



**Politecnico
di Torino**

Politecnico di Torino

Corso di Laurea Magistrale in INGEGNERIA DEL CINEMA E DEI
MEZZI DI COMUNICAZIONE

A.a. 2024/2025

Sessione di Laurea Aprile 2025

**Ruolo e impatto dell'automazione nei processi di
Quality Assurance nello sviluppo dei videogiochi**

Relatori:

Riccardo Coppola
Francesco Strada

Candidato:

Gabriele Massimiani

Abstract

L'automazione sta trasformando i processi di Quality Assurance (QA) nell'industria videoludica, influenzando il bilanciamento tra testing manuale e automatizzato. Questa tesi esplora il ruolo e l'impatto dell'automazione nel QA dei videogiochi, analizzando le modalità con cui le aziende scelgono tra test manuali e automatizzati, il livello di integrazione del QA nei processi di sviluppo e le principali sfide legate all'automazione e all'intelligenza artificiale (AI). Attraverso un'analisi basata su dati raccolti tramite un questionario rivolto a professionisti del settore, la ricerca evidenzia i fattori chiave che determinano l'adozione di tecniche automatizzate, il livello di collaborazione tra il QA e gli altri reparti, nonché le difficoltà percepite nell'introduzione di nuove tecnologie. I risultati mostrano una preferenza per un approccio ibrido, in cui l'automazione viene impiegata per test ripetitivi e di regressione, mentre il testing manuale rimane fondamentale per valutare l'esperienza utente e le dinamiche di gameplay. Oltre agli aspetti tecnici, la ricerca mette in luce anche le condizioni lavorative nel settore del QA, spesso caratterizzate da una scarsa valorizzazione del ruolo, turni di lavoro intensivi e una percezione del QA come posizione entry-level piuttosto che come una carriera strutturata. Questo porta a una costante rotazione del personale e a difficoltà nel trattenere talenti esperti, con ripercussioni sulla qualità del prodotto finale. In risposta alle criticità emerse, si propone un processo di testing che integri Embedded QA, CI/CD (Continuous Integration/Continuous Deployment) e strumenti avanzati di automazione, ottimizzando i flussi di lavoro senza sostituire il team di QA. Le conclusioni offrono una riflessione sulle possibili evoluzioni del QA nel settore videoludico, sottolineando la necessità di un equilibrio tra innovazione tecnologica e il valore insostituibile dell'intervento umano. La crescente integrazione dell'automazione, se accompagnata da un riconoscimento adeguato del ruolo del QA, potrebbe non solo migliorare la qualità del prodotto finale, ma anche contribuire a una maggiore professionalizzazione del settore, favorendo condizioni di lavoro più sostenibili e percorsi di carriera più strutturati per i professionisti del QA.

Indice

1 Introduzione	4
2 Quality Assurance e Automazione nei Videogiochi	6
2.1 Il ruolo del QA nei videogiochi	6
2.2 Metodologie di test nei videogiochi	10
2.3 Approcci di testing	12
2.4 Tool per l'automazione	16
2.5 Sfide Uniche del Testing dei Videogiochi	17
2.6 Progressi nel Testing Automatico dei Videogiochi	18
2.7 Embedded QA	21
3 Metodi	24
3.1 Struttura del questionario	24
3.2 Reclutamento dei Partecipanti	25
3.3 Metodologie	26
4 Risultati	32
4.1 Informazioni sugli intervistati	32
4.2 Risultati RQ1	37
4.3 Risultati RQ2	48
4.4 Risultati RQ3	61
4.5 Sintesi	73
5 Discussioni	77
5.1 Automazione E Manuale	77
5.2 Comunicazione e Integrazione	80
5.3 Sfide e AI	83
5.4 Strategie per un processo automatizzato	86
6 Conclusioni	92
Bibliografia e Sitografia	94

Appendice A Questionario	98
A.1 Domande demografiche	98
A.2 Confronto tra QA automatizzata e manuale	100
A.3 Automazione e collaborazione nel QA	103
A.4 Sfide e AI nel QA	104
A.5 Suggerimenti	105
Appendice B Grafici	106
B.1 Grafici Demografici	106
B.2 Grafici RQ1	112
B.3 Grafici RQ2	118
B.4 Grafici RQ3	133

