

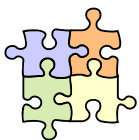
Architetture Software

Architetture software: Introduzione al corso

Dispensa ASW 010
ottobre 2014

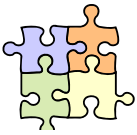
*L'inizio
è la parte più importante di un lavoro.*

Platone



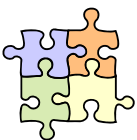
Architetture software

- Alcune caratterizzazioni di **architettura software**
 - the **software architecture** of a system is the set of **structures** needed to reason about the system, which comprise software elements, relations among them, and properties of both [SAP]
 - the **architecture** (of a software system) is the primary carrier of **system qualities** such as performance, modifiability, and security, none of which can be achieved without a unifying architectural vision [SEI]
 - a **software system's architecture** is the set of **principal design decisions** made about the system [Taylor, Medvidovic, Dashofy]
 - **architecture** is the **conceptual glue** that holds every phase of the project together for all its many stakeholders [SEI]



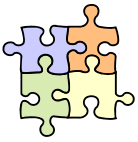
Che cosa sono le architetture software

- L'**architettura software** è il vettore principale delle **qualità** di un sistema software – come prestazioni, modificabilità e sicurezza – **nessuna delle quali può essere ottenuta senza una visione architeturale unificante**
 - nelle fasi iniziali di un progetto, l'analisi dell'architettura consente di garantire che l'approccio di progettazione prescelto conduca ad un sistema accettabile
 - l'architettura serve da progetto sia per il sistema che per il piano/progetto relativo al suo sviluppo – definendo le assegnazioni di lavoro che deve essere svolto dai team di progettazione ed implementazione
 - l'architettura ha un ruolo chiave anche per le attività di manutenzione successive al rilascio del sistema
 - in breve, l'architettura è il collante concettuale che tiene assieme ogni fase del progetto, per ciascuna delle sue numerose parti interessate



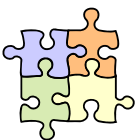
Che cosa sono le architetture software

- La disciplina delle **architetture software** – o **architettura del software** è interessata ai sistemi software complessi
 - alle *parti interessate* allo sviluppo di un sistema e ai loro *interessi* – sia funzionali che, soprattutto, di *qualità*
 - come prestazioni, modificabilità, sicurezza, disponibilità
 - alla *strutturazione fondamentale interna di un sistema*
 - quali elementi? con che responsabilità? quali collaborazioni?
 - sia dal punto di vista metodologico che da quello tecnologico
 - alle *relazioni tra la struttura interna di un sistema e le sue qualità esterne* – la comprensione di queste relazioni è alla base della progettazione dei sistemi software complessi
 - sulla base di questa comprensione, la *progettazione di un sistema complesso* può essere guidata dalle qualità fondamentali che esso deve esibire – procedendo mediante l'applicazione di principi, linee guida e soluzioni pre-esistenti



Percorso di studi

- Alcuni problemi affrontati dal corso di **Architetture software**
 - da un punto di vista metodologico
 - se organizzo il mio sistema in un certo modo, quali qualità sostengo? quali qualità penalizzo? perché?
 - se voglio ottenere una certa qualità, in che modo posso organizzare il mio sistema? perché?
 - da un punto di vista tecnologico
 - quali sono le principali tecnologie di middleware di supporto alla realizzazione di architetture software?
 - da un punto di vista sia metodologico che tecnologico
 - quale organizzazione per il sistema è supportata da una certo middleware? e quindi, quali qualità?
 - per organizzare il mio sistema in un certo modo, quale middleware? in che modo posso utilizzare in modo efficace quel middleware?



Obiettivo formativo

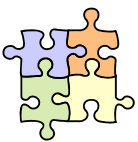
- Il corso di **Architetture software** ha l'obiettivo di presentare gli aspetti sia metodologici che tecnologici relativi alle architetture software, che hanno un ruolo fondamentale nel raggiungimento degli obiettivi di qualità (requisiti non funzionali) dei sistemi software.
- Specificamente, ha l'obiettivo di presentare metodologie e strumenti per la progettazione e la valutazione di architetture software e, in particolare, di architetture per sistemi software distribuiti e di tipo enterprise, incluse le architetture basate su componenti, le architetture orientate ai servizi e i sistemi software sul cloud.
- Alla fine del corso lo studente dovrebbe sapere impostare un progetto di un'architettura software, analizzandone dettagli e problematiche tecnologiche e metodologiche, e valutare l'architettura in termini di raggiungimento di obiettivi di qualità.



