

# Software Engineering

Competenze della professione



Roma, a.a. 2005-2006

# Contenuti

- Che cos'è una professione riconosciuta
- Responsabilità etica e professionale
- Competenze, discipline, professione e carriera
- Aree e livelli di competenza
- “Best practices” (poco conosciute o poco utilizzate)
- Software Engineering Body of Knowledge (SWEBOK)

## Riferimenti:

1. *Guide to the Software Engineering Body of Knowledge (SWEBOK)*,  
A project of the IEEE Computer Society, Professional Practices Committee,  
2004 Version
2. Steve McConnell, *Professional Software Development*, Addison-Wesley, 2003
3. Ian Sommerville, *Ingegneria del software*, 7° Edizione, Pearson – Addison Wesley, 2005  
Titolo originale: *Software Engineering, 07 Edition*, 2004
4. Roger S. Pressman, *Principi di Ingegneria del software*, quarta edizione, McGraw-Hill, 2005  
Titolo originale: *Software Engineering: A Practitioner's Approach, Sixth Edition*, 2004



# Che cos'è una professione riconosciuta?

*“Perché l'ingegneria del software sia ufficialmente riconosciuta come professione, occorre che ci sia consenso sul corpo delle competenze richieste.”*

*(Body of Knowledge)*



# Responsabilità etica e professionale

## **Codice etico e di pratica professionale dello sviluppo software.**

(ACM/IEEE-CS Joint Task Force on Software Engineering and Professional Practices)

*“Gli sviluppatori software devono impegnarsi a rendere il proprio mestiere (l’analisi, la specifica, il disegno, lo sviluppo, il test e la manutenzione del software) una professione rispettata e dagli effetti benefici. Gli sviluppatori devono pertanto aderire alle seguenti regole:*

*1. **Pubblico.** Gli sviluppatori software devono agire in linea con l’interesse pubblico.*

*2. **Cliente e datore di lavoro.** Gli sviluppatori software devono agire in un modo conforme agli interessi del loro cliente e datore di lavoro, restando in accordo con l’interesse pubblico (punto 1).*

*3. **Prodotto.** Gli sviluppatori software devono assicurare che i loro prodotti e le modifiche che vi applicano siano a livello di standard professionale più elevato possibile.*

*4. **Giudizio.** Gli sviluppatori software devono mantenere integrità ed indipendenza nel loro giudizio professionale.*

*5. **Management.** Manager e leader degli sviluppatori devono sottoscrivere e promuovere un approccio etico al management dello sviluppo e della manutenzione del software.*

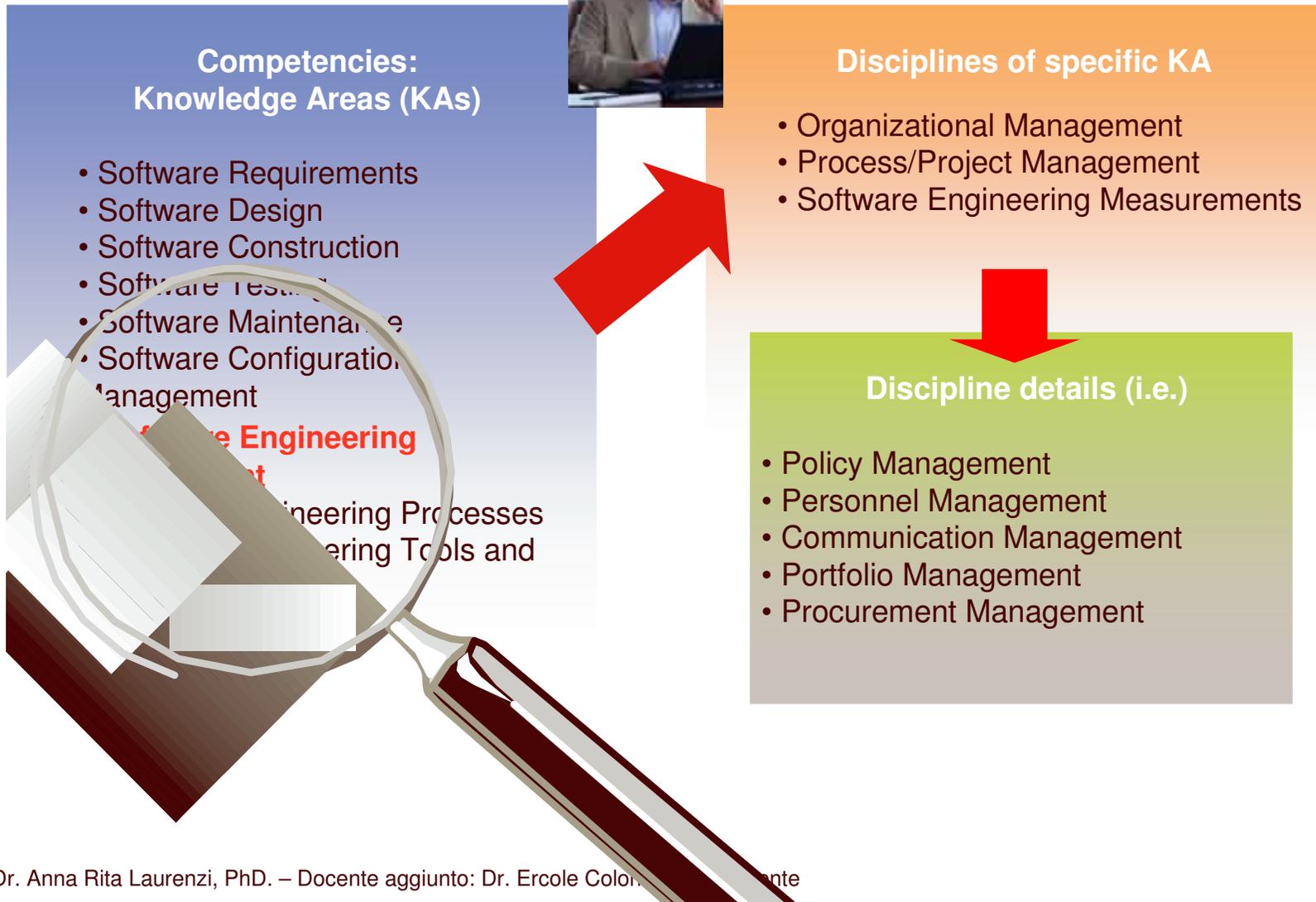
*6. **Professione.** Gli sviluppatori software devono far progredire l’integrità e la reputazione della professione, restando in accordo con l’interesse pubblico (punto 1).*

*7. **Colleghi.** Gli sviluppatori software devono essere leali e di supporto nei confronti dei loro colleghi.*

*8. **Se stessi.** Gli sviluppatori software devono, per tutta la durata della loro attività lavorativa, continuare la propria formazione sulla pratica della professione, e devono promuovere un approccio etico ad essa.”*