Introduzione

Negli ultimi decenni, l'industria dei videogiochi ha conosciuto una crescita esponenziale, affermandosi come uno dei settori più redditizi e innovativi dell'intrattenimento digitale. Grazie ai progressi tecnologici, i videogiochi sono diventati sempre più complessi, integrando grafica avanzata, intelligenza artificiale sofisticata e meccaniche di gioco sempre più immersive. Tuttavia, questa evoluzione ha reso sempre più articolati i processi di sviluppo e testing, ponendo nuove sfide nella garanzia della qualità del prodotto finale. Negli ultimi anni, però, questa crescita ha subito un rallentamento a causa di diversi fattori, tra cui investimenti non ottimali [14] e la difficoltà nel recuperare i costi di sviluppo dei progetti AAA [8]. Tali circostanze hanno determinato una serie di licenziamenti su larga scala e la chiusura di interi studi di sviluppo, generando ulteriori difficoltà produttive e contribuendo a un impoverimento artistico del prodotto videoludico [13].

La Quality Assurance riveste un ruolo fondamentale nello sviluppo dei videogiochi, garantendo che il software sia stabile, giocabile e privo di difetti critici. A differenza di altri settori del software, il testing nei videogiochi non si limita a verificare il corretto funzionamento delle funzionalità, ma deve anche assicurare un'esperienza di gioco coinvolgente e priva di frustrazioni per l'utente finale. Questo implica una combinazione di testing manuale, per valutare elementi soggettivi come il bilanciamento e il divertimento, e testing automatizzato, per eseguire controlli ripetitivi e complessi con maggiore efficienza.

L'automazione dei processi di QA sta assumendo un'importanza crescente nell'industria videoludica, poiché consente di ridurre i tempi di sviluppo e migliorare la qualità del prodotto finale. Tecniche avanzate come il testing automatizzato, l'utilizzo di bot basati su intelligenza artificiale e l'adozione di pipeline CI/CD permettono di identificare e correggere bug in modo più rapido ed efficace. Tuttavia, l'implementazione dell'automazione nel QA dei videogiochi non è priva di sfide, tra cui la necessità di standardizzazione, la difficoltà di testare esperienze non deterministiche, la resistenza al cambiamento da parte degli sviluppatori e la loro lacuna nelle competenze.

Questa tesi ha l'obiettivo di esplorare il ruolo e l'impatto dell'automazione nei processi di QA nello sviluppo dei videogiochi. In particolare, la ricerca si propone

di rispondere alle seguenti domande:

- RQ1: In che modo le aziende di sviluppo videoludico scelgono tra testing manuale e automatizzato, e quali sono i principali fattori che influenzano questa decisione?
- RQ2: In che modo il team di Quality Assurance integra l'automazione nei processi di sviluppo del gioco e collabora con altri reparti per ottimizzare la qualità del prodotto finale?
- RQ3: Quali sono le principali sfide affrontate dai team di Quality Assurance nei processi di automazione, e come percepiscono l'evoluzione del loro ruolo in relazione all'introduzione di tecnologie come l'intelligenza artificiale nel settore dei videogiochi?

La struttura della tesi è articolata in sei capitoli. Dopo questa introduzione, il Capitolo 2 (Quality Assurance e Automazione nei Videogiochi) fornisce una panoramica sul ruolo del QA nei videogiochi, le metodologie di testing adottate e le sfide specifiche del settore. Il Capitolo 3 (Metodi) descrive l'approccio adottato per la raccolta e l'analisi dei dati, con un focus particolare sulla progettazione del questionario e sulle metodologie di analisi quantitativa e qualitativa utilizzate. Il Capitolo 4 (Risultati) presenta i dati raccolti e le analisi effettuate, suddivise in base alle tre domande di ricerca. Il Capitolo 5 (Discussioni) interpreta i risultati, mettendoli in relazione con la letteratura esistente e delineando possibili implicazioni pratiche. Infine, il Capitolo 6 (Conclusioni) riassume i principali risultati della ricerca e suggerisce possibili sviluppi futuri per l'integrazione dell'automazione nei processi di QA nel settore videoludico.

