

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA

CORSO DI LAUREA IN INFORMATICA



**Realtà virtuale per colmare il divario tra
e-commerce e negozio fisico**

Tesi di laurea triennale

Relatore

Prof. Tullio Vardanega

Laureando

Simone Magagna
1009467

ANNO ACCADEMICO 2015-2016

Dedica

Sommario

Questo documento rappresenta la relazione finale di stage tenuta presso l'azienda The White Dog s.r.l. della durata complessiva pari a 308 ore.

Il primo capitolo tratta dell'azienda ospitante, dei prodotti e dei servizi che offre, in particolare all'azienda Diana Corp. della quale rappresenta il reparto ricerca e sviluppo. Il secondo capitolo descrive le strategie aziendali riguardanti l'attività di stage, di come il mio progetto si integri in esse, del piano di lavoro propostomi e degli obiettivi aziendali e personali.

Il terzo capitolo parla dell'effettivo sviluppo del progetto, descrivendo le fasi di pianificazione e ricerca effettuate con il mio tutor aziendale e il team di sviluppo, per arrivare a trattare dell'analisi dei requisiti, dei principi di progettazione perseguiti, dello sviluppo di alcune delle parti più peculiari e interessanti del progetto e della verifica e validazione.

Il quarto ed ultimo capitolo analizza i risultati ottenuti confrontandoli con gli obiettivi prefissati, valuta le conoscenze acquisite e discute quali tra queste debbano essere integrate nel corso di studi. Il capitolo si conclude con alcune considerazioni di carattere personale.

Ringraziamenti

In questa sezione ci saranno i ringraziamenti.

Indice

1	The White Dog s.r.l.	1
1.1	Chi è The White Dog s.r.l.	1
1.2	Prodotti e servizi	1
1.2.1	Servizio di supporto tecnologico	1
1.2.2	Il prodotto Live Story	3
1.3	Processi interni	4
1.4	Strumenti e tecnologie	8
1.4.1	Ambienti di sviluppo	8
1.4.2	Gestione dei progetti	8
1.4.3	Versionamento	9
1.4.4	Strumenti di automazione	9
1.4.5	Tecnologie di sviluppo	9
1.5	Ricerca e innovazione	9
2	Il quadro strategico	11
2.1	Strategie aziendali di stage	11
2.2	Il progetto di stage proposto	11
2.2.1	Piano di lavoro proposto	13
2.2.2	Obiettivi aziendali	17
2.2.3	Obiettivi personali	18
3	Il progetto di e-commerce VR	19
3.1	Pianificazione del lavoro	19
3.2	Ricerca e sperimentazione	19
3.3	Tecnologie adottate	19
3.4	Analisi dei requisiti	19
3.4.1	Caratteristiche degli utenti	19
3.4.2	Casi d'uso	19
3.4.3	Requisiti	20
3.5	Progettazione	20
3.5.1	Portabilità dell'applicazione	20
3.5.2	Usabilità dell'applicazione	20
3.5.3	Costruzione della scena 3D	20
3.5.4	Interazione con gli oggetti all'interno della scena	20
3.5.5	Progettazione e integrazione con AWS API Gateway	20
3.6	Sviluppo	20
3.6.1	Sviluppo degli oggetti interattivi	20
3.6.2	Creazione a runtime di oggetti interattivi	20

3.6.3	Dati persistenti attraverso le scene	21
3.6.4	Unity e il protocollo HTTP	21
3.6.5	Creazione e parsing di oggetti JSON in Unity	21
3.7	Verifica e validazione	21
4	Analisi retrospettiva	23
4.1	Bilancio dei risultati rispetto agli obiettivi prefissati	23
4.2	Bilancio formativo	23
4.3	Analisi critica del rapporto formativo tra stage e corso di laurea	23
4.4	Valutazioni personali	23
	Glossario	25
	Bibliografia	29

Elenco delle figure

1.1	Logo dell'azienda The White Dog s.r.l.	1
1.2	Ambiti aziendali gestiti dal sistema ERP	3
1.3	<i>Wall mood</i> creato tramite <i>Live Story</i> per l' <i>e-commerce</i> di una nota marca di occhiali	4
1.4	Ciclo di sviluppo della Metodologia Agile	5
1.5	Competenze necessarie alla metodologia di sviluppo DevOps	6
1.6	Ciclo di sviluppo software Extreme Programming	7
1.7	Esempio di <i>scrum board</i> all'interno del software Jira	8
2.1	Schema rappresentativo della differenze tra <i>single-channel</i> , <i>multi-channel</i> , <i>cross-channel</i> e <i>omni-channel</i>	12
2.2	Samsung Gear VR	15
2.3	Logo Unity	16
2.4	Stack tecnologico di sviluppo per Samsung Gear VR	17
2.5	Logo dell'evento Stage IT 2016	18

Elenco delle tabelle

Capitolo 1

The White Dog s.r.l.

1.1 Chi è The White Dog s.r.l.

The White Dog s.r.l. è una realtà aziendale nata nel 2008 con sede a Torreglia, in provincia di Padova. Essa è stata fondata dal signor Stefano Mocellini, fondatore e CEO di Diana Corp.¹, con la volontà di creare un *team* di lavoro focalizzato sulla ricerca e sviluppo.

The White Dog s.r.l. coordina e gestisce società tutte affini al settore *e-commerce*, come Diana Corp. e LiveStory². L'azienda possiede un reparto di ricerca e sviluppo denominato R&D, il quale esplora nuove tecnologie da applicare poi alle società figlie nel caso di esito positivo o facendo nascere nuovi progetti separati.



figura 1.1: Logo dell'azienda The White Dog s.r.l.

1.2 Prodotti e servizi

L'azienda svolge principalmente due attività: la prima legata al servizio di supporto tecnologico offerto a Diana Corp., la seconda legata alla gestione e sviluppo del prodotto Live Story.

1.2.1 Servizio di supporto tecnologico

Il principale servizio che l'azienda offre a Diana Corp. è la ricerca e lo sviluppo di nuove tecnologie da applicare nell'ambito del *fashion e-commerce*. Essa svolge l'attività

¹<http://www.dianacorp.com/>

²<http://www.livestory.nyc/>

di *testing*_G dei nuovi strumenti presenti nel mercato, li valuta attentamente in termini di prestazioni e costi, per poi renderli disponibili agli sviluppatori di Diana Corp. attraverso consulenze.

The White Dog s.r.l., inoltre, interviene direttamente nel codice dei servizi Diana Corp. se a quest'ultima vengono commissionate nuove *features* o applicazioni che, per tempistiche o competenze, non è in grado da sola di sviluppare.

