



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

Gamification e Serious Games per l'Insegnamento di Software Testing: una Revisione Sistemica della Letteratura

RELATORE

Prof. Fabio Palomba

Dott. Stefano Lambiase

Università degli Studi di Salerno

CANDIDATO

Jacopo Gambardella

Matricola: 0512105792

Anno Accademico 2022-2023

Questa tesi è stata realizzata nel

sesa^{lab}
SOFTWARE ENGINEERING
SALERNO

Abstract

Il testing è una pratica fondamentale per lo sviluppo di software di buona qualità, tuttavia dagli studenti di ingegneria del software è visto spesso come un compito noioso ed ostico, a cui avvicinarsi non è facile. Le conseguenze di questo mancato interesse possono portare ad un aumento di bug ed alla compromissione del corretto funzionamento di software moderni. *Gamification* e *serious game* sono possibili soluzioni a questi problemi. Con il termine *gamification* si intende l'utilizzo di meccanismi che si ritrovano comunemente nei giochi applicati in un contesto educativo; i *serious game* sono giochi progettati con l'obiettivo di educare e rendere l'esperienza formativa un'attività piacevole ed efficace. Obiettivo del presente lavoro è stato quello di: (1) determinare le strategie di gamification e serious games maggiormente utilizzate per l'insegnamento del testing e la loro rispettiva efficacia e (2) fornire una serie di consigli su quelli che potrebbero essere i prossimi passi nella ricerca nell'utilizzo di gamification e serious games nell'ambito del testing. Per raggiungere questi obiettivi si è condotta una systematic literature review interrogando i più famosi database scientifici e individuando 9 articoli riguardo l'argomento.

Indice

Elenco delle Figure	iii
Elenco delle Tabelle	iv
1 Introduzione	1
2 Stato dell'Arte	4
2.1 Background	4
2.1.1 Testing	4
2.1.2 Gamification	6
2.1.3 Serious Game	7
2.1.4 Challenge nell'insegnamento dell'ingegneria del software . .	8
2.2 Related Work	9
3 Metodo di Ricerca	11
3.1 Obiettivo e domande di ricerca	11
3.2 Definizione della stringa di ricerca e database	12
3.3 Criteri di Inclusione e Esclusione	13
3.4 Criteri di qualità	15
3.5 Estrazione dei dati	16

4	Analisi dei Risultati	17
4.1	Informazioni generali sul dataset	17
4.2	Tecniche di gamification utilizzate per il testing	18
4.3	Efficacia delle strategie di gamification	20
4.4	Serious game nel software testing	23
4.5	Efficacia dei Serious Games	25
5	Minacce alla validità	30
6	Conclusioni	32
	Bibliografia	34

Elenco delle figure

3.1	Risultati prodotti dalla stringa di ricerca	13
3.2	Risultati totali	14

Elenco delle tabelle

3.1	Inclusion-Exclusion criteria	14
3.2	Quality criteria	16
4.1	Elementi di gamification	21
4.2	Serious game e tecniche di testing	27
4.3	Dataset 1	28
4.4	Dataset 2	29

CAPITOLO 1

Introduzione

Contesto. Il testing è una fase fondamentale nello sviluppo del software, è il processo di valutazione e verifica che un prodotto software soddisfi i requisiti specificati. Un buon testing non mira soltanto a trovare errori nel sistema, ma cerca anche di migliorare il software in termini di efficienza, accuratezza e usabilità. Tipicamente il testing è classificato in tre categorie: testing funzionale, testing non funzionale o performance testing, e testing di manutenzione o di regressione. Ognuna di queste categorie assicura che un software sia di elevata qualità, sicuro in termini di vulnerabilità ed economicamente sostenibile. La presenza di bug nei software comporta un enorme spreco di risorse: causano crash delle applicazioni, il costo di costanti update sfiora i budget iniziali causando ingenti danni economici, come nel caso di sistemi bancari e servizi di pagamento online, danneggiano brand e riducono la fiducia del cliente nei confronti del prodotto. Pericoli ben più gravi si sono verificati in passato a causa di testing insufficiente, per esempio, tra il 1985 ed 1987 la macchina per la radioterapia Therac-25, a causa di un bug nel software, fornì dosi di radiazioni eccessive ai pazienti che ne fecero uso causandone la morte o ferendoli gravemente.¹ Conseguenza di questi pericoli è la sempre più alta richiesta di figure professionali

¹<https://ethicsunwrapped.utexas.edu/case-study/therac-25>

da parte delle aziende per prevenire i pericoli dovuti a software non adeguatamente testati.

Limitazioni. Tuttavia la difficoltà e la ripetitività del testing fa sì che gli studenti lo vedano come una fase trascurabile, noiosa da eseguire e che richiede tempo, siccome richiede la piena comprensione del contesto del programma e del codice su cui lo si svolge. Il testing richiede inoltre una solida capacità di scrivere codice, e questo può rappresentare una sfida per gli studenti che non possiedono una buona base di programmazione. A seguito di queste motivazioni, si nota una mancanza di interesse generale ed una scarsa motivazione degli studenti nell'apprendimento di questa importante disciplina. Questo può portare una riduzione dell'efficacia del testing stesso e alla mancanza di figure professionali.

Contributi e risultati. Per sopperire alla mancanza di interesse e di motivazione degli studenti si sono esplorate nuove tecniche di apprendimento, quali gamification e serious Games, anche definiti con il termine "educational games". Utilizzando queste due tecniche si cerca di trasformare un lavoro complesso e tedioso in un'attività divertente e positiva, facendo leva su elementi chiave e punti di forza quali: obiettivi, senso di progresso ed interazione sociale. La systematic literature review condotta mira a fornire una panoramica degli strumenti maggiormente utilizzati tramite gamification per raggiungere i vantaggi sopra elencati, e si promette di determinare quanto questi siano stati efficaci per l'insegnamento del testing. Al contempo si individueranno i serious games utilizzati nello stesso ambito e se ne determinerà la validità. Infine verranno proposti consigli su quelli che potrebbero essere i prossimi passi nella ricerca. Per condurre la systematic literature review, è stata formulata la seguente main research question:

"Quali sono gli approcci di gamification ed i serious games proposti in letteratura per l'insegnamento del testing?"

Le seguenti sottodomande sono poi state formulate:

- Quali sono le strategie di gamification utilizzate per l'insegnamento del testing?
- Quali sono i serious games utilizzati per l'insegnamento del testing?
- Quanto si sono dimostrate efficaci le strategie di gamification utilizzate?

- Quanto si sono dimostrati efficaci i serious games utilizzati?

Struttura dell'elaborato. La tesi è strutturata come segue. La Sezione 2 descrive lo stato dell'arte, in particolare si approfondiscono le tematiche principali di questa ricerca, si analizzano le definizioni di testing, gamification e serious games e si individuano vantaggi e svantaggi nell'apprendimento. La Sezione 3 descrive la metodologia di ricerca utilizzata per condurre una revisione sistematica della letteratura. Nella Sezione 4 si trovano le risposte alle domande di ricerca. La Sezione 5 è dedicata alla descrizione delle minacce alla validità della revisione sistematica della letteratura. Infine, la Sezione 6 riassume le principali conclusioni della ricerca e si forniscono possibili direzioni future per la ricerca in questo campo.

2.1 Background

2.1.1 Testing

Il testing, secondo la definizione di Bruegge, è la pratica atta ad individuare problemi nel comportamento del sistema che differiscono dal comportamento desiderato [1]. L'obiettivo principale del testing è trovare il massimo numero di fault nel software, ma non dimostrarne l'assenza, questo perchè è pressochè impossibile prendere in considerazione tutti i possibili input e scenari di utilizzo del software. Al fine di raggiungere questo scopo il testing è spesso messo in atto da sviluppatori inizialmente non coinvolti nello sviluppo del sistema [1]. Questo perchè gli sviluppatori che hanno lavorato al software potrebbero non essere in grado di individuare possibili difetti o potrebbero essere portati a non trovarne in quanto "ciechi" rispetto a problemi che non sono stati considerati e che non si aspettano. Per questo motivo, il testing dovrebbe essere assegnato ad un team di tester che lavorano in modo indipendente dal team di sviluppatori utilizzando tecniche specifiche per individuare difetti nel software. Le tecniche di testing includono testing manuale, coverage testing, testing automatizzato o testing basato su generazione casuale di input. Per facilitare

la lettura si sono classificate queste tecniche:

Coverage testing. Anche detto testing di copertura, è una tecnica di testing usata per valutare la qualità dei test eseguiti su un software. L'obiettivo del coverage testing è misurare quanti e quali parti del codice sorgente sono state eseguite durante l'esecuzione del test. Il codice sorgente è suddiviso in porzioni, come ad esempio linee di codice, funzioni e classi. In base al livello di copertura ottenuto è possibile verificare la quantità di codice testata ed individuare le parti del codice che non sono state testate a sufficienza.

