



Analisi dei requisiti

Anno accademico 2020/2021

Ingegneria del Software

Tullio Vardanega, tullio.vardanega@unipd.it





Acquisizione e fornitura

- ❑ **Il committente studia il problema e definisce le proprie attese sul prodotto**
 - Le fissa in un **capitolato d'appalto** che determina i «requisiti utente»
- ❑ **Chi si candida a fornitore deve capire come corrispondere a quelle attese**
 - Fissando le implicazione di interesse, di costo e di qualità: **studio di fattibilità**
- ❑ **Il progetto nasce quando committente e fornitore trovano un accordo contrattuale al riguardo**
 - Che fissa gli obiettivi (**analisi dei requisiti**) e i costi del progetto
- ❑ **Il contratto finisce con la valutazione del committente sul prodotto**
 - Collaudo, accettazione



□ **“Requisito”** secondo il glossario IEEE

1. Capacità/*capability* necessaria a un utente per risolvere un problema o raggiungere un obiettivo

○ Visione dal lato del bisogno

2. Capacità/*capability* che deve essere posseduta (o condizione che deve essere soddisfatta) da un sistema per adempiere a un obbligo

○ Visione dal lato della soluzione

3. Descrizione documentata di una capacità interpretata come in 1 o 2



- **Cause di abbandono (Standish Group 1995)**
 - **Requisiti incompleti**
 - **Insufficiente coinvolgimento del cliente/utente**
 - **Attese irrealistiche**
 - **Scarsità di risorse**
 - **Volatilità di specifiche e requisiti**
 - **Insufficiente competenza del fornitore**



Studio di fattibilità – 1

- **Valutare rischi, costi e benefici**
 - **Nell'ottica del cliente e del fornitore**
 - Competenze richieste/disponibili, prospettive future, competizione
 - **Studio basato su dati vari e spesso incerti**
 - **Definizione e valutazione di possibili scenari**

- **Decidere se procedere**
 - **Con l'obiettivo di restare entro un costo massimo prefissato**

- **Con le conoscenze immediatamente disponibili**
 - **E con un piano di formazione sostenibile**



Studio di fattibilità – 2

- **Fattibilità tecnico-organizzativa**
 - Strumenti e tecnologie per la realizzazione
 - Soluzioni algoritmiche e architettoniche
 - Piattaforme idonee per l'esecuzione
- **Rapporto costi/benefici**
 - Confronto tra il mercato attuale e quello futuro
 - Costo di produzione vs. redditività dell'investimento
- **Individuazione dei rischi**
 - Complessità e incertezze



Studio di fattibilità – 3

- **Valutazione delle scadenze temporali**
 - Risorse disponibili vs. risorse necessarie
- **Valutazione delle alternative**
 - **Scelte architettoniche**
 - Esempi: sistema centralizzato o distribuito; modello *client-server*, ...
 - **Strategie realizzative**
 - "*Make or buy*": riuso o sviluppo ex-novo



Sui requisiti e sull'analisi

- ❑ I requisiti riflettono l'ambiente d'uso, nella sua evoluzione da senza a con il prodotto, non la sua realizzazione
 - Capire il «senza» per comprendere il «con», con terminologia coerente con il dominio d'uso
- ❑ L'analisi riflette la struttura funzionale del prodotto
 - Lo scenario «Premere un pulsante scatena un calcolo che legge o scrive dati» va trattato come gerarchia, dall'esterno all'interno, non come flusso
 - Il punto di vista dell'analisi è sempre e solo quello del «lato utente»
- ❑ Questa è la prospettiva dei **diagrammi dei casi d'uso**
 - Relazione tra «attore» e «sistema»: l'attore è chi o cosa possa fare richieste alla parte di prodotto alla quale è esposto (sistema)



Attività di analisi

- ❑ Studio dei bisogni e delle fonti del dominio d'uso
- ❑ Comprensione del problema dal lato dei bisogni
 - Approfondimenti tramite **scenari** d'uso / *use case*
- ❑ Raggruppamento degli scenari per affinità
 - Per individuare possibili parti del sistema
- ❑ Classificazione e tracciamento dei requisiti
 - In dialogo con il committente (e gli altri *stakeholder*)



Studio del dominio

- ❑ **Domande base**
 - A quali bisogni risponde il prodotto atteso
 - Quali problematiche d'uso esso comporta
- ❑ **Acquisizione delle conoscenze**
 - Documentazione e soluzioni preesistenti
 - Interviste agli utenti
- ❑ **Consolidamento del glossario**
 - Raccoglie e definisce i termini chiave del dominio
 - Per interazione ordinata con il committente
 - Consolidato nel corso del progetto



Tecniche – 1

- ❑ **Dominio d'uso come fonte di requisiti impliciti**
 - Comportamenti dell'utente e dell'ambiente d'uso
- ❑ **Interazione con il cliente / committente**
 - Interviste
 - Generazione, analisi e discussione di scenari
- ❑ **Discussioni creative e collaborative (aka *brainstorming*)**
- ❑ **Prototipazione**
 - Interna (solo per il fornitore)
 - Esterna (per discussione con il cliente)

