

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea in Informatica per il Management

**STUDIO SUI PRINCÌPI PER FACILITARE
L'ADOZIONE DI UN LINGUAGGIO
DI PROGRAMMAZIONE ATIPICO**

Relatore:
Chiar.mo Prof.
DAVIDE SANGIORGI

Presentata da:
CRISTIAN ROMANELLO

Correlatori:
Dott. SAVERIO GIALLORENZO
Prof. MARCO RUFFINO

Sessione III
Anno Accademico 2017/2018

Ai miei genitori, alla mia famiglia

Abstract

Il *Service-Oriented Applications* (SOA) è un paradigma che sta prendendo sempre più piede, al fine di rispondere in modo più efficace all'esigenza di costruire applicazioni distribuite. In particolare, un nuovo stile architetturale orientato ai servizi, detto "a microservizi", sta acquisendo sempre più consensi nell'ambito industriale. Il focus di questo nuovo paradigma sono i microservizi.

JOLIE è un linguaggio di programmazione ad alto livello, nato nel 2006, progettato per creare microservizi e coordinarli. Esso è stato sviluppato all'interno del progetto europeo *SENSORIA* ed è interamente realizzato in Java e distribuito in formato Open source.

Questa nuova tecnologia ha molteplici vantaggi e potenzialità, i quali da soli non sembrano sufficienti per spingere ad una adozione massiva del linguaggio, sia in termini di utilizzatori (prevalentemente aziende) sia rispetto le dimensioni della comunità di utilizzatori e sviluppatori.

Questo elaborato riporta l'esperienza acquisita nell'analisi delle criticità legate alla comunicazione del progetto *JOLIE*, presentando la definizione di una strategia per migliorarla e la sua implementazione.

Introduzione

Al giorno d'oggi la maggior parte dei sistemi informatici industriali è sviluppata come un'applicazione distribuita. A livello astratto, tale architettura consiste in un insieme di processi interconnessi fra loro le cui comunicazioni avvengono attraverso lo scambio di messaggi.

Fino a tempo recenti, il principale stile architetturale seguito per sviluppare questi sistemi è il modello monolitico, il quale considera l'applicativo come un'intera singola entità. Questo approccio comporta notevoli svantaggi nel caso di sistemi dalle grandi dimensioni, in particolar modo nel momento in cui si devono effettuare manutenzioni, implementazioni di nuove funzionalità ed eventualmente scalare il sistema.

I **microservizi** rappresentano un'importante evoluzione del sistema monolitico adottato finora. Martin Fowler è un informatico e ingegnere britannico di software, autore di numerosi libri e considerato fra i massimi esperti nei settori della programmazione orientata agli oggetti. Egli è uno dei pionieri del modello a microservizi ed afferma che:

“Lo stile architetturale a microservizi è un approccio allo sviluppo di una singola applicazione come insieme di piccoli servizi, ciascuno dei quali viene eseguito da un proprio processo e comunica con un meccanismo snello, spesso una API HTTP.” [1]

I microservizi, proprio grazie al loro approccio compositivo, offrono molteplici vantaggi rispetto ai sistemi monolitici:

- Ogni microservizio si propone all'esterno come una *black-box*, infatti espone

solo una *Application Programming Interface* (API)¹. Tale astrazione permette di non specificare i dettagli implementativi delle singole funzionalità offerte, limitando le ripercussioni che aggiornamento e manutenzione di un singolo microservizio hanno sulle funzionalità dell'intera architettura. In contrasto, ogni aggiornamento di un sistema monolitico richiede di terminare l'esecuzione del sistema, rimpiazzarlo con la sua nuova versione e rimetterla in esecuzione;

- Velocizzazione dei tempi di rilascio del software, permettendo di reagire più velocemente alle esigenze del mercato. Disporre di un processo di *deployment*² veloce consente di poter aggiungere o modificare funzionalità di un sistema software in modo più efficace ed efficiente rispetto ad un monolite;
- Facilitazione della sperimentazione di nuove tecnologie. La principale difficoltà che spesso impedisce l'adozione di una nuova tecnologia consiste nel rischio associato all'utilizzo di un qualcosa di nuovo e del quale si ha generalmente poca esperienza. Riducendo questo rischio ad una piccola area di un sistema software, la nuova tecnologia è più facilmente testabile ed adottabile;
- Migliori performance grazie all'utilizzo di tecnologie "ad hoc". Ogni microservizio, essendo indipendente dagli altri, può implementare la miglior tecnologia per raggiungere il suo specifico obiettivo; questo indipendentemente dalle tecnologie implementate dagli altri microservizi;
- Resilienza. Quando un componente non funziona, non è automatico che tutto il sistema software smetta di funzionare. In molti casi è possibile isolare il problema ed intervenire mentre il resto del sistema continua a funzionare, cosa non possibile in un'architettura monolitica;

¹Una API (interfaccia di programmazione di un'applicazione) consiste in una serie di convenzioni adottate dagli sviluppatori software per definire il modo con il quale una determinata funzione di un'applicazione deve essere richiamata. Esse permettono, inoltre, di espandere le funzionalità di un programma, favorendo la comunicazione con altre applicazioni.

²Il termine *deployment* indica la consegna o rilascio al cliente, con relativa installazione e messa in funzione, di una applicazione o di un sistema software, tipicamente all'interno di un sistema informatico aziendale.

- Scalabilità. In generale risulta molto più semplice ed economico scalare un microservizio rispetto ad un sistema software monolitico di grandi dimensioni.

I vantaggi appena elencati sono ben compresi dagli esperti del settore, ma quasi sempre questa comprensione non si traduce poi in un'effettiva adozione della nuova tecnologia.

L'obiettivo di questa tesi è quello di analizzare le criticità legate alla comunicazione del progetto JOLIE e definire una strategia per migliorarla. La fase che ha richiesto più tempo è stata proprio quella in cui si è analizzato come poter trasmettere l'utilità di adottare JOLIE in modo più efficace rispetto alla metodologia attuale. Per svolgere al meglio questa attività ci si è basati anche su studi afferenti l'area cognitiva e la comprensione degli esseri umani.

La tesi è stata strutturata in 5 capitoli.

Nello specifico, nel **capitolo 1** viene introdotta la situazione attuale, dove si è effettuata un'analisi del problema della comunicazione di JOLIE dal punto di vista cognitivo.

Nel **capitolo 2** viene approfondito il tema di come la comunicazione debba essere modulata a seconda del profilo a cui è rivolta.

Il **capitolo 3**, invece, descrive i dettagli (corredati da esempi) dell'intera struttura della comunicazione, suddivisa per livelli.

A seguire, nel **capitolo 4** si approfondiscono più in dettaglio i principi per l'adozione di un linguaggio di programmazione.

A conclusione, nel **capitolo 5** è riportato e descritto il *mockup* del sito di JOLIE realizzato.

Indice

Introduzione	i
1 Situazione attuale	1
1.1 Analisi cognitiva	2
1.2 Possibile soluzione	6
1.2.1 Situazione futura	6
1.2.2 Analogia	7
1.2.3 <i>Bias</i> “positivi”	9
2 Analisi Target	13
2.1 Obiettivo	13
2.2 <i>Target</i>	13
2.3 Modalità	14
2.4 Analisi delle caratteristiche-tipo dei ruoli	15
2.5 Storia di “aggancio”	18
3 La Strategia	25
3.1 JOLIE Unitario e Plurivoco	26
3.1.1 <i>Semiotic Square</i>	26
3.1.2 Significato letterario	30
3.1.3 Significato per JOLIE	31
3.2 <i>Roadmap</i>	32
3.2.1 I° Livello	34
3.2.2 II° Livello	37

3.2.3	III° Livello	44
3.2.4	IV° Livello	45
3.2.5	V° Livello	47
4	Principi per l'Adozione di un Linguaggio di Programmazione	49
4.1	Il cambiamento	49
4.2	Le barriere	50
4.3	Diffusione dell'innovazione	51
5	Sito realizzato	61
5.1	Homepage	61
5.2	Download <i>workflow</i>	68
5.3	Technical Insights	72
	Conclusioni	77
	Bibliografia	79

Elenco delle figure

1.1	Il processo decisionale secondo D. Kahneman	2
1.2	Attuale impatto cognitivo della comunicazione di JOLIE	5
1.3	Futuro flusso di costi cognitivi della comunicazione di JOLIE	7
2.1	Dipendenze fra i <i>target</i> e le loro relazioni con JOLIE	14
2.2	Storia Businessman: situazione iniziale	19
2.3	Storia Businessman: situazione di “crisi”	20
2.4	Storia Businessman: soluzione JOLIE	21
3.1	Rappresentazione analitica del <i>Semiotic Square</i>	27
3.2	Esempio di utilizzo del <i>Semiotic Square</i>	28
3.3	Utilizzo del <i>Semiotic Square</i> per “unitario” e “plurivoco”	29
3.4	Strategia completa	32
3.5	Analogia OOP - JOLIE	35
3.6	Il Viaggio dell’Eroe	38
4.1	La <i>funzione valore</i> della teoria del prospetto	55
4.2	La <i>curva di Rogers</i> nella teoria della diffusione dell’innovazione	57
5.1	Menu e <i>slogan</i> della Homepage	62
5.2	Pagina spiegazione <i>slogan</i>	63
5.3	Sezione Homepage: Esplorazione profili	64
5.4	Sezione Homepage: <i>Focus</i> sui microservizi	65
5.5	Sezione Homepage: <i>Top Stories</i>	65
5.6	Sezione Homepage: <i>Top Technical Tutorials</i>	66

5.7	Sezioni Homepage: <i>Technical Insights</i> e <i>Recent News</i>	67
5.8	Sezione Homepage: <i>Academia</i>	68
5.9	Pagina <i>Download</i>	69
5.10	Pagina <i>Install Guide</i>	70
5.11	Pagina <i>The IDE</i>	71
5.12	Pagina <i>The IDE</i>	72
5.13	Pagina <i>Technical Insights</i>	73
5.14	Pagina <i>Technical Insights</i> : il menu	74

Capitolo 1

Situazione attuale

A partire dalla nascita di JOLIE, diverse sono state le conferenze tenute dai co-fondatori, tra cui il Dott. Ing. Claudio Guidi. Tra le più importanti si cita il “*Meeting on Microservices 2016 - visione, processi e tecniche*”, nonché “il primo appuntamento in Italia sui microservizi tra industria, innovazione e ricerca”¹. La platea spesso comprendeva le potenzialità dei microservizi e, in particolare, della tecnologia JOLIE, risultandone anche entusiasta. A seguito di queste conferenze non seguiva, però, quasi mai un’effettiva adozione della nuova tecnologia.

Quest’ultima è stata fortemente validata, porta ad ottimi risultati, risulta essere estremamente competitiva, però non è così diffusa e adottata quanto sperato. Non emergendo rilevanti motivazioni dal punto di vista tecnico, il problema risale, molto probabilmente, nella **comunicazione**. Questo non significa che la comunicazione effettuata fino ad ora non sia stata efficace, ma potrebbe semplicemente necessitare di una rivisitazione. Ma in quale modo?

Prima di giungere ad una possibile risposta al problema, è necessario analizzare il motivo per cui la comunicazione fino ad ora adottata non risulta essere vincente.

¹[<http://www.mokabyte.it/mblog/wp-content/uploads/2016/12/MoM2016.pdf>]

1.1 Analisi cognitiva

Una possibile analisi può essere effettuata dal punto di vista cognitivo rispetto alle persone che partecipano alle conferenze. Il risultato, riportato dal Dott. Claudio Guidi, consiste nella difficoltà di affrontare un salto tecnologico troppo elevato, e questo porta, una volta spiegato, prevalentemente ad emozioni negative. Nonostante la razionalità conduca le persone a determinare JOLIE (e più in generale i microservizi) come una tecnologia valida, l'aspetto emotivo vince su tutto. Esiste una spiegazione per questo fenomeno: il *Decision Making*.

Il *Decision Making* (processo decisionale) consiste nel processo che porta un individuo oppure un gruppo a prendere una decisione piuttosto che un'altra. Daniel Kahneman schematizza il processo decisionale come segue:



Figura 1.1: Rappresentazione grafica del processo decisionale secondo D. Kahneman[2]

L'informazione recepita da una persona attraversa, in primo luogo, il *System 1*. Esso è prevalentemente dominato dall'istinto, infatti risulta essere veloce, associativo ed emozionale. Nel momento in cui si vuole far comprendere un concetto ad una persona, questa è la prima barriera da affrontare. *System 1* è molto importante perché, anche se in modo involontario, condiziona la successiva (ed ultima) tappa: *System 2* (ragionamento). *System 1* rappresenta quindi una sorta di passaggio obbligatorio per raggiungere il pensiero razionale.

Stando all'interpretazione di Kahneman, le informazioni più facilmente assimilabili da *System 1* sono veloci, associative ed emozionali. Esse possono essere generalmente ottenute semplificando il concetto più complesso, ma al tempo stesso bisogna accertarsi di non alterarne la correttezza. Più la forma comunicativa sarà chiara, semplice e più in generale cognitivamente *effortless* (senza sforzo), più facilmente sarà poi assimilabile da *System 1*. Queste semplificazioni, appositamente create, e quelle di per sé effettuate dagli individui, portano ad imbattersi però in molteplici vincoli e *bias* cognitivi.

I *bias* consistono in giudizi sviluppati dalle persone sulla base delle interpretazioni di informazioni in loro possesso. Queste ultime possono anche essere non logicamente o apparentemente connesse tra loro, e possono portare, quindi, ad un errore di valutazione o ad una mancanza di oggettività nel giudizio. Dato il funzionamento della cognizione umana, i *bias* non sono mai completamente eliminabili, ma possono essere opportunamente affrontati, correggendo così la percezione (distorta) al fine di diminuire gli effetti negativi. In generale i *bias* derivano da esperienze individuali, contesti culturali e di credenze, giudizi altrui, schemi mentali e paure. La deformazione del giudizio deriva da molteplici cause, spesso anche interconnesse fra loro:

- *Problemi di elaborazione*: la naturale tendenza alla semplificazione dei problemi, prima di avviare un processo di scelta, comporta la riduzione intenzionale della quantità di informazione disponibile;
- *Effetto alone (Halo effect)*: la prima impressione significativa, in termini di impatto emozionale (positivo o negativo), dirige il successivo processo di interpretazione e scelta (p.e. attribuzione di un giudizio);

- *Insufficiente adeguamento dell'ancoraggio*: una volta espressa una stima iniziale in merito ad un evento (a partire da esperienze passate, da altre informazioni disponibili o casualmente), le persone tendono ad ancorare a questo giudizio ogni successiva valutazione, realizzando solo dei minimi aggiustamenti;
- *Confirmation bias*: le persone tendono a ricercare informazioni che confermino le proprie certezze, piuttosto che prove che le falsifichino;
- *Egocentrismo*: ricordare un evento in modo che soddisfi dei criteri di autostima;
- *Stereotipi*: si verifica quando vengono aggiunti o distorti elementi di un ricordo, in modo tale che questi non siano più coerenti con uno stereotipo familiare.

Queste sono solo alcune delle cause della deformazione di giudizi. Ogni persona ne è affetta, seppur in misura estremamente variabile. Solo dopo aver attraversato tutte queste deformazioni, l'informazione ricevuta in input arriva a *System 2* (ragionamento). Quest'ultimo è, per definizione, lento, cognitivamente costoso e retto da regole. Qui vengono prese le decisioni; il fatto che esse siano prese però in modo puramente razionale e senza distorsioni, come visto precedentemente, è estremamente raro, se non quasi impossibile.

Dal momento che la nuova tecnologia comunicata rappresenta un cambiamento dirompente della realtà esistente, tutti i *bias* cognitivi vengono amplificati, rendendo la comunicazione ancora più difficile e delicata allo stesso tempo.

Per la trasposizione di questo sistema di ragionamento in un grafico, ci si può basare sul valore e sul costo cognitivo percepiti da una persona, relativamente alla nuova tecnologia JOLIE.

Il *valore percepito* si riferisce alla sola percezione di esso per una persona; il *costo cognitivo percepito* corrisponde allo sforzo mentale che un individuo deve compiere (volontariamente o meno) per assimilare un'informazione.

Il grafico è diviso orizzontalmente da una linea arancione, la quale costituisce la *Soglia di Interesse*. Essa rappresenta il minimo interesse da scaturire in una persona affinché l'informazione ottenga sufficiente attenzione per essere anche solo vista,

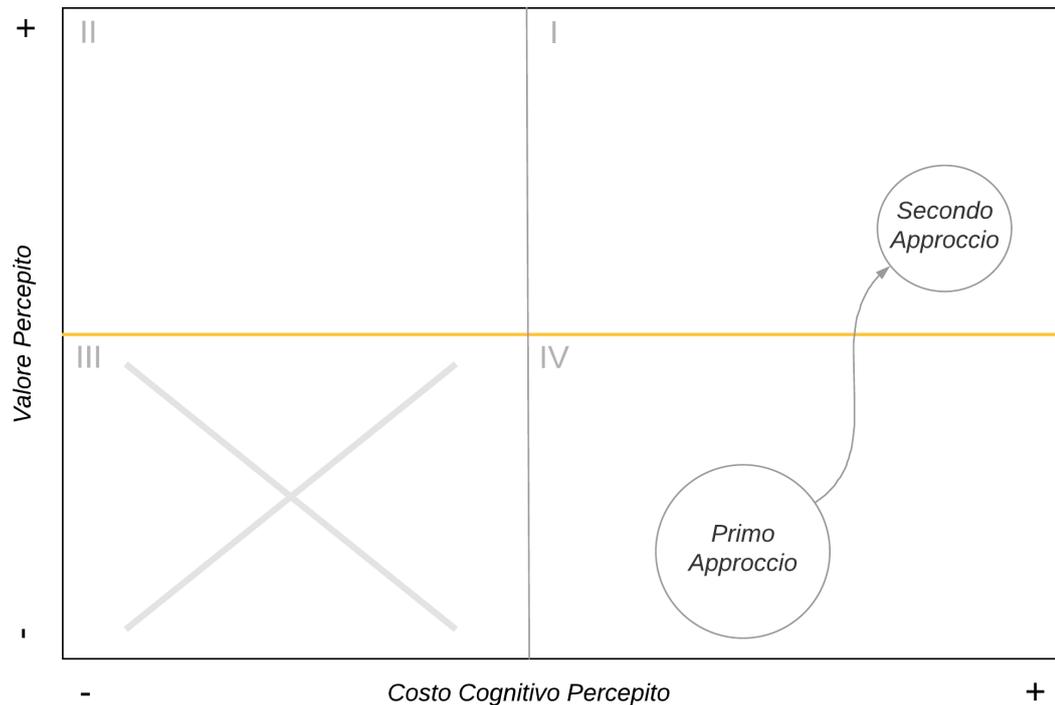


Figura 1.2: Rappresentazione grafica dell'attuale impatto cognitivo della comunicazione di JOLIE

udita o letta per un breve periodo di tempo. Il quadrante III del grafico non rappresenta un valido scenario nel quale posizionarsi, né come punto di partenza né (e soprattutto) come punto di arrivo. Questo perché esso costituisce prevalentemente l'area dell'obbligo, ovvero attività (spesso) di basso valore percepito e relativo basso costo cognitivo.