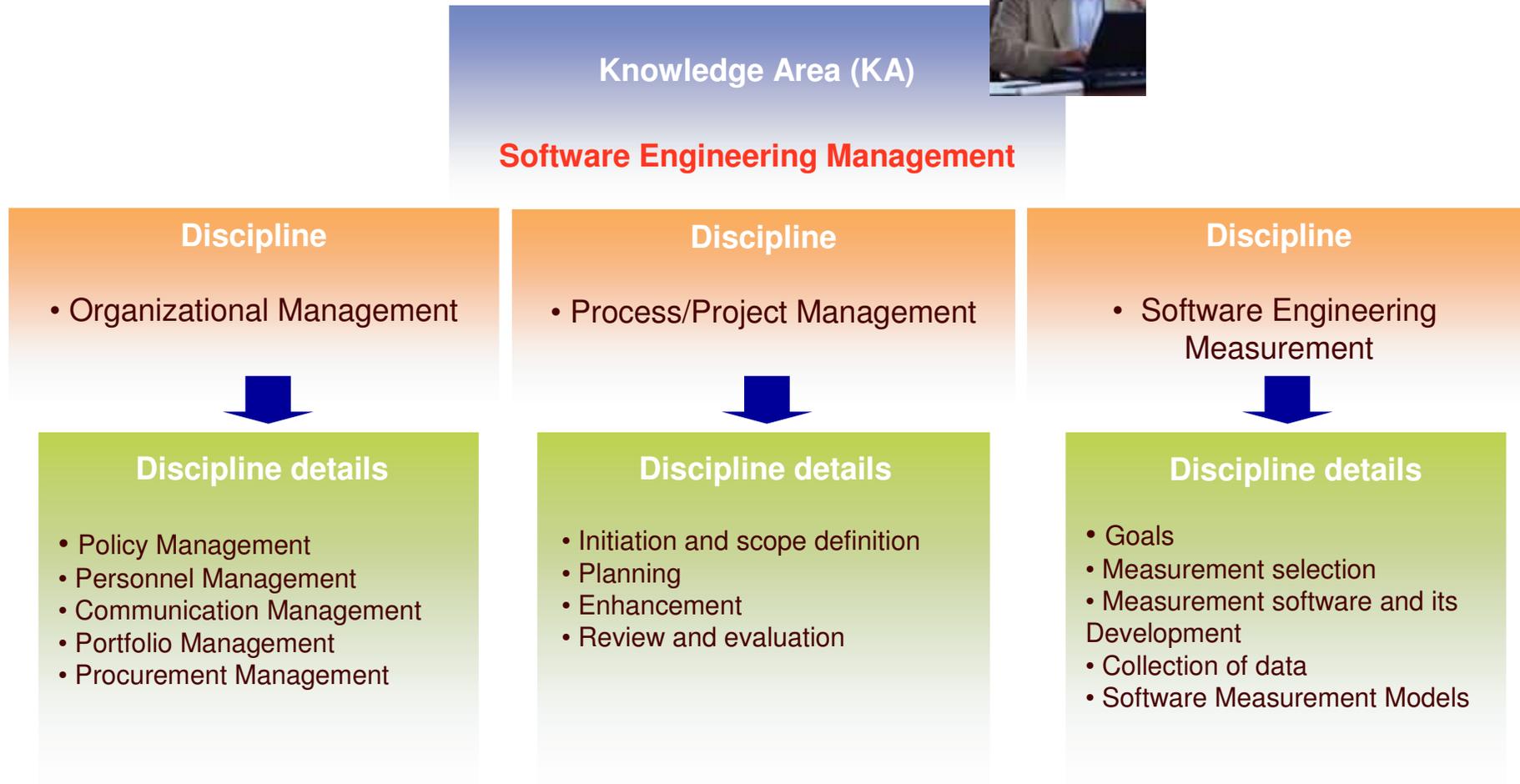
# Competenze e discipline

*es. Software Engineering Management*



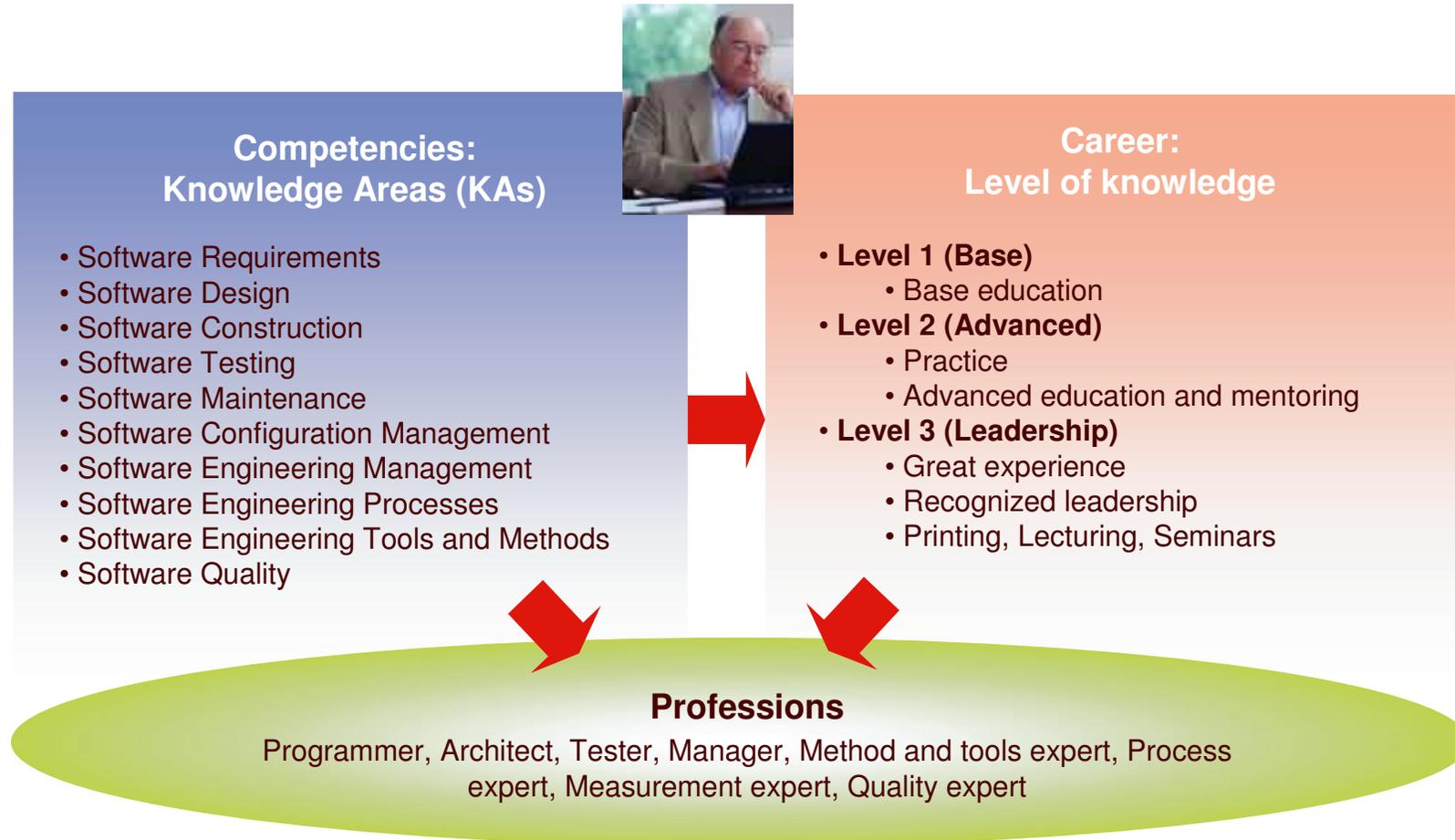
# Dettaglio su competenze e discipline

*es. Software Engineering Management*



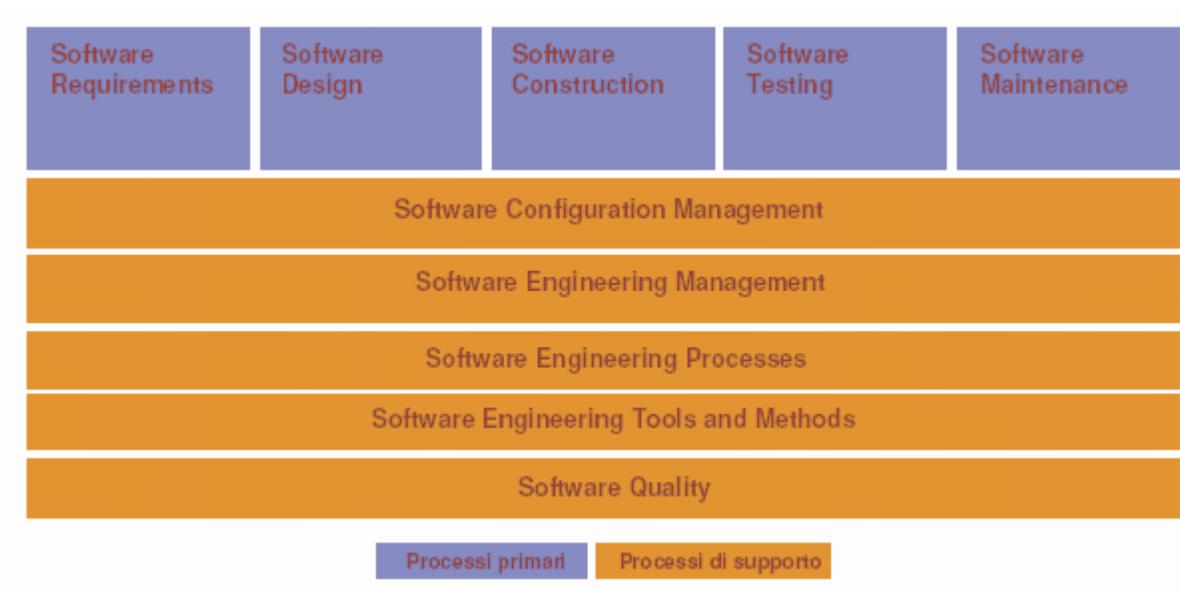
# Competenze, carriere e professioni

*A Project of the Software Engineering Coordinating Committee, from IEEE*



# Competenze e processi di sviluppo

*Secondo ISO 12207:1995, Norme sul ciclo di vita del software*



## Aree di competenze (KAs)

Area di competenza	Descrizione
<i>Software Requirements</i>	Scoprire, analizzare, modellare e documentare le funzioni da sviluppare nel software.
<i>Software Design</i>	Il ponte ideale tra i requisiti e lo sviluppo, la progettazione definisce la struttura e gli stati dinamici del sistema a differenti livelli di astrazione e punti di vista.
<i>Software Construction</i>	La creazione del software secondo la progettazione definita. L'attività primaria è quella di creare il software e la struttura dei dati che implementino le funzionalità tramite i linguaggi, le tecnologie e gli ambienti scelti.
<i>Software Testing</i>	Attività associata all'esecuzione del software per scoprire i difetti e valutare le funzionalità.
<i>Software Maintenance</i>	Attività relative all'installazione, rilascio, migrazione, e operatività del sistema software.
<i>Software Configuration Management</i>	La disciplina che definisce come organizzare e memorizzare i diversi artefatti del progetto software, come controllare e gestire le modifiche, come rilasciare il sistema al cliente.
<i>Software Quality</i>	Attività eseguite sulle parti statiche del sistema software (codice e documenti) per valutare il livello di conformità degli elementi ai requisiti tecnici.
<i>Software Engineering Management</i>	Tutti gli aspetti gestionali che vanno dalla gestione del business e delle persone alle tematiche del progetto software.
<i>Software Engineering Tools and Methods</i>	L'uso di strumenti (tools), tecnologie, metodologie e tecniche per l'ingegneria del software.
<i>Software Engineering Process</i>	Attività relative alla misurazione e al miglioramento della qualità del software sviluppato, alle tempistiche, efficienza, produttività e altre caratteristiche dei progetti software.

**KA** = Knowledge Area

## Livelli di competenza

Livello di competenza	Descrizione
<i>Base</i>	La persona esegue il lavoro di base in una specifica area di competenza ( <i>progettazione, sviluppo, testing, ecc.</i> ), generalmente sotto la supervisione di un esperto. La persona è impegnata ad accrescere le proprie conoscenze e competenze.
<i>Advanced</i>	La persona esegue il proprio lavoro in autonomia ed in maniera efficace nella propria area di competenza. Fa da modello nel proprio ruolo ad altri meno esperti e a volte fa loro da guida ( <i>coacher</i> ).
<i>Leadership/Senior</i>	La persona esegue un lavoro “esemplare” nella propria area di competenza. Fa da guida ( <i>coaching</i> ) in maniera regolare alle persone meno esperte, coordina progetti complessi e fornisce supporto qualificato all’azienda. E’ riconosciuta in azienda come una linea guida nell’ambito della sua competenza.
<i>Mastery</i>	Realizza lavori di riferimento nella propria area di competenza e possiede una profonda esperienza basata su molti progetti. Tiene generalmente seminari o corsi di formazione, scrive articoli o libri riconosciuti di valore nell’ambito dell’area di competenza. E’ riconosciuta come persona di riferimento nell’area di competenza nell’ambiente esterno.

## Livelli di carriera

Livello	Descrizione	KA coperte
9	La persona ha appena completato gli studi (diploma di laurea in ingegneria del software o titolo equivalente) ed è all'inizio della propria carriera. Lavoro sotto la stretta supervisione di un coordinatore.	Non applicabile
10	La persona ha maturato un minimo di esperienza nell'ingegneria del software e lavora da uno o due anni. E' in grado di lavorare con una limitata supervisione del coordinatore.	Base: tutte le 10 KA Advanced: 3 KA
11	La persona ha acquisito una maturato un rilevante competenza nell'ingegneria del software ed in grado di lavora in perfetta autonomia, secondo le richieste. Ha partecipato in uno o più progetti completati e ha esperienza in ogni fase del ciclo di sviluppo necessario per rilasciare un software di qualità.	Base: tutte le 10 KA Advanced: 6 KA Leadership: 1 KA
12	<b>Full Professional Status:</b> ha ottenuto successi in tutti progetti cui ha partecipato contribuendo in maniera significativa nelle aree di propria competenza. Gestisce in maniera efficace e risolutiva le problematiche affidate. E' innovativo, consistente e fornisce contributi significativi nelle attività che le vengono affidate. Fornisce guida tecnica e supporta le persone meno esperte che le vengono affidate.	Base: tutte le 10 KA Advanced: 8 KA Leadership: 3 KA

