Frontespizio

• Database NEO4J

TODO!!! Hapi.Js, Apache struts Grafico Poligonale per i pesi dei quality Attribute dei QM

Qualità spesso in contrasto con i costi!

PROPOSTE DI TITOLO PER LA TESI

MULTI-CRITERIA DECISION MAKING APPLICATO AL PROCESSO DI SELEZIONE DI UN FRAMEWORK DI SVILUPPO WEB

SELEZIONE DI UN FRAMEWORK DI SVILUPPO WEB CON IL METODO MULTI-CRITERIA DECISION MAKING.

MULTI-CRITERIA DECISION MAKING APPLICATO AL PROCESSO DI SELEZIONE DI UN WEB DEVELOPMENT FRAMEWORK.

SELEZIONARE UN WEB FRAMEWORK APPLICANDO IL METODO MCDM

APPLICAZIONE DEL MULTI-CRITERIA DECISION MAKING AL PROCESSO DI SELEZIONE DI UN WEB FRAMEWORK.

AUTORE: ANDREA SERAFINI

Parole Chiave: Multiple Criteria Decision Making; Analisi Multi Criteriale, Web Frameworks; Software Quality;

[1 Introduzione 6](#_Toc17210599)

[2 Background 9](#_Toc17210600)

[2.1 Le Entità 10](#_Toc17210601)

[2.2 Il Quality Model 12](#_Toc17210602)

[2.3 Aprroccio Generico di Analisi Multi-Criteriale 13](#_Toc17210603)

[2.4 l’analisi multi-criteriale applicata al nostro caso di studio 14](#_Toc17210604)

[3 Screening Preliminare 18](#_Toc17210605)

[3.1 Metodologia di Selezione 24](#_Toc17210606)

[3.2 Le Alternative 25](#_Toc17210607)

[4 Definizione del modello e degli attributi di Qualità 49](#_Toc17210608)

[4.1 I Quality Attributes (Mappare Con ISO le Metriche!?) 50](#_Toc17210609)

[4.2 Metriche individuate 50](#_Toc17210610)

[4.3 Assegnare i Pesi 51](#_Toc17210611)

[4.4 Quality Attribute Presentazione/Generiche 56](#_Toc17210612)

[4.5 Quality Attribute Trends 59](#_Toc17210613)

[4.6 Quality Attribute Tecnologie & Features 63](#_Toc17210614)

[4.7 il Quality model in sintesi 69](#_Toc17210615)

[5 Valutazione delle Alternative 71](#_Toc17210616)

[5.1 Normalizzazione dei dati 72](#_Toc17210617)

[5.2 Valutazione dei quality attribute 73](#_Toc17210618)

[5.3 Valutazione del Quality Model Presentazione/Generiche 73](#_Toc17210619)

[5.4 Valutazione del Quality Model Trends 76](#_Toc17210620)

[5.5 Valutazione del Quality Attribute Tecnologie e Supporto allo Sviluppo WEB 79](#_Toc17210621)

[5.6 Valutazione della qualità dei Frameworks 83](#_Toc17210622)

[6 Caso di Studio 85](#_Toc17210623)

[7 Related work 86](#_Toc17210624)

[8 Bibliografia 89](#_Toc17210625)

Glossario

API: Application programming Interface.

CMS: Contenent Management System.

CSRF:Cross-site request forgery (Vulnerabilità di Applicazioni Web).

CRUD: Create, Read, Update, Delete (operationi su un Database).

JDBC Java: Data Base Connectivity.

MCDM: Multiple Criteria Decision Making.

MCA/AMC: Analisi Multi-Criteriale.

MVC: Model-View-Controller (Design Pattern).

ODBC Open: Data Base Connectivity.

ORM: Object Relational Mapping.

REST

SOP Same Origin Policy.

XSRF: Cross-site request forgery.

# Introduzione

Il WEB ha subito dei cambiamenti inimmaginabili dalle prime definizioni di sito web come un sistema di gestione di informazioni come proposto da Tim Berners-Lee[[1]](#footnote-1) nel 1989 all’evoluzione iniziata nei primi anni 2000 fino ad oggi, i siti web sempre più complessi non possono essere più visti come semplici contenitori di informazioni ma come vere e proprie applicazioni per la fornitura di servizi ad utenti sempre più esigenti, perché se sono cambiate le applicazioni di pari passo sono cambiate anche le abitudini degli utenti.

Dal punto di vista dello sviluppatore questa evoluzione rende sempre più complesso il processo di sviluppo di un sito web, ma possiamo ancora chiamarlo sito web? Oggi si sente più spesso parlare di applicazioni web che di ‘sito’, questo in seguito all’esplosione di tecnologie mobili gli utenti hanno iniziato a richiedere diversi servizi, come le applicazioni ‘social’ o quelle di e-commerce, colossi di che tutti oggi conosciamo come Amazon, Facebook, Google hanno fatto la loro fortuna sulla fornitura di servizi basati sulla diffusione tramite internet che ha permesso la diffusione in scala mondiale dell’utilizzo di tali servizi attraverso le moderne applicazioni Web.

Dal punto di vista tecnico sono state sviluppate e rese disponibili moltissime soluzioni che permettono di creare moderne applicazioni web: possiamo infatti utilizzare CMS[[2]](#footnote-2), Toolkit[[3]](#footnote-3) e Frameworks che permettono di sviluppare applicazioni complesse e di qualità.

Un interessante articolo riguardante l’evoluzione del web è proposto da Bodje in [1] nel quale si chiariscono termini e concetti riguardanti la ‘Web Engineering’, una panoramica interessante è stata fatta per i ‘tool’ che ha a disposizione uno sviluppatore, nel nostro caso abbiamo deciso di prendere in esame i frameworks perché ci forniscono un ambiente dove lo sviluppatore ha la possibilità di riutilizzare sia il design oltre al riutilizzo di codice, a differenza dei toolkit che che sono orientati al riutilizzo di codice ma sono spesso indipendenti dallo scopo dell’applicazione e dei CMS che ci forniscono una web application preconfezionata e totalmente personalizzabile.

Un Framework di Sviluppo Web è uno strumento Software che permette allo sviluppatore di ridurre il carico di lavoro, fornisce soluzioni standardizzate a problemi ricorrenti che si verificano durante lo sviluppo di un’applicazione web. I Frameworks in particolare ci permettono di sviluppare un software ad hoc per le nostre esigenze ottimizzato e di qualità superiore a patto che si rispettino pattern e standard. I Frameworks ci garantiscono inoltre la possibilità di sviluppare software secondo un’architettura multi-tier in particolare secondo il modello MVC[[4]](#footnote-4). Nel corso degli anni sono stati proposti una moltitudine di Framework per lo sviluppo di applicazioni web sia frontend che backend con caratteristiche differenti, scritti con diversi linguaggi di programmazione (come Java, Javascript, PHP, Ruby e Python). Alcuni sono indicati per lo sviluppo di applicazioni mobili come single page application altri per la realizzazione di e-commerce, gestionali, cloud, streaming e molti altri tipi di servizi, è quindi importante scegliere lo strumento giusto.

Un problema complesso che un’azienda o uno sviluppatore deve risolvere prima di iniziare lo sviluppo è scegliere lo strumento o l’insieme di strumenti più adatti alle proprie esigenze.

In questo lavoro di tesi ci focalizzeremo sull’analisi e sulla definizione di criteri di selezione di framework di sviluppo Web.

“Using a framework allows the developer to significantly reduce the costs and time necessary for completing the project without limiting its functionality. Although frameworks can be seen as a universal base for a project, selecting the framework affects the cost and time needed for completing the project. Also, due to efficiency issues and the character of the industry, the framework selection affects the costs of application maintenance, particularly the demand for the resources of a production server. Consequently, it becomes crucial to be able to quickly and reliably compare the available frameworks in terms of their usefulness for developing a new web application.” Cit. SELECTING A PHP FRAMEWORK FOR A WEB APPLICATION PROJECT — THE METHOD AND CASE STUDY Conference Paper · March 2015 [2]

La scelta di utilizzare un Framework rispetto ad un altro può essere difficile, ed una scelta errata potrebbe costarci molto tempo di sviluppo in più, si potrebbe dover scrivere del codice per risolvere problemi già noti e gestiti da altri Framework solo perché l’obiettivo di un Framework non è corrispondente con il nostro obiettivo.

Al fine di risolvere questo problema è stato modellizzato un problema decisionale con criteri multipli, questo metodo è stato adattato alle necessità dettate dal nostro obiettivo ed integrato con una valutazione di caratteristiche non solo tecniche ma anche qualitative. Questo tipo di problemi sono identificati come Multiple Criteria Decision Making (MCDM), è sorprendente come questo tipo di problemi trovi riscontro in moltissimi campi di applicazione, durante la mia personale ricerca si soluzioni applicabili a questo tipo di problemi ho potuto verificare l’utilizzo in campi assai differenti, tra questi possiamo citare, software, meteorologia, gestione delle risorse energetiche, ottimizzazione dei costi, ecologia ed anche studi astratti che prescindono dal campo di applicazione. Questo mi ha portato ad abbracciare questo tipo di approccio nella risoluzione del mio problema, di combinare soluzioni già studiate con l’introduzione di modifiche in relazione al mio obiettivo. Il problema proposto come già accennato riguarda la selezione di un Framework di Sviluppo Web il mio personale punto di vista si basa su tre attributi di qualità, l’usabilità, il supporto tecnologico offerto dal Framework, e la community dello stesso.

Possiamo riassumere questa ‘Analisi Multi-Criteriale’ nelle seguenti attività: la prima attività è uno screening preliminare dei frameworks disponibili che rispettano un insieme di caratteristiche, definizione dei criteri di valutazione, ed infine una valutazione ed analisi comparativa dei singoli framework, queste attivita che vedremo in dettaglio nei prossimi capitoli ci hanno permesso di trovare una soluzione al nostro problema decisionale, questa soluzione nel nostro caso è proprio un Framework che si avvicina di più al nostro obiettivo rispetto agli altri presi in esame.

# Background

Al fine di affrontare la scelta del framework più idoneo il mio studio fa uso di strumentidi supporto ai problemi decisionali con criteri multipli (Multiple Criteria Decision Making) basato su un approccio di analisi Multicriteriale (MCA) combinato con l’utilizzo di un modello di qualità. L’Analisi Multi-Criteriale ci consente di di selezionare un Framework con caratteristiche adeguate ai nostri bisogni da utilizzare come strumento di supporto per lo sviluppo di un applicazione web, questo approccio ci permetterà di prendere una decisione basata sulla valutazione di metriche basate sulle esigenze degli sviluppatori e sui requisiti dell’applicazione, l’obiettivo è di selezionare il framework che ci permetterà di sviluppare l’applicazione con costi di sviluppo contenuti, avere una buona manutenibilità, e di essere conforme agli standard attuali inoltre avremo a disposizione tutti gli strumenti necessari per lo sviluppo.

L’Analisi Multi-Criteriale è uno strumento che viene utilizzato per prendere decisioni dove fattori spesso eterogenei giocano un ruolo importante nella valutazione di un processo decisionale. Questi fattori possono ingannare i decisori in quanto la natura di questi non è formale risulta difficile misurarli. In un generico processo decisionale possiamo individuare le seguenti entità: obiettivo, decisori, alternative, criteri e valutazioni. Queste entità prescindono dall’ambito di applicazione dell’analisi sono comuni ad ogni Analisi Multi-Criteriale.

In particolare la nostra Analisi Multi-Criteriale combinerà l’utilizzo classico con una valutazione basata su un Modello di Qualità come definito in [3], questa combinazione permette di definire il proprio criterio di valutazione e può essere modificato nel caso in cui l’obiettivo cambiasse.

Va sottolineato che questa tesi non vuole fornire un algoritmo decisionale ma uno strumento che il/i decisore/i possono utilizzare come strumento di supporto durante un processo decisionale. È infatti responsabilità del decisore individuare i criteri, le metriche, valutarli ed infine prendere una decisione.

## Le Entità

Nel nostro problema come già introdotto precedentemente troviamo diverse entità, ognuna di queste ha un ruolo fondamentale nella ricerca della soluzione ideale al nostro problema, di seguito elenchiamo e espletiamo in dettaglio come queste sono state modellizzate e come interagiscono in relazione al nostro problema.

**Obiettivo**

L’obiettivo definisce lo scopo per il quale viene effettuata l’Analisi, il risultato quindi dovrà avvicinarsi il più possibile all’obiettivo prefissato, spesso questo è definito da stakeholder che non partecipano attivamente alla decisione ma l’influenzano indirettamente. Nel nostro caso di studio l’obiettivo è selezionare un Framework per sviluppare un’applicazione web tradizionale che ci permetta di avere a disposizione gli strumenti necessari allo sviluppo, ridurre i costi di sviluppo, avere una buona usabilità e manutenibilità.

**Alternative**

Le Alternative rappresentano lo spazio di soluzioni, ovvero tutti gli elementi appartenenti ad un insieme con caratteristiche comuni che potrebbero essere scelti come soluzione.

(unito con i criteri di selezione per costruire l’insieme di alternative.) durante la prima fase di questo studio viene individuata una lista arbitraria di criteri di selezione, l’individuazione dei criteri di selezione utili al nostro scopo non è banale, influenzerà infatti l’intero svoglimento ed il metodo con cui verrà condotto il nostro studio ed ovviamente il risultato finale. Questi criteri ci permettono di restringere anche il campo delle possibili soluzioni ovvero dei Framework che potremmo individuare nelle fasi successive. Tra questi criteri possiamo citare: l’utilizzo di un determinato linguaggio di programmazione, il tipo di Licenza, l’adozione, il tipo di Framework, e la vitalità. I criteri di selezione possono essere definiti sia dal decisore che da stakeholder esterni.

dopo aver individuato un insieme di criteri di selezione si inseriranno in una lista un insieme di Frameworks anche questi scelti arbitrariamente, verrà selezionato l’insieme di Frameworks che rispettano i criteri di selezione che sono stati individuati nella fase precedente.

I Frameworks che rispettano i criteri di selezione andranno a formare lo spazio delle nostre possibili soluzioni questi Frameworks verranno chiamati ‘alternative’ e si analizzeranno in dettaglio.

**Criteri di Valutazione**

I Criteri di Valutazione in un processo decisionale rappresentano le metriche in base alle quali verranno effettuate le valutazioni. Nella nostra analisi i criteri di valutazione sono sostituiti dal quality model, all’interno del quale troviamo gli attributi di qualità (Quality Attributes), all’interno dei quali troviamo le metriche.

**Decisore**

Il Decisore in un Analisi Multi-Criteriale è l’attore che ha il compito di definire i criteri di selezione e valutazione delle alternative, nella maggiornaza dei casi il decisore non è una sola persona ma un gruppo di persone che individuano e definiscono i criteri, spesso possono costruire anche lo spazio delle soluzioni altre volte lo spazio delle soluzioni è un insieme di alternative predefinito dagli stakeholder. Nel caso di studio esposto vi è un unico decisore le decisioni sono prese utilizzando un approccio basato sulla conoscenza e sull’esperienza personale, in supporto alle decisioni prese è stata utilizzata anche la bibliografia elencata.

**(Vecchio)**

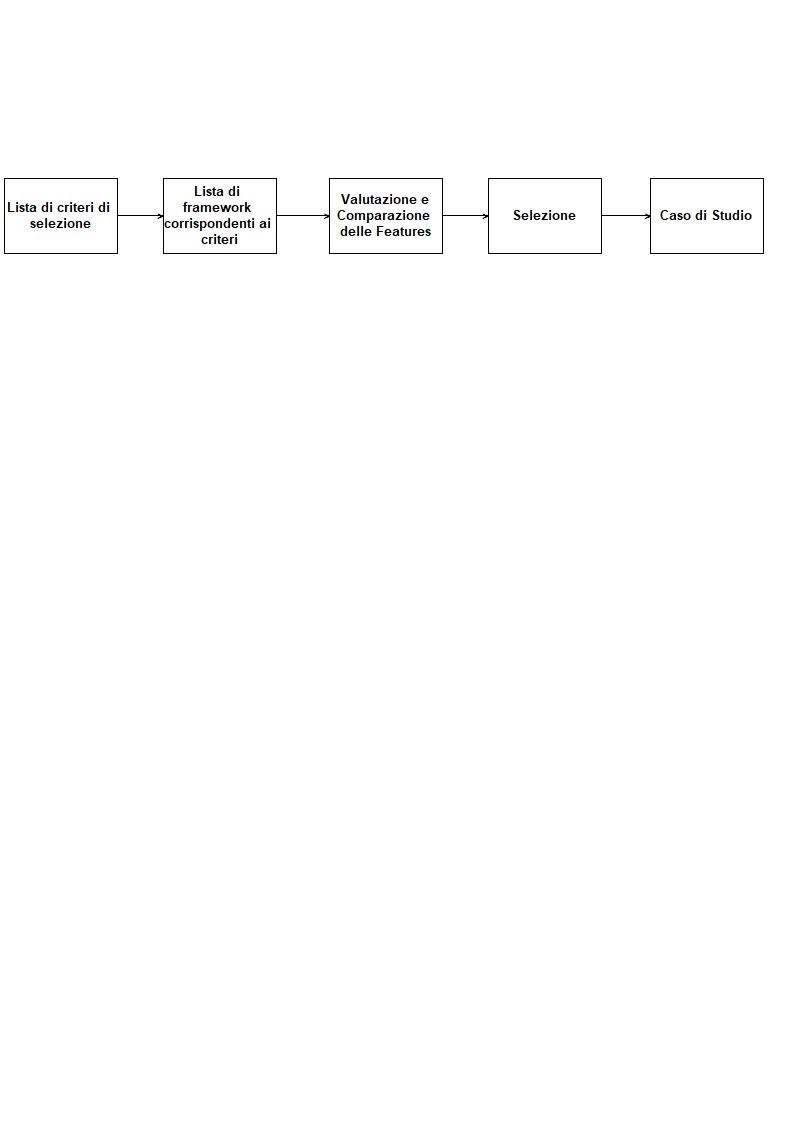
Analizziamo le Features significative per il nostro obiettivo di ogni Framework selezionato, assegneremo un ‘peso’ ad ogni Feature ed attraverso una metrica verrà assegnato un punteggio ad ogni Framework, verrà prodotta una tabella comparativa che ci permetterà infine di giungere ad una scelta finale in base ad un metodo di valutazione e selezione (o ad una combinazione di scelte nel caso il nostro scopo fosse quello di sviluppare ad esempio un’applicazione sia backend che frontend).

Il metodo di selezione consiste nell’individuazione di Features e Tecnologie messe a disposizione dai Frameworks che siano importanti per il raggiungimento del nostro obiettivo (lo sviluppo di una web application con determinati requisiti). In questa fase verranno elencate i pro e i contro che incontreremo scegliendo un determinato Framework, del tempo necessario allo sviluppatore per poter padroneggiare il Framework, si produrranno dei casi d’uso dove si evidenzieranno le caratteristiche dei diversi ambienti di sviluppo.

**Soluzione**

La soluzione è L’alternativa che si avvicina di più all’obiettivo. Nel nostro caso sarà il Framework che ha ottenuto il punteggio maggiore.

**Confronto Empirico:** Per avvalorare il nostro studio basato sul metodo MCDM viene fatto un confronto empirico tra il risultato ottenuto dalla nostra analisi multi-criteriale (MCDM) ed una serie di casi d’uso dove testeremo i 3/5 FullStack framework con il punteggio maggiore.Si è scelto di utilizzare il sott’insieme di Frameworks che ci permetta di sviluppare tutti i diversi livelli dell’applicazione. Infine metteremo a confronto le due classifiche e confronteremo le differenze tra i due metodi applicati.



Nel diagramma precedente possiamo distinguere le differenti fasi dell’intero MCDM, in particolare attraverso le prime due fasi ovvero l’associazione dei Framework con i Criteri di Selezione andiamo a creare l’insieme delle possibili soluzioni che verrà chiamato ‘Alternative’. La fase 3 e 4 rappresenta il processo decisionale del MCDM ed infine i casi di studio rappresentano una fase opzionale che ci permetterà di verificare la bontà dei risultati del MCDM.