L'attuale situazione è caratterizzata da un primo approccio che avviene nel quadrante IV dove, a fronte di un discreto costo cognitivo, si inizia a percepire, seppur in piccole dosi, il valore della nuova tecnologia. Il problema è che solo a fronte di un importante impegno (e quindi costo) cognitivo, la persona può veramente comprendere il valore aggiunto di JOLIE. Questo si verifica nel secondo approccio. Da questa situazione iniziale nasce la necessità di modificare l'approccio comunicativo di JOLIE, rendendolo cognitivamente poco costoso (con il giusto equilibrio) e, al tempo stesso, aumentare il valore percepito.

1.2 Possibile soluzione

1.2.1 Situazione futura

Esiste un aspetto molto importante della comunicazione: il **costo cognitivo**. Nella precedente rappresentazione grafica, gli approcci mostrati erano solamente due, il che significa che una persona, per raggiungere il secondo approccio, deve sostenere un unico e rilevante aumento di costo cognitivo. Una tecnica per alleggerire il carico (cognitivo) sostenuto consiste nell'aumentare il numero di "passaggi", così come anche il numero di approcci intermedi. La somma del costo sostenuto per attraversare tutti i passaggi potrebbe risultare maggiore (analiticamente) rispetto alla situazione con i soli due approcci, ma il carico cognitivo sostenuto per attraversare ogni passaggio diminuisce notevolmente; questo facilita il superamento (incrementale) di ogni singolo approccio. Una rappresentazione grafica, appositamente costruita, è la seguente.

L'analisi di questo capitolo si sofferma esclusivamente sulla struttura del grafico proposto, lasciando la descrizione del contenuto dei vari approcci (da qui in seguito chiamati *Livelli*) ai capitoli successivi.

La struttura comunicativa, quindi, cambia radicalmente rispetto alla situazione precedente (con i soli due approcci). In questo scenario si evita comunque il quadrante III, poiché non di interesse, ma anche il quadrante IV, perché all'interno di esso le informazioni vengono recepite come di basso valore.

Gli approcci utilizzati fino ad ora, vengono sostituiti dai **Livelli**. Essi consistono, dal punto di vista teorico, sempre nello stesso oggetto, ovvero una "tappa" fondamentale per raggiungere le successive; l'accento viene però posto sulla scalabilità di tali livelli, infatti la visione che si è cercato di creare è proprio quella di una sorta di "percorso guidato" a livelli incrementali. Quest'ultimo inizia dal Livello I, il primo con il quale una persona si avvicina, per poi proseguire, a seconda della tipologia di interessi, con i successivi livelli. Per attraversare ognuno di essi, è necessario percorrere, a differenza del precedente modello, solo un breve "passo" e con un contenuto costo cognitivo. L'intero percorso posa al di sopra della soglia di interesse (evidenziata in

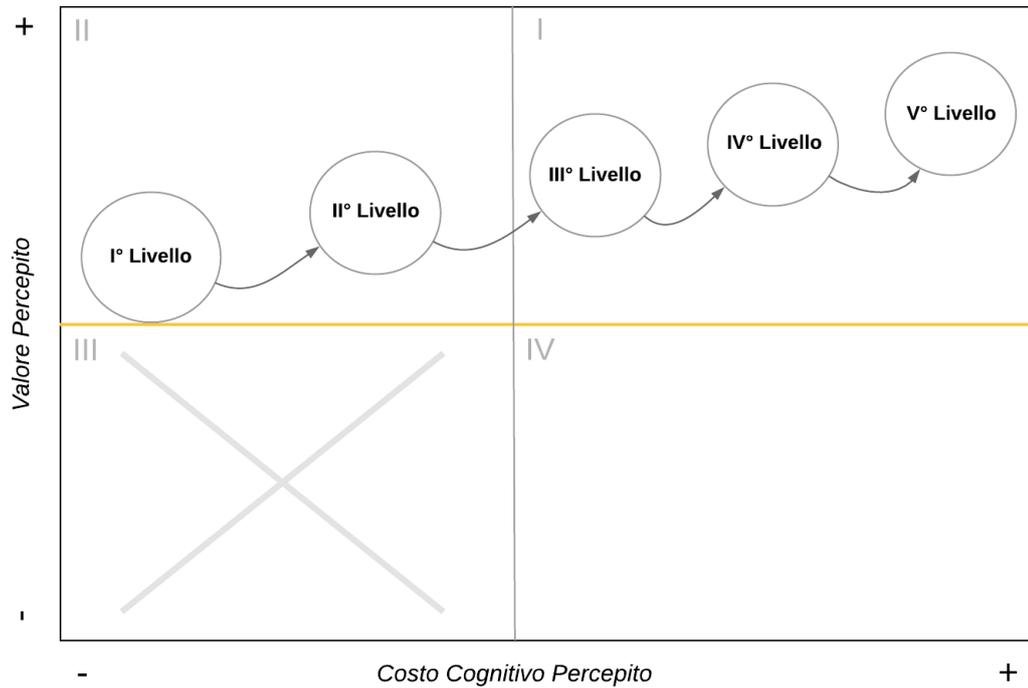


Figura 1.3: Rappresentazione grafica del futuro flusso di costi cognitivi della comunicazione di JOLIE

arancione nel grafico), per far sì che tutte le informazioni recepite, a prescindere dal livello, abbiano un **alto valore percepito**. Il costo cognitivo (totale) è superiore al modello con i due approcci, ma venendo suddiviso in tanti piccoli passi, è percepito (nella sua totalità) come inferiore. Questo grazie al fatto che ogni passaggio, fra un livello e il successivo, poggia sul costo sostenuto per arrivare fino a tale livello.

1.2.2 Analogia

Analizzando il comportamento di *System 1* e *System 2* si evince dunque l'importanza del **primo approccio** (Livello I) rispetto la comunicazione della nuova tecnologia, e quanto essa possa condizionare positivamente (o negativamente) le successive scelte "razionali". L'elaborazione di un'ottima comunicazione per il Livello I fa sì che l'intera catena di livelli diventi, nel suo complesso, più accessibile.

Precedentemente si sono elencate le caratteristiche affinché un messaggio venga facilmente “accettato” da *System 1*: velocità, associatività ed emozionalità. Quest’ultima risulta essere la componente più rilevante ed efficace, ma anche quella più complessa da costruire (dal punto di vista comunicativo). Riguardo alle emozioni provate da coloro che partecipano alle conferenze su JOLIE, esse sono prevalentemente negative; questo a causa del dirompente impatto che la nuova tecnologia porta con sé. La necessità di trasformare queste emozioni da negative in positive diventa, quindi, la priorità. La modalità ritenuta più efficace è stata l’**analogia**².

In particolare, l’analogia permette di creare le basi per un nuovo concetto (i microservizi e JOLIE) partendo dal confronto con qualcosa che le persone del settore conoscono già: l’*Object-oriented programming* (OOP)³. Lo scopo dell’analogia consiste nel comunicare la possibilità di raggiungere lo stesso risultato ottenuto dall’OOP tramite un approccio migliore (JOLIE e i microservizi).

L’analogia da costruire deve, però, essere al contempo: corretta (dal punto di vista rappresentazionale) e stimolante (dal punto di vista cognitivo). Se *System 1* viene opportunamente stimolato, allora *System 2* sarà più propenso ad elaborare le informazioni ricevute, ma non solo. Esso lavorerà, inoltre, in modo subcosciente rispetto ai *bias* entrati in fusione in *System 1*: per questi motivi è estremamente importante che l’analogia sia ad effetto e sufficientemente intrigante da stimolare ulteriore curiosità.

Tutto questo però non è sufficiente, poiché è necessario, per ottenere un risultato ottimale, fornire continue conferme a *System 2* durante la sua elaborazione. Queste conferme saranno meno “ad effetto” rispetto all’analogia iniziale, e consisteranno prevalentemente in spiegazioni di natura tecnico/analitica, ma costantemente presenti in tutti i livelli, seppur in diverse forme e modalità.

²L’analogia è il rapporto che la mente coglie fra due o più caratteristiche che possiedono qualche tratto comune nella loro costituzione, nel loro comportamento o nei loro processi.

³L’*Object-Oriented Programming* (programmazione orientata agli oggetti) è un paradigma di programmazione che permette di definire oggetti software in grado di interagire fra loro attraverso lo scambio di messaggi.

1.2.3 *Bias* “positivi”

All'interno dell'Analisi Cognitiva, si sono elencati alcuni *bias* cognitivi che influenzano il processo decisionale. Tuttavia, ne esistono alcuni considerati (dal nostro punto di vista) come “positivi” e che inconsciamente favoriscono il passaggio ad una nuova frontiera (in questo caso tecnologica), facilitando il superamento di alcune barriere psicologiche.

- *Effetto della superiorità dell'immagine visiva*: uno stesso elemento è più facilmente richiamabile alla memoria se è stato presentato in forma visiva rispetto che in forma scritta.

La comunicazione di JOLIE deve essere quindi, in parte ed in fase iniziale, espressa tramite immagini;

- *Effetto della positività*: le persone richiamano più facilmente ricordi piacevoli rispetto quelli negativi.

JOLIE deve evocare emozioni positive, e non negative;

- *Effetto della rilevanza per il Sé*: le informazioni che sono collegate alla propria persona e alle proprie attività sono ricordate meglio di informazioni simili ma riferite ad altre persone.

La comunicazione dovrà differenziarsi (per modalità e contenuti) in base alla persona a cui è esposta;

- *Effetto von Restorff*: un elemento che si distingue dagli altri per qualche caratteristica è ricordato meglio[3].

JOLIE e i microservizi si distinguono ampiamente rispetto l'OOP e i sistemi monolitici;

- *Hyperbolic discounting*: date due ricompense simili, gli umani dimostrano una preferenza per quella che arriva temporalmente prima, rispetto a quella futura. Questo perché la ricompensa nel lontano futuro viene “scontata” al momento della valutazione, riducendone così drasticamente il valore. Più la ricompensa è futura e meno valore viene ad essa attribuito.

JOLIE permette di creare microservizi riducendo il numero di righe di codice necessarie, così facendo il tempo impiegato per la sua stesura è minore.

- *Action bias*: le persone tendono ad agire anche quando l'azione è meno vantaggiosa dell'omissione.

Questo *bias* è stato studiato da Fagerlin, Zikmund-Fisher e Ubel in un loro esperimento[4]: nel caso di una diagnosi di cancro, i pazienti preferivano sottoporsi a trattamenti (azione), piuttosto che a semplici controlli (inazione), anche se i trattamenti risultavano più dannosi o meno efficaci dell'inazione. Tuttavia, una variabile importante che potrebbe aver influenzato i risultati dello studio è la gravità del cancro dei soggetti.

Questo tipo di *bias* è confermato anche da un altro studio, effettuato da Bar-Eli, Azar, Ritov, Keidar-Levin e Schein [5]. In esso si osserva il fatto che i portieri di calcio durante i rigori, pur sapendo che la strategia ideale per loro sarebbe rimanere al centro della porta (inazione), molto spesso, invece, si tuffano in una delle due direzioni laterali (azione).

Quest'ultimo *bias* è il più controverso di tutti, ma anche il più importante; necessita quindi di un ulteriore approfondimento.

Lo studio effettuato sui pazienti affetti da cancro e la relativa analogia non vogliono in alcun modo mancare di rispetto o dimostrarsi insensibili di fronte la drammaticità degli eventi; hanno, infatti, puramente lo scopo di analizzare il comportamento umano in situazioni richiedenti un altissimo costo cognitivo.

Spesso aziende e imprese si trovano ad apportare continue modifiche alle applicazioni da loro sviluppate (secondo i principi dell'OOP). Questa situazione può diventare difficile da sostenere nel tempo, se il codice non è stato scritto rispecchiando nel migliore dei modi i principi della progettazione software. Tuttavia, anche rispettando al meglio tali principi, l'azienda può trovarsi di fronte a sfide che non aveva previsto, come ad esempio l'inserimento di un nuovo componente (interno oppure esterno all'applicazione o all'azienda) e la necessità di comunicare e cooperare con esso. Questi eventi vengono affrontati in modo non ottimale, semplicemente perché non possono essere (sempre) previsti. Il paragone a cui si può pensare è proprio

quello di un'impresa che si trova in questa disperata situazione, dove continui cambiamenti (interni ed esterni) rendono l'applicativo sempre più complesso e difficile da mantenere nel tempo.

La soluzione che si propone a tale problema è proprio JOLIE, il quale permette di creare applicazioni come insieme di tanti piccoli microservizi, ognuno indipendente (dal punto di vista implementativo) dagli altri.

L'aspetto cognitivo e psicologico da analizzare di questo fenomeno è il fatto che le persone, a fronte di una grave malattia, così come le imprese (formate anch'esse da esseri umani) in situazioni non più sostenibili, si trovano entrambi in circostanze che determinano la loro possibilità di vita futura. Il comportamento generalmente adottato consiste nell'utilizzo di una "cura" (nel caso delle aziende, JOLIE), che permetta la sopravvivenza. Ovviamente questo caso è volutamente estremizzato, al solo scopo di far comprendere come le persone reagiscono di fronte a situazioni (umane e aziendali) drammatiche.

Lo studio effettuato sui portieri di calcio, invece, evidenzia come le persone possano reagire (apparentemente) in modo non convenzionale. Focalizzando l'attenzione sul portiere, nel momento del calcio di un rigore, se questi rimanesse fermo al centro, avrebbe sicuramente la possibilità di scegliere da quale lato buttarsi, ovvero lo stesso della direzione della palla. Questo però non è un comportamento adottato poiché la velocità della palla è così elevata che per il portiere è quasi impossibile riuscire, in quella frazione di secondo, a cogliere la traiettoria della palla e soprattutto a muovere il proprio corpo in tale direzione.

La scelta dei migliori portieri ricade, invece, nella valutazione di quale possa essere la più probabile direzione della palla, e comportarsi di conseguenza. Se la loro valutazione si rivelasse corretta, le probabilità di una parata aumenterebbero esponenzialmente.

Questo paragone può rispecchiarsi in un'azienda che vuole "buttarsi" e cogliere le opportunità del futuro. Se si interpreta l'azienda (portiere) e il futuro (palla), tanto più brava sarà l'azienda a cogliere la direzione della palla (opportunità), tanto maggiori saranno i guadagni ottenibili da questa mossa (benefici economici, posizionamento di mercato e non solo).

Capitolo 2

Analisi Target

In questo capitolo verranno delineati i principali ruoli dei *target* a cui JOLIE si rivolge. Si analizzeranno i caratteri distintivi e i punti di contatto che questi possono avere con JOLIE. Da tale analisi verranno successivamente sviluppate delle strategie di comunicazione che vanno a chiarire perché JOLIE è una risorsa strategia per ogni ruolo individuato.

2.1 Obiettivo

L'obiettivo della comunicazione di JOLIE consiste nel far intuire prima, e comprendere poi, il valore aggiunto della nuova tecnologia. La comunicazione, però, deve essere modulata nel contenuto e nella forma a seconda della tipologia alla quale la persona appartiene.

2.2 *Target*

La comunicazione è riferita principalmente a *target* (obiettivi) selezionati: dalle persone del settore (non necessariamente esperti) fino ad attraversare l'intera scala gerarchica aziendale. In particolare, si sono individuati i seguenti profili:

- Businessman: l'uomo d'affari cui compete la gestione economica aziendale;

- IT Manager: supporta il Businessman nelle scelte sistemistiche che riguardano l'azienda e nello sfruttamento della tecnologia a sostegno dello sviluppo e della gestione del *business*;
- Tecnico: implementa e mantiene le scelte approvate dall'IT Manager e si occupa dello sviluppo di progetti informatici
 - Esperto: conosce molto bene molteplici tecnologie e lavora nel settore da diverso tempo;
 - Beginner: possiede basi limitate e tipicamente non ha grande esperienza professionale.

Il Beginner non viene analizzato nel dettaglio poiché può corrispondere ad un caso specifico del Tecnico Esperto.

Di seguito si illustra un grafico che esprime le dipendenze fra i *target* appena elencati e le rispettive relazioni con JOLIE.

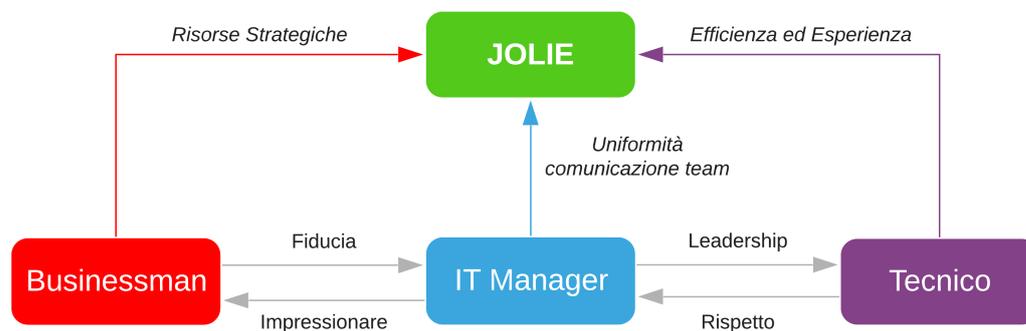


Figura 2.1: Rappresentazione grafica delle dipendenze fra i *target* e le loro relazioni con JOLIE

2.3 Modalità

La principale modalità attraverso la quale JOLIE viene comunicato consiste nel rispettivo sito web.

