Luca Cabibbo

Architettura dei Sistemi Software

Qualità del software e progettazione per le qualità

dispensa asw140

marzo 2021

It is quality rather than quantity that matters.

Seneca

- Riferimenti

- Luca Cabibbo. **Architettura del Software**: Strutture e Qualità. Edizioni Efesto, 2021.
  - Capitolo 4, Qualità del software e progettazione per le qualità
- [POSA1] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M. **Pattern-Oriented Software Architecture**: A System of Pattern, Volume 1 (**POSA1**), Wiley, 1996.
- [DSA] Cervantes, H. and Kazman, R. **Designing Software Architectures**: A Practical Approach. Addison Wesley, 2016.

- Obiettivi e argomenti

- Obiettivi
  - introdurre il concetto di qualità
  - presentare i principali approcci architetturali per la progettazione per le qualità
- Argomenti
  - introduzione
  - qualità
  - progettare per le qualità
  - discussione

* Introduzione

- Abbiamo più volte detto che l'architettura di un sistema software ha un ruolo fondamentale nel raggiungimento degli attributi di qualità del sistema
  - che cosa si intende per qualità?
  - in che modo l'architettura sostiene le qualità di un sistema? quali sono gli approcci di progettazione per controllare gli attributi di qualità?

* Qualità

- Due tipologie principali di requisiti/interessi del software
  - requisiti/interessi funzionali
  - requisiti/interessi non-funzionali
    - chiamati anche **attributi di qualità**, **proprietà di qualità** o, semplicemente, **qualità**
  - i requisiti architetturalmente significativi sono soprattutto attributi di qualità – da considerare nel contesto di un insieme di requisiti funzionali principali

Premessa: qualità e uso previsto

- Da Wikipedia (2020)
  - La **qualità**, nell'ambito dell'economia, dell'ingegneria e della produzione, indica una misura delle caratteristiche o delle proprietà di una entità (una persona, un prodotto, un processo, un progetto) in confronto a quanto ci si attende da tale entità, per un determinato impiego.
  - Nell'accezione più usata del termine, ci si riferisce alla qualità di un bene, materiale o immateriale, che viene prodotto per un determinato utilizzo.
  - L'uso che si intende fare del bene in oggetto è importante, poiché la valutazione della qualità varia a seconda dell'utilizzo. Per esempio, una persona può essere un ottimo scrittore, ma avere una valutazione molto bassa come atleta. Allo stesso modo, un gruppo di dati può avere un'alta qualità quando usati come informazione generica, divulgativa, ma una bassa qualità per un utilizzo di alta precisione.
  - Il concetto di qualità è applicabile […] ogni volta che un oggetto, una persona o altro, viene confrontato con le performance attese (*ovvero, quello che ci si attende da esso*).

Alcune qualità del software

- Prestazioni (performance)
  - la capacità del sistema di eseguire in modo prevedibile entro il profilo di prestazioni richiesto
- Sicurezza (security)
  - la capacità del sistema di resistere ad usi non autorizzati – e nel tempo stesso fornire servizi ai suoi utenti legittimi
- Modificabilità (modifiability)
  - la capacità del sistema di essere flessibile a fronte di cambiamenti inevitabili dopo il rilascio, in modo bilanciato rispetto ai costi di fornire tale flessibilità

Alcune qualità del software

- Affidabilità (reliability)
  - in generale, il livello con cui l'utente può fare affidamento sulle funzionalità del sistema per svolgere i propri compiti
  - in modo più specifico, la capacità di un sistema di eseguire le funzionalità specificate per un periodo di tempo specificato, senza fallimenti
  - spesso specificata in termini di altre qualità, come la disponibilità, la tolleranza ai guasti e la capacità di recupero

Alcune qualità del software

- Disponibilità (availability)
  - la capacità di un sistema di essere completamente o parzialmente funzionante come e quando richiesto
- Tolleranza ai guasti/resilienza (fault tolerance/resilience)
  - la capacità del sistema di operare come richiesto, anche a fronte di guasti di componenti hardware o software del sistema
- Capacità di recupero (recoverability)
  - riguarda la capacità di recupero del sistema dai fallimenti – per rendere il sistema nuovamente operativo e recuperare i suoi dati dopo un fallimento, entro tempi predefiniti e accettabili

