Lezione 10 19/11/2024

Processo di definizione di un’architettura software

La definizione dell’architettura software è un’insieme di attività che permettono ad un insieme di stakeholder di andare a definire la soluzione architetturale di un sistema.

Questo deve essere un processo evolutivo, perché il software stesso evolve nel tempo.

L'attività principale è la progettazione architetturale. Il processo include anche altre attività fondamentali, come:

- Analisi
- Valutazione
- Documentazione
- Facoltativamente, implementazione iniziale

Queste attività sono **evolutive** ed **iterative**, perché parto da un disegno di partenza e faccio analisi di ciò che va inserito, raffinando sempre di più finché quello che ho prodotto ha un livello di dettaglio sufficiente.

- Evolutiva: L'architettura si evolve nel tempo man mano che i requisiti cambiano o emergono nuovi vincoli
- Iterativa: Il processo prevede cicli di progettazione, valutazione e perfezionamento per migliorare progressivamente l'architettura e garantire che rimanga allineata agli obiettivi del sistema

Questo approccio assicura flessibilità e adattività.

Sintesi:

1. **Architectural Design**:
    1. The **central activity** of the SSA process
    2. Involves creating a high-level blueprint of the system, detailing components, their interactions, and how they collectively achieve the desired functionality
    3. This step ensures a clear definition of system structures, enabling effective communication among stakeholders
2. **Analysis**:
    1. Focuses on understanding and refining system requirements and constraints
    2. Examines architectural options to identify trade-offs in terms of performance, scalability, reliability, and cost
    3. Involves techniques like use case analysis, scenario modeling, and quality attribute prioritization
3. **Evaluation**:
    1. Verifies that the proposed architecture meets the system's functional and non-functional requirements
    2. Methods may include simulation, prototyping, and architectural reviews
    3. Ensures risks are identified and mitigated early in the design process
4. **Documentation**:
    1. Provides a **comprehensive description** of the architecture, including diagrams, specifications, and rationale for decisions made
    2. Serves as a reference for developers, testers, and stakeholders throughout the system's lifecycle
5. **Initial Implementation (Optional)**:
    1. In some cases, the process includes developing a **proof of concept** or prototype
    2. Helps validate the feasibility of the design and identify potential issues before full-scale implementation

I moduli che abbiamo identificato nell'architettura logica, andremo a dire qual è l'interfaccia di un componente, e come interagisce con il resto del sistema.

Discuteremo il modello di Hofmeister, che è un modello generalizzato che raccoglie a fattor comune le caratteristiche dei principali modelli industriali.

è essenziale rendere bene chiari quali sono gli obiettivi complessivi, insieme agli input e gli output attesi, del processo di definizione di architettura.

La definizione dell'architettura ha come obiettivo quello di raccogliere i bisogni e gli interessi degli stakeholder, e poi progettare una soluzione architetturale che soddisfa le necessità.

Quindi ho l'obiettivo, devo capire quali sono gli interessi degli stakeholders, come input ho gli architectural concerns che possono essere gli obiettivi ma anche altre informazioni, e poi in output ho un disegno architetturale che è stato validato secondo gli ASR.

Hofmeister model

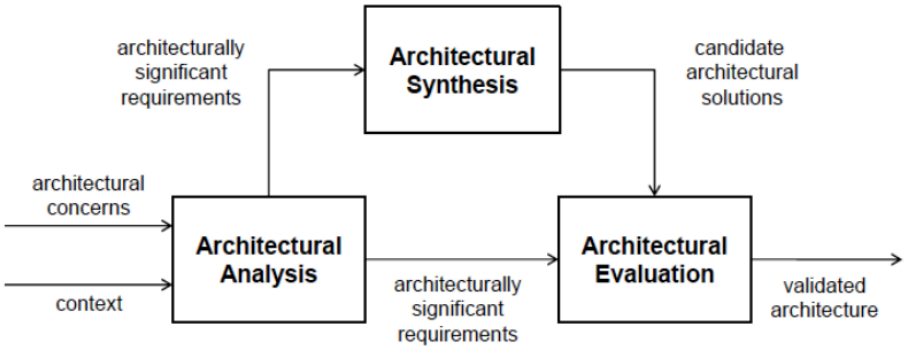
Hofmeister ha sviluppato un modello generale per il processo di definizione e progettazione dell'architettura software.