## Il Quality Model

L’introduzione di un Modello di Qualità in sostituzione dei classici criteri di valutazione è stato necessario per poter valutare la qualità del software (dei frameworks), in accordo con *Basciani, et al.* In [3], alcuni modelli di qualità limitano la valutazione di qualità in quanto non sono personalizzabili secondo le esigenze di un caso specifico, grazie al meta modello specificato in [3] si può personalizzare una gerarchia di caratteristiche di qualità, permettendoci quindi di costruire un proprio modello di qualità. In particular, each quality attribute consists of an expression defining

how the values of the sub-attributes have to be aggregated [3]. In questo modello di qualità le foglie del modello sono metriche i cui valori sono calcolati da un ‘metric provider’. *‘The proposed approach is artifact independent in the sense that it can be applied to assess the quality of any kind of modeling artifacts. We are not aiming at proposing “yet another quality model”, but an holistic approach providing users with the means to specify and customize their own quality definition’* [3]*.* Essendo un approccio generalizzato per la valutazione di un artefatto, indipendente dall’artefatto stesso, è stato abbracciato questo tipo di approccio di valutazione. In questo meta modello viene specificata una struttura gerarchica dove la radice è il Quality Model costituito da QalityAttributes, ValueTypes e MetricProvider. Gli attributi di qualità modellizzano un aspetto che contribuisce alla valutazione di qualità dell’artefatto. Verrà quindi specificato dal quality attribute come aggregare gli attributi. A differenza dell’approccio proposto in ***Automated Quality Assessment of Modeling Artifacts*** [2] non utilizzeremo MdeForge con lo scopo di automatizzare il processo di valutazione ma utilizzeremo solo la parte teorica riguardante il meta-modello ed il modello di qualità. Ogni attributo in questo modello ha un valore che rappresenta il suo valore calcolato e può essere di due tipi, SingleValue o AggregatedValue per semplicità il valore assunto dai nostri quality attribute sarà SingleValue e calcolato da semplici regole che sostituiranno il MetricProvider. Di seguito una immagine descrive l’approccio proposto da Basciani ET AL in [3].

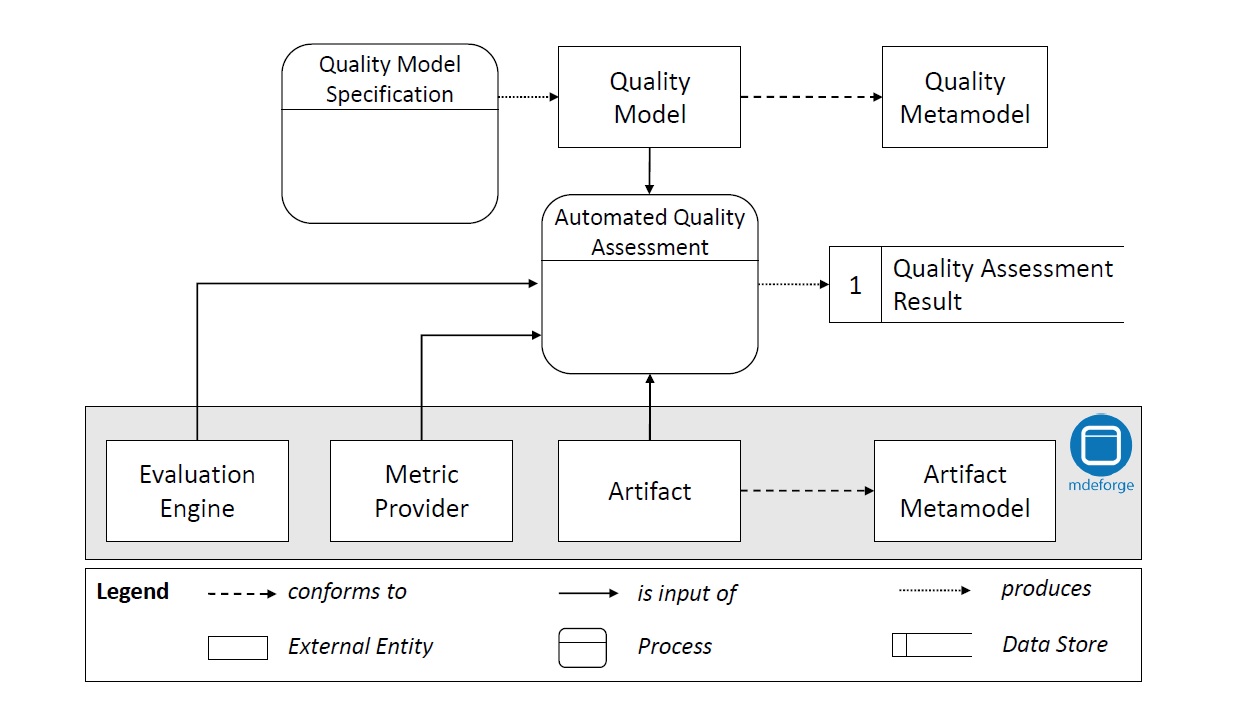
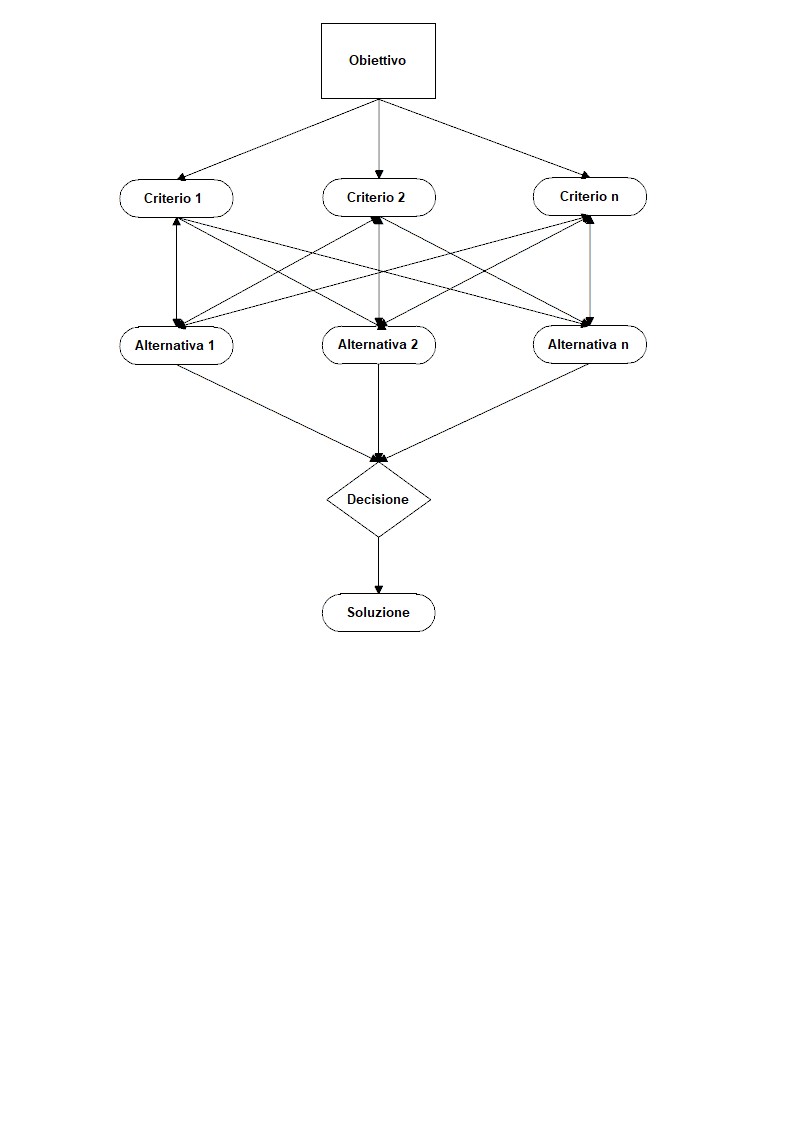


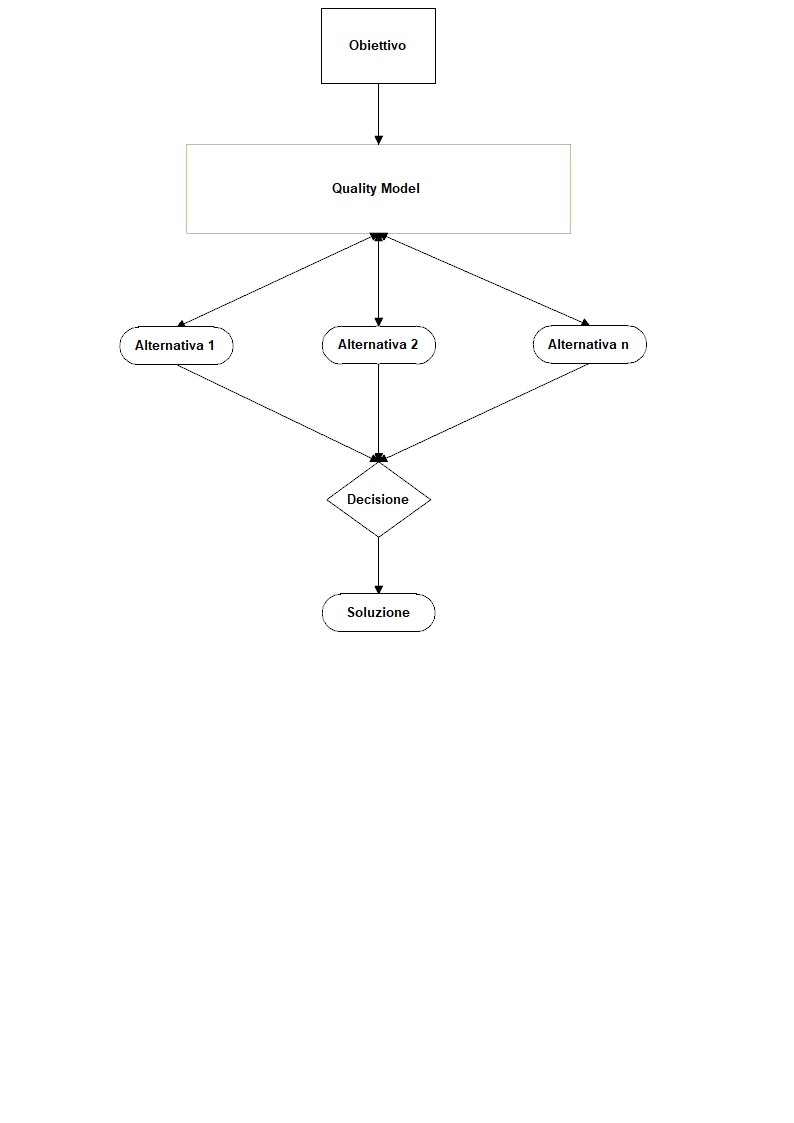
Figura In questo diagramma possiamo analizzare le entità che compongono il Quality Model definito in [3] .

## Aprroccio Generico di Analisi Multi-Criteriale

Nella Figura 1 possiamo vedere le relazioni tra le entità del nostro processo, l’entità Obiettivo influenza direttamente le metriche scelte per la valutazione, le metriche nel nostro caso particolare corrispondono al Quality Model che specificheremo in dettaglio successivamente, queste metriche a loro volta hanno una relazione con le Alternative che nel nostro corrispondono proprio ai frameworks presi in esame.

## l’analisi multi-criteriale applicata al nostro caso di studio

Nella Seguente Figura possiamo vedere come l’uso del Quality Model cambi il classico approccio di un’Analisi Multi-Criteriale. In questo approccio i criteri di valutazione sono definiti nel modello di qualità questo influenzerà direttamente la valutazione delle alternative in quanto all’interno di esso vengono definiti sia i criteri che le metriche di valutazione.



In Accordo con Zhi-Gang ET AL in [4] formalizziamo analogamente il nostro problema come un Multiple Attribute Decision Making sostituendo infatti il Quality Model ai Criteri, la valutazione verte sui Quality Attribute del modello di qualità. Di seguito formalizziamo il nostro problema come:

# Screening Preliminare

In questo capitolo viene definito lo spazio di soluzioni, che d’ora in poi chiameremo Alternative, le alternative formano l’insieme di possibili soluzioni che rispettano determinate proprietà, partiremo quindi da un insieme più grande di frameworks e costruiremo un sott’insieme chiamato Alternative, questa formalizzazione è necessaria in quanto l’insieme delle alternative non è dato, va quindi costruito attraverso la definizione di regole. Formalizzando l’insieme delle Alternative come un insieme matematico possiamo dire con certezza se un elemento appartiene o no all’insieme, questo ci permetterà ‘filtrare’ i framework e concentrarci solo su quelli che hanno la possibilità di essere selezionati come ‘Soluzione’ risparmiando molto tempo nella successiva fase di analisi. Le proprietà che ogni Framework deve rispettare riguardano: il linguaggio di programmazione, il tipo di licenza software, il tipo di Framework, e l’utilizzo del framework stesso.

*A = {f A : f (lp, ls, t, a)};*

*dove:*

*A= Insieme delle Alternative (Frameworks)*

*lp = Linguaggio di Programmazione; l = Licenza Software;*

*t=tipo di framework; ad=adozione;*

Quindi un Framework appartiene all’insieme delle Alternative se e solo se rispetta i criteri di selezione definiti nella sezione successiva, è bene specificare che anche i criteri di selezione possono essere modificati in base alle necessità.

**Adozione**

Un aspetto fondamentale che rende vincente un Framework rispetto ad un altro è l’adozione, ovvero l’utilizzo dello stesso da parte di aziende di rilievo, vi sono anche aziende che hanno sviluppato alcuni dei frameworks presi in esame e che hanno sposato la politica open source, questo ci permette di affidarci a dei frameworks con una grande ‘community’ alle spalle, tra queste aziende possiamo citare colossi come Microsoft, Google, Facebook, Netflix, Accenture, IBM, Apple, Amazon, Twitter, Samsung, Sony e molte altre. L’Adozione da parte di una di queste aziende è un’indice di qualità che non possiamo ignorare oggigiorno, queste aziende hanno come punto chiave la diffusione dei loro servizi digitali e fisici attraverso la rete, è stato proprio questo aspetto che le ha fatte esplodere negli ultimi anni, un fattore altresì importante è stata la diffusione degli smartphone che hanno reso fruibili applicazioni web in qualsiasi posto vi sia una connessione web, questo aspetto va considerato in quanto alcuni Frameworks sono nati per lo sviluppo di applicazioni web ottimizzate per il loro utilizzo su dispositivi mobili. É quindi necessario per gli scopi di questo studio che i Frameworks selezionati non siano solo uno strumento didattico ma anche uno strumento di Business.

**Vitalità (DRAFT TRASFERIRE ED INTEGRARE QUESTA PARTE CON IL QM DEL CAP 3)**

La Vitalità del Framework è un aggregato di attributi detti quality attribute rilevanti per valutare la qualità del framework. È importante che il metodo di valutazione sia uniforme per tutti i Frameworks, andremo quindi a definire un Quality Model regolato da un Quality Meta-model.

Consequently, a quality assessment approach

should be open to custom quality models that users can easily define

to support the quality assessment problem at hand. [Citazione*Automated Quality Assessment of Modeling Artifacts pag 11****].***

The Quality Metamodel (QMM)

It plays a key role in the proposed approach since it enables the specification

of quality models according to user requirements. The metamodel is shown

in Fig. 2 and consists of a number of constructs as explained in the following.

QualityModel is the root element consisting of QualityAttributes,

ValueTypes, and MetricProviders.

A QualityAttribute represents a quality aspect that is considered to be

relevant for contributing to the quality assessment of a given artifact. A quality

attribute, like maintainability can be an aggregation of other attributes, like

changeability and modularity. Thus, each quality attribute specifies how to aggregate

the contained attributes in order to provide an overall quality value for the

considered attribute as described below. [Basciani ET AL Automated Quality Assessment of Modeling Artifacts]

Tra le metriche rilevanti la nostra valutazione troviamo:

* Numero di Commit
* Contributors
* Forks
* Release Frequency

Gli attributi devono rispettare la seguente metrica:

Relesease Frequency >= 1 / 2 per Anno

Contributors >= 100

Commits >= 12 per Anno

Forks >= 5

Un criterio importante per il nostro studio sarà la vitalità di un software per ‘vitalità’ è intesa la valutazione di alcuni dati estrapolati da repository come GitHub e SourceForge, infatti all’interno di questi possiamo trovare dati di interesse come il numero di Commit, di release, i progetti paralleli nati dal Framework, questo sarà argomento verrà approfondito successivamente con i dati raccolti nei repository, viene considerato un indice di qualità il numero di Fork di un Frameworks, l’apprezzamento, i contributors, e non poco importante la data in cui sono avvenute queste operazioni, accedere a delle infografiche che ci permettono di capire se il Framework è in evoluzione,in declino o se è un progetto ormai ‘morto’.

**Adozione**

Un criterio di selezione non trascurabile del nostro screening preliminare è il tipo di Licenza Software, al fine di selezionare uno strumento cercando di minimizzare i costi di licenza software, ci orienteremo su Frameworks distribuiti con licenza opensource. Il requisito che la Licenza software deve rispettare e che sia una licenza FOSS, ovvero riconosciuta o dalla Open Source Initiative o dalla Free Software Foundation, questo aspetto è imprescindibile dalla scelta del nostro Frameworks, questo ci permetterà oltre che al mero utilizzo del Framework anche di sviluppare ed ampliare il framework secondo le nostre necessità.

**Licenza Software**

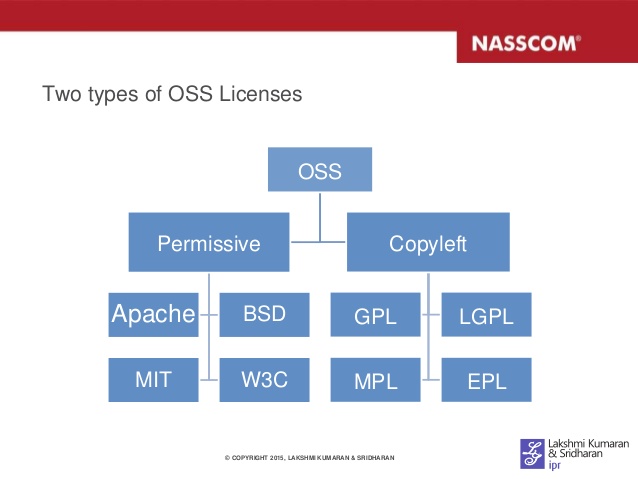
Un criterio di selezione non trascurabile del nostro screening preliminare è il tipo di Licenza Software, al fine di selezionare uno strumento cercando di minimizzare i costi di licenza software, ci orienteremo su Frameworks distribuiti con licenza opensource. Il requisito che la Licenza software deve rispettare e che sia una licenza FOSS, ovvero riconosciuta o dalla Open Source Initiative o dalla Free Software Foundation, questo aspetto è imprescindibile dalla scelta del nostro Frameworks, questo ci permetterà oltre che al mero utilizzo del Framework anche di sviluppare ed ampliare il framework secondo le nostre necessità.

Uno strumento utile per l’individuazione delle licenze OS che sono piu calzanti al nostro bisogno è stato il libro di St. Laurent [5] e l’articolo di T. Gomez-Diaz [6]. In entrambi si fa riferimento ad un punto della definizione di Software Open Source per quanto riguarda la ri-distribuzione:

*“1. Free Redistribution The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.”* [5][6]

Una licenza OpenSource infatti non obbliga lo sviluppatore a pagare delle ‘royalty’ in caso di vendita di un prodotto software sviluppato su una base Open Source inoltre ci permetterà di accedere al codice del framework senza alcuna restrizione. La Categoria di licenze che permettono questo tipo di utilizzo è detta ‘Permissive’. Sul sito Open Source Initiative possiamo trovare una lista di licenze software Open Source riconosciute effettueremo una comparazione tra la licenza del framework e la lista di cui sopra, in casi particolari in cui la licenza non è nella lista questa verrà ricercata all’interno della lista delle licene riconosciute da Free Software Foundation altrimenti verrà valutata singolarmente (e.g. nel caso ad esempio che fosse una licenza in fase di approvazione da una delle due organizzazioni citate precedentemente).

Manca qualcosa…



**Linguaggio**

Tra i criteri di selezione per il nostro screening prleiminare sceglieremo un insieme di linguaggi di programmazione che riteniamo essere importanti per il nostro studio. Ci sono infatti linguaggi di programmazione che nel corso degli anni hanno mutato il loro utilizzo, ad esempio il PHP è un linguaggio nato con lo scopo specifico di realizzare applicazioni web Java nacque per la gestione di piccoli dispositivi il suo utilizzo per le applicazioni web avvenne solo dopo il 1993 con l’esplosione di internet, il C# un linguaggio sviluppato da Microsoft simile al C++ venne utilizzato inizialmente per applicazioni desktop ma successivamente utilizzato per lo sviluppo di ASP.NET uno dei Framework Web più utilizzati.

Il Linguaggio di programmazione determina una scelta importante, è infatti il mezzo che ci permette di scrivere il nostro codice che amplierà le funzioni fornite dal Framework, in questa sezione sceglieremo un insieme di linguaggi che saranno conformi con il criterio di selezione. Il Linguaggio deve essere utilizzato e molto diffuso nell’ambito di sviluppo di applicazioni web, abbiamo preso in considerazione due indici per la scelta dei linguaggi uno è il sondaggio annuale di Stackoverflow e l’altro è l’indice Tiobe.

