



Analisi dei requisiti

IS

Anno accademico 2019/2020

Ingegneria del Software

Tullio Vardanega, tullio.vardanega@unipd.it



□ “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



V & V = Qualifica

□ **Verifica**

- **Accertare che l'esecuzione di specifiche attività non abbia introdotto errori**
 - *Did I build the system right?*
- **Attenzione rivolta ai processi (al *way of working*)**
 - Svolta sui prodotti dei processi per accertare il rispetto delle regole, convenzioni e procedure vigenti

□ **Validazione**

- **Accertare che il prodotto corrisponda alle attese**
 - *Did I build the right system?*
- **Attenzione rivolta ai prodotti finali**



Attività necessarie

- ❑ **Svolgimento dell'analisi**
 - Studio dei bisogni e delle fonti del dominio applicativo
 - Prima classificazione dei requisiti
 - Prerequisito al **tracciamento**
 - **Modellazione concettuale del sistema**
 - Dal punto di vista dei bisogni (*visione Use Case*)
 - **Assegnazione dei requisiti a parti distinte del sistema**
 - Dal punto di vista dei bisogni (*visione Use Case*)
 - **Negoziazione con il committente**
 - Consolidamento della **classificazione** dei requisiti
- ❑ **Redazione del Piano di Qualifica**
 - Definizione delle strategie di V&V
 - Metodi, tecniche, procedure, strumenti e tempi



Attività di analisi – 1

□ Studiare e definire il problema da risolvere

○ Identificare il prodotto da commissionare

- Compito del cliente (**committente**)

○ Capire cosa deve essere realizzato

- Compito del cliente e del fornitore

○ Definire gli accordi contrattuali

- Compito del cliente e del fornitore

Attività implicate
dai processi di
acquisizione e
fornitura

□ Verificare le implicazioni di costo e di qualità

○ La soddisfazione del cliente è relativa ai requisiti

- Espliciti o impliciti
- Diretti o derivati



Attività di analisi (lato fornitore) – 2

- ❑ **Studio dei bisogni e delle fonti**
 - **Identificare, specificare e classificare i requisiti**
- ❑ **Modellazione concettuale del sistema**
 - **Partizionamento in componenti (ambiti) a scopo di assegnazione dei requisiti**
 - Con diagrammi dei casi d'uso e solo sotto-casi
 - Non è progettazione!
 - **Catturare il punto di vista dell'attore sul sistema**
 - Cosa può essere richiesto – non cosa serve fare