Il modello cattura e generalizza le caratteristiche chiave comuni di cinque approcci industriali prominenti alla progettazione dell'architettura software.

Il modello identifica 3 attività principali, che vanno eseguite in maniera iterativa:

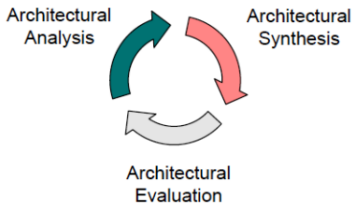
- **Analisi architetturale**
- **Sintesi architetturale (design)**
- **Valutazione architetturale**

- **Architectural Analysis**
  - Aims to **define, identify, clarify, and/or negotiate** the problem (architecturally significant requirements, ASR) that the architecture must address
- **Architectural Synthesis (Design)**
  - The core activity of **architecture design**, which focuses on creating the architecture based on the identified ASRs
- **Architecture Evaluation**
  - Ensures that the architectural design decisions are **appropriate** and effectively meet the identified ASRs



Il contesto è molto utile, perché per esempio il paese nel quale deve essere sviluppato il software può dare informazioni molto utili.

Le tre principali attività del modello di Hofmeister vengono svolte in modo **iterativo** (evolutivo). Questo approccio consente un continuo affinamento e adattamento dell'architettura man mano che si approfondisce la comprensione dei requisiti e emergono nuove intuizioni durante il processo.



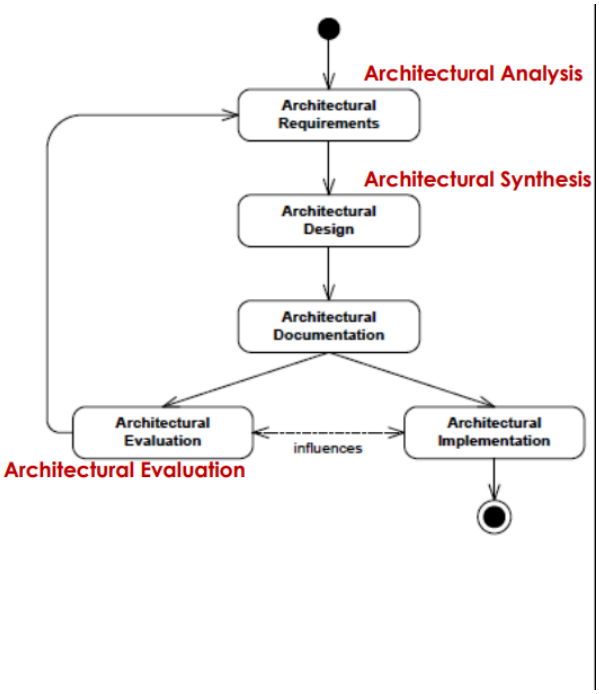
L'**analisi** prevede che gli interessi, gli ASR, siano esaminati, rivisitati e consolidati. Quando io termino questa fase, voglio che gli ASR siano prioritizzati. Devo poter dire quanto è importante ciascun requisito, e devo dire quanto è difficile poter soddisfare quel requisito.

Gli ASR vanno in input al processo di design, dove vengono scelti gli elementi dell'architettura che risultano più idonei. Vengono considerate delle opzioni (design options), devo fare una valutazione per capire se tra queste design options che sono idonee per risolvere il problema, per trovare quella migliore (per esempio perché se la scelgo, so che tutti gli altri ASR si sposano bene).

Nell'attività di valutazione, l'architettura è validata per vedere se soddisfa gli ASR, se necessario si itera. Il processo di definizione dell'architettura termina quando l'architettura è considerata sufficientemente buona.