Infine, offre consulenza a Diana Corp. per l'utilizzo delle seguenti piattaforme di *management*:

- * **SAP:** sistema informativo aziendale per la gestione di tutti i processi dell'azienda: vendite, acquisti, gestione magazzino, contabilità eccetera. SAP supporta il sistema di gestione ERP, che permette di ottimizzare la gestione di tutte le attività del business intervenendo nel risparmio dei costi, nei tempi decisionali aziendali e negli investimenti.

Le principali caratteristiche di un sistema ERP sono:

- Database comune a tutte le applicazioni e unico;
- Struttura modulare ed integrata;
- Scalabilità del sistema.

- * **Magento:** è una piattaforma *e-commerce open source* che offre la possibilità di avviare una attività online di commercio elettronico con ampia flessibilità sia nella grafica, sia nelle funzionalità che nei contenuti.

L'interfaccia di amministrazione fornisce strumenti di marketing, SEO e gestione del catalogo, in modo da offrire ai commercianti la possibilità di creare siti a misura delle proprie esigenze di business. Progettato per essere completamente scalabile e per essere ripristinato facilmente.

- * **WordPress:** è una piattaforma software di *personal publishing e content management system*_G, ovvero un programma che consente la creazione e distribuzione di un sito internet formato da contenuti testuali o multimediali, facilmente gestibili ed aggiornabili in maniera dinamica.



figura 1.2: Ambiti aziendali gestiti dal sistema ERP

1.2.2 Il prodotto Live Story

Live Story è un *social management system*_G che gestisce contenuti *social* e li rende acquistabili. Il *concept* di Live Story è nato nel reparto R&D di The White Dog s.r.l., *concept* che è diventato poi azienda nel 2015 con sede a New York.

Live Story colleziona foto degli utenti dei *social network* marcate con un particolare *hashtag* che rappresenta l'azienda che vuole utilizzare il servizio. Il sistema accoppia la foto ad un particolare prodotto presente nel catalogo e genera automaticamente le richieste di permesso di utilizzo della foto e la invia all'utente interessato. Se l'utente approva e il moderatore ritiene conforme la foto, l'azienda può utilizzare il contenuto nel proprio sito o *e-commerce*.

Una volta approvati, i contenuti possono comporre un *wall*, ovvero un pannello visibile in una pagina web, nel quale sono posizionati in maniera *custom*. Vi sono due tipi di *wall* che Live Story permette di comporre:

- * **Mood:** *wall* creati con contenuti e foto scelti dall'azienda. In questo caso non vi è nessuna interazione con gli utenti dei social network, ma solo una creazione più rapida dei contenuti visualizzabili nell'*e-commerce*;
- * **Feed:** *wall* creati automaticamente tramite i contenuti marcati dall'*hashtag* aziendale presenti nei *social network*.

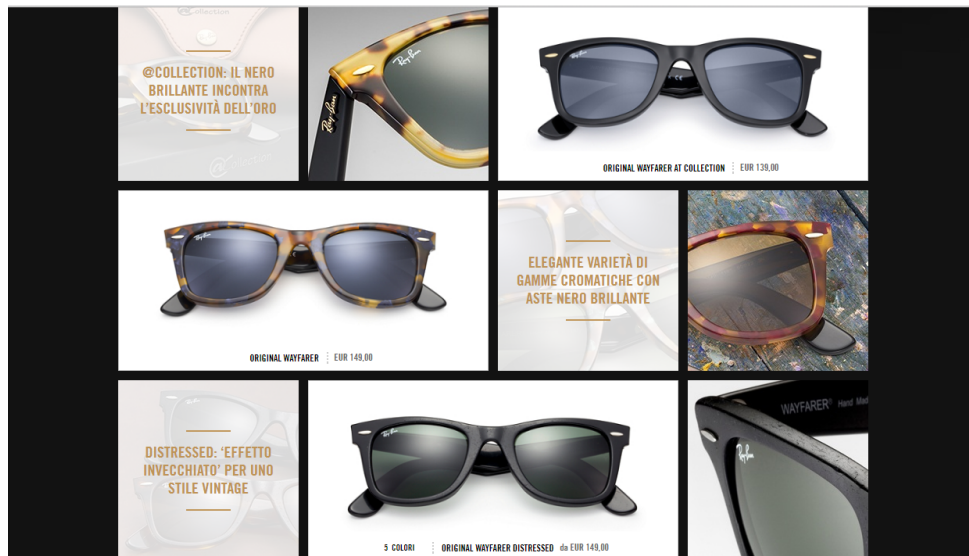


figura 1.3: Wall mood creato tramite *Live Story* per l'e-commerce di una nota marca di occhiali

1.3 Processi interni

Lo sviluppo del software a The White Dog s.r.l. segue una metodologia tipicamente Agile³.

La Metodologia Agile si riferisce ad un insieme di metodi di sviluppo software fondati su principi comuni, direttamente o indirettamente derivati dai principi del *Manifesto per lo Sviluppo Agile di Software*:

- * **Gli individui e le interazioni** più che i processi e gli strumenti;
- * **Il software funzionante** più che la documentazione esaustiva;
- * **La collaborazione col cliente** più che la negoziazione dei contratti;
- * **Rispondere al cambiamento** più che seguire un piano.

I metodi agili si contrappongono al modello a cascata e ad altri processi software tradizionali, proponendo un approccio meno strutturato e focalizzato sull'obiettivo di consegnare al cliente, in tempi brevi e frequentemente, software funzionante e di qualità. Fra le pratiche promosse dai metodi agili ci sono la formazione di team di sviluppo piccoli, cross-funzionali e auto-organizzati, lo sviluppo iterativo e incrementale, la pianificazione adattiva e il coinvolgimento diretto e continuo del cliente nel processo di sviluppo.

³<http://agilemanifesto.org/>

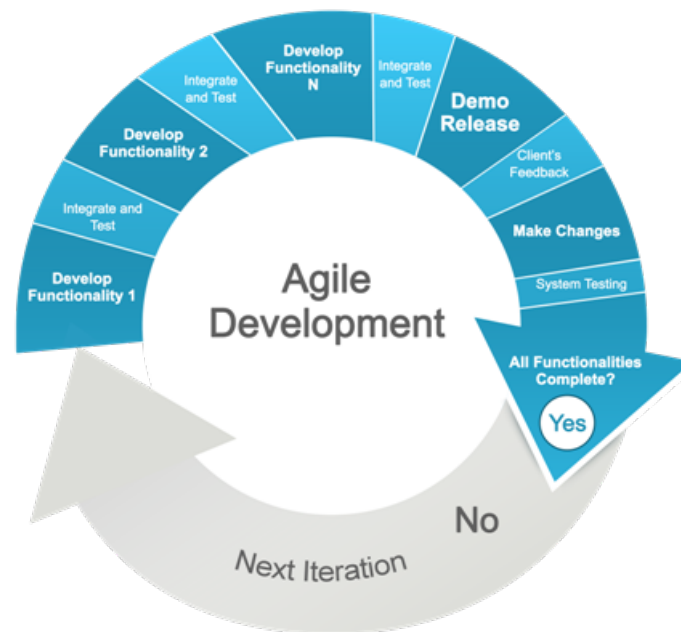


figura 1.4: Ciclo di sviluppo della Metodologia Agile

L'azienda The White Dog s.r.l. è composta da un *team* di sviluppo formato da sole tre persone, le quali possiedono competenze sia software, sia decisionali che di controllo della qualità. Questo permette all'azienda di essere molto agile sia nello sviluppo che nel rilascio del software.