Processi di supporto implicati

❑ Documentazione

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

❑ Gestione e manutenzione dei prodotti

○ Tracciamento dei requisiti

- Essenziale per il controllo sistematico di conformità

○ Impostazione e gestione della configurazione

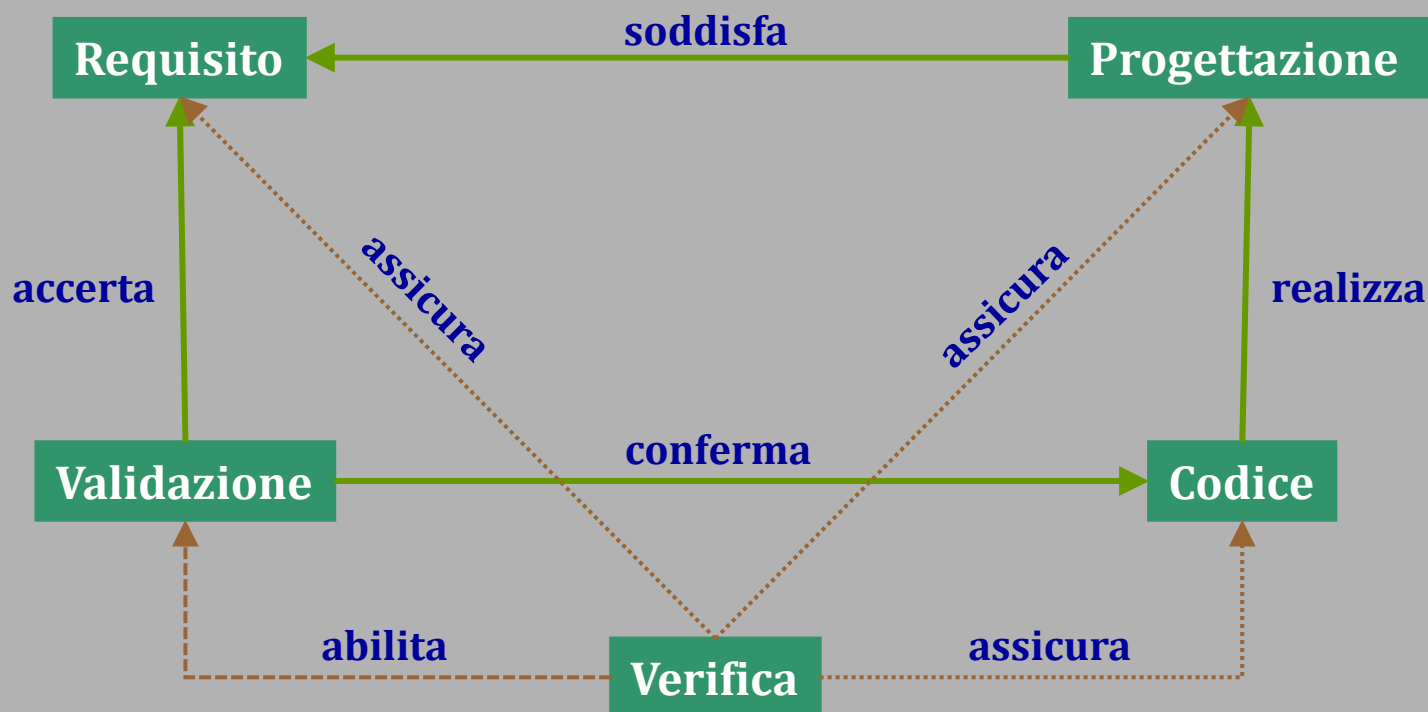
- La prima *baseline* riguarda i requisiti