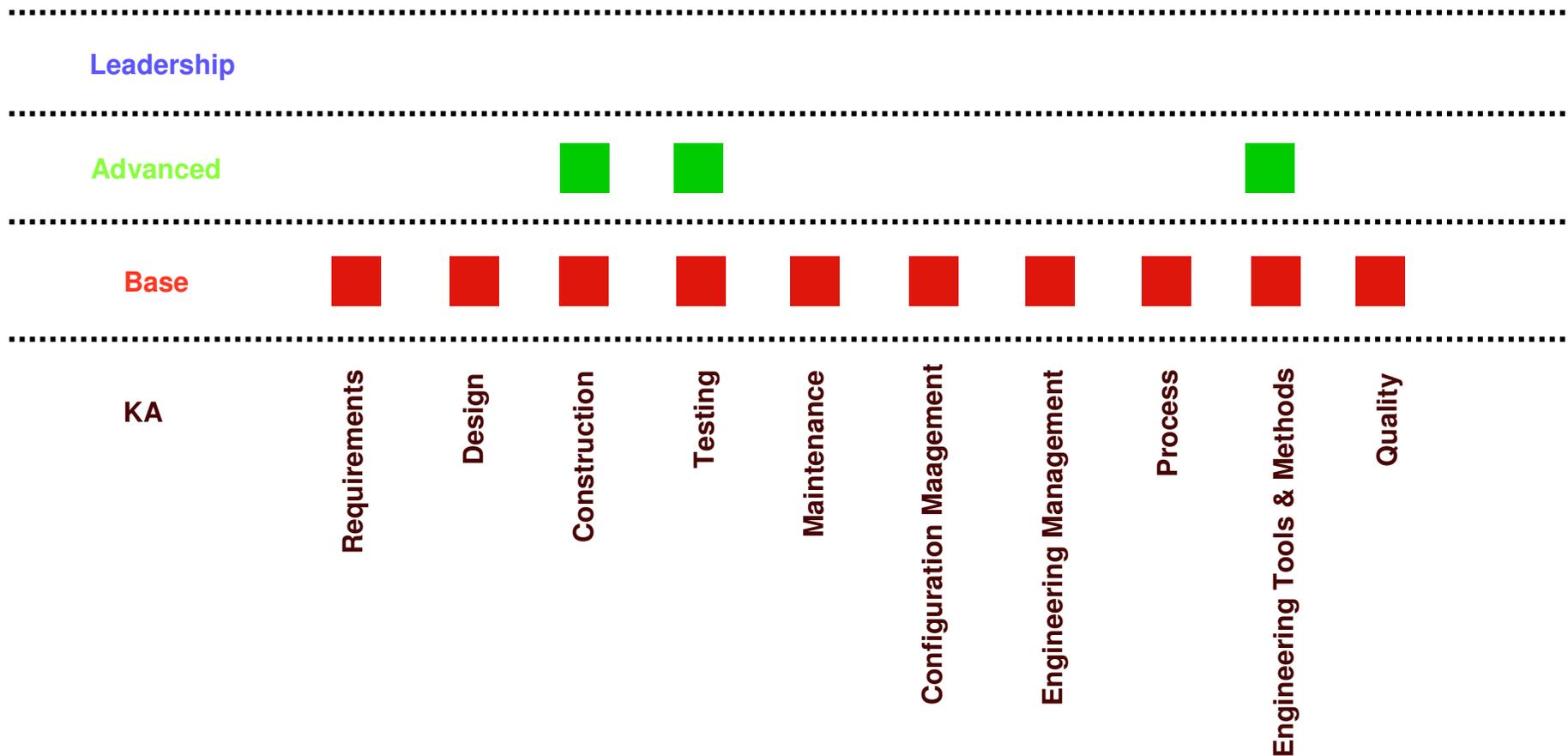
## Livelli di carriera

Livello	Descrizione	KA coperte
13	E' considerato un punto di riferimento ( <i>champion</i> ) nella propria area di competenza ed in grado di considerare sia aspetti interni che esterni del progetto assicurando che essi siano consistenti e correttamente gestiti. Assume la piena responsabilità del progetto coordinato e fornisce un contributo significativo. Le decisioni prese hanno generalmente un impatto positivo sul successo e la profittabilità del progetto.	Base: tutte le 10 KA Advanced: 8 KA Leadership: 5 KA Mastery: 1 KA
14	La persona costituisce una risorsa tecnica di grande valore all'interno dell'azienda. Spesso risolve problemi di grande complessità tecnica e prende decisioni importanti per la riuscita del progetto su cui lavora. E' conosciuto internamente ed esternamente come una persona che contribuisce allo sviluppo dell'arte dell'ingegneria del software.	Intentionally left blank
15	La persona risulta essere indispensabile per il successo dell'azienda. Lavora in maniera consistente alla progettazione e realizzazione di prodotti software di successo e di riferimento per il mercato.	Intentionally left blank

LL = Ladder Level

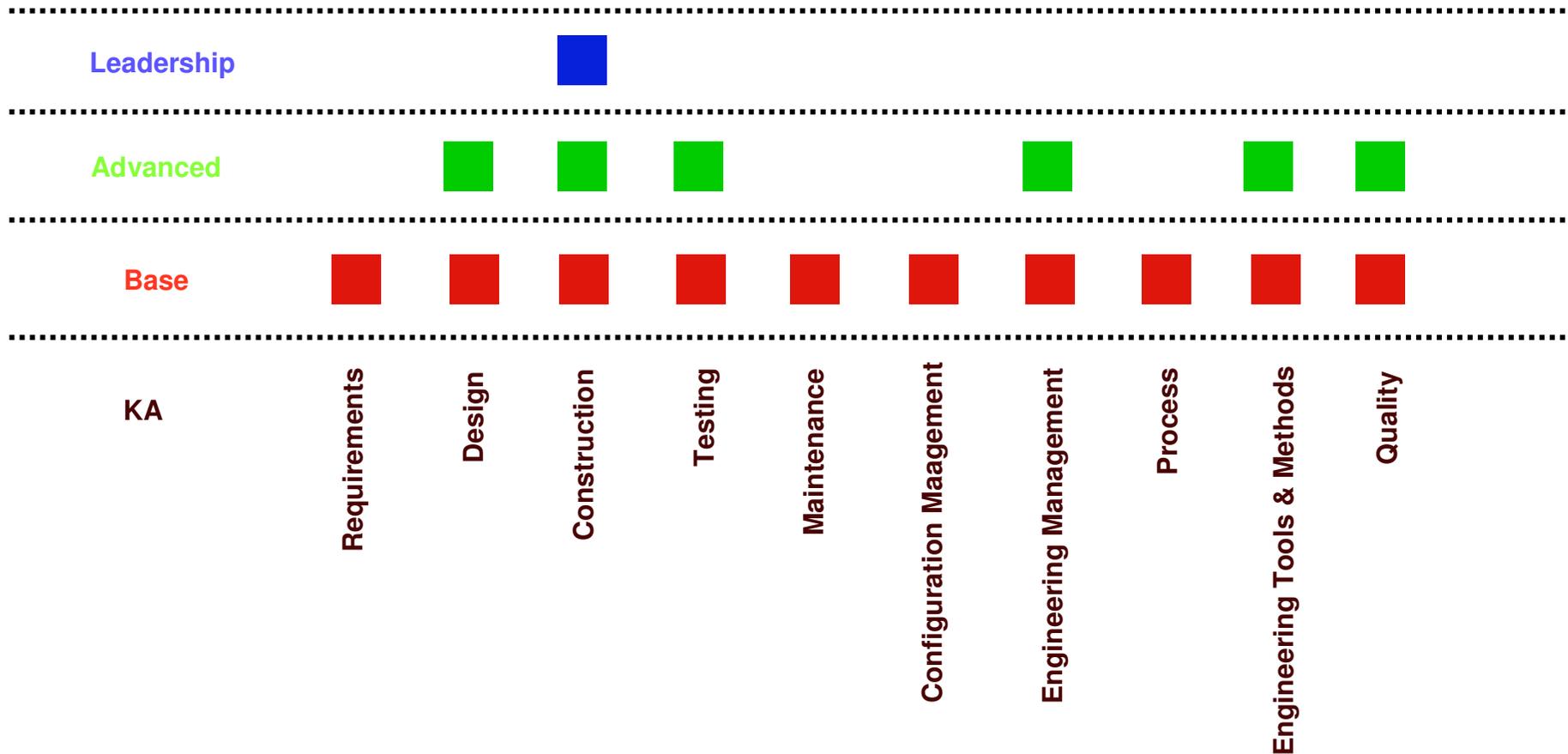
# Avanzamento nella carriera

*Esempio di sviluppo professionale per raggiungere il “livello 10”*



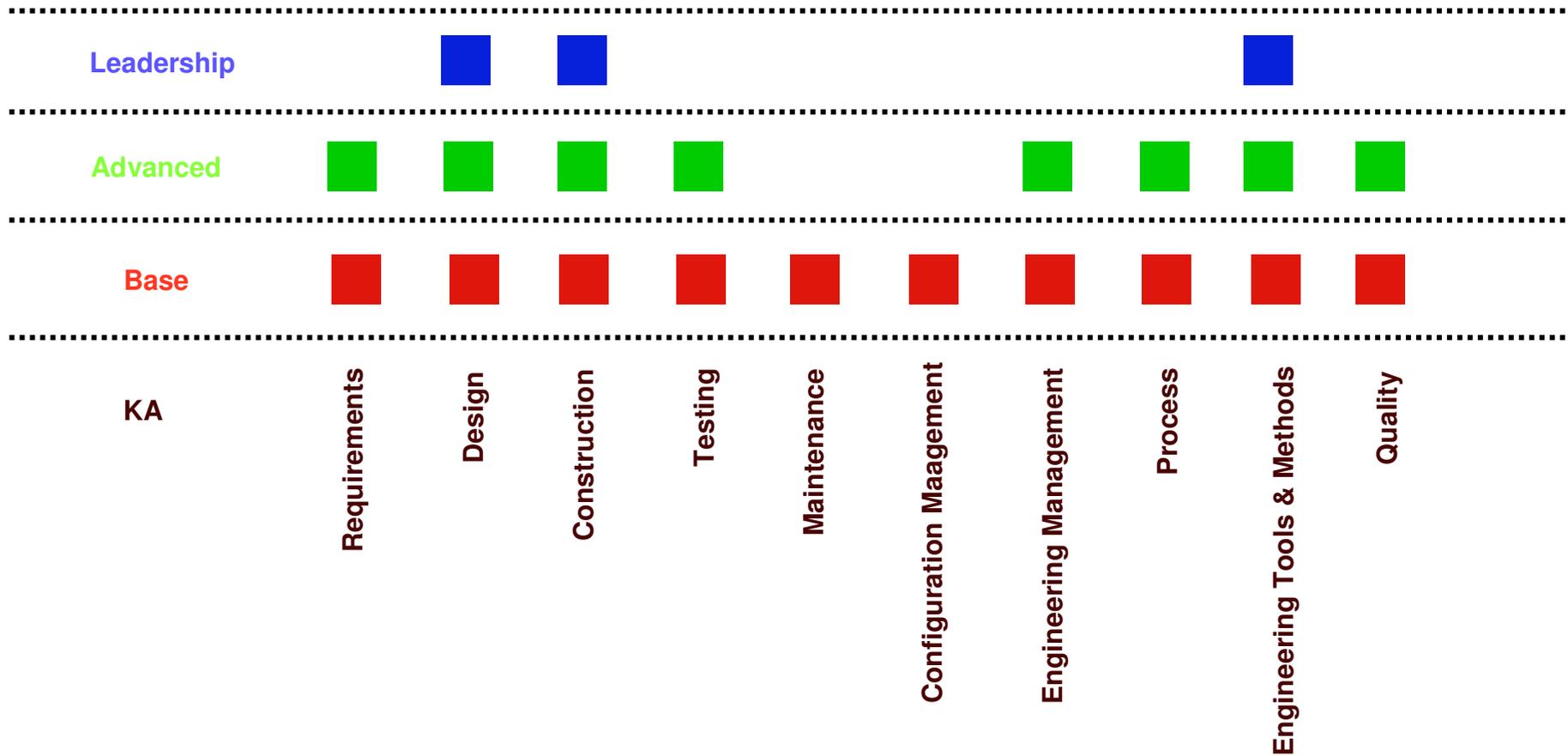
# Avanzamento nella carriera

Esempio di sviluppo professionale per raggiungere il “livello 11”



# Avanzamento nella carriera

*Esempio di sviluppo professionale per raggiungere il “livello 12”*



## Requisiti a livello “Base”

### *Esempio di sviluppo professionale in “Engineering Management”*

Attività	Requisiti
Lecture	<p>Lettura “analitica” del seguente materiale:</p> <ul style="list-style-type: none"> <li>• <i>Software Engineering Code of Ethics and Professionalism</i>, ACM/IEEE-CS</li> <li>• <i>Software Project Survival Guide</i>, Steve McConnell</li> <li>• “They Write the Right Stuff”, Charles Fishman</li> </ul> <p>Lettura “ispettiva” del seguente materiale:</p> <ul style="list-style-type: none"> <li>• <i>Software Engineering</i>, Parte I + Capitoli 22 e 23, Ian Sommerville (Nota 1)</li> <li>• <i>Software Engineering: A practitioner’s Approach</i>, Capitolo 4, Roger Pressman (Nota 2)</li> <li>• <i>201 Principles of Software Development</i>, Alan Davis</li> </ul>
Esperienza di lavoro	<ul style="list-style-type: none"> <li>• Revisione di un piano di progetto</li> <li>• Apprendimento di una tecnica di stima</li> <li>• Pianificazione e registrazione delle attività personali</li> </ul>
Corsi	Nessuno
Certificazione	Nessuna
Partecipazione esterna	Nessuna

**Nota 1:** Ian Sommerville, Parte I (Capp. 1-5): Descrizione; Cap. 22: Verifica e convalida; Cap. 23: Test del software.