Essa dà molta importanza agli individui presenti e all'interazione tra di loro, eliminando completamente la gerarchia lavorativa classica e impegnandosi molto per mantenere un ambiente lavorativo di reciproco rispetto paritario.

La collaborazione col cliente avviene in maniera assidua e giornaliera tramite riunioni nell'ufficio R&D o con trasferte. Se entrambe le soluzioni non sono attuabili, vengono optate sessioni di video conferenze. Questa pratica è agevolata dal fatto che per volontà del fondatore, The White Dog s.r.l. ha sede all'interno dello stesso stabilimento di Diana Corp., suo principale cliente.

Infine la natura dell'azienda la obbliga a rispondere al cambiamento in maniera molto veloce e repentina, dovendo pianificare così molti cicli di *refactoring*_G per lo sviluppo software.

Le tre principali metodologie di sviluppo, derivanti da quella Agile, che l'azienda adotta sono:

DevOps

Metodologia di sviluppo software che punta alla comunicazione, collaborazione e integrazione tra gli sviluppatori e addetti alle *operations*_G dell'*information technology*.

DevOps vuole rispondere all'interdipendenza tra sviluppo software e IT *operations*_G, puntando ad aiutare un'organizzazione a sviluppare in modo più rapido ed efficiente prodotti e servizi.

L'integrazione *DevOps* ha come obiettivo il rilascio del prodotto, il collaudo del software, l'evoluzione e il mantenimento in modo tale da aumentare affidabilità e sicurezza e rendere più veloci i cicli di sviluppo e rilascio.

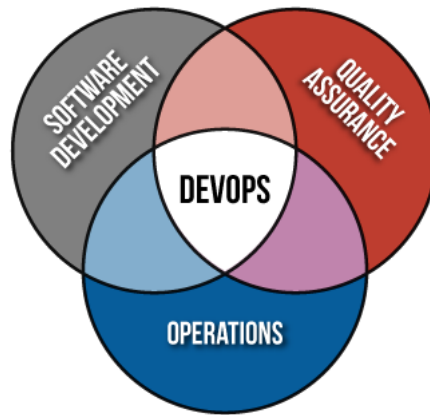


figura 1.5: Competenze necessarie alla metodologia di sviluppo DevOps

In The White Dog s.r.l. questo principio è concretizzato dal fatto che ogni membro possiede sia le competenze di sviluppo, sia amministrative che di controllo della qualità, migliorando così di molto l'efficienza e l'agilità nello sviluppo del software. Questa metodologia aiuta e migliora anche i continui rilasci che l'azienda giornalmente è costretta a fare per migliorare i servizi di Diana Corp. o per aumentarne le potenzialità con nuove tecnologie.

Extreme Programming

Metodologia di sviluppo software che enfatizza la scrittura di codice di qualità e la rapidità di risposta ai cambiamenti di requisiti. Prescrive lo sviluppo iterativo e incrementale, soprattutto in brevi cicli di sviluppo. Suggerisce inoltre l'uso sistematico di *unit testing*_G e *refactoring*_G, vietando ai programmatori di sviluppare codice non strettamente necessario. Sostiene la chiarezza e la semplicità del codice, preferisce strutture gestionali non gerarchiche e dà molta importanza alla comunicazione diretta e frequente fra sviluppatori e cliente e fra gli sviluppatori stessi.

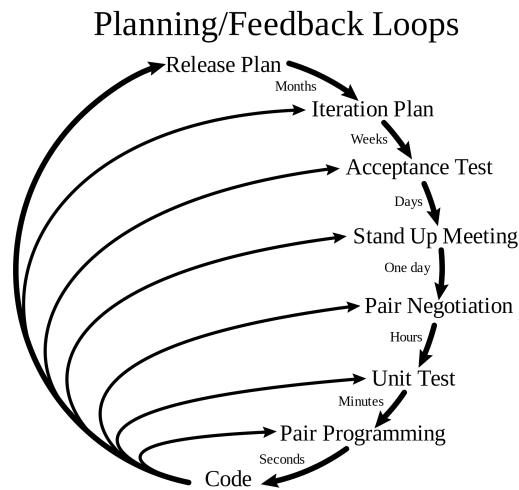


figura 1.6: Ciclo di sviluppo software Extreme Programming

Il *team* di sviluppo di The White Dog s.r.l. fa ampio utilizzo di questa metodologia, spingendo molto sulla semplicità del codice prodotto, che andrà a migliorare e ampliare i servizi *e-commerce* di Diana Corp. e Live Story e che quindi dovrà poi essere compreso dai loro sviluppatori.

Prevede continui cicli di *refactoring*_G per adattarsi alla volubilità del cliente finale e di *unit testing*_G per garantirgli codice funzionante e di qualità.

Scrum

*Framework*_G agile di sviluppo software, iterativo ed incrementale, concepito per gestire progetti e prodotti software. Esso enfatizza tutti gli aspetti di gestione di progetto legati a contesti in cui è difficile pianificare in anticipo. Vengono utilizzati meccanismi propri di un processo di controllo empirico, in cui i cicli di *feedback*, che ne costituiscono le tecniche di *management* fondamentali, risultano in opposizione alla gestione basata sul concetto tradizionale di *command-and-control*_G. Il suo approccio alla pianificazione e gestione dei progetti è quello di portare l'autorità decisionale al livello di proprietà e certezze operative.