Esito documentato
in **verbali** con
valore contrattuale



Qualità dell'analisi – 1

- La specifica dei requisiti deve essere
 - Priva di ambiguità (UNAMBIGUOUS)
 - Corretta (CORRECT)
 - Completa (COMPLETE)
 - Verificabile (VERIFIABLE)
 - Consistente (CONSISTENT)
 - Modificabile (MODIFIABLE)
 - Tracciabile (TRACEABLE)
 - Ordinata per rilevanza (RANKED)

IEEE 830-1998

*Recommended Practice for
SW Requirements Specifications*



Qualità dell'analisi – 2

GOOD REQUIREMENTS CHECKLIST

COMMENTARY BELOW THE BULLETED CHECKLIST ITEMS MAY GUIDE REQUIREMENTS IMPROVEMENTS EFFORTS

The following 10-item list (with supplemental notes) serves to assess areas where requirements quality may be improved, guide discussions among stakeholders, and—only if absolutely necessary—produce a metric to report to management. The statements below apply to both individual requirements and the specification as a whole.

- **Complete:** Contains sufficient information to drive forward the work of those who need to use the information at this point in the lifecycle, at an acceptable level of risk.

"Completeness" cannot be a single binary toggle and is not merely a reflection of whether all fields in a template are filled in but must be assessed continually, based on the changing needs of both the project and developers over time. Who are the consumers of the artifacts you create? What do they need today to do their work, and what do you know today about the substance of the artifacts? How will your knowledge and their needs progress together across the lifecycle to produce the highest-quality, most useful content at precisely the right time?

- **Correct:** Stakeholder and SME reviews locate no errors, content is consistent with source materials, and artifacts have been reviewed and accepted by appropriate constituents.

Who gets to decide if requirements are "done"? I ask this in every class I teach at work, usually to a few moments of silence before someone speaks up: "Maybe the person who has to actually use them?" Yes—consumers of the requirements are the ultimate arbiters of requirements. But see also *Traceable*, where we move a step beyond mere evidence that a requirement can be traced to source material.

- **Concise:** Addresses a single idea or concept, expressed in as few words as possible. Supporting information is separate from the requirement statement itself.

The advent of commercially available natural language processing (NLP) requirements tools has made assessment of a statement's conciseness (also called "atomic" by some checklists) somewhat simpler. Mere mechanical assessment of conjunctions or verb counts provide insight, but what else can be explored through a discussion of the intent behind the statement? Does the level of abstraction of the requirement matter?

- **Feasible:** Requirements are known to be feasible through use in prior products, prototyping, or analysis.

I have the privilege of working with some of the most brilliant people I've encountered anywhere in the world, people who take "not possible" as a challenge to make it happen anyway. Propose an idea to them and ask if it is feasible, and the answer will invariably be "Yes." Nothing's impossible, given enough time and resources—but time and resources are finite, so bound the discussion and, perhaps, discuss how to stage steps toward the end goal, particularly if it currently appears to violate all known practice.

- **Necessary:** Market segment, business strategy, usage model, technical feasibility, or legal or sustainability constraints require that this statement exist.

Separating needs from wants is always a challenge. There are no free features—any requirement artifact is a claim on planning, architecture, development, test, and other resources, human as well as financial. Do all stakeholders share the same vision of product needs?

- **Prioritized:** Tradeoffs between requirements are clear, explicit, and understood by stakeholders, and they have been assessed across multiple dimensions of market and technical considerations.

Some programs may find value in a strict high–medium–low prioritization scheme; others may simply

(Continued on next page)

GOOD REQUIREMENTS CHECKLIST (CONT.)

COMMENTARY BELOW THE BULLETED CHECKLIST ITEMS MAY GUIDE REQUIREMENTS IMPROVEMENTS EFFORTS

assess whether requirements are present or absent for a particular variant—included, or left in a backlog to address later—if at all. More significant than whether a priority has been assigned to a requirement may be the question of whether prioritization is assessed frequently through a process of backlog grooming, ensuring that what is done at any particular time is the highest-value, greatest-need work.

- **Unambiguous:** Each requirement is well understood by the intended audience and has only one possible interpretation.

NLP tools also shine at identifying ambiguity in requirements. However, just as requirements prioritization is not an objective science, neither is the effect of ambiguity, as increasing recent research suggests. If a set of requirements all contain the canonically ambiguous verb "support," must they all be revised? The only reasonable answer is, "It depends." Does the risk posed by the ambiguity of "support" exceed the risk of taking the time to explore a better verb to express a concept that may be well understood and even already coded and tested on other projects? Sometimes yes—but not always.

- **Verifiable:** Implementation of each requirement can be effectively established prior to product release, and qualitative requirements are quantified with a target value and, where applicable, a description of the scale and meter with which they will be measured.

If you can't test it, how will you know if you've met it? Agile provides one possible answer in the form of a product owner or other empowered customer representative who can raise the "good enough" flag. Particularly in contexts in which tools and work pro-

cess support extensive reuse, tailoring the target value while leaving the core of the requirement itself alone can accelerate progress.