All'interno di esso ogni singolo profilo (*target*) deve percorrere, attraverso la navigazione, una diversa **storia**, appositamente costruita e narrata secondo gli **obiettivi** e le specifiche **necessità**.

Il singolo *target*, leggendo la storia per lui pensata, deve poter provare empatia affinché essa lo coinvolga veramente. Questo avviene (inconsciamente) attraverso l'identificazione della **vision** e delle **fragilità** caratteristiche del proprio ruolo. Per *vision* si intende l'insieme di ideali, valori e aspirazioni di ogni *target*. In particolare modo, l'essere umano sviluppa più empatia rispetto ai propri punti deboli (fragilità), anziché ai punti di forza. Le storie devono, quindi, tener conto sia della *vision* che delle fragilità. Inoltre, è necessario esplicitare il diretto valore aggiunto che JOLIE porta con sé (rinforzando quindi le fragilità).

2.4 Analisi delle caratteristiche-tipo dei ruoli

Una volta definiti i ruoli della comunicazione e le modalità attraverso le quali attuarla, è necessario analizzare più in dettaglio ogni singolo profilo. Lo scopo, per ognuno di essi, consiste nell'identificare una descrizione del ruolo, la *vision*, le criticità e i punti di adesione con JOLIE.

Businessman

La figura del Businessman è quella posta al vertice della catena decisionale (dal punto di vista aziendale), ma per quanto riguarda le scelte tecnologiche, si affida all'IT Manager, il quale a sua volta si rivolge ai Tecnici per la realizzazione.

Vision *“Il futuro è un'incertezza continua, ma pieno di opportunità. Devo saperle cogliere al momento giusto e sfruttarle al massimo. Il mio compito consiste anche nello scoprire nuovi strumenti che possano garantire alla mia azienda un vantaggio competitivo. La mia conoscenza in campo tecnico però non è sufficiente per confermare l'effettiva efficacia di tali strumenti; ed è proprio per questo motivo che ho bisogno dell'approvazione da parte dell'IT Manager di cui mi fido.”*

Criticità La continua angoscia che vive il businessman è quella del fallimento, aziendale in primis, ma anche personale.

Adesione a JOLIE Adottare JOLIE, prima delle aziende concorrenti, consente al Businessman di raggiungere un notevole vantaggio competitivo in termini economici sul risparmio dei costi di sviluppo (e soprattutto mantenimento) degli applicativi.

IT Manager

L'IT Manager è la figura più complessa dal punto di vista relazionale, poiché si interpone, nella scala gerarchica aziendale, fra il Businessman ed il Tecnico Esperto. La sua funzione principale è quella di gestire il proprio *team* (squadra), conoscendone al meglio i singoli componenti e le caratteristiche di ognuno (*skills*). Al di sopra di esso vige il vincolo economico (imposto dal Businessman), in base al quale dovrà poi stabilire le *deadline*¹ del suo *team*. Viene valutato in base alle proprie performance, in termini di tempo e di qualità del risultato ottenuto, i quali dipendono poi, a loro volta, dai risultati della propria squadra.

Vision *“Devo raggiungere gli obiettivi che mi sono stati assegnati, e per farlo devo gestire al meglio il mio team. Saper intuire i punti di forza e debolezza di ogni membro del mio team è una priorità, e devo riuscire a farli lavorare insieme al fine di raggiungere gli obiettivi prestabiliti. Voglio che il Businessman mi valorizzi il più possibile. Il mio sogno più grande è realizzare un grande progetto con un grande team.”*

Criticità La paura più grande dell'IT Manager è quella di non avere più il controllo (psicologico) del *team* e di non raggiungere gli scopi prefissati, così come perdere la posizione guadagnata all'interno dell'azienda.

Questo può essere causato da una insufficiente comprensione delle componenti del proprio *team* e/o da una malagestione di essi. Egli potrebbe quindi non essere

¹La *deadline* consiste nel termine ultimo per l'esecuzione o la consegna di un lavoro ed è una scadenza improrogabile (specialmente nella produzione aziendale).

un buon *leader* (caposquadra), cadendo in errori ancora più gravi, come ad esempio quello di non saper organizzare efficientemente le *deadline* o avere aspettative troppo alte nei confronti del proprio *team* (causate da una scarsa conoscenza dello stesso).

Adesione a JOLIE Per un IT Manager, l'adozione di JOLIE si traduce in una minore dipendenza da componenti esterne (p.e. librerie o *frameworks*), minori costi di formazione (soprattutto in relazione ai cambiamenti delle componenti esterne) ed una più efficace comunicazione fra i membri del *team*.

Tecnico Esperto

Principalmente il ruolo del Tecnico Esperto concerne l'eseguire i compiti a lui assegnati dall'IT Manager, ed ha solitamente poche relazioni con gli altri livelli verticali dell'azienda. Egli è comunque un grande esperto in campo tecnico.

Vision *“Voglio conoscere tutte le tecniche e tecnologie esistenti, e saper fare la scelta giusta al momento giusto. Il mio sogno principale è quello di diventare un tecnico esperto il più possibile riconosciuto; sogno anche di poter realizzare un grande progetto tecnico di portata internazionale.”*

Criticità La debolezza principale del Tecnico Esperto è quella di scoprire che il proprio bagaglio conoscitivo, in un breve periodo di tempo, non serve a nulla e, a causa di questo, la sua autorevolezza tecnica viene messa in discussione. L'adozione di una nuova tecnologia mette paura per molteplici motivi: il principale è l'assenza di conoscenza pregressa su di essa, nonostante i molti anni di esperienza nel campo. A questo si aggiunge però anche la difficoltà di apprendimento, ad esempio, per via dell'età, *forma mentis* o abitudini.

Tutto questo lo rimette in discussione, e quindi torna ad essere nuovamente considerato un *beginner* (nonostante i molti anni di esperienza): questa è la sua vera paura.

Adesione a JOLIE L'adozione di JOLIE, per un Tecnico Esperto, si traduce in minor codice da scrivere (e quindi più velocità di stesura) e soprattutto in una più agevole gestione e mantenimento degli applicativi. Oltre a questi vantaggi, egli è altresì in grado di comunicare con gli altri membri del proprio *team* senza ambiguità o diversità fra linguaggi di programmazione.

2.5 Storia di “aggancio”

Ogni *target* ha bisogno di un primo approccio, il quale inizia la comunicazione di JOLIE. Esso viene affrontato principalmente facendo immedesimare il singolo ruolo in una storia di “aggancio” corrispondente alla sua *vision* e debolezze. L'empatia prodotta da questo approccio fa sì che l'intero flusso comunicativo susseguente venga “accolto” più facilmente: prima dal punto di vista emotivo, e poi elaborato attraverso il ragionamento. La realizzazione di questa fase avviene attraverso la creazione di diverse **storyboard**², alcune testuali, altre grafiche.

Businessman

Il Businessman è principalmente interessato al risvolto economico dell'adozione di questa nuova tecnologia, quindi la sua storia verterà essenzialmente su questo aspetto.

L'analogia che si è cercata di creare consiste nel confronto fra due sistemi (monolite e “a microservizi”), entrambi costruiti per livelli fra loro comunicanti. Di seguito è riportata una *storyboard* grafica.

La storia vissuta dal Businessman inizia con la rappresentazione della situazione attuale, dove ogni livello ha un proprio costo di collegamento con quelli adiacenti (oltre che di mantenimento).

²Lo storyboard (“sceneggiatura disegnata”) è una tecnica inventata da *Walt Disney* nel 1920 per pre-visualizzare un filmato prima della sua realizzazione. Lo storyboard consiste in un documento visuale statico della sceneggiatura dove si precisano le scelte di inquadratura e di cosa si vede e cosa no in ogni dato momento

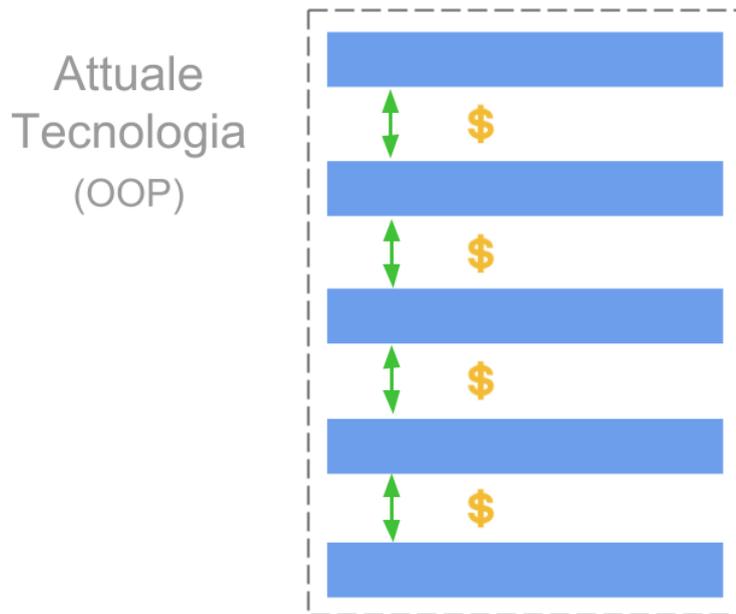


Figura 2.2: Rappresentazione grafica della situazione iniziale (applicazione sviluppata con OOP)

In questa situazione ogni livello comporta un costo (ipotetico) di comunicazione con gli altri. I singoli strati possono essere pensati come componenti di un software monolite: interfaccia utente lato client, database e applicazione lato server. Tutte le componenti fanno però parte di un'unica entità, dove esse sono ben integrate ma difficilmente separabili.

Nel caso in cui, però, il collegamento fra due livelli qualsiasi venisse a mancare (o non funzionasse correttamente), i costi (in termini di perdita di denaro e non solo) aumenterebbero esponenzialmente. La situazione di “crisi” viene rappresentata come segue.

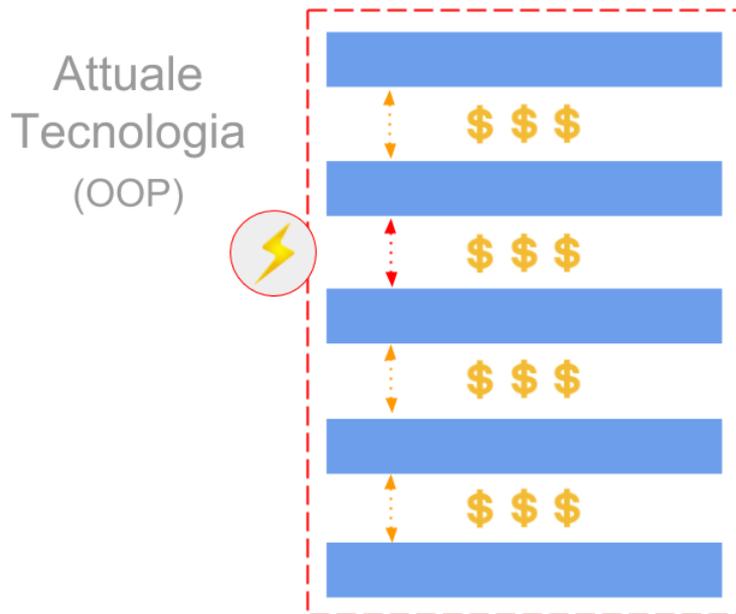


Figura 2.3: Rappresentazione grafica della situazione di “crisi” (malfunzionamento della comunicazione fra livelli)

Più a lungo il sistema è sottoposto a deficit, e più aumentano i costi. La soluzione proposta da JOLIE, invece, punta a ridurre direttamente i costi iniziali, ma anche quelli di mantenimento e riparazione. L'evidente risparmio monetario porta il Businessman a considerare la nuova tecnologia, dal punto di vista economico, come migliore rispetto a quella attualmente adottata. La soluzione proposta diventa quindi graficamente come segue:

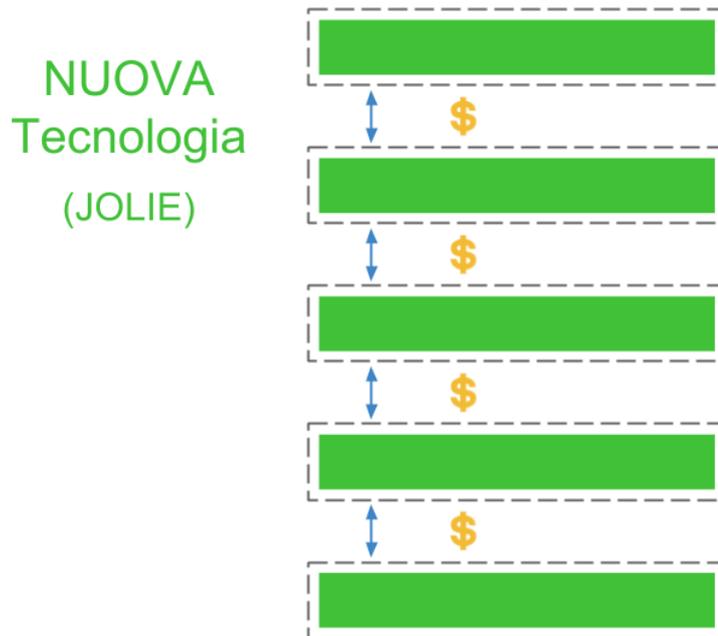


Figura 2.4: Rappresentazione grafica della soluzione JOLIE

Ogni livello della precedente struttura monolitica diventa ora, nel modello distribuito, una componente a sé stante.

Il momento più importante della comunicazione al Businessman si ha quando viene espresso, successivamente, il risultato come effetto di azione-benefici: il primo che riesce ad implementare la nuova tecnologia (in modo scalabile) ottiene enormi benefici in termini di risparmio, diventando così estremamente più competitivo e riuscendo, quindi, a sopravvivere (dal punto di vista aziendale) e a vincere.

La *call to action*³ deve essere, dunque, quella di implementare oggi la nuova tecnologia con lo scopo di prevenire gli elevati (e inevitabili) costi futuri, causati dall'attuale tecnologia.

³La *call to action* è una frase testuale, o inserita su una creatività grafica, che spinge l'utente a compiere una specifica azione. Il principio è quello che non devono esserci dubbi sull'azione da compiere da parte dell'utente.

IT Manager

La storia dell'IT manager ha inizio con egli stesso responsabile di un gruppo di persone (*team*) e produce risultati positivamente valutati dal Businessman. Col passare del tempo vengono aggiunti nuovi componenti alla sua squadra, e ciò rallenta le *deadline*, in quanto le nuove persone inserite devono comunicare più spesso ed essere istruite. Questo porta ad una notevole incrinazione dei rapporti col Businessman. La situazione diventa ancora più insostenibile quando viene aumentata la complessità dei progetti a lui assegnati, causando una maggior dipendenza dalle componenti esterne (e relativi cambiamenti). Il tutto continua a peggiorare, fino ad arrivare ad un punto di rottura e crollo delle performance.

La risoluzione ai problemi posti è l'adozione del linguaggio di programmazione JOLIE:

L'implementazione della nuova tecnologia semplifica notevolmente la gestione del *team*, uniformando il livello tecnico e mantenendo nel tempo la conoscenza acquisita. Inoltre migliora, in termini di efficienza ed efficacia, la comunicazione fra i componenti del *team*, poiché “parlano” tutti lo stesso linguaggio (JOLIE). Questo si traduce in una maggiore velocità implementativa e minor dipendenza da componenti esterne (*addons, framework, ecc*). Il risultato finale è un aumento positivo delle performance con le quali l'IT Manager viene valutato.

Tecnico Esperto

Il coinvolgimento in un qualcosa di più grande entusiasma il Tecnico Esperto. Egli possiede capacità e conoscenze acquisite nel tempo, e quindi conosce ampiamente il proprio mestiere. Quello di cui ha bisogno è sentirsi parte di un progetto più ampio a cui collaborano molte altre persone oltre a lui. Per “convincere” il Tecnico Esperto bisogna mostrare lui brevi *snippet* di codice, così da poter fargli cogliere nell'immediato le potenzialità del nuovo linguaggio.

Capitolo 3

La Strategia

L'obiettivo principale da raggiungere, per creare un'efficace comunicazione, consiste nella definizione di una strategia uniforme da seguire.

Il primo passo da compiere è quello di riuscire ad identificare uno *slogan*, il quale deve essere:

- Breve: poche parole sono più facilmente memorizzabili dalle persone;
- Unico: non deve esistere un altro *slogan* uguale (o troppo simile). Questo non deve necessariamente essere vero a livello globale, ma è sufficiente che lo sia per le singole persone che lo vedono. Per esempio: se esiste un altro *slogan*, il quale però è utilizzato da un'azienda che opera prevalentemente dall'altra parte del mondo, le probabilità che questo sia conosciuto dalle persone a cui ci si rivolge, sono estremamente basse.

L'unicità, proprio grazie alla sua naturale diversità da tutto il resto, è più agevolmente memorizzabile;

- Originale: deve non essere banale o eccessivamente scontato, e non dovrebbe ripetere modelli precedenti o già esistenti.

La banalità viene automaticamente “esclusa” dall'essere umano e, quindi, nemmeno presa in considerazione. La presenza, invece, di due *slogan* simili fra loro, rischia di creare confusione nella mente umana, e quindi impedirne la corretta memorizzazione e/o ricordo;

- **Credibile:** lo *slogan* non deve dar modo ad una persona di pensare che non sia vero o irrealistico.
Se le persone non credono a ciò che viene detto loro, è frequente che queste non lo ricordino. Esagerare può anche essere un ottimo modo per attirare attenzione nel breve periodo, ma nel medio/lungo termine è più facilmente dimenticabile;
- **Evocativo:** deve suscitare emozioni ed esprimere al meglio il concetto, riconducendolo al prodotto/servizio.
Le emozioni provate grazie allo *slogan* fanno sì che questo rimanga più impresso (e quindi anche riutilizzabile e rievocabile) nella mente delle persone;
- **Evidenziare le qualità** che rendono il prodotto/servizio il migliore.

