

Capitolo 1

Introduzione

1.1 Descrizione del problema

Uno dei vari problemi che un'azienda deve affrontare, dal momento in cui decide di orientare le sue scelte strategiche del settore ICT verso l'e-business¹, è quello di riuscire a scegliere le applicazioni più adeguate alle proprie esigenze (valutando contenuti, prestazioni e costi), ma soprattutto di essere in grado di fornire all'utente quello che cerca e veramente desidera. È infatti ormai concetto assodato che la *Customer satisfaction* (soddisfazione dell'utente) sia fondamentale per il successo di un'attività, in genere, ma in particolare *online*. La differenza infatti con l'*offline* non sta certo nel fatto che i Clienti siano meno importanti o esigenti, cambiano invece gli strumenti a disposizione e le aspettative in termini di efficacia (fare la cosa giusta), tempo di risposta ed affidabilità (efficienza) dei servizi fruiti.

Spesso ci si trova di fronte ad una situazione contrastante: il business evolve, con un modello² che, oltretutto, cambia anche velocemente, amplificando la separazione con la struttura che tende a rappresentarlo, cioè il sistema informativo, un'entità dotata di un'inertza naturale. Ridurre questo "gap" è una delle aree di intervento più recenti e ambite nell'ICT delle organizzazioni medio/grandi. Tale attività di miglioramento deve passare attraverso modifiche e trasformazioni sia dei processi aziendali (*Business Process Reengineering*) dell'organizzazione ICT (adeguandosi a Standard e Best Practices del settore), sia delle modalità utilizzate per produrre e distribuire le applicazioni software.

L'insieme di attività che si propongono di garantire l'allineamento completo dell'organizzazione ICT con le priorità strategiche del business fanno parte dell'area di **BTO (Business Technology Optimization)**. La BTO è composta da soluzioni di

¹ **Electronic Business**, indica le modalità digitali (tramite soluzioni ICT) con le quali vengono svolte le attività dei processi aziendali. Tipicamente è riferito a soluzioni che utilizzano la tecnologia Internet.

² Un **Modello di Business** è dato dall'insieme di struttura e azioni attraverso cui una organizzazione opera all'interno del mercato, ovvero il modo in cui essa genera reddito e trae guadagno.

Application Delivery, Application Maintenance e IT Governance, che traducono in realtà una serie di metodi e tecniche studiate, da una parte per ottimizzare gli investimenti effettuati da un'azienda nell'ICT e per sostenere quelle organizzazioni che intendono sviluppare nuovi progetti, dall'altra per assicurare la massima reattività verso gli utenti finali.

Uno dei parametri, di notevole importanza per la valutazione del successo dell'ICT è appunto la capacità di distribuire applicazioni affidabili ad alte prestazioni che soddisfino sia le esigenze e i costi dell'azienda fornitrice, sia le aspettative del Cliente. È necessario dunque che l'ICT disponga di tutte le informazioni necessarie per gestire l'intero portafoglio di applicazioni aziendali e le infrastrutture in modo proattivo, durante l'intero ciclo di vita del software e in perfetta linea con gli obiettivi di business.

La Qualità di un'applicazione Web (riferendosi all'e-business, d'ora in avanti il contesto delle applicazioni considerate è sempre quello Web-based) dipende non solo dal contenuto e dalle funzionalità applicative, ma anche dal modo in cui le informazioni sono presentate all'utente e dalle prestazioni che egli percepisce nell'utilizzare l'applicazione stessa. Diviene perciò importante per entrambi le parti coinvolte, Azienda e Cliente, che l'ICT disponga di processi e strumenti idonei alla corretta e puntuale valutazione delle applicazioni utilizzate per erogare il proprio servizio. Tali applicazioni costituiscono infatti un "primo biglietto da visita", oltre ad essere il contatto iniziale con i Clienti. Per l'utente infatti non hanno, giustamente, grande importanza le più svariate questioni tecniche che possono sottostare al funzionamento più o meno efficace di un'applicazione; ciò che essi cercano è **efficienza** nelle operazioni richieste e **disponibilità** immediata, due esigenze strettamente dipendenti una dall'altra. Se infatti un'applicazione (supponiamo l'invio di un ordine oppure un pagamento online) porta a buoni risultati dal punto di vista della correttezza (giunge a buon fine) ma il tempo di risposta (efficienza) per completare l'operazione è troppo elevato, il Cliente non sarà del tutto soddisfatto.