Functional testing. Il testing funzionale mira a verificare che il software soddisfi le specifiche funzionali richieste dal cliente o dall'utente finale. Il testing funzionale può includere diversi tipi di test, come ad esempio, test di regressione, test di accettazione, test di integrazione, test di sistema e test di validazione. Il tipo di test scelto dipende ovviamente dalle esigenze del progetto.

Unit Testing. Il testing di unità è una tecnica di testing che si concentra sulla verifica del comportamento e della funzionalità di una singola unità di codice, come una funzione, un metodo o una classe. Il testing di unità è solitamente svolto dagli sviluppatori stessi durante lo sviluppo del software per correggere eventuali errori o problemi di funzionamento nel codice il prima possibile. Il testing di unità può essere eseguito sia in maniera manuale sia con un approccio automatico tramite strutture di test come Junit per il linguaggio di programmazione Java, che semplificano la creazione e l'esecuzione dei test di unità.

Exploratory Testing. Il testing esplorativo è una tecnica di testing che si basa sull'esplorazione, la sperimentazione e l'osservazione del software con un approccio non pianificato. Il tester esplora il software come se fosse un utente reale, cercando di identificare problemi o errori che potrebbero non essere evidenti tramite le tecniche di testing tradizionali. Questa tecnica richiede al tester grande creatività, flessibilità ed esperienza.

Mutation Testing. La mutation testing è una tecnica di testing del software che prevede la creazione di versioni modificate (mutanti) del codice sorgente

originale. Questi mutanti differiscono dal codice originale in un piccolo e specifico modo, come ad esempio la modifica di un operatore aritmetico o logico, l'inserimento di una condizione errata o la rimozione di una linea di codice. Lo scopo della mutation testing è di valutare l'efficacia dei test di unità, ovvero verificare se i test siano in grado di individuare eventuali difetti introdotti dalle mutazioni. In pratica, si eseguono i test di unità sulle versioni mutate del codice e si controlla se questi test falliscono, indicando quindi la presenza di un difetto.

Per testare un sistema in maniera efficace, un tester deve possedere una buona e dettagliata conoscenza del sistema e delle tecniche da utilizzare per rispettare budget, tempo e requisiti qualitativi. In particolare, il tester deve essere in grado di identificare i rischi del progetto e di definire un piano di test adeguato per coprire tutti gli aspetti critici del sistema. Inoltre, deve essere in grado di selezionare le tecniche di test più appropriate per le diverse fasi del ciclo di vita del software, e di valutare i risultati dei test per individuare e correggere i difetti. Pensiero comune ed errato dei manager è vedere il testing come un lavoro che può essere svolto anche da principianti, lasciando figure più esperte e con maggiore esperienza su attività ritenute più impegnative [1]. Per questi motivi, si può sottovalutare l'importanza del testing, ed a non dedicare le risorse e l'attenzione necessarie per assicurare una buona qualità del prodotto software.

2.1.2 Gamification

Gamification è l'utilizzo delle meccaniche di gioco per coinvolgere le persone, motivarle e promuovere l'apprendimento [2]. Il termine "meccaniche" include tutti gli elementi presenti nei giochi moderni, tra questi si trovano:

- **Badge:** sono la rappresentazione di un obiettivo raggiunto [3].
- **Punti:** sono ottenuti al termine di un compito, il punteggio è utilizzato come sistema di misura di successo [3].
- **Livelli:** dividere l'attività in livelli permette di dare un senso di progresso al giocatore. I livelli iniziali richiederanno meno sforzo al giocatore per essere

completati, la difficoltà di questi aumenterà di pari passo col progresso del giocatore [3].

- **Storyline:** la narrativa viene utilizzata abitualmente nei giochi per aumentare la motivazione, il giocatore viene coinvolto nella narrazione e vuole contribuire a farla progredire [3]. Ne consegue un'attività svolta con maggior entusiasmo ed attenzione.
- **Leaderboard:** rappresenta la classifica dei punteggi dei giocatori e favorisce un ambiente competitivo tra giocatori [3].

La combinazione di questi elementi contribuisce a creare quel che viene definito da Kapp "Game Thinking", cioè l'idea di pensare ad una esperienza abitudinale e convertirla in un'attività che ha elementi di competizione, cooperazione e storytelling, applicabile quasi in tutti i contesti [2]. Per sfruttare al meglio il "Game Thinking" è bene chiarire che non è sufficiente inserire gli elementi sopra descritti in una qualsiasi attività per ottenere ottimi risultati, è necessaria una buona conoscenza dell'attività in cui questi elementi verranno impiegati ed è importante conoscere come un giocatore si interfaccia ad un gioco. Il giocatore inizia a giocare ottenendo feedback istantanei mentre cerca di completare una challenge, definita da regole precise, il completamento della challenge scatena una reazione emotiva nel giocatore, che ne ottiene senso di gratificazione e premi quantificabili [2]. Inoltre, è importante considerare che l'applicazione del Game Thinking non è adatta a tutte le attività e a tutti i contesti. L'obiettivo deve essere quello di creare esperienze coinvolgenti e motivanti che siano in grado di migliorare la partecipazione e l'apprendimento, ma è necessario fare attenzione a non creare distrazioni o a non compromettere la qualità dell'attività svolta.

2.1.3 Serious Game

I "Serious Games" sono giochi designati per l'addestramento, la pubblicità, la simulazione e l'educazione che possono essere utilizzati su pc o console [4]. Per chiarire meglio la definizione di serious game di seguito si evidenziano alcune differenze con il termine gamification. Benchè gli obiettivi dei serious games siano

quasi del tutto simili a quelli definiti da gamification, questi si differenziano nel metodo utilizzato. Mentre con il processo di gamification si utilizzano meccaniche di gioco applicate ad azioni reali, i serious games sono utilizzati questi elementi nel contesto di un gioco, che sia un gioco da tavolo o un gioco web-based [2]. I serious game sono strumenti di insegnamento molto efficaci ed altamente motivanti poichè riescono a comunicare in maniera ottimale concetti ed argomenti. I giocatori di serious game assumono all'interno di questi un ruolo, prendono decisioni, formulano strategie e ricevono veloci feedback come conseguenza delle loro azioni. Si può dedurre quindi che un vantaggio intrinseco dei serious game è la possibilità dei giocatori di poter esplorare e commettere errori tentando di trovare soluzioni a problematiche reali in un ambiente privo di rischi e senza alcuna conseguenza. Esempio di applicazioni di questa metodologia posso essere trovate in diversi campi, come la formazione militare, medica, aziendale e scolastica. Possono essere utilizzati per simulare situazioni di emergenza o per insegnare competenze come la gestione del tempo, la collaborazione o la risoluzione di conflitti in un ambiente di lavoro.

2.1.4 Challenge nell'insegnamento dell'ingegneria del software

Di pari passo all'insegnamento dell'ingegneria del software sorgono nuove sfide e difficoltà da affrontare. Coinvolgere gli studenti è una delle principali challenge, causa la forte componente teorica e l'insufficienza di praticità nelle classi [5]. Trovare quindi il giusto equilibrio tra la componente pratica e quella teorica sembra essere tutt'oggi un problema da risolvere nei corsi di ingegneria del software. Cercare di colmare le differenze tra scenari teorici e scenari reali è un'altra grande sfida da affrontare per gli esperti della materia, di fatti è difficile trovare tool e tecnologie da integrare in una classe, che siano contemporaneamente gratuite ed utilizzati all'interno dell'ambiente accademico [5]. L'evoluzione continua e veloce dell'informatica rappresenta in aggiunta, una difficoltà per gli insegnanti, che devono aggiornare al contempo le proprie conoscenze sulle ultime metodologie e tecnologie utilizzate in campo lavorativo. Con il corso degli anni si è tentato di proporre soluzioni alle problematiche descritte; internet ha avuto un ruolo fondamentale, permettendo l'accesso ad una illimitata riserva di informazioni per rispondere alle nuove complicazioni

sorte in questo campo [6].