Roadmap di quello che vedremo:

- **Architecture Requirements**
  - **Docs**: Review of relevant documentation to identify architectural needs.
  - **Interviews (QAW - Quality Attribute Workshops)**: Engaging with stakeholders to uncover and prioritize quality attribute requirements
  - **Understanding Goals (PALM)**: Using the PALM (Pedigreed Attribute eLicitation Method) approach to understand the system's goals and related attributes
- **Architectural Design**
  - **Attribute-Driven Design (ADD)**: A methodology that drives the design process based on the system's quality attribute requirements
- **Architectural Evaluation**
  - **Architecture Tradeoff Analysis Method (ATAM)**: A structured approach for evaluating architectural decisions by analyzing trade-offs in quality attributes.
- **Documenting an Architecture**
  - Proper documentation of the architecture to ensure clarity and traceability of decisions made during the design and evaluation processes



Analysis (or Architectural Requirements)

Vogliamo recuperare i requisiti significativi.

Per un architetto, non tutti i requisiti sono uguali. Ci sono dei requisiti particolarmente importanti, chiamati **ASR (architecturally significant requirement)**, e sono quelli che impattano in maniera significativa sulla soluzione architetturale da scegliere.

Possiamo avere 2 tipi di ASR:

- **Requisiti esplicitamente significativi**, come quelli di natura tecnica come la performance, l'interfacciamento con altri sistemi, il numero di utenti che deve servire contemporaneamente...
- **Requisiti implicitamente significativi**, possono definire la natura del comportamento funzionale del sistema, cioè qual è il comportamento

funzionale che deve avere un requisito, come il “come fare un acquisto online”, la procedura.

Come si decide che un requisito è significativo dal punto di vista architetturale? Devo capire se è **critico per uno stakeholder**, oppure se richiede una soluzione architetturale che da un punto di vista potrebbe essere **molto d’impatto sul resto**.

Come li possiamo identificare? Ho 3 modi:

- Leggere i **documenti** che specificano i requisiti che il sistema deve realizzare
- Coinvolgo gli **stakeholders** (interviste) per raccogliere informazioni e chiarire dubbi
- Andare a capire quali sono i **business goals**

Documenti

Un luogo ovvio in cui cercare potenziali ASR (Architecturally Significant Requirements) è nei documenti dei requisiti o nelle user stories... purtroppo questo non è solitamente il caso. Perché?

Perché molti progetti non hanno neanche una documentazione.

La maggior parte di ciò che è contenuto in una specifica dei requisiti non influisce sull'architettura, di solito si parla di funzionalità.

Inoltre, nessun architetto si limita ad aspettare che i requisiti siano "completati" prima di iniziare il lavoro. L'architetto deve iniziare mentre i requisiti sono ancora in fase di evoluzione.

Anche se i documenti esistono, di solito non sono così utili all’architetto, perché quando gli attributi di qualità sono catturati, non sono scritti bene, sono molto generali. E anche perché la maggior parte di quello di cui ha bisogno l’architetto non è presente neanche nei documenti fatti meglio.

Gli ASR spesso derivano dagli obiettivi aziendali dell'organizzazione di sviluppo stessa. Le qualità di sviluppo (come le ipotesi relative al lavoro in team, in base alle loro competenze) sono anch'esse non menzionate di solito.

Se un requisito influisce sulla presa di una decisione di progettazione architetturale, per definizione è un **ASR** (Architecturally Significant Requirement).

Bisogna cercare nel documento requisiti che possano influire su una decisione di progettazione architetturale.

Some specific things to look for are the following **categories** of information:

- Usage
  - User roles, internationalization, language distinctions
- Time
  - Timeliness and element coordination
- External elements
  - External systems, protocols, sensors or actuators (devices), middleware
- Networking
  - Network properties and configurations (including their security properties)
- Orchestration
  - Processing steps, information flows

- Security properties
  - User roles, permissions, authentication
- Data
  - Persistence and currency
- Resources
  - Time, concurrency, memory footprint, scheduling, multiple users, multiple activities, devices, energy usage, soft resources (e.g., buffers, queues), scalability requirements
- Project management
  - Plans for teaming, skill sets, training, team coordination
- Hardware choices
  - Processors, families of processors, evolution of processors.
- Flexibility of functionality, portability, calibrations, configurations
- Named technologies, commercial packages

Interviste

Quando non ho documenti comprensivi che vengono prodotti, oppure non saranno pronti in tempo, allora posso intervistare gli stakeholders.