Nei Successivi paragrafi parliamo dei linguaggi di programmazione scelti. Elenchiamo le caratteristiche principali. Faremo una stima dell’utilizzo di tali linguaggi attraverso una collezione di dati come il numero di Progetti scritti in quel determinato linguaggio presenti su Repository come GitHub e SourceForge, il numero di ricerche effettuate su Google, Tiobe Index (mensilmente viene stimata l’utilizzazione in base a 25 Search Engine)

**Java**

È un Linguaggio di Programmazione di alto livello orientato agli oggetti, un grande vantaggio di questo linguaggio è la sua portabilità, infatti il software non ha bisogno di essere ricompilato, ma viene virtualizzata una macchina java (Java virual machine). Il Linguaggio ha visto il su momento di maggiore crescita con la nascita di Internet dopo la prima metà degli anni ’90 dove è stato molto utilizzato per applicazioni client-server. È uno dei linguaggi più utilizzati al mondo.

Attualmente secondo Tiobe Index è considerato il Linguaggio più utilizzato (<https://www.tiobe.com/tiobe-index/> in data 29/01/2019).Effettuando una ricerca su Google con la stringa +”java programming” otteniamo ben 9.030.000 corrispondenze (in data 29/01/2019). È un Linguaggio orientato agli oggetti, e dispone di numerosi framework per lo sviluppo web e per l’utilizzo di basi di dati (Librerie ORM).

**PHP**

È il linguaggio di programmazione più longevo per quanto riguarda le applicazioni lato server, è interpretato, ha la possibilità di gestire oggetti, un linguaggio di alto livello con costrutti derivanti dal “C”.Esistono molte librerie che permettono di interfacciarsi con molti DBMS ( [MySQL](https://it.wikipedia.org/wiki/MySQL" \o "MySQL), [PostgreSQL](https://it.wikipedia.org/wiki/PostgreSQL" \o "PostgreSQL), [MariaDB](https://it.wikipedia.org/wiki/MariaDB" \o "MariaDB), [Oracle](https://it.wikipedia.org/wiki/Oracle_Database), [Firebird](https://it.wikipedia.org/wiki/Firebird_SQL" \o "Firebird SQL), [IBM DB2](https://it.wikipedia.org/wiki/DB2), [Microsoft SQL Server](https://it.wikipedia.org/wiki/Microsoft_SQL_Server) )

Attualmente secondo Tiobe Index è all’ottavo posto (<https://www.tiobe.com/tiobe-index/> in data 29/01/2019).Effettuando una ricerca su Google con la stringa +”PHP programming” otteniamo ben 1.740.000 corrispondenze (in data 29/01/2019).

**Javascript**

Javascript è un Linguaggio imprescindibile per lo sviluppo di applicazioni Web, È utilizzato maggiormente nel lato frontend (e.d. Angular,React) per rendere dinamiche ad esempio utilizzando il ben noto Ajax che nel lato backend per scrivere funzioni ed accedere a dati presenti su un database (e.d. Framework Node.js), permettendo allo sviluppatore di scrivere applicazioni completamente in Javascript con il Framework Full Stack Javascript MEAN.

Attualmente secondo Tiobe Index è al sesto posto (<https://www.tiobe.com/tiobe-index/> in data 29/01/2019).Effettuando una ricerca su Google con la stringa +”Javascript programming” otteniamo ben 1.080.000 corrispondenze (in data 29/01/2019).

**Python**

Attualmente secondo Tiobe Index è al terzo posto ed è inoltre il linguaggio che nel 2018 ha avuto una crescita di ricerche maggiore rispetto agli altri linguaggi. (<https://www.tiobe.com/tiobe-index/> in data 29/01/2019).Effettuando una ricerca su Google con la stringa +”Javascript programming” otteniamo ben 5.220.000 corrispondenze (in data 29/01/2019).

**Ruby**

È un Linguaggio di programmazione ad oggetti con una sintassi semplice ed elegante che premia la produttività, presenta inoltre caratteristiche di linguaggi imperativi e funzionali. Ha avuto una crescita significativa in seguito allo sviluppo del Framework Ruby on Rails. Effettuando una ricerca su Google con la stringa +”PHP programming” otteniamo ben 657.000 corrispondenze (in data 29/01/2019).

**C#**

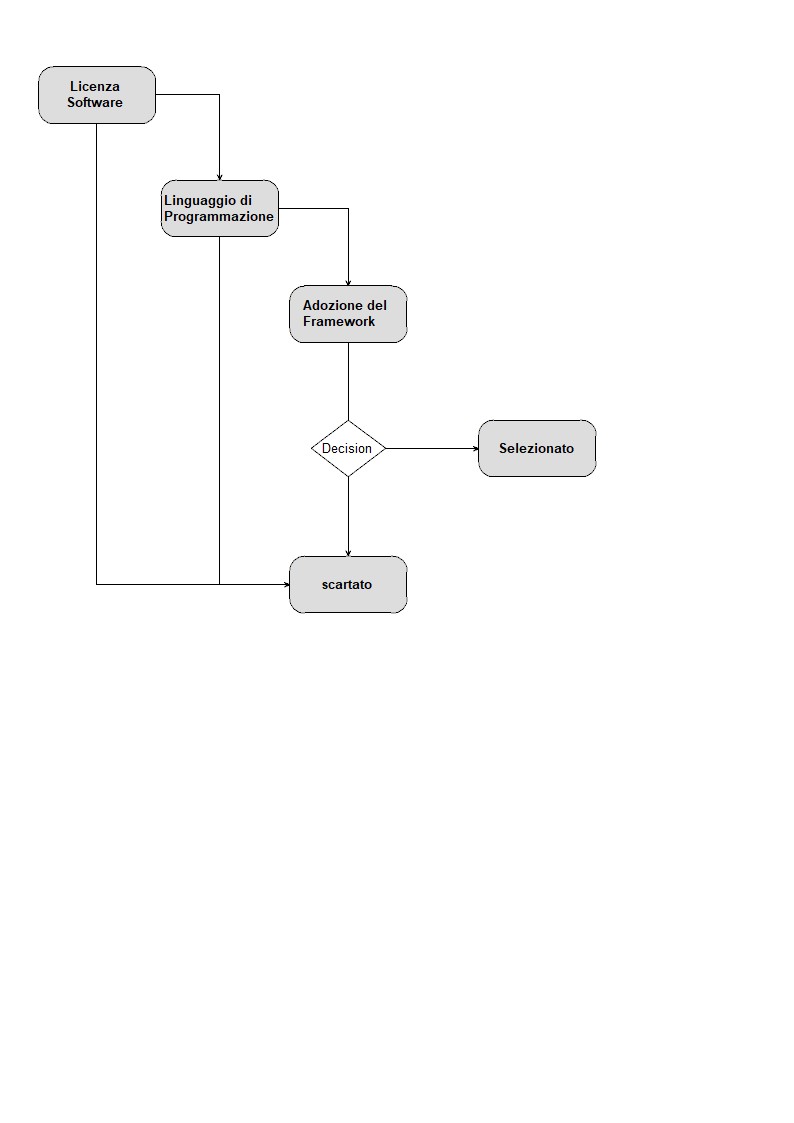
Linguaggio di programmazione Microsoft

di corrispondenza dei criteri di selezione

## Metodologia di Selezione

Questo diagramma riassume il flusso di lavoro che è stato effettuato per selezionare i framework che andranno a formare il nostro insieme Alternative. Questo diagramma a cascata è unidirezionale, la conformità del Framework rispetto ai criteri che abbiamo scelto è obbligatoria, devono essere soddisfatti tutti contemporaneamente, questo ci permetterà di scartare dalla soluzione finale i Frameworks che non sono utili al conseguimento dell’obiettivo finale. Non è importante in che ordine vengono verificati, l’importante è che vengano rispettati tutti contemporaneamente, è consigliabile utilizzare un ordine di verifica delle proprietà che ci permetterà di ridurre il tempo impiegato in questa fase di selezione, nel nostro processo abbiamo iniziato analizzando prima la conformità di criteri facili da verificare come ad esempio il tipo di Licenza e terremo per ultimi criteri come adozione e vitalità che richiedono uno sforzo maggiore per la loro verifica.

L’ordine sarà il seguente:



## Le Alternative

Nel seguente paragrafo facciamo una panoramica dei Frameworks, elencandone le caratteristiche in dettaglio, Sceglieremo i Framework che soddisfano i criteri di selezione. la conformità dei Framework con i criteri di selezione verrà effettuata seguendo le fasi del ‘Diagramma di corrispondenza dei criteri di selezione’ mostrato nel Capitolo 1. Attraverso questa verifica di Conformità dei Frameworks con i Criteri di Selezione verrà prodotta una tabella che ci permetterà di costruire il nostro spazio di possibili soluzioni ovvero l’insieme delle Alternative.

Per ogni framework preso in esame esponiamo una breve introduzione dove elenchiamo i punti di forza, le caratteristiche, la paternità, l’adozione, il linguaggio utilizzato/ambiente di sviluppo, il tutto viene posi sintetizzato in forma tabellare in modo da rendere immediata la lettura delle proprieta del framework al fine di verificarne l’appartenenza all’insieme delle Alternative.

**SPRING FRAMEWORK (JAVA, BACKEND,MVC)**

Spring MVC Framework, più brevemente “Spring MVC” è un Framework OpenSource per lo sviluppo di applicazioni web basate sul design pattern MVC scritte in Linguaggio Java. Possiamo accedere al codite tramite GitHub dove possiamo notare che l’ultima release risale al 13/02/2019 e che ci sono 13 braches di cui 7 attivi e vanta ben 348 contributors. Tra le sue caratteristiche troviamo *l’inversion of control e l’aspect oriented programming.* Spring si occupa di mappare classi e metodi con gli url e di gestire diversi tipi di Views. Spring sposa il design pattern MVC pienamente il suo funzionamento è il seguente: il Model corrisponde a classi ed oggetti che ci permettono di accedere al database, le view corrispondono a file Java Servlet Page (.jsp), il Controller è rappresentato da classi ‘Controller’ che si occupano di gestire la logica business dell’applicazione ed interagiscono con le Views ed il Model. Esso è anche un Framework che adotta il concetto Convention over Configuration che ci permette di iniziare ad utilizzarlo senza particolari settaggi iniziali, infatti ci sono delle impostazioni di default che rendono già funzionante il Framework rendendo la configurazione del WebApplicationContext molto facile. La configurazione iniziale della Servlet ‘DispatcherServlet’ viene fatta attraverso un file XML (web.xml), dove vengono inseriti pochi ed essenziali parametri e direttive J2EE. Il Framework si avvale dell’utilizzo di Maven per l’accesso alle risorse esterne ed al download il che facilita ulteriormente l’installazione di librerie esterne ed aggiornamenti. Per ogni Servlet vengono instanziati dei WebApplicationContext che si avvalgono allo standard ServletContext ed all’interno troviamo dei Bean che configurano il nostro Spring come l’HandlerMapping, l’HandlerExceptionResolver e ViewResolver questi Bean sono sovrascrivibili con dei Bean personalizzati dallo sviluppatore.

Utilizzato da: Wix,TicketMaster,Billguard

Con Spring Abbiamo tutto quello che ci serve per sviluppare facilmente applicazioni in Java.

In Sintesi:

* Linguaggio: Java
* Licenza: OpenSource
* Tipo: Backend/Fullstack
* Integrazioni: AngularJS,Bootstrap,Spring Boot,Hibernate, Spring Security, Spring Data. (6 tools fonte www.stackshare.io)
* Adozione: Buona Samsung Microservice (fonte www.stackshare.io)
* Vitalità: attualmente disponibile nella versione.
* Statistiche Git Hub: 29.9K Stars 19.1K Forks Last Commit 15/06/19 giorno (dati del 15/06/19)

|  |  |
| --- | --- |
| Linguaggio | Java |
| Licenza | OpenSource |
| Tipo | Backend/Fullstack |
| Integrazioni | … |
| Vitalità | … |
| Stars | … |
|  |  |
| Forks | … |
| Last commit | … |

**Laravel (PVP, MVC, FS)**

Laravel è un Framework opensource scritto in linguaggio PHP orientato agli oggetti che ci permette di sviluppare applicazioni web con il design pattern MVC, è un framework maturo, nato nel 2011 è attualmente uno dei Framework PHP più diffusi, il codice è accessibile su GitHub dove possiamo trovare la sua ultima versione attualmente la sua ultima release è stata fatta il 12/03/2019 all’interno di GitHub possiamo trovare ben 13 branches 5 dei quali attivi, più di 390 release e 1920 contributors (accesso il 17/03/2019). Laravel utilizza il Facade Pattern per rendere elegante e leggibile il codice, vengono infatti create delle API attraverso l’uso di interfacce. Laravel ci mette a disposizione i Controller, che sono classi con metodi pubblici, tra queste possiamo trovare anche classi per l’implementazione di controller RESTful nel caso di utilizzo di API RESTful. Ha dei meccanismi di caching del routing che ci permettono di velocizzare la nostra applicazione e di richiesta dei cookie dall’oggetto request.

In sintesi:

* Linguaggio: PHP
* Licenza: OpenSource
* Tipo: Fullstack/Backend
* Integrazione: PHP, Sentry, Auth, Bugsnag, Laravel SPark ed altri (19 tools fonte www.stackshare.io)
* Adozione: Discreta 9GAG, Union, Webedia, Sellsuki, Boot Ventures, GeoCodio. (fonte www.stackshare.io)
* Vitalità: il Framework risulta molto utilizzato ed aggiornato.

Statistiche Git Hub: 52.9K Stars 16.2K Forks Last Commit 15/06/19 dati del 15/06/19

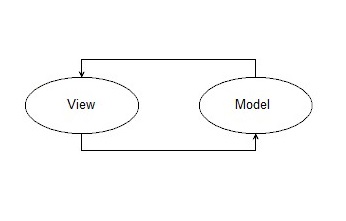
Verdetto: Selezionato

**AngularJS (Fe,JS, Mean)**

AngularJs è un framework frontend open-source sviluppato da Google nel 2010 scritto in Javascript, nato per agevolare lo sviluppo delle applicazioni a pagina singola, fornisce gli strumenti per la separazione dei livelli ma allo stesso tempo il design pattern MVC non è vincolante, Angularjs è infatti molto flessibile, può essere visto come un framework Model View Whatever dove whatever significa ‘ciò che vuoi’ è stato anche definito dai suoi autori Model View-View Model (MVVM). Possiamo accedere al codice su GitHub dove possiamo scaricare l’ultima release che attualmente è la v1.7.8 rilasciata in data 11/03/2019 ha 16 branches di cui 1 attivo un totale di 204 release e 1596 contributors.

Il ModelView-ViewModel ci permette di interagire con il Model di dati direttamente dalla View e viceversa, questo meccanismo è chiamato Two-Way Binding.

**Fig 4: Model View**



Il Flusso di lavoro di AngularJS è il seguente: legge il file HTML che incapsula dei token riconosciuti da AngularJS come comandi, questi vengono legati al modello rappresentato da variabili standard Javascript queste variabili possono essere cablate nel codice o accessibili come risorse JSON statiche o dinamiche.

In particolare una View sarà un Template HTML con al suo interno direttive AngularJS. i Controller sono applicazioni scritte in JavaScript che definiscono l’interazione tra la View ed il Model, essi sono definiti in due passi,con il metodo module() dell’oggetto globale Angular definiamo un modulo che a sua volta contiene i controller che vengono definiti con il metodo controller().

Essendo un framework frontend per un completo sviluppo di un applicativo client-server dobbiamo ricorrere a delle Librerie esterne per la realizzazione del lato server, per l’interazione del frontend con il backend possiamo utilizzare lo standard JSON e quindi avere un frontend indipendente dalle tecnologie usate per lo sviluppo del backend che esse siano scritte in PHP,Java o in Javascript (e.g. Node.js)

In Sintesi:

* Linguaggio: JavaScript
* Licenza: OpenSource
* Tipo: Frontend
* Integrazioni Spring MVC, MEAN, Bugsnag Auth0, Stencil e molti altri (39 tools fonte www.stackshare.io)
* Adozione: Molto Buona Google, Udemy, MIT e molti altri
* Vitalità: il Framework risulta molto utilizzato ed aggiornato data ultima release 11/03/2019
* Statistiche Git Hub: 59.6K Stars 28.9K Forks Last Commit 15/06/19 giorno (dati del 15/06/19) Code Frequency Buona

**Verdetto: Selezionato come Alternativa**

**Node.Js**

È un framework Open-source per lo sviluppo di applicazioni web scritto in Javascript accessibile su GitHub attualmente è disponibile la versione 11.12.0 rilasciata il 15/03/2019 vi sono ben 142 branches molti dei quali attivi e ben 2417 contributors. È probabilmente uno dei framework di sviluppo web più innovativi degli ultimi anni in quanto fornisce gli strumenti necessari per sviluppare funzionalità server-side in JavaScript, questo permette di sviluppare un applicativo client-server completamente in JavaScript. È un Framework relativamente giovane supportato da una grande community, è facile da utilizzare ed è molto indicato per lo sviluppo di applicazioni Mobile.

Node.js è basato sul Javascript Engine V8 il motore Runtime utilizzato da Google anche in Chrome utilizza un approccio asincrono event-driven, questo è un grande vantaggio in termini di efficienza in quanto il nostro applicativo non si ferma ad attendere una risposta come nel caso di un approccio sincrono, questa caratteristica si rispecchia anche nella scrittura del codice dove si farà affidamento alle *callback* questo lo rende molto efficiente per le situazioni con carichi di traffico elevato.

Node.js dispone di tutte le API necessarie alla gestione/interazione di database, http, livello di rete, soket, SSL, DNS, incluse le api CommonJS che sono uno standard per gli interpreti di comandi JavaScript e la possibilità di creare applicazioni con il design pattern MVC. Nativamente vi è la possibilità di accedere attraverso il registro NPM ad un repository che rende disponibili tutte le librerie sviluppate appositamente per Node.js. Tra queste troviamo un dispatcher singleton per la gestione del traffico, …

Le funzioni in Node.js si trovano all’interno di moduli che è possibile richiamare con la funzione ‘require’. Questi possono essere moduli ‘core’ di Node.js o moduli definiti dall’utente, in questo caso la funzione cerca ricorsivamente il modulo nella cartella padre fino a trovare il modulo richiesto, questo sistema ci permette di condividere i moduli tra diverse applicazioni residenti nella stessa root.

In sintesi:

* Linguaggio: JavaScript
* Licenza: OpenSource
* Tipo: Backend/MEAN
* Integrazioni MEAN, Bugsnag Auth0 e molti altri (115 tools fonte www.stackshare.io)
* Adozione: Molto Buona MIT, Ebay, Netflix, Twitter.
* Vitalità: il Framework risulta molto utilizzato ed aggiornato attualmente disponibile nella versione 11.12.0 rilasciata il 15/03/2019
* Statistiche Git Hub: 62.1K Stars 13.9K Forks Last Commit 15/06/19 giorno (dati del 15/06/19) Code Frequency Molto Alta

**Verdetto: Selezionato come Alternativa**

**Google Web Toolkit**

Google Web Toolkit è un Framework scritto in Java che ci permette di produrre applicazioni web Ajax. GWT è nato in un periodo in cui il linguaggio JavaScript non era ancora ben strutturato questo creava problemi di manutenibilità del codice causato dalla non modularità del linguaggio stesso, ma allo stesso tempo questo linguaggio ci permetteva di ottimizzare applicazioni web con chiamate asincrone grazie all’utilizzo di Ajax, per fare fronte ai problemi ma allo stesso tempo usufruire dei vantaggi di JavaScript è stato sviluppato questo Framework che ci permette di scrivere codice in linguaggio Java tradotto successivamente in codice Javascript ed HTML. Questo permette allo sviluppatore di poter sviluppare l’applicazione all’interno di un IDE, di effettuare debug, test, integrare, librerie esterne, compilare il codice e produrre file HTML e Javascript. Questa soluzione è stata rivoluzionaria per il modo in cui si poteva sviluppare un’applicazione web. Tra le caratteristiche principali ed uniche di GWT troviamo: un compilatore di codice da java a JavaScript, un’emulazione JRE scritto in Javascript, le API GWT ed un’interfaccia per definire metodi nativi in JavaScript la JSNI. Il Compilatore è il cuore del Framework perché ci permette di sviluppare un’applicazione interamente in Java e produrre allo stesso tempo codice HTML e Javascript. Dopo l’uscita di Framework scritti Javascript come AngularJS il Google Web Toolkit è stato via via abbandonato, attualmente infatti l’ultima release risale al 18/10/2019, questo purtroppo non ci permetterà di selezionare il GWT perché non rispetta uno dei criteri di selezione che abbiamo definito nel capitolo precedente.