Quality Assurance e Automazione nei Videogiochi

Il ruolo del QA nei videogiochi

La Quality Assurance è un insieme complesso di metodi utilizzati in tutte le fasi dello sviluppo del software, dall'ingegneria dei requisiti e dalla progettazione del software alla codifica e al testing [26]. Nel contesto dello sviluppo dei videogiochi, il team di QA è essenziale per garantire che il gioco finale sia stabile, giocabile e divertente. Mentre i test tradizionali del software si concentrano su come il software esegue determinate funzioni, i test sui videogiochi riguardano anche l'esperienza del giocatore, inclusi aspetti come divertimento, equilibrio e usabilità [33].

I tester del team di QA hanno il compito di identificare e segnalare bug, problemi di usabilità e altri difetti che potrebbero compromettere l'esperienza di gioco. Il loro lavoro non si limita semplicemente a testare il gioco in modo superficiale, ma richiede un approccio metodico e analitico per verificare in modo approfondito tutti gli aspetti del prodotto [9].

La QA riveste un ruolo centrale nello sviluppo di videogiochi di alta qualità, come quelli classificati AA o AAA. I videogiochi sono prodotti software estremamente complessi che integrano grafica avanzata, meccaniche di gioco sofisticate e supporto per diverse piattaforme hardware. Un team di QA dedicato è indispensabile per individuare e correggere problematiche che potrebbero influenzare negativamente l'esperienza del giocatore, quali bug, glitch, problemi di prestazioni o interfacce utente poco intuitive.

Nel contesto dello sviluppo di giochi Indie, invece, il ruolo della QA assume una dimensione ancora più critica a causa delle risorse spesso limitate. Gli sviluppatori indipendenti, a differenza delle grandi case di produzione, raramente dispongono di team di QA dedicati e si affidano al contributo di programmatori, designer o addirittura di conoscenti per testare i propri giochi. Questo approccio può portare a una minore attenzione ai processi di QA e a un numero maggiore di bug nel prodotto finale [5]. Tuttavia, disponendo di un budget più ampio, è possibile esternalizzare il QA, affidandosi a aziende specializzate o a una divisione QA fornita

dal publisher con cui lo studio indie ha stretto accordi.

Nel settore videoludico, i giochi AAA e Indie rappresentano due estremi interconnessi. I giochi AAA, caratterizzati da budget elevati e team numerosi, sono veri e propri blockbuster supportati da grandi editori come Xbox e Playstation. Questi titoli puntano su tecnologie avanzate e un ampio pubblico, garantendo un'esperienza tecnica di alto livello, ma con costi di sviluppo e marketing che solo pochi possono sostenere, come nel caso di *Cyberpunk 2077* [36].

I giochi Indie, invece, sono sviluppati da piccoli team o singoli individui con risorse limitate ma maggiore libertà creativa, che consente loro di esplorare idee innovative spesso ignorate dai giochi AAA. Titoli come *Celeste* [28] e *Hollow Knight* [4] dimostrano il loro impatto creativo.

Tra queste due categorie si collocano altre classificazioni: i Triple-I, titoli indipendenti che offrono una qualità paragonabile agli AAA, e i Double-A, prodotti da studi di medie dimensioni con budget intermedi [2].

Queste classificazioni servono principalmente a semplificare la categorizzazione dei giochi per il pubblico generale. Tuttavia, spesso alimentano l'errata convinzione che i giochi AAA siano intrinsecamente migliori rispetto ai titoli indie. Questa visione, oltre a essere troppo riduttiva, ignora il valore creativo e l'innovazione che i giochi indie portano al panorama videoludico.

Indipendentemente dalla categoria del gioco, i team di QA si occupano di garantire e monitorare la qualità del prodotto lungo l'intero ciclo di sviluppo, dalla pre-produzione fino al supporto post-lancio, contribuendo in modo decisivo alla riuscita del progetto. Durante la pre-produzione, analizzano documenti di progettazione, anticipano problemi di usabilità e offrono feedback sull'esperienza prevista. In produzione, eseguono test approfonditi su funzionalità, prestazioni, grafica e audio, utilizzando metodi manuali e automatizzati. Inoltre, mantiene una comunicazione costante con il resto del team di sviluppo per segnalare e risolvere eventuali problemi. Dopo il lancio, continuano a monitorare il gioco, raccogliendo feedback dai giocatori, identificando problemi residui e collaborando con gli sviluppatori per pubblicare patch e aggiornamenti, garantendo così un'esperienza ottimale anche dopo il rilascio.