Gli stakeholder spesso non hanno idea di quali siano i requisiti di qualità che sono importanti per il sistema.

Se insisti su requisiti quantitativi per i QA, è probabile che tu ottenga numeri arbitrari. Almeno alcuni di questi requisiti saranno molto difficili da soddisfare.

Gli architetti invece conoscono bene gli attributi di qualità che devono raggiungere, derivanti dall'esperienza passata. Quindi vanno coinvolti, e le interviste devono puntare a capire cosa gli stakeholders **sanno** e cosa **necessitano**.

I risultati delle interviste con gli stakeholder dovrebbero includere:

- Un elenco di **driver architetturali**, cioè il insieme di requisiti che hanno un'influenza significativa sulla tua architettura.
- Un set di scenari **QA** (Quality Attributes) prioritizzati.

Definizioni

Architectural Drivers: i driver architetturali sono i fattori di alto livello che influenzano significativamente l'architettura del sistema. Esempi:

- Attributi di qualità come prestazioni, scalabilità o sicurezza.
- Vincoli aziendali come budget o tempi.
- Requisiti esterni, come la conformità a standard.

Scenari di Attributi di Qualità (QA) Prioritizzati: gli scenari QA sono descrizioni specifiche e concrete di situazioni o sfide che riflettono il comportamento richiesto del sistema in relazione agli attributi di qualità (requisiti non funzionali). Esempi:

- **Prestazioni:** "Il sistema deve elaborare 1000 transazioni al secondo durante le ore di punta con una latenza massima di 1 secondo."
- **Sicurezza:** "Solo gli utenti autenticati devono accedere ai record confidenziali entro 2 secondi dalla richiesta."

Quality attribute workshop

Questo è un metodo focalizzato sugli stakeholders che permette di **generare**, **prioritizzare** e **perfezionare** gli scenari di attributi di qualità **prima** che l'architettura software sia completata.

È un metodo di intervista strutturato progettato per raccogliere e prioritizzare i requisiti di attributi di qualità dagli stakeholder. Facilita l'identificazione precoce dei requisiti architetturalmente significativi (ASR) coinvolgendo gli stakeholder in un processo collaborativo.

Steps:

- **Step 1: Presentazione e Introduzioni al QAW**

  I facilitatori del QAW descrivono la motivazione del QAW e spiegano ogni passaggio del metodo.

- **Step 2: Presentazione del Business/Missione**

  Lo stakeholder che rappresenta le preoccupazioni aziendali dietro il sistema presenta il contesto aziendale del sistema, i requisiti funzionali generali, i vincoli e i requisiti noti relativi agli attributi di qualità.

- **Step 3: Presentazione del Piano Architetturale**

  L'architetto presenta i piani architetturali del sistema così come sono al momento.

- **Step 4: Identificazione dei Driver Architetturali**

  I facilitatori del QAW condivideranno l'elenco dei principali driver architetturali che hanno raccolto durante i passaggi 2 e 3, chiedendo agli stakeholder chiarimenti, aggiunte, eliminazioni e correzioni.

  L'obiettivo è raggiungere un consenso su un elenco sintetico di driver architetturali che includa requisiti, vincoli aziendali, vincoli tecnologici e attributi di qualità.

- **Step 5: Brainstorming sugli Scenari**

  Ogni stakeholder esprime uno scenario che rappresenta le sue preoccupazioni rispetto al sistema.

  I facilitatori del QAW garantiscono che esista almeno uno scenario rappresentativo per ogni driver architetturale elencato nel passaggio 4.