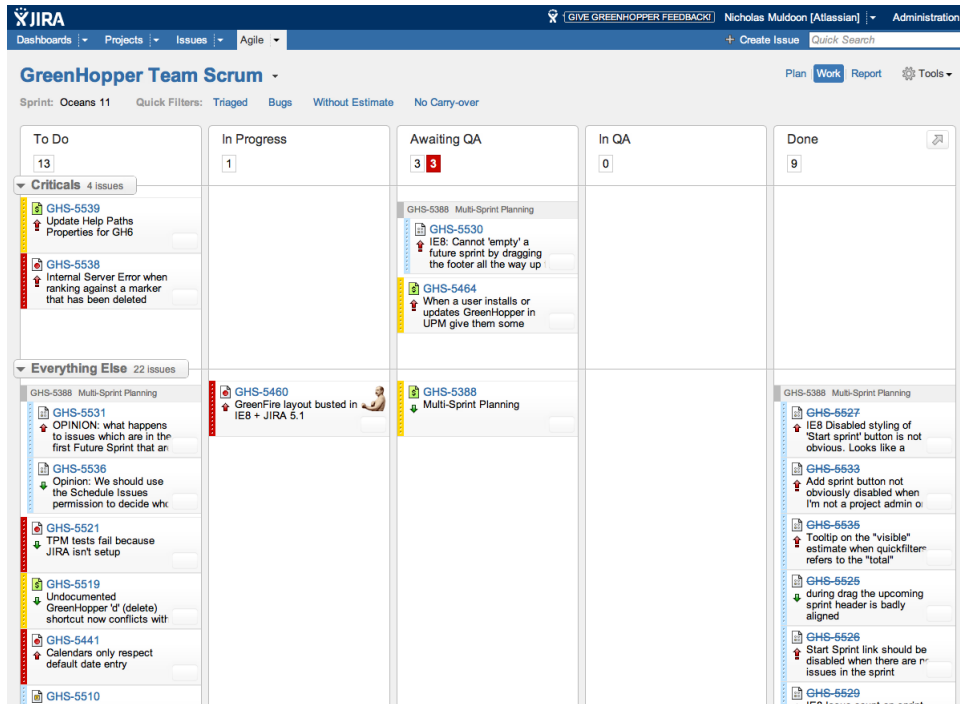


figura 1.7: Esempio di *scrum board* all'interno del software Jira

Data la forte difficoltà aziendale a pianificare in anticipo lo sviluppo software, causata dalla continua richiesta di modifiche e nuove funzionalità dei servizi *e-commerce* che ha in gestione, l'azienda fa ampio utilizzo di questa metodologia di pianificazione del lavoro utilizzando strumenti di *project management* come Wrike⁴ e Jira⁵. I due software supportano il principio *Scrum* fornendo all'utente una *Scrum board*: tavola virtuale nella quale è possibile visualizzare il flusso di lavoro suddiviso in sezioni. Ad ogni sezione sono assegnati uno o più sviluppatori che annotano l'avanzamento del lavoro, i *bug* e i problemi riscontrati.

1.4 Strumenti e tecnologie

1.4.1 Ambienti di sviluppo

Il sistema operativo adottato dall'azienda è Mac OS X installato su macchine iMac. L'ambiente di sviluppo predefinito è Eclipse⁶, ma, data la natura aziendale, varia molto spesso in base al prodotto in fase di sviluppo, che può cambiare in maniera repentina.

1.4.2 Gestione dei progetti

I due principali strumenti utilizzati da The White Dog s.r.l. per il *project management* e l'*issue tracking* sono rispettivamente Wrike e Jira.

⁴<https://www.wrike.com/it/it/>

⁵<https://www.atlassian.com/software/jira>

⁶<https://eclipse.org/>

Wrike, sviluppato dall'omonima casa, è uno strumento per la collaborazione e il *project management*. Permette ai suoi utenti di modificare progetti, classificare le attività per importanza, tenere traccia dei programmi e collaborare con altri utenti dello stesso gruppo.

Jira, prodotto dall'azienda Atlassian, è un software di *bug tracking*_G, *issue tracking*_G e *project management*. Esso permette di tenere traccia delle azioni e dei problemi degli utenti, di distribuire i compiti all'interno del *team*, discutere del lavoro in atto con una visibilità completa e migliorare le prestazioni della squadra visualizzando dati in tempo reale.

1.4.3 Versionamento

Il principale software di controllo di versione distribuito utilizzato dall'azienda è Git⁷. Git supporta lo sviluppo non lineare con diramazione e fusioni rapide e continue e comprende strumenti specifici per visualizzare e navigare una cronologia di sviluppo. Permette ad ogni sviluppatore una copia locale dell'intera cronologia di sviluppo e le modifiche vengono importate da un *repository*_G ad un altro. I *repository*_G possono essere pubblicati facilmente tramite protocolli HTTP, FTP, SSH, RSYNC o uno speciale protocollo git.

1.4.4 Strumenti di automazione

The White Dog s.r.l. fa ampio utilizzo di strumenti di automazione per facilitare il *deployment*_G del software prodotto.

Il principale strumento utilizzato dall'azienda è Docker⁸. Docker è un progetto *open-source* che automatizza il *deployment*_G delle applicazioni all'interno di *container software*_G, fornendo un'astrazione addizionale grazie alla virtualizzazione a livello di sistema operativo Linux. Docker utilizza le funzionalità di isolamento delle risorse del kernel Linux, come ad esempio *cgroups* e *namespaces* per consentire a container indipendenti di coesistere sulla stessa istanza di Linux, evitando l'installazione e la manutenzione di una macchina virtuale.

1.4.5 Tecnologie di sviluppo

Vista la varietà delle ricerche e dei prodotti sviluppati dall'azienda The White Dog s.r.l., le tecnologie di sviluppo sono sempre in continua evoluzione e cambiamento. I servizi di mantenimento e sviluppo di nuove funzionalità offerti agli *e-commerce* di Diana Corp. la portano comunque a lavorare giornalmente con le seguenti tecnologie web come: Java, JavaScript, Node.js, MongoDB, PHP, HTML e CSS.

1.5 Ricerca e innovazione

R&D rappresenta il reparto di ricerca e sviluppo dell'azienda The White Dog s.r.l.. Ha a disposizione diversi dispositivi per la ricerca come *smartphone* di ultima generazione, *smart TV*, *smartwatch* e numerosi dispositivi per lo sviluppo *AR*_G e *VR*_G

⁷<https://git-scm.com/>

⁸<https://www.docker.com/>

come *Google Glass*⁹, *Oculus Rift Development Kit 2*¹⁰, *Gear VR*¹¹, *Google Cardboard*¹² e *Leap Motion*¹³. Attraverso questi dispositivi l'azienda studia e sviluppa nuove modalità di interazione che l'utente finale può utilizzare nell'acquisto nei propri *stores* digitali.

Un esempio di progetto R&D è rappresentato proprio il mio progetto di stage che, attraverso le tecnologie *VR_G* sopracitate, mira a creare un *e-commerce* in realtà virtuale dove è possibile visualizzare l'interno di un negozio, i prodotti esposti, potendoli anche acquistare.