**Nota 2:** Roger S. Pressman, Cap. 4: Metriche di processo e di progetto.

# Requisiti a livello “Advanced”

## Esempio di sviluppo professionale in “Engineering Management”

Attività	Requisiti
Lecture	<p>Lettura “analitica” del seguente materiale:</p> <ul style="list-style-type: none"> <li>• <i>No Silver Bullets-Essence and Accidents of Software Engineering</i>, Fred Brooks</li> <li>• <i>Programmer Performance and the Effects of the Workplace</i>, DeMarco and Lister</li> <li>• <i>Rapid Development</i>, Steve McConnell</li> <li>• <i>Manager’s Handbook for Software Development</i>, NASA Goddard Space Flight Center</li> </ul> <p>Lettura “ispettiva” del seguente materiale:</p> <ul style="list-style-type: none"> <li>• <i>Software Chronicles Crisis</i>, Way Gibbs</li> <li>• <i>Recommended Approach to Software Development</i>, NASA Goddard Space Flight Center</li> </ul>
Esperienza di lavoro	<ul style="list-style-type: none"> <li>• Partecipazione come revisore ai manufatti di un progetto</li> <li>• Partecipazione all’organizzazione di un progetto e creazione di un piano di progetto</li> <li>• Partecipazione alla creazione delle stime di un progetto; diventare esperto della tecnica di stima bottom-up; coordinare un’attività di stima</li> <li>• Diventare esperto nella creazione dei rapporti di stato di avanzamento dei progetti</li> <li>• Diventare esperto di pianificazione e relative tecniche (work breakdown structure, stime, gestione dei costi)</li> </ul>
Corsi	<ul style="list-style-type: none"> <li>• Software Project Survival Guide (2 gg)</li> <li>• Rapid Development (2 gg)</li> <li>• Software Estimation (2 gg)</li> </ul>
Certificazione	Nessuna
Partecipazione esterna	Nessuna

# Requisiti a livello “Leadership”

## Esempio di sviluppo professionale in “Engineering Management”

Attività	Requisiti
Lecture	Le lecture sono personalizzate: le aree da approfondire come pure la selezione degli articoli e libri sono concordati con il <i>mentore</i> ed evidenziati nel Piano di sviluppo personale. Un obiettivo di massima può essere costituito da circa 1.000 pagine.
Esperienza di lavoro	<ul style="list-style-type: none"> <li>• Coordinare le stime, la pianificazione ed il controllo di progetto significativo; costruire l'organizzazione del progetto ed il <i>business case</i>; creare il piano di un progetto di piccole dimensioni ed uno di grandi dimensioni; creare il piano delle attività di un progetto significativo; esperto nella scelta, personalizzazione e pianificazione del ciclo di sviluppo più adatto; diventare esperto delle tecniche di gestione del rischio; esperto delle tecniche di gestione dei problemi tipici di un progetto software.</li> <li>• Esperto in tecniche di stime (per analogia, parametriche, Delphi, ecc.); produrre la stima iniziale di un progetto con la tecnica top-down.</li> <li>• Creare la tempificazione (<i>scheduling</i>) di un progetto significativo con le <i>milestone</i>; produrre la tempificazione di dettaglio di una <i>milestone</i>; esperto delle tecniche di controllo dei cammini critici e di schedulazione (Gantt e PERT); esperto delle tecniche di reporting dello stato di avanzamento dei progetti.</li> <li>• Partecipazione in attività di consulenza o di coaching in Engineering Management.</li> </ul>
Corsi	<ul style="list-style-type: none"> <li>• Effective Software Project Management (3 gg)</li> <li>• Risk Management (2 gg)</li> <li>• Project Outsourcing (2 gg)</li> </ul>
Certificazione	<p>Una certificazione riconosciuta, per es.:</p> <ul style="list-style-type: none"> <li>• PMI's Project Management Professional certification, oppure</li> <li>• IEEE Computer Society's Certified Software Development Professional certification.</li> </ul>

## Requisiti a livello “Leadership”

*Esempio di sviluppo professionale in “Engineering Management”*

Attività	Requisiti
Partecipazione esterna	Partecipazione attiva e costruttiva al mondo dell'ingegneria del software con: <ul style="list-style-type: none"><li>• Creazione di corsi sul tema della gestione dello sviluppo software (serali, fine settimana, ecc.)</li><li>• Insegnamento all'interno dell'azienda o presso enti/università</li><li>• Partecipazione a gruppi di lavoro sul tema (comitati, panel, standard board, ecc.)</li><li>• Presentazioni in conferenze</li><li>• Pubblicazione di articoli in riviste specializzate</li><li>• Revisione di manoscritti per libri</li><li>• <i>Mentoring e coaching</i> di altri persone all'interno dell'azienda sui temi di cui si ha la leadership.</li></ul>

## “Best practice” poco conosciute o poco usate

*Utilizzate solo da organizzazioni di successo*

Best practice	Anno in cui è stata resa disponibile
<b>Project planning and management:</b> <ul style="list-style-type: none"> <li>• Automated estimation tools (strumenti di automazione delle stime)</li> <li>• Software measurement (misurazione del software)</li> <li>• Productivity environment (ambienti di produttività)</li> <li>• Risk management planning (gestione del rischio)</li> </ul>	<p>1973</p> <p>1977</p> <p>1984</p> <p>1981</p>
<b>Requirements engineering practices:</b> <ul style="list-style-type: none"> <li>• Change Board (gestione delle modifiche)</li> <li>• User interface prototyping (prototipazione delle interfacce utente)</li> <li>• JAD sessions (sessioni JAD)</li> </ul>	<p>1978</p> <p>1975</p> <p>1985</p>
<b>Design practices:</b> <ul style="list-style-type: none"> <li>• Information hiding (mascheramento delle informazioni)</li> <li>• Design for change (progettazione evolutiva)</li> </ul>	<p>1972</p> <p>1979</p>
<b>Construction practices:</b> <ul style="list-style-type: none"> <li>• Source code control (controllo del codice sorgente)</li> <li>• Incremental integration (integrazione incrementale)</li> </ul>	<p>1980</p> <p>1979</p>

Continua ...

# “Best practice” poco conosciute o poco usate

*Utilizzate solo da organizzazioni di successo*

Best practice	Anno in cui è stata resa disponibile
<b>Quality assurance practice:</b> <ul style="list-style-type: none"><li>• Branch-coverage testing (validazione delle condizioni di scelta)</li><li>• Inspections (ispezioni/revisioni tecniche)</li></ul>	1979 1976
<b>Process improvement:</b> <ul style="list-style-type: none"><li>• Software Engineering Institute’s Software Capability Maturity Model (SEI-CMM)</li><li>• Software Engineering Process Group (SEPG)</li></ul>	1987 1989