- **Step 6: Consolidamento degli Scenari**

  Gli scenari simili vengono consolidati quando possibile.

- **Step 7: Prioritizzazione degli Scenari**

  La prioritizzazione degli scenari avviene assegnando a ciascuno stakeholder un numero di voti, generalmente pari al 30% del numero totale degli scenari.

- **Step 8: Raffinamento degli Scenari**

  I facilitatori del QAW aiutano gli stakeholder a mettere gli scenari prioritari nel formato a sei parti: **fonte**, **stimolo**, **artefatto**, **ambiente**, **risposta** e **misura della risposta**.

Understanding business goals

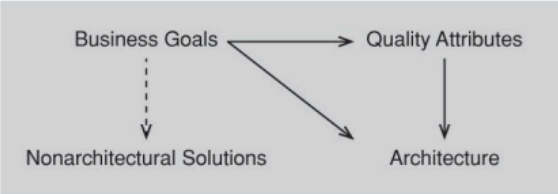
I business goals sono la ragione di esistere di un sistema.

Gli obiettivi aziendali comuni includono, naturalmente, il conseguimento di un profitto, ma la maggior parte delle organizzazioni ha molte altre preoccupazioni oltre al semplice profitto. Gli obiettivi aziendali spesso generano requisiti il cui raggiungimento (o mancato raggiungimento) segnala un progetto architettonico di successo (o meno).

Gli obiettivi aziendali conducono frequentemente direttamente agli ASR.

- There are three possible relationships between business goals and an architecture

1. *Business goals often **generate** quality attribute requirements*
   - Every quality attribute requirement should originate from some higher purpose that can be described in terms of *added value*
   - *E.g., response time vs competitors*
2. *Business goals may directly **affect** the architecture without inducing a quality attribute requirement at all*
   - For example, database team to be involved even if not necessary
3. *No influence at all*
   - For example, a business goal to “reduce cost” may be realized by lowering the thermostats in the winter or reducing employees’ salaries



Come si fanno a capire i business goals?

Gli architetti spesso acquisiscono consapevolezza del business e degli obiettivi aziendali di un'organizzazione per "osmosi". Questo avviene lavorando, ascoltando, parlando e assimilando gradualmente gli obiettivi in atto nell'organizzazione.

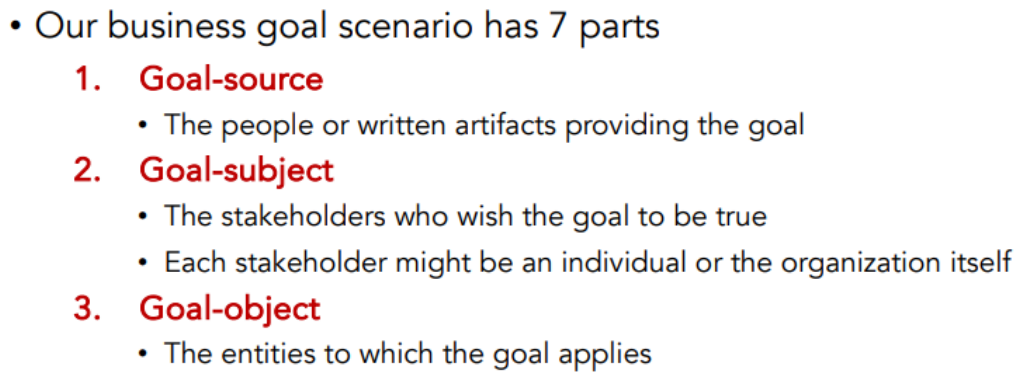
Sebbene l'osmosi abbia i suoi benefici, esistono modi più sistematici... vedremo il **PALM** (**Pedigreed Attribute eLicitation Method**).

PALM

Obiettivi Chiave del PALM (Pedigreed Attribute eLicitation Method):

- **Identificare** gli attributi di qualità più importanti per il sistema (es. prestazioni, scalabilità, sicurezza).
- **Stabilire una chiara origine** per ciascun attributo, documentando la fonte e la motivazione della sua inclusione.
- **Prioritizzare** gli attributi di qualità in base alle necessità degli stakeholder e all'impatto sull'architettura.
- Fornire **tracciabilità e giustificazione** per le decisioni relative agli attributi di qualità.