2.2 Related Work

In questa sezione si andranno a presentare e descrivere i lavori vicini al presente lavoro di tesi e in che modo ci si è differenziati da essi.

Nel systematic mapping di Pedreira et al. [7] sono state raccolte informazioni esistenti nella letteratura, classificate ed analizzate poi secondo 4 criteri: (1) la fase dello sviluppo del software coinvolta, (2) gli elementi di gamification utilizzati, (3) il metodo di ricerca applicato e (4) il tipo di pubblicazione. I risultati sono stati valutati successivamente in termini di metodologia, replicabilità ed evidenza empirica. Si osserva che gli studi presi in considerazione incorporavano il processo di gamification in tool adhoc indipendenti, indicando una insufficiente presenza di guidelines e soluzioni che aiutino ad integrare questa tecnica in ecosistemi esistenti. La ricerca effettuata nel 2015 indica che l'adozione di gamification applicata all'ingegneria del software procede in maniera più lenta rispetto a campi già ben largamente esplorati come quello del marketing. Infine, sono stati proposti nuovi consigli e futuri campi da esplorare sulla base dei risultati ottenuti.

Kosa et al. [8] pone come obiettivo principale quello di ottenere una panoramica dell'attuale educazione game-based nell'ingegneria del software. Per fare ciò sono stabiliti i seguenti sotto-obiettivi: (1) si cerca di comprendere quali sono i tipi di giochi principalmente utilizzati in questo campo, (2) investigare i vantaggi ottenuti dai giochi utilizzati, (3) esplorare le metodologie, se presenti, utilizzate per lo sviluppo di serious game nell'ingegneria del software. Lo studio mostra che in generale, i serious game incrementano la motivazione, fungendo da possibile punto di accesso per studenti demotivati e, da valore aggiunto a studenti già con sufficiente motivazione allo studio della materia. Si nota anche una prevalenza di serious game digitali rispetto all'approccio non digitale, viene però specificato che l'uso di un serious game non digitale dovrebbe comunque essere preso in considerazione per argomenti particolarmente complessi che coinvolgono aspetti sociali.

Backlund et al. [9] conduce una meta-analisi degli educational game nel campo dell'istruzione per studiarne l'efficacia. Lo studio è limitato al contesto scolastico:

scuole elementari, scuole superiori, college ed università. I risultati ottenuti sono stati selezionati sulla base di tre criteri: (1) paper nell'intervallo tra 2002 e prima metà del 2012, (2) studi che hanno valutato empiricamente l'efficacia, (3) studi pubblicati su riviste scientifiche. I 40 studi ottenuti dalla ricerca mostrano che 29 di questi hanno ottenuto risultati positivi.

Queste ricerche hanno analizzato le tecniche di gamification e serious game focalizzandosi singolarmente su una di esse. Tuttavia una systematic literature review che comprenda entrambe può fornire una visione più ampia e completa del campo. Inoltre, la comparazione di gamification e serious game può evidenziare un quadro completo sulle loro potenzialità ed i loro limiti, offrendo un'idea migliore su come applicarli in modo efficace e mirato nell'ingegneria del software. Gli studi citati hanno rivolto particolare attenzione ad un argomento molto ampio quale l'ingegneria del software, la seguente revisione sistematica concentra le analisi sul testing, un campo più ristretto e specifico, al fine di migliorare ed ottimizzare un'aspetto importante della materia.

Metodo di Ricerca

3.1 Obiettivo e domande di ricerca

L'obiettivo della ricerca è identificare le tecniche di gamification ed i serious game maggiormente utilizzati per l'insegnamento del testing nei corsi di ingegneria del software. In particolare si cercherà di comprendere l'efficacia di queste tecniche nella promozione dell'apprendimento degli studenti e nella loro motivazione ad imparare, si cercherà inoltre di fornire una panoramica completa ed aggiornata di esse identificando le migliori pratiche per il loro utilizzo. I risultati della ricerca potranno essere utilizzati per migliorare la progettazione dei programmi di studio e degli strumenti didattici nei corsi di ingegneria del software, al fine di migliorare l'efficacia dell'apprendimento degli studenti. Infine si forniranno, sulla base dei risultati ottenuti, consigli per direzioni future nella ricerca in questo campo.

Le domande di ricerca poste sono:

RQ₁. Quali sono le strategie di gamification maggiormente utilizzate per l'insegnamento del testing?

RQ₂. Quanto sono efficaci le strategie di gamification per l'insegnamento del testing?

RQ₃. Quali sono i serious game maggiormente utilizzati per l'insegnamento del testing?

RQ₄. Quanto è efficace l'utilizzo dei serious game per l'insegnamento del testing?

Per rispondere alle domande di ricerca è stata condotta una systematic literature review. La metodologia è composta da una serie di tecniche utilizzate durante il processo di ricerca: definizione della stringa di ricerca, definizione dei criteri di inclusione ed esclusione, definizione dei criteri di qualità ed estrazione dei dati.

3.2 Definizione della stringa di ricerca e database

Sono stati selezionati i database scientifici:

- IEEEXplore
- Scopus
- ACM Digital Library

Dopo la selezione dei motori di ricerca su cui reperire il materiale accademico, si sono individuate le parole chiave per comporre la seguente query di ricerca:

("gamification" OR "serious games") AND ("software testing") AND ("learning" OR "education" OR "teaching")

Risultati prodotti dalla stringa Successivamente alla composizione della stringa di ricerca, come mostrato nella figura 3.1, si sono ottenuti 104 risultati. Successivamente, sono stati rimossi 16 duplicati. Inoltre, tramite snowballing backward, sono stati ottenuti ulteriori tre studi. Dopo aver eliminato i duplicati e completato la fase di snowballing, si è giunti ad un risultato complessivo di 91 documenti scientifici. La figura 3.2 mostra i risultati ottenuti.

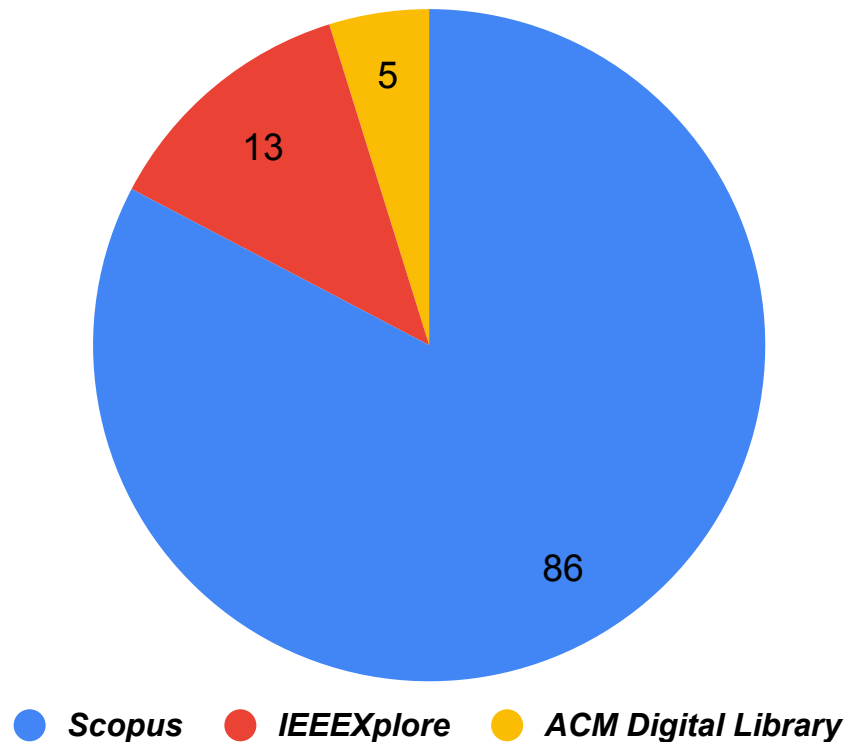


Figura 3.1: Risultati prodotti dalla stringa di ricerca

3.3 Criteri di Inclusione e Esclusione

In seguito, per ridurre il campo di analisi è stato necessario applicare criteri di inclusione ed esclusione per determinare quali documenti fossero utili alla ricerca e quali invece dovessero essere scartati. L'applicazione dei criteri di inclusione ed esclusione ha permesso di selezionare gli studi rilevanti per la revisione sistematica e di eliminare quelli non pertinenti. In questo modo, è stata garantita la qualità e la pertinenza della ricerca condotta sulla gamification e sui serious game per l'insegnamento del testing nell'ingegneria del software. Nella tabella 3.1 sono riportati i criteri di inclusione ed esclusione applicati. Nella prima riga si trovano i criteri utilizzati per considerare un documento, nella seconda riga sono riportati i criteri utilizzati per scartare un documento. Qualora almeno uno dei criteri di esclusione sia soddisfatto l'articolo sarà scartato. Il rispetto di almeno uno dei criteri di inclusione determina l'inclusione dell'articolo nella ricerca. Dei 91 risultati totali soltanto 12 soddisfacevano i criteri definiti.