**Guide to the Software Engineering  
Body of Knowledge**

**2004 Version**

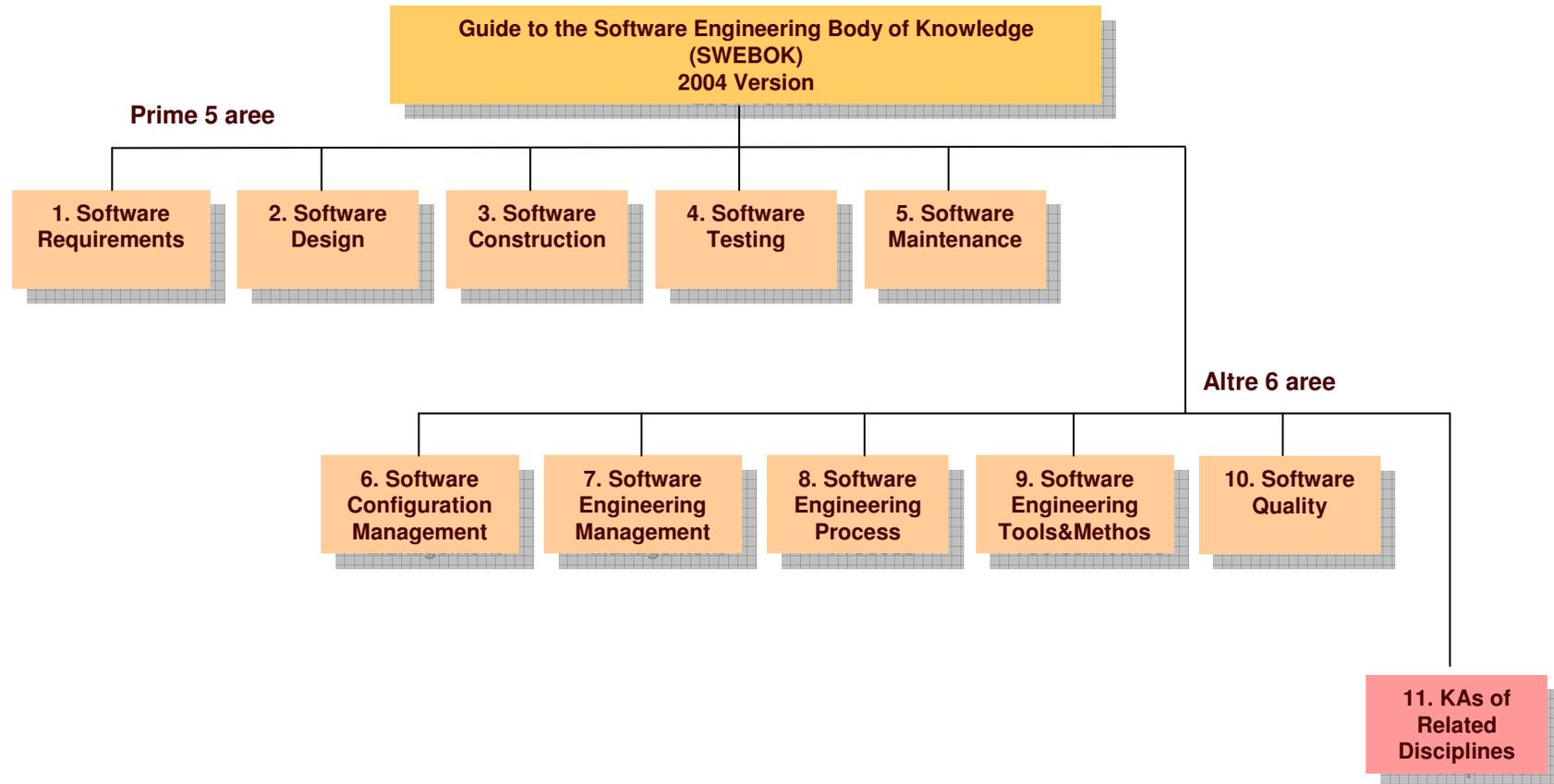
**SWEBOK\***

**A project of the IEEE Computer Society  
Professional Practices Committee**

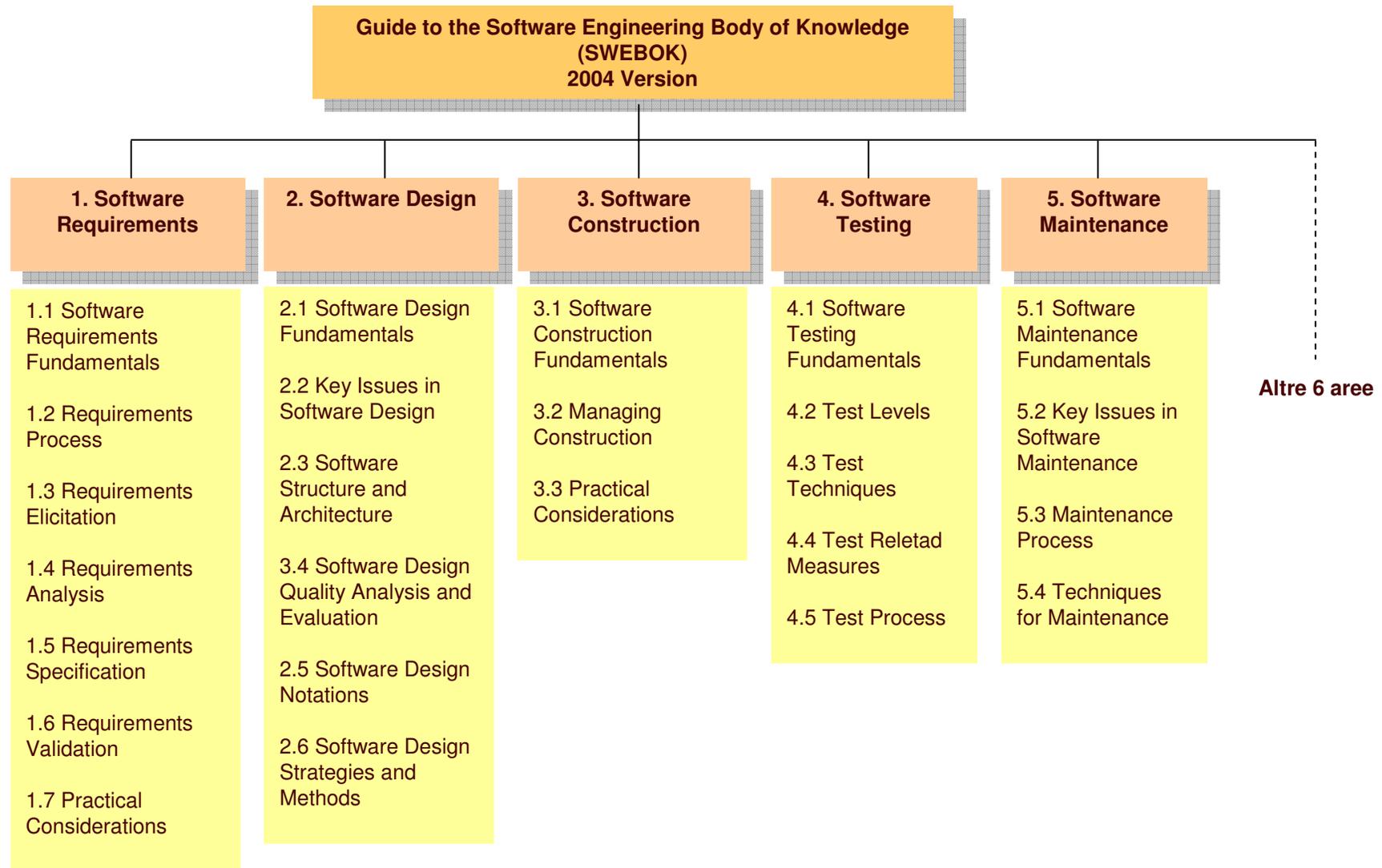
© IEEE – 2004 Version

® SWEBOK is an official service mark of the IEEE

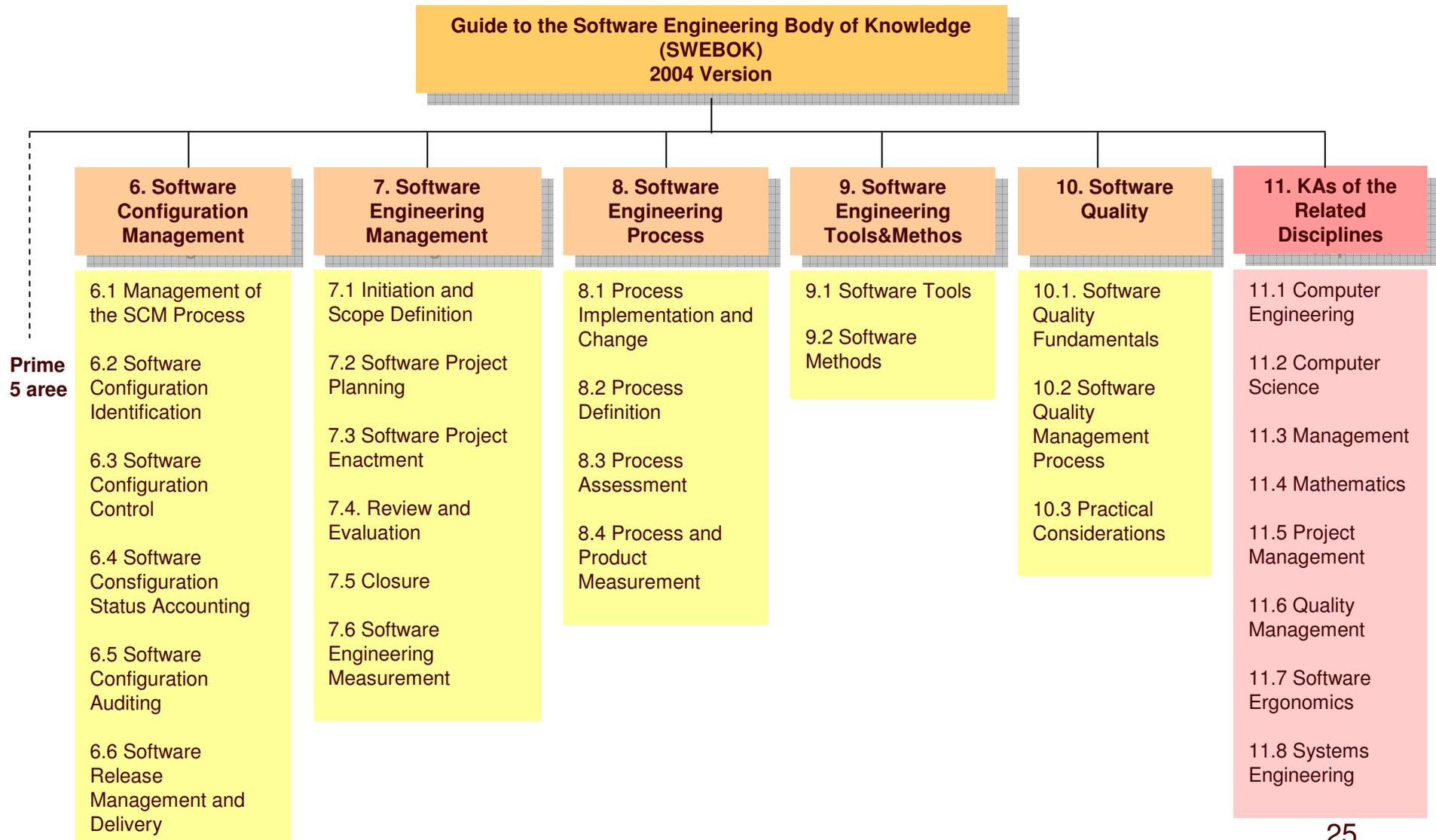
# Una visione d'insieme



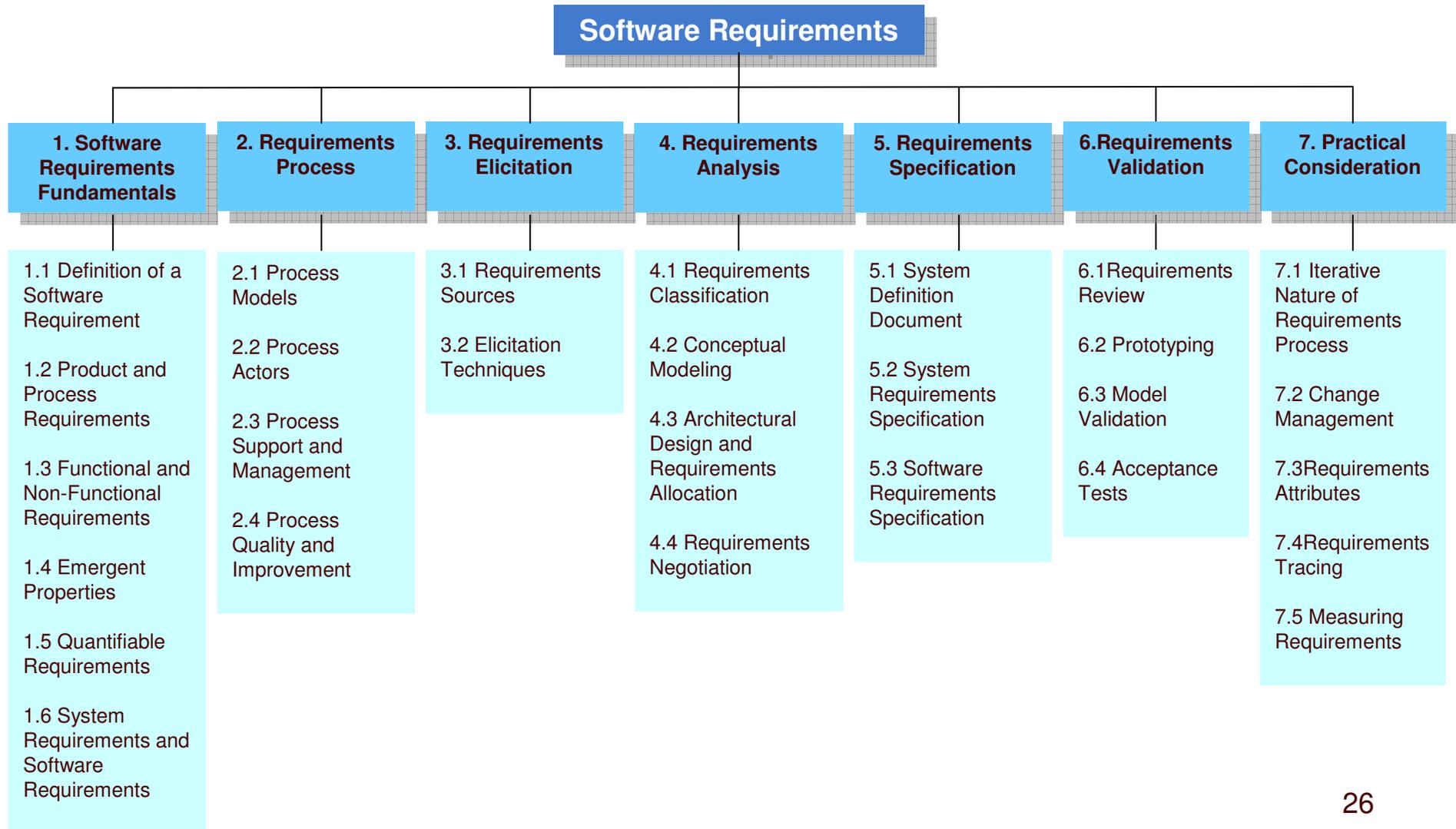
# Le prime 5 aree di competenze



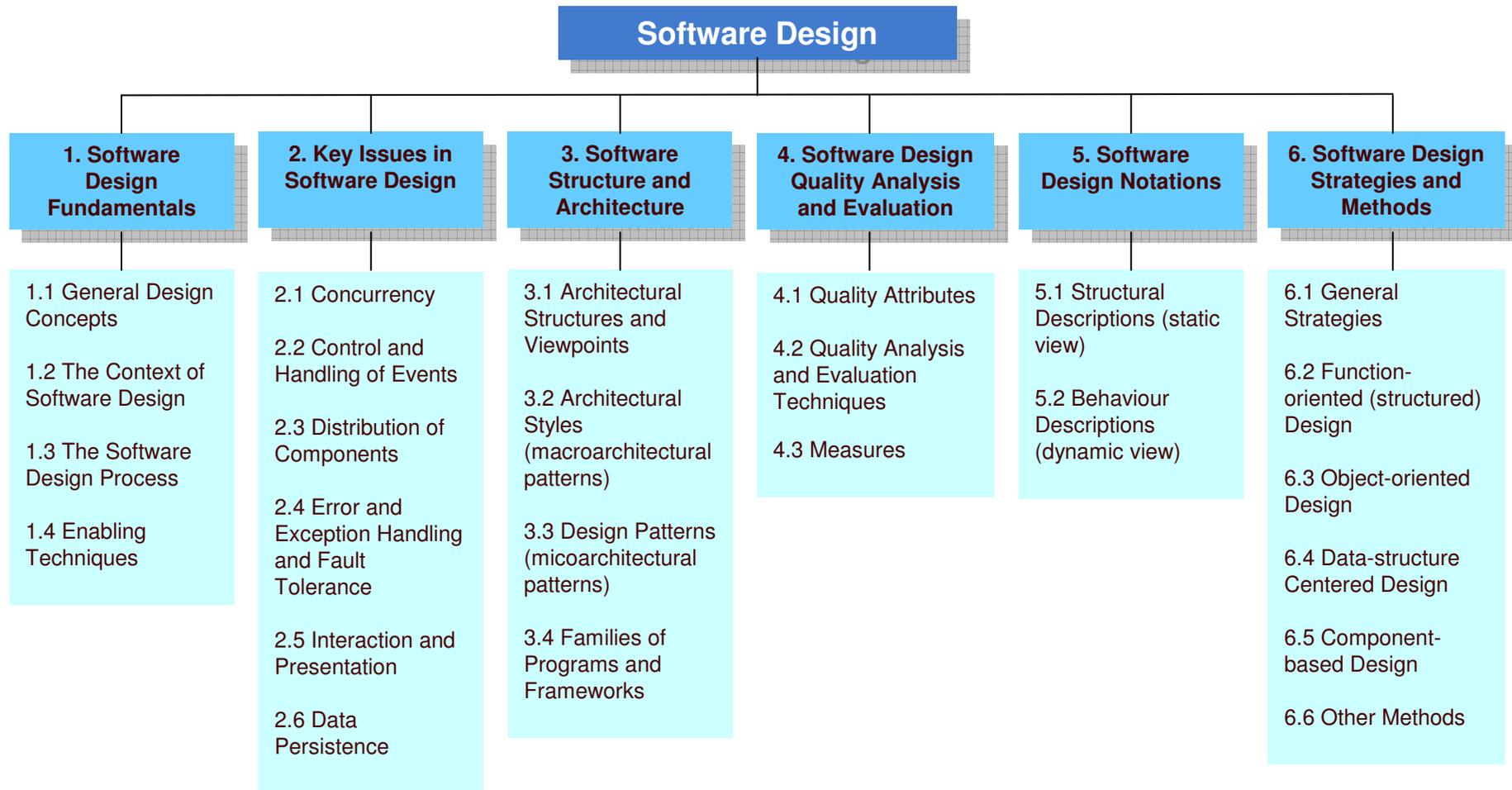
# Le altre 6 aree di competenza



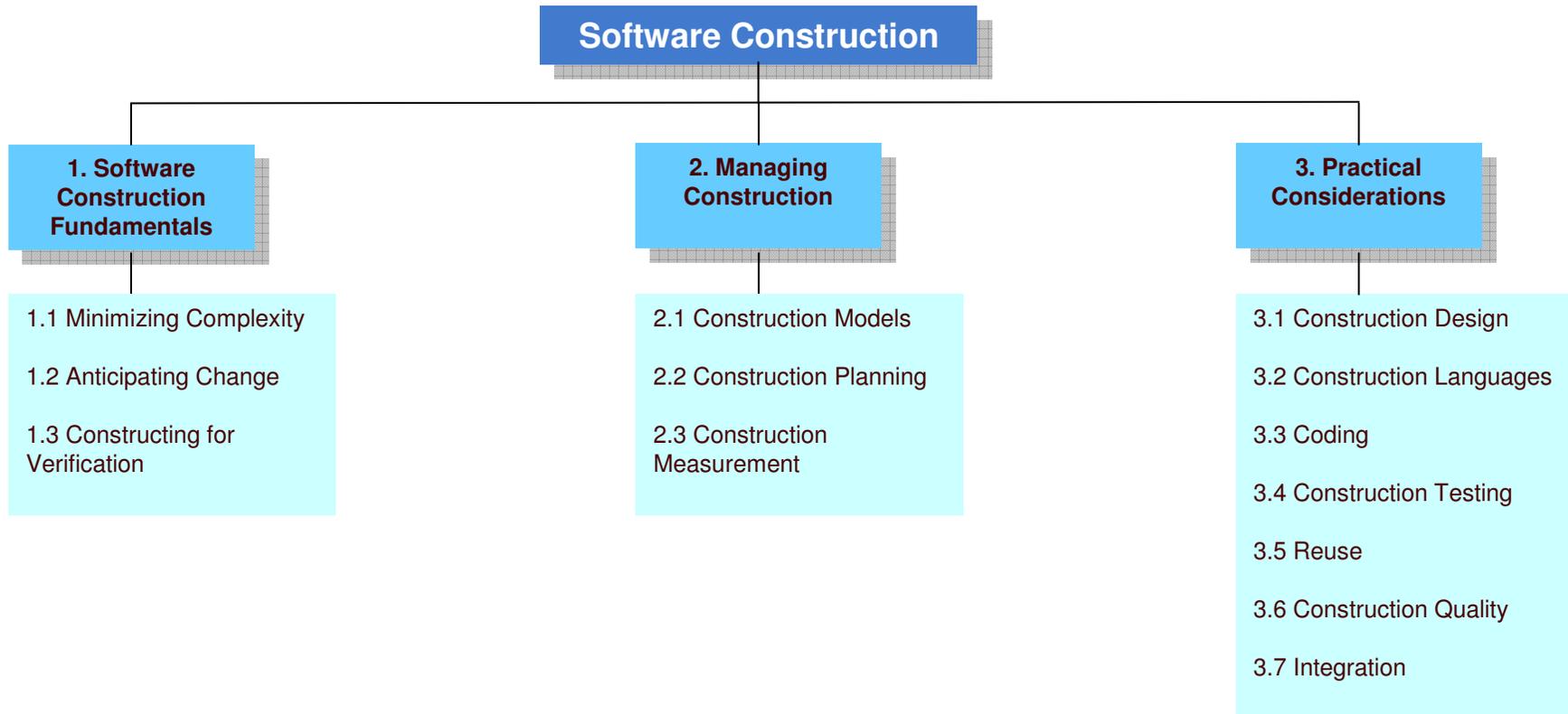
# 1. Requisiti del software



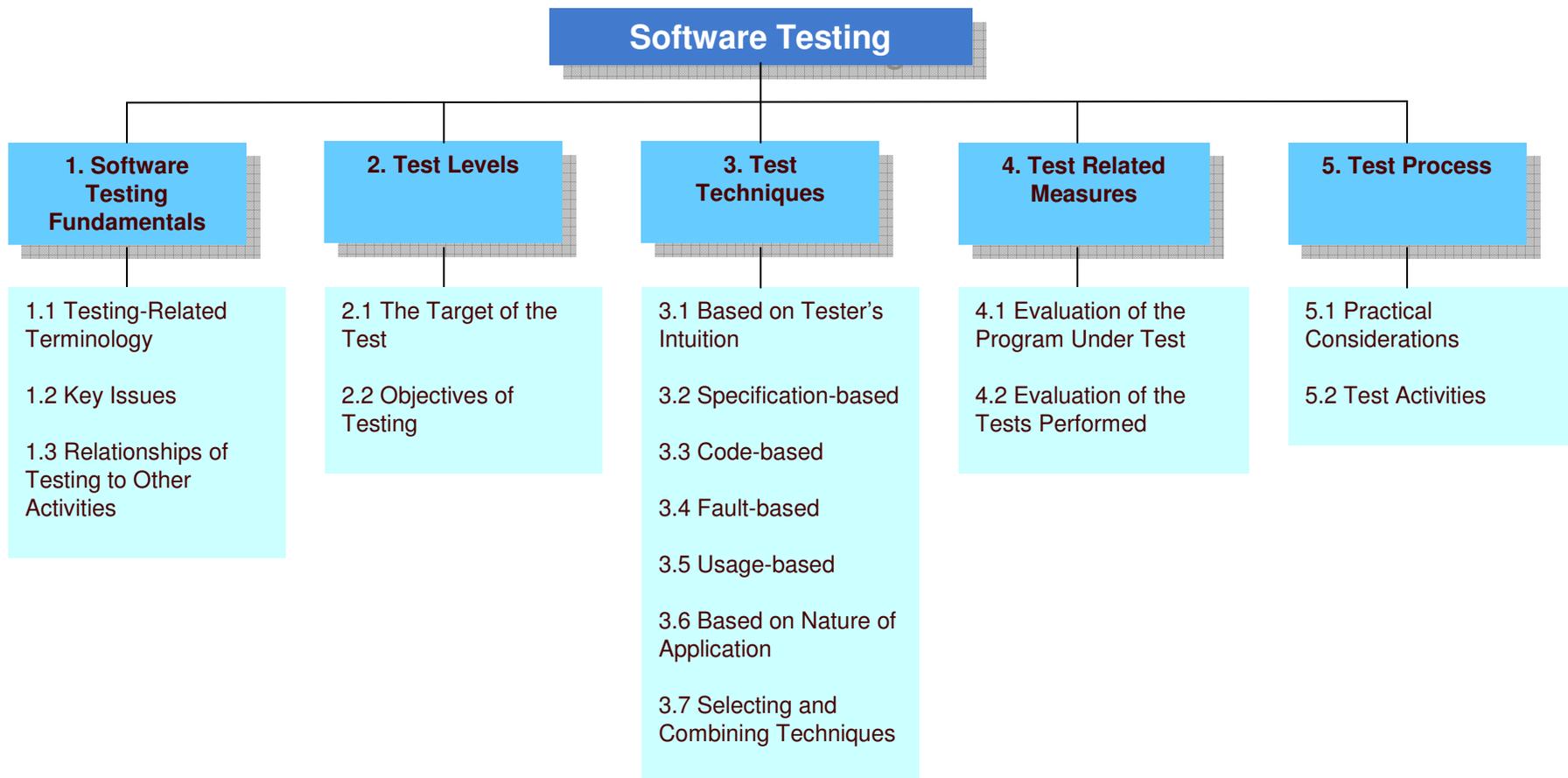
