

## Architetture Software

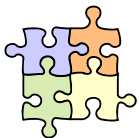
# Ottenere qualità: stili, tattiche e prospettive architeturali

Dispensa ASW 160

ottobre 2014

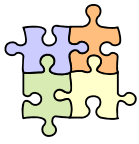
*Semplifica, combina ed elimina.*

*Suzaki*



### - Fonti

- [SAP] Chapter 4, Understanding Quality Attributes
- [POSA1] Chapter 1, Patterns
- [SSA] Chapter 4, Architectural Perspectives



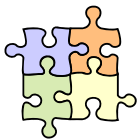
## - Obiettivi e argomenti

### ▣ Obiettivi

- ▣ introdurre il concetto di qualità
- ▣ introdurre i principali approcci architeturali per ottenere qualità

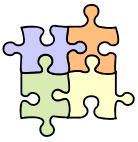
### ▣ Argomenti

- ▣ proprietà di qualità
- ▣ tattiche architeturali
- ▣ stili architeturali
- ▣ prospettive architeturali
- ▣ discussione



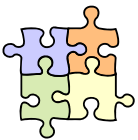
## - Wordle





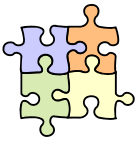
## Introduzione

- Abbiamo più volte detto che le architetture software hanno un ruolo fondamentale nel raggiungimento degli attributi di qualità
  - che cosa si intende per qualità? in che modo l'architettura software sostiene le qualità di un sistema?
  - quali sono gli approcci a disposizione del progettista per controllare gli attributi di qualità? in particolare, come è possibile controllare, in modo efficace e coerente, anche quelle qualità che corrispondono a interessi trasversali (*crosscutting concerns*) a più viste architeturali?



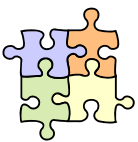
## \* Proprietà di qualità

- Requisiti/interessi del software
  - requisiti/interessi funzionali
  - requisiti/interessi non-funzionali
    - anche chiamati **proprietà di qualità** – oppure **attributi di qualità**
    - oppure, più semplicemente, **qualità**



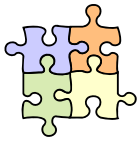
## Alcune qualità

- ❑ Prestazioni
  - la capacità del sistema di eseguire in modo prevedibile entro il profilo di prestazioni richiesto
- ❑ Scalabilità
  - la capacità del sistema di rispondere a variazioni nel suo carico di lavoro
- ❑ Modificabilità
  - 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
- ❑ Verificabilità
  - la facilità con cui è possibile dimostrare ed identificare errori del software mediante dei test



## Alcune qualità

- ❑ Sicurezza
  - la capacità del sistema di resistere ad usi non autorizzati – e nel tempo stesso fornire servizi ai suoi utenti legittimi
- ❑ Disponibilità
  - la capacità di un sistema di essere completamente o parzialmente funzionante come e quando richiesto, anche a fronte di guasti di componenti del sistema
- ❑ Resilienza
  - riguarda le modalità di recupero dai fallimenti di un sistema – la capacità di rendere nuovamente operativo un sistema, dopo un suo fallimento, entro tempi predefiniti ed accettabili



## Alcune qualità

### □ Interoperabilità

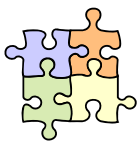
- riguarda il grado in cui due o più sistemi possono interagire in modo utile, scambiando tra loro informazioni significative, tramite interfacce, in un contesto particolare – questa qualità riguarda non solo la capacità di scambiare dati, ma anche la possibilità di interpretare correttamente i dati scambiati

### □ Usabilità

- riguarda la facilità per l'utente nell'utilizzo del sistema per eseguire un compito desiderato – nonché il supporto all'utente fornito dal sistema

### □ Time to market

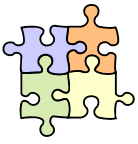
- il tempo di realizzazione di un sistema o di un nuovo servizio – una qualità importante soprattutto in presenza di pressioni competitive o di piccole finestre temporali circa l'opportunità di un sistema o servizio