In sintesi:

* Linguaggio: Java
* Tipo Frontend
* Licenza: OpenSource
* Adozione: Sostituito Google Inbox ed altri.
* Vitalità: Ultima Release 23 gennaio 2014 criterio non rispettato
* Statistiche Git Hub: 1.1K Stars 271 Forks Last Commit 25/03/19 giorno (dati del 10/06/19) Code Frequency molto bassa

**Verdetto: Rigettato**

**Vaadin (Java)**

Vaadin è un Framework OpenSource scritto in Java per lo sviluppo di interfacce utente nato con lo scopo di semplificare lo sviluppo di applicazioni Web fornendo strumenti di altro livello senza essere costretti a gestire tecnologie web di basso livello. Il concetto cardine del framework è scrivere codice Java e fare un render della UI in HTML sul browser, Vaadin automatizza inoltre la comunicazione tra il client ed il server. Vaadin è stato sviluppato per essere compatibile con Java, Spring, Kotlin e Java EE. Vaadin puo’ essere installato su Eclipse uno dei migliori IDE per lo sviluppo di Software scritti in Java. Possiamo accedere al sorgente del Framework su GitHub dove possiamo verificare che il progetto è attualmente alla versione 7.7.17 e che l’ultima release è recente, risale infatti al 15/03/2019 inoltre sulla repository notiamo che vi sono ben 68 Branches molti dei quali attivi, un dato che ci permette di affermare che il Framework non è in disuso. La Licenza è una Apache 2.0 una licenza OpenSource ben nota. Nella pagina ufficiale di Vaadin possiamo anche accedere ad una lista di aziende che adottano questo Framework, tra quelle di spicco possiamo citare: VolksWagen, Motorola, Bank of America, GSK e molti altri progetti di

interesse come un Firewall sviluppato da UPS con Vaadin.

In sintesi:

• Linguaggio: Java

• Licenza: OpenSource Apache 2.0

• Tipo: Frontend

• Adozione: Buona

• Vitalità: il Framework risulta molto utilizzato ed aggiornato attualmente disponibile nella versione 7.7.17 rilasciata in data 15/03/2019

**Spark**

È un Framework minimale con licenza Opensource scritto in Java nato per lo sviluppo di applicazioni web Java e Kotlin. Spark punta ad incrementare la produttività ma allo stesso tempo a mantenere un codice pulito grazie alla sua sintassi con espressioni dichiarative.

In sintesi:

Linguaggio: Java

Licenza: OpenSource

Tipo: Backend

Integrazioni MEAN, Bugsnag Auth0 e molti altri (115 tools fonte www.stackshare.io)

Adozione: Sufficiente Apache Software Foundation,

Vitalità: attualmente disponibile nella versione 11.12.0 rilasciata il 15/03/2019

Statistiche Git Hub: 8.3KStars 1.5K Forks Last Commit 24/05/19 giorno (dati del 15/06/19) Code Frequency Bassa

Verdetto: Rigettato???

**ASP.NET**

ASP.NET è un Framework OpenSource per applicazioni web sviluppato da Microsoft , nello specifico ASP.NET è formato da 3 Framework; ASP.NET MVC, Web Form e ASP.NET Web Pages, tutti e tre utilizzano il .NET Framework, ognuno di essi è destinato ad uno stile di sviluppo differente, ma la scelta di uno non preclude quella di un altro perlo sviluppo di una sola applicazione qeb, si possono infatti utilizzare contemporaneamente per sviluppare parti diverse della stessa applicazione.

* Web Form: Orientato ad un approccio visuale Drag & Drop.
* MVC: si avvale del design pattern MVC permette di separare le competenze per sviluppare applicazioni web complesse seguendo gli standard piu recenti.
* Web Pages: Permette di sviluppare applicazioni dinamiche con un approccio classico, combinando il Markup Language HTMLed il codice al livello server side.

La nostra attenzione ovviamente ricade sul Framework ASP.NET MVC che ci permette di sviluppare seguendo il design pattern MVC.

Attualmente il Framework ASP.NET MVC è reperibile attraverso la repository di GitHub dove possiamo prendere nota di dati come il numero di releases 34 l’ultima risale al 29/11/2018 il numero di contributors attualmente 65 e la licenza software opensource Apache 2.0.

Per quanto riguarda il linguaggio di programmazione supportato ASP.NET si appoggia al framework .NET il che ci permette di utilizzare diversi linguaggi di alto livello come Visual Basic, C#,J#,Perl e Python.

Un aspetto importante ed allo stesso tempo rassicurante di questo Framework è che è il principale Framework di sviluppo web sostenuto e sviluppato da Microsoft, è inoltre consolidato e maturo ed adottato da moltissime aziende in tutto il mondo tra queste possiamo sottolineare l’adozione di ASP.NET da parte di Siemens, UPS, StackOverflow, Alaska Airlines, Fujifilm, Sony, Mitsubishi Hitachi Power System MHPS, T-Systems, LG CNS ed ovviamente anche da Microsoft.

In sintesi:

* Linguaggio: Multi-Language, C#
* Licenza: OpenSource
* Tipo: Backend/fullstack
* Integrazioni (fonte www.stackshare.io)
* Adozione:
* Vitalità
* Statistiche Git Hub: Code Frequency

**Verdetto: Selezionato come Alternativa**

**Javalin**

È un Framework opensource semplice e leggero che ci permette di sviluppare applicazioni web in Java e Kotlin, è stato sviluppato tenendo a mente l’interoperabilità tra i due linguaggi infatti vi è un’unica versione che supporta entrambe le tecnologie. Javalin è nato come fork del Framework SparkJava ma ha subito un’influenza dal Framework KOA.Js è così diventato un progetto autonomo e diverso da SparkJava. Javalin gira sul server HTTP JETTY, uno dei server più stabili ed utilizzati per JVM, ci fornisce quindi tutte le Features offerte da jetty.

Per quanto riguarda l’adozione di Javalin possiamo citare Microsoft, RedHat, Jhon Hopkins University, Norwegia Universityof Science and Technology e molti altri.

Sulla pagina ufficile del Framework <https://javalin.io> possiamo trovare Documentazione e Lessons per apprenderne l’utilizzo. Molto interessante anche un Sondaggio messo a disposizione riguardante l’utilizzo del Framework dove possiamo notare che il tipo di utente che utilizza maggiormente il Framework è lo sviluppatore 68% di RESTAPI , WebSoket e Website.

Il Framework è scaricabile attraverso GitHub dove possiamo anche accedere ad alcuni dati rilevanti come la data dell’ultima Release attualmente rilasciata il 25/03/2019, con una comunity di 53 contributors ed 11 Branches, il Software è rilasciato con la ben nota licenza opensource Apache 2.0.

***Like Sinatra, Javalin is not aiming to be a full web framework,*** *but rather just a lightweight REST API library (or a micro framework, if you must). There is no concept of MVC, but there is support for template engines, WebSockets, and static file serving for convenience. This allows you to use Javalin for both creating your RESTful API backend, as well as serving an index.html with static resources (in case you’re creating an SPA). This is practical if you don’t want to deploy an apache or nginx server in addition to your Javalin service. If you wish to use Javalin to create a more traditional website instead of a REST APIs, there are several template engine wrappers available for a quick and easy setup. [cit* <https://javalin.io/about>*]*

In sintesi:

* Linguaggio: Java, Kotlin
* Licenza: OpenSource Apache 2.0
* Tipo: Backend/minimale
* Integrazioni: JAVA,Kotlin (fonte www.stackshare.io)
* Adozione: Buona
* Vitalità: il Framework risulta utilizzato ed aggiornato attualmente disponibile nella versione 2.8.0 rilasciata in data 25/03/2019
* Statistiche Git Hub: 2.9KStars 250K Forks Last Commit 17/06/19 giorno (dati del 17/06/19) Code Frequency Bassa

**Polymer (HTML,Js)**

Polymer è un Framework OpenSource per lo sviluppo di applicazioni web Frontend, l’obiettivo del progetto Polymer è di migliorare il Web esistente, è sviluppato dal team di Chrome (Google), tra le features possiamo citare: includin shadow DOM, Custom elements e Model Driven Views. Il progetto attualmente è molto attivo su GitHub, siamo attualmente alla versione 3.2.0 rilasciata il 20/03/2019 inoltre possiamo sottolineare il numero di branches e contributors rispettivamente 64 e 141, il software è rilasciato con la licenza OpenSource BSD. Un tasto dolente riguardante questo Framework è la documentazione che risulta frammentata tra le diverse pagine del sito del progetto, non immediato capire dove e come reperire la documentazione del Software.

Per quanto riguarda l’adozione del Framework risulta utilizzato da:  [Netflix](https://en.wikipedia.org/wiki/Netflix" \o "Netflix), [Electronics Arts](https://en.wikipedia.org/wiki/Electronic_Arts" \o "Electronic Arts), [ING](https://en.wikipedia.org/wiki/ING_Group), [Coca-Cola](https://en.wikipedia.org/wiki/Coca-Cola), [McDonald's](https://en.wikipedia.org/wiki/McDonald%27s), [BBVA](https://en.wikipedia.org/wiki/BBVA), [IBM](https://en.wikipedia.org/wiki/IBM) and [General Electric](https://en.wikipedia.org/wiki/General_Electric).

In sintesi:

* Linguaggio: Javascript,HTML
* Licenza: OpenSource
* Tipo: Frontend
* Integrazioni: (fonte www.stackshare.io)
* Adozione: Buona, Bloomberg, Google Cloud, :  [Netflix](https://en.wikipedia.org/wiki/Netflix" \o "Netflix), [Electronics Arts](https://en.wikipedia.org/wiki/Electronic_Arts" \o "Electronic Arts), [ING](https://en.wikipedia.org/wiki/ING_Group" \o "ING Group), [Coca-Cola](https://en.wikipedia.org/wiki/Coca-Cola" \o "Coca-Cola), [McDonald's](https://en.wikipedia.org/wiki/McDonald%27s" \o "McDonald's), [BBVA](https://en.wikipedia.org/wiki/BBVA" \o "BBVA), [IBM](https://en.wikipedia.org/wiki/IBM" \o "IBM) and [General Electric](https://en.wikipedia.org/wiki/General_Electric" \o "General Electric).
* Vitalità: il Framework risulta utilizzato ed aggiornato attualmente disponibile nella versione 1.12.0 rilasciata il 19/05/19
* Statistiche Git Hub: 21.1KStars 2K Forks Last Commit 17/06/19 giorno (dati del 17/06/19) Code Frequency Buona

**Verdetto: in fase di valutazione**

**React.JS (Javascript Mantained by Facebook Huge Community)**

React è un Framework Opensource scritto in Javascript per lo sviluppo Frontend. React ci permette di sviluppare la componente View del modello MVC, è stata la prima libreria nata all’interno di Facebook, nasce con lo scopo di permettere di sviluppare in modo semplice applicazioni web complesse con design single application page.

Il codice è accessibile su Github dove possiamo subito notare i numeri a favore del Framework come gli oltre 1000 contributors le 116 release attualmente il Framework è alla versione stabile 16.8.6 rilasciata il 27/03/2019. Sul sito ufficiale del Framework possiamo reperire documentazione e tutorial. Il Framework è rilasciato con licenza OpenSource MIT.

Per quanto riguarda l’adozione possiamo citare Instagram e Facebook che sono attualmente i due Social network più utilizzati al mondo.

In sintesi:

* Linguaggio: Js, Html
* Licenza: OpenSource MIT
* Tipo: Frontend/MEAN
* Integrazioni: Aut0, Bugsnag , Firebase, Stencil (tot 131 tool fonte www.stackshare.io)
* Adozione: Instagram, Facebook, Yahoo, Udemy, Reddit, MIT e molte altre. (fonte www.stackshare.io)
* Vitalità: il Framework risulta utilizzato ed aggiornato attualmente disponibile nella versione 16.9.0 alpha rilasciata il 03/04/2019 (accesso github 17/06/19)
* Statistiche Git Hub: 131.1KStars 24.2K Forks Last Commit 17/06/19 giorno (dati del 17/06/19) Code Frequency Bassa

**Sails.JS (JavaScript)**

È un Framework OpenSource per lo sviluppo in Javascript di applicazioni web. Sails è un Framework MVC in un ambiente Node.JS creato per soddisfare i requisiti delle moderne applicazioni data-driven, scalabile con un’architettura service oriented. Tra i punti di forza di Sails.js possiamo citare: la possibilità di sviluppare un applicazione interamente in JavaScript,la sua versatilità nelle connessioni a diversi tipi di database, un set di progetti (blueprints) che ci permettono di utilizzare codice Backend pronto all’uso, indipendenza dal framework frontend, integrazione web socket, supporto commerciale fornito da Sails ed una buona documentazione reperibile sul sito ufficiale del progetto. Tra le cose da citare troviamo anche la compatibilità con altri Framework Frontend conosciuti come: Angular,React, iOS,Android,Windows Phone.

Il Framework è reperibile su GitHub dove notiamo subito numeri importanti come i 55 Branches, 211 contributors e la data dell’ultima release della versione 1.2.2 del 16/05/2019.

È adottato da grandi e piccole aziende, tra quelle di maggiore spicco troviamo: Philips, Microsoft,DellEMC e molte altre.

In sintesi:

* Linguaggio: Javascript
* Licenza: OpenSource MIT
* Tipo: Backend Minimale
* Integrazioni: Node.JS, MySql,PstgreSql, MongoDB, Soket.IO. (tot 9 tool fonte www.stackshare.io)
* Adozione: Buona Philips, Microsoft,DellEMC,TestLegends, HYPH, and Giguana (fonte www.stackshare.io)
* Vitalità: attualmente disponibile nella versione 1.2.2 rilasciata il 16/05/2019 (accesso github 17/06/19)
* Statistiche Git Hub: 20.6KStars, 1.8K Forks, Last Commit 04/06/19 (dati del 17/06/19), Code Frequency Media.

**Koa.Js**

Koa è un Framework minimale backend Opensource per lo sviluppo di applicazioni web in ambiente Node.JS, sviluppato dal team di sviluppo di Express.

Linguaggio: Javascript

Licenza: OpenSource

Tipo: Backend Minimale

Integrazioni: Node.JS, Sentry, Postgraphile (tot 3 tool fonte www.stackshare.io)

Adozione: Hostr, Dial Once, Pubu (fonte www.stackshare.io)

Vitalità: attualmente disponibile nella versione 2.7.0 rilasciata il 28/01/2019 (accesso github 17/06/19)

Statistiche Git Hub: 26.4KStars, 2.4K Forks, Last Commit 10/06/19 (dati del 17/06/19), Code Frequency Media.

POCHE INFORMAZIONI REPERIBILI

**Express (Javascript)**

Express è un framework scritto in Javascript basato su Node.js ci fornisce infatti Features aggiuntive a quelle offerte da Node.Js. È utilizzato da molte compagnie quali Accenture, IBM e Uber. È un Framework minimale e non definisce un design di sviluppo. Vi sono molti Frameworks per lo sviluppo di applicazioni web che si basano sull’ambiente fornito da express, tra questi possiamo certamente citare Sails.js. Express fa parte di Node.Js Foundation, un progetto per lo sviluppo di API web supportato dalla stessa IBM.

Il codice è reperibile su GitHub attualmente disponibile nella versione 4.17.1 rilasciata il 26/05/19 , attualmente è accessibile anche la versione pre-release 5.0 alpha.

In sintesi:

• Linguaggio: Javascript

• Licenza: OpenSource

• Tipo: Backend Minimale

• Integrazioni: Node.JS, MEAN, Bugsnag, Postgraphile (tot 19 tool fonte www.stackshare.io)

• Adozione:Molto Buona,Twitter, Accenture, IBM e Uber (fonte www.stackshare.io)

• Vitalità attualmente disponibile nella versione 4.17.1 rilasciata il 26/05/19 (accesso github 17/06/19)

• Statistiche Git Hub: 44.3KStars, 7.4 K Forks, Last Commit 16/05/19 (dati del 18/06/19), Code Frequency Media.

**Ember(Js,frontend)**

Ember è un Framework OpenSource per lo sviluppo di applicazioni web frontend scritto in Javascript. Il Framework è orientato alla produttività ma ci permette allo stesso tempo di sviluppare applicazioni rispettando il pattern MVC (MVVM). Il codice è reperibile su GitHub dove troviamo 69 branches 759 contributors attualmente disponibile nella versione nella versione 3.10.1 rilasciata il 10/06/19.

In sintesi:

• Linguaggio: Javascript

• Licenza: OpenSource MIT

• Tipo: Frontend MVC

• Integrazioni: Auth0, Bugsnag, Airbrake, Stencil (tot 19 tool fonte www.stackshare.io)

• Adozione:Ottima, Twitch, Microsoft,Google,Netflix,linkedin,zendesk Uber (fonte www.stackshare.io)

• Vitalità attualmente disponibile nella versione 3.10.1 rilasciata il 10/06/19 (accesso github 18/06/19)

• Statistiche Git Hub: 21KStars, 4.2K Forks, Last Commit 18/06/19 (dati del 18/06/19), Code Frequency Alta.

**Symfony (PHP, Backend)**

Symphony è un Framework OpenSource per lo sviluppo di applicazioni web lato Backend in ambiente PHP con design pattern MVC. Symphony può essere visto non solo come un Framework ma come un insieme di componenti riutilizzabili per risolvere problemi comuni, le singole componenti possono essere utilizzate in modo indipendente, infatti all’interno di altri Framework come Laravel o Drupal troviamo componenti di Symphony. Il Framework è provvisto di tutto quello che ci serve per sviluppare il lato backend fornisce tutte le funzionalità di PHP come la programmazione agli oggetti e il testing di PHPUnit lo sviluppatore ha piena liberta durante lo sviluppo e la logica del pattern MVC deve essere sviluppata dal programmatore che può avvalersi di strumenti che implementano ORM come Doctrine. Il sito web ufficiale mette a disposizione sia un’ottima documentazione che esempi e referenze, Symphony non ha bisogno di presentazioni vista la sua popolarità e diffusione.

Il Codice del Framework è reperibile su GitHub dove possiamo verificare e confrontare alcuni dati forniti dal sito ufficiale come Contributors 1800 430 release attualmente disponibile nella versione 4.3.1 rilasciata il 06/06/19.

In sintesi:

• Linguaggio: PHP

• Licenza: OpenSource MIT

• Tipo: Full Stack/Backend MVC

• Integrazioni:Bugsnag, Airbrake,PHP, KeyCDN, Sentry (tot 9 tool fonte www.stackshare.io)

• Adozione: Molto Buona, Accenture, Sky Vegas,Business Insider, (fonte www.stackshare.io)

• Vitalità attualmente disponibile nella versione 4.3.1 rilasciata il 06/06/19.19 (accesso github 18/06/19)

• Statistiche Git Hub: 21KStars, 7K Forks, Last Commit 18/06/19 (dati del 18/06/19), Code Frequency Molto Buona.

**Cake PHP (PHP, Backend)**

CakePHP è un Framework OpenSource per lo sviluppo di applicazioni web lato Backend in ambiente PHP. Tra le caratteristiche possiamo citare: Generazione di Codice, la possibilità di non configurare il Framework (convention over configuration), gran numero di componenti Built-in come traduttore, caching, accesso al database, design pattern MVC, gestione della sicurezza implementata dal framework, come la prevenzione di SQL injection,XSS, e validazione delle form. Cake offre allo sviluppatore la documentazione i tutorial ed anche un servizio premium customer support. Tra le aziende che adottano Cake possiamo citare BMW, Billabong.

Il Framework è scaricabile da GitHub attualmente è disponibile la versione 3.7.8 rilasciata il 28//05/2019.

• Adozione: Molto Buona

• Vitalità: il Framework risulta utilizzato ed aggiornato attualmente disponibile nella versione 3.7.7 del 24/04/19

In sintesi:

• Linguaggio: PHP

• Licenza: OpenSource MIT

• Tipo: Full Stack/Backend MVC

• Integrazioni: PHP, KeyCDN, Raygun, Deployer (tot 4 tool fonte www.stackshare.io)

• Adozione: Buona, BMW, Billabong (fonte www.stackshare.io)

• Vitalità attualmente disponibile nella versione 3.7.8 rilasciata il 28//05/2019 (accesso github 18/06/19)

• Statistiche Git Hub: 5.7KStars, 2.9K Forks, Last Commit 17/06/19 (dati del 18/06/19), Code Frequency Bassa.

**Zend Framework** (PHP Backend/FullStack)

Zend è un Framework Opensource Full Stack per lo sviluppo di applicazioni web in ambiente PHP, con la versione Zend Framework 2 è possibile sviluppare applicazioni in codice PHP Object Orientedsfruttando tutte le features offerte da PHP 5.3, namely,namespace,late state binding, lambda function e closure.

In sintesi:

• Linguaggio: PHP

• Licenza: OpenSource MIT

• Tipo: Full Stack/Backend MVC

• Integrazioni: PHP, KeyCDN, SlickStack, Deployer (tot 4 tool fonte www.stackshare.io)

• Adozione: Bassa, Webedia, Scandlearn, MV Labs (fonte www.stackshare.io)

• Vitalità: attualmente disponibile nella versione 2.4.13 del 13/07/17 (accesso github 18/06/19)

• Statistiche Git Hub: 7.9KStars, 3.4K Forks, Last Commit 22/05/19 ma risulta un vuoto nei commit dal 17/02/17 (dati del 18/06/19), Code Frequency Molto Bassa.