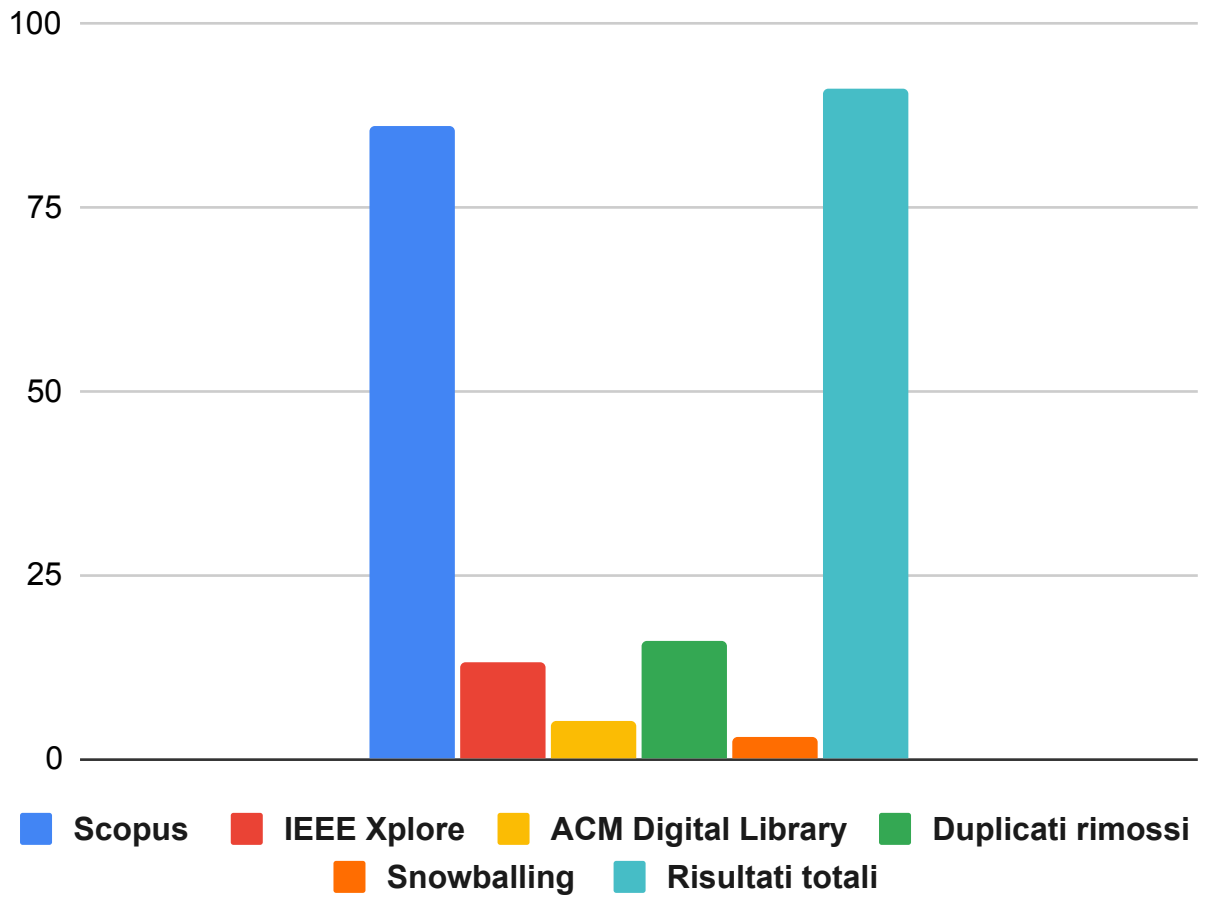


Figura 3.2: Risultati totali

Exclusion Criteria	Inclusion Criteria
Non è scritto in inglese	È scritto in inglese
Non riguarda gamification o serious games relativi al testing nell'ingegneria del software	Mostra dati che aiutano a studiare l'efficacia di gamification nel testing
Non è di almeno 3 pagine	È almeno 3 pagine
	Propone strategie di gamification o serious games per l'insegnamento del software testing

Tabella 3.1: Inclusion-Exclusion criteria

3.4 Criteri di qualità

Dopo aver superato le fasi iniziali di filtraggio, sono stati stabiliti i criteri di qualità, per garantire la validità e l'affidabilità della revisione sistematica. In particolare, i criteri di qualità considerati nella SLR si basano su una combinazione di elementi oggettivi e soggettivi, che riflettono sia la qualità metodologica degli studi inclusi, sia la pertinenza degli obiettivi di ricerca rispetto all'argomento trattato. I seguenti criteri sono stati stabiliti:

- **Q1:** *Le informazioni rilevanti degli studi sono chiaramente riportate ed indicate nell'articolo?*
- **Q2:** *Le conclusioni sono riportate chiaramente nell'articolo?*
- **Q3:** *Le domande di ricerca sono chiaramente definite?*
- **Q4:** *Le informazioni di base/gli studi sono stati descritti adeguatamente?*

A ciascuno degli articoli scientifici è stato assegnato un punteggio tra 0, 0.5 ed 1, dove 0 indica che l'articolo non ha rispettato il criterio di qualità, 0.5 che l'articolo ha rispettato sufficientemente il criterio di qualità, ed 1 che l'articolo ha rispettato pienamente il criterio di qualità. Questi criteri di qualità sono stati utilizzati come strumento di valutazione degli studi inclusi nella revisione sistematica, al fine di determinare la loro idoneità a contribuire all'analisi e di garantire la validità e l'affidabilità dei risultati ottenuti. L'assegnazione del punteggio ha consentito di stabilire una gerarchia degli studi in base alla qualità metodologica e alla rilevanza dei risultati rispetto agli obiettivi della ricerca. L'utilizzo di criteri di qualità chiaramente definiti e di un sistema di punteggio ha quindi rappresentato uno strumento essenziale per garantire l'obiettività e la trasparenza del processo di selezione degli studi inclusi nella revisione sistematica. Nella tabella 3.2 sono riportati gli studi che hanno superato la prima fase di filtraggio, sono inoltre mostrati i punteggi che hanno raggiunto per ogni criterio qualitativo. Gli articoli sono identificati nella tabella dagli stessi identificativi utilizzati nell'appendice online [10]. I paper che non hanno raggiunto un punteggio minimo di 2 sono stati scartati.

ID paper	Q1	Q2	Q3	Q4
P19	1	1	0.5	1
P24	0	0	0	0
P32	1	1	1	1
P33	0.5	1	0.5	0.5
P49	1	0.5	0.5	1
P65	0	1	0	0.5
P70	1	1	0.5	1
P83	1	0.5	0.5	1
P86	0.5	0	0.5	0.5
P89	1	1	1	1
P90	1	1	0.5	1
P91	1	1	0.5	1

Tabella 3.2: Quality criteria

3.5 Estrazione dei dati

Durante la fase di estrazione dei dati, sono state analizzate le informazioni rilevanti per la ricerca: le strategie di gamification utilizzate, i serious game proposti, quanto si sono rivelati efficaci i metodi di gamification e serious game proposti, la tipologia di testing si proponeva di insegnare, il nome degli autori e l'anno di pubblicazione. Tramite questa analisi dei dati finale, sono state sintetizzate le informazioni essenziali alla ricerca ed identificate le principali tendenze e le attuali lacune nella letteratura.