Programma (preliminare) del corso

□ Architetture software

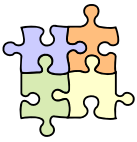
- architetture software: introduzione e concetti di base
- processo di definizione dell'architettura software
- parti interessate e interessi – requisiti e qualità
- descrivere l'architettura – viste e punti di vista
- progettare per le qualità – tattiche architettoniche e stili architettonici
- valutazione di architetture software (cenni)



Programma (preliminare) del corso

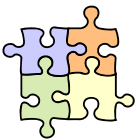
□ Architetture dei sistemi distribuiti

- architetture distribuite: client/server, peer-to-peer, DOA, messaging e integrazione di applicazioni, architetture a componenti, web services, SOA, gestione di dati distribuiti, cloud computing
- ### □ Middleware e tecnologie per sistemi distribuiti
- tecnologie DOA (RMI), messaging (JMS), componenti (EJB), web services, NoSQL, tecnologie per il Cloud Computing



Collocazione concettuale del corso

- Nell'ambito della Laurea Magistrale in Ingegneria Informatica, il corso di Architetture software
 - ambisce ad offrire una visione di alto livello – “unificante” – relativamente alla progettazione di sistemi software complessi
 - si colloca pertanto ad un livello di astrazione maggiore rispetto ad altri corsi di natura “tecnologica”
 - così come i sistemi software comprendono sia basi di dati, che infrastrutture di rete, che altri elementi
 - quindi conoscenze relative a questi altri corsi sono utili
 - ma non indispensabili – una conoscenza “di dettaglio” non è solitamente richiesta
 - *inoltre, per certi versi, il corso di Sistemi informativi si colloca ad un livello di astrazione ancora maggiore*
 - *o volendo, complementare, poiché si interessa soprattutto di aspetti di natura organizzativa ed economica*



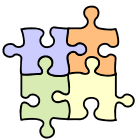
Prerequisiti

- **Architetture software**
 - è un corso per gli studenti del *secondo anno* della *Laurea Magistrale in Ingegneria informatica (ordinamento 270)*
 - tutto ciò che è stato studiato (o viene studiato) sarà utile
 - *Analisi e progettazione del software*
 - *Sistemi informativi su web, Sistemi operativi, Basi di dati I*
 - *Basi di dati II, Programmazione concorrente, Sicurezza dei sistemi informatici e delle reti, Big data, ...*
 - ma niente indispensabile (?)



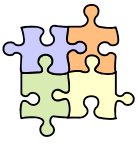
Prerequisiti

- Relazione tra *Analisi e progettazione del software* (APS) e *Architetture software* (ASW)
 - APS ha l'obiettivo di
 - presentare alcuni aspetti dell'ingegneria del software (processi, requisiti, analisi, progettazione)
 - con riferimento ad uno specifico metodo per lo sviluppo del software (processi iterativi, requisiti funzionali espressi come casi d'uso, OOA, OOD guidata dall'assegnazione di responsabilità)
 - ASW ha l'obiettivo di
 - presentare ulteriori aspetti dell'ingegneria del software
 - con particolare attenzione alla progettazione per i requisiti di qualità
 - consapevoli dell'esistenza di una molteplicità di metodi per lo sviluppo del software



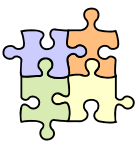
Architetture software

- Docente: **Luca Cabibbo**
- Rivolto a: studenti del secondo anno della LM II (270)
- CFU: 6
- Lezioni
 - nella primo semestre
 - dal 1 ottobre 2014 al 23 gennaio 2015
 - con sospensione delle lezioni dal 21 dicembre 2014 al 6 gennaio 2015
 - **lunedì e giovedì, dalle 9:00 alle 11:00**
 - in aula **N5**



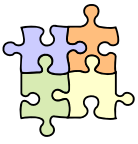
Architetture software

- Docente: **Luca Cabibbo**
- Ricevimento studenti
 - durante il primo semestre – salvo cambiamenti nell’orario di un mio altro corso
 - **mercoledì dalle 14:00 alle 15:30 dalle 14:30 alle 16:00**
 - oppure, **per appuntamento** (da concordare per posta elettronica)
 - negli altri periodi, l’orario di ricevimento potrebbe cambiare
 - DIA, secondo piano – stanza 211



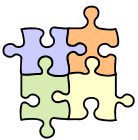
Sito del corso e materiale didattico

- Sito web del corso
 - <http://cabibbo.dia.uniroma3.it/asw>
- Il materiale didattico di riferimento può essere reperito sul sito del corso
 - dispense a cura del docente
 - (link ad) articoli e presentazioni