## 2. Progettazione del software



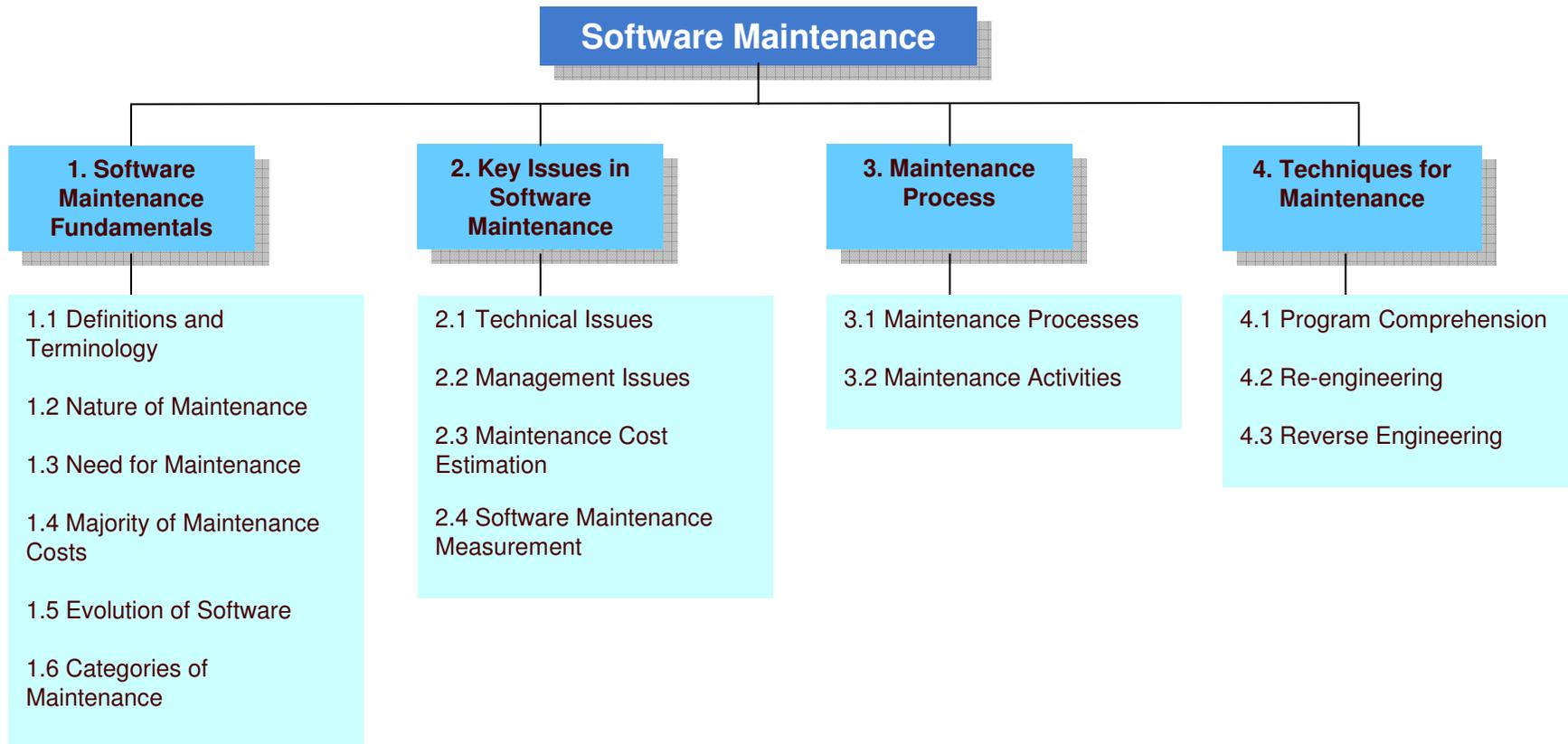
## 3. Sviluppo del software



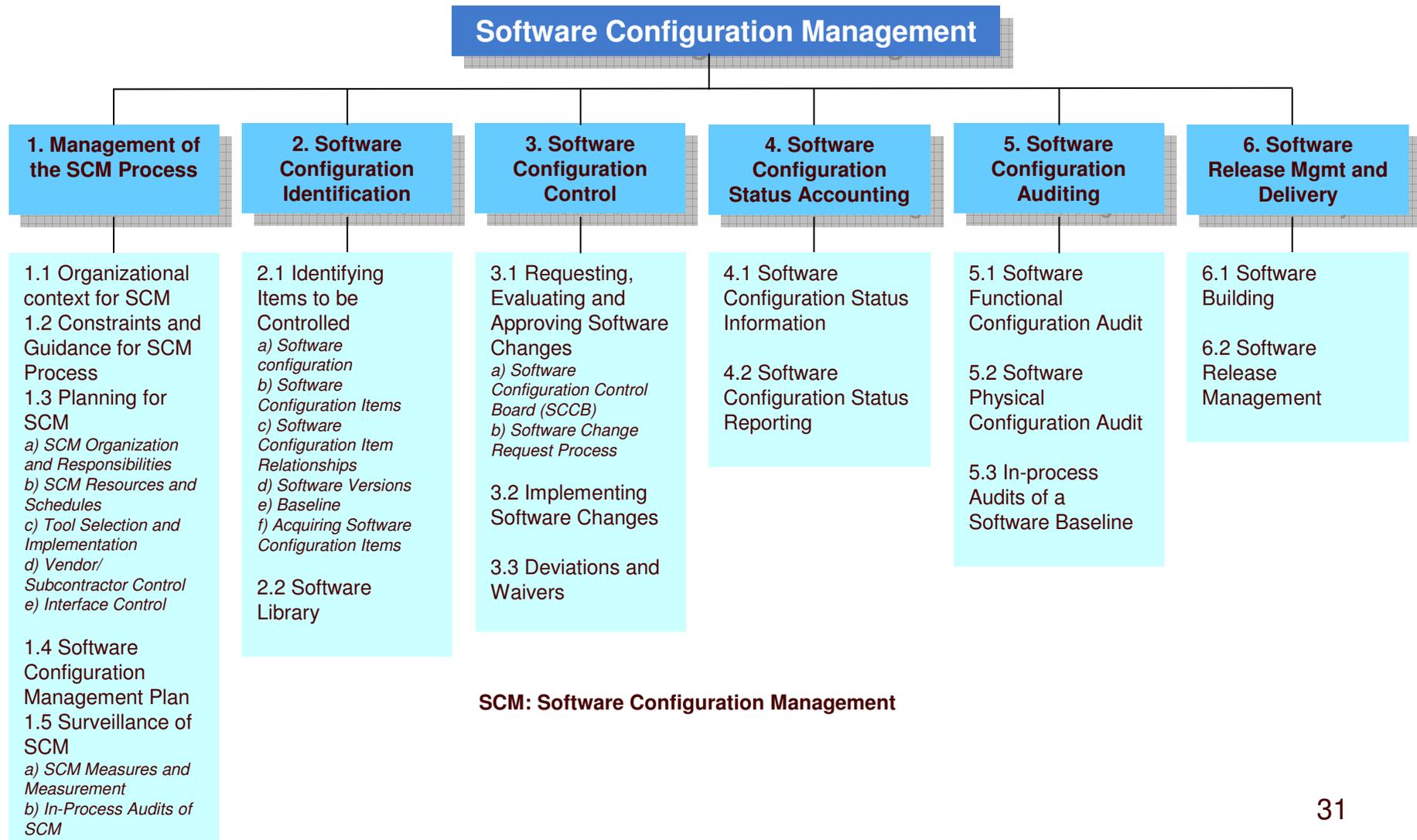
## 4. Testing del software



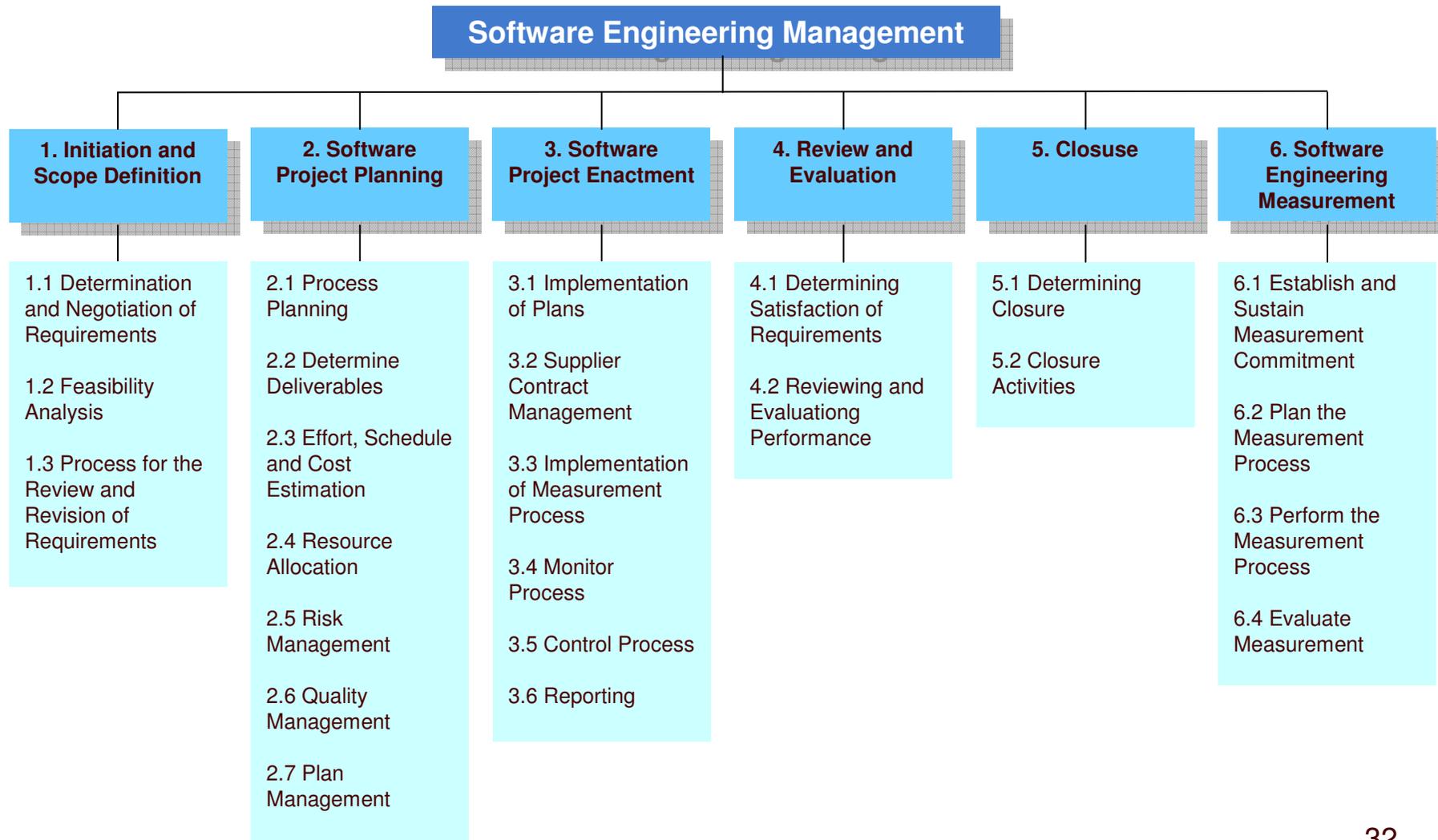
## 5. Manutenzione del software



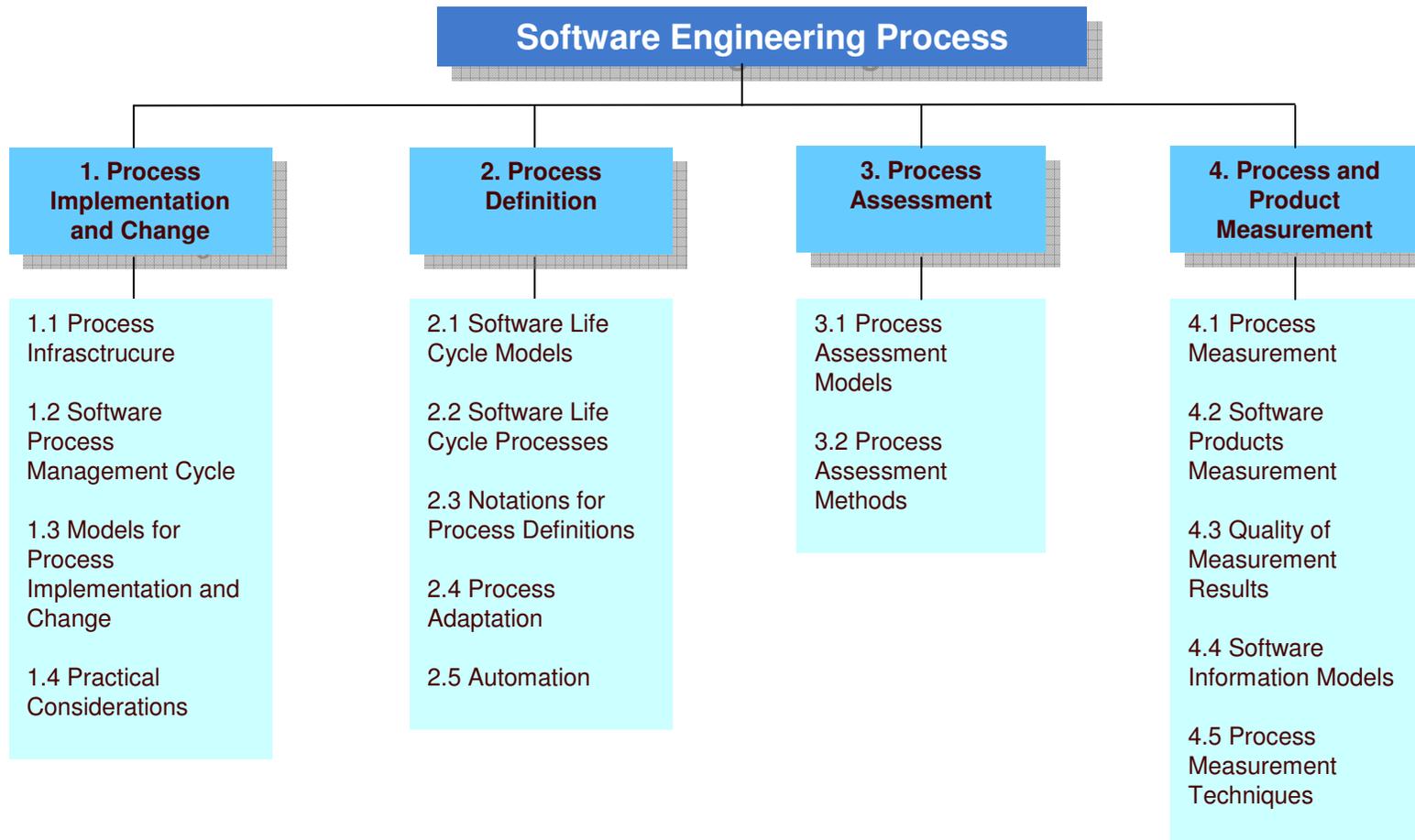
## 6. Gestione della configurazione software



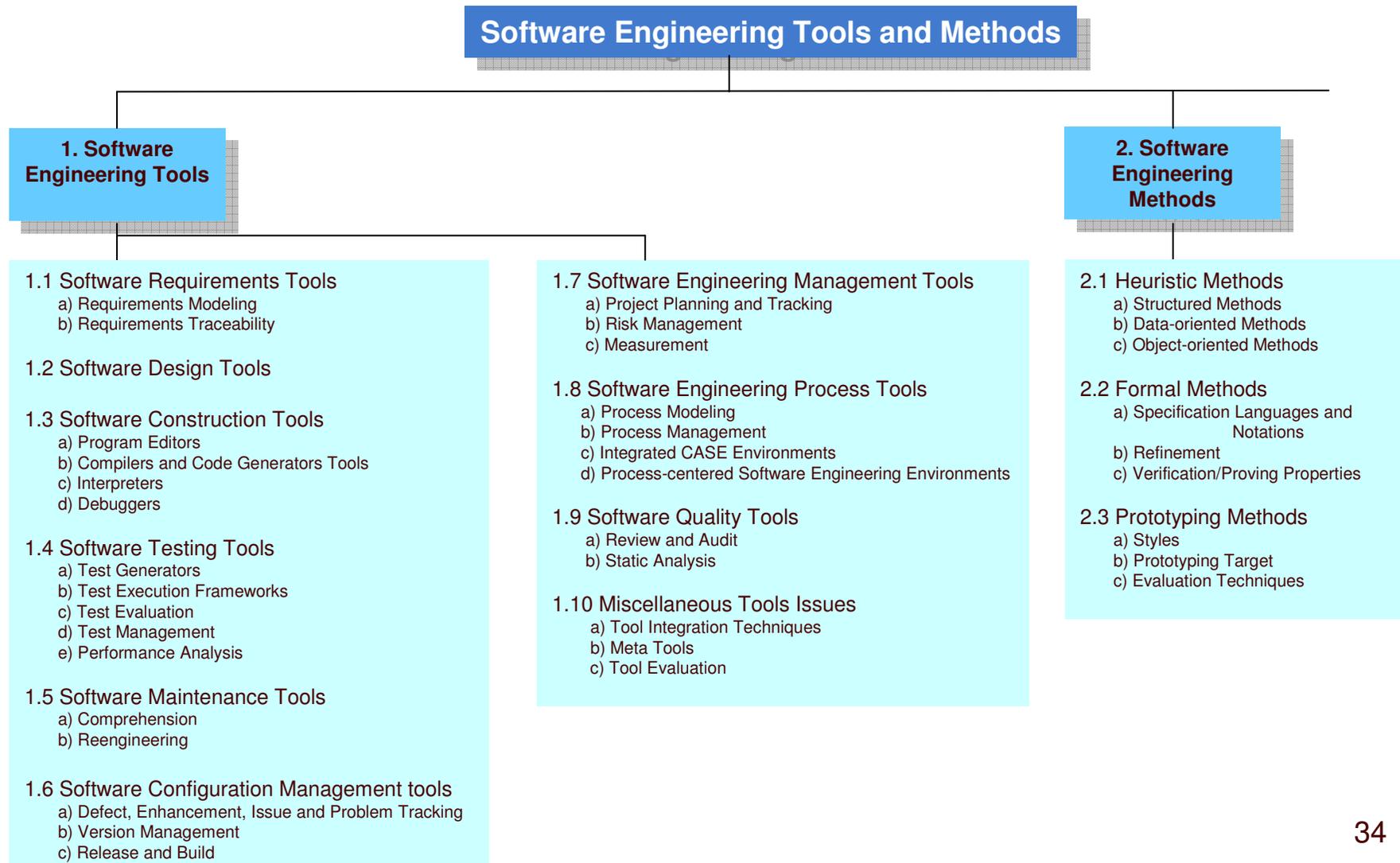
## 7. Gestione dello sviluppo del software



## 8. Processo di sviluppo software



# 9. Strumenti e metodi per lo sviluppo del software

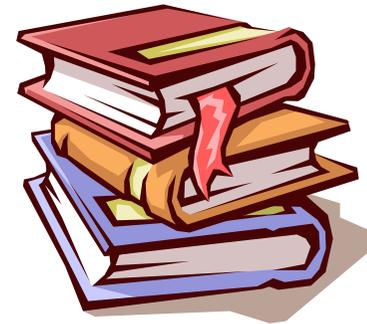