L'adozione di un processo di QA strutturato offre numerosi vantaggi per gli sviluppatori di videogiochi. Permette di ridurre i costi, intervenendo sui bug nelle

prime fasi dello sviluppo, quando le correzioni sono meno onerose [24]. Inoltre, migliora la qualità complessiva del gioco, garantendo un'esperienza fluida e priva di errori [5]. Un gioco ben testato aumenta la soddisfazione dei giocatori e rafforza la reputazione degli sviluppatori, consolidando la loro credibilità nel settore [11].

Questa è solo una delle possibili soluzioni, forse la più bilanciata, ma non sempre rispecchia la realtà. Soprattutto nelle grandi aziende, il team di QA è spesso il più trascurato, sia dal punto di vista economico che psicologico. Inoltre, le donne nel settore devono affrontare frequenti episodi di discriminazione e molestie. I tester di QA, nella maggior parte dei casi, ricevono stipendi bassi, affrontano turni di lavoro estenuanti e hanno contratti temporanei, con poche possibilità di stabilizzazione, poiché vengono spesso lasciati a casa alla fine del progetto [40].

Nel contesto di un team di Quality Assurance per lo sviluppo di videogiochi, esistono diversi ruoli chiave che contribuiscono al buon esito del processo di testing. Il **QA Lead** o **QA Manager** svolge un ruolo centrale, essendo responsabile della gestione complessiva delle attività di testing. Il suo compito è coordinare il team, definire le strategie di testing, assegnare i compiti e monitorare i progressi, assicurandosi che ogni fase del processo venga rispettata e che vengano raggiunti gli obiettivi prefissati. Inoltre, il QA Lead si occupa di garantire la qualità complessiva dei test, supervisionando la pianificazione e l'esecuzione delle attività di verifica, e ottimizzando continuamente le metodologie e le risorse impiegate [11]. Il QA Manager non è un elemento isolato, ma un gestore di reparto che interagisce con altri manager e lead di diversi settori. Il suo ruolo combina competenze organizzative e operative, richiedendo una profonda conoscenza del prodotto sotto ogni aspetto, dal game design alla narrativa. L'esperienza pregressa come tester e una solida base di conoscenze nei vari ambiti dello sviluppo videoludico sono essenziali per svolgere il lavoro con efficacia. La gestione del team varia in base alle dimensioni dell'azienda e alla complessità del progetto, ma è fondamentale evitare il micromanagement, privilegiando un'organizzazione strutturata dei task tramite strumenti specifici. Il lavoro di QA include la pianificazione predittiva dei test, considerando non solo il percorso principale del gioco, ma anche scenari imprevedibili che potrebbero comprometterne la stabilità. Il processo si articola in test plan e test run, con i risultati documentati attraverso bug ticket che vengono assegnati ai reparti competenti per la risoluzione. Il QA Manager funge inoltre da ponte tra lo studio di sviluppo e il publisher, coordinando le attività di testing e garantendo la condivisione della documentazione necessaria per il rilascio del prodotto sul mercato [1].

Il **QA Tester** è il professionista incaricato di testare il gioco per identificare e segnalare bug, glitch o problemi che potrebbero influire sull'esperienza dell'utente. Il loro compito principale è documentare con precisione i problemi rilevati, fornendo informazioni dettagliate per consentire ai programmatori di riprodurre e risolvere i difetti. Inoltre, i QA Tester possono eseguire test specifici, come quelli legati alle prestazioni o alla localizzazione, per assicurarsi che il gioco funzioni correttamente su diverse piattaforme e sia adatto a diverse lingue e culture. Questo ruolo richiede attenzione ai dettagli, metodo e pazienza, oltre alla capacità di simulare scenari complessi per individuare potenziali problemi nascosti durante lo sviluppo [9].

Il **QA Analyst**, invece, si occupa di un'analisi più approfondita del processo di sviluppo del gioco, utilizzando la conoscenza del game design e della programmazione per individuare potenziali problematiche prima che si trasformino in veri e propri bug. Collabora strettamente con sviluppatori, designer e produttori per garantire che il gioco rispetti gli standard di qualità e soddisfi i requisiti di prestazioni. Il QA Analyst può anche contribuire alla progettazione dei test, fornendo indicazioni strategiche su quali aspetti del gioco necessitano di maggiore attenzione e sviluppando piani per ottimizzare il processo di QA.