- **Consistent:** Each requirement is represented only once in a specification and is internally and externally consistent with all others.

In a complex, multicustomer system-of-systems context, we are exploring the concept that a colleague has dubbed "purposeful duplication." Different stakeholders may have the same requirements or different ones and, likely, a combination of each. Maintaining the tension of the inconsistency of asks while assessing their overlaps and commonality as well is a delicate balancing act, and it may also require some acceptance of inconsistency until the last responsible moment for a decision to be made about a release, which may be far beyond our traditional comfort zones. This tension can only be maintained with careful and well-executed traceability.

- **Traceable:** Requirements are traceable if they are concise (only one concept per requirement, please) and uniquely and persistently identified with a tool-generated or other identifier.

However, mere traceability of requirements is insufficient to truly satisfy this attribute in a context that measures quality as "quality in use." On the programs with which I work now, we are beginning to assess the quality of a requirement's traceability if it is, in fact, in use and traced. Why is this artifact here? What need does it satisfy, and is that need itself traced to a higher-order request that includes a rationale for its claim on resources? Have the consumers of that requirement signed off on its fitness for purpose?



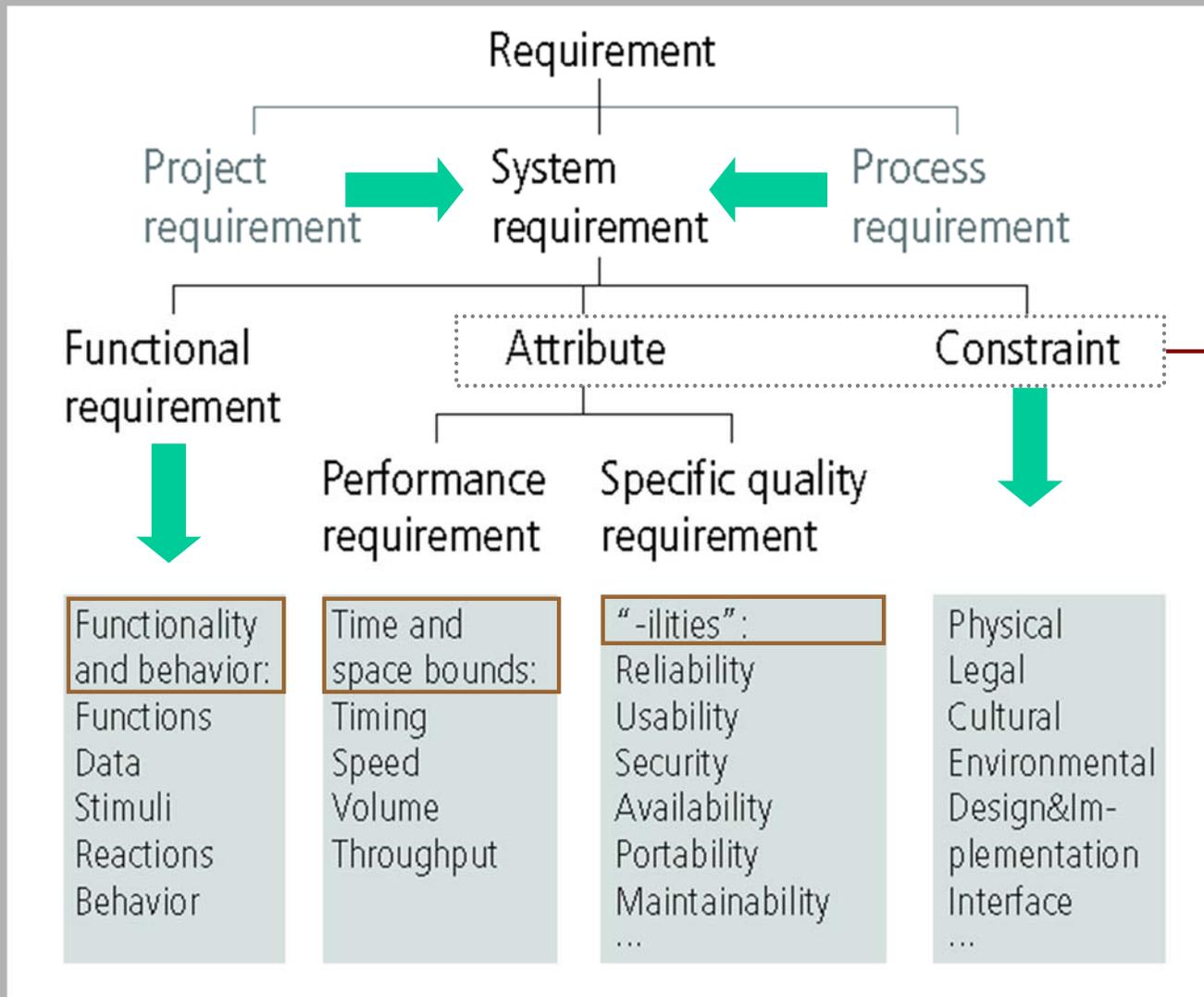
Tecniche – 2

- Per avere quelle qualità conviene che i requisiti siano «a grana fine»
 - I requisiti utente spesso sono «a grana grossa»
 - Lo studio del problema li affina, suddividendoli, e ne aggiunge, ampliando la ricerca

- È utile «sapere dove cercare»
 - La classificazione dei requisiti aiuta a farlo
 - Rispondendo alla domanda: che tipo di requisiti applicano al prodotto?