○ Gestione dei cambiamenti

- Ha bisogno di regole, procedure, e strumenti di versionamento



Tracciamento dei requisiti



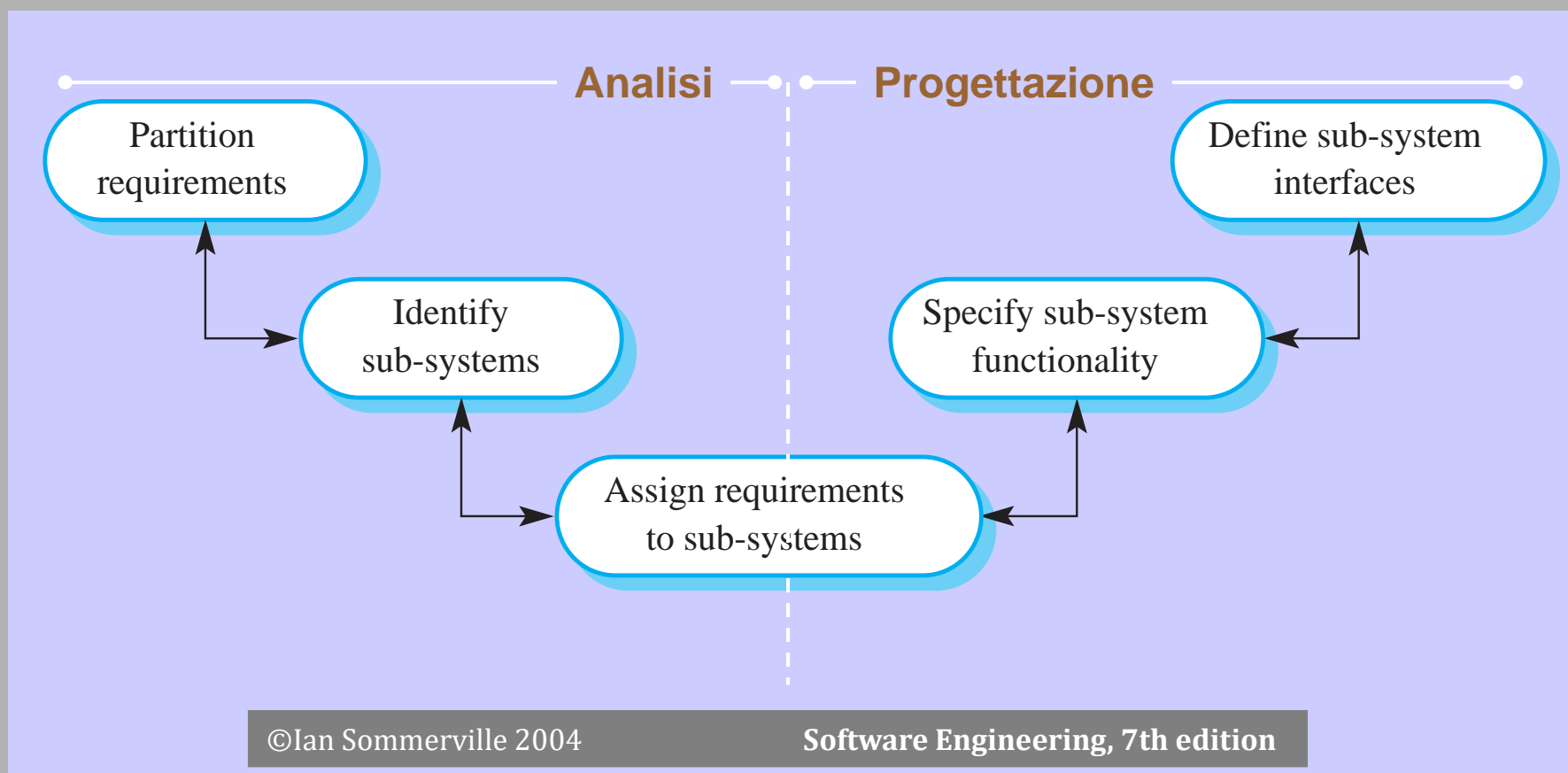


Prodotti documentali

- ❑ **Definizione dei bisogni (prima di utente e poi SW)**
 - **Capitolato d'appalto** → responsabilità del cliente
 - I requisiti utente sono vincoli contrattuali e specificano il cosa
 - I requisiti SW specificano il come
- ❑ **Specifica dei requisiti SW**
 - **Studio di Fattibilità** → documento interno del fornitore
 - **Analisi dei Requisiti** → documento contrattuale
- ❑ **La ripartizione dei requisiti svolta con i casi d'uso è la prima modellazione concettuale del sistema SW**
 - Da essa inizia la progettazione



Confine tra analisi e progettazione





Diversi approcci

- ❑ Procedimento *top-down*
 - Studio il sistema immaginando le parti in cui può essere decomposto
 - Senza elementi preconcepi: esplorazione funzionale
- ❑ Procedimento *bottom-up*
 - Concepisco il sistema ipotizzando le parti che possono comporlo
 - Tipico dell'OOP, fortemente orientato a riuso e specializzazione
- ❑ Procedimento *agile*
 - Perseguendo consolidamento incrementale
 - Nella cattura dei requisiti e nella realizzazione del prodotto



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 architetture
 - 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



Tecniche di analisi

□ Analisi dei bisogni e delle fonti

○ Comprensione del dominio

- Osservazione dei comportamenti dell'utente finale e dell'ambiente d'uso
- Fonte di requisiti impliciti

○ Interazione con il cliente

- Interviste
- Generazione, analisi e discussione di scenari

Esito documentato in minute
e con valore contrattuale

○ Discussioni creative e collaborative

- *Brainstorming*

○ Prototipazione

- Interna (solo per il fornitore)
- Esterna (per discussione con il cliente)