Analisi dei Risultati

4.1 Informazioni generali sul dataset

Il dataset mostrato in figura 4.3 include studi scientifici che esplorano l'utilizzo di tecniche di gamification e di serious game nel contesto del software testing. Nei risultati mostrati si cerca di identificare i seguenti aspetti:

- L'efficacia di tali tecniche nel migliorare la qualità del testing
- Gli aspetti motivazionali dell'utilizzo di gamification e serious game per il software testing
- L'effetto sul coinvolgimento degli utenti dopo l'utilizzo di tali tecniche
- L'impatto di gamification e serious game sull'apprendimento del software testing
- La valutazione della qualità delle tecniche di gamification e serious game utilizzate

Il dataset finale mostrato nelle tabelle 4.3 e 4.4 è composto da 9 paper e copre un periodo di tempo compreso tra il 2007 ed il 2020.

4.2 Tecniche di gamification utilizzate per il testing

RQ₁. Quali sono le strategie di gamification maggiormente utilizzate per l'insegnamento del testing?

Il primo obiettivo di ricerca era: **stabilire le strategie di gamification maggiormente utilizzate per l'insegnamento del software testing**. Negli studi analizzati, sono state identificate diverse tipologie di testing a cui sono state applicate tecniche di gamification. Per facilitare la lettura si sono classificate queste tecniche:

Coverage testing. Il coverage testing è una tecnica di testing del software che mira a valutare la copertura dei test rispetto al codice sorgente.

Nello specifico, per promuovere l'apprendimento dei concetti base del testing di copertura, Sherif et al. ha utilizzato un'applicazione che presentava agli studenti una serie di sfide di difficoltà crescente. Gli elementi di gamification utilizzati includevano *badge*, *punti esperienza* ed una *classifica* personale, un *combat system*, grafiche e musiche. Questi elementi sono stati implementati per migliorare la comprensione dei concetti base del coverage testing e per motivare gli studenti. L'applicazione sviluppata dagli autori copre diversi aspetti del testing di copertura, come ad esempio lo *statement coverage*, il *branch coverage* e la *path coverage*. L'applicazione presentava inoltre una modalità che consentiva di sfidare altri giocatori, promuovendo così l'aspetto competitivo tra i partecipanti.

Functional testing. Il test funzionale si concentra sulla verifica del comportamento dell'applicazione rispetto ai requisiti funzionali specificati.

Nello studio di Jesus et al. gli autori hanno sfruttato tecniche di gamification per insegnare ad un gruppo di studenti universitari concetti basilari sul testing ed il testing funzionale. L'ambiente gamificato presentava molti elementi già presenti nello studio descritto precedentemente, come *badge*, *classifiche*, *punteggi*, *ricompense* e *punti esperienza*. Lo strumento sviluppato dagli autori ha permesso di fornire un feedback istantaneo alle performance dei partecipanti.

Oltre agli elementi di gamification già descritti, lo strumento presentava al giocatore la possibilità di personalizzare la propria esperienza attraverso la scelta di un *avatar* e la partecipazione ad un forum, per aumentare l'aspetto sociale.

Unit Testing. Il testing di unità è una tecnica di testing software che prevede la scrittura e l'esecuzione di test specifici per verificare il corretto funzionamento di una singola unità di codice.

Marabesi e Silveira pone come obiettivo principale lo sviluppo di un tool per insegnare il testing di unità utilizzando tecniche di gamification e valutarne l'efficacia. Lo strumento risultante fa uso di elementi quali: *punti esperienza*, una *valuta* di gioco, reward casuali ottenuti dal progresso del giocatore, sfide di difficoltà crescente, e reward aggiuntivi in caso il giocatore decida di condividere lo strumento sui social media.

Exploratory Testing. Il testing esplorativo è una tecnica di testing software che prevede l'esecuzione di test in modo dinamico e adattivo, senza seguire un piano di test prestabilito.

Una strategia sistematica per insegnare il testing esplorativo attraverso l'utilizzo della gamification è proposta da Costa e Oliveira. La strategia si basa su tre fasi principali: (1) preparazione, (2) esecuzione e (3) valutazione. Nella fase di preparazione, gli studenti vengono introdotti ai principi del testing esplorativo ed ai concetti base di gamification. Nella fase di esecuzione, gli studenti applicano ciò che hanno imparato in un'attività gamificata di testing esplorativo. Infine, nella fase di valutazione, gli studenti ricevono un feedback sul lavoro svolto e vengono valutati sulla base dei risultati ottenuti. Nell'attività di progettata da Costa e Oliveira i giocatori guadagnano punti completando attività di testing esplorativo e raggiungendo obiettivi specifici. Gli elementi di gamification impiegati sono numerosi: *reward* per il completamento di attività, *monete* di gioco, *avatar* da personalizzare, *sfide* contro altri studenti per incentivare la competizione, *livelli* e *classifiche*.

Un tentativo di nascondere la tediosità del testing agli studenti è stato fatto nello studio di Bell, Sheth e Kaiser. In questo studio gli autori propongono "HALO" (Highly

Automated dL-pOwered testing), un framework integrabile con numerosi IDE. HALO utilizza elementi di gamification per rafforzare l'aspetto sociale e competitivo durante le fasi di testing, motivando gli studenti tramite classifiche, reward, monete scambiabili per crediti extra a discrezione degli insegnanti, sfide, challenge e storyline basate su alcuni dei più famosi MMORPG (Massively Multiplayer Online Role Playing Game).

Risposta alla domanda RQ1 ed osservazioni

Come si osserva dalla tabella 4.1, si individua una tendenza all'incentivare la competizione e l'aspetto sociale degli studenti tramite elementi di gamification, come (i) l'assegnazione di punteggi e classifiche, (ii) la partecipazione a sfide e competizioni e (iii) l'utilizzo di badge e livelli di competenza. Inoltre, la tabella mostra che gli elementi di gamification più utilizzati sono quelli che prevedono la possibilità di raggiungere obiettivi e di ottenere premi per il loro raggiungimento, il che può motivare gli studenti a impegnarsi di più nell'apprendimento delle tecniche di testing. In generale, l'utilizzo di elementi di gamification può rendere l'apprendimento del software testing più coinvolgente e motivante per gli studenti, migliorando così la qualità dell'insegnamento e dell'apprendimento.

4.3 Efficacia delle strategie di gamification

RQ₂. Quanto sono efficaci le strategie di gamification per l'insegnamento del testing?

Per ottenere una panoramica sull'efficacia degli elementi riportati nella tabella 4.1 sono stati esaminati i risultati ottenuti dagli studi precedentemente descritti. tuttavia, dei cinque studi esaminati sopra [11, 12, 13, 14, 15], soltanto due hanno presentato prove empiriche:

- Il paper di Sherif et al. ha proposto l'utilizzo di un tool di gamification per insegnare ai partecipanti i concetti di base del software testing e del testing di copertura. Lo strumento è stato sottoposto a un gruppo di 20 studenti universi-

Titolo	Keywords	Tipologia di testing	Elementi di Gamification
Gamification to Aid the Learning of Test Coverage Concepts	Computer aided instruction; gamification, software testing; statement coverage	Coverage testing	Badge, leaderboard, punti esperienza, livelli, challenge, combat system, grafiche, musiche
Is it worth using gamification on software testing education? An experience report	Education; gamification; game elements; software testing; quality assurance; punti esperienza	Functional testing	Badge, leaderboard, punti esperienza, reward, avatar, forum
Towards a gamified tool to improve unit test teaching	Unit testing; cs education; gamification	Unit testing	Reward, challenge, monete
A Systematic Strategy to Teaching of Exploratory Testing using Gamification	Gamification; learning; ad-hoc testing; teaching; education; exploratory testing	Exploratory testing	Reward, monete, avatar, challenge, livelli, leaderboard
Secret ninja testing with HALO software engineering	Software Testing Education; Social Testing; Serious Games	Concetti base di testing	Quest, challenge, leaderboard, monete, avatar, storyline

Tabella 4.1: Elementi di gamification

tari, suddivisi in due sottogruppi. Il primo sottogruppo è stato utilizzato come gruppo di controllo e composto da 10 studenti, i quali hanno utilizzato una versione del tool senza gli elementi di gamification. Il secondo sottogruppo è stato utilizzato come gruppo sperimentale, composto dagli altri 10 studenti, che hanno utilizzato la versione completa del tool, comprensiva degli elementi di gamification. Dopo l'esperimento, è stato osservato un aumento delle performance del gruppo sperimentale rispetto al gruppo di controllo. In particolare, i partecipanti al gruppo sperimentale hanno impiegato meno tentativi per superare le sfide e i livelli proposti, aumentando così l'efficacia dei test. Inoltre, il 90% dei partecipanti ha valutato positivamente lo strumento di gamification utilizzato nel corso dell'esperimento.