Verdetto: Rigettato

Django (Python, Backend)

Django è un Framework FullStack per lo sviluppo di applicazioni web OpenSource scritto in Python. Django è un Framework di alto livello orientato allo sviluppo veloce ed allo stesso tempo pulito, tra le sue caratteristiche possiamo citare la possibilità di sviluppare applicazioni MVC, la facilità di apprendimento del framework, lo sviluppo rapido,

In Sintesi:

• Linguaggio: Python

• Licenza: OpenSource

• Tipo: Backend

• Adozione: Buona Utilizzato da Google,Youtube,Instagram,Pinterest,Accenture e molti altri.lar

• Vitalità: attualmente disponibile nella versione 2.2 del 1/04/19

• Statistiche Git Hub: 42.2K Stars 18.1K Forks Last Commit 1 giorno dati del 15/06/19

**Rails (Ruby, Backend,MVC)**

Rails è un Framework opensource Fullstack/Backend per lo svipuppo di applicazioni web MVC in ambiente Ruby. Tra i punti di forza di Rails possiamo citare: Convention over Configuration, la rapidità di sviluppo, la grande community, DB Migration, la facilità di apprendimento e l’eleganza del codice scritto in Ruby.

In sintesi:

• Linguaggio: Ruby

• Licenza: OpenSource MIT

• Tipo: Full Stack/Backend MVC

• Integrazioni: Ruby, Sentry, Haml, Raygun, Bugsnag (tot 40 tool fonte www.stackshare.io)

• Adozione: Ottima, Twitch, Kickstarter, Airbnb, Groupon ed altre 2300 aziende hanno dichiarato di usare Rails (fonte www.stackshare.io)

• Vitalità: attualmente disponibile nella versione 5.2.3 del 24/04/19

• Statistiche Git Hub: 43.4KStars, 17.4K Forks, Last Commit 18/06/19 (dati del 18/06/19), Code Frequency Buona.

**Vue.JS (Javascritp,Frontend)**

Vue è un Framework OpenSource per lo sviluppo di applicazioni web frontend scritto in Javascript. È Orientato allo sviluppo di interfacce web e single page application. Può essere considerato piu come una libreria per le interfacce utente che un Framework ma è comunque utilizzato moltissima ed ha un rating altissimo su Github.

In sintesi:

• Linguaggio: Javascript

• Licenza: OpenSource

• Tipo: Frontend/Interfacce

• Integrazioni: Sentry, Bugsnag, VuePress, Stencil (tot 40 tool fonte www.stackshare.io)

• Adozione: Ottima, 9GAG, Sellsuki, Google Cloud 813 aziende hanno dichiarato di usare Vue (fonte www.stackshare.io)

• Vitalità: attualmente disponibile nella versione 2.6.10 del 20/03/19

• Statistiche Git Hub: 141.6KStars, 20.4K Forks, Last Commit 18/06/19 (dati del 18/06/19), Code Frequency Buona.

**Backbone (Javascritp,Frontend,MVC)**

Backbone.js è un framework Opensource per lo sviluppo di applicazioni web in ambiente Javascript, è uno strumento che ci permette di sviluppare il lato frontend dell’applicazione web in linea con il pattern MVC. Il codice è reperibile su GitHub dove troviamo 4 branches 290 contributors, è attualmente disponibile nella versione 1.4.0 rilasciata il 19/02/19.

In Sintesi:

• Linguaggio: JavaScript

• Licenza: OpenSource MIT

• Tipo: Frontend

• Adozione: Buona Wordpress, Vox, Walmart mobile, Airbnb.

• Vitalità: attualmente disponibile nella versione 1.4.0 rilasciata il 19/02/19.

Frameworks in Draft

Code Igniter

è un framework Opensource per lo sviluppo di applicazioni web lato Backend/FullStack in ambiente PHP Il codice è reperibile su GitHub, è attualmente disponibile nella versione rilasciata il.

In Sintesi:

• Linguaggio: PHP

• Licenza: OpenSource

• Tipo: Backend/Full Stack

• Integrazioni

• Adozione: Buona Sufficiente Buffer,Webedia,Accenture.

• Vitalità: attualmente disponibile nella versione.

• Git Hub Stats: 17.4 K Stars 7.7K Forks Last Commit 1 giorno (dati del 15/06/19)

**Flask**

Flask è un framework Opensource minimale per lo sviluppo di applicazioni web lato Backend scritto in Python, il Framework è dipendente da due librerie Jinja un template engine e Werkzeug WSGI Toolkit la documentazione ed i tutorial sono facilmente reperibili sul sito ufficiale di Flask. Il codice è reperibile su GitHub, attualmente disponibile nella versione 1.0.2 rilasciata il 02/05/2019

In Sintesi:

• Linguaggio: Python

• Licenza: OpenSource

• Tipo: Backend Minimale

• Adozione: Buona Netflix,Mit e molti altri

• Vitalità: attualmente disponibile nella versione 1.0.2 rilasciata il 02/05/2019.

• Statistiche Git Hub: 44.7K Stars 12.6K Forks Last Commit 15/16/19 (dati del 15/06/19), Code Frequency Buona.

**MEAN (Javascript FullStack)**

MEAN non è un Framework ma un Bundle di Framework che implementa lo sviluppo Full Stack in un ambiente Javascript, è stato uno dei punti di forza dello sviluppo delle moderne applicazioni web, che ha finalmente permesso allo sviluppatore di scrivere un’applicazione interamente in Javascript.

MEAN è l’acronimo di MongoDB, Express, Angular e Node, che sono Framework già inseriti nello studio e conformi ai nostri criteri di selezione (escluso MongDB che è un DBMS e che analizzeremo come attributo di qualità per la compatibilità con le basi di dati). Esso rappresenterà la nostra alternativa FullStack per l’ambiente Javascript, mentre Node.js, MongoDB ed Express.Js sono mandatori la componente AngularJS puo’ essere sostituita da Framework frontend come Ember.Js, Backbone.Js, React.Js. Essendo quindi il Panorma dei Framework javaScript vasto ma allo stesso tempo popolato da Frameworks di piccole dimensioni nella valutazione si considererà MEAN al pari di uno Framework FullStack classico come Spring MVC o Laravel. [2]

In sintesi:

• Linguaggio: Javascript

• Licenza: OpenSource

• Tipo: Fullstack

• Integrazione:

• Adozione:

• Vitalità:

• Statistiche Git Hub:

Verdetto: Selezionato come Alternativa

MEAN (Javascript FullStack)

In sintesi:

ASP.NET MVC

https://docs.microsoft.com/it-it/aspnet/core/mvc/overview?view=aspnetcore-2.2

• Linguaggio: Javascript

• Licenza: OpenSource

• Tipo: Fullstack

• Integrazione:

• Adozione:

• Vitalità:

• Statistiche Git Hub:

Verdetto: Selezionato come Alternativa

La seguente Tabella contiene l’insieme di Frameworks che formano lo spazio delle nostre possibili soluzioni ognuno dei quali è detto ‘alternativa’ la tabella nelle colonne indica la conformità dei singoli Frameworks con i criteri di selezione

Tabella riassunto del risultato dello screening preliminare.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Framework | Language | Type | License | Vitalità | Adozione | Conformità  Ai Criteri di selezione |
| Spring Framework | Java | Backend | Apache License 2.0 | Si | Si | Conforme |
| Google Web Toolkit | Java | Frontend | OpenSource | No | Si | Non Conforme |
| Spark | Java | - | - | - | - | Non Conforme |
| Vaadin | Java | Frontend | OpenSource | Si | Si | Conforme |
| Javalin | Java | - | - | - | - | Conforme |
| Laravel | PHP | Backend | MIT | Si | Si | Conforme |
| Symfony | PHP | Backend | MIT | Si | Si | Conforme |
| Cake PHP | PHP | Backend | OpenSource | Si | Si | Conforme |
| AngularJs | Javascript | Frontend | opensource | Si | Si | Conforme |
| Polymer | Javascript | Frontend | OpenSource | Si | - | - |
| React.JS | Javascript  /HTML | Frontend | OpenSource | Si | Si | Conforme |
| Express | Javascript | Backend | OpenSource | Si | Si | Conforme |
| Node.js | Javascript | Backend | MIT | Si | Si | Conforme |
| Sails.js | Javascript | Backend | OpenSource | Si | Si | Conforme |
| Ember.Js | JavaScript | - | - | - | - | - |
| Koa.Js | Javascript | Backend | OpenSource | Si | Si | Conforme |
| Django | Python | Backend | Django  License | - | - | Conforme |
| Ruby on Rails | Ruby | - | MIT | - | - | - |
| ASP.NET | Multi | Backend | Apache 2.0 | Si | Si | Conforme |
| Backbone.js | Javascript | Frontend | OpenSource | Sì | Sì | Conforme |
| Code Igniter | PHP | FullStack | OpenSource | Si | Si |  |
| Flask | Python | Backend | OpenSource |  | Si |  |
| Vue.js | Javascript | Frontend/  interfacce | Opensource |  |  |  |
| Zend | PHP | Fullstack | Opensource | No | No | Non Conforme |

# Definizione del modello e degli attributi di Qualità

4 Definizione del modello e degli attributi di Qualità 49

4.2 Metriche utilizzate

4.1 I Quality Attributes (Mappare Con ISO le Metriche!?)

Quality Attribute Presentazione/Generiche

Quality Attribute Trends

Quality Attribute Tecnologie & Features

4.3 Assegnare i Pesi 51

4.7 Instanza del quality model

Questo capitolo è un preambolo dell’analisi multi-criteriale vera e propria vengono infatti definiti criteri e regole di valutazione, individuando quality attribute significativi per l’analisi instanziamo un modello di qualità come definito dal metamodello in accordo con Basciani ET AL in [3], vengono individuate 3 tipi di metriche in corrispondenza dei raggruppamenti definiti dagli attributi di qualità, questi sono definiti dal Decisore arbitrariamente, è quindi molto importante fare le scelte giuste e scegliere solo metriche significative per la nostra analisi. Questo capitolo può essere suddiviso in 4 fasi distinte:

* Individuazione delle metriche e aggregazione in Quality Attributes
* Definizione dei Pesi.
* Assegnazione di un peso ai Quality Attributes
* Istanziare il Quality Model

Istanziare il modello di qualità come definito in [3] ci permette di aggiungere, togliere o sostituire dei Quality Attribute dalla nostra analisi, di fatto i quality attribute rappresentano i criteri di valutazione della nostra Analisi, sono un aggregato di metriche, allo stesso tempo dall’esterno può essere visto come una componente atomica intercambiabile con un suo peso. Per semplicità la somma dei pesi dei Quality Attribute del modello è pari a 1 (100%) cambiare un attributo significa bilanciare anche i pesi, il bilanciamento dei pesi è molto importante in quanto può spostare “l’ago della bilancia”. Questo implica che il Punteggio Massimo di ogni attributo al fine di una corretta valutazione deve essere uguale per tutti gli attributi, il problema del punteggio massimo ottenibile da ogni attributo è dato dalla differenza delle metriche che lo compongono, supponiamo che il punteggio massimo di ogni attributo sia 100 dovremmo trasformare il punteggio ottenuto dalla sommatoria delle metriche in centesimi con la seguente proporzione:

Pa : X = PAmax : 100

Pa = Punteggio Attributo; X = Punteggio in Centesimi; PAmax = Punteggio Massimo Attributo.

Per calcolare quindi il punteggio finale di ogni framework sarà sufficiente sommare il punteggio degli attributi moltiplicato per il rispettivo peso:

Pf= pa1\*w1+…+pan\*wn;

pa = punteggio attributo; w= peso attributo;

L’output di questo processo è un insieme di tabelle contenenti metriche pesate che serviranno ad effettuare la valutazione di qualità dei frameworks durante la fase di analisi del capitolo successivo.

## I Quality Attributes (Mappare Con ISO le Metriche!?)

Definiamo il modello di qualità su tre quality attribute in modo similare a Basciani ET AL in [3], la differenza è che non verrà utilizzato MDEForge per la valutazione. I tre Quality Attributr individuati sono Presentazione, Trend, e Tecnologie e Features.

## Metriche individuate

Al fine di Istanziare il quality model individueremo un set di metriche queste metriche sono state raggruppate in quality attributes, e poterli cosi’ valutare formalmente ed omogeneamente. Come specificato in [3] possiamo Istanziare un modello di qualità gerarchico Ad Hoc per il nostro studio. Si assume che le metriche siano individuate dagli stakeholder, ovviamente piu stakeholder individuano un set maggiore di metriche con priorità diverse, sarà quindi necessario trovare un modo per identificare quali sono le metriche significative e quelle che non lo sono. Nel nostro caso essendoci un unico decisore le metriche sono state individuate in base alle personali esperienze e preferenze.

Attributo Presentazione

* Tipo di Framework: Molto Alta
* Documentazione Molto Alta

Attributo Trend

* Frequenza delle release Media
* Stars peso Media
* Forks peso Media
* Pull-Request Media
* Issues Bassa
* Maturità peso Molto Bassa
* Trend Linguaggio di programmazione Media

Attributo Tecnologie e Features

* MVC Media
* Compatibilità DB Molto Alta
* Templates Molto Alta
* Caching Alta
* Moduli Molto Bassa
* Event Driven Programming Molto Bassa
* Namespaces Molto Bassa
* Test support Alta
* Sicurezza Molto Alta

**Mappatura con ISO/IEC 25010**

Al fine di avere un riscontro riconosciuto per le metriche individuate come indicatori di qualità mappiamo alcune di esse con le caratteristiche specificate nella ISO/IEC 25010 partendo dal modello di qualità del prodotto[[5]](#footnote-5) possiamo mappare alcune metriche individuate con quelle definite dallo standard.

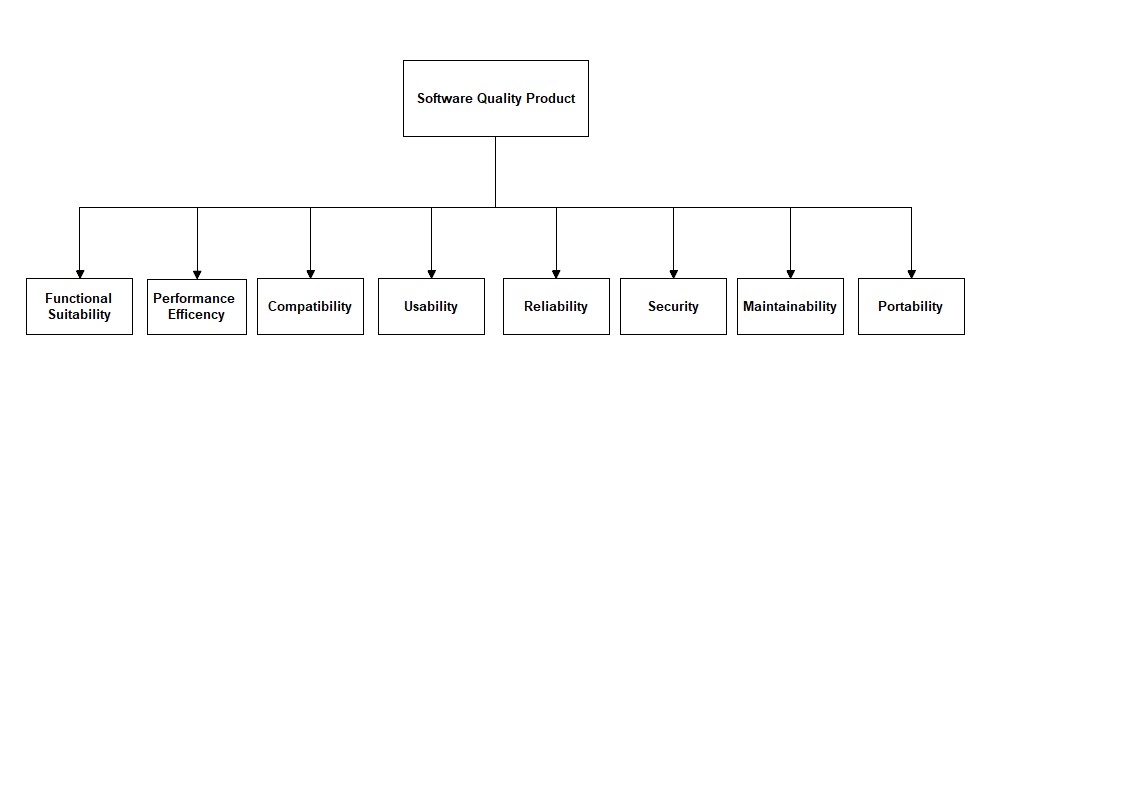
Figura ISO/IEC 25010:2011

Tabella Mappatura degli attributi con la ISO/IEC 25010:2011

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Metrica | Mappatura ISO/IEC 25010:2011 | Importanza |  |
| Documentazione | Usability:Learnability | Molto Alta |  |
| Tipo di Framework |  |  |  |
| Frequenza delle release |  |  |  |
| Stars |  |  |  |
| Forks |  |  |  |
| Pull-Request |  |  |  |
| Issues |  |  |  |
| Maturità |  |  |  |
| Trend Linguaggio di programmazione |  |  |  |
| MVC | Modularity |  |  |
| Compatibilità DB | Compatibility |  |  |
| Templating | Maintainability: Reusability |  |  |
| Testing | Maintainability: Testability |  |  |
| Sicurezza | Security |  |  |
| Caching Alta | Performance Efficency |  |  |
| Form Validation | Reliability: Fault Tolerance |  |  |

## Assegnare i Pesi

Un problema importante in questa fase è l’assegnazione dei pesi alle metriche/attributi questo perché il decisore utilizza un linguaggio informale è quindi necessario definire una metodologia di assegnazione dei pesi che non sia ambigua al fine di assegnare dei valori dei valori numerici corrispondenti a delle variabili linguistiche. Come primo passo di questo processo è quindi importante andare a definire una ‘scala linguistica’ a cui corrisponderanno dei valori numerici univoci. Definiamo quindi una corrispondenza biunivoca tra due insiemi, un insieme di parole utilizzate per definire l’importanza di una metrica e un insieme di numeri Naturali che rappresentano la scala di importanza corrispondente. È molto importante quindi Formalizzare un metodo di assegnazione dei in questo caso è stato scelto un metodo semplice ed efficace che ci permette di creare sotto insiemi discreti di metriche in base all’importanza assegnata calcoleremo quindi il peso in modo uniforme per ogni attributo.

;

L’associazione viene fatta dal decisore

I pesi verranno cosi costruiti: sia n il numero delle importanze non vuote si divide il peso comeplessivo in n porzioni ogni porzione si suddivide per gli elementi che appartengono all’importanza ennesima

Il peso di ogni metrica associato ad un grado di importanza è dato da:

Definiamo il calcolo del peso di una Metrica come

i coefficenti a,b,c,d possono assumere i valori {0,1}

;

IL divisore di ogni componente puo’ essere anche visto come la somma delle n metriche con importanza maggiore o uguale a quella i-esima.

~~MI è l’insieme con importanza maggiore a M~~

|  |  |  |
| --- | --- | --- |
| **Metriche** | **Importanza** | **Peso** |
| **MVC** | media | 0,055 |
| **Compatibilità DB** | Molto Alta | 0,2343 |
| **Templating** | Molto Alta | 0,2343 |
| **Testing** | Alta | 0,121 |
| **Sicurezza** | Molto Alta | 0,2343 |
| **Caching** | Alta | 0,121 |

somma totale dei pesi è pari a 0,9999

L’insieme delle importanze è cosi formato:

Attributo Tecnologie e Features

MVC Media

Compatibilità DB Molto Alta

Templating Molto Alta

Moduli Non Rilevante

Event Driven Programming Non Rilevante

Namespaces Non Rilevante

Test support Alta

Sicurezza Molto Alta

Caching Alta

Bassa 0 elementi

Media 1 elemento 1;

Alta 2 elementi 2;

Molto Alta 3 elementi 3;

L’insieme delle importanze presi in esame è

Formula di costruzione dei pesi:

peso complessivo

Con ogni w è calcolato nel seguente modo sia

Esempio

Peso delle metriche con importanza media = 0.33 /5 =0.066

Peso delle metriche con importanza media = 0.33 /5 + 0.33/4 =0.066+0.0825= 0.1485

Peso delle metriche con importanza media = 0.33 /5 + 0.33/4+0.34/2 = 0,066+0.0825+ 0.17=

La somma per verifica deve essere uguale ad 1

0.066+0.297+0.627= 0.99

Funziona da formalizzare!!!!