Comprensione 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

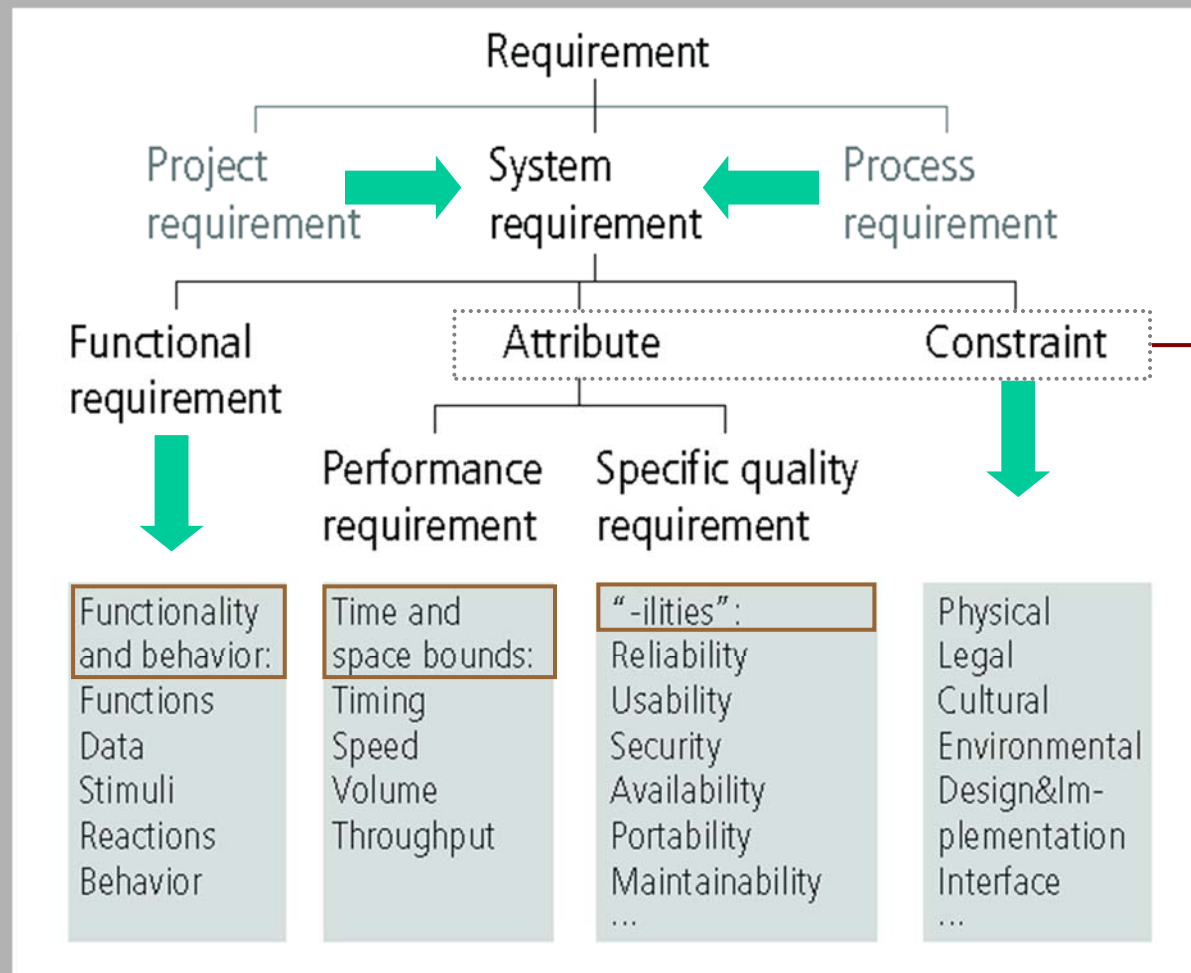


Classificazione dei requisiti – 1

- ❑ Mettere ordine nei requisiti facilita comprensione, manutenzione e tracciamento
- ❑ **Attributi di prodotto**
 - Domanda: **cosa devo fare?**
 - Requisiti funzionali, prestazionali, di qualità
- ❑ **Attributi di processo**
 - Domanda: **come devo farlo?**
 - Requisiti di vincolo (realizzativo, normativo, contrattuale)



