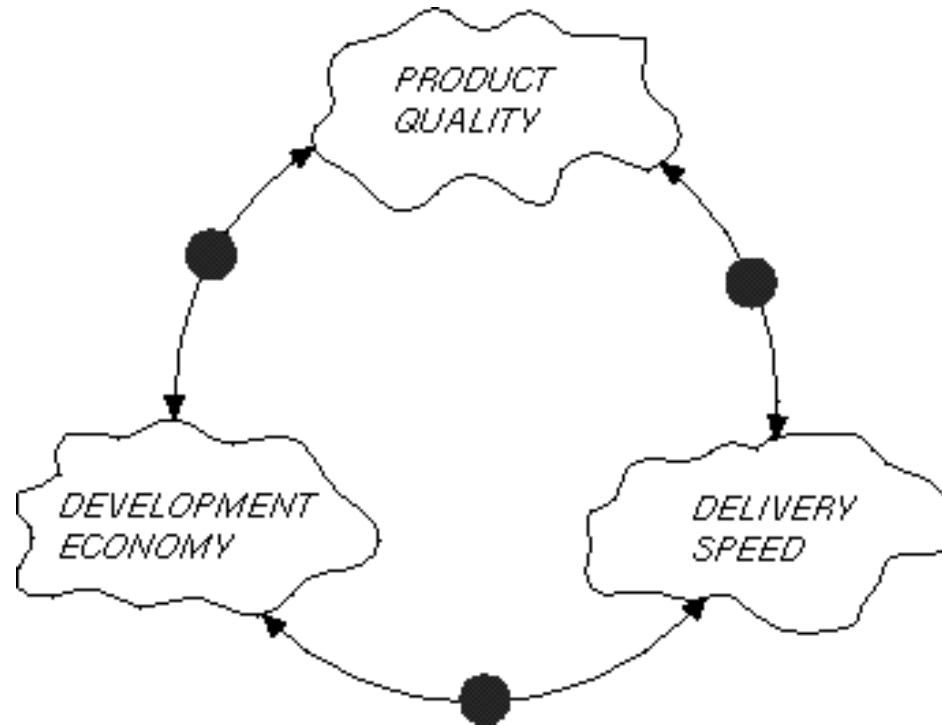
INGEGNERIA DEL SOFTWARE

Gerald Weinberg, *"Quality Software Management - Vol. 4, Anticipating Change"*,
Dorset House, 1997

Software Engineering: the art and science of designing and building, with economy and elegance, applications, objects, frameworks, and other information systems, so that they are correct, reliable, extensible, reusable, secure, efficient, easy to maintain and to use.

INGEGNERIA DEL SOFTWARE

La "triade" dell'Ingegneria del Software:



INGEGNERIA DEL SOFTWARE

Basta saper programmare?

I termini "programmatore" e "Software Engineer" sono spesso usati come sinonimi.

Dunque per svolgere il lavoro di Software Engineer è sufficiente saper programmare?

Un programmatore:

- possiede tutte le conoscenze tecniche relative ai linguaggi ed agli ambienti di programmazione da lui utilizzati
- sviluppa un programma completo a partire da una definizione precisa delle specifiche
- lavora individualmente

Un Software Engineer:

- progetta un componente da integrare con altri allo scopo di creare un prodotto
- progetta componenti riusabili all'interno di altri sistemi
- progetta secondo i principi della programmazione object-oriented

INGEGNERIA DEL SOFTWARE

- opera all'interno di un team di persone che si occupano di fasi diverse dello sviluppo del prodotto e dei suoi componenti
- lavora seguendo i principi del "Codice etico e di pratica professionale dello sviluppo software"

Dunque saper programmare è condizione necessaria ma non sufficiente per essere un Software Engineer.

INGEGNERIA DEL SOFTWARE

Lo sviluppo software è un processo di problem solving

Nel 1945 il matematico G.Polya scrisse "*How to solve it?*", un libro sui *metodi di risoluzione* dei problemi matematici, destinato agli insegnanti di matematica. Polya si chiese: "come possiamo insegnare a qualcuno come si risolve un problema di matematica?"

Il metodo suggerito da Polya consiste di 4 fasi:

1. **Comprensione del problema** Qual è l'incognita? Quali sono i dati? Qual è la condizione sufficiente a determinare l'incognita in funzione dei dati?
2. **Compilazione di un piano risolutivo** Il problema è già noto? È un caso particolare di un altro problema noto?
3. **Sviluppo del piano risolutivo** Si può dimostrare che ciascun passaggio è corretto?
4. **Verifica del risultato** Si può verificare il risultato? Si può verificare il procedimento? Si può ottenere il risultato in qualche altro modo più semplice?

INGEGNERIA DEL SOFTWARE

Il processo di produzione di un sistema software deve seguire un metodo analogo a quello che si usa nella soluzione di problemi matematici.