Per questo motivo risulta necessario apportare dei miglioramenti all'intero ciclo di vita del software, che esiste, più o meno articolato, in tutte le realtà organizzative, attraverso l'integrazione con nuovi **processi** (fasi e attività), **metodi** (linee guida, Standard, Best Practices) e **ruoli** che aiutano a raggiungere la Qualità voluta sia dal Cliente che dall'Azienda. I processi e i metodi da adottare per ottenere servizi di Qualità (conformi ai requisiti espliciti e impliciti) sono molteplici e riguardano l'insieme di attività che consentono ad un prodotto software di possedere attributi di: funzionalità, affidabilità, efficienza e usabilità. Innanzitutto bisogna essere sensibili ai concetti e ai principi di Qualità relativi al prodotto software e al processo per poterlo realizzare. Solo a partire da questa visione è possibile comprendere appieno l'importanza che hanno, lungo il ciclo di vita del software, le fasi riguardanti la gestione dei requisiti e il Software Testing, nonché i processi trasversali di Quality Assurance.

Questa tesi vuole offrire un contributo alla Qualità del Software, con la ricerca di metodi, tecniche e Best Practices da applicare per la misurazione e gestione delle prestazioni dei sistemi software, delineando una metodologia apposita:

Application Performance Management (APM). La metodologia APM definisce quali processi, strumenti e tecniche sono da utilizzare nella fase di rilascio o *Delivery* (**Performance Testing**) e nella fase di Gestione Operativa o *Operations* (**End User Monitoring**).

1.2 Obiettivi

Application Performance Management (APM) è una metodologia che nasce con l'obiettivo di migliorare sistematicamente ogni livello di un sistema informatico complesso, per garantire requisiti qualitativi in termini di prestazione. Questo obiettivo cardine, si raggiunge mediante un processo **ciclico** che prevede di:

- 1) **stimare** in fase di progettazione il fabbisogno di risorse tecnologiche in funzione dei requisiti di business e di sistema attraverso la creazione di modelli matematici e/o di simulazione (**Capacity Planning**);
- 2) **validare** nella fase di **Testing** i livelli di servizio attesi e apportare le ottimizzazioni necessarie (**Tuning**);
- 3) **monitorare** continuamente e in tempo reale, durante la gestione operativa, la disponibilità dei servizi applicativi e le prestazioni così come vengono percepite dall'utente finale (**end-to-end**).

La metodologia definita in questa tesi si focalizza sull'impiego di **tecniche di misurazione** per le fasi di testing e manutenzione, proprie del ciclo di vita del software. È costituita da due processi: Performance Testing (PT) ed End User Monitoring (EUM), che hanno il compito di quantificare le prestazioni mediante metodi, strumentazione adeguata, modelli di caratterizzazione del carico applicativo e tecniche di misurazione ed analisi, che differiscono per tipologie, modalità e ambito con cui vengono applicate.

Il **Performance Testing** (o processo di collaudo prestazionale) è un'attività finalizzata a verificare, attraverso tecniche di test di carico, se e in che misura, un'applicazione sia in grado di erogare un livello di servizio predeterminato dall'utente (e tradotto in specifiche di requisiti dell'architettura hardware/software), prima del rilascio in produzione. Per quanto le applicazioni siano testate, il mondo di produzione è diverso, pieno di rischi per la continuità del servizio da erogare. Il processo di **End User Monitoring** consente, in fase di esercizio, di conoscere in ogni istante lo "stato di salute" dei propri sistemi ICT e ottenere valutazioni oggettive sui tempi richiesti ai Clienti. Attivare questo processo aiuta a garantire servizi continuativi e qualitativamente validi ed intervenire in tempo in caso di guasti (errori o *failure*), grazie alla segnalazione di situazioni critiche o di allarme, fornita da opportuni meccanismi.

Per raggiungere gli **scopi** prefissati risulta necessario:

- Analizzare il processo di produzione del software in ottica di Qualità, identificando, a partire dai lavori di ricerca e dagli standard di Software &

Quality Engineering, una terminologia propria del processo software e le metodologie di sviluppo esistenti.