## Alcune qualità

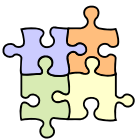
### □ Integrità concettuale

- I will contend that *conceptual integrity* is the most important consideration in system design – it is better to have a system omit certain anomalous features and improvements, but to reflect one set of design ideas, than to have one that contains many good but independent and uncoordinated ideas [Brooks, 1975]
- ad esempio
  - in Unix, la nozione di file è unificante
  - in Smalltalk, ogni cosa è un oggetto
  - in SQL, i dati sono sempre tabelle
  - in Lisp, ogni cosa è una lista



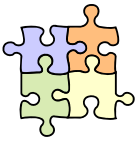
## La qualità è relativa all'uso previsto

- Da Wikipedia (ottobre 2013)
  - Questa voce riguarda la qualità nell'accezione più usata, essenzialmente nell'ambito dell'ingegneria, dell'economia e della produzione, quando ci si riferisce ad un bene, materiale o immateriale, che viene prodotto per un determinato utilizzo.
  - In generale, la misura della **qualità** 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.
  - L'uso che si intende fare 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 quello che ci si attende da lui.



## Architetture e qualità

- La definizione dell'architettura software è la prima fase nella creazione di un sistema software in cui vengono presi in considerazione gli attributi di qualità
  - le architetture sono critiche nel raggiungimento degli attributi di qualità desiderati dalle parti interessate
  - il modo in cui le funzionalità del sistema sono poste in corrispondenza con le strutture software determina il sostegno offerto dall'architettura alle qualità
  - pertanto, le qualità desiderate vanno tenute in considerazione durante tutta la progettazione dell'architettura



## Architetture e qualità

### □ Alcuni esempi

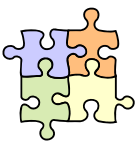
- per ottenere alte prestazioni, bisogna gestire le caratteristiche temporali del comportamento degli elementi, nonché frequenza e volume della comunicazione tra elementi
- se la modificabilità è importante, le responsabilità vanno assegnate agli elementi in modo tale che ciascun cambiamento atteso abbia effetto su uno o pochi elementi
- se il sistema deve essere altamente sicuro, la comunicazione tra gli elementi va gestita e protetta opportunamente – potrebbe essere richiesta l'introduzione di elementi specializzati
- se la scalabilità è importante, l'uso delle risorse deve essere opportunamente localizzato, in modo da poter facilitare l'introduzione di risorse con maggior capacità
- se si vuole riusabilità, l'accoppiamento tra elementi deve essere opportunamente ridotto

13



Ottenere qualità: stili, tattiche e prospettive architettrurali

Luca Cabibbo – ASw



## Architetture e qualità

### □ Alcuni esempi

- per ottenere alte prestazioni, bisogna gestire le caratteristiche temporali del comportamento degli elementi, nonché frequenza e volume della comunicazione tra elementi
- se la modificabilità è importante, le responsabilità vanno assegnate agli elementi in modo tale che ciascun cambiamento atteso abbia effetto su uno o pochi elementi
- se il sistema deve essere altamente sicuro, la comunicazione tra gli elementi va gestita e protetta opportunamente – potrebbe essere richiesta l'introduzione di elementi specializzati
- se la scalabilità è importante, l'uso delle risorse deve essere opportunamente localizzato, in modo da poter facilitare l'introduzione di risorse con maggior capacità
- se si vuole riusabilità, l'accoppiamento tra elementi deve essere opportunamente ridotto

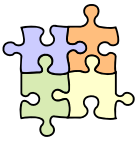
In questi esempi, si può notare che la gestione di queste qualità avviene soprattutto a livello architeturale. Infatti, in tutti questi esempi, si parla di elementi, proprietà esterne degli elementi, relazioni tra elementi – e delle caratteristiche che essi devono possedere per poter ottenere una certa qualità.