⁹<https://www.google.com/glass/start/>

¹⁰<https://www3.oculus.com/en-us/rift/>

¹¹<https://www3.oculus.com/en-us/gear-vr/>

¹²<https://vr.google.com/cardboard/>

¹³<https://www.leapmotion.com/>

Capitolo 2

Il quadro strategico

2.1 Strategie aziendali di stage

L'azienda The White Dog s.r.l. accoglie e offre l'attività di stage per due principali motivi:

- * **Sperimentazione su progetti innovativi:** data la forte propensione alla ricerca dell'azienda, esistono numerosi campi che essa vorrebbe esplorare ma che a causa di altri progetti più prioritari e scarsità di tempo non può studiare. Offre quindi allo studente universitario un progetto di ricerca e sviluppo su tecnologie innovative ed interessanti, non pretendendo alcun risultato da subito inseribile nel mercato. Questo permette allo studente di vivere l'esperienza dello stage in piena libertà e serenità, riuscendo così a portare un notevole valore aggiunto personale che l'azienda è ben felice di accogliere;
- * **Valutazione dello stagista:** l'azienda è in continua crescita e necessita di nuovo personale preparato e soprattutto capace di lavorare in costante sintonia col gruppo. Lo stage universitario permette all'azienda di scoprire persone che soddisfano questi due importanti requisiti per una futura assunzione.

Da parte sua The White Dog s.r.l. offre molto agli stagisti. I tutor aziendali supportano lo studente per tutto il periodo lavorativo, consigliandolo sia per quanto riguarda il piano di lavoro, sia sulle tecnologie da utilizzare sia effettuando proficue discussioni in vista della relazione finale. Allo studente viene offerto un ambiente di lavoro accogliente e strumenti aggiornati e all'avanguardia, supportandolo anche economicamente prevedendo un rimborso spese.

2.2 Il progetto di stage proposto

Il progetto propostomi nasce dalla costante volontà aziendale di ricercare nuove metodologie di interazione da proporre agli utenti dei suoi *e-commerce* in ottica *omni-channel*.

Il modello *omni-channel* si sta lentamente ma inesorabilmente affermando come principale modello di *retailing*_G a livello globale e si basa sulle seguenti caratteristiche:

- * Concezione e management unitario della distribuzione;

- * Processi basati sull'interazione, comunicazione e interdipendenza tra i *team* dedicati ai singoli canali;
- * Approccio dinamico al consumatore, che richiede un monitoraggio in tempo reale delle evoluzioni dei comportamenti di acquisto e delle risposte alle iniziative promosse;
- * Predisposizione di adeguati strumenti IT e marketing in grado di sfruttare ed assecondare il fenomeno della cross-canalità dei processi di acquisto;
- * Impatto competitivo decisivo delle scelte organizzative e di investimento nell'IT e nel marketing digitale;
- * Impiego di indicatori di prestazioni e di sistemi di *monitoring* adeguati al nuovo contesto.

Questo significa da parte delle aziende offrire un'unica *customer experience* capace di rispondere in modo adeguato alle aspettative del consumatore omnicanale che si informa sui prodotti in mobilità, li prova e li sperimenta in negozio e per poi acquistarli in loco o online. In termini pratici, questo si implementa costruendo un unico profilo del consumatore, immagazzinandovi dati sulle ricerche da lui effettuate, sulle scelte e sui dati personali immessi per poi sfruttarli in ogni tipologia di *store*, sia digitale che fisico.

La realtà virtuale si pone nella strategia *omni-channel* come punto di connessione tra *store* digitale e fisico permettendo all'utente di esplorare il negozio nella sua totalità e di osservarne i prodotti esposti da più angolazioni o indossati da modelli o manichini nel caso dei capi di abbigliamento, il tutto in un alto livello di definizione. Nasce così l'idea di un *e-commerce VR*, progetto in grado di colmare, in parte, quel divario che da sempre ha distanziato *store* virtuale e negozio fisico.

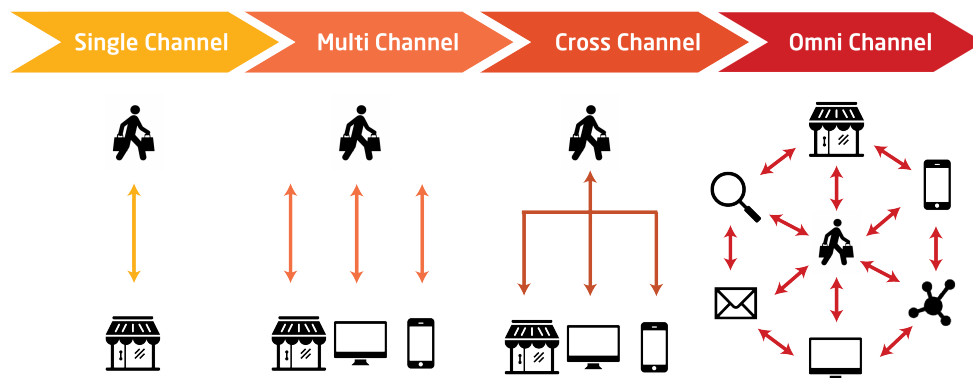


figura 2.1: Schema rappresentativo della differenze tra *single-channel*, *multi-channel*, *cross-channel* e *omni-channel*

L'obiettivo di stage è, dunque, un'esplorazione tecnologica nel campo della *virtual reality*. Il progetto mira ad arrivare ad un prototipo di *virtual showroom* dove poter esplorare ed interagire con i prodotti e permetterne l'acquisto. Il progetto era stato inizialmente diviso in due parti, per due differenti studenti:

- * La prima parte riguardava la progettazione e realizzazione del movimento in uno spazio 3D ed interazione con gli oggetti;
- * La seconda parte trattava invece la progettazione e realizzazione in un'interfaccia di presentazione del prodotto, con integrazione al processo di acquisto mediante l'uso di sistemi *cloud* esterni.

Purtroppo, nessun altro studente oltre a me ha aderito al progetto, dunque si è dovuto rivisitare l'obiettivo di stage. Dopo una riunione effettuata prima dell'inizio dello stage con il mio tutor aziendale, abbiamo deciso di mantenere intatti tutti gli obiettivi di ricerca, abbassando il livello qualitativo richiesto. Questo perché lo scopo ultimo di questo stage non era sviluppare un'applicazione o un servizio immediatamente vendibile, ma di studiare le potenzialità e i limiti di questa nuova tecnologia. Questa volontà è stata dettata anche dal fatto che il *team* aziendale inizialmente non aveva alcuna certezza che la tecnologia *VR_G* fosse applicabile al mondo *e-commerce*.

2.2.1 Piano di lavoro proposto

Piano temporale

In accordo col tutor aziendale, la durata massima dello stage è stata fissata a 320 ore, divise in 8 settimane lavorative di 5 giorni, 8 ore al giorno.