Classificazione dei requisiti – 2



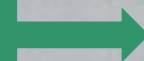
Extra-functional requirement

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

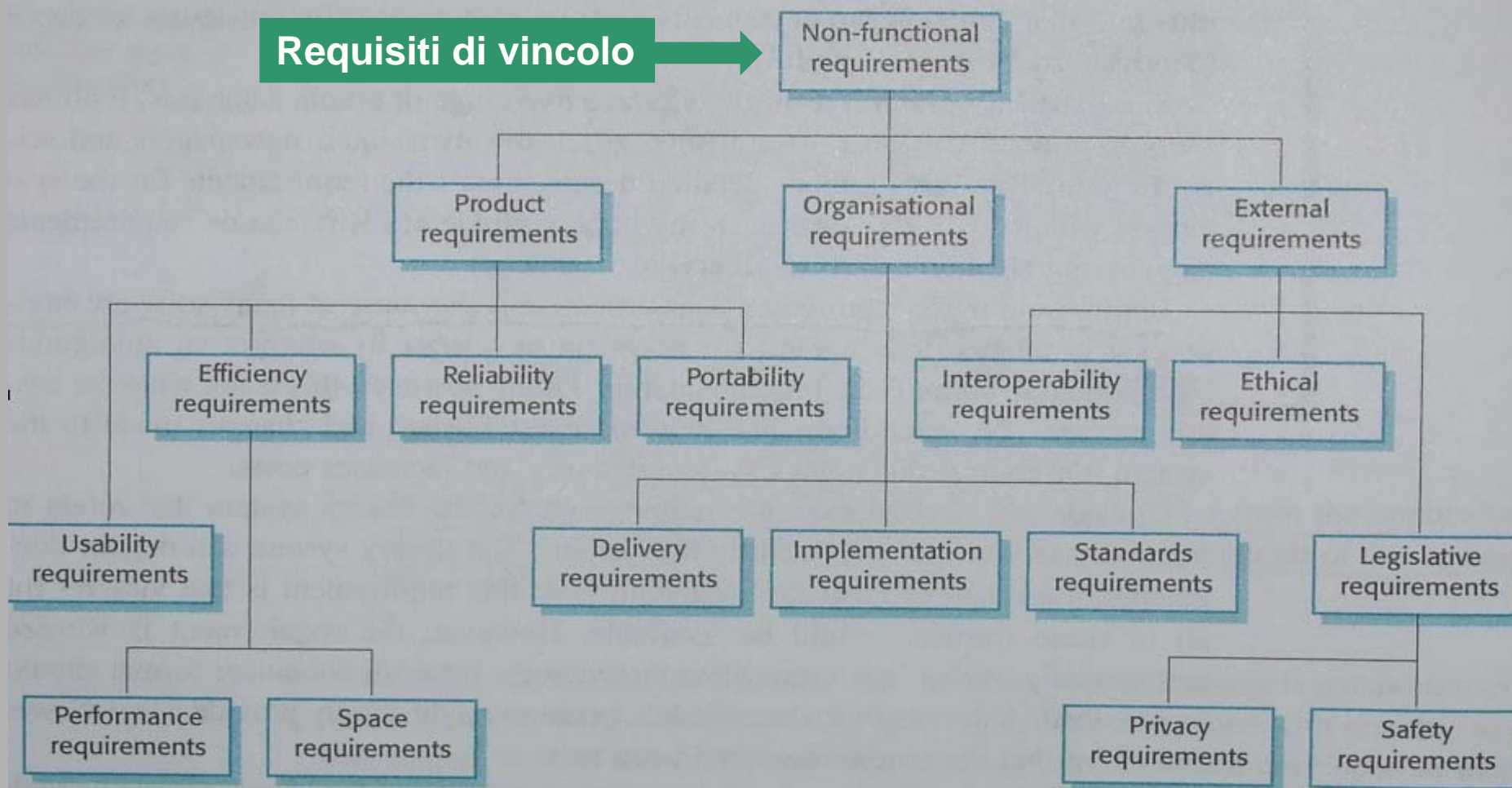


Classificazione dei requisiti – 3

Requisiti di vincolo



Non-functional requirements





Classificazione dei requisiti – 4

- ❑ I requisiti devono essere verificabili
- ❑ Chi fissa un requisito deve immaginare come accertarne il soddisfacimento
 - Requisiti funzionali → *test*, dimostrazione, revisione
 - Requisiti prestazionali → misurazione
 - Requisiti qualitativi → tecniche ad hoc
 - Requisiti dichiarativi (vincoli) → revisione
- ❑ Attenzione al costo e complessità di verifica