14



Ottenere qualità: stili, tattiche e prospettive architettrurali

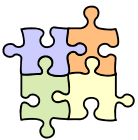
Luca Cabibbo – ASw



## Qualità e compromessi

### □ Attenzione

- 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 con attenzione tutti i possibili compromessi – e le conseguenze di ciascuna possibile decisione di progetto



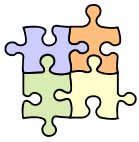
## Architetture e qualità

- L'architettura costituisce le fondamenta per le qualità del sistema
  - tuttavia, da sola, non ne garantisce il raggiungimento
  - infatti, il raggiungimento delle qualità va tenuto in considerazione sia durante la progettazione che durante l'implementazione

### □ Alcuni esempi

- modificabilità
  - architettura – scelta e organizzazione dei moduli
  - implementazione – stile di codifica
- usabilità
  - architettura – possibilità di annullare operazioni, fare undo e redo, riusare dati immessi in precedenza, ...
  - implementazione – tipo di carattere, tipo di componente grafico, disposizione dei componenti sullo schermo, ...





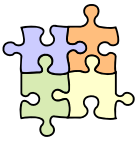
## Qualità e viste

- In alcuni casi, la gestione di una qualità individuale può avvenire localmente a una singola vista
  - alcuni interessi – ad esempio, la modificabilità – possono essere gestiti con riferimento (soprattutto) a una singola vista
- Tuttavia, altre qualità corrispondono a interessi trasversali (*crosscutting concerns*) alle varie viste
  - alcuni interessi – ad esempio, la scalabilità – richiedono considerazioni nell'ambito di più viste
  - come è possibile gestire anche queste qualità in modo efficace e coerente?



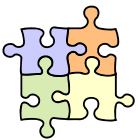
## Qualità e viste

- Si consideri ad esempio la qualità della *sicurezza* – che è un aspetto vitale di molti sistemi
  - ad es., restringere l'accesso a certi dati/funzionalità solo a classi appropriate di utenti
- Che relazione c'è tra la qualità della sicurezza e le viste dell'architettura?
  - la sicurezza può essere gestita nell'ambito di una singola vista – in particolare, tra quelle incontrate in precedenza?
  - altrimenti, potrebbe avere senso definire una vista della Sicurezza? (affinché abbia senso dovrebbe essere, per quanto possibile, indipendente dalle altre viste, altrimenti non ci sarebbe separazione degli interessi)



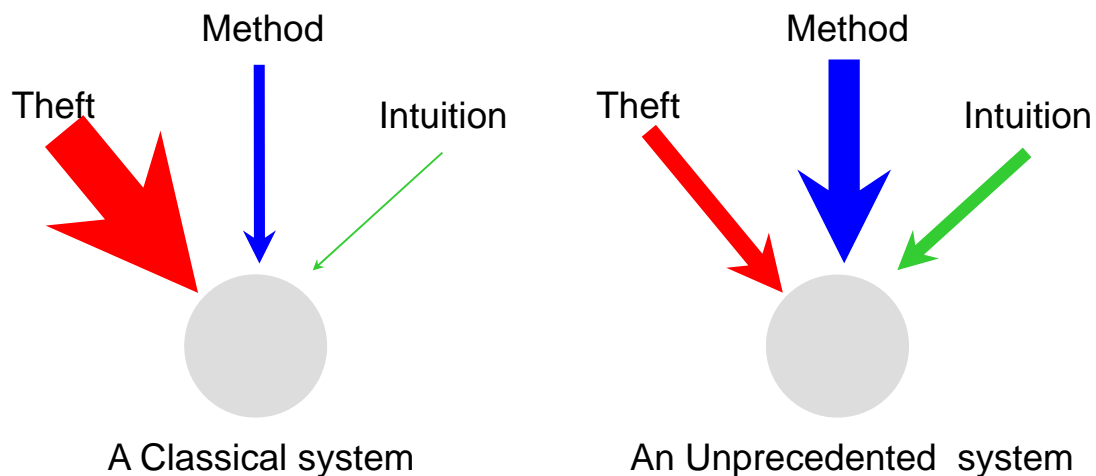
## Sicurezza

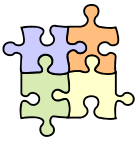
- La sicurezza è un interesse che riguarda diverse viste
  - il sistema deve avere la capacità di identificare e autenticare utenti – questo è di interesse per la vista funzionale
  - il sistema deve controllare diversi classi di accesso alle informazioni (CRUD) – applicando controlli a livelli variabili di granularità – vista delle informazioni
  - il sistema deve essere in grado di creare, gestire e distribuire informazioni segrete (chiavi e password) ed essere aggiornato con le ultime patch sulla sicurezza – vista operativa
  - ...
- La qualità “il sistema deve essere sicuro” viene decomposta attraverso i vari punti di vista in qualità più specifiche
  - in pratica, non ha senso considerare una qualità in isolamento – né avere una vista indipendente per ciascuna qualità



## Architetture e qualità

- Le architetture sono critiche nel raggiungimento di certe qualità
  - quali approcci è possibile utilizzare in pratica per la progettazione per gli attributi di qualità?