Come possiamo scrivere un business goal dopo che l'abbiamo capito? Uno scenario è un modo pratico, uniforme e chiarificatore per esprimere gli obiettivi aziendali.



4. **Environment**
   - The context for this goal
     - Environment may be social, legal, competitive, customer, and technological
5. **Goal**
   - Any business goal expressed by the goal-source
6. **Goal-measure**
   - A testable measurement to determine how one would know if the goal has been achieved
   - The goal-measure should usually include a *time* component, stating the time by which the goal should be achieved
7. **Pedigree and value**
   - The degree of ***confidence*** the person who stated the goal has in it
   - The goal's ***volatility*** and ***value***

Esempio:

Goal-Source: The company's board of directors expressed the goal in their annual report, emphasizing the need to improve environmental sustainability.

Goal-Subject: The goal is of particular interest to the company's shareholders and investors, who are looking for environmentally responsible practices.

Goal-Object: The goal applies to all company manufacturing facilities, targeting the reduction of carbon emissions.

Environment: The goal is set within the context of an industry facing increased scrutiny for its environmental impact and a growing demand for eco-friendly products.

Goal: The goal is to reduce carbon emissions by 20% across all manufacturing facilities within the next two years.

Goal-Measure: The goal will be measured by tracking and reporting the reduction in carbon emissions using a standardized carbon footprint calculation.

Pedigree and Value: The goal's pedigree traces back to the company's commitment to sustainability, driven by the CEO's vision. Achieving this goal holds significant value for the company's reputation, cost savings through energy efficiency, and its ability to attract environmentally conscious investors.

Altro esempio:

Gli elementi possono essere combinati in una frase del tipo:

For the system being developed
<goal-subject> desires that <goal-object> achieve <goal>
in the context of <environment> and will be satisfied if <goal-measure&value>

- Some sample business goal scenarios include the following:
  - For MySys, the ***project manager*** has the goal that ***his family's stock*** in the ***company will rise by 5 percent*** (as a result of the success of MySys)
  - For MySys, the **portfolio manager** has the goal that MySys ***will make*** the ***portfolio 30 percent more profitable***
  - For MySys, the ***project manager*** has the goal that ***customer satisfaction will rise*** by ***10 percent*** (as a result of the increased quality of MySys)

- A general scenario is a template for constructing concrete scenarios
- A general scenario uses the generic structure of a scenario to supply a list of possible values for each variable part of a scenario

Goal-subject	. . . has the goal that	Goal-object	. . . achieves	Goal	. . . in the context of	Environment	. . . and will be satisfied if	Goal-measure	Value
Any stakeholder or stakeholder group identified as having a *legitimate* interest in the system		Individual		Contributing to the growth and continuity of the organization		Social (includes political)		Time that business remains viable	1–*n*
		System		Meeting financial objectives		Legal		Financial performance vs. objectives	1–10
		Portfolio		Meeting personal objectives		Competitive		Promotion or raise achieved in period	H-M-L
		Organization's employees		Meeting responsibility to employees		Customer		Employee satisfaction; turnover rate	Resources willing to expend
		Organization's shareholders		Meeting responsibility to society		Technological		Contribution to trade deficit/ surplus	
		Organization		Meeting responsibility to state				Stock price, dividends	
		Nation		Meeting responsibility to shareholders				Market share	
		Society		Managing market position				Time to carry out a business process	
				Improving business processes				Quality measures of products	
				Managing quality and reputation of products				Technology-related problems	
				Managing change in environmental factors				Time window for achievement	

Un metodo per raccogliere e documentare gli obiettivi aziendali è il metodo **Pedigreed Attribute eLicitation Method** o PALM.

PALM utilizza il formato dello scenario degli obiettivi aziendali. PALM è un metodo composto da sette fasi. Vi partecipano architetti e stakeholder che possono discutere degli obiettivi aziendali pertinenti.