Il piano lavorativo è stato dunque pianificato per settimana nel seguente modo:

- * **Settimana 1:** settimana dedicata completamente alla ricerca, per colmare il *deficit* culturale personale e aziendale sulle tecnologie *VR_G*. Le attività principali previste sono: analisi dei requisiti funzionali del sistema da sviluppare e studio delle tecnologie e linguaggi disponibili riguardanti la realtà virtuale;
- * **Settimana 2:** in base ai risultati ottenuti nella prima settimana, viene richiesta una scelta dell'hardware da utilizzare e un *framework_G* di sviluppo, testandoli con un primo prototipo di scena 3D;
- * **Settimana 3:** previste attività di affinamento della scena 3D, progettazione e sviluppo degli oggetti e loro comportamento nello spazio 3D. Viene creato così un primo prototipo di *user interaction*;
- * **Settimana 4:** previste attività di progettazione e sviluppo integrazione tra sistema *VR_G* e *e-commerce*. Progettazione di *user interaction* per la fruizione dei contenuti provenienti dall'*e-commerce*;
- * **Settimana 5:** settimana dedicata all'approfondimento di *user interaction* e del comportamento degli oggetti nell'ambiente virtuale;
- * **Settimana 6:** previste attività di studio e prototipazione del possibile processo d'acquisto all'interno dell'ambiente virtuale;
- * **Settimana 7:** la settima settimana rappresenta una *milestone* importante per il progetto: conclusione del prototipo e relativa documentazione, raggiungendo così gli obiettivi minimi;
- * **Settimana 8:** l'ultima settimana viene dedicata completamente allo studio del modello emergente *omni-channel_G* e come la realtà virtuale possa estendere questo modello. Vengono così raggiunti gli obiettivi massimi.

Piano metodologico

Assieme al tutor aziendale, abbiamo fin da subito concordato la mia presenza durante l'orario d'ufficio, permettendo così un'interazione intensa e costante. Il lavoro di ricerca e sviluppo che ho effettuato è stato totalmente autonomo, con giornaliere interazioni con il personale solo per raccogliere e analizzare la documentazione, requisiti e *feedback* sull'andamento del progetto.

Le revisioni di progetto sono avvenute secondo la seguente metodologia:

- * Riunione breve di 15 minuti ogni mattina;
- * Riunione di 1 ora alla fine di ogni settimana come analisi retrospettiva.

Alle revisioni, oltre a me, hanno partecipato:

- * Valentino Baraldo, *cloud engineer* e tutor aziendale. Oltre a svolgere il compito di tutor aziendale, mi ha supportato sulla progettazione architettuale del progetto e sull'utilizzo del servizio *API Gateway* di *Amazon Web Services*¹;
- * Francesco Paggin, *front-end developer*. Ha supervisionato il mio lavoro grafico nell'ambiente di sviluppo *Unity*².

Piano tecnologico

Inizialmente lo *stack* tecnologico propostomi riguardava solamente l'hardware che l'azienda aveva acquistato per questo progetto, senza alcun vincolo software. I dispositivi che permettevano la sperimentazione *VR_G* erano:

- * **Oculus Rift Development kit 2:** visore per la realtà virtuale per uso desktop. Possiede uno schermo Samsung OLED 2160x1200 pixel (1080x1200 per occhio), con un *refresh rate* a 90 Hz e un ampio angolo di visione a 110 gradi. Dotato di accelerometro, giroscopio, magnetometro e *tracking* posizionale a 360 gradi. Viene accoppiato ad una telecamera infrarossi per il rilevamento di profondità, assieme a 40 emettitori infrarossi all'interno dell'*headset*. Monta due lenti in alta definizione possedendo 6 gradi di libertà di rotazione;
- * **Samsung Gear VR:** visore per la realtà virtuale per uso *mobile*. Possiede: accelerometro, giroscopio e sensore di prossimità, permettendo un campo visivo di 96 gradi. Il visore incorpora inoltre un'interfaccia utente fisica: *touch pad*, tasto indietro e tasto per il volume. Necessita l'inserimento di uno *smartphone* Samsung a partire dalla versione *Galaxy S6*;
- * **Google Cardboard:** con il termine *Google Cardboard* non si intende specificare un particolare visore per la realtà virtuale prodotto fisicamente da *Google*, ma un insieme di linee guida suggerite da questa per costruire un dispositivo a basso costo per l'uso *mobile*. In azienda erano presenti due visori che implementavano tali linee guida: *Unofficial Cardboard* e *Tera VR Box*;
- * **Leap Motion:** piccola periferica USB progettata per essere posta su una scrivania reale rivolta verso l'alto. Usando 2 telecamere e 3 LED infrarossi essa osserva un'area approssimativamente a forma di semisfera di circa un metro. E' progettata per identificare dita o oggetti simili come una penna, con una precisione di 0,01 mm.

¹<https://aws.amazon.com/it/>

²<https://unity3d.com/>



figura 2.2: Samsung Gear VR

Dopo un periodo di ricerca e *testing* su queste tecnologie, abbiamo deciso di intraprendere la strada *mobile*, a discapito di quella *desktop*. Questa decisione è stata dettata principalmente da due fattori:

- * **Requisiti hardware elevati:** per poter offrire un'esperienza fluida e piacevole, *Oculus Rift Development kit 2* abbisogna di un PC dall'hardware elevato, non accessibile all'utenza media:
 - **GPU:** NVIDIA GTX 970 / AMD R9 290;
 - **CPU:** Intel i5-4590;
 - **RAM:** 8GB;
 - **Video output:** HDMI 1.3;
 - **USB Ports:** 3 porte 3.0 più una porta 2.0;
 - **OS:** Windows 7 SP1 64 bit o superiore.
- * **Obiettivi aziendali:** anche se fin da subito mi era stato chiarito che non veniva preteso alcun prodotto finale utilizzabile, l'azienda sperava però di riuscire con questo stage di avere un primo prototipo di *virtual showroom* da poter mostrare alle fiere tecnologiche alle quali partecipa. In quest'ottica, l'utilizzo di *Oculus Rift Development kit 2* sarebbe risultato troppo scomodo sia per il trasporto e l'installazione, che per l'utilizzatore finale.

Si è deciso dunque di sviluppare sia per *Samsung Gear VR* che per *Google Cardboard*, entrambi dispositivi *mobile* a costo contenuto.

Lo stack software che si è andato a formare poi, presa questa decisione, ha avuto un tempo decisionale assai breve. Dopo alcune ricerche da me effettuate, ho potuto riscontrare come *Unity* fosse l'unico *framework*_G per lo sviluppo *VR*_G gratuito e ricco di documentazione, cosa estremamente fondamentale data la mia iniziale totale ignoranza sull'argomento.



figura 2.3: Logo Unity

Unity è uno strumento di *authoring*_G integrato e multiplatforma per la creazione di videogiochi 3D o altri contenuti interattivi, quali visualizzazioni architettoniche o animazioni 3D in tempo reale. Esso, tramite gli *SDK*_G forniti da Samsung, per il dispositivo Gear VR, da Google, per il dispositivo Cardboard, e da Andriod, per l'interfacciamento con lo smartphone, permette lo sviluppo di applicazioni *VR*_G fornendo un interfaccia per modellare "a mano" gli oggetti e i linguaggi JavaScript e C# per sviluppare il loro comportamento.