W= g \* (1/(n-1));

W = peso; g= grado di importanza assegnato alla variabile linguistica;

n = numero delle variabili linguistiche;

Di seguito la formula e la tabella con i pesi corrispondenti alle variabili linguistiche.

|  |  |  |
| --- | --- | --- |
| Nome Variabile | Importanza | Peso |
| Non rilevante | 0 | 0 |
| Bassa | 1 | 0.25 |
| Media | 2 | 0.50 |
| Alta | 3 | 0.75 |
| Molto Alta | 4 | 1 |

Questa tabella ci permette di distribuire in modo uniforme il peso delle variabili ma non è uno strumento che automatizza il processo di assegnazione dei pesi, infatti la valutazione dell’importanza di una metrica deve essere fatta dal decisore, nel nostro caso particolare l’importanza delle metriche è stata valutata in base all’esperienza ed alle conoscenze personali tenendo in considerazione gli obiettivi.

## Quality Attribute Presentazione/Generiche

Questo Quality Attribute è probabilmente il più semplice e intuitivo da valutare, infatti esso nasce dall’aggregazione di 3 metriche che servono ad inquadrare il Framework ma che saranno strumento di analisi come ad esempio la documentazione,

**Calcolo del Peso delle Metriche**

In questo attributo il peso è ripartito tra sole due metriche verrà quindi calcolato nel seguente modo:

di seguito la tabella riassuntiva dell’attributo presentazione:

|  |  |  |  |
| --- | --- | --- | --- |
| Quality Attribute | Metriche | Importanza | Peso |
| Presentazione | Documentazione | Molto Alta | 0,75 |
| Tipo di Framework | Alta | 0,25 |

**Documentazione**

La documentazione ha un peso importante in questa analisi è infatti una delle metriche con il peso maggiore, perché influisce sia sulla valutazione di attributi legati a features built-in del frameworks che sulla Usabilità del Framework parte dello sviluppatore, una documentazione incompleta o poco comprensibile potrebbe portarci ad una valutazione di qualità errata anche per quanto riguarda aspetti tecnici, questo perché la documentazione non verrà considerata solo come uno strumento per l’apprendimento ma anche di verifica di alcune caratteristiche e features del Frameworks (questo aspetto verrà ripreso in dettaglio anche nel quality model Tecnologie e Supporto allo Sviluppo Web).

Un buon Framework deve mettere a disposizione due tipi di documentazione: una d. tecnica incentrata sul codice sorgente (API Reference Documentation), e una d. utente costituita dai tutorials. Entrambe le documentazioni sono valutate in modo indipendente, per il materiale ufficiale fornito, non verranno quindi considerati in alcun modo articoli e tutorial su forum o blog non ufficiali, se alcuni aspetti sono poco chiari o assenti all’interno della pagina ufficiale verranno ritenuti assenti. Questo approccio potrebbe sembrare drastico ma tiene conto del tempo che un programmatore dovrebbe spendere per reperire informazioni non presenti sulla documentazione ufficiale, un fattore cruciale per mantenere bassi i costi dello sviluppo.

La valutazione è fatta utilizzando un approccio Knowledge-Based riguardante la mia esperienza nel campo di ingegneria del software e nello sviluppo software. Di seguito una tabella con i possibili punteggi ottenuti dalla documentazione.

|  |  |  |  |
| --- | --- | --- | --- |
| Documentazione | Punteggio Documentazione Utente | Punteggio Documentazione Tecnica |  |
| Completa: Documentazione ben strutturata, comprensibile, e facile da reperire ed organizzata gerarchicamente. | 1 | 1 |  |
| Sufficiente: Documentazione ben strutturata, con piccole mancanze su features o pochi commenti su alcune API Ref. Doc. | 0,67 | 0,67 |  |
| Incompleta: Presenza di metodi o features non commentati o documentati | 0,33 | 0,33 |  |
| Assente: mancanza della lingua inglese, API reference Doc. senza commenti | 0 | 0 |  |

**Tipo di Framework**

La tipologia di framework analizzato è una delle caratteristiche più importanti da tenere in considerazione possiamo infatti trovare frameworks FullStack che ci permettono di sviluppare l’intera applicazione senza ricorrere a librerie esterne o Backend/Frontend che ci permettono di sviluppare solo alcune componenti dell’applicazione web. Nella nostra analisi diamo un peso maggiore ai framework FullStack in quanto ci forniscono un ambiente di sviluppo completo. Nel Caso particolare dei FullStack Javascript Framework si prende in considerazione il modello di Framework MEAN considerando l’intercambiabilità della componente ‘Angular’ come descritto da Adhikari in [7].

Il tipo di Framework può essere di 3 tipologie ed assume i valori come nella seguente tabella:

|  |  |
| --- | --- |
| Tipo di Framework | Punteggio |
| Fullstack | 1 |
| Frontend | 0 |
| Backend | 0 |

## Quality Attribute Trends

Questo attributo raggruppa le metriche relative al trend di utilizzo del framework, ovvero metriche relativi alla community, al supporto, ed all’utilizzo, queste ci permetteranno ipotizzare uno scenario futuro per i Frameworks, con lo scopo di capire in che fase del ciclo di vita del software ci troviamo, un framework che è al tramonto sarà sicuramente meno appetibile di un framework all’apice del suo ‘ciclo’. La maggioranza delle informazioni relative a questo attributo sono reperite sulle pagine delle repository tenendo in considerazione gli attributi presenti nella seguente tabella, questo attributo può essere molto soggettivo, ma attraverso l’analisi sistematica ed una adeguata assegnazione dei punteggi in base a semplici regole è stato formalizzato, la il processo di valutazione è riproducibile applicando le regole definite ad Hoc per il nostro attributo.

**Calcolo del Peso delle Metriche**

In questo attributo il peso è ripartito tra sole due metriche verrà quindi calcolato nel seguente modo:

Verifica 0,094+0,339+0,566=1

di seguito la tabella riassuntiva dell’attributo presentazione:

|  |  |  |
| --- | --- | --- |
| Git Hub Contributors | Media | 0.047 |
| GitHub Stars | Alta | 0,113 |
| GitHub Pull Request | Molto Alta | 0,283 |
| Forks | Alta | 0,113 |
| Release per Anno | Media | 0.047 |
| Trend Linguaggio | Molto Alta | 0,283 |
| Dependency | Alta | 0,113 |

**Perche GitHub?**

Nella valutazione di questo attributo è stato fortemente utilizzato il riscontro di utilizzo della community presente su GitHub, perché è il repository più utilizzato al mondo dove possiamo accedere non solo al codice sorgente ma possiamo estrapolare anche molti dati interessanti che possono essere utilizzati come indici di qualità come già visto in [8]. I dati presi in considerazione sono riguardanti gli la community, tra questi troviamo:

**GitHub Contributors:** Il numero di Contributors per il Framework, ci permette di capire quanti sviluppatori attivi contribuiscono al progetto.

***GitHub Stars:*** è una metrica di gradimento del framework.

***GitHub Pull-Request:*** le pull-request ci indicando quanto sono attivi i contributors su un determinato repository.

***GitHub Forks:*** le forks ci indicano quanti nuovi progetti sono nati dal codice sorgente di un repository, questo dato puo’ essere considerato come una metrica per l’adozione del framework.

***GitHub Release Per Anno:*** è una metrica personalizzata questo rapporto può essere utilizzato per monitorare i cambiamenti significativi durante la storia di un frameworks e con che frequenza sono avvenuti

**GitHub Reposity:** per Linguaggio questa metrica è stata presa in considerazione per fare un confronto con il trend del linguaggio di programmazione preso da Stackoverflow.

**GitHub Dipendenze:** le dipendenze sono una metrica di Adozione del Software, molti repositorydipendenti da frameworks analizzato

**GitHub Peak Coding:** è un valore che possiamo ottenere dall’infografica del repository, possiamo relazionarlo con la Stabilità del software, in quanto un software che ha avuto un picco di codice in passato difficilmente un nuovo picco, un framework giovane invece potrebbe essere ancora in fase di sviluppo e quindi subire cambiamenti consistenti, questo può avvenire se ad esempio il picco corrisponde all’anno corrente, o ad esempio se negli ultimi 2 o 3 anni vi è una code frequency costante, influendo negativamente sulla stabilità.

**Repository relative ai Linguaggi (GitHub)**

JavaScript: 3,848,957 Repositories Rank 1

Java: 3,577,564 Repositories Rank 2

Python 1,874,470 Repositories Rank 3

PHP 1,229,110 Repositories Rank 4

Ruby 1,168,778 Repositories Rank 5

C# 858,778 Repositories Rank 6

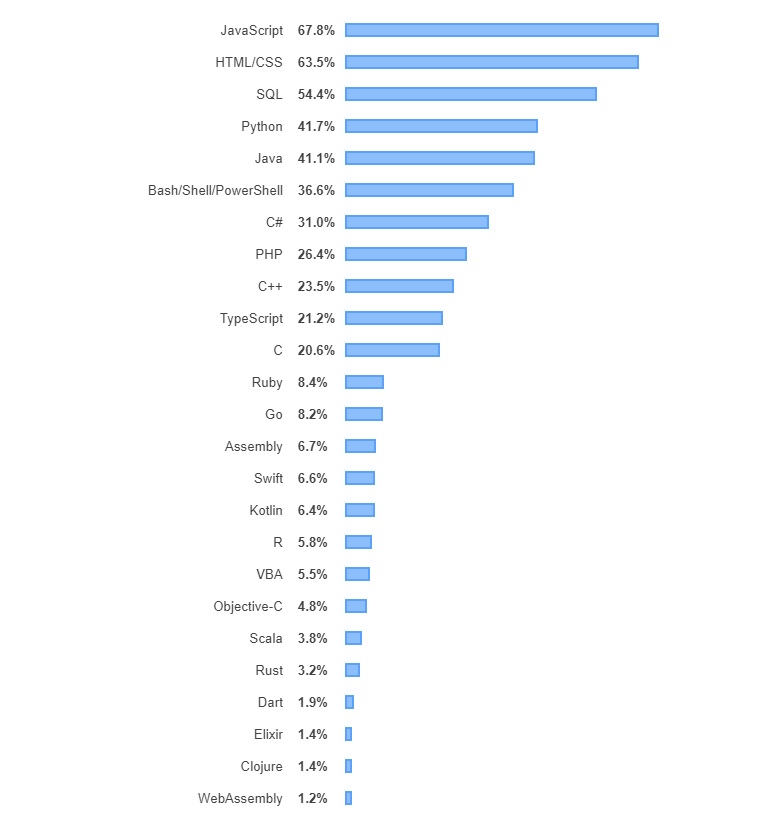
Statistiche online anche su https://madnight.github.io da controllare!!!

<https://pypl.github.io/IDE.html>

**Trend di utilizzo del Linguaggio**

In seguito ad una ricerca più approfondita ho trovato più attendibile il sondaggio annuale di Stackoverflow effettuato su un campione di 100.000 sviluppatori in particolare la domanda relativa alla popolarità dei linguaggi di programmazione che vede come linguaggi più popolari i seguenti.

Fig. 5 Trend di utilizzo dei linguaggi di programmazione secondo il sondaggio di stackoverflow.com



Con questo sondaggio possiamo quindi costruire una classifica per i linguaggi di programmazione presi in esame ai quali assegniamo un ranking ‘relativo’ attraverso la normalizzazione otteniamo un punteggio che viene utilizzato per valutarne il trend di utilizzo del linguaggio. Questa metriche è un esempio nel quale il valore minore indica una migliore qualità, si utilizzerà quindi la formula Z=1-Z.

In questa tabella notiamo in particolare 2, il linguaggio attualmente più utilizzato è il Javascript, ed il PHP in penultima posizione ormai in calo, questi valori sono effettivamente in linea con le metriche prese in esame nel repository Github dove ormai predominano i framework scritti in Javascript.

## Quality Attribute Tecnologie & Features

È il Quality Attribute che pesa di più sulla nostra analisi, in quanto tutti i sui attributi sono dipendenti da tecnologie e tecniche messe a disposizione dal Framework, può essere visto come il Quality Attribute delle Features del Framework, ma sarebbe riduttivo chiamarlo così in quanto aggrega anche metriche che non sono Features come ad esempio meccanismi di supporto architetturali come il pattern MVC. Di seguito le metriche individuate:

**Calcolo del Peso delle Metriche**

MVC: possibilità di strutturare l’applicazione secondo il pattern di design MVC.

Testing Tools: Presenza di strumenti per effettuare i test sull’applicazione.

Caching

Template Engine

Form Validation

Compatibilità Database

Sicurezza

In questo attributo il peso è ripartito tra ben 7 metriche, è stata assegnata l’importanza in base all’esperienza personale, un decisore con esperienza diversa può quindi assegnare le metriche in modo differente, e ricalcolare i pesi. In questo caso troviamo 1 sola metrica con importanza ‘media’, 2 con importanza ‘alta’, e 3 con importanza ‘molto alta’.

|  |  |  |  |
| --- | --- | --- | --- |
| Quality Attribute | Metriche | Importanza | PESO  Tecnologie |
| Tecnologie  &  Features | MVC | Media | 0,055 |
| Compatibilità DB | Molto Alta | 0,2343 |
| Templating | Molto Alta | 0,2343 |
| Testing | Alta | 0,121 |
| Sicurezza | Molto Alta | 0,2343 |
| Caching Alta | Alta | 0,121 |
| Form Validation | Alta | 0,121 |

**Design Pattern**

Il Framework può definire delle linee guida per quanto riguarda il design pattern da utilizzare, la maggior parte dei moderni Framework di Sviluppo web permettono di sviluppare applicazioni web secondo il classico design pattern MVC [8] [9], alcuni mettendo a disposizione delle API specifiche come nel caso di Spring MVC con le classi Model,View,Controller, altri Framework non di tipo FullStack potrebbero mettere a disposizione solo alcune delle 3 componenti. Preso atto di questi diversi scenari la valutazione viene fatta considerando singolarmente le componenti M, V, C nel caso in cui siano presenti tutte le componenti il Framework avrà un punteggio pari a 1 altrimenti verranno sommati i valori delle singole componenti come indicato nella tabella seguente.

|  |  |
| --- | --- |
| Design Pattern | Punteggio |
| MVC | 1 |
| MV/MC/VC | 0,67 |
| M/V/C | 0,33 |
| Assente | 0 |

**Compatibilità Database**

La compatibilità del database è un aggregato di attributi. Il valore viene calcolando tenendo in considerazione le tecnologie supportate con peso 0.4 e la presenza di features CRUD con peso 0.6 (invertire i pesi). Le informazioni relative alla compatibilità dei database sono reperite analizzando la documentazione dei Frameworks.

**Valutazione Features CRUD**

La nostra applicazione web come descritto nel caso di studio è dipendente dalle informazioni memorizzate nel database, è quindi di Fondamentale importanza avere a disposizione dei meccanismi per effettuare le operazioni CRUD nel modo piu semplice e veloce possibile. La Feature ha un valore binario 1 se è presente 0 altrimentiCRUD Punteggio

Presenti 1

Assenti 0

**Tabella di valutazione delle Tecnologie DB supportate**

Tecnologia Valore

Mysql/SqlLite/MongoDB 1

Altro 0.5

Assente 0

**Facilità di Testing**

Un buon Framework uno strumento per semplificare allo sviluppatore l’attività di testing, come il supporto per lo sviluppo guidato dai test (TDD) o per gli Unit Test, questa metrica è stata verificata analizzando la documentazione ed il codice dei frameworks. I Framework che forniscono un sistema di Tesing built-In avranno un punteggio massimo mentre i framework che specificano nella documentazione l’utilizzo di pacchetti esterni avranno un punteggio buono in caso invece che la documentazione riguardante i meccanismi di testing abbia delle gravi mancanze o non sia specificato assegneremo un punteggio minimo.

Testing Tools Punteggio

Built-in 1.00

Esterno 0.80

Assente 0.00

**Meccanismi di Caching**

L’applicazione da sviluppare è interamente dinamica, è quindi importante avere a disposizione un meccanismo di caching che ci permetta di migliorare le prestazioni della nostra applicazione web, nel nostro caso non andremo a determinare la differenza tra un meccanismo di caching rispetto ad un altro ma ne verifichiamo solo la presenza all’interno del framework o la compatibilità con una libreria esterna ben documentata, il metodo di analisi anche in questo caso è legato alla documentazione dei Frameworks. I Framework che forniscono un sistema di Caching built-In avranno un punteggio più alto.

Caching Punteggio

Built-in 1.00

Esterno 0.80

Assente 0.00

**Template Engine**

Il Template Engine ci permette di combinare la view con il modello di dati, questo ci permette di realizzare applicazioni dinamiche facilmente. In genere un Template Engine utilizza un linguaggio proprio detto template language attraverso il quale possiamo costtruire cicli, gestire variabili, funzioni e sostituzioni di testo. Generalmente un template engine prende un template sorgente in HTML con direttive riconosciute dal parser testuale del Template Processor e lo riempie con dati provenienti dal Model solitamente DB, ORM o JSON, l’utilizzo delle tecniche di templating è imprescindibile dal design pattern MVC. I Linguaggi di programmazione scelti mettono a disposizione funzioni Built-In o pacchetti e moduli esterni come JSP, ASP , Genshi, eRuby. Nel nostro insieme di frameworks abbiamo soluzioni che includono dei Template Processor Built-in e soluzioni con moduli esterni, nel nostro caso diamo una valutazione migliore a soluzioni integrate, la motivazione è che il programmatore spesso preferisce avere un pacchetto completo per risparmiare tempo di sviluppo, solo in caso di necessità o di notevoli miglioramenti prestazionali lo sviluppatore è spinto a ricorrere a soluzioni esterne. La seguente tabella riassume la valutazione applicata al quality attribute in base ai diversi casi possibili. [7]

Template Engine Punteggio

Built-in 1.00

Esterno 0.80

Assente 0.00

**Sicurezza [9] (DRAFT)**

Garantire la sicurezza è un aspetto fondamentale delle moderne applicazioni web, con la crescita degli utenti e dei dispositivi presenti in rete si riscontra anche una crescita delle applicazioni web, queste possono esporre a soggetti terzi sia i dati dei consumatori che i dati dell’azienda anche le istituzioni negli ultimi anni si stanno attivando per garantire la sicurezza dei dati infatti il 25 maggio 2018 è entrato in vigore il nuovo regolamento europeo sulla privacy, il GDPR 2018 [8] che attribuisce al titolare del servizio tutte le responsabilità sul trattamento dei dati ed introduce il diritto all’oblio, una falla nella sicurezza dell’applicazione potrebbe permettere attacchi di tipo Cross-site request forgery ed esporre dati sensibili a terzi, senza entrare troppo nei particolari legislativi si vuole sottolineare come la sicurezza della privacy e dei dati sia un fattore importante per quanto concerne lo sviluppo delle moderne applicazioni web. Vi sono anche altri tipi di attacchi e di falle nella sicurezza di un’applicazione web, come gli attacchi SQL injection, o Distribuited Denial of Service che possono influire sulla sulla reliability. Anche in questo caso assegnamo un punteggio maggiore ai frameworks che implementano sistemi di gestione della sicurezza, mentre ai frameworks che si appoggiano a pacchetti esterni indicati nella documentazione ufficiale daremo una valutazione molto buona, in caso invece di mancanza o di informazioni insufficienti o poco chiare riguardanti la sicurezza verrà valutato negativamente.

Sicurezza Punteggio

Built-in 1.00

Esterno 0.80

Assente 0.00

**Form Validation**

Il Meccanismo di validazione delle form è da sempre stato un aspetto fondamentale per un’applicazione web, una buona gestione della validazione dei dati inseriti dall’utente può influire sia sulla correttezza dei dati memorizzati nel database nel lato server che sull’usabilità dell’applicazione. Un errato inserimento dei dati potrebbe infatti compromettere i dati e questo può essere un problema che riguarda il lato server, un problema del lato client potrebbe riguardare l’usabilità dell’applicazione, mentre infatti un tempo era ‘accettato’ da parte dell’utente ricompilare una form a seguito di una risposta di errore da parte del server oggi potrebbe spingere l’utente ad abbandonare il servizio o comunque a dare una valutazione negativa dell’applicazione web è quindi necessario che le validazioni dei dati inseriti vengano fatte anche dal lato client preferibilmente in tempo reale all’inserimento con opportuni ‘Hints’ di errore. Come nei casi precedenti adottiamo la politica di valorizzare i framework che forniscono questi meccanismi Built-in, assegnando un punteggio buono a frameworks che specificano l’utilizzo di pacchetti esterni all’interno della documentazione ufficiale e un punteggio minimo altrimenti.