Lo *slogan*, per essere ad “effetto”, deve, inoltre, attirare l’attenzione ed essere al tempo stesso intrigante. Quest’ultima caratteristica garantisce che la persona che lo legge sia stimolata da un po’ di curiosità e si spinga a comprenderne meglio il significato (facilmente raggiungibile).

3.1 JOLIE Unitario e Plurivoco

3.1.1 *Semiotic Square*

Il termine da cui è iniziata l’analisi è stato “plurivoco”. Questo perchè l’essere plurivoco consente al linguaggio di dialogare utilizzando diversi protocolli e formati. In questo modo JOLIE può adattarsi al tipo di comunicazione da usare in base all’applicativo con il quale interloquisce.

Si è cercato poi di trovare un ulteriore termine da affiancare a quello già trovato, così da rinforzare ed esprimere meglio i punti di forza di JOLIE, sempre attraverso uno *slogan*, e si è utilizzata una tecnica nota per l’individuazione dei termini appropriati: il ***Semiotic Square***.

Il *Semiotic Square* (quadrato semiotico) è stato introdotto dal linguista lituano e studioso di semiotica Algirdas Julien Greimas, e deriva dal quadrato delle opposizioni

di Aristotele. Il quadrato semiotico viene costruito partendo da due concetti opposti fra loro (S_1 e S_2), assumendo l'esistenza di altri due concetti: $\sim S_1$ e $\sim S_2$. Le relazioni fra gli elementi sono espresse come segue:

- S_1 e S_2 sono contrari;
- S_1 e $\sim S_1$, S_2 e $\sim S_2$ sono contraddittori;
- $\sim S_1$ e $\sim S_2$ sono subcontrari (possono avere in comune delle zone intermedie);
- S_1 e $\sim S_2$, S_2 e $\sim S_1$ sono complementari (sono legati da una relazione di implicazione);
- Il punto di incrocio fra le due diagonali corrisponde ad una contraddizione.

Di seguito la rappresentazione analitica del *Semiotic Square*.

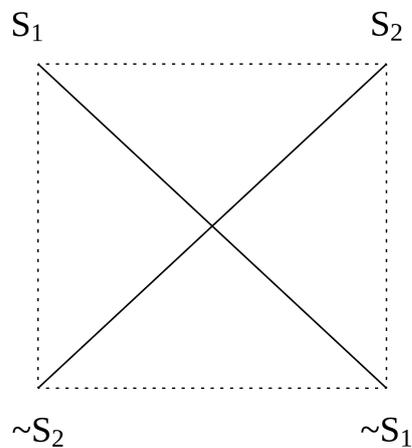


Figura 3.1: Rappresentazione analitica del *Semiotic Square*

Di seguito, invece, è proposto un esempio applicativo.

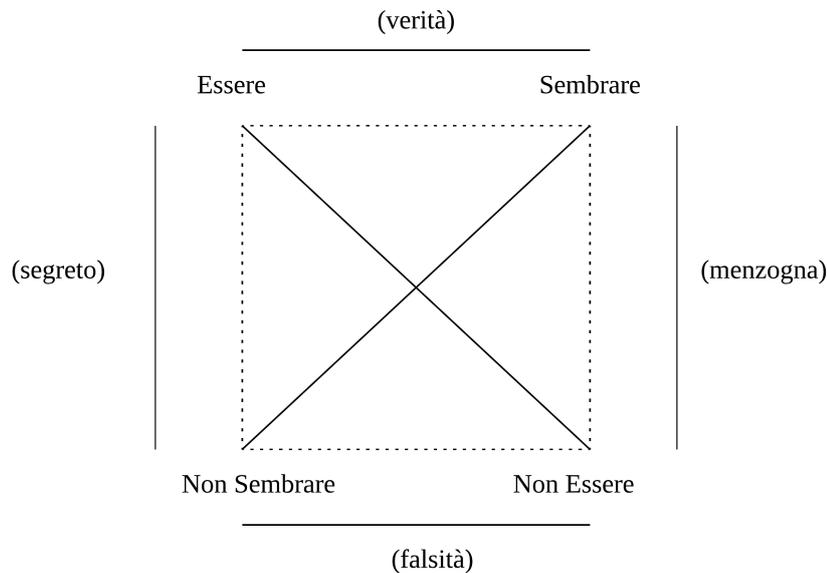


Figura 3.2: Esempio di utilizzo del *Semiotic Square*

Nell'esempio proposto la partenza consiste nei due termini "essere" e "sembrare" (fra loro opposti). Si procede poi a definire (al vertice delle diagonali opposti) i rispettivi subalterni: per "essere" è "non essere", mentre per "sembrare" è "non sembrare". Dal quadrato ottenuto, si tracciano successivamente i quattro lati del perimetro, ognuno dei quali corrisponde (se esiste) ad un termine che racchiude in sé stesso i significati di entrambi i concetti posti ai vertici di tale lato. Si analizzi ora l'esempio.

Partendo dal lato superiore, il concetto che esprime, allo stesso tempo, "essere" e "sembrare" è "verità"; "menzogna", invece, rispecchia perfettamente il "sembrare" e "non essere". Per quanto riguarda il "non sembrare" e il "non essere", il concetto si ritrova chiaramente in "falsità"; mentre l'"essere" e il "non sembrare" si riconduce a "segreto".

Di seguito, invece, si riporta l'applicazione pratica per quanto riguarda lo *slogan* creato, dove, in particolare, si analizza il termine "plurivoco".

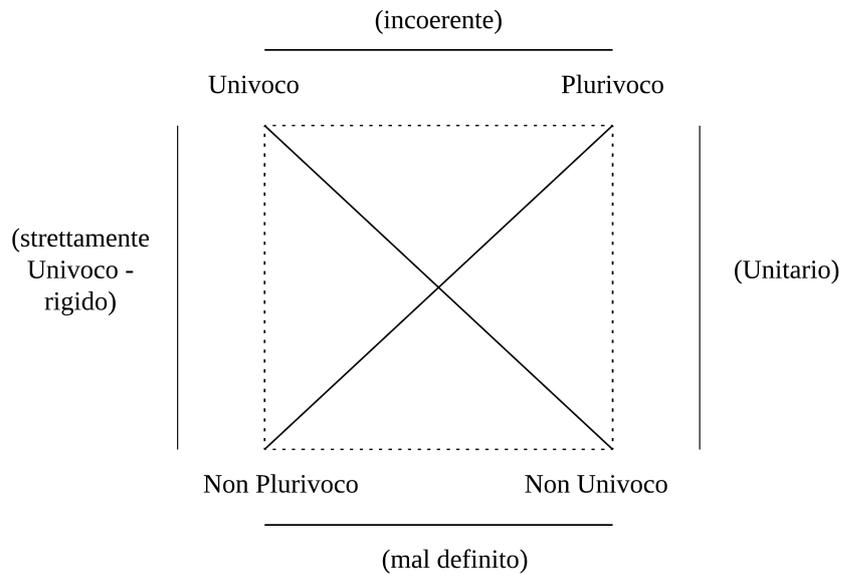


Figura 3.3: Utilizzo del *Semiotic Square* per “unitario” e “plurivoco”

Il termine di partenza è “plurivoco”, dal quale si deduce poi il suo contrario, ovvero “univoco”. I rispettivi subalterni sono “non plurivoco” e “non univoco”. Analizzando il quadrato in senso antiorario, dal lato superiore non è possibile esplicitare un termine che descriva al tempo stesso “plurivoco” e “univoco”, infatti si tratterebbe di un’incoerenza. Il concetto che si troverebbe poi sul lato fra “univoco” e “non plurivoco” risulterebbe eccessivamente stringente (rigido) rispetto a ciò che si vuole esprimere. La definizione di “non plurivoco” e “non univoco” delineerebbe un concetto mal definito (vago). Il termine, invece, che esprime al meglio “plurivoco” e “non univoco” è “unitario”.

L’utilizzo del *Semiotic Square* ha permesso quindi l’individuazione di un’ulteriore parola (“unitario”) che esprima ancora meglio la natura di JOLIE, rispetto al solo termine “plurivoco”.

3.1.2 Significato letterario

Lo *slogan* deve essere appropriato e corretto, oltre che efficace. Per questi motivi si è analizzato, in primo luogo, la semantica di entrambi i termini.

Unitario Il termine “unitario” è ampiamente definito, rispetto a “plurivoco”. Esso ritrova infatti diverse definizioni:

- Uniforme, armonico, organico: stile unitario¹;
- Che mira all’unità, all’unificazione (intese anche come valore); che forma una struttura organica; che ha trovato tutti d’accordo: si sta cercando una soluzione al caso²;
- Che si fonda sull’unità, si ispira o tende all’unità, alla solidità di tutte le componenti³;
- Soluzione unitaria, unica per una serie di problemi o per i diversi aspetti di un problema³;
- Con riferimento a opere letterarie o artistiche, che ha unità, coerenza e organicità tra le varie parti⁴;
- In tecnologia, operazione di natura, rispettivamente fisica o chimica che, facendo parte di un processo, conserva un’identità caratteristica indipendentemente dalla particolare destinazione del processo stesso⁴.

In inglese, il termine “unitario” può essere tradotto in “*unitary*”, “*uniform*” o “*unitive*”. Delle tre traduzioni, si è ritenuta più appropriata ed evocativa “*unitary*”.

¹[<https://www.garzantilinguistica.it/ricerca/?q=unitario>]

²[http://dizionari.corriere.it/dizionario_italiano/U/unitario.shtml]

³[<http://dizionari.repubblica.it/Italiano/U/unitario.php>]

⁴[<https://dizionario.internazionale.it/parola/unitario>]

Plurivoco Per quanto riguarda “plurivoco”, invece, le traduzioni si riducono sensibilmente:

- In matematica, di funzione a più valori (in contrapposizione a univoco)⁵;
- In linguistica, termine talora usato come sinonimo di polisemico⁵.

Per quanto riguarda l’inglese, “plurivoco” può essere tradotto in “*plurivocal*” o “*many-side*”. Il termine scelto, in questo caso per via delle semplicità rispetto all’altro, è stato “*plurivocal*”.

3.1.3 Significato per JOLIE

I termini appena visti non sono stati scelti solo per il loro significato, ma per come possono rappresentare (in modo espressivo) JOLIE. La “chiave di lettura” cambia, quindi, come segue.

Unitario La proprietà di essere unitario consente a JOLIE di poter affrontare tutte le problematiche di integrazione digitale con un’unica modalità, senza cambi tecnologici né di approccio o progettazione.

Plurivoco La proprietà di essere plurivoco consente al linguaggio di dialogare utilizzando diversi protocolli e formati. In questo modo JOLIE può adattarsi al tipo di comunicazione da usare in base all’applicativo con il quale interloquisce.

Unitario e Plurivoco Essere al tempo stesso unitario e plurivoco rende JOLIE, a suo modo, unico e speciale. Esso permette, infatti, sia di progettare un sistema integrato in modo uniforme, sia di poter dialogare con le componenti già esistenti e con quelle nuove che arriveranno in futuro.

Risultato Il risultato finale è stato lo *slogan* in inglese:

“The Unitary and Plurivocal Programming Language”.

⁵[<http://www.treccani.it/vocabolario/plurivoco>]

3.2 Roadmap

Introduzione

Nel capitolo I si è definita la situazione futura, suddivisa per livelli. Il cuore della strategia, però, non è solamente la suddivisione dei due approcci in più livelli, ma è anche il vero e proprio contenuto di tali livelli. Di seguito ne è mostrata una rappresentazione grafica.

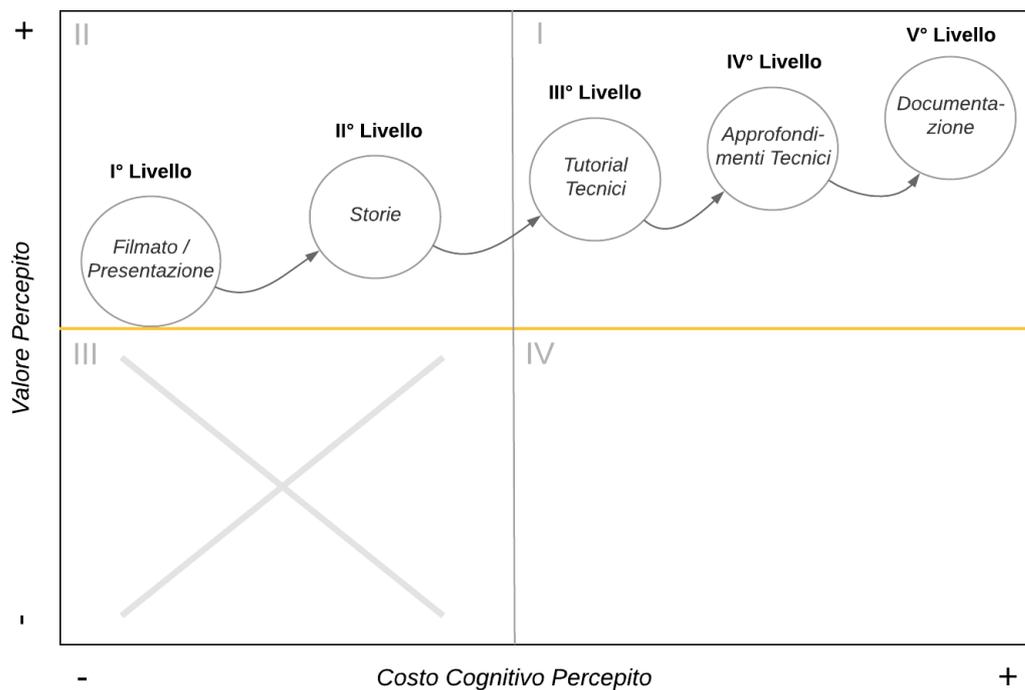


Figura 3.4: Rappresentazione grafica della strategia completa (livelli e contenuti)

L'aspetto più importante della strategia è l'incrementale valore e costo cognitivo percepiti. Il I° Livello, infatti, sarà molto più accessibile (e quindi modulato in tale senso) rispetto al IV° Livello, dove sarà necessario un maggiore impegno e sforzo mentale da parte di una persona. Ogni livello è costruito appositamente per garantire il giusto carico cognitivo, attraverso l'utilizzo (o meno) di testi, immagini e video.

La descrizione dettagliata delle modalità realizzative di tutti i livelli è contenuta all'interno della *roadmap*.

Definizione

La *roadmap* ha lo scopo di organizzare le varie fasi della comunicazione vissuta dall'utente in un primo approccio al sito di JOLIE. Essa avviene attraverso 5 livelli principali, dove ognuno dei quali ha un ben definito scopo e forma rappresentativa attraverso la quale viene comunicato. Il percorso ideale è identificato contestualmente al *target* di riferimento:

- **Businessman:** dato il suo ruolo, molto probabilmente è solo interessato ad esplorare i livelli I e II;
- **IT Manager:** necessita di conoscenze decisamente più tecniche e approfondite rispetto al Businessman. Si relazionerà, quindi, maggiormente con i livelli III, IV e V;
- **Tecnico Esperto:** poiché non possiede conoscenze pregresse, potrebbe realmente attraversare l'intera catena a livelli, avendo bisogno di affrontare sia il primo approccio conoscitivo, sia quello più tecnico.

Si viene così a formare una catena di apprendimento crescente, a partire dal I° Livello dove l'utente, appena interessato, si avvicina, fino ad arrivare al V° ed ultimo livello.

Lo scopo dei primi livelli (I - II) è quello di massimizzare il valore percepito della nuova tecnologia, minimizzando allo stesso tempo il costo cognitivo richiesto. Ogni livello successivo al I° viene "costruito" sui precedenti, ed esso potrà introdurre un nuovo costo cognitivo (sempre maggiore rispetto al precedente). Questo è possibile (e soprattutto sostenibile) perché l'incremento del valore percepito di ogni nuovo livello parte da quello del precedente, per essere poi aumentato di livello in livello.

3.2.1 I° Livello

Il primo livello in cui l'utente si avvicina è quello più importante, e proprio per questo è necessario ottenere il massimo valore percepito, con il minimo costo cognitivo possibile.

Lo scopo principale è quello di far percepire subito l'innovazione tecnologica che sta per avvenire, e come trarre un evidente vantaggio economico con essa.

Lo scopo ultimo è quello di far passare l'utente alla "scoperta" del livello successivo, stimolando in lui curiosità ed interesse.

Il *target* di riferimento comprende prevalentemente *business manager* (IT e non). La forma rappresentativa consiste per lo più in un **breve video** (2 minuti circa), diverse immagini e poco testo, così da garantire un basso costo cognitivo.

Analogia

Nel capitolo I si è analizzato in dettaglio il funzionamento di *System 1*, dove è emersa la necessità di creare un'analogia per facilitare la comprensione della nuova tecnologia. Di seguito è riportata una bozza appositamente ideata, la quale può essere utilizzata nel filmato di presentazione.

Technological Level	Analogy	Mainstream Technologies	JOLIE
Cloud Information System	 <p>Society</p>	 SOA tech.	
Informative system	 <p>Human Body</p>	 Java / JS / Python + Libraries + Frameworks + SOA tech.	
Microservices	 <p>Organs</p>	 Java / JS / Python + Libraries + Frameworks	
Computing	 <p>Molecules</p>	 Java / JS / Python	

Figura 3.5: Rappresentazione grafica dell'analogia OOP - JOLIE *

* *Le spunte verdi indicano che per realizzare il rispettivo livello tecnologico è sufficiente utilizzare solo JOLIE, senza altre componenti esterne.*

Il punto esclamativo rosso, invece, indica la necessaria dipendenza da diverse componenti esterne (specificate).