- Lo studio condotto da Jesus et al. ha utilizzato una piattaforma online che utilizzava elementi di gamification per insegnare ai partecipanti, composti da un totale di 53 studenti universitari, i concetti base del testing funzionale. I partecipanti sono stati divisi in tre gruppi: il gruppo A, composto da 13 partecipanti, il gruppo B, composto da 18 partecipanti, e il gruppo C, un gruppo di controllo composto da 22 partecipanti. I risultati dello studio hanno mostrato una maggiore motivazione nei gruppi A e B rispetto al gruppo C, con livelli superiori di interesse ed intrattenimento nei partecipanti dei gruppi sperimentali. Tuttavia, lo studio ha riferito che le performance del gruppo di controllo sono state superiori rispetto ad entrambi i gruppi sperimentali. In sintesi, l'utilizzo della gamification ha dimostrato di avere un impatto positivo sulla motivazione degli studenti, ma non ha prodotto una significativa differenza nelle prestazioni in termini di apprendimento.

Risposta alla domanda RQ2 ed osservazioni

Dai dati raccolti dai due studi esaminati, emergono risultati divergenti sull'efficacia delle tecniche di gamification nell'insegnamento del software testing. Sebbene entrambi gli studi abbiano rilevato un aumento della motivazione nei partecipanti ai gruppi sperimentali, solo uno di essi ha dimostrato una differenza significativa nelle prestazioni in termini di apprendimento. In particolare, l'utilizzo della gamification ha dimostrato di essere efficace nel migliorare le prestazioni dei partecipanti nel superamento delle sfide proposte ma non ha avuto un impatto significativo sulle loro capacità di apprendimento a lungo termine. Tuttavia, la maggior parte degli studi analizzati presenta una minaccia alla validità comune: il numero di partecipanti allo studio basso (esempio, 25 o 30 partecipanti). Per tale motivo, ulteriori studi sono necessari per valutare in modo più approfondito l'efficacia delle tecniche di gamification nell'ambito dell'insegnamento del software testing, utilizzando strategie che coinvolgano un alto numero di partecipanti.

4.4 Serious game nel software testing

RQ₃. Quali sono i serious game maggiormente utilizzati per l'insegnamento del testing?

Il secondo obiettivo più importante della ricerca era stabilire i serious game maggiormente utilizzati per l'insegnamento del software testing. Come per la prima domanda di ricerca, per facilitare la lettura, si sono classificati gli studi in base alle tecniche di testing che si proponevano di insegnare:

Mutation testing. una tecnica di testing software che prevede la modifica controllata del codice sorgente al fine di valutare l'efficacia dei test automatizzati.

Un serious game sull'insegnamento del mutation testing è presentato da Rojas e Fraser. Il serious game presentato nell'articolo è "Code Defenders", che utilizza la tecnica del mutation testing per migliorare la qualità del software e la capacità dei programmatori nello scrivere codice di alta qualità. Il gioco si basa sulla sfida di due giocatori, o due squadre, che assumono rispettivamente i ruoli

di "difensore" ed "attaccante". Lo scopo degli attaccanti è scrivere varianti del programma sotto test, codice mutante. I difensori dovranno scrivere unit test per rilevare il codice mutante degli attaccanti. La fazione degli attaccanti guadagna punti se il codice mutante non viene rilevato dai difensori, viceversa, il punteggio della squadra che difende aumenterà se riuscirà a rilevare il codice mutante con successo. Il gioco, basato a turni, alternati da round di attacchi e difesa, si avvale di un ambiente di sviluppo online. Code Defenders permette ai giocatori di scrivere codice online, eseguire i test e confrontarli con quelli degli altri giocatori. Il gioco incoraggia i giocatori a scrivere codice di alta qualità ed affina la loro abilità nell'identificare errori.

Unit testing, tecnica di testing software che mira a testare le singole unità

Gomes e Lelli propone un serious game chiamato "GAMUT" per l'insegnamento del testing di unità. Il gioco utilizza la narrativa per introdurre i concetti dello unit testing in modo divertente ed interessante. GAMUT è progettato per essere giocato individualmente, il giocatore veste i panni di un soldato in un ambiente medioevale, sconfiggendo mostri tramite diversi tentativi di test. La difficoltà del gioco è data dal numero limitato di test che il giocatore può eseguire per sconfiggere il nemico e dai punti di vita del giocatore che diminuiranno in base agli errori commessi. Il gioco è stato sviluppato inizialmente con un prototipo in forma cartacea, e successivamente in forma digitale tramite Javascript e JQuery.

Il paper di Valle et al. descrive un gioco chiamato "Testing Game" progettato per insegnare il testing funzionale, il testing strutturale e il mutation testing, e che può essere integrato nei corsi di ingegneria del software. Il gioco è web-based e progettato in 2D, con tre livelli rappresentati da porte, corrispondenti alle fasi di testing descritte. Ogni livello offre diverse sfide e può essere accessibile solo ottenendo la chiave del livello precedente. Ci sono otto sfide nel testing funzionale, dieci nel testing strutturale e cinque nel mutation testing. Il gioco fornisce un libro digitale all'inizio di ogni sfida per spiegare i concetti di testing necessari per superarla.

In Elbaum et al. è proposto Bug Hunt, un serious game sviluppato per introdurre in modo accattivante il testing del software nelle prime fasi dell'educazione

informatica. Bug Hunt è un'applicazione web sviluppata in Java e basata sul modello model-view-controller. Per i nuovi giocatori Bug Hunt presenta inizialmente un tutorial che permette di familiarizzare con l'interfaccia dell'applicazione che presenta i concetti fondamentali del testing. Il gioco consiste nel trovare e nel documentare il maggior numero di bug, incoraggia gli studenti a lavorare in modo collaborativo per all'interno della propria squadra. Ogni bug individuato permetterà al team di mettere a segno punti, il punteggio ottenuto varia in base alla difficoltà di individuazione. Il programma da testare è lo stesso presentato ad ogni squadra, inoltre, viene presentato ai team una lista dei bug noti e presenti all'interno del software. Il vincitore del gioco sarà il team che avrà totalizzato il numero maggiore di bug trovati. Il gioco Bug Hunt è strutturato quindi con l'obiettivo di incoraggiare gli studenti a collaborare, a sviluppare le loro capacità di testing e ad imparare divertendosi.

Risposta alla domanda RQ3 e osservazioni

Dagli studi analizzati emerge che diversi serious game sono stati sviluppati seguendo pattern scelti appositamente per la tecnica di testing che si propone di insegnare. Si nota tuttavia che elementi come teamwork, competizione e classifiche sono presenti quasi in tutti gli studi presi in analisi. Sebbene l'obiettivo finale di questi serious game sia l'insegnamento del testing, non esiste un'applicazione che copra tutte le tecniche descritte in precedenza. Attualmente, si tende a selezionare un argomento specifico e creare un serious game che ruoti intorno ad esso, allo scopo di suscitare l'interesse degli studenti. Questa osservazione suggerisce che l'elaborazione di un serious game completo ed universale per tutte le tecniche di testing potrebbe essere una sfida complessa ma remunerativa in termini di apprendimento degli studenti.

4.5 Efficacia dei Serious Games

RQ4. Quanto è efficace l'utilizzo dei serious game per l'insegnamento del testing?

Dei quattro studi della tabella 4.2 si sono analizzati i risultati presentati da [17, 18, 19].