- Classificare e definire i requisiti non funzionali o QoS prestazionali, riferendosi specificatamente alle applicazioni Web ed evidenziare l'importanza di annotarli correttamente e correlarli ai requisiti funzionali, durante le prime fasi del ciclo di vita. Inoltre è opportuno gestire le possibili modifiche correttive/evolutive future dei requisiti stessi. La gestione dei requisiti non funzionali, in generale, è spesso gestita con priorità bassissima o addirittura del tutto tralasciata, anche se trascurandola si può arrivare a sviluppare un prodotto finale insoddisfacente o addirittura inutile.
- Illustrare il Software Testing secondo due visioni: come insieme di tecniche disponibili e come processo completo (quali momenti lo caratterizzano, il ciclo di vita seguito per ognuno di essi e la documentazione necessaria), per far poi proprio questo criterio per la definizione del Performance Testing.
- Per descrivere i processi cardine della metodologia, occorre definire: la terminologia specifica per le attività di misurazione delle prestazioni, delle tecniche di test mirate e la strumentazione necessaria per poterle effettivamente attuare e poter dunque automatizzare parte del processo di testing e di monitoring.
- Un *Modello di WorkLoad* per la caratterizzazione e descrizione (individuando linguaggi opportuni) del comportamento degli utenti nei confronti dell'applicazione, allo scopo di riprodurre il carico applicativo previsto il più fedelmente possibile e ottenere dunque risultati accurati dai test prestazionali.
- Verificare sperimentalmente, ove possibile, i risultati ottenuti per convalidare o meno quanto formulato.

1.3 Approccio adottato

Per la definizione della **metodologia APM**, il presente elaborato si propone dapprima di **analizzare il contesto** e lo stato dell'arte esistente, per comprendere quali elementi e conoscenze risultino indispensabili per potersi dedicare ad affrontare suddette tematiche. Definita la metodologia, o meglio ancora effettuati in contemporanea i passi necessari alla sua definizione, occorre **verificare sperimentalmente** l'effettiva applicabilità di quanto è stato delineato.

È evidente che per la fase di analisi e ricerca di queste tematiche, possono essere sufficienti gli aspetti prettamente teorici (raccolti nei libri, negli articoli e nei siti Web specifici del settore), mentre per gli stadi di definizione e verifica, contribuisce in modo decisivo il contesto di una realtà industriale. L'inserimento in un'azienda che si occupa dei suddetti concetti, o che ha l'esigenza di metterli in atto, permette di apprendere direttamente le effettive necessità, di proporre e

sperimentare le idee formulate, ma soprattutto di disporre di quella strumentazione all'avanguardia per poterle concretamente attuare. Per di più, l'inserimento in strutture organizzative ben consolidate, offre la possibilità di confrontarsi con persone esperte (*mentor*) che concorrono all'attuazione delle attività di BTO.

Combinare quanto si studia con applicazioni concrete e interessanti della realtà industriale, ha rappresentato per me uno degli aspetti cruciali che mi sono continuamente preposta come obiettivo lungo il mio percorso formativo. Per l'appunto, la metodologia APM è stata formulata durante un periodo di stage (da Marzo ad Agosto 2006) presso un Cliente dell'azienda **AIVE S.P.A.**, società per cui ho svolto lo stage. L'azienda si occupa di consulenza, progettazione e sviluppo software per organizzazioni di medie/grandi dimensioni con sede principale a Marcon (VE) e filiali dislocate in diversi punti del territorio nazionale. Da Dicembre 2005, un gruppo di consulenti Aive si occupa presso **GENERALI SERVIZI INFORMATICI S.P.A (GSI S.P.A)** della progettazione, esecuzione di test di carico e delle attività di monitoring delle applicazioni informatiche sviluppate e contribuisce inoltre alla definizione di una metodologia di lavoro per l'azienda stessa. All'interno di questo gruppo di lavoro si è collocata la mia esperienza di stage.

GSI S.p.A, con sede a Mogliano Veneto (TV), è una società che offre servizi per la gestione dell'infrastruttura ICT (Information & Communication Technology) alle aziende del Gruppo Generali, una delle più importanti realtà assicurative e finanziarie internazionali, che ha come Azienda capofila **Assicurazioni Generali S.p.A.** Uno dei progetti correnti di questa società, è l'industrializzazione del testing funzionale e dei concetti di **Performance Management**, attraverso la definizione di processi e l'impiego di strumentazione di **Mercury Interactive**, società leader globale nel settore del software e dei servizi di BTO, per la gestione del testing funzionale, testing delle prestazioni, Monitoring e IT Governance.