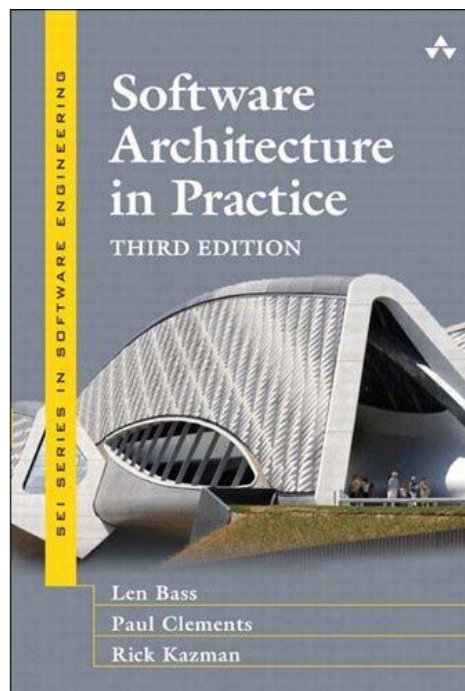


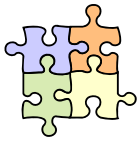
Ulteriore materiale didattico

- La parte metodologica del corso, e il relativo materiale didattico, è ispirata principalmente ai seguenti testi, utili per la consultazione (disponibili in biblioteca)
 - **Software Architecture in Practice [SAP]**
 - **Len Bass & Paul Clements & Rick Kazman**
 - Addison Wesley, 2013 (third edition)
 - **Pattern-Oriented Software Architecture (vol. 4) – a pattern language for distributed computing [POSA4] o [POSA]**
 - **Frank Buschmann & Kevlin Henney & Douglas C. Schmidt**
 - John Wiley & Sons, 2007
 - **Software Systems Architecture [SSA]**
 - **Nick Rozanski & Eoin Woods**
 - Addison Wesley, 2012 (second edition)

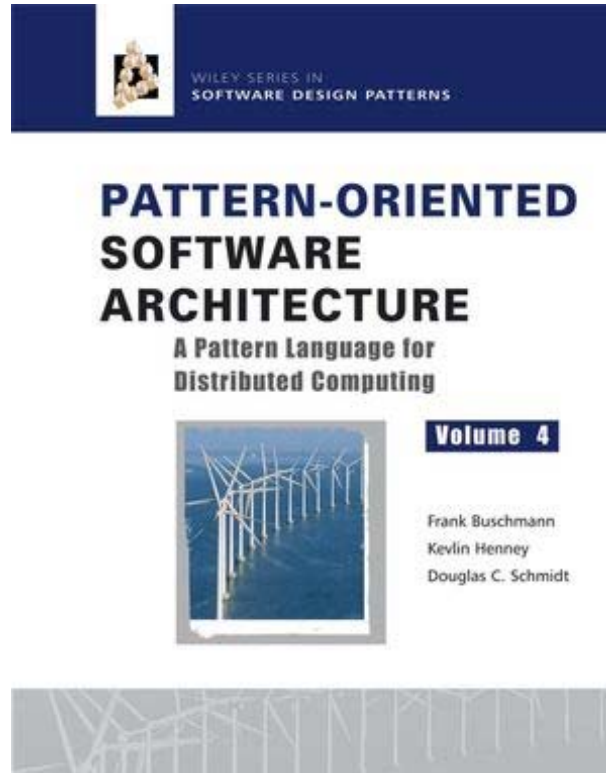


Ulteriore materiale didattico [SAP]





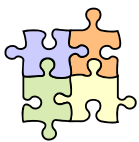
Ulteriore materiale didattico [POSA4]



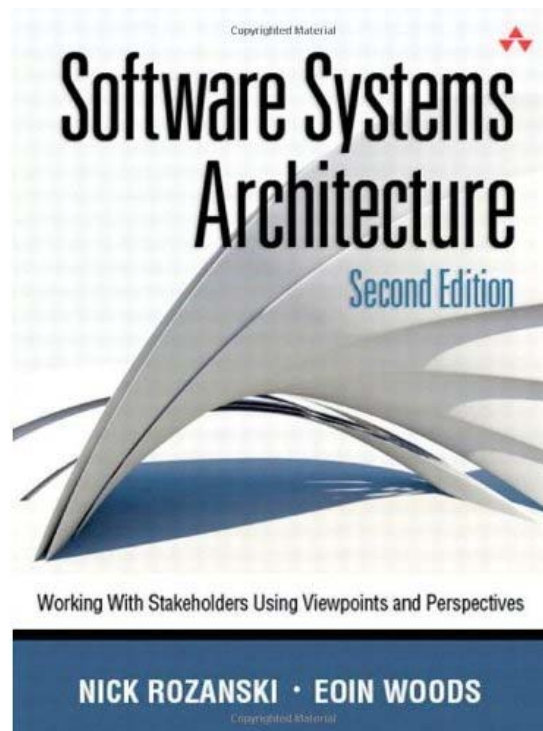
17

Architetture software: Introduzione al corso

Luca Cabibbo - ASw



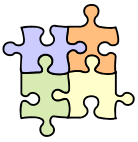
Ulteriore materiale didattico [SSA]