L'analogia parte dal livello più inferiore (Molecole), il quale rappresenta (per JOLIE) la sola computazione, realizzabile in modo equivalente con OOP anche tramite Java, Javascript oppure Python. Proseguendo verticalmente, si trovano poi gli Organi, derivanti dall'insieme (e interazione) di Molecole, dove, per creare artefatti analoghi con OOP, è necessario utilizzare più strumenti: *libraries* e *frameworks*. Successivamen-

te, gli organi possono comporre il Corpo Umano; per creare un sistema informativo tramite OOP, è necessario avvalersi dell'utilizzo di tutte le tecnologie viste fino ad ora, con l'aggiunta di SOA⁶. L'ultimo livello dell'analogia è la Società, composta da singoli esseri umani che interagiscono fra loro; questo è il livello informaticamente più il complesso e difficile da creare (e soprattutto gestire). La costante è il punto di forza di JOLIE: esso può essere utilizzato singolarmente per sviluppare qualunque livello dell'analogia, senza alcuna dipendenza da altri linguaggi di programmazione o da ulteriori tecnologie.

Dal punto di vista cognitivo, per una persona è estremamente semplice concepire il livello più alto dell'analogia (Società) come composizione di tutti i livelli inferiori (dal Corpo Umano fino alle Molecole), ed è proprio ciò che accade utilizzando JOLIE. L'analogia è stata volutamente semplificata, così da ridurre il costo cognitivo necessario per la sua comprensione, ma senza alterarne la correttezza. Il fatto di utilizzare "noi stessi" (inteso come persone) e ciò di cui siamo composti (Organi, Molecole e Proteine) rende l'analogia estremamente delicata. Il rischio è infatti quello di causare nelle persone emozioni eccessivamente vive, dato che si sta parlando di ciò di cui siamo composti, vanificando così l'efficacia dell'intera successiva comunicazione. Per evitare ciò, si è scelto di utilizzare (seppur a scopo esclusivamente dimostrativo) immagini estremamente semplici e gradevoli alla vista, senza presentarne alcune che potrebbero sembrare, seppur biologicamente corrette, visivamente (ed emozionalmente) "disgustose".

Analizzando, invece, l'analogia dal punto di vista tecnologico, si ha che la creazione (verticale) di tutti i livelli illustrati può essere implementata attraverso l'utilizzo di JOLIE, senza ricorrere a "frameworks" o librerie esterne; indispensabili, invece, nel caso di sistemi sviluppati tramite OOP.

Questa analogia dovrà essere espressa nel video di presentazione, in modo da poter

⁶SOA è l'acronimo di *Service-Oriented Architecture* (Architettura Orientata ai Servizi). L'*Organization for the Advancement of Structured Information Standards* definisce SOA come "un paradigma per l'organizzazione e l'utilizzo delle risorse distribuite che possono essere sotto il controllo di domini di proprietà differenti. Fornisce un mezzo uniforme per offrire, scoprire, interagire ed usare le capacità di produrre gli effetti voluti consistentemente con presupposti e aspettative misurabili".

far comprendere il più semplicemente possibile il valore di JOLIE, a prescindere dal profilo della persona che lo visualizzerà.

3.2.2 II° Livello

Una volta descritti i principali aspetti del cambiamento tecnologico (e le opportunità che esso porta con sé, se sfruttato), si passa ad illustrare, quasi raccontando, alcune **storie** vere di casi d'utilizzo di JOLIE. Lo scopo è quello di rendere, attraverso le testimonianze, il più interessanti possibili i vantaggi (economici e non solo) dell'adozione della nuova tecnologia.

In questa fase vengono meno i video, e l'utente si dedica per lo più alle immagini e al testo.

Le storie descritte in questo livello differiscono dalle storie di “aggancio” (definite nel capitolo II) per il contenuto: le storie di questo livello consistono in vere e proprie testimonianze, mentre quelle di “aggancio” sono create principalmente attorno al profilo che le leggerà.

Per la realizzazione di una qualsiasi storia, sia essa appositamente pensata oppure una vera testimonianza, viene seguita una comune tecnica denominata “Il Viaggio dell'Eroe”.

Il Viaggio dell'Eroe

Il Viaggio dell'Eroe consiste in un modello narrativo per creare, raccontare ed analizzare storie. Esso è stato dedotto dallo sceneggiatore statunitense Christopher Vogler[8], grazie anche agli studi del linguista russo Vladimir Jakovlevič Propp (“Schema di Propp”)[9]. Le applicazioni di questo modello sono ritrovabili nella maggior parte dei film e cartoni animati prodotti da *Walt Disney*.

Tutte le storie, sostiene Vogler, sono riconducibili ad uno schema narrativo elementare: l'*Eroe* riceve una *Chiamata* che lo strappa al suo *Mondo Ordinario*, istruito da un *Mentore* vince la sua *Paura*, supera la *Prima Soglia* ed entra nel *Mondo Straordinario*, poi accede alla *Caverna più Profonda*, affronta la *Prova Centrale*, ottiene

la *Ricompensa* e, dopo aver attraversato una *Resurrezione*, torna a casa con l'*Elisir*. Di seguito una rappresentazione grafica del Viaggio dell'Eroe:

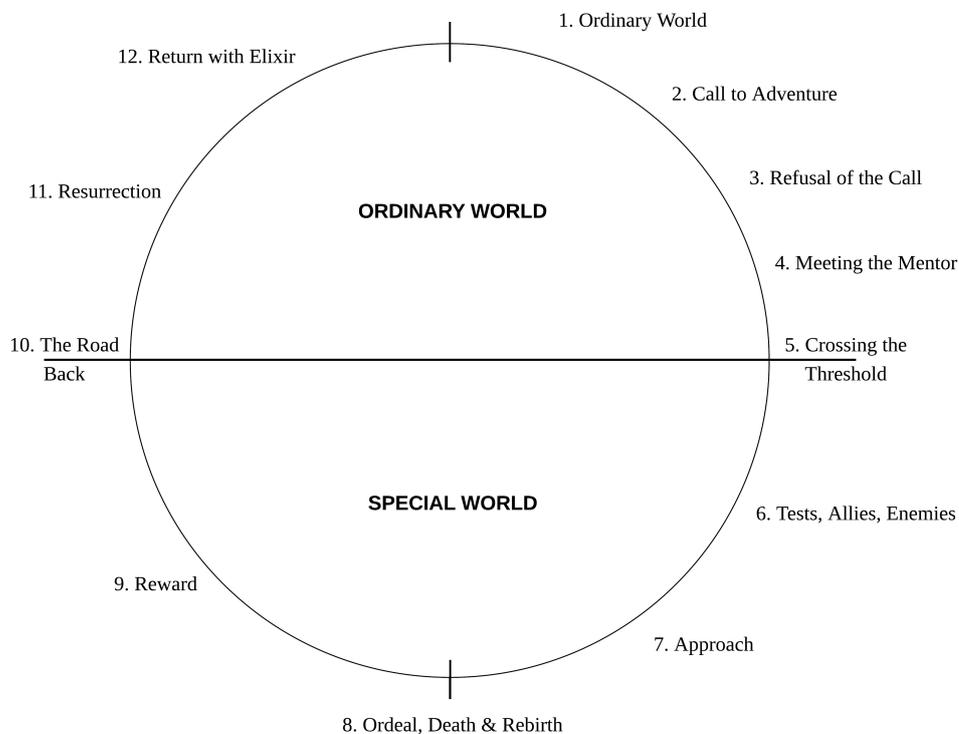


Figura 3.6: Rappresentazione grafica del Viaggio dell'Eroe

1. **Ordinary World** (Mondo Ordinario). L'eroe lascia un mondo a lui conosciuto per cominciare un viaggio, ed entra in un altro mondo (*Special World*). In un certo senso è l'ultimo posto da cui si proviene, dato che in un racconto e nella vita si passa attraverso la successione di mondi speciali che lentamente ridiventano ordinari appena ci si abitua. D'altra parte non si può percepire qualcosa come novità, stranezza o straordinarietà senza fare comparazione con ciò che non lo è. Il Mondo Ordinario è il mondo consueto dell'eroe, quello da cui parte. Una ferita, anche se nascosta, rende un eroe più umano e segna il confine della zona in cui è più vulnerabile. Ecco perché alcuni tratti possono anticipare nel mondo ordinario il disegno o l'incombenza del mondo speciale.

In particolare, per rinforzare il racconto, in questa fase si evidenzia qual è la posta in gioco, cosa si lascia e dove si spera di tornare.

2. **Call to Adventure** (Chiamata all'Avventura). Essa consiste nella sfida per l'eroe, dove si stabilisce l'obiettivo e il percorso da percorrere. La chiamata all'avventura risalta il rischio o il prezzo da pagare e rende chiaro l'obiettivo dell'eroe. Un evento iniziale scatenante (casuale oppure determinato) è necessario per far iniziare la storia. Esso può essere una chiamata interiore, l'appello da parte di un messaggero oppure una serie di coincidenze ricorrenti. A volte può essere una mancanza, l'improvviso "rapimento" di qualche bisogno dell'eroe o l'incidente procurato a una figura a lui cara, in ogni caso una sottrazione che lascia senza scelta.
3. **Refusal of the Call** (Rifiuto della chiamata). Ci sono casi di eroi riluttanti (proprio perché in questo modo aumentano la percezione del rischio che correranno) che cercano di "negare" la chiamata e sfuggire ad essa. Il timore mette in evidenza il fatto di non poter essere a conoscenza del futuro. A volte serve un'ulteriore influenza, nonostante il richiamo all'avventura, come un cambiamento delle circostanze o l'incoraggiamento di un mentore.
4. **Meeting the Mentor** (Incontro col mentore). Quello di cui ha bisogno l'eroe per mettersi in viaggio: consigli, direzione, guida, uno strumento "magico" da portarsi (una conoscenza, la fiducia in se stesso, la protezione, l'allenamento). I mentori sono in genere coloro che hanno già affrontato il viaggio in precedenza.
5. **Crossing the Threshold** (Superamento della prima soglia). L'eroe accetta la sfida. Da questo momento si entra nel mondo speciale del racconto. Egli si lancia nel viaggio, dal momento in cui non può più tornare indietro. È una soglia su cui ci sono guardiani, esseri che cercano di fermare l'eroe e che vanno ignorati, assorbiti o trasformati in alleati. L'attraversamento del confine tra mondo ordinario e mondo speciale può anche essere un incidente, l'affronto diretto del mondo speciale, laddove lo slancio di fiducia potrebbe trasformarsi in "crisi", e tutte le illusioni romantiche sul mondo nuovo potrebbero essere riviste al vero contatto con esso.

6. **Tests, Allies, Enemies** (Prove, nemici, alleati). In questa zona l'eroe fa i primi incontri, viene coinvolto nelle sfide che servono a imparare le regole del mondo speciale. Qui si rivela il vero carattere dell'eroe, si mettono in evidenza i sentimenti, i ritmi, le priorità, i valori e le regole che contano.
7. **Approach** (Avvicinamento alla caverna più recondita). L'eroe si avvicina all'apice, al posto pericoloso. In questa fase, spesso egli si ferma sul cancello per preparare, pianificare e sorpassare i "guardiani dell'ombra". L'eroe sta formando una nuova percezione di sé e degli altri, ha tempo per indossare una divisa o le "armi" prima dell'incontro. È anche la fase dove i compagni di viaggio diminuiscono e la lotta si fa più dura, con la sorpresa di veder emergere nuove qualità nei personaggi. Ci si accerta che ognuno sia d'accordo sugli obiettivi.
8. **Ordeal, Death & Rebirth** (Prova centrale). La Prova centrale è il momento critico della battaglia. La prova in cui l'eroe rischia davvero di morire o muore per poter rinascere di nuovo. Nel momento cruciale in cui egli o i suoi obiettivi sono a rischio, in genere c'è un temporaneo rovescio di fortuna. In questa fase può avvenire anche l'incontro con la divinità o con il "tentatore". Durante la prova centrale l'eroe si trova a dover fronteggiare le sue più recondite paure, con il fallimento dell'impresa, o con la fine di un rapporto. Dopo niente sarà mai più lo stesso. Tutto avviene sulla cima della montagna, nella zona più profonda della caverna, nel cuore della foresta, nella parte più riposta d'una terra straniera, o nel posto più segreto della sua anima. È il momento più profondo della discesa prima della risalita (qui spesso un testimone vede l'eroe che sembra morto), dove l'ombra si rivela (sia essa un nemico o un rivale, o la paura stessa dell'eroe, le sue qualità rifiutate e proiettate in altri).
9. **Reward** (Ricompensa). L'eroe, sopravvissuto, "festeggia" (anche per l'aver imparato qualcosa). Se c'è un tesoro da prendere, questo è il momento. Finita la burrasca, ci si misura, si diventa consapevoli della propria volontà o forza, raggiungendo il rispetto di sé. La ricompensa, comunque, non sempre viene data. A volte l'eroe si rende conto anche di quanto sia stato stupido o testardo,

di quanto abbia rischiato più del necessario. In questa fase una rivelazione oppure un riconoscimento sono sempre presenti.

10. **The Road Back** (Via del ritorno). L'eroe è trasformato e sulla via del ritorno, ma non è ancora fuori dalla "foresta". La storia deve raccontare la sua decisione di tornare al mondo ordinario (la scelta di non rimanere nel mondo speciale, con la paura che il premio possa sparire in giorni ritornati comuni).
11. **Resurrection** (Resurrezione). Non è la prova più grande, ma la definitiva. È come un esame finale per provare che si è imparata la lezione, dato che una cosa utile nel mondo straordinario non è detto che sia trasportabile in quello ordinario. È la purificazione, l'adattamento al ritorno che a volte prende la forma di un'ulteriore sfida (magari la minaccia questa volta non riguarda l'eroe, ma l'intero suo mondo). Nonostante la vittoria, è ancora il momento difficile in cui l'eroe può morire. A volte in questa fase si risvegliano i falsi pretendenti, quelli che vogliono approfittare delle conquiste dell'eroe per farle proprie e rubargli il premio.
12. **Return with Elixir** (Ritorno con l'elisir). L'eroe è tornato rinato, definitivamente cambiato, e ha portato con sé l'esperienza raggiunta, un dono da usare nel mondo ordinario. Ci sono due possibilità: il ritorno al punto di partenza, ma ora il vecchio mondo sembra diverso, e la forma aperta, il cui finale continua all'interno di colui al quale la storia è stata raccontata. La funzione di questa fase è quella di concludere la storia.

Le tappe del viaggio appena illustrate sono paragonabili alle fasi che un'azienda attraversa durante l'adozione di JOLIE.

Intervista

Per realizzare il contenuto di questo livello ci si è basati principalmente su storie vere di aziende che hanno adottato JOLIE. Si è proposto loro una serie di domande a cui rispondere, in modo così da poter realizzare una breve **intervista**. Di seguito

le otto domande proposte:

1. Potete descrivere brevemente di cosa si occupa l'azienda e in quale settore opera?
2. Qual è lo scenario in cui è stata applicata la tecnologia?
3. Quali erano le difficoltà riscontrate prima dell'uso della tecnologia?
4. In che modo Jolie ha permesso di risolvere la problematica descritta in risposta alla domanda precedente?
5. Come si presenta ora lo scenario, dopo aver implementato la soluzione con Jolie? La soluzione finale vi ha soddisfatto? In che modo?
6. Rispetto alla problematica descritta in risposta al punto 2, potreste elencare almeno (ma non solo) 2 pregi e 2 difetti trovati nell'adozione di Jolie?
7. Jolie è stato adottato anche in altri ambiti all'interno dell'azienda?
8. A posteriori, consigliereste Jolie ad altre aziende come soluzione alternativa a problemi simili rispetto a quelli riscontrati?

Una delle aziende che ha gentilmente risposto all'intervista, e acconsentito alla sua pubblicazione, è stata *"Monrif Group S.p.A."*. Quest'ultima opera nel settore dei quotidiani, della pubblicità, dei nuovi media, dell'ospitalità e della stampa. Di seguito l'intervista completa.

1. Potete descrivere brevemente di cosa si occupa l'azienda e in quale settore opera?
"Monrif è una azienda che opera principalmente nel mondo dell'editoria, attraverso la sua controllata Poligrafici Editoriale."

2. Qual è lo scenario in cui è stata applicata la tecnologia?

“Lo scenario informatico presente nelle aziende del Gruppo Monrif è estremamente eterogeneo, sia dal punto di vista delle richieste operative che della composizione tecnologica delle soluzioni adottate. JOLIE si è inserito come tecnologia di middleware nel framework della system integration.”

3. Quali erano le difficoltà riscontrate prima dell’uso della tecnologia?

“Le difficoltà incontrate prima dell’introduzione erano quelle tipiche di un sistema eterogeneo con un alto livello di cross-component coupling, il quale rendeva lo sviluppo di qualsiasi nuova soluzione di integrazione difficile da sviluppare, gestire e monitorare.”

4. In che modo Jolie ha permesso di risolvere la problematica descritta in risposta alla domanda precedente?

“JOLIE ha portato come dote il nuovo approccio architetturale a microservizi. I microservizi, per definizione, sono altamente componibili, scalabili e modulabili, rendendo le nuove implementazioni molto più flessibili e riducendo gradualmente il cross-component coupling. Il valore aggiunto di JOLIE è dato dalla sua completa coerenza linguistica con l’architettura a microservizi. Lo sviluppatore JOLIE può concentrarsi sugli aspetti di orchestrazione e composizione tipici di un architettura a microservizi, senza perdere tempo con aspetti di migrazione tra paradigmi.”

5. Come si presenta ora lo scenario, dopo aver implementato la soluzione con JOLIE? La soluzione finale vi ha soddisfatto? In che modo?

“Negli ultimi tre anni, l’introduzione di JOLIE nell’azienda ha permesso lo sviluppo di soluzioni di systems integration utilizzando solo risorse interne, con un considerevole impatto sia sul costo delle soluzioni che nella gestione di progetto. Le soluzioni sviluppate in JOLIE ci soddisfano sia dal punto di vista tecnico che operativo, dimostrando di essere stabili, efficienti e facilmente modificabili; queste caratteristiche hanno avuto un forte impatto sull’operatività del reparto tecnico, sia in fase di progettazione che di gestione delle soluzioni.”

6. Rispetto alla problematica descritta in risposta al punto 2, potreste elencare almeno (ma non solo) 2 pregi e 2 difetti trovati nell'adozione di JOLIE?

“I pregi riscontrati superano di gran lunga i difetti. In particolare, sono da evidenziare la facilità di apprendimento della tecnologia e la facilità con cui si possono sviluppare, deployare e combinare microservizi sviluppati in JOLIE. I due difetti principali riscontrati sono: l'assenza di un vero e proprio IDE, la presenza di un editor (ATOM) con supporto per JOLIE, ha mitigato parzialmente questo problema, e l'incoerenza della documentazione online, che, a volte, risulta di difficile comprensione e navigazione.”