Classificazione dei requisiti – 1



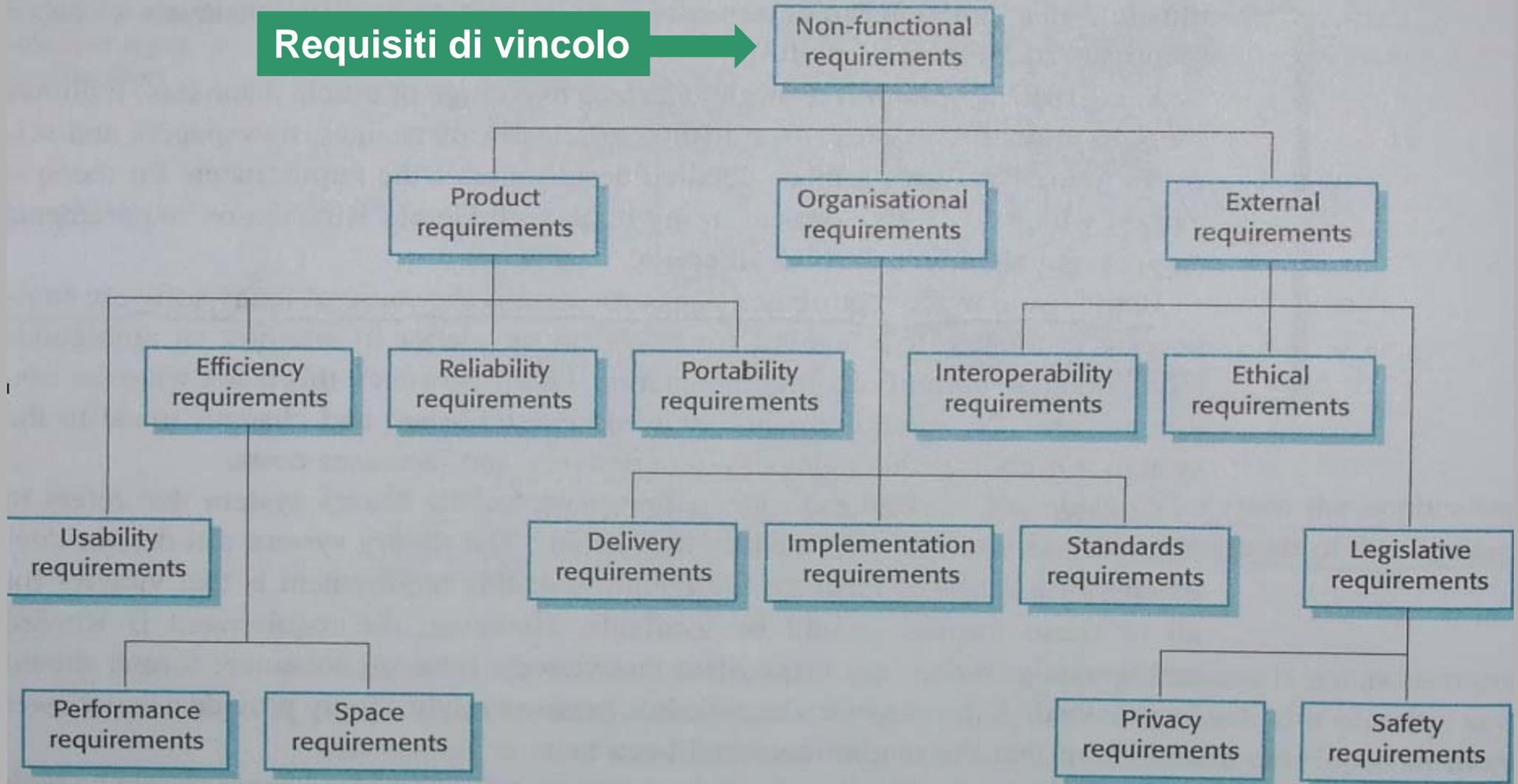
Extra-functional requirement

M. Glinz,
On Non-Functional Requirements.
15th IEEE International Conference
on Requirements Engineering.
October 2007, 21-26



Classificazione dei requisiti – 2

Requisiti di vincolo





Classificazione dei requisiti – 3

- ❑ **I requisiti hanno diversa rilevanza e utilità, che va negoziata e concordata con il committente**
 - **Obbligatoria**
 - Irrinunciabili per qualcuno degli *stakeholder*
 - **Desiderabili**
 - Non strettamente necessari ma a valore aggiunto riconoscibile
 - **Opzionali**
 - Relativamente utili oppure contrattabili più avanti nel progetto
- ❑ **Non devono essere tra loro contraddittori**