Il linguaggio di *scripting* da me scelto per modellare il comportamento degli oggetti nello spazio tridimensionale è C# poiché è risultato essere il linguaggio più utilizzato nella *community Unity*.

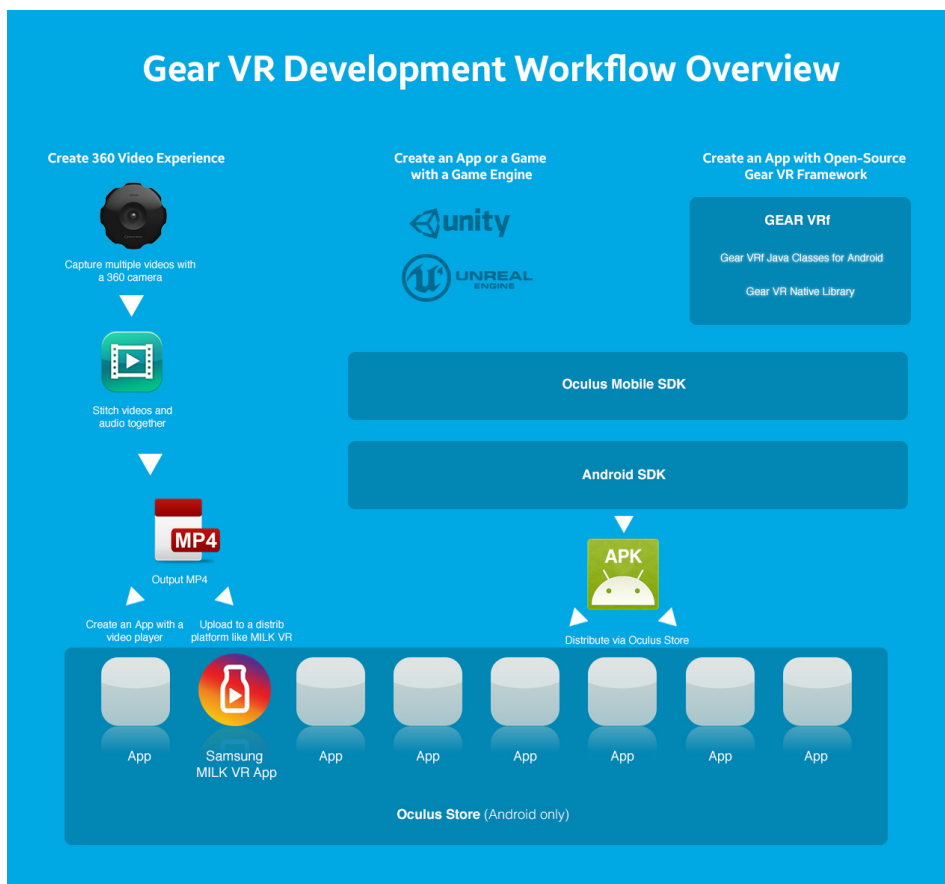


figura 2.4: Stack tecnologico di sviluppo per Samsung Gear VR

2.2.2 Obiettivi aziendali

Nel *Piano di Lavoro* presentatomi, l'azienda espone gli obiettivi minimi e massimi che si aspetta di veder raggiunti alla fine delle 320 ore di stage:

*** Obiettivi minimi:**

1. Studio delle tecnologie disponibili in ambito VR_G e stesura di un documento riassuntivo che offra un *overview* dello stato attuale della realtà aumentata;
2. Progettazione e sviluppo di un ambiente virtuale con: una scena e oggetti definiti, un comportamento associato agli oggetti, un prototipo di *user interaction* e scambio di informazioni di base con un sistema di *e-commerce*.

*** Obiettivi massimi:**

1. Studio e prototipazione di diversi modelli di *user interaction* con l'ambiente e con gli oggetti finalizzati alla presentazione di un bene vendibile;
2. Studio e implementazione di possibili nuovi processi di acquisto in ambito VR_G .

2.2.3 Obiettivi personali

Sono venuto a conoscenza di questo progetto durante l'evento di Stage IT 2016, organizzato da Confindustria Padova in collaborazione con l'Università di Padova e Venezia. Mi ha fin da subito colpito e appassionato per le tecnologie che mi avrebbe permesso di studiare, come ad esempio *Unity*. La computer grafica è da sempre un mio personale interesse e la realtà aumentata è un ambito per me molto affascinante e ricco di opportunità.



figura 2.5: Logo dell'evento Stage IT 2016

Le tecnologie proposte dall'azienda purtroppo non rientrano nel percorso di studi, dunque gli obiettivi formativi personali che mi sono posto riguardano lo studio e la sperimentazione delle tecnologie, senza pretendere di arrivare ad un risultato non prototipale:

* **Obiettivi minimi:**

1. Conoscenza ad alto livello delle tecnologie (hardware e software) attualmente disponibili nel mercato atte a creare ambienti virtuali;
2. Conoscenza ad alto livello dei concetti principali di *e-commerce* e relative tecnologie di riferimento usate per la vendita online.

* **Obiettivi massimi:**

1. Capacità di identificare, progettare e sviluppare ambienti virtuali, selezionando le tecnologie attualmente disponibili più appropriate per il caso d'uso;
2. Presa di coscienza dei concetti multi-channel e omni-channel e come le nuove modalità di vendita si integrino con questi modelli emergenti.

Capitolo 3

Il progetto di e-commerce VR

In questo capitolo andrò a trattare nel dettaglio tutte le fasi dello stage che hanno portato allo sviluppo del progetto.

3.1 Pianificazione del lavoro

In questa sezione tratterò della pianificazione del lavoro effettuata assieme al mio tutor, delle fasi che l'hanno caratterizzata e del ciclo di vita adottato.

3.2 Ricerca e sperimentazione

In questa sezione descriverò la fase di ricerca e sperimentazione delle tecnologie utilizzate, inizialmente a me sconosciute. Ho deciso di dedicare una sezione a questa fase perché ha avuto una rilevante importanza all'interno del mio stage e rappresenta uno dei principali obiettivi aziendali.

3.3 Tecnologie adottate

In questa sezione descriverò come le ricerche e le sperimentazioni effettuate mi hanno portato a scegliere un particolare stack tecnologico.

3.4 Analisi dei requisiti

All'interno di questa sezione tratterò dell'attività di analisi dei requisiti che il team ha effettuato prima della progettazione e dello sviluppo del progetto.

3.4.1 Caratteristiche degli utenti

In questa sottosezione descriverò le tipologie di utenti previsti per l'applicazione.

3.4.2 Casi d'uso

In questa sottosezione elencherò tutti i casi d'uso previsti per l'applicazione.

3.4.3 Requisiti

In questa sottosezione elencherò tutti i requisiti estratti che l'applicazione soddisfa.