7. JOLIE è stato adottato anche in altri ambiti all'interno dell'azienda?

“JOLIE rimane principalmente legato ad aspetti di systems integration, ma è stato utilizzato anche come tecnologia per lo sviluppo di backend per applicazioni WEB e API pubbliche.”

8. A posteriori, consigliereste Jolie ad altre aziende come soluzione alternativa a problemi simili rispetto a quelli riscontrati?

“Sì, JOLIE può rappresentare la chiave di volta per le aziende che vogliono sviluppare una maggiore integrazione tra i componenti del loro sistema informatico. L'impatto positivo che JOLIE ha avuto in Monrif può essere facilmente proiettato su scenari e aziende simili a Monrif.”

3.2.3 III° Livello

Il III° livello contiene i “Tutorial Tecnici”. Questi ultimi vertono sui principali argomenti tecnici emersi dalle Storie del II° livello. I contenuti dei tutorial sono affrontati dal punto di vista tecnico-teorico, ma senza entrare in dettagli implementativi. All'interno di essi sono presenti collegamenti diretti ad “Approfondimenti Tecnici” (descritti nel IV° Livello).

Lo scopo di questo livello è quello di fornire una guida tecnica per raggiungere un singolo obiettivo. Di seguito l'elenco dei Tutorial Tecnici proposti:

- Orchestrator

- API Gateway
- Web Application
- Load Balancer
- Containerize a JOLIE microservice

3.2.4 IV° Livello

In questo livello, invece, vengono proposti una serie di “Tutorial di accompagnamento” per l’utente, attraverso l’esplorazione (a difficoltà incrementale) di tutti i fondamentali per programmare in JOLIE.

Il percorso è proposto tramite interfaccia *web*, dove l’utente può scrivere codice JOLIE ed eseguirlo direttamente dall’interfaccia stessa. La struttura della pagina si suddivide in due parti: a destra le due schermate “Server” e “Client”, dove poter scrivere codice, al di sotto delle quali ci sono i rispettivi pulsanti per far eseguire i microservizi. L’output è osservabile tramite due distinte *console*. A sinistra, invece, vi sono le tracce tutorial da seguire. Esse sono visualizzate in modo da poterle scorrere verticalmente. Il contenuto è organizzato in modo da fornire una traccia da seguire per svolgere un piccolo esercizio, al fine di apprendere le competenze riguardanti un singolo micro-argomento. A completare questa parte vi sono anche i rispettivi “spoiler” delle soluzioni, sempre visualizzabili da parte dell’utente (su richiesta tramite il click di un pulsante).

Ogni argomento citato nei tutorial è propriamente referenziato alla pagina di descrizione completa della Documentazione (Livello V), ivi incluse le singole parole usate all’interno della descrizione e spiegazione.

L’elenco dei tutorial è organizzato come segue:

1. BASIC: semplici esempi illustrativi delle capacità di base di un programma JOLIE
 - Server - Client: semplice esempio di classica comunicazione Client - Server;

- Server always ON: come far diventare il Server scritto al punto precedente un generico servizio che accetta più comunicazioni;
 - Sequence operations (“;”): spiegazione di cosa significhi eseguire in sequenza due istruzioni JOLIE;
 - Parallel operations (“|”): a completamento del punto precedente, le istruzioni JOLIE possono essere eseguite anche in parallelo;
 - From SODEP to HTTP: esempio di come un programma (unitario) JOLIE riesca ad essere espresso in vari formati (plurivoco). In particolare, si passa dal protocollo binario SODEP allo standard HTTP;
 - Calls from Browser: dato che al punto precedente si è creato un server always on per HTTP, ora è possibile richiamarlo tramite un comune browser;
2. INTERMEDIATE (Step 1): spiegazione (corredata da esempi) di come utilizzare gli elementi base di JOLIE per definire la comunicazione tra microservizi;
- Interfaces
 - Types
 - Ports
3. INTERMEDIATE (Step 2): spiegazione (corredata da esempi) di come programmare logiche complesse con gli operatori JOLIE;
- Choice
 - Procedures
 - Error handling
4. ADVANCED: spiegazione con esempi di come un programma JOLIE possa ospitare l'esecuzione di sotto-servizi, possibilmente scritti in linguaggi diversi;
- Embedding JOLIE
 - Embedding Java

- Embedding Javascript
5. EXPERT: spiegazione con esempi dei costrutti di JOLIE per la creazione di architetture complesse;
 - Aggregation
 - Redirections
 6. MASTER: spiegazione corredata da esempi di come creare una sessione che permetta a due o più programmi JOLIE di coordinarsi tra loro;
 - Sessions

3.2.5 V° Livello

Il V° ed ultimo livello fornisce tutti i dettagli di cui un programmatore JOLIE ha bisogno: la Documentazione. Essa contiene un elenco di macro-argomenti analizzati poi nello specifico, che aiutano sia il Tecnico Esperto che il Beginner. Tra i vari argomenti proposti si citano: Basic data types, Data structures, Protocols, Locations e Databases.

Capitolo 4

Principi per l'Adozione di un Linguaggio di Programmazione

In questo capitolo si analizzano le diverse tematiche riguardanti l'adozione di un nuovo linguaggio di programmazione da parte (prevalentemente) dei programmatori. Lo scopo consiste nell'individuare alcuni principi da seguire per facilitare ulteriormente l'adozione di JOLIE e il suo mantenimento nel tempo.

Lo studio

Leo A. Meyerovich e Ariel Rabkin hanno formulato, nel 2012, una serie di studi ed ipotesi[6] su quali siano i principi alla base dell'adozione di un linguaggio di programmazione. Di seguito se ne citano e analizzano alcuni, riconducendo l'utilità di questi a JOLIE e le eventuali correlazioni con la Strategia descritta nei capitoli precedenti.

4.1 Il cambiamento

L'adozione di un linguaggio di programmazione nuovo a programmatori ed IT Manager è rappresentabile attraverso una funzione, denominata “*Funzione di cambiamento*”.

$$\text{Funzione di cambiamento} = \frac{\text{Crisi Percepita}}{\text{Piano di Adozione Percepito}}$$

Il problema, secondo gli studi sopra citati, sembra risiedere principalmente nel fatto che le nuove funzionalità dei linguaggi (siano esse quantitative o qualitative) non si occupano sufficientemente di far sì che il denominatore (*Piano di Adozione Percepito*) sia il più elevato possibile. Quest'ultimo risulta essere più gestibile e controllato rispetto alla *Crisi Percepita*. I due fenomeni sono, infatti, inversamente proporzionali: un minore *Piano di Adozione* comporta un aumento considerevole della *Crisi Percepita*, a prescindere dal fatto che l'adozione sia realmente difficoltosa (oppure no). Esiste tuttavia un ulteriore fattore che potrebbe rendere la *Funzione di cambiamento* più sostenibile: i benefici percepiti. Essi vengono, però, considerevolmente temperati dal costo percepito. Quest'ultimo è generalmente più elevato rispetto a quello reale, ed è dovuto alla soggettività umana, così come descritto nel capitolo I ("Analisi Cognitiva"). La soggettività umana è però anche influenzata dalla demografia, infatti l'età di un programmatore o di un IT Manager può determinare la pensione (o meno) all'adozione di un nuovo linguaggio di programmazione.

4.2 Le barriere

Le barriere consistono in tutti quegli ostacoli che impediscono (oppure limitano) l'adozione di un linguaggio di programmazione. Esse sono principalmente di tipo cognitivo, e per il loro superamento è necessario sostenere dei "costi di adozione". Quest'ultimi possono essere monetari oppure psicologici e tendono ad essere posticipati nel tempo da parte dei programmatori, così da (apparentemente) evitarli, oppure al fine di acquisire più informazioni possibili.

Tuttavia, le barriere non costituiscono sempre e solo un ostacolo; infatti, esse possono essere appositamente sfruttate con lo scopo di aumentare la durabilità del linguaggio stesso. Inoltre possono fortemente limitare il manifestarsi di comportamenti "free

*riding*¹ e forzare (quasi inconsciamente) coloro che hanno adottato il linguaggio a comportarsi tutti allo stesso modo, ovvero a partecipare alla comunità e al suo sviluppo. Coerentemente con quanto appena descritto, la tendenza dei programmatori, a fronte di un non continuo aggiornamento del linguaggio, è quella di abbandonare lo stesso. Il motivo di una così importante rilevanza attribuita alla comunità e allo sviluppo risiede nel fine di garantire, a tutti gli utilizzatori, maggiore produttività e un corretto utilizzo del linguaggio. Spesso, infatti, può accadere che un linguaggio estremamente competitivo dal punto di vista delle funzionalità offerte, non sia egualmente diffuso rispetto a linguaggi molto più utilizzati e con una comunità più consistente. Questo perchè uno sviluppatore ha necessità di poter far affidamento su una comunità attiva e che mantenga il linguaggio costantemente aggiornato nel tempo.

4.3 Diffusione dell'innovazione

Everett M. Rogers è stato un teorico e sociologo della comunicazione americana, e ha dato origine alla “*Diffusion of Innovations Theory*” (teoria della diffusione dell'innovazione)[7]. Quest'ultima identifica cinque fasi che compongono il processo dell'adozione di un'innovazione (come appunto una nuova tecnologia o un linguaggio di programmazione):

1. *Knowledge* (conoscenza): un individuo è consapevole dell'innovazione ma non l'ha ancora approfondita;
2. *Persuasion* (persuasione): l'individuo è interessato e in cerca informazioni sull'innovazione;
3. *Decision* (decisione): un individuo valuta i pro e i contro e prende la decisione di adottare l'innovazione;

¹Il fenomeno del *free rider* ha luogo quando, all'interno di un gruppo di individui, si ha un membro che evita di dare il proprio contributo al bene comune, poiché ritiene che il gruppo possa funzionare ugualmente nonostante la sua astensione.

4. *Implementation* (implementazione): un individuo adotta l'innovazione e ne analizza gli usi;
5. *Confirmation* (conferma): l'individuo finalizza la decisione di adozione, ad esempio implementandola nell'intero proprio sistema e pubblicizzandola.

La prima fase (conoscenza) è quella più delicata, dove l'innovazione ha necessità di essere espressa con semplicità, al fine di essere più facilmente ricordata dalle persone. Come descritto nel capitolo I, si tratta di *System 1*.

La seconda, invece, consiste nella persuasione, dove l'individuo deve essere "accompagnato" e guidato nel ragionamento. Quest'ultimo corrisponde proprio a *System 2*, al quale è necessario fornire una chiara spiegazione (sia essa scritta oppure vocale) di quali sono i vantaggi dell'innovazione e per quali motivi dovrebbero essere interessanti per la persona a cui li si sta spiegando.

La terza fase corrisponde alla decisione, dove un individuo dovrebbe avere la possibilità di provare lui stesso l'innovazione, osservandone direttamente i risultati. La valutazione dei pro e contro (relativamente anche a ciò che già esiste o si utilizza) avviene però senza doverla implementare effettivamente nel proprio sistema, ma già pensando a come e se questo potrebbe avvenire.

La penultima fase è quella dell'implementazione vera e propria, dove l'individuo ha già deciso di adottare la nuova tecnologia e inizia ad utilizzarla realmente, analizzandone i diversi possibili usi. Per un IT manager, ad esempio, significa effettuare un'analisi del proprio sistema, individuando una possibile porzione di applicativo nel quale provare la nuova tecnologia ed infine implementarla.

L'ultima fase corrisponde alla conferma, ovvero alla finalizzazione della decisione di adozione. Questa avviene nel momento in cui la nuova tecnologia è impiegata effettivamente in tutto il sistema e la si pubblicizza ad altri.

Fattori di successo

Rogers, inoltre, esplicita cinque fattori di successo di un'innovazione, corrispondenti ad ogni fase precedentemente descritta.

1. *Relative advantage* (vantaggio relativo): l'innovazione deve essere percepita come migliore rispetto alle soluzioni già disponibili o adottate. Il grado di "vantaggio relativo" può essere misurato in termini economici, ma altre componenti possono entrare in gioco, quali i fattori di prestigio locale, la convenienza o la soddisfazione personale;
2. *Compatibility* (compatibilità): l'innovazione deve essere percepita come coerente rispetto ai valori esistenti, all'esperienza precedente e ai bisogni di chi la dovrebbe adottare;
3. *Complexity* (complessità): si tratta del grado in cui un'innovazione è percepita come difficile da comprendere e da utilizzare. Alcune innovazioni sono facili da capire e si diffondono più rapidamente rispetto a quelle complesse;
4. *Triability* (sperimentabilità): l'innovazione deve avere caratteristiche tali da poter essere sperimentata su basi limitate. Innovazioni "non divisibili" (che devono cioè essere assunte nel loro complesso, senza poter essere prima provate) si diffondono con maggiore difficoltà;
5. *Observability* (osservabilità): l'innovazione che produce risultati visibili ha maggiori possibilità di diffusione.

Il processo di adozione può fallire in una qualsiasi delle fasi appena descritte. Per questo motivo è estremamente importante far sì che esse vengano ideate e realizzate nel migliore dei modi, così da garantire un'alta probabilità di un'effettiva adozione.

L'incertezza del futuro

La principale barriera che i programmatori devono affrontare è quella dell'incertezza del futuro, dovuta all'imprevedibilità (umana e ambientale) e alla mancanza di informazioni. In particolar modo, questa barriera viene amplificata dalla naturale condizione di un'innovazione emergente: la ristretta dimensione della comunità; questo fa sì che i programmatori siano ancor più reticenti nei confronti dell'adozione di una nuova tecnologia.

Nel capitolo I si è approfondita l'analisi cognitiva grazie allo studio di Kahneman[2] sul processo decisionale. Quest'ultimo, infatti, si rispecchia perfettamente con la scelta di un programmatore di adottare un nuovo linguaggio.

La teoria del prospetto La teoria del prospetto è una teoria della decisione formulata dagli psicologi israeliani Daniel Kahneman e Amos Tversky nel 1979[10]. Essa si propone di fornire una descrizione di come gli individui effettivamente si comportino di fronte ad una decisione, in particolare in condizioni di rischio. Attraverso numerosi esperimenti di psicologia cognitiva, infatti, viene dimostrato come le scelte degli esseri umani violino sistematicamente i principi della razionalità economica. In particolare, i due autori focalizzano l'attenzione su tre importanti fenomeni psicologici, in realtà collegati fra loro:

- **Framing effect** (effetto contesto): il *frame*, cioè il contesto in cui l'individuo si trova a operare la scelta, ha un effetto determinante sulla scelta stessa. Il modo in cui il problema viene formulato influisce sulla percezione riguardo il punto di partenza rispetto cui valutare i possibili esiti delle proprie azioni;
- **L'avversione alle perdite**: per la maggior parte degli individui, la motivazione per evitare una perdita è superiore alla motivazione per realizzare un guadagno. Questo principio psicologico generale (probabilmente collegato a una sorta di istinto di sopravvivenza) fa sì che la stessa decisione possa dare origine a scelte opposte, se gli esiti vengono rappresentati al soggetto come perdite, piuttosto che come mancati guadagni. Ad esempio, è più facile rinunciare a un possibile sconto piuttosto che accettare un aumento di prezzo, anche se la differenza tra il prezzo iniziale e quello finale è la stessa[11];
- **Isolation effect** (effetto isolamento): la mente umana fatica a processare molte informazioni complesse, tutte insieme; decide perciò di selezionarne ed isolarne solo alcune, senza considerarle tutte facenti parte di un unico sistema. Il rischio diventa, quindi, quello di non operare scelte corrette perché non basate sulle giuste informazioni (derivanti invece dalla globalità). I due

psicologi hanno dimostrato che, in condizioni d'incertezza, la mente tende a semplificare al massimo il flusso di informazioni che riceve e deve processare, ragionando così secondo euristiche (“scorciatoie di pensiero”), che hanno l'effetto di produrre comportamenti non in linea con scelte razionali.

La teoria del prospetto comprende anche il concetto della *funzione valore*, la quale rappresenta graficamente i principi appena descritti.

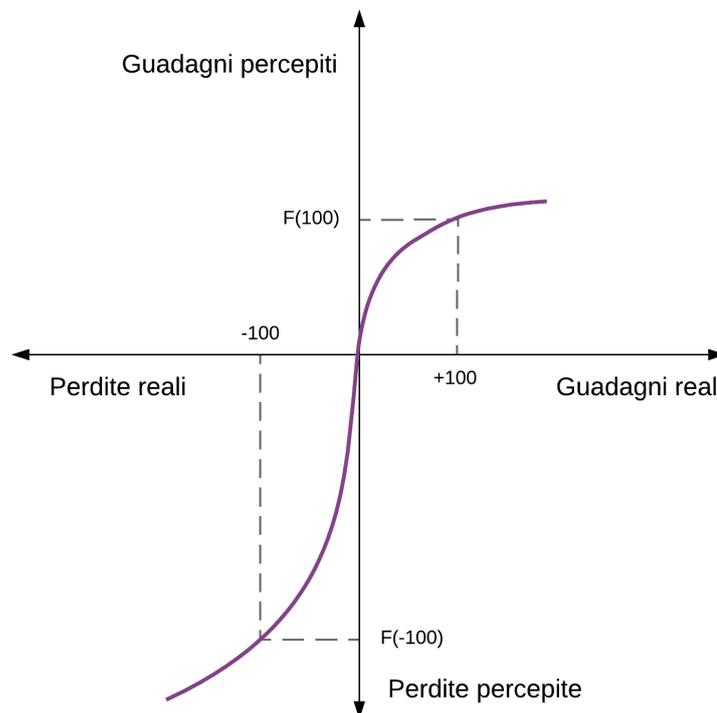


Figura 4.1: Rappresentazione grafica della *funzione valore*

Kahneman e Tversky osservano che la “differenza in termini di valore soggettivo [cioè percepito] tra un guadagno di \$100 e uno di \$200 è maggiore della differenza soggettiva tra un guadagno di \$1.100 ed un guadagno di \$1.200”. Il processo decisionale, secondo la *funzione valore*, ha inizio al centro del grafico, il quale corrisponde alla ricchezza di una persona al momento della valutazione delle possibili decisioni. Essa effettua, però, solo minimi spostamenti rispetto alla situazione iniziale, ed è per

questo motivo che è estremamente importante far sì che sia il quanto più possibile non condizionato dai *bias* (descritti nel capitolo I). La situazione iniziale è determinata anche dal *Framing effect*, ovvero le credenze, i sistemi di cultura e di pensiero degli individui.