Alcune qualità del software

- Scalabilità (scalability)
  - la capacità del sistema di rispondere a variazioni nel suo carico di lavoro, senza ripercussioni su prestazioni e disponibilità
- Usabilità (usability)
  - riguarda la facilità per l'utente nell'utilizzo del sistema per eseguire un compito desiderato – nonché il supporto all'utente fornito dal sistema
- Interoperabilità (interoperability)
  - riguarda il grado in cui due o più sistemi possono interagire in modo utile, scambiando tra loro informazioni significative, tramite interfacce, in un contesto particolare – nonché la capacità di interpretare correttamente i dati scambiati

Alcune qualità del software

- Verificabilità (testability)
  - la facilità con cui è possibile dimostrare e identificare errori del software mediante dei test
- Monitorabilità/osservabilità (monitorability/observability)
  - riguarda la capacità di osservare un sistema software mentre è in esecuzione nell'ambiente di produzione
- Supporto per rilasci e aggiornamenti (deliverability)
  - il rilascio di nuove versioni del software agli utenti finali è importante e rischioso – e deve talvolta poter essere effettuato in modo rapido, frequente e affidabile

Alcune qualità del software

- Tempo di realizzazione (time to market)
  - il tempo di realizzazione (sviluppo e rilascio) di un nuovo sistema o di un nuovo servizio
  - una qualità importante soprattutto in presenza di pressioni competitive o di piccole finestre temporali circa l'opportunità di un sistema o servizio
- Costo (cost)
  - anche i costi di realizzazione del software sono importanti
  - sono possibili diversi indicatori di costo (a secondo delle voci di costo che comprendono o meno) – ad es., il TCO (Total Cost of Ownership)
- Semplicità (simplicity)
  - i sistemi progettati in modo semplice sono più semplici da implementare, verificare e far evolvere

Architettura e qualità

- La definizione dell'architettura software è la prima fase nel ciclo di vita di un sistema software in cui vengono presi in considerazione gli attributi di qualità
  - l'architettura ha un ruolo critico nel raggiungimento degli attributi di qualità desiderati
  - nell'architettura di un sistema, le strutture software determinano il sostegno alle qualità
  - le qualità desiderate vanno tenute in considerazione durante tutta la definizione dell'architettura

Qualità e compromessi

- In genere, gli attributi di qualità non possono essere raggiunti in isolamento
  - ottenere una qualità può avere un effetto negativo (o positivo) su un'altra
  - la progettazione dell'architettura deve considerare le conseguenze di ciascuna decisione di progetto e i possibili compromessi

Architettura e qualità

- L'architettura costituisce le fondamenta per le qualità del sistema
  - tuttavia, il raggiungimento delle qualità va tenuto in considerazione anche durante la progettazione e l'implementazione

Qualità e viste

- Relazione tra qualità e viste architetturali
  - in alcuni casi, la gestione di una qualità può avvenire localmente a una singola vista
  - altre qualità richiedono considerazioni nell'ambito di più viste – i cosiddetti interessi trasversali (*crosscutting concerns*)
  - sono comunque necessari degli approcci per gestire e controllare le diverse qualità in modo efficace e coerente

* Progettare per le qualità

- La disciplina dell'architettura del software ha l'obiettivo di distillare la conoscenza acquisita da esperienze passate nel raggiungimento degli attributi di qualità del sistema
  - la codifica sistematica delle relazioni tra strutture e qualità è di fondamentale importanza nel processo di progettazione dell'architettura
- Ecco alcune soluzioni proposte in letteratura per la progettazione per gli attributi di qualità
  - pattern architetturali
  - tattiche architetturali
  - prospettive architetturali
  - design concept
  - tutti questi approcci sono basati sul riuso di conoscenza relativa alla "soluzione esemplare di problemi significativi e ricorrenti"

- Pattern architetturali

- Un **pattern architetturale** [POSA1] (o **stile architetturale** [SSA])
  - descrive un particolare *problema* di progettazione ricorrente che si manifesta in uno specifico contesto di progettazione
  - e presenta un ben provato schema generico per la sua *soluzione* – che è specificato dalla descrizione dei componenti che lo costituiscono, le loro responsabilità e relazioni, e il modo in cui essi collaborano
- Esempi – *Layers*, *Pipes-and-Filters*, *Microkernel*, *Client/Server*, *Peer-to-Peer*, ...
- I pattern architetturali sostengono la progettazione iniziale di un sistema software con alcune proprietà di qualità ben definite