3.5 Progettazione

In questa sezione andrò a descrivere le più importanti fasi di progettazione.

3.5.1 Portabilità dell'applicazione

In questa sottosezione tratterò di come la progettazione del software sia stata ampiamente influenzata dalla volontà di portabilità dell'applicazione su tutti i dispositivi VR Android.

3.5.2 Usabilità dell'applicazione

In questa sottosezione descriverò gli studi effettuati riguardo l'usabilità dell'applicazione VR e delle scelte di progettazione che hanno portato tali studi.

3.5.3 Costruzione della scena 3D

In questa sezione andrò a descrivere le fasi di progettazione della scena 3D presente nell'applicazione e visibile tramite dispositivo VR.

3.5.4 Interazione con gli oggetti all'interno della scena

All'interno di questa sottosezione parlerò della progettazione riguardante le modalità di interazione tra il visore VR e gli oggetti presenti all'interno della scena.

3.5.5 Progettazione e integrazione con AWS API Gateway

All'interno di questa sezione tratterò della progettazione riguardante l'API Mock creata tramite AWS API Gateway e della sua integrazione con l'applicazione.

3.6 Sviluppo

In questa sezione andrò a descrivere in dettaglio lo sviluppo delle più significative e peculiari funzionalità dell'applicazione.

3.6.1 Sviluppo degli oggetti interattivi

In questa sottosezione descriverò come si costruiscono degli oggetti interattivi in Unity per i dispositivi VR.

3.6.2 Creazione a runtime di oggetti interattivi

In questa sottosezione tratterò della creazione a runtime di oggetti interattivi in Unity.

3.6.3 Dati persistenti attraverso le scene

In questa sezione spiegherò come si costruiscono oggetti persistenti che vivono attraverso le scene.

3.6.4 Unity e il protocollo HTTP

In questa sottosezione parlerò di come Unity si integri con il protocollo HTTP.

3.6.5 Creazione e parsing di oggetti JSON in Unity

In questa sottosezione parlerò di come si creino e si manipolino oggetti JSON in Unity.

3.7 Verifica e validazione

All'interno di questa sezione parlerò della fase di verifica e validazione effettuata per questo progetto.

Capitolo 4

Analisi retrospettiva

In questo capitolo analizzerò i risultati ottenuti confrontandoli con gli obiettivi prefissati, le conoscenze acquisite e quali tra queste ritengo debbano essere integrate nel corso di laurea.

4.1 Bilancio dei risultati rispetto agli obiettivi prefissati

In questa sezione analizzerò i risultati ottenuti e li confronterò con gli obiettivi che l'azienda si era prefissata di ottenere.

4.2 Bilancio formativo

In questa sezione analizzerò le conoscenze, le abilità e le competenze apprese durante l'attività di stage.

4.3 Analisi critica del rapporto formativo tra stage e corso di laurea

In questa sezione discuterò quali conoscenze, apprese durante lo stage, ritengo debbano essere integrate nel corso di laurea.

4.4 Valutazioni personali

In questa sezione effettuerò delle valutazioni personali riguardo al progetto e allo stage.

Glossario

A

- * **Authoring:** gli applicativi d'autore sono quei software verticali che consentono la realizzazione di una comunicazione multimediale, articolata e riproducibile su personal computer. L'intento è quello di poter produrre e veicolare contenuti (immagini statiche, animazioni grafiche, filmati video, commenti sonori, effetti audio e altro) su supporti come CD-ROM, DVD, via web, ma anche attraverso un circuito, chiuso o aperto, di display distribuiti e connessi tra loro in rete.
- * **Augment reality:** arricchimento della percezione sensoriale umana mediante informazioni, in genere manipolate e convogliate elettronicamente, che non sarebbero percepibili con i cinque sensi.

B

- * **Bug tracking:** applicativo software usato generalmente dai programmatori per tenere traccia delle segnalazioni di bug all'interno dei software, in modo che tali errori siano mantenuti sotto controllo, con una descrizione della riproducibilità e dei dettagli ad essi correlati.

C

- * **Command-and-control:** principio di *management* dove si afferma il mantenimento dell'autorità in un processo decisionale distribuito.
- * **Content management system:** è uno strumento software, installato su un server web, il cui compito è facilitare la gestione dei contenuti di siti web, svincolando il webmaster da conoscenze tecniche specifiche di programmazione web.
- * **Container software:** soluzione al problema di come ottenere il software eseguibile in modo affidabile quando viene spostato da un ambiente informatico all'altro. Sono costituiti da un intero ambiente di runtime: un'applicazione, oltre a tutte le sue dipendenze, librerie e altri file binari e file di configurazione necessari per eseguirlo, impacchettati in un unico pacchetto.

D

- * **Deployment:** consegna o rilascio al cliente, con relativa installazione e messa in funzione o esercizio, di una applicazione o di un sistema software tipicamente all'interno di un sistema informatico aziendale.

E**F**

- * **Framework:** architettura logica di supporto (spesso un'implementazione logica di un *design pattern*) su cui un software può essere progettato e realizzato.

G**H****I**

- * **Issue tracking:** pacchetto software che gestisce e mantiene liste di problemi in maniera organizzata.

J**K****L****M****N****O**

- * **Operations:** funzioni di un'impresa coinvolte nella messa a disposizione per il cliente di un determinato prodotto o servizio.

P**Q****R**

- * **Retailing:** o vendita al dettaglio, costituisce l'ultimo anello della catena di distribuzione. Il venditore al dettaglio (negozi, supermercato, eccetera) acquista

quantità, relativamente elevate, di merce dal produttore o da un grossista e rivende quantità più contenute ai consumatori per ottenere un profitto.

- * **Refactoring:** tecnica strutturata per modificare la struttura interna di porzioni di codice senza modificarne il comportamento esterno, applicata per migliorare alcune caratteristiche non funzionanti del software.
- * **Repository:** è un ambiente di un sistema informativo, in cui vengono gestiti i metadati, attraverso tabelle relazionali; l'insieme di tabelle, regole e motori di calcolo tramite cui si gestiscono i metadati prende il nome di metabase.

S

- * **Software development kit:** insieme di strumenti per lo sviluppo e la documentazione di software.
- * **Social management system:** software che permette la gestione dei propri social network, collezionando contenuti presenti in essi o interagendovi automaticamente effettuando operazioni desiderate.

T

- * **Testing:** il *software testing* è un'attività di investigazione condotta per fornire alle parti interessate informazioni sulla qualità del prodotto o del servizio in prova.

U

- * **Unit testing:** per test di unità si intende l'attività di *testing* di singole unità software. Per unità si intende il minimo componente di un programma dotato di funzionamento autonomo.

V

- * **Virtual reality:** realtà simulata attraverso dispositivi elettronici, come visori, cuffie e sensori di movimento.

W

X

Y

Z

Bibliografia