Form Validation Punteggio

Built-in 1.00

Esterno 0.80

Assente 0.00

## il Quality model in sintesi

Questa tabella è una sintesi del Quality Model, dalla quale possiamo reperire tutte le informazioni necessarie per proseguire la valutazione delle Alternative, nella Quality Attribute sono troviamo in nomi assegnati agli attributi di qualità, nella seconda le metriche che li compongono, nella terza il valore che può assumere una metrica con il relativo punteggio, ed infine nella 4 colonna il peso. Per come abbiamo assegnato i punteggi e per il metodo con il quale sono stati ripartiti i pesi delle metriche, ogni attributo avrà un punteggio pari a 1 questo ci permette di avere una valutazione omogenea tra i vari attributi, durante la valutazione avremo la possibilità di assegnare un peso per ogni attributo potendo così personalizzare ulteriormente la valutazione e l’importanza di ogni attributo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality Model** | | | |
| **Quality Attribute** | **Nome Metrica** | **Importanza Metriche** | **Peso Metriche** |
| **Presentazione** | Documentazione |  |  |
| Tipo |  |  |
| **Tecnologie**  **&**  **Features** | MVC | Media | 0,055 |
| Compatibilità DB | Molto Alta | 0,2343 |
| Templating | Molto Alta | 0,2343 |
| Testing | Alta | 0,121 |
| Sicurezza | Molto Alta | 0,2343 |
| Caching Alta | Alta | 0,121 |
| Form Validation | Alta | 0,121 |
| **Trends** | Git Hub Contributors | Media | 0.047 |
| GitHub Stars | Alta | 0,113 |
| GitHub Pull Request | Molto Alta | 0,283 |
| Forks | Alta | 0,113 |
| Release per Anno | Media | 0.047 |
| Trend Linguaggio | Molto Alta | 0,283 |
| Dependency | Alta | 0,113 |

# Applicazione dell’analisi criteriale al quality model

In questo capitolo applichiamo l’Analisi Multi-Criteriale basata Quality Model, questo processo avviene in due fasi con due metodi differenti, la prima fase consiste nella valutazione dei Quality Attribute per ogni framework, una seconda fase empirica di approfondimento per i framework che hanno ottenuto un punteggio più alto secondo la metrica applicataValutazione dei Frameworks utilizzando il Quality Model

A questo punto del processo di analisi valutare un framework corrisponde a fare una valutazione di qualità che rispetti le regole definite dal quality model.

Il Quality model definito nel capitolo precedente ci permette di valutare i Frameworks con l’applicazione della seguente formula:

Per ogni framework analizzato assegnamo un punteggio, a questo punto del processo il decisore è chiamato in casua per l’assegnazione dei pesi degli attributi, vi sono 2 possibilità: assegnare un peso per ogni attributo o non assegnare un peso e valutare gli attributi alla pari.

Formalizziamo quindi come il risultato finale sarà ottenuto:

*Max(PF) è una funzione che restituisce il framework con punteggio massimo.*

*A= è l’insieme delle Alternative.*

*PF è il punteggio calcolato per ogni Framework.*

Il Calcolo dei punteggi dei quality attribute è ottenuto applicando la seguente formula:

## Normalizzazione dei dati

Prima di calcolare un punteggio per gli attributi dei frameworks procediamo con la normalizzazione dei valori delle metriche attraverso la tecnica di scaling conosciuta come min-max scaling.

Vi sono casi in cui un valore inferiore indica una qualità migliore, in questo caso analogamente a Zurkiewicz ET AL [2] se il valore di un’attributo è inversamente proporzionale alla qualità ovvero se un valore basso indica una qualità maggiore allora il valore normalizzato sarà dato dalla seguente formula:

## Valutazione dei quality attribute

Per ogni Quality Attribute procediamo come definito precedentemente, si analizza ogni metrica, alla quale assegnamo un punteggio, tutti i punteggi ottenuti dalle metriche saranno normalizzati, questo ci permetterà di avere dei dati omogenei che risultino per il lettore anche più comprensibili e leggibili.

## Valutazione del Quality Model Presentazione/Generiche

Come definito nel capitolo precedente questo attributo è composto da due metriche: Documentazione e Tipo di Framework che verranno valutate secondo le regole specificate precedentemente.

**Valutazione della Docmuntazione**

Nella seguente tabella troviamo i valori assegnati alla documentazione dei framework in base alla metrica definite nel capitolo precedente. Prendiamo come esempio alcuni risultati scaturiti dall’analisi della documentazione, nel caso ad esempio della prima riga della tabella in riferimento al framework SpringMVC possiamo notare che il punteggio assegnato è massimo, questo perché sia la documentazione Tutorial che quella del codice sorgente sono complete, mentre nel caso di Laravel possiamo notare che ha ottenuto un punteggio minimo per la documentazione del codice questo perche sulla pagina ufficiale non è reperibile. In questa valutazione il decisore potrebbe commettere degli errori di valutazione, ma grazie alla definizione della metrica del capitolo precedente si limita l’errore umano. Il punteggio ottenuto dalle due differenti documentazioni viene normalizzato per essere omogeneo agli altri punteggi.

|  |  |  |  |
| --- | --- | --- | --- |
| Framework | Doc Tutorials | Doc Source Code | Punteggio |
| Spring MVC | 1 | 1 | 2 |
| Vaadin | 1 | 1 | 2 |
| Javalin | 1 | 0 | 1 |
| Laravel | 1 | 0 | 1 |
| Symfony | 1 | 1 | 2 |
| Cake PHP | 1 | 0.67 | 1.67 |
| Code Igniter | 1 | 1 | 2 |
| AngularJs | 1 | 1 | 2 |
| Polymer | 1 | 1 | 2 |
| React.JS | 1 | 1 | 2 |
| Express.js |  |  |  |
| Node.js | 1 | 1 | 2 |
| Sails.js | 1 | 1 | 2 |
| Koa.Js | 1 | 1 | 2 |
| Ember.Js | 1 | 1 | 2 |
| Backbone.js | 0,67 | 0,67 | 1,34 |
| Vue.js |  |  |  |
| Django |  |  |  |
| Flask |  |  |  |
| Rails |  |  |  |
| ASP.NET |  |  |  |
| ASP.NET MVC (archived 2019) |  |  |  |
| Apache Struts |  |  |  |
| Sinatra |  |  |  |
| Yii |  |  |  |

NOTE

Spark api doc generata poco comprensibile

**Valutazione del tipo di Framework**

Il Tipo con il quale viene presentato il framework influisce indicativamente sulle funzionalità dello stesso, in genere un framework presentato come Fullstack permette allo sviluppatore di sviluppare l’intera applicazione, mentre i framework Frontend/Backend solo una parte. In questa tabella possiamo notare in particolare alcuni valori, nel caso di Spring MVC il punteggio è massimo mentre in Node.js ed AngularJs si ha un punteggio minimo questo perche Node.Js è un Framework Backend ed AngularJs è Frontend per valutare in modo omogeneo anche l’ambiente JS abbiamo aggiunto il Bundle MEAN come se fosse un Framework Fullstack e come Spring ha un punteggio massimo.

|  |  |  |
| --- | --- | --- |
| Framework | Tipo di Framework | Punteggio |
| Spring MVC | Fullstack | 1 |
| Vaadin | Frontend | 0 |
| Javalin | Backend | 0 |
| Laravel | Fullstack | 1 |
| Symfony | Fullstack | 1 |
| Cake PHP | Fullstack | 1 |
| Code Igniter | Fullstack | 1 |
| AngularJs | Frontend/MEAN | 0 |
| Polymer | Frontend | 0 |
| React.JS | Frontend/MEAN | 0 |
| Express.js | Backend/MEAN | 0 |
| Node.js | Backend/MEAN | 0 |
| Sails.js | Backend | 0 |
| Koa.Js | Backend | 0 |
| Ember.Js | Frontend | 0 |
| Backbone.js | Frontend | 0 |
| Vue.js | Frontend | 0 |
| Django | Fullstack | 1 |
| Flask | BackEnd | 0 |
| Rails | Fullstack | 1 |
| ASP.NET | Fullstack | 1 |
| ASP.NET MVC (archived 2019) | Fullstack | 1 |
| Apache Struts | Fullstack | 1 |
| Sinatra |  |  |
| Yii |  |  |

## Valutazione del Quality Model Trends

Questa Tabella riassume i dati acquisiti durante l’analisi delle metriche specificate nell’attributo ‘Stato del Framework’ è l’attributo dove il decisore può commettere meno errori rispetto agli altri in quanto i dati analizzati sono numerici, i dati acquisiti da GitHub e da Stackoverflow vengono normalizzati, pesati ed utilizzati per assegnare il punteggio all’attributo. Si può notare come vi sia una relazione tra il Trend di utilizzo del linguaggio di programmazione acquisito su stackoverflow ed i dati acquisiti su Githhub, possiamo notare infatti che i Framework scritti in javascript risultano più attivi su Github e che secondo il sondaggio di stackoverflow il Javascript risulta essere il linguaggio più in voga in quest’ ultimo anno, vi sono comunque delle discordanze dovute al periodo di tempo analizzato in quanto alcuni dei progetti sono presenti su Github da piu di 10 anni.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Framework | Contributors | Stars | Pull-Request | Forks | Dependency  Dependent/Repo/Package[[6]](#footnote-6) | Releases | Years on GitHub | Pull-Request  Merged/Proposed [[7]](#footnote-7) | Issues  Closed / New[[8]](#footnote-8) |
| Spring MVC | 381 | 30.983 | 187 | 20.006 | 0/124.294/0 | 156 | 10 | 7/11 | 110/64 |
| Vaadin | 181 | 1.513 | 11 | 707 | 49/7.341/0 | 248 | 12 | 20/3 | 16/10 |
| Javalin | 65 | 3.162 | 2 | 264 | 41/637/0 | 55 | 2 | 15/2 | 21/7 |
| Laravel | 504 | 53.883 | 0 | 16.527 | 11/0/0 | 94 | 7 | 8/0 | 0/0 |
| Symfony | 1.905 | 21.296 | 203 | 7.085 | 1/0/0 | 445 | 8 | 303/56 | 182/76 |
| Cake PHP | 530 | 7915 | 23 | 3.405 | 0/0/0 | 363 | 13 | 37/4 | 22/8 |
| Code Igniter | 440 | 17.494 | 34 | 7.731 | 2/0/0 | 34 | 12 | 0/3 | 8/5 |
| AngularJs | 1.591 | 4.175 | 76 | 28.888 | 76/101.164/4690 | 204 | 8 | 0/1 | 8/0 |
| Polymer | 142 | 21.141 | 26 | 2.005 | 28/4.693/2788 | 153 | 6 | 6/3 | 1/2 |
| React.JS | 1.304 | 133.587 | 191 | 24.850 | 82/2.258.967/93.109 | 116 | 5 | 80/23 | 80/52 |
| Express.js | 233 | 44.891 | 55 | 75.545 | 48/4.504.803/53.321 | 279 | 9 | 0/2 | 16/5 |
| Node.js | 2.510 | 63.177 | 281 | 14.420 | 102/0/0 | 556 | 9 | 41/65 | 126/62 |
| Sails.js | 214 | 20.730 | 36 | 1.853 | 63/19.583/484 | 271 | 6 | 0/2 | 7/19 |
| Koa.Js | 176 | 26.860 | 18 | 2.461 | 32/88.373/6.342 | 81 | 5 | 4/1 | 4/4 |
| Ember.Js | 760 | 21.127 | 84 | 4169 | 104/17.844/3.340 | 380 | 7 | 20/14 | 8/11 |
| Backbone.js | 291 | 27.520 | 35 | 5.687 | 10/31.254/1669 | 31 | 8 | 0/0 | 0/2 |
| Vue.js | 277 | 144.994 | 102 | 21.120 | 74/953.590/28.200 | 248 | 3 | 6/12 | 86/13 |
| Django | 1775 | 43.095 | 222 | 18.537 | 26/277.796/7388 | 224 | 13 | 64/19 | 0/0 |
| Flask | 546 | 45.653 | 2 | 12.760 | 15/296.291/4.346 | 30 | 8 | 13/1 | 23/2 |
| Rails | 3.862 | 43.736 | 763 | 17.616 | 2.770/1.180.808/12.514 | 367 | 13 | 155/25 | 114/23 |
| ASP.NET | 626 | 12.986 | 102 | 3.186 | 2381/32.047/26 | 75 | 5 | 312/51 | 584/301 |
| ASP.NET MVC (archived 2019) | 150 | 5778 | 0 | 2.237 | 80/298.045/2.831 | 49 | 5 | 0/0 | 0/0 |
| Apache Struts | 40 | 959 | 7 | 638 | 446/1/0 | 115 | 12 | 0/0 | 0/0 |
| Sinatra | 361 | 10.636 | 36 | 1912 | 90/145.288/3.197 | 133 | 11 | 0/1 | 1/3 |
| Yii | 268 | 4.865 | 0 | 2.227 | 2/0/0 | 43 | 10 | 0/0 | 2/0 |
|  |  |  |  |  |  |  |  |  |  |

\* Periodo esaminato: 1 mese

\*\* Dependency: Dependent from / Repository dependant by /Package dependant by

\*\*\* Accesso ai dati di Github il 31/07/2019

**Tabella Riassuntiva di valutazione del quality models**

Questa tabella ci mostra il punteggio ottenuto valutando il quality attribute Trends per ogni Framework.

|  |  |
| --- | --- |
| Framework | Punteggio |
| Spring MVC | 1,320 |
| Vaadin | 0,870 |
| Javalin | 1,068 |
| Laravel | 1,037 |
| Symfony | 1,580 |
| Cake PHP | 0,732 |
| Code Igniter | 0,518 |
| AngularJs | 2,092 |
| Polymer | 1,474 |
| React.JS | 2,833 |
| Express.js | 2,708 |
| Node.js | 3,002 |
| Sails.js | 1,733 |
| Koa.Js | 1,411 |
| Ember.Js | 2,025 |
| Backbone.js | 1,335 |
| Vue.js | 3,339 |
| Django | 1,969 |
| Flask | 1,420 |
| Rails | 1,844 |
| ASP.NET | 0,828 |
| ASP.NET MVC (archived 2019) | 0,576 |
| Apache Struts | 0,689 |
| Sinatra | 0,289 |
| Yii | 0,332 |

## Valutazione del Quality Attribute Tecnologie e Supporto allo Sviluppo WEB

**(DRAFT due tabelle una valori numerici e una features tesuali? inoltre i punteggi corretti sono nel foglio excel)**

Le seguenti Tabelle contengono i dati raccolti durante l’analisi degli attributi del quality model ‘Tecnologie e Supporto allo Sviluppo Web’. I dati sono stati prodotti analizzando la documentazione ed il codice sorgente, in alcuni casi particolari laddove la documentazione non è stata ritenuta esaustiva sono state effettuate ricerche esterne alla documentazione poi verificate confrontando i dati con il codice del Framework. Per semplificare la comprensione delle tabelle riassuntive sono stati separati i dati testuali dai rispettivi valori numerici.

**Tabella Testuale del QA Tecnologie e Supporto allo Sviluppo WEB**

In questa tabella troviamo i dati testuali che verranno poi tradotti in valori numerici secondo le regole definite nel capitolo 4. Prendiamo in considerazione la prima riga della tabella relativa al framework Spring MVC in questo caso avendo una documentazione completa, tutte le informazioni sono chiare e ben definite, la prima colonna indica il tipo di framework, essendo un framework di tipo fullstack ci aspettiamo che tutte le celle della riga siano ‘complete’, la seconda ci indica che il framework offre un pieno supporto al design pattern MVC, nella cella successiva ‘Compatibilità DB’ troviamo elencate le tecnologie supportate che anche in questo caso sono conformi con le nostre metriche, le successive celle : Test, Sicurezza, Caching, Form Validation e Template Engine indicano il supporto alle rispettive funzionalità offerte dal framework, mentre per le prime nella documentazione abbiamo trovato informazioni complete riguardanti meccanismi Built-in come ad esempio la Spring Tool Suite (STS), nell’ultima cella quella riguardante il Template Engine troviamo solo informazioni riguardanti pacchetti sviluppati da terzi, questo non è un indicatore negativo ma in linea con le metriche definite otterrà un punteggio leggermente inferiore in quanto è stato premiato il framework che offre funzionalità Built-in che permette allo sviluppatore di scegliere quale tecnologia usare, se una Built-in oppure una sviluppata da terzi, nel caso particolare comunque possiamo notare che Spring MVC fa affidamento a tecnologie ben collaudate come ad esempio JSP .

Tabella Quality Attribute ‘Tecnologie & Features’ in forma tabellare.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Framework | Tipo | Design Pattern | Compatibilità  DB | Test | Sicurezza | Caching | Form  Validation | Template Engine |
| Spring MVC | FullStack | MVC | MySQL,MongoDB  … | Spring Tool Suite  Unit,Mock,Built-in | Spring Security  Built-in | Spring Caching  Built-in | Spring Validation  Built-in | JSP,Thymeleaf, Velocity… |
| Vaadin | Frontend | V | SQLContainer  (permette di creare una connessione ad un database) | JUnit5,Testbench (vaadin github Archiviato) |  | SQLContainer (database caching) | Vaadin  Validator,Binder Built-in | Vaadin Template  Built-in |
| Javalin | Backend  Minimale | - | Assente?/CRUD Assente | Assente?  Non Specificato nella Documentazione | Pac4J  (progetto di testing per Javalin GitHub)  , Javalin-jwt | Javalin Caching  Built-in | Javalin Validator  Built-in | Thymeleaf, Velocity,Mustachee |
| Laravel | Backend  Fullstack | MVC | MySQL,SQLite  … | PHPUnit | Laravel Security  Built-in | Memcached,Redis | Laravel Validation  Built-in | Laravel Template Engine Blade  …  Built-in |
| Symfony | Backend  Fullstack | MVC | MySQL, MongoDB | PHPUnit | Symfony  Security  Built-in | Cache contracts,PSR-6 | Symfony Validator  Built-in | Symfony Template Parser  …  Built-in |
| Cake PHP | Backend  Fullstack | MVC | MySQL, SQLite  … | PHPUnit,  Cake Test Support.  Built-in | Cake SecurityComponent  Built-in | File,Apcu, Wincache, Memcached, Redis, Array. | Cake Validator  Built-in | CakePHP Views  …  Built-in |
| Code Igniter | FullStack  Backend | MVC | MySQL | Code Igniter unit test, PHP Unit  Built-in | Code Igniter Security Class  Built-in | File,Apcu, Wincache, Memcached, Redis, Dummy Cache. | CodeIgniter  Validator  Built-in | CodeIgniter Parser Class  …  Built-in |
| AngularJs | Frontend/ MEAN | MV | Assente? Verificare! | Unite, [Karma](https://docs.angularjs.org/guide/unit-testing#karma), [Jasmine](https://docs.angularjs.org/guide/unit-testing#jasmine), [angular-mocks](https://docs.angularjs.org/guide/unit-testing#angular-mocks). | AngularJS Security  Built-in | AngularJs caching Object (verificare)  Built-in | AngularJS Validator  Built-in | Templates  Built-in???? |
| Polymer | Frontend |  | Assente? Verificare! |  |  | Assente? Verificare! |  |  |
| React.JS | Frontend/MEAN | V | Assente | React-unit, Unexpected-react  Built-in | Assente? Verificare! | Assente? Verificare! | React Validator  Built-in | React  Built-in |
| Express.js | Backend/ MEAN | M,C | MongoDB | Verificare!  Mocha |  | Verificare |  |  |
| Node.js | Backend/MEAN | M,C | MongoDB | Mocha |  | Assente? Verificare!  Node-cache esterno |  |  |
| Framework | Tipo | Design Pattern | Compatibilità  DB | Test | Sicurezza | Caching | Form  Validation | Template Engine |
| Sails.js | Backend |  | MySQL, MongoDB, PostgreSQL | Mocha |  | sailsjs-cacheman  (Plugin Verificare) |  |  |
| Koa.Js | Backend |  | - | Mocha |  | koa-static-cache  (Plugin Verificare) |  |  |
| Ember.Js | Frontend/MEAN |  | - | QUnit Ember Built-in,Mocha… |  | Assente? Verificare! |  |  |
| Backbone.js | Frontend/MEAN |  | - |  |  |  |  |  |
| Vue.js | Frontend/  interfacce | - | - |  |  |  |  |  |
| Django | Fullstack | MVC | MySQL, SQLite, Oracle… |  |  | CachingVerificare |  |  |
| Flask | Backend |  | SQLite |  |  | Flask-Caching Est  Verificare |  |  |
| ASP.NET | Backend/  Fullstack |  | MySQL, SQLite,  SQL Server  … |  |  |  |  |  |
| Rails | Backend/  Fullstack | MVC | MySQL, SQLite, PostgreSQL |  |  | Rails Caching molti meccanismi presenti  Verificare!  Built-in |  |  |

Tabella i valori numerici del quality attribute Tecnologie e Features.