Il **QA Engineer** ha il compito di ampliare la mentalità della qualità all'interno del team e di fornire strumenti e ambienti che permettano a sviluppatori, game designer, grafici e altri professionisti di testare il prodotto in modo rapido ed efficiente. Combinando competenze di sviluppo software e testing, questo ruolo si occupa della creazione di strumenti personalizzati, framework di automazione e script per l'esecuzione di test automatizzati, riducendo l'errore umano e garantendo una copertura più ampia. L'automazione consente di gestire volumi elevati di test in tempi ridotti, permettendo al team di QA di concentrarsi su aspetti più complessi e contribuendo a ottimizzare il processo di verifica e a velocizzare il rilascio del prodotto [10].

Questi ruoli, pur essendo distinti, lavorano in stretta collaborazione per garantire che il gioco sviluppato sia privo di bug e soddisfi gli elevati standard di qualità richiesti nel mercato. Inoltre, la struttura dei ruoli può variare tra i diversi team, adattandosi alla dimensione del gruppo e alla tipologia di progetto in sviluppo. Vi sono, inoltre, sottoruoli specifici legati al QA, spesso altamente specializzati e presenti solo in determinate aziende, come ad esempio i tester nel reparto di *Localization Quality Assurance (LQA)*. Altri ruoli specifici possono essere *Animation QA*, *VFX QA*, *Narrative QA*, *Audio QA* [23].

Metodologie di test nei videogiochi

Nel caso dei videogiochi, la tradizionale piramide dei test, caratterizzata da una base larga e una punta stretta, viene messa in discussione. Solitamente alla base si colloca lo Unit Testing, seguito dall'Integration Testing e, al vertice, dall'End-To-End o System Testing.

- **Unit Testing:** verifica il corretto funzionamento di singole unità di codice, come classi o funzioni, in isolamento.
- **Integration Testing:** testa l'interazione tra unità di codice, verificando che lavorino correttamente insieme.
- **System Testing/End-To-End Testing:** valuta il sistema software completo (SUT) simulando l'esperienza dell'utente finale, garantendo il corretto funzionamento globale.

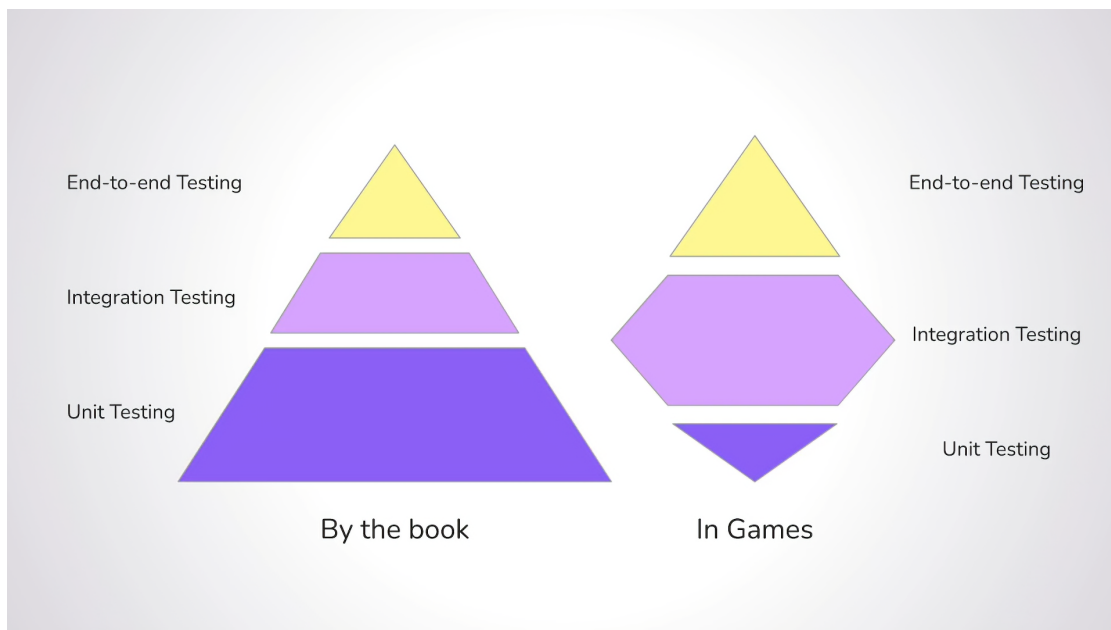


Figura 2.1: Fonte: Slide tratta dalla live su LinkedIn di Romain Gauthier, The challenges of test automation in video games, 5 nov 2024