- Tattiche architetturali

- Una **tattica** (**architetturale**) [SAP] è una decisione di progetto che influenza il controllo (e dunque, il conseguimento) della risposta di un attributo di qualità
  - ciascuna tattica è un'opzione di progettazione, volta a controllare una specifica qualità
- Esempi di tattiche per migliorare le prestazioni – sono *trasformazioni* per ridurre il tempo di risposta a un evento
  - *increase resource efficiency*, *increase resources*, *reduce overhead*, *introduce concurrency*, *maintain multiple copies of computations*, …
- Le tattiche architetturali sostengono il raffinamento di un progetto – in cui il sostegno per alcune qualità è insoddisfacente

- Prospettive architetturali

- Una **prospettiva architetturale** (**prospettiva**) [SSA] è
  - una collezione di attività, tattiche e linee guida usate per garantire che un sistema esibisca una particolare proprietà di qualità che richiede riflessioni attraverso diverse viste architetturali del sistema
  - ciascuna prospettiva si riferisce a una specifica qualità o a uno specifico insieme di qualità correlate tra di loro
- Le prospettive sono un modo per codificare e sistematizzare le conoscenze circa la progettazione per le proprietà di qualità
  - utili per condividere e riusare le conoscenze dei bravi architetti – e dunque anche per l'apprendimento
  - costituiscono una *guida efficace* alla risoluzione di fattori architetturali

- Design concept

- Un **design concept** [DSA] è un qualunque "blocco" costruttivo con cui è possibile creare o raffinare le strutture di un'architettura
  - alcuni design concept sono più popolari nelle università
    - i *pattern architetturali* e le *tattiche architetturali*
  - altri design concept sono invece più popolari tra i professionisti
    - le *architetture di riferimento* (ad es., le applicazioni web e le applicazioni a servizi), i *pattern di rilascio* (ad es., non distribuito, distribuito, client/server a 2/3/4 livelli) – nonché i *componenti sviluppati*, tra cui le *famiglie di tecnologie* (ad es., DBMS), i *prodotti* (ad es., MySQL), i *framework* (ad es., Spring e Hibernate) e le *piattaforme* (ad es., Java EE e Google App Engine)

* Discussione

- Esistono diversi approcci architetturali per progettare per le qualità
  - i pattern architetturali sono utili soprattutto per effettuare una *decomposizione iniziale* (a grana grossa) del sistema
  - le tattiche architetturali sono utili per il *raffinamento* (a grana fine) dell'architettura
  - le prospettive architetturali sono più generali, perché hanno l'obiettivo/ambizione di descrivere il corpo della conoscenza sulla progettazione per una specifica qualità
  - i design concept generalizzano questi e altri possibili approcci di progettazione dell'architettura

Discussione

- Relazione tra gli approcci per le qualità e le viste
  - l'applicazione di un pattern architetturale porta all'identificazione/ definizione degli elementi (e delle relazioni tra elementi) in una o più viste architetturali – di solito in una sola vista
  - l'applicazione di una tattica architetturale porta al raffinamento degli elementi (e delle relazioni tra elementi) in una o più viste architetturali
  - l'applicazione di una prospettiva richiede in genere di intervenire su diverse viste architetturali

Discussione

- Vantaggi legati ad approcci architetturali basati sul "riuso" di buone esperienze di progettazione
  - guida alla progettazione, analisi, valutazione, comunicazione dell'architettura
  - adozione di soluzioni accettate
  - possibilità di lavorare in modo sistematico
  - possibilità di avere framework e/o generatori di codici specializzati per un certo pattern architetturale
- Rischi
  - possibilità di suggerimenti conflittuali
  - in alcuni casi i suggerimenti sono (troppo) generici

Discussione

- La progettazione per le qualità, nel seguito del corso
  - la Parte II presentano la progettazione per alcuni specifici attributi di qualità – soprattutto in termini di tattiche architetturali, anche se talvolta con un approccio da prospettiva architetturale
  - la Parte III (e in parte anche le Parti IV e V) presentano invece la progettazione dell'architettura soprattutto in termini di pattern architetturali

Concetti e relazioni fondamentali