Classificazione dei requisiti – 5

❑ I requisiti hanno diversa utilità strategica

○ **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 contraddittori**

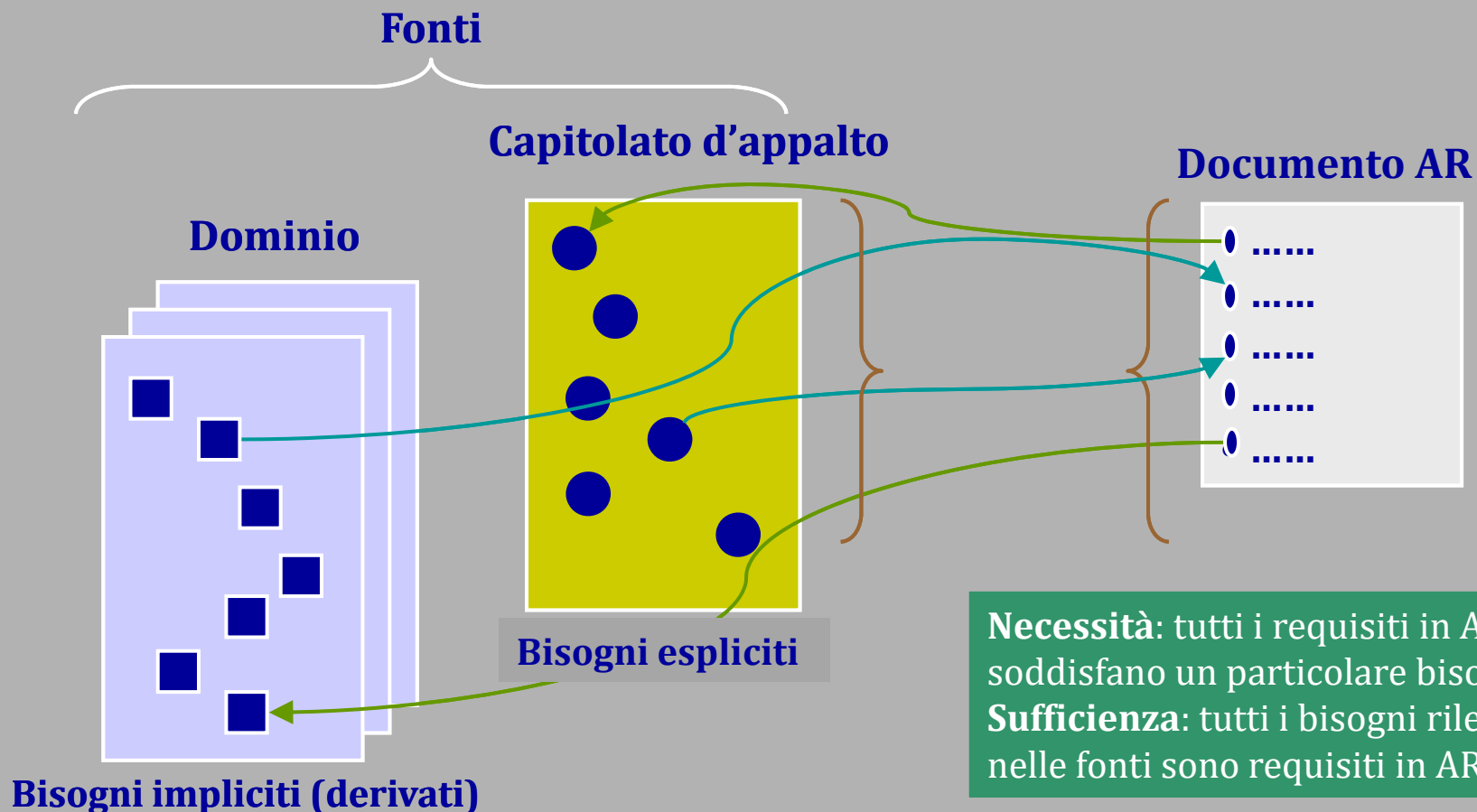


Attività di analisi – 3

- ❑ Accertare la soddisfacibilità dei requisiti rispetto ai vincoli di processo
- ❑ Effettuare **tracciamento** per assicurare che i requisiti concordati siano tutti e soli quelli necessari e sufficienti
 - Nessun bisogno trascurato (chiusura: quelli sufficienti)
 - Nessuna caratteristica superflua (sinteticità: quelli necessari)
- ❑ Determinare con il cliente l'utilità strategica dei requisiti concordati