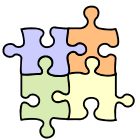
## Architetture e qualità

- Le architetture sono critiche nel raggiungimento di certe qualità
  - quali approcci è possibile utilizzare in pratica per la progettazione per gli attributi di qualità?
  - intuitivamente, è di fondamentale importanza poter sfruttare l'esperienza passata per produrre progetti migliori
    - “la codifica sistematica delle relazioni tra architetture e qualità migliora molto il processo di analisi e progettazione” [ESA]
  
- Alcune soluzioni proposte in letteratura
  - stili architettonici [POSA]
  - tattiche architettonici [SAP]
  - prospettive architettonici [SSA]
  
- Sono approcci il cui obiettivo è il riutilizzo di conoscenza relativa alla “soluzione esemplare di problemi significativi e ricorrenti”

21

Ottenere qualità: stili, tattiche e prospettive architettonici

Luca Cabibbo – ASw



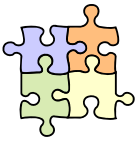
## \* Stili architettonici

- Uno **stile** (o **pattern**) **architettonico** [POSA]
  - 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*
  - lo schema di soluzione è specificato dalla descrizione dei componenti che lo costituiscono, le loro responsabilità e relazioni, e il modo in cui essi collaborano

22

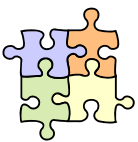
Ottenere qualità: stili, tattiche e prospettive architettonici

Luca Cabibbo – ASw



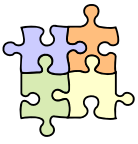
## Stili architetturali

- Un esempio di stile architetturale – Layers
  - problema
    - progettare un sistema complesso – sostenere sviluppo indipendente delle parti, modificabilità e/o portabilità
  - soluzione
    - scomponi il sistema in una pila di strati, con dipendenze dall'alto verso il basso
    - criterio di decomposizione – livello di astrazione delle funzionalità
  - sostegno alle qualità
    - positivo su modificabilità – ma anche sicurezza e affidabilità
    - potrebbe avere impatto negativo sulle prestazioni
- Altri esempi di stili architetturali
  - Pipes-and-Filters, MicroKernel, Client/Server, Peer-to-Peer, ...



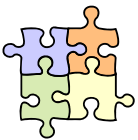
## Stili architetturali

- Osservazioni sugli stili architetturali
  - uno stile architetturale documenta delle esperienze di progettazione esistenti e ben provate
  - uno stile architetturale descrive una schema per *l'organizzazione strutturale fondamentale di sistemi software*
  - uno stile architetturale non è caratterizzato solo da un problema e da una soluzione
    - comprende anche criteri e linee guida per la selezione degli elementi software e delle relazioni tra di essi
    - comprende anche una giustificazione logica che collega problema e soluzione
  - in generale, ogni stile architetturale sostiene la costruzione di sistemi software con alcune proprietà di qualità ben definite
    - il problema affrontato è, di solito, il raggiungimento di alcune qualità – nell'ambito di un qualche specifico contesto



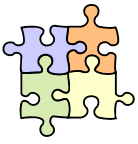
## \* Tattiche architetturali

- Una **tattica (architetturale)** [SAP] è una decisione di progetto che influenza il controllo (e dunque, il conseguimento) della risposta di un attributo di qualità
  - la progettazione di un sistema è basata su un insieme di decisioni – alcune decisioni sono relative alle funzionalità del sistema, mentre altre aiutano a raggiungere gli attributi di qualità
  - ciascuna tattica è/descrive un'opzione di progettazione per l'architetto, volta a controllare una certa qualità – le tattiche hanno un impatto diretto sulla risposta del sistema a qualche stimolo
  - le tattiche non vengono “inventate” – piuttosto sono usate per catturare quello che gli architetti fanno in pratica