1. occorre *definire il sistema*, comprendendo e analizzando i requisiti richiesti dall'utente
2. occorre *compilare una specifica del piano di produzione*, cercando di scomporlo in sottoproblemi che siano più semplici da risolvere
3. occorre *sviluppare la specifica*, badando alla *correttezza formale* di tutte le trasformazioni
4. occorre *formalizzare, verificare e convalidare* sia il *prodotto* che il *procedimento* seguito nello sviluppo della produzione

INGEGNERIA DEL SOFTWARE

Lo sviluppo software è una attività di produzione.

Definizione:

"Produzione = il risultato di una serie di trasformazioni operate dall'uomo"

Nelle attività produttive di tipo *industriale* il ciclo di produzione è un processo che viene descritto, controllato, ottimizzato e poi rigorosamente applicato; in genere sono coinvolte moltissime persone.

Altre attività produttive sono invece regolate meno rigorosamente (produzione *artigianale*), ma si presenta comunque il problema del controllo delle risorse e della qualità del prodotto; è coinvolto un numero ridotto di persone.

Altre attività produttive sono invece completamente sregolate e si basano sulla creatività (produzione *artistica*); è coinvolta di norma una singola persona.

In molti casi l'approccio artistico, artigianale e industriale si mescolano:

- costruzione di un edificio
- produzione di un film

INGEGNERIA DEL SOFTWARE

L'insieme delle attività con cui si porta avanti in modo ordinato lo sviluppo e la manutenzione di un prodotto software può essere considerato un *processo di produzione*.

Produzione di sistemi software

L'analogia tra processo di produzione di manufatti industriali e processo di produzione di sistemi software non vale in alcuni aspetti fondamentali:

- I prodotti software sono *intangibili* e *invisibili*: questo è un problema sia nel valutarne il costo, sia nel pretendere il giusto prezzo dai clienti
- La produzione del software è un'attività *creativa* in cui la conoscenza viene acquisita, sperimentata e applicata in modo *incrementale*
- I *clienti* del prodotto *sono coinvolti nella produzione*
- Mentre un ciclo di produzione industriale evolve molto lentamente, un *ciclo di produzione del software* non può essere facilmente definito a priori e spesso all'inizio non è completamente definito
- Specialmente su Internet, chi produce software deve riflettere bene e capire quale servizio corrisponde in modo redditizio al software che vuole produrre

INGEGNERIA DEL SOFTWARE

La progettazione del software

Compito degli ingegneri del sw è la definizione delle proprietà del prodotto ("specifica dei requisiti") e la stesura della sua descrizione ("progetto"), che verranno usate dai programmatori per guidare la realizzazione ("codifica").

L'ingegnerizzazione dei sistemi software comporta lo sviluppo di un insieme di sottoprocessi dei quali i principali sono:

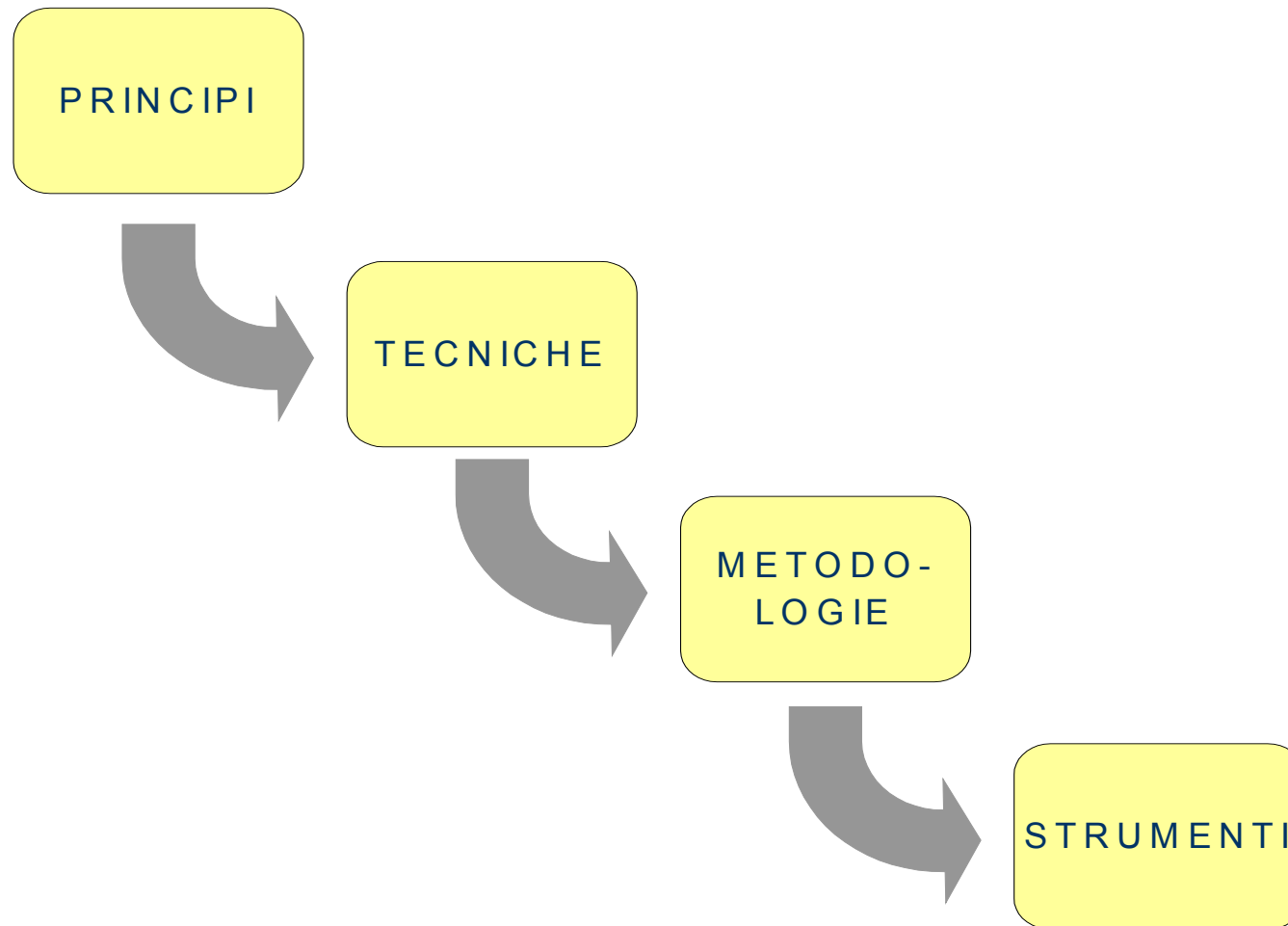
- *Analisi e specifica* dei requisiti del sistema da produrre o integrare
- *Studio di fattibilità e stima delle risorse* necessarie alla produzione, e conseguente *pianificazione* di processo
- *Progetto* di una soluzione basata su un sistema informatico
- Eventuale *prototipazione*, con validazione da parte dell'utente
- *Realizzazione* del progetto in forma di programmi
- *Testing* da parte degli autori dei programmi stessi
- *Verifica della qualità* del prodotto realizzato e della sua documentazione
- *Validazione* del prodotto realizzato rispetto ai requisiti da parte del cliente

INGEGNERIA DEL SOFTWARE

L'Ingegneria del Software è quindi una disciplina metodologica che studia metodi, tecniche e strumenti che supportano tali fasi del processo produttivo.

INGEGNERIA DEL SOFTWARE

Dai *principi* otteniamo delle *tecniche*, da queste delle *metodologie*, e da queste degli *strumenti*.



INGEGNERIA DEL SOFTWARE

L'obiettivo finale è un *prodotto software di qualità*, cioè che:

- Soddisfi i requisiti dell'utente
- Non contenga errori (o ne contenga "pochi")
- Sia semplice da capire e modificare
- Impieghi componenti e sottosistemi appropriati o disponibili
- Abbia una buona efficienza
- Bilanci gli obiettivi con i vincoli di sistema e d'ambiente

INGEGNERIA DEL SOFTWARE

Bruegge, Dutoit, *"Object-Oriented Software Engineering"*, Prentice Hall 2000

Software Engineering is:

a **modelling** activity -- software engineers deal with complexity through modelling, by focusing at any one time on only the relevant details and ignoring everything else.

- *model* -- an abstraction of reality
- *analysis* -- constructing a model of the *problem domain*
- *design* -- constructing a model of the *solution domain*
- In OO methods, the solution domain model is an extension of the problem domain model, so that the structure of the software reflects that of the problem.

a **problem-solving** activity -- models are used to search for an acceptable solution

- driven by experimentation
- reuses pattern solutions

INGEGNERIA DEL SOFTWARE

- incremental evolution of the system toward one acceptable to the client
- revised in response to change

a **knowledge acquisition** activity -- in modelling the application and solution domain, software engineers collect data, organize it into information, and formalize it into knowledge.

- nonlinear -- new information may invalidate previous knowledge
- risk-based development -- identify high-risk components to avoid late surprises
- issue-based development -- execute development activities in parallel, organizing according to issues which still need resolution
- iterative development -- design and implement the high-risk (difficult) parts first

a **rationale-driven** activity -- software engineers need to capture the context in which decisions were made and the rationale behind these decisions in order to understand the implications of a proposed change when revisiting a decision.

- assists in dealing with changing systems

INGEGNERIA DEL SOFTWARE

- useful in the maintenance phase