- **Business goals elicitation**
  - Capture from stakeholders the set of important business goals for the system
  - Elaborate the business goals and express them as business goal scenarios
  - Consolidate almost-alike business goals to eliminate duplication
  - Have the participants prioritize the resulting set to identify the most important goals

- **Identify potential QAs from business goals**
  - For each important business goal scenario, have the participants describe a QA and response measure value that (if architected into the system) would help achieve the goal

Steps:

- **Step 1: PALM overview presentation**
  - Overview of PALM, the problem it solves, its steps, and its expected outcomes
- **Step 2: Business drivers' presentation**
  - Briefing by the project management on the business drivers
    - What are the goals of the customer organization for this system?
    - What are the goals of the development organization?
- **Step 3: Architecture drivers' presentation**
  - Briefing by the architect on the driving *business* and *quality attribute requirements*
- **Step 4: Business goals elicitation**
  - Business goals are elaborated and expressed as *scenarios*
  - Consolidate almost-alike business goals to eliminate duplication
  - Participants prioritize the resulting set to identify the most important goals

- **Step 5: Identification of potential quality attributes from business goals**
  - For each important business goal scenario, participants describe a *quality attribute* that (if architected into the system) would help achieve it
- **Step 6: Assignment of pedigree**
  - For each architectural driver (step 3), identify which business goals it support
  - We establish its pedigree by asking for the source of the quantitative part (e.g., why is there a 40 ms performance requirements? Why not 60?)
- **Step 7: Exercise conclusion**
  - Review of results, next steps, and participant feedback

Capturing ASRs in a Utility Tree

Infine, voglio rappresentare tutti gli ASR che ho identificato in un modo che sia facilmente fruibile. Di solito si usa l'utility tree, che elenca tutti i requisiti non funzionali, a cui appiccica uno scenario, di modo che si possa capire quale requisito è associato a quale attributo di qualità.

Un **Utility Tree** è una struttura che consente di registrare tutti gli **ASR** in un unico posto e di stabilire la priorità di ogni **ASR** in base a:

- L'impatto sull'architettura.
- Il valore aziendale o la missione.

La radice dell'albero è un nodo segnaposto chiamato "Utility".

Il secondo livello contiene categorie generali di **QA** (Qualità degli Attributi), ad esempio, "performance".

Il terzo livello affina tali categorie, ad esempio, "tempo di attesa dell'utente" e "throughput delle transazioni".

Il quarto livello cattura gli **ASR** (sotto forma di scenari). Sotto ogni affinamento, vengono registrati gli **ASR** appropriati.