Nei videogiochi, data la loro natura di sistemi altamente interconnessi, come il sistema dell'inventario, che può essere extra-diegetico o diegetico (ad esempio in *Death Stranding* [31]), l'interdipendenza tra i componenti è particolarmente evidente. Un inventario diegetico, infatti, genera numerosi collegamenti con altri

elementi del gioco. Per questo motivo, l'Integration Testing assume un ruolo centrale e predominante, assumendo una forma romboide più ampia rispetto allo Unit Testing, che risulta invece ridotto (Fig. 2.1) [16].

Esistono diversi tipi di test, ognuno progettato per raggiungere obiettivi specifici attraverso metodi distinti:

- **Functional Testing:** che verificano la correttezza delle funzionalità del gioco, come le meccaniche di gioco, l'interfaccia utente e l'intelligenza artificiale;
- **UX Testing:** Si concentra sull'esperienza dell'utente, valutando la facilità d'uso, l'intuitività e la soddisfazione del giocatore.
- **Localization Testing:** Verifica la correttezza della traduzione e dell'adattamento culturale del gioco per i diversi mercati.
- **Performance Testing:** Valuta la velocità ed efficienza del gioco, inclusi framerate, tempi di caricamento e utilizzo delle risorse.
- **Compatibility Testing:** Verifica il corretto funzionamento del gioco su diverse configurazioni hardware e software.
- **Regression Testing:** Assicura che le modifiche al codice non abbiano introdotto nuovi bug o riattivato vecchi problemi.
- **Platform Certification & Compliance Testing:** Questo tipo di testing assicura che il gioco sia compatibile con le diverse piattaforme di destinazione, come PC, console e dispositivi mobili, e che rispetti tutte le normative e gli standard di settore. I principali hardware manufacturer, ovvero i licenziatari delle piattaforme console come *Sony*, *Microsoft* e *Nintendo*, impongono requisiti tecnici rigorosi per i titoli pubblicati sulle loro piattaforme. Ad esempio, *Sony* fornisce una *Technical Requirements Checklist (TRC)*, *Microsoft* pubblica i *Xbox Requirements (XR)* e *Nintendo* offre un insieme di linee guida noto come *Lotcheck*.
- **Gameplay Testing:** Si concentra sull'esperienza di gioco, valutando aspetti come il bilanciamento, il divertimento e l'engagement del giocatore.

Oltre a queste categorie principali, esistono moltissimi altri test più o meno specifici come Destructive, Accessibility, Data-driven, Smoke, Security, Recovery, Stability, Error handling, Deployment, Installation, Scalability, Balance, Vulnerability, Fault tolerance, Attack flow, Load, Stress, Volume, ecc [18].

Inoltre, è fondamentale coinvolgere il team di QA fin dalle fasi iniziali dello sviluppo, nonostante ciò rappresenti una sfida per un processo di sviluppo agile. Tuttavia, avere i tester presenti nelle riunioni di design può aiutare a prevenire errori da parte di chi ha una visione complessiva del progetto. Questo argomento sarà approfondito in seguito nella tesi [18].

Approcci di testing

Gli approcci nel testing dei videogiochi sono diversi e vanno scelti in base al tempo a disposizione e agli obiettivi specifici da raggiungere.

Nel *Black Box testing*, il tester analizza il gameplay e l'interazione tra gli elementi del gioco, simulando l'esperienza del videogiocatore e identificando eventuali problematiche, senza accedere al codice sorgente. In questo caso, il tester si concentra, per esempio, su aspetti come l'adattamento di un nuovo veicolo all'ambiente, senza considerare direttamente il funzionamento tecnico sottostante. Il tester è consapevole solo delle funzionalità del gioco, ma non del suo codice interno, e progetta test per verificare se un dato input porta al risultato atteso.