Come sostiene l'amministratore delegato di Mercury Italia, in uno dei suoi interventi riportati nella rivista *Computer Business Review Italy* (Novembre 2005), il mercato di riferimento è segmentato secondo diversi livelli di maturità nei confronti dell'approccio alla verifica operativa e prestazionale delle applicazioni. Si parte dall'assoluta inazione, per giungere, attraverso test sulle singole applicazioni e test sistemistici, a costruire dei veri e propri centri di eccellenza dove consolidare tutti i processi migliori e raggruppare le persone più competenti in merito. Questa osservazione evidenzia gli ingenti costi derivanti dall'adozione di queste attività. Le spese sono necessarie in modo particolare per poter disporre di personale e strumentazione adeguati, sebbene sia chiaro che più che di un costo si tratta di un investimento, che genera ritorni molto significativi, dato l'impatto positivo e l'effetto amplificante che la Qualità ha sul business. Infatti, in Italia, attualmente solo poche strutture (es. Banche, Assicurazioni e aziende operanti nel settore delle telecomunicazioni e nei servizi di publishing) possono permettersi di affrontare con rigore queste discipline.

Questa tesi, oltre ad essere frutto delle attività apprese durante l'esperienza di stage, è dipesa dalla passione personale che mi lega alle materie coinvolte: Ingegneria e Qualità del software, Project Management, Prestazioni e Affidabilità dei

Sistemi. Infatti la tesi può essere vista come l'unione, correlazione e rivisitazione dei concetti di suddette materie di studio, dando maggiore peso all'aspetto di Ingegneria e Qualità del Software nella definizione della metodologia. Corsi universitari intrapresi, così come seminari e approfondimenti personali, sono da considerarsi le "fondamenta" che mi hanno agevolato nell'avvio di questo lavoro.

1.4 Organizzazione della tesi

Allo scopo di perseguire gli obiettivi preposti per la realizzazione del lavoro e allo stesso tempo evidenziare l'approccio empirico seguito e l'importanza personalmente attribuitagli, la tesi è suddivisa in tre parti e si articola nei seguenti capitoli:

I. ANALISI DEL CONTESTO

▪ Capitolo 2: QUALITÀ NELLA PRODUZIONE DEL SOFTWARE

Per comprendere come definire e posizionare la metodologia APM, questo capitolo presenta sia l'insieme di attività per la conduzione dei progetti software e i criteri con cui queste vengono intraprese (**cicli di vita**), sia le **metodologie di sviluppo** da seguire per ottenere dei prodotti software che rispondono a requisiti di Qualità. Gli obiettivi del capitolo sono duplici: da un lato definire e strutturare, a diversi livelli di astrazione, una **terminologia comune** per il processo di produzione del software e applicarla poi coerentemente nella metodologia APM, dall'altro illustrare i principali concetti legati alla **Qualità del software**. Essi costituiscono i parametri secondo i quali giudicare la bontà di una soluzione informatica.

▪ Capitolo 3: GESTIONE DEI REQUISITI NON FUNZIONALI

In un modello di ciclo di vita, qualunque esso sia, l'Ingegneria dei requisiti (**Requirements Engineering, RE**) è forse la fase più delicata e allo stesso tempo più critica dell'Ingegneria del software e uno dei criteri di ingresso che facilita l'introduzione e l'applicazione della metodologia APM. In questo capitolo sono definiti i concetti e le attività di analisi, ricerca e **modellazione dei requisiti** che consentono di descrivere un problema e la sua soluzione. Questi concetti vengono applicati nello specifico ai requisiti non funzionali evidenziando dei metodi per superare la loro difficoltà di espressione e quantificazione, motivi che portano a trascurarli o posizzionarli in secondo piano. L'analisi termina con la classificazione e descrizione degli attributi o requisiti di **QoS (Quality of Service) di efficienza e affidabilità**, utili per misurare ed ottenere il raggiungimento dei livelli prestazionali attesi dagli utenti finali. Per la definizione di ogni requisito di performance ci si riferisce prettamente all'ambito delle applicazioni Web.

▪ **Capitolo 4: SOFTWARE TESTING**

Questo capitolo definisce i concetti e l'importanza del Software Testing per il processo di Quality Assurance (QA), distinguendo gli obiettivi e l'utilizzo dei momenti di verifica e validazione. Vengono presentate le molteplici tecniche di testing: **approccio empirico**, che si distingue in statico (*Walkthrough* e *Software Inspection*) e dinamico (*White-Box* e *Black-Box*) e le **tecniche formali**. Viene illustrato, riferendosi a standard e linee guida presenti sul mercato e direttamente sperimentate, le fasi del processo di software testing (dal test di modulo al test d'installazione) e il ciclo di vita da seguire in ogni momento di test (**Software Testing LifeCycle**). Inoltre si evidenziano le necessità organizzative, i ruoli, la documentazione adeguata e gli strumenti che consentono l'automazione delle attività di testing. L'analisi di questi concetti favorisce la corretta introduzione del **Performance Testing**: sia in termini di processo (posizionandolo correttamente all'interno delle fasi dell'intero processo di Software Testing) sia come insieme di tecniche di test specifiche.