Tracciamento dei requisiti





Attività di analisi – 4

- **La documentazione in linguaggio naturale genera rischi di ambiguità interpretativa**
 - 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 invece di testo e disegni in stile libero



Attività di analisi – 5

- ❑ **IEEE 830-1998: *Recommended Practice for Software Requirements Specifications***
- ❑ **La specifica 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: Struttura documento AR

1. Introduzione

- Scopo del documento
- Scopo del prodotto
- Glossario (definizioni, acronimi, abbreviazioni)
- Riferimenti (normativi, informativi)
- Struttura del documento

2. Descrizione generale

- Prospettive sul prodotto (per chi è inteso, quale contesto d'uso)
- Funzioni del prodotto
- Caratteristiche degli utenti
- Vincoli generali
- Assunzioni e dipendenze

3. Specifica dei requisiti

- Definizione dei requisiti utente
- Prima decomposizione del sistema
- Definizione dei requisiti di sistema
- Evoluzione attesa del sistema

Requisiti espressi in linguaggi

- semi-formali (grafici o algebrici)
 - operazionali (diagrammi di flusso)
 - dichiarativi (diagrammi E/R)
 - misti (UML)
- formali
 - operazionali (automi, algebre)
 - dichiarativi (logiche)
 - misti (macchine astratte)

Norma confluita in: ISO/IEC/IEEE 29148:2011



Verifica dei requisiti – 1

- ❑ Eseguita su un documento organizzato
- ❑ Tramite *walkthrough*
 - Lettura a largo spettro
- ❑ Oppure **ispezione**
 - Lettura mirata e strutturata
- ❑ Matrice delle dipendenze (necessità e sufficienza)
 - A fini di tracciamento



Verifica dei requisiti – 2

❑ **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



Gestione dei requisiti

❑ Identificazione, classificazione

- **Identificatore unico**
 - Garantito da DBMS
- **Numerazione sequenziale basata sulla struttura del documento**
 - Esempio: 2.4.7
- **Coppie <CATEGORIA, NUMERO>**

❑ Gestione dei cambiamenti

- Valutazione di fattibilità tecnica ed impatto sul progetto

❑ Tracciabilità

- Requisiti ↔ parti della specifica ↔ componenti del sistema
- Strumenti di supporto informatico



Implicazioni di costo e di qualità

- **Cause di abbandono (Standish Group 1995)**
 - **Requisiti incompleti**
 - **Insufficiente coinvolgimento del cliente/utente**
 - Cliente e utente non sono necessariamente la stessa entità
 - **Scarsità di risorse**
 - **Attese irrealistiche**
 - **Volatilità di specifiche e requisiti**
 - **Insufficiente competenza tecnologica/metodologica del fornitore**



Wisdom of the sages – 1

- Studiosi del *requirements engineering* hanno individuato alcune debolezze nelle pratiche convenzionali e hanno proposto corrispondenti contromisure
 - Pamela Zave & Michael Jackson
Four Dark Corners of Requirements Engineering
ACM Transactions on Software Engineering and Methodology, 6(1):1-30, January 1997
- Vediamole ...



Wisdom of the sages – 2

- ❑ **La terminologia dell'analisi dei requisiti deve riflettere il dominio (l'ambiente) d'uso del prodotto SW**
 - Quindi non la prospettiva interna del prodotto
- ❑ **La trasformazione dei requisiti in specifiche implementabili passa dal consolidamento della conoscenza del dominio d'uso**
 - Capire bene cosa deve accadere nell'ambiente d'uso e come ciò va percepito
- ❑ **I requisiti non devono descrivere il prodotto, ma l'ambiente d'uso, visto senza e con il prodotto in uso**
 - Capire prima il «senza», per meglio comprendere il «con»
- ❑ **La descrizione delle azioni specificate nell'analisi deve individuarne gli attori distinguendo se essi siano nell'ambiente o nel prodotto**
 - Rivedremo questa metodica nei diagrammi dei casi d'uso



Wisdom of the sages – 3

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 as 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?



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*