## Tattiche architetturali

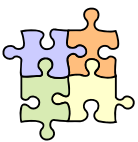
- Alcuni esempi di tattiche per le prestazioni
  - *increase available resources*
    - riduce il tempo di risposta a un evento usando risorse (processori, memorie, reti, ...) più veloci
  - *introduce concurrency*
    - riduce il tempo di risposta a un evento svolgendo in parallelo alcune parti dell'algoritmo
  - *reduce computational overhead*
    - riduce il tempo di risposta a un evento intervenendo sulla parte di interazione (non algoritmica) dell'elaborazione



## \* Prospettive architeturali



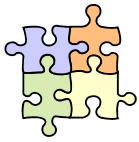
- Una **prospettiva architeturale (prospettiva)** [SSA] è una collezione di attività, tattiche e linee guida usate per garantire che un sistema esibisca un particolare insieme di proprietà di qualità correlate che richiedono riflessioni attraverso diverse viste architeturali del sistema



## Prospettive architeturali



- Una prospettiva ambisce a codificare il “corpo della conoscenza” circa la progettazione per una certa qualità
  - un modo per **sistematizzare** le competenze dell’architetto
  - per condividere e **riusare** le conoscenze dei bravi architetti
  - una **guida efficace** alla risoluzione di fattori architeturali
    - che cosa mi devo chiedere? quali aspetti/interessi devo prendere in considerazione? che cosa devo fare? quali attività devo svolgere? con quali modelli? con quali linee guida? quali stili e tattiche architeturali posso utilizzare? quali ulteriori problemi mi posso aspettare? quale bibliografia posso consultare?
    - utile specialmente dove un architetto non è familiare con gli interessi, i problemi e le soluzioni comuni



# Prospettive architeturali

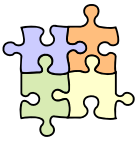


- [SSA] descrive ciascuna prospettiva in una forma standard
  - definizione della qualità di interesse
  - applicabilità – alle viste
  - interessi – qualità affrontate
  - attività – da eseguire per applicare la prospettiva alle viste
  - tattiche architeturali
  - problemi e rischi
  - bibliografia essenziale



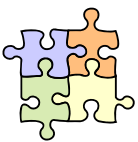
## \* Discussione

- Stili, tattiche e prospettive architeturali affrontano la progettazione per le qualità – ma in modo diverso tra loro
  - gli stili architeturali si pongono l'obiettivo di soddisfare simultaneamente un certo numero di attributi di qualità
    - sono utili soprattutto per effettuare una *decomposizione iniziale* del sistema, per garantire che il sistema sostenga effettivamente un certo insieme di qualità
  - le tattiche architeturali hanno l'obiettivo di migliorare il controllo di un singolo attributo di qualità
    - sono utili soprattutto per il *raffinamento* dell'architettura
  - le prospettive architeturali sono più generali, perché hanno l'obiettivo (o l'ambizione) di descrivere il corpo della conoscenza sui modi per controllare un'intera qualità
    - sono certamente un punto di partenza in tal senso



## Discussione

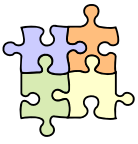
- Relazione tra gli approcci per le qualità e le viste
  - l'applicazione di uno stile architettuale porta all'identificazione/definizione degli elementi (e delle relazioni tra elementi) in una o più viste architeturali (di solito di una sola vista)
  - l'applicazione di una tattica architettuale porta al raffinamento degli elementi (e delle relazioni tra elementi) in una o più viste architeturali
  - l'applicazione di una prospettiva richiede di intervenire su diverse viste architeturali – al fine di analizzare e validare le qualità dell'architettura e di guidare ulteriormente le decisioni architeturali



## Discussione

- Vantaggi legati ad approcci architeturali 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 stile architettuale
- Rischi
  - possibilità di suggerimenti conflittuali
  - in alcuni casi i suggerimenti sono (troppo) generici





# Relazioni tra concetti fondamentali