Nello studio di Gomes e Lelli il gioco GAMUT è stato presentato ad un gruppo di 25 studenti universitari con l'obiettivo di insegnare il testing di unità. Per misurare l'efficacia di GAMUT il gruppo è stato sottoposto ad un questionario, a cui solo 15 studenti hanno risposto. Tutti i soggetti hanno concordato sull'attrattiva e l'intuitività del gioco proposto, la maggioranza degli studenti ha affermato di sentirsi in grado di scrivere test di unità dopo aver giocato a GAMUT. Il serious game proposto è stato valutato dagli studenti come un ottimo approccio per l'introduzione del testing di unità ed una buona alternativa ad un approccio standard. In particolare, si è osservato che il gruppo sperimentale ha dimostrato un maggiore interesse per lo unit testing dopo aver giocato a GAMUT.

Lo studio di Valle et al. ha invitato 15 studenti a giocare e valutare Testing Game. L'86% degli studenti ha valutato positivamente il gioco proposto definendolo facile da utilizzare ed interessante. La valutazione dell'apprendimento è stata anch'essa positiva, la maggioranza degli studenti, ha valutato positivamente Testing Game, definendolo più efficace di altre attività tradizionali e che questo aveva contribuito all'apprendimento del software testing. In seguito, quasi tutti i partecipanti all'esperimento hanno definito il gioco come divertente e motivante, affermando che le sfide proposte all'interno, contribuivano ad aumentare il livello di immersione all'interno di Testing Game.

Oltre 400 studenti hanno utilizzato Bug Hunt di Elbaum et al. Durante lo studio, agli studenti è stato somministrato un questionario anonimo con l'obiettivo di valutare l'efficacia del serious game "Bug Hunt". Il questionario si è focalizzato su tre punti: (1) l'utilità del gioco educativo, (2) la possibilità che il gioco possa sostituire il materiale didattico tradizionale sul testing e (3) i concetti appresi grazie al gioco. Tutte le domande hanno ricevuto risposte positive, la maggioranza degli studenti ha convenuto che "Bug Hunt" ha notevolmente facilitato l'apprendimento del testing del software e dei suoi concetti base. La sezione dei commenti del questionario ha mostrato anche che i partecipanti hanno apprezzato il tutorial all'interno del gioco educativo e hanno dimostrato un forte interesse nell'applicazione di un approccio di apprendimento simile.

Titolo	Serious Game	Tipologia di testing
Code Defenders: A Mutation Testing Game	Code Defenders	Mutation testing
GAMUT: GAME-based learning approach for teaching Unit Testing	GAMUT	Unit testing
Educational games: A contribution to software testing education	Testing game	Testing funzionale, testing strutturale, mutation testing
Bug Hunt: Making Early Software Testing Lessons Engaging and Affordable	Bug Hunt	Concetti base e diverse strategie

Tabella 4.2: Serious game e tecniche di testing**Risposta alla domanda RQ4**

Sulla base dei risultati analizzati, si può affermare che i serious game rappresentano un approccio efficace per l'insegnamento del software testing. In particolare GAMUT [17] è stato valutato positivamente dagli studenti, che hanno dimostrato di aver appreso concetti di unit testing in modo efficace. Risultati simili sono stati raggiunti anche da Testing Game e Bug Hunt [18, 19]. Tuttavia sembra prematuro affermare che i serious game possano essere ottimi sostituti alle attività didattiche tradizionali. Risulta importante sottolineare che i campioni di studenti coinvolti nella valutazione di questi serious game sono limitati in termini numerici. Si potrebbe però considerare l'opzione di integrare i serious game nelle attività didattiche senza sostituirle completamente.

Titolo	Autori	Anno	# Pagine	Venue
Gamification to Aid the Learning of Test Coverage Concepts	Sherif E., Liu A., Nguyen B., Lerner S., Griswold W.G.	2020	5	32nd IEEE Intl. Conference on Software Engineering Education & Training
Is it worth using gamification on software testing education? An experience report	de Jesus G.M., Ferrari F.C., Paschoal L.N., Souza S.R.S.	2019	10	SBQS '19: Proceedings of the XVIII Brazilian Symposium on Software Quality
Towards a gamified tool to improve unit test teaching	Marabesi M., Frango Silveira I.	2019	7	2019 XIV Latin American Conference on Learning Technologies
A systematic strategy to teaching of exploratory testing using gamification	Costa I.E.F., Oliveira S.R.B.	2019	7	ENASE 2019 - Proceedings of the 14th International Conference on Evaluation of Novel Approaches to Software Engineering
Code Defenders: A Mutation Testing Game	Rojas J.M., Fraser G.	2016	6	2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)

Tabella 4.3: Dataset 1

Titolo	Autori	Anno	# Pagine	Venue
Secret ninja testing with HALO software engineering	Robles G., González-Barahona J.M., Moral A.	2011	4	Proceedings of the 4th international workshop on Social software engineering
GAMUT: GAME-based learning approach for teaching Unit Testing	Renata Faria Gomes, Valéria Lelli	2021	11	ACM International Conference Proceeding Series, 27
Educational games: A contribution to software testing education	Pedro Henrique Dias Valle; Armando Maciel Toda; Ellen Francine Barbosa; José Carlos Maldonado	2017	8	2017 IEEE Frontiers in education Conference (FIE)
Bug Hunt: Making Early Software Testing Lessons Engaging and Affordable	Sebastian Elbaum, Suzette Person, Jon Dokulil, Matt Jorde	2007	10	29th International Conference on Software Engineering (ICSE'07)

Tabella 4.4: Dataset 2

Minacce alla validità

Le minacce alla validità di una revisione sistematica della letteratura sono tipicamente legate all'imprecisione dell'estrazione dei dati ed alla presenza incompleta di studi a causa di limitazioni dei termini di ricerca, alla selezione dei motori di ricerca accademici ed alla presenza di un bias del ricercatore riguardo i criteri di inclusione/esclusione. In questa sezione si sono identificate ed affrontate attentamente le potenziali minacce alla validità [20, 21] dello studio adottando misure per mitigarle o minimizzarle.

Validità interna. Le minacce in questa categoria riguardano l'approccio di selezione delle fonti e dei dati, nonché la possibilità che i risultati siano influenzati da fattori causali che il ricercatore non conosce [20]. Si è applicato un approccio sistematico precedentemente proposto in letteratura da Kitchenham et al. e descritto nella Sezione 3. Sono stati definiti e riportati con attenzione tutti i passaggi del processo, ovvero i motori di ricerca, i termini di ricerca e i criteri di inclusione/esclusione, per garantire la replicabilità. La presente ricerca è stata effettuata utilizzando parole chiave definite, seguite da una ricerca manuale nelle referenze del pool iniziale del campo studiato. Per quanto riguarda i motori di ricerca, sono stati inclusi database accademici ufficiali e completi: Scopus, IEEE Xplore ed ACM Digital library.

Validità di costruito. Le minacce in questa categoria riguardano fino a che punto l'oggetto di studio rappresenta veramente la teoria alla base dello studio e sono principalmente dovute all'imprecisione delle misurazioni effettuate [20]. Una minaccia in questa categoria riguarda la mancanza di prove empiriche in alcuni degli studi analizzati.

Validità delle conclusioni. Le minacce in questa categoria riguardano il grado di credibilità delle conclusioni a cui si arriva[20]. Per ridurre le probabilità di errore, si è adottato un approccio sistematico e organizzato definito da Kitchenham et al. e sono state seguite le linee guida di Keele et al. Si è inoltre seguita una rigorosa valutazione dei criteri di qualità, e si sono estratte le fonti dai database più utilizzati e conosciuti.

Validità esterna. Le minacce in questa categoria riguardano il grado in cui i risultati di questo studio possono essere generalizzati [20]. Innanzitutto, la presente revisione sistematica della letteratura è stata condotta sui database scientifici più conosciuti, per aumentare la qualità delle fonti e la generalizzabilità dei risultati. Inoltre, si sono estratti studi solo in lingua inglese, escludendo quindi studi scritti in altre lingue, mettendo a rischio la generalizzabilità. Tuttavia, (i) le principali venue di ingegneria del software accettano solo articoli scritti in inglese e (ii) la maggior parte delle revisioni sistematiche della letteratura già presenti in letteratura includono criteri di esclusione basati sulla lingua inglese.

CAPITOLO 6

Conclusioni

L'obiettivo della ricerca era effettuare una system literature review per fornire una panoramica delle tecniche di gamification e dei serious game maggiormente utilizzati per l'insegnamento del software testing, cercando di misurare anche l'efficacia di questi.