▪ **Capitolo 5: MISURARE LE PRESTAZIONI DEI SISTEMI**

L'analisi quantitativa delle prestazioni dei sistemi software consente di valutare, (attraverso metodi di **modellazione del sistema** oppure per sistemi preesistenti, metodi basati su **tecniche di misurazione**), le caratteristiche di efficienza e affidabilità delle applicazioni e dell'infrastruttura ICT, servendosi di metriche ben definite. Il capitolo fornisce una panoramica dei diversi metodi di valutazione delle prestazioni da applicare durante le diverse fasi del ciclo di vita del software (Design, Testing & Deploy, Operation & Maintenance). Dall'analisi di lavori di ricerca, riferimenti e contributi apportati da società di consulenza nell'ambito del **Software Performance Measurement** sono stati definiti: i **concetti fondamentali** per intraprendere un'attività di testing prestazionale; le **tecniche di performance test** (standard e particolari); ed infine la configurazione della **strumentazione di Performance Testing** utilizzata per simulare il carico di lavoro a cui l'applicazione/infrastruttura sarà sottoposta (*Protocol-simulation Tool*).

II. LA METODOLOGIA APM

▪ **Capitolo 6: PERFORMANCE TESTING E MONITORING**

In questo capitolo viene presentata la metodologia APM, proposto un modello di WorkLoad per la rappresentazione e la progettazione dello scenario per i test di performance ed infine vengono descritte le caratteristiche e le attività proprie dei due processi che costituiscono la metodologia APM: Performance Testing e End User Monitoring.

III. RISULTATI SPERIMENTALI

▪ Capitolo 7: IL CONTESTO AZIENDALE

In questa sezione, si evidenzia un quadro complessivo della realtà per cui e dove effettivamente si è svolta l'esperienza di stage: l'azienda **Aive S.p.A** come fornitore e **Generali Servizi Informatici S.p.A (GSI S.p.A)** come Cliente. Oltre che alla storia, *mission* e struttura organizzativa della società GSI S.p.A, si è dedicato maggiore spazio nella presentazione del progetto **SOFTWARE LIFECYCLE**. Il suddetto progetto si propone di definire e adottare le Best practices di riferimento per le discipline che compongono il ciclo di sviluppo del software nel rispetto delle priorità e dei piani operativi previsti, capitalizzando gli asset (beni, risorse) e gli skill esistenti. Un ambito di questo progetto riguarda proprio la definizione di un **PROCESS FRAMEWORK**, una base di processi e metodologie di sviluppo comuni a tutti i gruppi di sviluppo ed al personale coinvolto.

▪ Capitolo 8: CASI DI STUDIO

Infine, viene riportata una descrizione sommaria degli strumenti utilizzati e della prima industrializzazione del Performance Testing in **GSI S.p.A**. Sono descritti due di test di performance eseguiti durante il periodo di stage e il **Progetto di Normalizzazione e Gestione Sonde** (Implementazione dei Processi e definizione di Standard) che ha consentito la prima industrializzazione dello End User Monitoring di tipo attivo, del quale mi sono occupata in prima persona durante lo stage.

1.4.1 Suggerimenti per la consultazione

Nella seguente figura vengono indicate le interdipendenze fra i vari capitoli e di questa tesi. Nel grafo è presente un arco orientato fra il Capitolo *i* e il capitolo *j* se il contenuto del capitolo *i* è necessario alla comprensione del capitolo *j*. In certi casi, viene riportato sugli archi del grafo il/i paragrafi la cui lettura è da considerarsi un prerequisito fondamentale. Implicita la lettura nella sua interezza, qualora non venga riportato nulla.