**Tabella Scalare del QM Tecnologie e Supporto allo Sviluppo WEB**

Questa Tabella ci mostra il risultato dell’analisi dei framework in relazione con le metriche definite dall’attributo di qualità Tecnologie & Features, questa tabella è la trasformazione dei valori testuali in valori numerici, tutti i valori sono stati normalizzati. Ritornando all’esempio precedente riguardante Spring MVC possiamo vedere come abbia ottenuto un punteggio massimo in tutte le celle tranne in quella relativa al ‘Template Engine’ questo perchè nella metrica è specificato che la stessa può assumere 3 valori numerici corrispondenti: 0 se non vi è supporto alla tecnologia, 0,80 se vi è supporto ma sviluppato da terzi, 1 se vi è supporto Built-in. Una riga importante riguarda la valutazione del Bundle MEAN, questa in particolare è data dalla combinazione dei frameworks che compongono il fullstack in Javascript e sono Mongo, Express, AngularJS e Node.js .(continua con MEAN)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Framework | Punteggio Tipo di Framework | Punteggio Design Pattern | Punteggio  Compatibilità  DB | Punteggio Test | Punteggio  Sicurezza | Punteggio Cahing | Template Engine |
| Spring MVC | 1 | 1 | 1 | 1 | 1 | 1 | 0,80 |
| Vaadin | 0 | 0.33 | 1 | 1 |  | 1 | 1 |
| Javalin | 0 | 0.33 | 0 | 0 | 1 | 0 | 1 |
| Laravel | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Symfony | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Cake PHP | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Code Igniter | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| AngularJs | 1 | 0.67 | 0 | 1 | 1 | 1-+ | 1 |
| Polymer | 0 | 0 | 0 |  |  | -+ |  |
| React.JS | 1 |  | 0 | 1 |  | -+ |  |
| Express.js | 1 | 0.67 | 1 |  |  | -+ |  |
| Node.js | 1 | 0.67 | 1 |  |  | - |  |
| Sails.js | 0 |  | 1 |  |  | - |  |
| Koa.Js | 0 |  | 0 |  |  |  |  |
| Ember.Js | 1 |  | 0 |  |  |  |  |
| Backbone.js | 1 |  | 0 |  |  |  |  |
| Vue.js | 0 |  | 0 |  |  |  |  |
| Django | 1 | 1 | 1 |  |  | 1 |  |
| Flask | 0 |  | 1 |  |  |  |  |
| ASP.NET | 1 |  | 1 |  |  |  |  |
| Rails | 1 | 1 | 1 |  |  | 1 |  |
| MEAN |  |  |  |  |  |  |  |

## Valutazione della qualità dei Frameworks

Dopo aver valutato singolarmente gli attributi possiamo fare una valutazione complessiva dei framework utilizzando la seguente formula:

La soluzione S corrisponderà l'alternativa che ha la riga con sommatoria degli elementi più alta.

**Calcolo del Peso dei Quality Attribute**

Si è deciso di assegnare un peso ad ogni attributo anche in questo caso utilizziamo la scala precedente e lo stesso metodo di calcolo dei pesi, si è scelto di fare questo in modo da rendere gli attributi di qualità intercambiabili, cosi un decisore potrebbe decidere di eliminare, sostituire o aggiungere un attributo di Qualità senza dover modificare il procedimento di calcolo. In dettaglio il peso degli attributi di qualità è stato cosi calcolato:

Verifica peso totale 0,9999

|  |  |  |
| --- | --- | --- |
| Quality Attribute | Importanza | Peso |
| Presentazione | Alta | 0,1666 |
| Stato del Framework | Molto Alta | 0,4166 |
| Tecnologie & Features | Molto Alta | 0,4166 |

Utilizzando la normalizzazione abbiamo ottenuto dei risultati relativi all’insieme preso in esame, abbiamo construito una classifica con punteggio massimo pari a 1 e minimo pari a 0, questo non significa che l’alternativa con punteggio minore valga ‘Zero’ ma semplicemente che è la piu lontana dall’obiettivo prefissato, ovvero trovare il framework piu calzante ai nostri bisogni.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Framework | Punteggio Quality Attribute | Punteggio Quality Attribute | Punteggio Quality Attribute | Punteggio  Finale |
| Spring MVC |  |  |  |  |
| Vaadin |  |  |  |  |
| Javalin |  |  |  |  |
| Laravel |  |  |  |  |
| Symfony |  |  |  |  |
| Cake PHP |  |  |  |  |
| Code Igniter |  |  |  |  |
| AngularJs |  |  |  |  |
| Polymer |  |  |  |  |
| React.JS |  |  |  |  |
| Express.js |  |  |  |  |
| Node.js |  |  |  |  |
| Sails.js |  |  |  |  |
| Koa.Js |  |  |  |  |
| Ember.Js |  |  |  |  |
| Backbone.js |  |  |  |  |
| Vue.js |  |  |  |  |
| Django |  |  |  |  |
| Flask |  |  |  |  |
| Rails |  |  |  |  |
| ASP.NET |  |  |  |  |
| ASP.NET MVC (archived 2019) |  |  |  |  |
| Apache Struts |  |  |  |  |
| Sinatra |  |  |  |  |
| Yii |  |  |  |  |

# Caso di Studio

Il caso di studio riguarda lo sviluppo di un’applicazione web che ci permetta di visualizzare/inserire/cancellare un framework da una lista con i seguenti attributi: nome, tipo, punteggio. Sono stati scelti solo FullStack Frameworks per consentire lo sviluppo dell’applicazione in tutte le sue componenti backend e frontend.

# Related work

In questo capitolo introduciamo e discutiamo le fonti esterne che abbiamo studiato e che ci hanno portato a sviluppare la nostra analisi multi-criteriale in questo modo. Il primo articolo preso in esame è stato ***selecting a php framework for a web application project — the method and case study*** [2] in questo articolo si risolve il problema della selezione di un framework php per sviluppare un applicazione web, in questo metodo di analisi comparativa i criteri di valutazione sono Features del framework stesso, queste vendono classificate in 3 tipi: ‘Stimulants’,’Destimulants’ e ‘Nominants’ queste variabili possono essere costituite anche da sotto-variabili le quali attraverso un processo di normalizzazione verranno valutate con due metodi differenti, il ‘Non-reference-method’ tramite il quale viene creata una ‘Synthetic Feature’ sommando il il valore normalizzato e pesato delle sotto-feature; ed il ‘Reference method’ il quale consiste nel determinare un valore ottimale ed utilizzarlo come referenza per la valutazione considerando i valori che vi sia avvicinano di piu come migliori. Analogamente a questo tipo di analisi abbiamo usato la tecnica di normalizzazione delle variabili utilizzando gli stessi concetto di ‘Stimulants’ e ‘Destimulants’ per gestire le variabili che portano a risultati negativi, ed abbiamo abbracciato il ‘Non-reference method’ valutando le variabili in normalizzando i valori. Diversamente da questa analisi non esaminiamo ‘features’ ma Quality Attribute come specificato in [3]. Il problema riscontrato in questo particolare studio è il campo di soluzioni che risulta molto ristretto rispetto al nostro problema di analisi, infatti vengono analizzati solo frameworks PHP questo permette al decisore di utilizzare solo features, nel nostro caso lo spazio di soluzioni è piu ampio, si mettono in discussione ambienti di sviluppo diverso come Java, PHP, Javascript, Python ed altri, questa situazione ci porta a valutare metriche che non sono solo features ma anche altri ‘indici di qualita’ non puramente tecnici.

Un altro articolo *preso in esame è: ‘****A quantity model for controlling and measuring software quality based on the expert decision-making algorithm****’* in questo studio basato su un Multiple Criteria Decision Making

*‘Multiple criteria decision making (MCDM [[9]](#footnote-9)) is a methodology that helps decision makers make preference decisions (e.g. assessment, ranking, selection) regarding a finite set of available alternatives (courses of action) characterized by multiple, potentially conflicting attrib-utes [9,10]. MCDM provides a formal framework for modeling multi-attribute decision problems, particularly problems whose nature demands systematic analysis, including analysis of decision complexity, regularity, significant consequences, and the need for accountability [9]. Among those well-known methods, MCDM has only relatively recently been employed to evaluate software quality performance. MCDM-based decision-making is a wide method for the measuring the software quality [4,11–13]. Among those well-known evaluation methods, MCDM has been employed relatively recently to evalu-ate organizational performance and it uses a set of attrib-utes to resolve decision-making issues.’*[*p.82,* 10]

In particolare si utilizza un modello di quantità per misurare la ‘software quality’, l’aspetto interessante di questo approccio è che formalizza un metodo attraverso il quale assegnare un peso agli attributi, il problema che risolve questo articolo è che un decisore non può conoscere in anticipo il peso di un attributo si usa quindi una tecnica chiamata Analityc Hierarchic Process (AHP).

*‘Assume that we have n different and independent criteria (C1, C2, …, Cn) and they have the weights (W1, W2, …, Wn), respectively. The decision-maker does not know in advance the values of Wi, i = 1, 2, …, n, but he is capable of making pair-wise comparison between the different criteria.’ [p.82, 10].* In questo caso i pesi sono prestabiliti e vengono assegnati ad un attributo facendo un confronto a coppie tra tutte le possibili coppie di attributi.

Un approccio interessante sul MCDM di Mehmet Kabaka ET AL in [10] risolve il problema delle ambiguità del linguaggio naturale approssimando i criteri di valutazione attraverso l’utilizzo di un insieme sfocato, costruendo una scala linguistica con la quale assegnare pesi e valutare l’impatto dei criteri sull’obiettivo, anche se questo metodo è stato applicato per risolvere un problema decisionale riguardante le performance energetiche degli edifici è interessante l’idea di trovare un metodo per eliminare le ambiguità del linguaggio naturale.

Un approccio che combina la valutazione della software quality con TFN è presentato da Fu Zhi-Gang ET AL in [4], in questo articolo gli autori modellizzano il problema come un Multiple Attribute Decision Making, gli autori focalizzano lo studio su caratteristiche del processo produttivo giapponese in collaborazione con il team di progetto Hitachi sottolineando come la valutazione della qualità si in stretta relazione con il processo di sviluppo del software. Nell’articolo è presentato anche un esempio pratico del metodo proposto.La particolarità di questo approccio è la modellizzazione delle preferenze associate alle alternative come una matrice A=[m][n] in cui ogni elemento della matrice è un TFN che esprime le preferenze normalizzate dei tre Decision Maker per l’alternativa i-esima, questo approccio è molto utile nel caso in cui vi siano 3 valori di preferenza in relazione agli attributi: Lower, Modal, Upper modellizabile con i Triangular Fuzzy Number[[10]](#footnote-10).

Questa approssimazione che fa uso degli insiemi sfocati al fine di modellizzare un contesto reale è stata largamente trattata da Lehmann ET AL in [11] e da Zimmerman nel libro Fuzzy Set Theory and Its Application [12], richiami di logica fuzzy possono essere visionati online in [13]. Questa approssimazione tuttavia per la natura del nostro problema non è necessaria, in quanto è stato utilizzato un modello di valutazione basato sulla qualità software che attraverso la difinizione di semplici regole e formule è in grado di restituirci dei valori non ambigui.

Tornando allo studio condotto da Zhi-Gang ET AL in [4] gli aspetti interessanti dal nostro punto di vista vi è la modellizzazione del problema MDMA[[11]](#footnote-11) dove vengono definiti l’insieme discreto delle Alternative , l’insieme degli attributi , le informazioni riguardanti il peso degli attributi .

**Alcuni articoli correlati con i quality attribute e le metriche individuate.**

Un articolo interessante riguardante i dati estrapolati da GitHub è *“A Large Scale Study of Programming Languages and Code Quality in Github”* [14]nello stesso viene condotto uno studio riguardante la relazione tra un linguaggio di programmazione e la qualità del software, nel nostro caso utilizzeremo la medesima fonte di informazioni ma utilizzeremo metriche differenti per costruire un attributo di qualità che si basa sulla diffusione e sulla community del framework.

**Related Works riguardanti le tecnologie utilizzate per lo sviluppo di applicazioni web.**

Al fine di individuare metriche che abbiano un riscontro con tecnologie e tecniche utilizzate in contesti reali sono stati visionati anche articoli e libri che trattano nello specifico lo viluppo di applicazioni web tra questi possiamo certamente citare Bodje ET AL in [1], in questo articolo si chiarisce la terminologia e si fanno cenni riguardanti l’ingegneria del web. In questo articolo in si fa anche una distinzione del tipo di strumento che si può utilizzare per lo sviluppo di applicazioni web, e.g. Toolkit, CMS, Frameworks, inoltre è stato anche considerato l’aspetto qualitativo del software nella seguente immagine possiamo vedere come è stata definita la Software Quality da L. Olsina riproposta anche da Bodje.

****

Figura Software Quality individuata da Olsina in [15] riproposta da Bodje in [1]

Approfondendo possiamo notare che la definizione di Software Quality si basa sulla famosa norma ISO/IEC 9126, questa definizione può infatti essere vista come una mappatura a tale norma con l’omissione della qualità relativa alla portabilità, essendo ormai una norma obsoleta la noi nel nostro lavoro di ricerca consideriamo lo standard ISO/IEC 25010, durante la nostra ricerca abbiamo trovato un interessante documento della British Standard Institution che implementa in [16] lo standard ISO/IEC 25010 parte dello SQuaRE, nel nostro studio mappiamo alcune metriche con la questo Standard, altre saranno metriche personalizzate.

# Bibliografia

|  |  |
| --- | --- |
| [1] | N. N.-R. Bodje e N. G.M, «Diving in Web Engineering: Definition of Words,» *International Journal of Research in Computer Applications & Information Technology,* vol. 2, n. 3, pp. 94-102, 2014. |
| [2] | Z. Adrian e M. Marek, «SELECTING A PHP FRAMEWORK FOR A WEB APPLICATION PROJECT — THE METHOD AND CASE STUDY,» Lublin University of Technology (POLAND), March 2015. |
| [3] | F. Basciani, J. Di Rocco, D. Di Ruscio, L. Iovino e A. Pierantonio, «Automated Quality Assessment of Modeling Artifacts,» L'Aquila, May 2018. |
| [4] | F. Zhi-Gang e L. Yong, «An Approach to Software Quality Assessment with Triangular Fuzzy,» *International Journal of Advancements in Computing Technology,* vol. IV, n. 9, pp. 170-1716, 2012. |
| [5] | A. M. S. Laurent, Understanding Open Source and, Sebastopol: O’Reilly Media, Inc, 2004. |
| [6] | T. Gomez-Diaz, «Free software, Open source software, licenses. A short,» Paris, 2014. |
| [7] | A. Adhikari, «Full Stack JavaScript: Web Application - Development with MEAN,» 2016. |
| [8] | T. Reenskaug, «Model-View-Controller,» 1979. |
| [9] | T. Murat e Y. Alex, «PROFESSIONAL Java® EE Design Patterns,» 2014, p. 193. |
| [10] | M. Kabaka, E. Köseb, O. Kırılmaza e S. Burmaoglu, «A fuzzy multi-criteria decision making approach to assess buildingenergy performance,» *Elsiever,* vol. Energy and Buildings, n. 72, p. 382–389, 27 dicembre 2013. |
| [11] | I. Lehmann, W. Richard e Z. Hans-Jürgen, «Fuzzy set theory,» *Operations Research-Spektrum.,* vol. 14, n. 9, pp. 1-9, 1992. |
| [12] | Z. Hans-Jurgen, Fuzzy Set Theoryand Its Applications, New York: Springer Science+Business Media, 2001. |
| [13] | A. Vari, «Fuzzy Logica,» [Online]. Available: https://it.wikipedia.org/wiki/Logica\_fuzzy. [Consultato il giorno 16 08 2019]. |
| [14] | B. Ray, D. Posnett, V. Filkov e P. Devanbu, *A Large Scale Study of Programming Languages,* Hong Kong: Department of Computer Science, University of California, Davis, CA, 95616, USA, 2014. |
| [15] | C.-W. CHANG, D.-J. HORNG e H.-L. LIN, «A Quantity Model for Controlling and Measuring Software Quality Based on the Expert Decision-Making Algorithm,» 2009. |

fine

1. Architectural Styles and the Design of Network-based Software Architectures Roy Thomas Fielding 2000

2. Template Processor https://en.wikipedia.org/wiki/Template\_processor

3. REGOLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO E DEL CONSIGLIO del 27 aprile 2016

4. Dolière Francis Some. Web applications security and privacy. Cryptography and Security [cs.CR]. Université Côte d’Azur, 2018. English. ffNNT : 2018AZUR4085ff. fftel-01925851v2f

1. Grey relational analysis model software quality assessment with triangular fuzzy information Yang-Rui, Yang ; He-Chuang, Wang ; Yan-Hui, Xin International Journal of nowledge-Based and Intelligent Engineering Systems, 2017, Vol.21(2), pp.97-102 [Rivista Peer Reviewed]

1.1.1 Documentazione Online dei Frameworks

SpringMVC [https://spring.io/docs/](https://spring.io/docs/reference)

Laravel <https://laravel.com/docs/5.8>

Symphony [https://symfony.com/doc/current/index.html#gsc.tab=0](https://symfony.com/doc/current/index.html" \l "gsc.tab=0)

Django <https://docs.djangoproject.com/en/2.2/>

Ruby on Rails <https://guides.rubyonrails.org/>

AngularJs <https://docs.angularjs.org/api>

Node.js <https://nodejs.org/it/docs/>

Sails.js <https://sailsjs.com/documentation/reference>

Google Web Toolkit

Cake PHP <https://book.cakephp.org/3.0/en/contributing/documentation.html>

React.js <https://reactjs.org/docs/getting-started.html>

ASP.NET <https://docs.microsoft.com/it-it/aspnet/>

Vaadin <https://vaadin.com/docs> fullstack example <https://vaadin.com/start/latest/full-stack-spring>

Express <https://expressjs.com/it/guide/routing.html>

Spark <https://spark.apache.org/docs/latest/index.html>

DA APPROFONDIRE

https://www.appdynamics.com/blog/engineering/a-practical-guide-to-popular-node-js-mvc-frameworks/

View con i template http://www.corej2eepatterns.com/Patterns2ndEd/CompositeView.htm

hapi vs sails https://github.com/hapijs/hapi

https://hackernoon.com/node-js-framework-comparison-hapi-js-and-sails-js-e31f05cc1da

• Four key metrics tecnica di valutazione

• Lista di Framework diffusi https://www.g2.com/categories/web-frameworks

8.1.1 Esempi e tutorial

• MVC MEAN https://code.tutsplus.com/tutorials/build-a-complete-mvc-website-with-expressjs--net-34168 altro https://itnext.io/a-new-and-better-mvc-pattern-for-node-express-478a95b09155

• MVC NODE EXPRESS https://www.youtube.com/watch?v=GLL3UIPgU\_o

TREND E STRATISTICHE

• https://trends.builtwith.com/framework

• http://hotframeworks.com/ punteggio github

• https://trends.google.com

• https://insights.stackoverflow.com/survey/2019

• https://insights.stackoverflow.com/survey/2018

• https://stackshare.io/spring#description

• https://en.wikipedia.org/wiki/Comparison\_of\_JavaScript\_frameworks

documentazione Frameworks

spring https://spring.io/docs

Laravel https://laravel.com/docs

Node.js https://nodejs.org/it/docs/

Angular https://docs.angularjs.org/api

Polymer https://polymer-library.polymer-project.org/3.0/api

Altri Strumenti e Librerie

SQL con JAVASCRIPT sequelize.js, offre ORM in JS utilizzabile in NODE.JS https://sequelize.readthedocs.io/en/v3/

Appunti provvisori

• Architettura nei Criteri di Selezione?

• API?

• Template Engine supportati (https://en.wikipedia.org/wiki/Comparison\_of\_web\_template\_engines) https://en.wikipedia.org/wiki/Mustache\_(template\_system)

Da Leggere

https://hackr.io/blog/tag/back-end-development

https://hackr.io/blog/tag/web-development/page/2

https://hackr.io/blog/why-should-you-learn-angular-in-2018

https://hackr.io/blog/django-vs-laravel

https://hackr.io/blog/10-best-javascript-frameworks-2019

1. Cern 1989 <http://info.cern.ch/Proposal.html> [↑](#footnote-ref-1)
2. Contenent Management System: è uno strumento software che permette al web master di operare su un’applicazione web svincolandolo da conoscenze tecniche. [↑](#footnote-ref-2)
3. I Toolkit sono insiemi di classi , librerie e funzioni, per lo sviluppo di applicazioni. [↑](#footnote-ref-3)
4. Model View Controller. [↑](#footnote-ref-4)
5. Product Quality Model [↑](#footnote-ref-5)
6. Dependency: Dependent from / Repository dependant by /Package dependant by [↑](#footnote-ref-6)
7. Accesso ai dati su GitHub.com il 31/07/2019 [↑](#footnote-ref-7)
8. Periodo analizzato un mese (Luglio 2019) [↑](#footnote-ref-8)
9. MCDM: Multiple Criteria Decision Making [↑](#footnote-ref-9)
10. Triangular Fuzzy Number è una struttura matematica formata da una tripla e.g.

    i tre numeri possono rappresentare 3 valori di preferenza per un attributo. [↑](#footnote-ref-10)
11. Il Multiple Attribute Decision Making è un problema decisionale che basa l’analisi sugli attributi di un artefatto. [↑](#footnote-ref-11)