# 10. Qualità del software



## Lecture consigliate (raccomandate)

- ❖ Steve McConnell – *Professional Software Development* - Addison-Wesley (2003)
- ❖ IEEE Professional Practices Committee – *Guide to the Software Engineering Body of Knowledge* – IEEE Computer Society (2004)
- ❖ Roger S. Pressman – *Software Engineering: A Practitioner's Approach* – McGraw-Hill (2000)
- ❖ Ian Sommerville – *Software Engineering 07 Edition* – Pearson Addison Wesley (2004)
- ❖ Frederick P. Brooks, Jr . - *The Mythical Man-Month* - Addison-Wesley (1975, 1995 Anniversary Edition)
- ❖ Capability Maturity Model® Integration, Version 1.1 – Staged Representation - SEI (2002)
- ❖ Glenford J. Myers - *The art of Software Testing* - Jhon Wiley & Sons, New York (1979)



# Siti da visitare

**Software Engineering Institute (SEI):** <http://www.sei.cmu.edu/sei-home.html>

**Software Engineering Body of Knowledge (SWEBOK):** <http://www.swebok.org/>

**Professional Software Development:** [www.construx.com/profession](http://www.construx.com/profession)

**Ian Sommerville:** [www.pearsoned.co.uk/sommerville](http://www.pearsoned.co.uk/sommerville)

**R.S.Pressman:** [www.mcgraw-hill.it](http://www.mcgraw-hill.it)

[www.ateneonline.it/informatica](http://www.ateneonline.it/informatica)

[www.hyperbook.it](http://www.hyperbook.it)