Inoltre, la curva rappresentata è più ripida nel quadrante negativo delle perdite. Ciò significa che le persone sono avverse al rischio, il che si traduce in una maggiore propensione al non guadagno, rispetto alla perdita. Analizzando l'esperimento dei \$100/\$200 contro i \$1.100/\$1.200, si evince che l'attribuzione di valore a questa differenza (reale) fra gli importi, è soggettivamente influenzata rispetto a come viene percepita. Per una persona che possiede \$100, arrivare ad ottenerne \$200 significherebbe un incremento del 100% della propria ricchezza; mentre per la persona che ne possiede \$1.100, raggiungerne \$1.200 comporterebbe un aumento di solo poco più del 9%. Questo ragionamento razionale non viene però seguito se, a fronte comunque di guadagni, è previsto anche un rischio di perdita. In tale caso il ragionamento (istintivo) attuato dagli individui assume caratteri opposti. La persona che possiede \$100 preferisce non scommettere (e quindi non mettere a rischio) per ottenere altri \$100, poiché se li perdesse non avrebbe più denaro (\$0).

L'intera analisi è riconducibile all'adozione di un'innovazione, dal momento in cui il futuro e la non completa e perfetta informazione determinano la creazione di uno scenario di incertezza (rischio).

Tuttavia, il problema principale, ovvero le dimensioni della comunità dell'innovazione, non sarà mitigato se non vi sono programmatori che iniziano ad adottare l'innovazione stessa. Per questo motivo ci si avvale di studi sulla cognitività umana[12]: migliorare sempre più la comunicazione di tali innovazioni al fine di semplificarle e agevolarne l'adozione.

Le categorie di utilizzatori

Rogers, nel suo studio sulla teoria della diffusione dell'innovazione[7], identifica una curva (denominata "curva di Rogers") la quale classifica in varie categorie coloro che adottano le innovazioni, in base all'idea che alcuni individui sono inevitabilmente

più propensi all'innovazione rispetto ad altri.

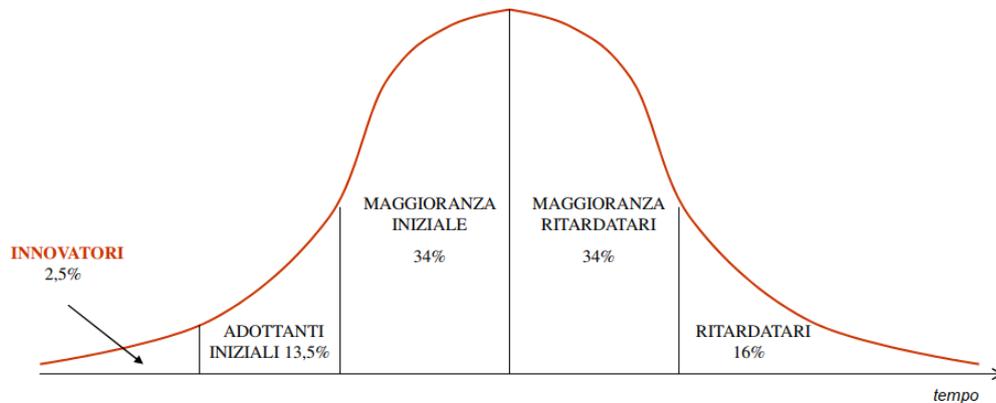


Figura 4.2: Rappresentazione grafica della *curva di Rogers*

La *curva di Rogers* è una curva normale “a campana” e distingue le seguenti categorie:

- **Innovatori**: sono caratterizzati da un elevato livello di istruzione, sono avversi al rischio (anche in termini di risorse), possiedono abilità specifiche nella comprensione e nell'applicazione delle conoscenze tecniche e sono esposti a più fonti di informazione;
- **Adottanti iniziali**: sono dotati di alti livelli di istruzione, elevata reputazione nella comunità, capacità di svolgere una funzione di leadership sociale e con esperienze di successo alle spalle. All'interno delle aziende sono influenzatori di opinioni, cioè fungono da chiave per convincere gli altri, perchè i potenziali adottanti si rivolgono a loro per avere consigli e informazioni sull'innovazione;
- **Maggioranza iniziale**: caratterizzata da soggetti che hanno una forte interazione con i pari; spesso hanno la tendenza a seguire un processo deliberativo di maggiore durata prima di adottare una nuova tecnologia, ma è grazie a loro che viene avviata la diffusione dell'innovazione su ampia scala;

- **Maggioranza ritardataria:** comprende soggetti tendenzialmente scettici, tradizionalisti, con uno status economico relativamente basso, prudenti e che subiscono molto le eventuali pressioni sociali che spingono all'adozione. Di solito vogliono avere la possibilità di osservare le conseguenze dell'innovazione prima di adottarla;
- **Ritardatari:** individui normalmente isolati, sospettosi, con relazioni sociali ridotte (solo vicini o parenti), con un processo decisionale lento e dotati di risorse limitate.

Le categorie appena descritte sono state delineate in termini di popolazione, ma possono essere facilmente ricondotte ai potenziali utilizzatori di JOLIE (aziende e singoli sviluppatori). Inoltre, è possibile dedurre le seguenti informazioni:

- La propensione di una persona all'adozione di un'innovazione influisce sul tasso di assorbimento (diffusione) delle stesse nel tempo;
- Gruppi diversi hanno propensioni diverse ad investire in innovazione e lo fanno per ragioni diverse e con aspettative diverse;
- Le persone che appartengono alle categorie degli innovatori e adottanti iniziali sono tendenzialmente più facili da convincere ad acquisire nuove soluzioni;
- La maggioranza (iniziale e ritardataria) che costituisce il 68% di una popolazione è quella che determina il successo o meno di una innovazione;
- Gli innovatori possono richiedere più flessibilità e minor controllo, mentre la maggioranza può richiedere una maggiore stabilità e supporto.

La rapida espansione della maggior parte delle innovazioni di successo si verifica quando i fattori sociali e tecnici lo consentono e la alimentano.

Il concetto di categorie è importante perché mostra il fatto che tutte le innovazioni devono attraversare un processo naturale, prevedibile, e talvolta lungo, prima di diventare ampiamente adottate all'interno di una popolazione, azienda o anche singoli individui stessi.

Capitolo 5

Sito realizzato

In questo capitolo si affronta l'analisi della struttura e della forma dei contenuti presenti nel *mockup*¹ del sito realizzato. I principi che hanno guidato la stesura del *mockup* sono gli stessi descritti e articolati in tutti i capitoli precedenti.

Premessa L'attuale sito *web* di JOLIE e il relativo nuovo *mockup* sono stati sviluppati utilizzando **Leonardo**, il *server web*² creato interamente in JOLIE, congiuntamente con altri appositi linguaggi di programmazione (ad es. HTML, CSS e Javascript) e relative librerie (Bootstrap³ e jQuery⁴). Per agevolare l'internazionalizzazione di JOLIE, si è deciso di scrivere l'intero sito in lingua inglese.

5.1 Homepage

La Homepage del sito, di seguito chiamata "Home", è la pagina più importante, poiché è tendenzialmente la prima che un utente visita. Data la poca diffusione di JOLIE, la Home deve poter "attirare" e trattenere il maggior numero di visitatori

¹Il *mockuping* consiste nel costruire un progetto finale dal punto di vista grafico, ma senza l'interattività di un prototipo, rappresentando nel dettaglio i vari contenuti e le funzionalità base dell'applicazione *web* in maniera statica.

²Un *server web* è un'applicazione *software* che, in esecuzione su un *server*, è in grado di gestire le richieste di trasferimento di pagine web di un *client*, tipicamente un *web browser*.

³[<https://getbootstrap.com>]

⁴[<https://jquery.com>]

possibili. Questa fase corrisponde al Livello I della *roadmap* della Strategia.

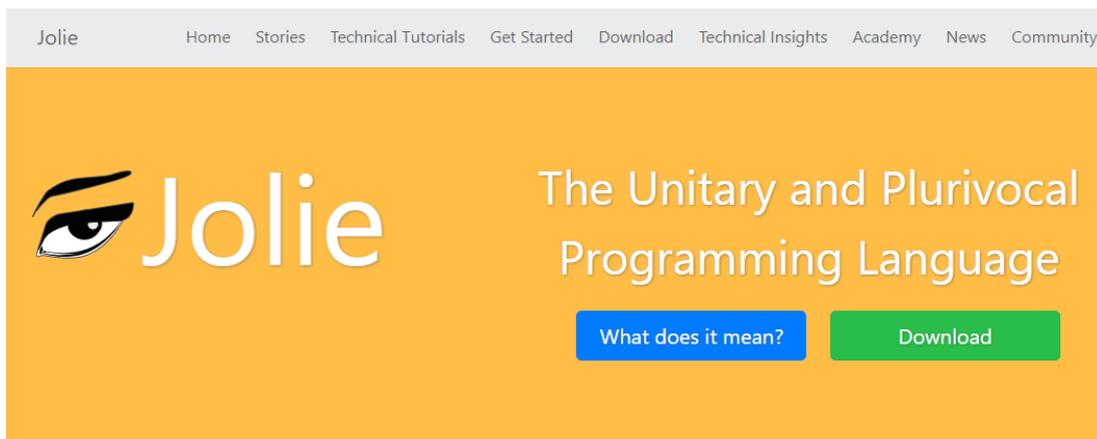


Figura 5.1: Menu e *slogan* della Homepage

La parte superiore contiene il menu: una serie di collegamenti rapidi alle pagine più importanti del sito.

Nella parte inferiore, invece, è posto a sinistra il nome “Jolie” a caratteri grandi e ben visibili, con affianco il suo logo originale. Alla sua destra è presente lo *slogan*: “*The Unitary and Plurivocal Programming Language*”. Data la possibile curiosità (volutamente) suscitata, è presente un bottone “What does it mean?” che riconduce ad una pagina di spiegazione di entrambi i termini (riportata di seguito).

The Unitary and Plurivocal Programming Language

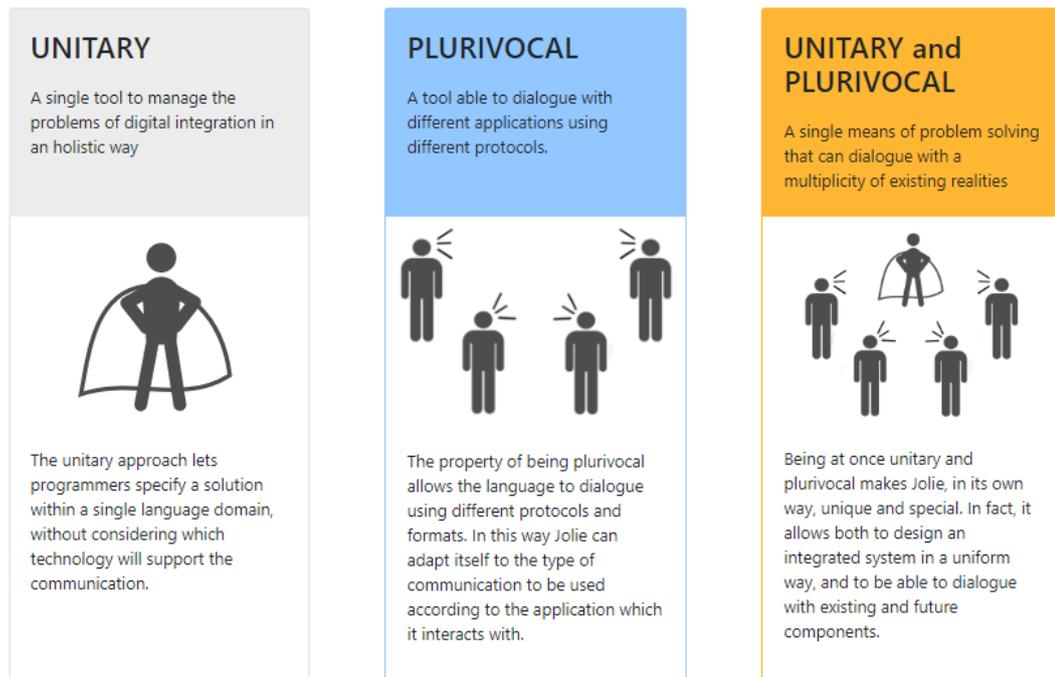


Figura 5.2: Pagina spiegazione *slogan*

Nella Home, accanto al pulsante “What does it mean?” è presente quello di “Download”, in modo da favorire la propensione a provare il linguaggio nell’immediato. Al di sotto della prima sezione, è presente la diramazione per i diversi profili:

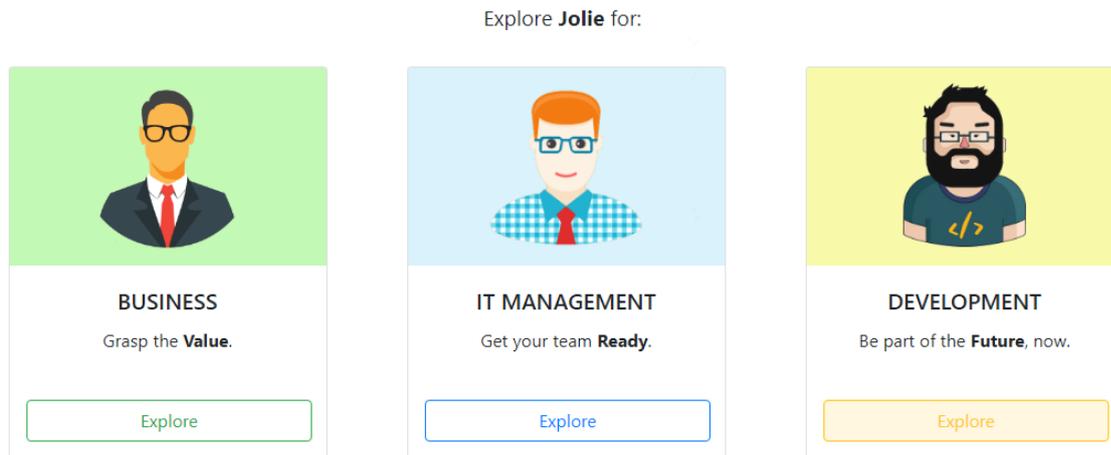


Figura 5.3: Sezione Homepage: Esplorazione profili

Lo scopo di questa breve sezione è quello di conoscere JOLIE guidando il visitatore secondo il ruolo che più gli si avvicina, attraverso la narrazione di un'apposita storia (Livello II della Strategia). Per ognuno di essi si è costruito un sotto-*slogan* (volutamente molto breve e conciso) il quale cogliesse il valore che JOLIE assume per loro. Il Businessman è prevalentemente interessato al valore monetario, l'IT Manager al proprio *team*, mentre il *developer* al futuro. Il titolo di ogni ruolo è stato scelto in base alla professione, così da garantire limitata impersonificazione dell'utente visitatore, già molto coinvolto dall'*avatar* (immagine) in questa fase, e dalla storia in quella successiva.

Proseguendo nella Home si trova poi un *focus* sui microservizi, il cui scopo è quello di risaltare l'efficacia di JOLIE rispetto ad essi.



Focus on Microservices

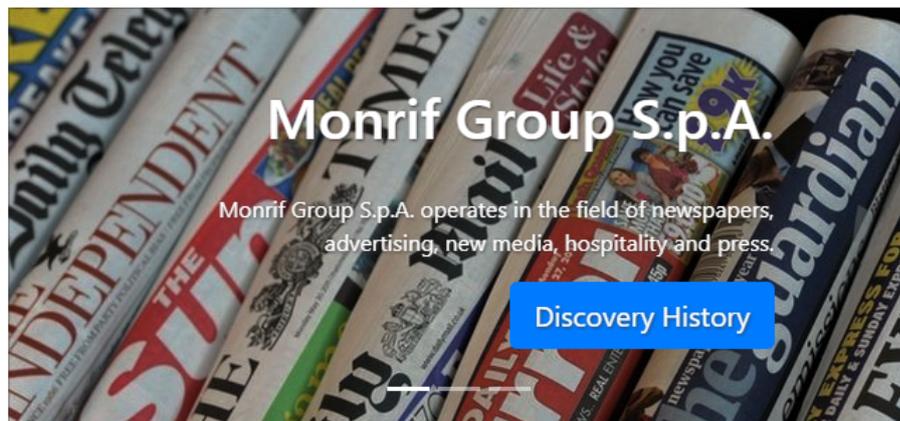
Thinking in Microservices [Read More](#)

Jolie crystallises the programming concepts of microservices as native language features: the basic building blocks of software are not objects or functions, but rather **Services** that can always be relocated and replicated as needed. Distribution and reusability are achieved by design.

Figura 5.4: Sezione Homepage: *Focus* sui microservizi

Al di sotto del *focus* sui microservizi vi sono le “Top Stories”, ovvero le principali storie di aziende che hanno adottato JOLIE. La prima presentata è quella di “Monrif Group S.p.A.”, la cui intervista è stata interamente riportata nel Livello II della strategia (capitolo 2).

Top Stories



Monrif Group S.p.A.

Monrif Group S.p.A. operates in the field of newspapers, advertising, new media, hospitality and press.