Punto di arrivo

- ❑ **Culmine del progetto è il collaudo (**validazione**) del prodotto**

 - In esso il fornitore dimostra che tutti i requisiti siano soddisfatti
 - La validazione concerne i requisiti di lato committente, ma i requisiti dello sviluppo sono molti di più
 - Il prodotto è efficace se soddisfa pienamente tutti i requisiti

- ❑ **Il fornitore assicura che i requisiti non esposti al collaudo siano anch'essi validati**
 - Usando un processo di verifica incrementale, che non ne dimentica alcuno (**tracciamento**)
 - L'analisi deve facilitare la verifica dei requisiti
- ❑ **Chi fissa un requisito deve immaginare come verificarne il soddisfacimento**
 - Facendo attenzione al costo e complessità di verifica



□ Verifica

- Accertare che l'esecuzione di attività non introduca errori
 - *Did I build the system right?*
- Attenzione rivolta al *way of working*
 - Rispetto delle regole, convenzioni e procedure vigenti nell'attuazione dei processi

□ Validazione

- Accertare che il prodotto corrisponda alle attese
 - *Did I build the right system?*
- Attenzione rivolta al prodotto finale

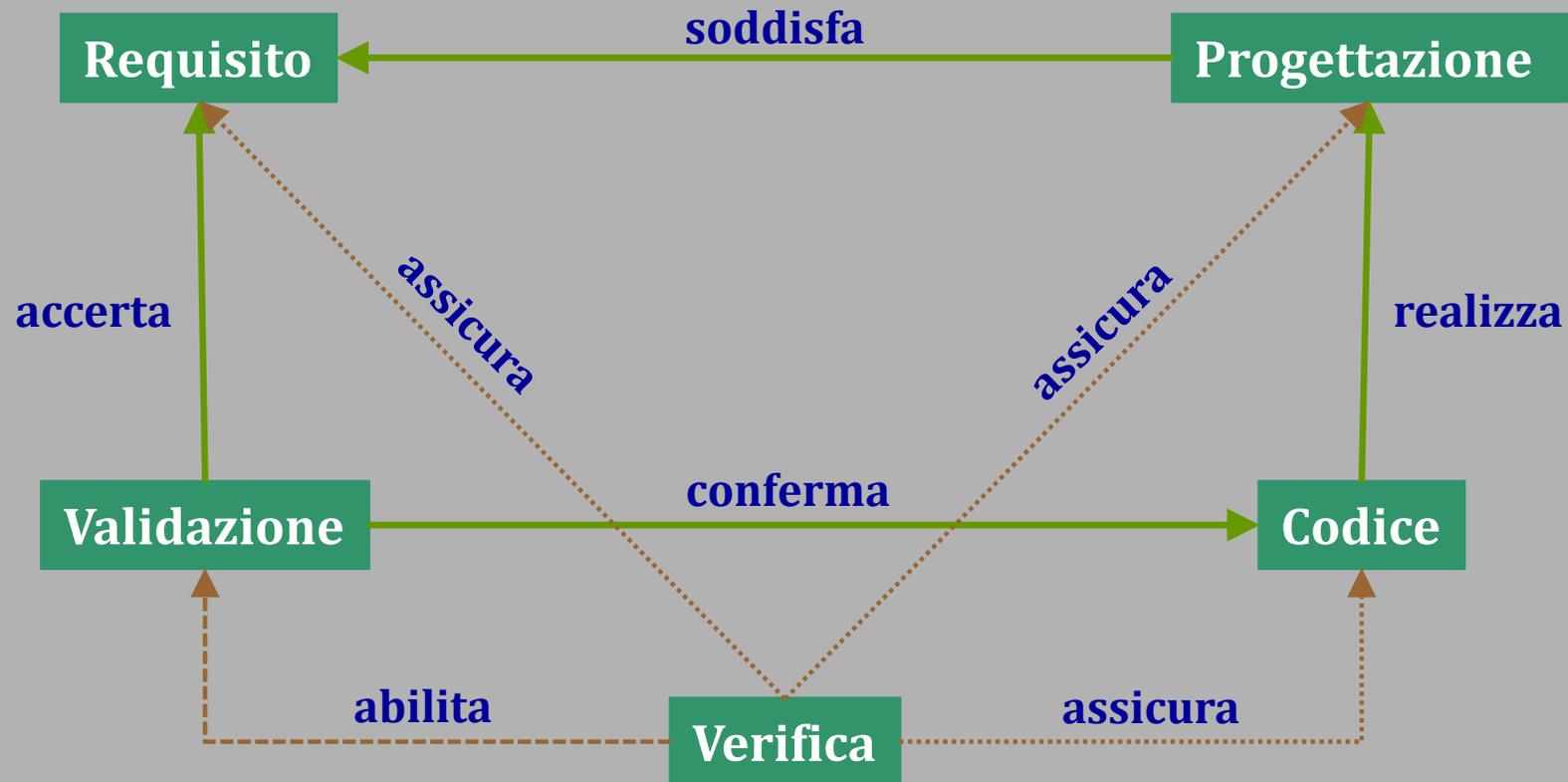
V & V = Qualifica

□ Piano di qualifica

- Dire come svolgeremo le attività di V&V nel progetto
 - Metodi, tecniche, procedure, strumenti e tempi