18

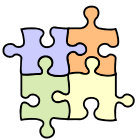
Architetture software: Introduzione al corso

Luca Cabibbo - ASw



Esami

- L'esame prevede lo svolgimento delle seguenti prove "finali"
 - prova scritta – circa 15 punti su 30
 - due domande da scegliere su tre – circa 80 minuti
 - prova orale – circa 15 punti su 30
 - discussione dell'elaborato della prova scritta
 - un paio di ulteriori domande
 - nei giorni immediatamente successivi alla prova scritta
- Calendario degli esami
 - gennaio-febbraio 2015, giugno-luglio 2015, settembre 2015
- Al massimo tre partecipazioni nell'ambito dell'anno accademico



Homework

- Durante lo svolgimento del corso saranno proposte alcune prove intermedie – "homework"
 - ovvero da svolgere autonomamente – "a casa" – consegnando i relativi elaborati entro tempi opportuni
 - possibili argomenti delle prove intermedie
 - lettura di articoli scientifici
 - con domande relative alla loro comprensione e correlazione con altri argomenti studiati nel corso
 - sperimentazione di tecnologie
 - piccoli progetti di utilizzazione del middleware
 - non obbligatorie – ma consigliate a chi frequenta il corso e ha intenzione di sostenere l'esame al primo appello



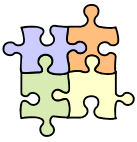
Esame con homework

- Modalità *alternativa* di svolgimento dell'esame – primo appello (gennaio-febbraio 2015)
 - se la valutazione delle prove intermedie è positiva, allora viene “mediata” con la valutazione (se almeno sufficiente) delle prove scritte e orale, con i seguenti pesi
 - homework – circa 5 punti su 30
 - scritto – circa 12.5 punti su 30
 - orale – circa 12.5 punti su 30
- Esame con homework, appelli successivi al primo
 - chi partecipa ad un appello, ma senza verbalizzare (perché respinto o rifiuta il voto), agli appelli successivi perde il “voto” dei suoi homework
 - chi non partecipa al primo appello, negli appelli successivi il “peso” dei suoi homework viene ridotto al 50%



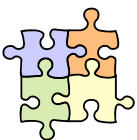
Registrazione al corso

- Gli studenti che frequentano il corso di Architetture software e/o intendono sostenere il relativo esame sono invitati a registrarsi al corso
 - modulo: <http://cabibbo.dia.uniroma3.it/asw/registrazione>
 - perché
 - per sapere quanti e chi siete
 - per conoscere i vostri indirizzi di posta elettronica (se serve, per contattarvi)



Da Ingegneria del sw ad Architetture sw

- Il corso di **Architetture software** (per la *LM II ord. 270, 6 CFU*) nasce come trasformazione del corso di **Ingegneria del software** (per la *LM II ord. 509, 5 CFU*)
 - ne condivide in modo significativo obiettivi formativi, programma, contenuti e materiale didattico
 - tuttavia, enfatizza ancora di più gli aspetti relativi alla progettazione di sistemi software complessi e l'attenzione agli obiettivi di qualità
- Gli studenti del corso di **Ingegneria del software** (*LM II ord. 509*) sono invitati a contattare il docente al più presto



Per gli studenti Erasmus

- Informazioni per gli studenti Erasmus
 - con *alcuni* studenti Erasmus – per fortuna non con tutti – si sono verificate alcune situazioni spiacevoli – che è bene che non si ripetano più
 - a causa di ciò, gli studenti Erasmus devono
 - contattare il docente per incontrarlo e farsi conoscere, il prima possibile – per verificare che siano effettivamente interessati al corso e ne posseggano i prerequisiti necessari – altrimenti è bene che cambino il loro *learning agreement*
 - andare dal docente, a ricevimento studenti, con largo anticipo rispetto a quando intendono sostenere l'esame – per far sì che la preparazione proceda in modo corretto
 - il docente sottolinea che finora ha trattato – e continuerà a trattare – gli studenti Erasmus *allo stesso modo* – dunque, né meglio né peggio – degli studenti locali