Inoltre vengono suggerite tre modalità di consultazione, a seconda della tipologie di persone interessate al lavoro di tesi:

- **STUDENTI UNIVERSITARI e RICERCATORI**: che vogliono comprendere i concetti e le tecniche di base utilizzate per condurre e gestire un progetto di sviluppo software, orientato ai principi di Qualità e ottimizzazione dell'ICT. È consigliata la lettura della tesi nella sua interezza.
- **ESPERTI NELL'AMBITO ICT**: persone con conoscenza generale alle problematiche di progettazione, sviluppo, manutenzione e assistenza di sistemi, ad esempio perché operano già in una organizzazione ICT. Ad esempio Responsabile dei Sistemi Informativi, Responsabili applicativi, analisti e sviluppatori, amministratori di database). A queste figure

professionali è utile conoscere la metodologia ad un livello prevalentemente concettuale (quadro generale dei processi e delle tecniche di test), in quanto possono essere direttamente coinvolte dai gruppi incaricati nelle attività di Performance Testing e Monitoring, per collaborare alla definizione del test attraverso l'identificazione delle esigenze, il reperimento di informazioni, dati applicativi e la preparazione degli ambienti necessari per eseguire il test stesso. Si consiglia in particolare la lettura mirata dei paragrafi: (2.1,2.2), (3.3), (4.3,4.5), (5.3, 5.4, 5.5), (6.1, 6.3, 6.4).

- **ESPERTI NELL'AMBITO DEL SOFTWARE TESTING:** professionisti che ricoprono già funzioni in ambito del Controllo e Assicurazione Qualità o mansioni di Project Manager e che desiderano costruire per la propria organizzazione dei centri o laboratori specializzati alla verifica prestazionale delle applicazioni. In questo caso si consiglia la lettura integrale dei capitoli 5 e 6 (che prevede come prerequisito minimo la lettura dei paragrafi 2.1,2.2,3.3,4.3,4.5) e dei risultati sperimentati presentati nel capitolo 8.

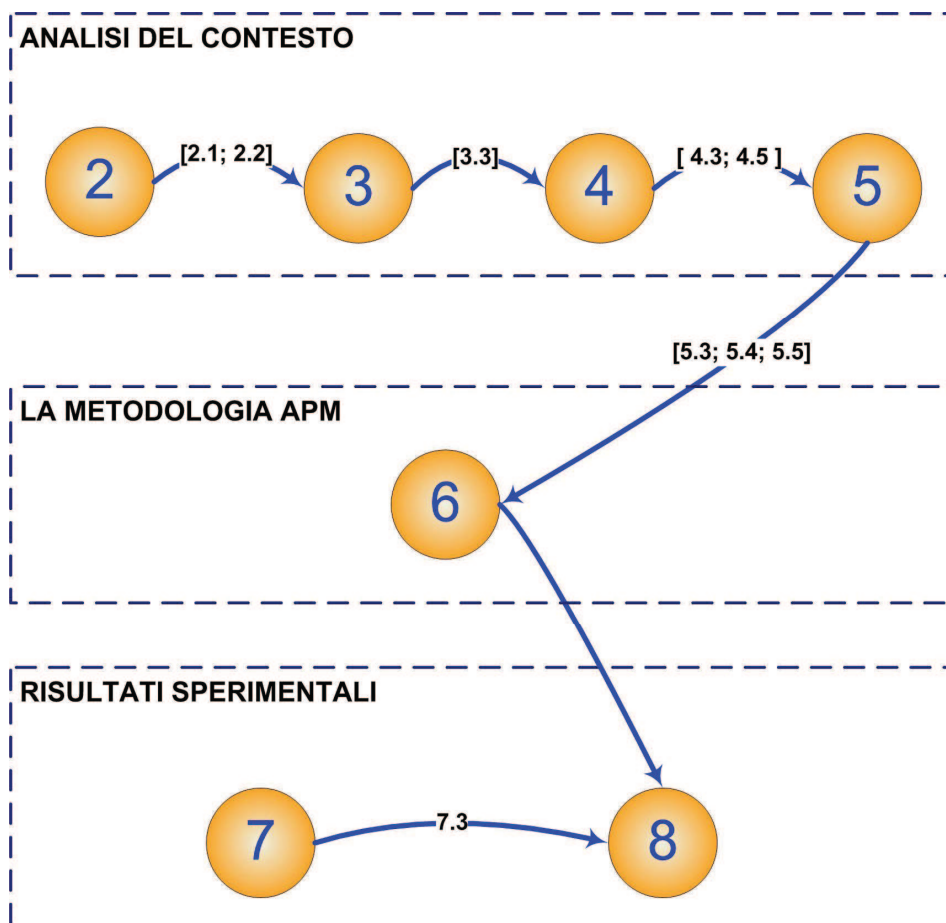


Figura 1.1: Interdipendenza dei capitoli della tesi

CAPITOLO 1. INTRODUZIONE

Parte I

Analisi del contesto