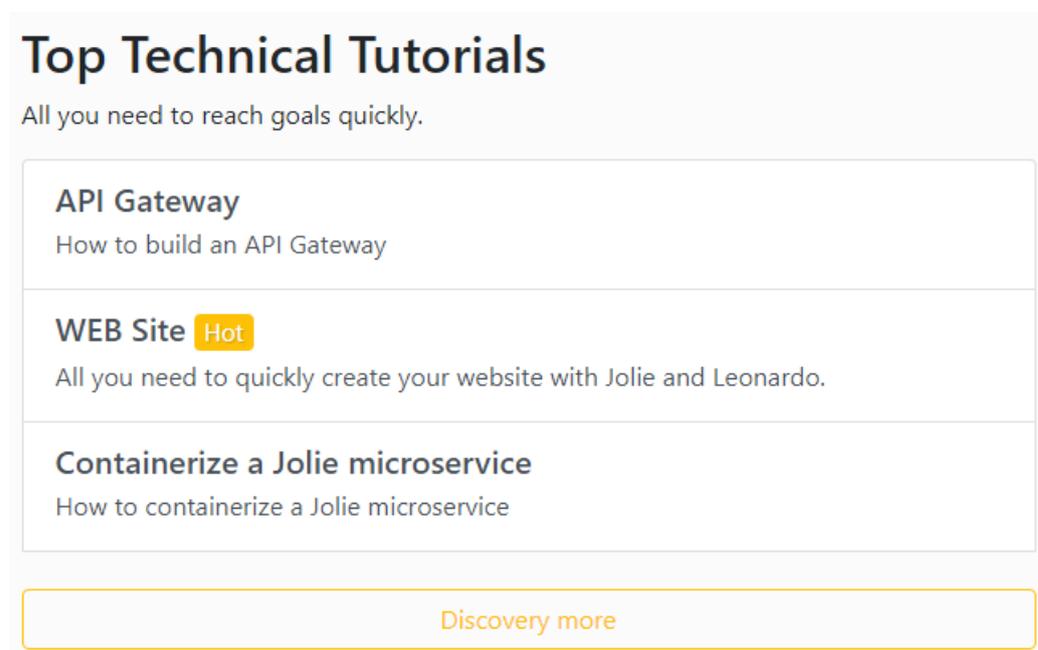
[Discovery History](#)

Figura 5.5: Sezione Homepage: *Top Stories*

Lo scopo di questa sezione è quello di far emergere il fatto che aziende vere hanno già adottato JOLIE ottenendo grandi risultati e traendo (tuttora) importanti bene-

fici.

Proseguendo la navigazione verso il basso, si trovano alcuni “Top Technical Tutorials” (Tutorial Tecnici), ovvero il Livello III della strategia. Offrendo questa “scorciatoia” si fa sì che coloro (prevelentemente sviluppatori) che vogliono implementare velocemente alcuni argomenti, abbiano la possibilità di farlo in modo guidato e preciso.



Top Technical Tutorials
All you need to reach goals quickly.

API Gateway
How to build an API Gateway

WEB Site Hot
All you need to quickly create your website with Jolie and Leonardo.

Containerize a Jolie microservice
How to containerize a Jolie microservice

[Discovery more](#)

Figura 5.6: Sezione Homepage: *Top Technical Tutorials*

Nella sezione successiva si hanno, sulla sinistra, i “Technical Insights” (approfondimenti tecnici, Livello IV della strategia) evidenziati per difficoltà (incrementale), mentre, sulla destra, le “Recent News” (ultime notizie), suddivise per tipologia (evento oppure notizia).

Technical Insights

Practical "hand-on" guides on specific topics.

Server - Client Easy

A Jolie program defines a service and is a composition of two parts, called *behaviour* and deployment.

Server always ON Easy

Server is a computer program or a device that provides functionality for other programs or devices.

Interfaces Intermediate

Thus an interface is a collection of operation types, a list of One-Way and Request-Response operation declarations.

Choice Intermediate

The input choice implements input-guarded choice. Namely, it supports the receiving of a message for any of the statements...

Sessions Master

Having multiple instances of a behaviour running in a service introduces the problem of routing incoming messages to...

[Find out more](#)

Recent News

Latest news from Blog and Events.

A new look for the Jolie documentation pages Event today

The Jolie documentation pages, which you can find at...

Jolie code is now recognised by Github News 5 days ago

A few months ago, Github merged support for recognising Jolie...

Jolie 1.6.2 released News 7 days ago

Jolie 1.6.2 has been released! Go get it from our download page.

A Vision for the Microservice Revolution News 10 days ago

Is there only one way to develop Microservices? Or is there...

Jolie 1.6.1 released News 15 days ago

Jolie 1.6.1 has been released! Go get it from our download page.

[Read more](#)

Figura 5.7: Sezioni Homepage: *Technical Insights* e *Recent News*

A concludere la Home vi è una sezione “Academia”, dedicata principalmente ai “Latest Scientific Papers”:

Academia

Latest Scientific Papers

Service-oriented Programming with Jolie Fabrizio Montesi, Claudio Guidi, Gianluigi Zavattaro. In Web Services Foundations.	10 Gen, 2018
Process-aware Web Programming with Jolie (Extended Version) Interesting Fabrizio Montesi Submitted for publication	5 Dec, 2017
Implementing Elastic Capacity in a Service Oriented PaaS A. Zuccato, T. Vardanega 4th Int'l Workshop on Adaptive Services for Future Internet (WAS4FI 2014), 2 September 2014 (to appear)	25 Sep, 2017

[See others](#)

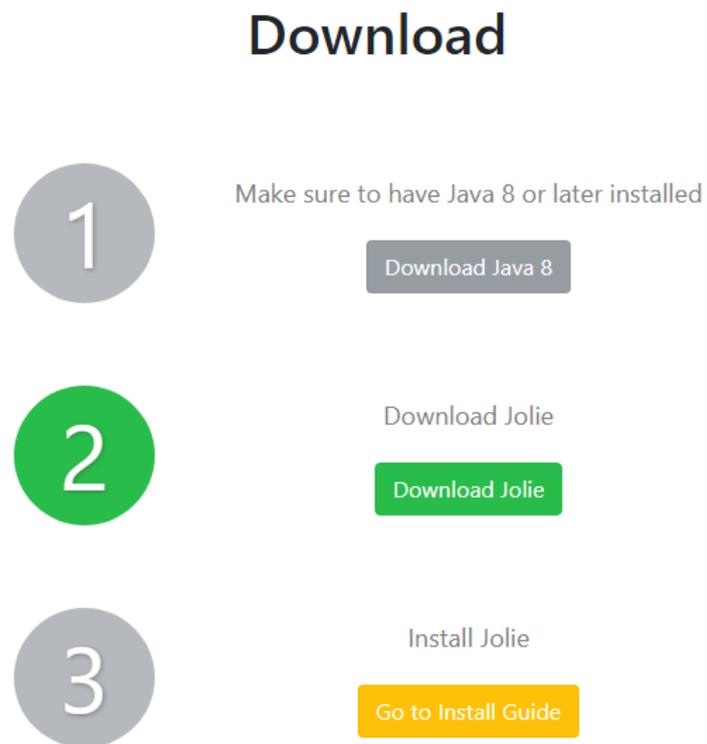
Figura 5.8: Sezione Homepage: *Academia*

5.2 Download *workflow*

La sezione dedicata al *Download* è stata estremamente semplificata e riorganizzata rispetto alla versione preesistente. Lo scopo è quello di far sì che l'utente, che per la prima volta si approccia a JOLIE, possa scaricarlo e soprattutto eseguire i programmi in modo chiaro e semplice. La nuova versione prevede una serie di pagine raccolte in un flusso navigabile (*workflow*).

1) Download

Nella pagina di download ci si assicura che l'utente abbia installato (oppure installi) i requisiti, che possa scaricare JOLIE e che passi, infine, alla sezione successiva: "Install Guide".

Figura 5.9: Pagina *Download*

2) Install Guide

Nella “Install Guide” viene dettagliatamente descritta la procedura per installare JOLIE. Il passo successivo consiste nell’installazione dell’IDE (ambiente di sviluppo integrato).

Install Guide

In order to install Jolie successfully, follow these instructions:

1. Run the Shell Terminal as ADMINISTRATOR
2. Change your path corresponding where you have downloaded the Jolie Installer
3. Execute the command "java -jar jolie-1.6.2.jar"
 - on Linux/MacOS, by executing the installer as root, write the command "sudo java -jar jolie-1.6.2.jar"
4. Follow the displayed instructions

Great! Now you're ready to enjoy Jolie:

[Setup the IDE for Jolie](#)

Figura 5.10: Pagina *Install Guide*

3) Integrated Development Environment

Questa sezione ha un duplice scopo: far installare all'utente l'IDE (Integrated Development Environment) appropriato e inicializzarlo per programmare in JOLIE. La pagina successiva ("Run Jolie") consiste nell'eseguire il primo programma.

Integrated Development Environment

We suggest to install the IDE "Atom":

[Download Atom](#)

Setup Atom for Jolie

Once installed Atom:

1. Open Atom
2. Click on "File" > "Settings"
3. Select the left tab "Install"
4. Search for "Jolie" Packages
5. Install the package "language-jolie"

If you'd like to have also a linter (code and error highlighter), install also the package "linter-jolie".

Now you're ready to run your first Jolie program:

[Run my first Jolie program](#)

Figura 5.11: Pagina *The IDE*

4) Run Jolie

Dal momento in cui l'utente è riuscito ad installare JOLIE e ad inizializzare l'IDE, è necessario fornire lui una base su cui iniziare a programmare, in modo da ottenere subito dei risultati visualizzabili. Per questo motivo si è creato un apposito programma - "Hello, World Jolie!" - che l'utente può scaricare ed eseguire facilmente, grazie alle istruzioni riportate.

Run Jolie

"Hello, World" Jolie!

This is the simplest program you can make with Jolie. It prints a "Hello, World!" string as output. Try it now to run your first Jolie program!

[Download "Hello, World!"](#)

Run It

To run a Jolie program you have only to start a shell command where the jolie file is located, and write "jolie [file name].ol".

For example "jolie hello_world.ol".

Go Further!

Are you ready to go beyond the simple "Hello, World!" program? Find out something a bit more interesting here:

[Go to Guide](#)

Figura 5.12: Pagina *The IDE*

5.3 Technical Insights

La sezione dedicata ai "Technical Insights" (approfondimenti tecnici) è stata realizzata seguendo le specifiche descritte nel Livello IV della strategia (capitolo 2).

 BASIC - "Server and Client"



Description A Jolie program defines a service and is a composition of two parts, called behaviour and deployment. A behaviour defines the implementation of the functionalities offered by a service; behavioural primitives include communication and computation constructs. However, these do not deal with how communications are supported: they abstractly refer to communication ports, which are assumed to be correctly defined in the deployment part.

Goal A simple Server - Client application: the client sends a number (5) to the server. The server return the number received multiplied by two, which will be printed by the client.

[Show Tips](#)

Server	Client
<pre>main { twice(number)(response) { response = number * 2; println@Console(number) } }</pre>	<pre>main { twice@TwiceService(5)(response); println@Console(response) }</pre>
<p>Output:</p> <pre>5</pre>	<p>Output:</p> <pre>10</pre>
<p>Download & Run</p>	

[Go to Next Insight](#)

Figura 5.13: Pagina *Technical Insights*

Il riquadro a sinistra contiene la descrizione dell'approfondimento tecnico e il suo scopo (*Goal*). Sulla parte destra, invece, ci sono le due rappresentazioni del codice contenuto in Server e in Client, al di sotto delle quali vi è il rispettivo *output*. L'utente può scaricare ed eseguire (manualmente) il codice proposto. Se, invece, si clicca sull'icona del menu in alto a sinistra (affianco al nome dell'argomento), apparirà l'elenco di tutti i Technical Insights disponibili, organizzati per livelli di difficoltà incrementale.

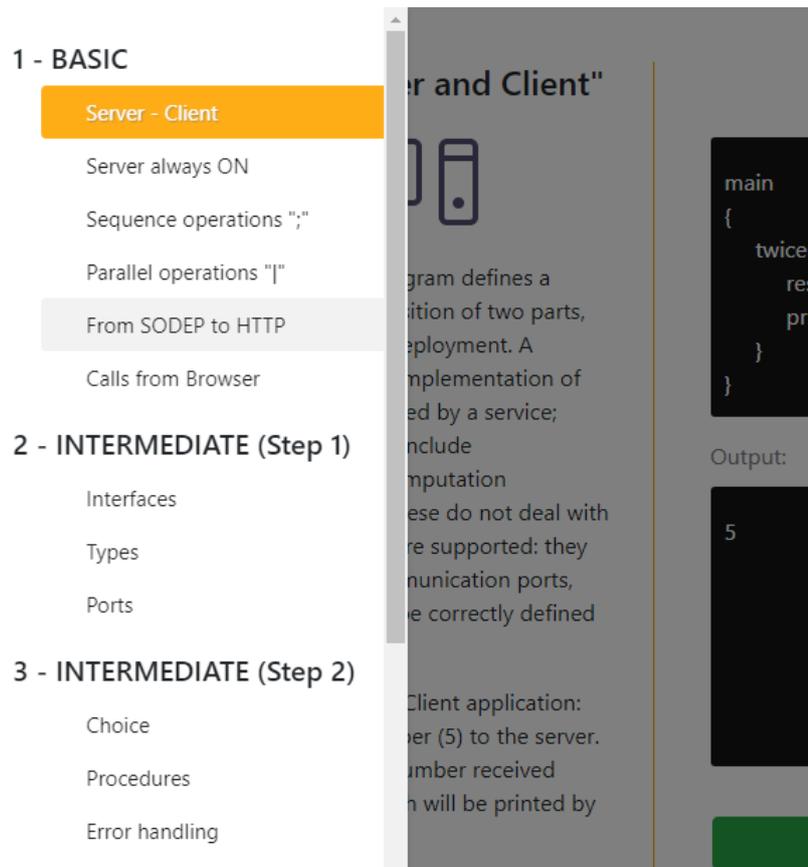


Figura 5.14: Pagina *Technical Insights*: il menu

Conclusioni

La scarsa diffusione di JOLIE ha creato la necessità di rivisitarne la comunicazione. Quest'ultima è stata riorganizzata attraverso la formulazione di una nuova strategia, avvalendosi di principi dedotti da studi sulla psicologia e cognitività umana.

Durante la fase iniziale di analisi si sono approfonditi gli aspetti e i motivi per cui la comunicazione fino ad ora adottata non ha portato ai risultati auspicati. La formulazione di uno scenario futuro, partendo da quello attuale, ha permesso di determinare l'obiettivo finale della nuova comunicazione.

La *roadmap* prodotta descrive dettagliatamente ogni fase della comunicazione, la quale è stata pensata e costruita per facilitare (in modo equilibrato ed opportuno) la comprensione di questa nuova tecnologia da parte delle persone. In particolare, si sono studiati ed analizzati i profili a cui la comunicazione è rivolta, facendo risaltare, coerentemente con le proprie inclinazioni professionali, i vantaggi che ognuno di essi può trarre dall'adozione di JOLIE.

La nuova strategia comunicativa realizzata, dunque, ha lo scopo di aumentare l'adozione di JOLIE da parte delle aziende e dei sviluppatori.

Bibliografia

- [1] Martin Fowler, “Microservices”, 2014.
[<https://martinfowler.com/articles/microservices.html>]
- [2] Daniel Kahneman, “Maps of Bounded Rationality: Psychology for Behavioral Economics”, 2003.
- [3] von Restorff, Hedwig, “Über die Wirkung von Bereichsbildungen im Spurenfeld” [The effects of field formation in the trace field], 1933.
[<https://link.springer.com/article/10.1007/BF02409636>]
- [4] Angela Fagerlin, Brian J. Zikmund-Fisher, Peter A. Ubel, “Helping Patients Decide: Ten Steps to Better Risk Communication”, 2011.
[<https://academic.oup.com/jnci/article/103/19/1436/899532>]
- [5] Michael Bar-Eli, Ofer Azar, Ilana Ritov, Yael Keidar-Levin, Galit Schein, “Action bias among elite soccer goalkeepers: The case of penalty kicks”, 2007.
[<https://econpapers.repec.org/RePEc:pra:mprapa:4477>]
- [6] Leo A. Meyerovich, Ariel Rabkin, “Socio-PLT: Principles for Programming Language Adoption”, 2012.
[<https://dl.acm.org/citation.cfm?id=2384597>]
- [7] Everett M. Rogers, “Diffusion of innovations”, 1995.
- [8] Christopher Vogler, “Il viaggio dell’eroe. La struttura del mito ad uso di scrittori di narrativa e di cinema”, 2010.
- [9] Vladimir Jakovlevič Propp, “Morfologia della fiaba”, 1928.

- [10] Daniel Kahneman, Amos Tversky, “Prospect Theory: An Analysis of Decision Under Risk”, 1979.
- [11] Amos Tversky e Daniel Kahneman, “Rational Choice and the Framing of Decisions”, 1986.
- [12] Daniel Kahneman, “Thinking, fast and slow”, 2011.

Ringraziamenti

Desidero innanzitutto ringraziare il professor. Davide Sangiorgi, relatore di questa tesi, per avermi permesso di realizzarla.

Un ringraziamento particolare va al mio correlatore, Saverio Giallorenzo, per la disponibilità, l'aiuto e i consigli che mi ha dato durante la realizzazione di questa tesi. Ringrazio inoltre Claudio Guidi, il quale mi ha permesso di svolgere il tirocinio per tesi nella sua azienda, dove abbiamo lavorato fianco a fianco per migliorare la comunicazione di questa nuova tecnologia.

Un ringraziamento davvero sentito al professor. Marco Ruffino, nonché correlatore, che ha contribuito con tutta la sua esperienza alle analisi in ambito cognitivo-psicologico e al perfezionamento di molti aspetti legati alla comunicazione.

Un grazie enorme e speciale va alla mia fidanzata, Mary, “semplicemente” per amarmi e darmi ogni giorno la forza di cui ho bisogno.

Un grande grazie lo rivolgo a Claudio, mio amico di una vita, e a tutti i miei amici di Santa Maria, con i quali ho e condivido una vita piena di risate e divertimento. Ringrazio inoltre i miei colleghi, nonché amici, arbitri di Pallacanestro, insieme ai quali condivido dure sfide e sacrifici, ma anche grandi soddisfazioni.

Ringrazio anche i miei colleghi di corso, in particolare Lorenzo ed Anna, che hanno vissuto insieme a me questa esperienza, rendendola più ricca ed entusiasmante.

Il ringraziamento più grande di tutti, però, è rivolto ai miei genitori: la soddisfazione di questo lavoro e dei traguardi raggiunti va a loro.