Per ciò che concerne l'applicazione di gamification all'insegnamento di software testing, si può notare dagli articoli analizzati che le tecniche più adottate sono simili a quelle già presenti in altre attività didattiche. Oltre ciò, riguardo l'uso di serious game, gli studi revisionati hanno mostrato che l'utilizzo di tale approccio risulta essere apprezzato dagli studenti, oltre ad avere un impatto positivo sul loro apprendimento. Tuttavia, dagli studi individuati e analizzati sorge una minaccia alla generalizzabilità (comune sia a gamification che serious game) dei risultati ottenuti: il numero di partecipanti alle sperimentazioni condotte risulta essere inferiore a studi simili in diversi ambiti. In tal senso, ulteriori studi dovrebbero concentrarsi su campioni di numero maggiore, al fine di generalizzare i già molti risultati ottenuti dallo stato dell'arte.

Sviluppi futuri potrebbero includere:

- Esplorare l'efficacia di gamification e serious game su studenti con differenti background culturali

- Sviluppare nuove tecniche di gamification e serious game per il software testing e testarle in diversi contesti didattici
- Sviluppare un serious game che possa integrare tutte le tecniche di testing moderne
- Testare le tecniche di gamification ed i serious game già presenti in questo campo su un campione più grande.

La presente ricerca potrebbe avere un impatto significativo sia sulla comunità accademica e sui ricercatori di serious game e gamification che sugli insegnanti. I risultati ottenuti suggeriscono che l'uso di queste tecniche può essere efficace per l'insegnamento del testing, ma ci sono ancora alcune aree che potrebbero essere approfondite. Per esempio, potrebbe essere utile esplorare le differenze tra i risultati ottenuti in gruppi di studenti con diversi livelli di conoscenza ed esperienza. Inoltre, si potrebbe esplorare l'utilizzo di questi serious game su studenti con differenti background culturali. Infine, i risultati della presente ricerca potrebbero essere utilizzati per sviluppare nuove tecniche di gamification e serious game. Oltre ciò, i docenti potrebbero utilizzare i risultati ottenuti come compendio per modellare e pianificare lezioni innovative per supportare l'apprendimento degli studenti.

Bibliografia

- [1] Bernd Bruegge e Allen H Dutoit. "Object-oriented software engineering. using uml, patterns, and java". In: *Learning* 5.6 (2009), p. 7.
- [2] Karl M Kapp. *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.
- [3] Mourya Reddy Narasareddy Gari, Gursimran Singh Walia e Alex David Radermacher. "Gamification in computer science education: A systematic literature review". In: *2018 ASEE Annual Conference & Exposition*. 2018.
- [4] Tarja Susi, Mikael Johannesson e Per Backlund. "Serious games: An overview". In: (2007).
- [5] Sofia Ouhbi e Nuno Pombo. "Software engineering education: Challenges and perspectives". In: *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE. 2020, pp. 202–209.
- [6] Mehdi Jazayeri. "The education of a software engineer". In: *Proceedings. 19th International Conference on Automated Software Engineering, 2004*. IEEE. 2004, pp. xviii–xxvii.
- [7] Oscar Pedreira, Félix Garcia, Nieves Brisaboa e Mario Piattini. "Gamification in software engineering–A systematic mapping". In: *Information and software technology* 57 (2015), pp. 157–168.

- [8] Mehmet Kosa, Murat Yilmaz, Rory O'Connor e Paul Clarke. "Software engineering education and games: a systematic literature review". In: *Journal of Universal Computer Science* 22.12 (2016), pp. 1558–1574.
- [9] Per Backlund e Maurice Hendrix. "Educational games-are they worth the effort? A literature survey of the effectiveness of serious games". In: *2013 5th international conference on games and virtual worlds for serious applications (VS-GAMES)*. IEEE. 2013, pp. 1–8.
- [10] Jacopo Gambardella. *Online Appendix*. 2023. URL: https://drive.google.com/drive/folders/11W4_yx1zCSQ4KUP1VSLrMDQhAlhdnLFT?usp=share_link/edit?usp=share_link.
- [11] Eman Sherif, Andy Liu, Brian Nguyen, Sorin Lerner e William G Griswold. "Gamification to aid the learning of test coverage concepts". In: *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*. IEEE. 2020, pp. 1–5.
- [12] Gabriela Martins de Jesus, Leo Natan Paschoal, Fabiano Cutigi Ferrari e Simone RS Souza. "Is it worth using gamification on software testing education? an experience report". In: *Proceedings of the XVIII Brazilian Symposium on Software Quality*. 2019, pp. 178–187.
- [13] Matheus Marabesi e Ismar Frango Silveira. "Towards a gamified tool to improve unit test teaching". In: *2019 XIV Latin American Conference on Learning Technologies (LACLO)*. IEEE. 2019, pp. 12–19.
- [14] Igor Ernesto Ferreira Costa e Sandro Ronaldo Bezerra Oliveira. "A Systematic Strategy to Teaching of Exploratory Testing using Gamification." In: *ENASE*. 2019, pp. 307–314.
- [15] Jonathan Bell, Swapneel Sheth e Gail Kaiser. "Secret ninja testing with HALO software engineering". In: *Proceedings of the 4th international workshop on Social software engineering*. 2011, pp. 43–47.
- [16] José Miguel Rojas e Gordon Fraser. "Code defenders: a mutation testing game". In: *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE. 2016, pp. 162–167.

- [17] Renata Faria Gomes e Valéria Lelli. "GAMUT: GAME-based learning approach for teaching Unit Testing". In: *XX Brazilian Symposium on Software Quality*. 2021, pp. 1–11.
- [18] Pedro Henrique Dias Valle, Armando Maciel Toda, Ellen Francine Barbosa e José Carlos Maldonado. "Educational games: A contribution to software testing education". In: *2017 IEEE Frontiers in education Conference (FIE)*. IEEE. 2017, pp. 1–8.
- [19] Sebastian Elbaum, Suzette Person, Jon Dokulil e Matt Jorde. "Bug hunt: Making early software testing lessons engaging and affordable". In: *29th International Conference on Software Engineering (ICSE'07)*. IEEE. 2007, pp. 688–697.
- [20] Claes Wohlin, Per Runeson, Martin H"ost, Magnus C Ohlsson, Bj"orn Regnell e Anders Wessl'en. *Experimentation in software engineering*. Springer Science Business Media, 2012.
- [21] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey e Stephen Linkman. "Systematic literature reviews in software engineering—a systematic literature review". In: *Information and software technology* 51.1 (2009), pp. 7–15.

Ringraziamenti

Vorrei ringraziare il professore Palomba, relatore di questa tesi di laurea, per avermi dato l'opportunità di lavorare a questo progetto di ricerca e per la sua gentilezza e disponibilità costante. Un ringraziamento speciale va al Dott. Stefano Lambiase, correlatore di questa tesi, per avermi sempre aiutato con le sue osservazioni ed i suoi consigli a migliorare il mio lavoro. Grazie al suo sostegno ed alla sua supervisione attenta, ho potuto affrontare e superare le difficoltà incontrate lungo il percorso.

Ringrazio i miei genitori. Ringrazio mio padre Antonio, che nonostante tutte le difficoltà ha fatto sempre in modo di essere presente per me e mi ha fatto capire che nessun ostacolo è insormontabile con il duro lavoro e la determinazione. Ringrazio mia madre Ofelia, che mi è sempre rimasta accanto e che non ha mai smesso di supportarmi su qualunque cosa facessi, la ringrazio per avermi sempre incoraggiato ad inseguire i miei sogni e a non mollare mai. Un ringraziamento speciale va a mia zia Tiziana, per avermi aiutato a rialzarmi in un momento difficile, e per avermi mostrato gentilezza, forza ed energia ineguagliabili.

Infine, desidero ringraziare i miei amici Viviana, Davide, Carmine, Vincenzo e Stefano, per il loro sostegno e la loro amicizia. Grazie a voi, ho avuto l'opportunità di imparare cose nuove, di superare le difficoltà e di affrontare le sfide con maggiore determinazione. Siete stati sempre presenti nei momenti di difficoltà, pronti ad ascol-

tare e a incoraggiare. Vi ringrazio per aver vissuto con me momenti indimenticabili ed aver condiviso con me risate, delusioni e successi.

Grazie a tutti voi.