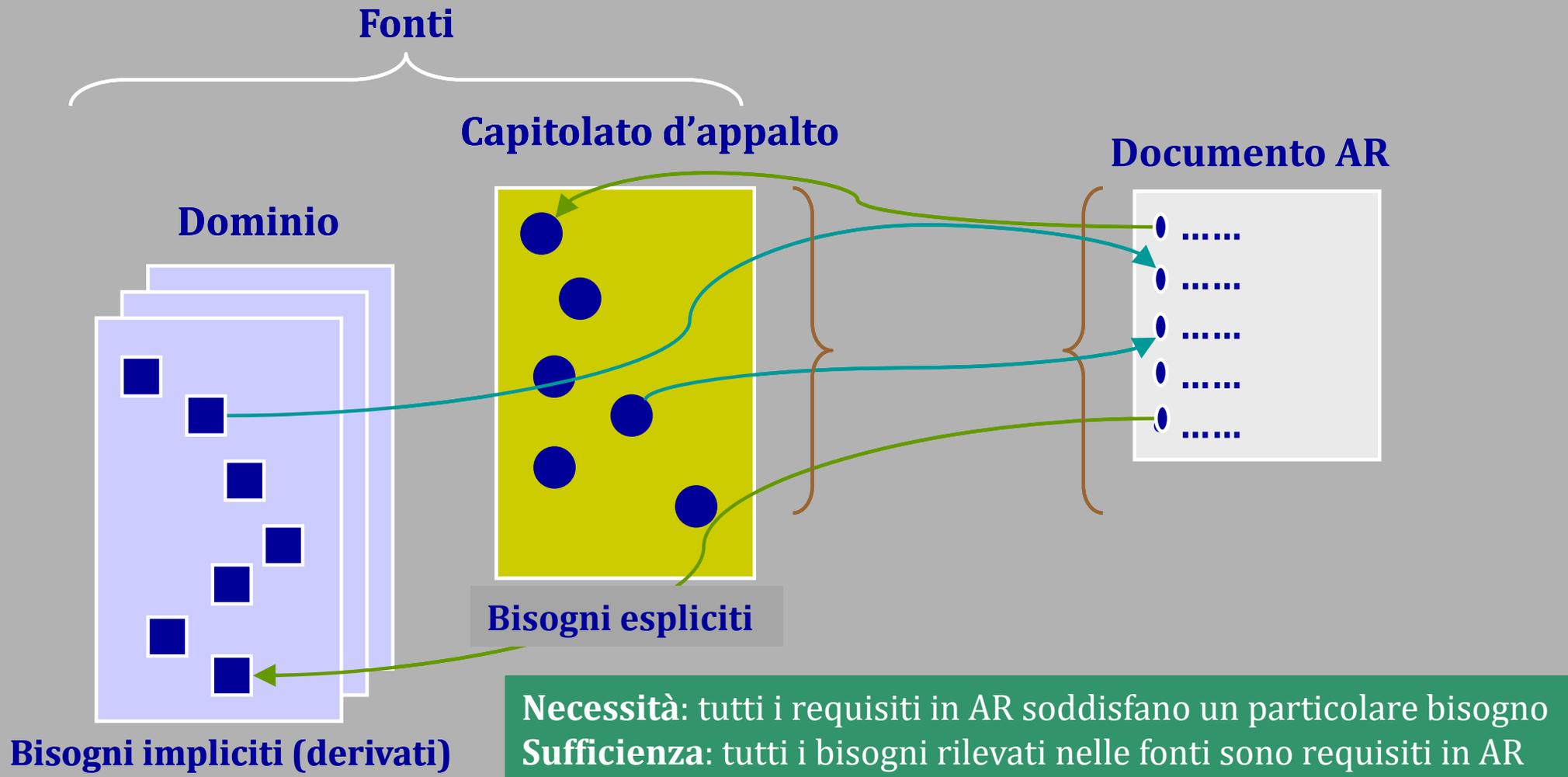


Tracciamento dei requisiti



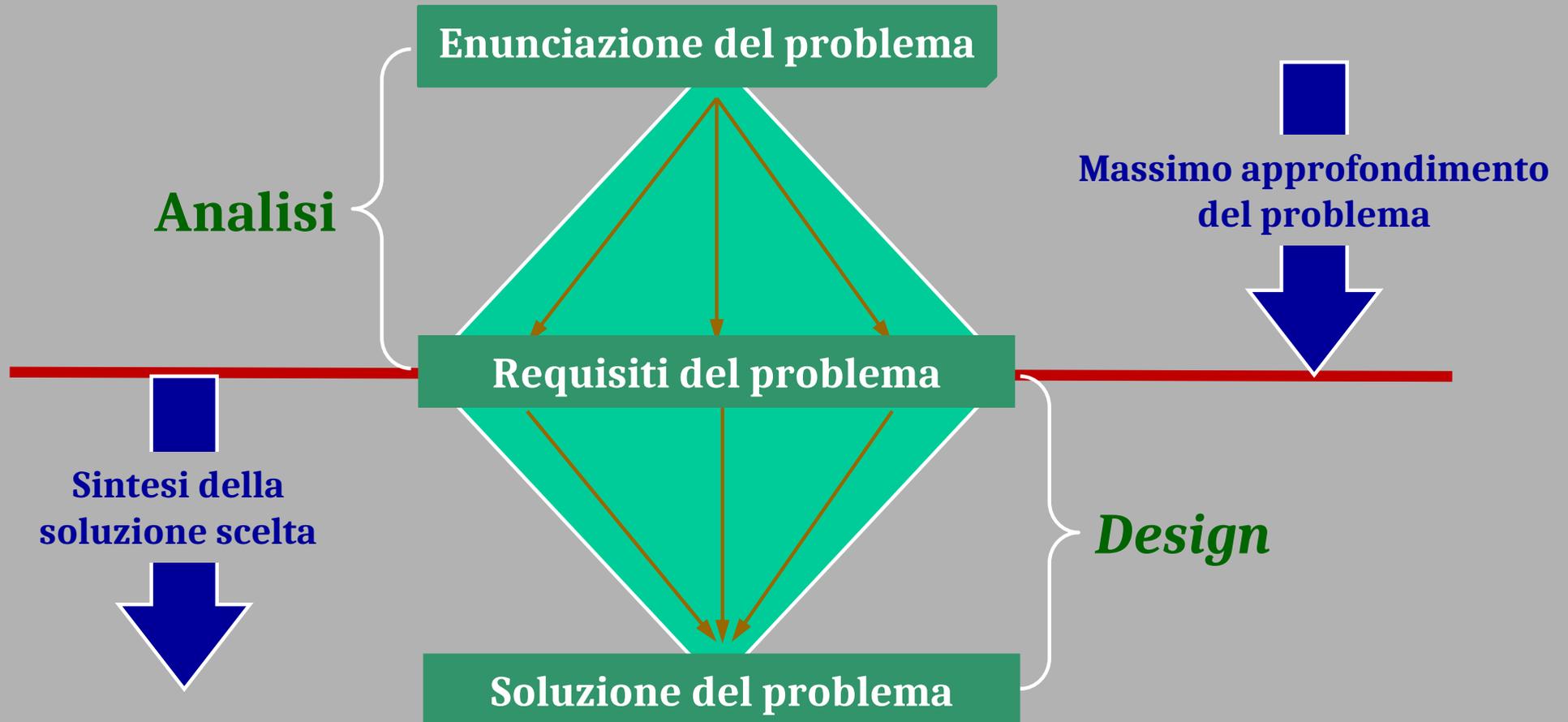


Esempio di tracciamento



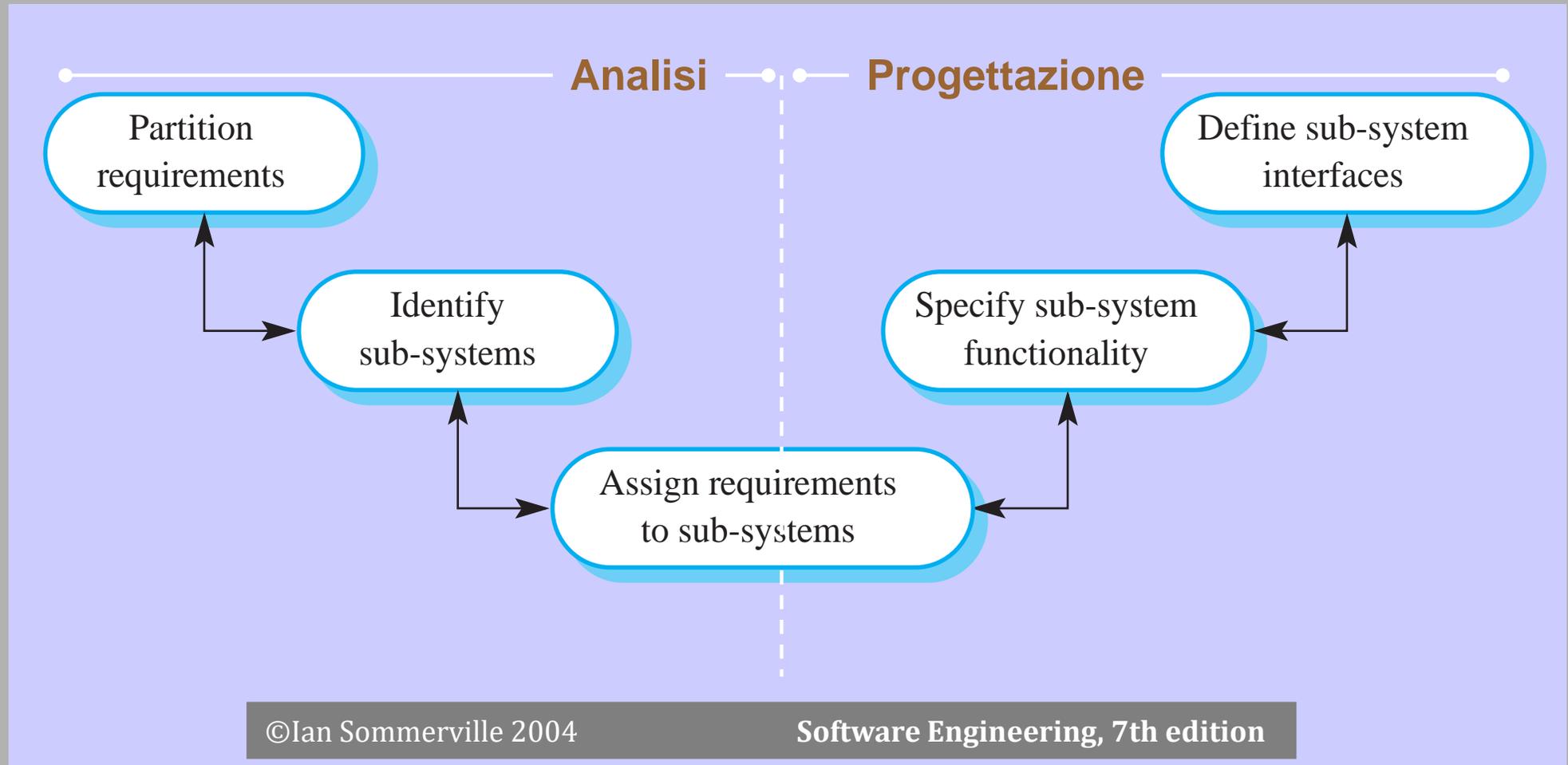


Confine tra analisi e *design* – 1





Confine tra analisi e *design* – 2





Processi di supporto all'analisi

□ Documentazione

- Per raccogliere i risultati dello studio di fattibilità
- Per specificare i requisiti

□ Verifica

□ Gestione e manutenzione dei prodotti

○ **Tracciamento dei requisiti**

- Essenziale per il controllo sistematico di conformità

○ **Gestione di versione e configurazione**

- I prodotti dell'analisi non sono monoliti e possono subire variazioni

○ **Gestione dei cambiamenti**

- Cambiare i requisiti ha implicazioni delicate e va fatto con metodo e procedure



Documentazione dell'analisi – 1

- La documentazione in linguaggio naturale genera rischi di ambiguità interpretativa
 - Servono norme redazionali per evitare espressioni ambigue
 - Glossario per garantire terminologia consistente
- L'uso di metodi (semi-)formali aiuta a ridurre gli errori di interpretazione
 - Diagrammi e formule (**UML**) invece di testo e disegni in stile libero