Procedure

1. Build the tree

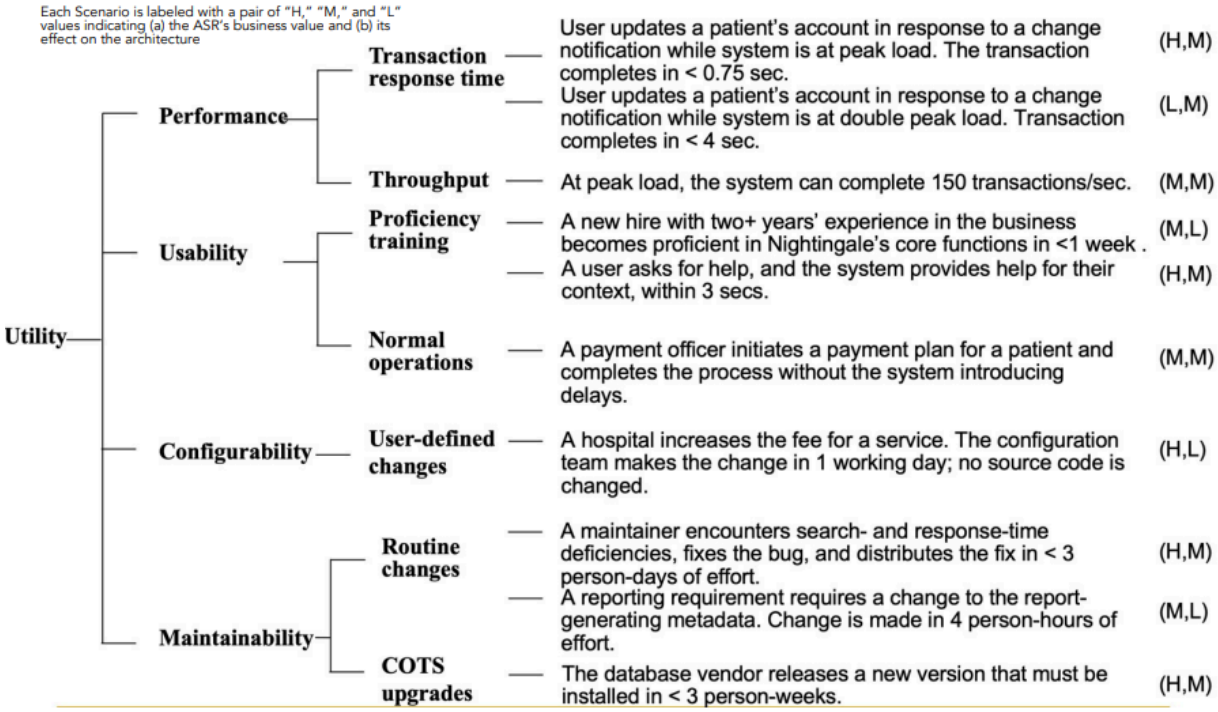
Build the tree by placing ASRs in the tree

2. Valuate Scenarios

Valuate each ASR against the two criteria: the **business value** of the candidate ASR and the **architectural impact** of including it

- You can use any scale you like, e.g.,"**H**" (high), "**M**" (medium), "**L**" (low)
- For business
  - **High**: a **must-have** requirement
  - **Medium**: an **important** requirement but would not lead to project failure if omitted
  - **Low**: a **nice** requirement to have but not something worth much effort
- For architectural impact
  - **High** means that meeting this ASR will **profoundly** affect the architecture
  - **Medium** means that meeting this ASR will **somewhat** affect the architecture
  - **Low** means that meeting this ASR will have **little** effect on the architecture

Excerpt from Utility Tree for a Health Care Application Called Nightingale



Utilizza l'**Utility Tree** per effettuare controlli importanti, ad esempio:

- Un **QA** o un affinamento del **QA** senza alcun **ASR** non è necessariamente un errore o un'omissione. Occorre prestare attenzione a cercare eventuali **ASR** non registrati in quell'area.
- Gli **ASR** classificati con un punteggio (H,H) (High Impact, High Priority) sono quelli che meritano maggiore attenzione. Un numero molto elevato di **ASR** con punteggio (H,H) potrebbe essere motivo di preoccupazione: il sistema è realizzabile?
- Gli stakeholder possono esaminare l'**Utility Tree** per assicurarsi che le loro preoccupazioni siano affrontate.

I requisiti, che siano stati raccolti o meno, cambiano continuamente.

Gli architetti devono adattarsi e aggiornarsi per garantire che le loro architetture siano ancora quelle giuste per il successo del progetto.

Mantieni sempre un canale aperto con i principali stakeholder che determinano gli **ASR**, così da rimanere al passo con i requisiti in evoluzione.

Sommario

- Architectures are driven by architecturally significant requirements: requirements that will have profound effects on the architecture
  - Architecturally significant requirements may be captured from requirements documents, by interviewing stakeholders, or by conducting a Quality Attribute Workshop
- Be mindful of the business goals of the organization
  - Business goals can be expressed in a common, structured form and represented as scenarios
  - Business goals may be elicited and documented using a structured facilitation method called PALM
- A useful representation of quality attribute requirements is in a utility tree
  - The utility tree helps to capture these requirements in a structured form
  - Scenarios are prioritized
  - This prioritized set defines your “marching orders” as an architect