Documentazione dell'analisi – 2

1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Product overview
 - 1.3.1 Product perspective
 - 1.3.2 Product functions
 - 1.3.3 User characteristics
 - 1.3.4 Limitations
- 1.4 Definitions

2. References

3. Requirements

- 3.1 Functions
- 3.2 Performance requirements
- 3.3 Usability requirements
- 3.4 Interface requirements
- 3.5 Logical database requirements
- 3.6 Design constraints
- 3.7 Software system attributes
- 3.8 Supporting information

4. Verification

(parallel to subsections in Section 3)

5. Appendices

- 5.1 Assumptions and dependencies
- 5.2 Acronyms and abbreviations

ISO/IEC/IEEE 29148:2018
System and software engineering –
Life cycle processes –
Requirements engineering

Example outline of
Software Requirements Specification



Documentazione dell'analisi – 3

- **Ricerca chiarezza espressiva**
 - L'uso del linguaggio naturale rende difficile coniugare chiarezza con facilità di lettura

- **Ricerca chiarezza strutturale**
 - Separazione tra requisiti funzionali e non-funzionali
 - Classificazione precisa, uniforme e accurata

- **Ricerca atomicità e aggregazione**
 - Requisiti elementari
 - Correlazioni chiare ed esplicite



Verifica dell'analisi

- ❑ Eseguita su un documento organizzato
 - Non ripete il lavoro di analisi, ma si accerta che esso sia stato svolto in modo conforme alle attese
- ❑ Tramite *walkthrough*
 - Lettura a largo spettro
- ❑ Oppure *ispezione*
 - Lettura mirata e strutturata
- ❑ Accertando necessità e sufficienza dei requisiti specificati
 - Esaminando la matrice delle dipendenze (documentazione di tracciamento)



Stati di progresso per SEMAT – 1

□ *Conceived*

- Il committente è identificato e gli *stakeholder* vedono sufficienti opportunità per il progetto

□ *Bounded*

- I bisogni macro sono chiari, i meccanismi di gestione dei requisiti (configurazione e cambiamento) sono fissati

□ *Coherent*

- I requisiti sono classificati e quelli essenziali (obbligatori) sono chiari e ben definiti



Stati di progresso per SEMAT – 2

❑ *Acceptable*

- I requisiti fissati definiscono un sistema soddisfacente per gli *stakeholder*

❑ *Addressed*

- Il prodotto soddisfa i principali requisiti al punto da poter meritare rilascio e uso

❑ *Fulfilled*

- Il prodotto soddisfa abbastanza requisiti da meritare la piena approvazione degli *stakeholder*