Quando il testing necessita di una maggiore profondità tecnica, si ricorre al *White Box testing*, in cui il tester ha accesso al codice sorgente e una comprensione della struttura logica del software. Con queste informazioni, il tester è in grado di progettare casi di test mirati che esplorano diversi percorsi nel codice, predicendo gli output corrispondenti. Sebbene il *White Box testing* sia comunemente utilizzato a livello di unità, può essere applicato anche durante i test di integrazione e di sistema, analizzando i flussi di codice all'interno di singole unità e le interazioni tra unità e sottosistemi.

Il *Grey Box testing*, ovvero la maggior parte delle situazioni di testing, combina gli approcci *Black Box* e *White Box*: il tester ha una conoscenza parziale del funzionamento interno del software, utilizzando questa consapevolezza insieme alle specifiche di alto livello per progettare test che verificano sia il comportamento esterno che il flusso del codice.

Un altro aspetto importante del testing riguarda l'approccio *Autentico* o *Sintetico*. L'approccio *Autentico* cerca di replicare l'esperienza del giocatore, ad esempio testando un nuovo veicolo seguendo il percorso naturale di scoperta nel gioco, mentre quello *Sintetico* isola l'oggetto da testare, utilizzando codici di debug per forzare situazioni specifiche senza il coinvolgimento emotivo del giocatore. Altri approcci,

come il *Metodico* o *Caotico*, dipendono molto dalla personalità del tester. Il tester *Metodico* analizza i problemi con attenzione, mentre quello *Caotico* tende a testare ogni possibile interazione, generando un numero maggiore di bug e crash. Infine, l'approccio *Oggettivo* o *Soggettivo* si ricollega al *Black Box testing*, in cui l'oggettività riguarda la valutazione di problemi evidenti nel gioco, come la mancanza di un collider, mentre la soggettività si riferisce a aspetti legati all'esperienza di gioco, come la valutazione di un veicolo che, pur avendo una copertura modellata per attutire i danni, nel gioco permette di distruggere completamente il carrarmato, a causa di una errata rappresentazione del veicolo [18].

Oltre a quelle già citate possiamo osservare una differente suddivisione di approcci:

- **Playtesting manuale:** in cui i tester giocano al gioco per identificare bug ed errori, pratica ancora molto diffusa nell'industria dei videogiochi. Questo metodo, spesso considerato il più affidabile per valutare l'esperienza di gioco complessiva, si basa sull'intuizione e sulla conoscenza dei tester per scoprire problemi imprevisi [30]. Questo tipo di testing può essere esteso ai videogiocatori stessi quando il gioco è abbastanza avanzato da permettere la raccolta di feedback costruttivi. Tuttavia, questo approccio ha due facce: da un lato, consente di valutare il divertimento e l'efficacia del gameplay loop, oltre a raccogliere dati su aspetti nascosti al giocatore; dall'altro, se il gioco non è sufficientemente rifinito, potrebbe rischiare di generare cattiva pubblicità a causa di un'esperienza di gioco deludente.
- **Test automatizzati:** Sebbene il playtesting manuale sia fondamentale, l'utilizzo di test automatizzati sta diventando sempre più importante per gestire la crescente complessità dei videogiochi moderni. Il testing automatizzato dei videogiochi si basa sull'impiego di script o framework di test finalizzati alla simulazione delle interazioni dell'utente, tra cui la pressione di pulsanti, i movimenti del mouse e le interazioni con l'interfaccia grafica. Tali script possono essere progettati per svolgere azioni mirate, analizzare specifici scenari o eseguire una sequenza prestabilita di casi di test. Gli strumenti di automazione eseguono gli script e restituiscono un'analisi del comportamento del videogioco, evidenziando eventuali anomalie, malfunzionamenti o difetti. Si rivela particolarmente utile per test necessari ma ripetitivi, o per eseguire scenari complessi difficilmente gestibili manualmente, riducendo così la necessità di intervento umano. Per esempio gli agenti autonomi basati sull'intelligenza artificiale possono esplorare gli ambienti di gioco e testare diverse modalità e situazioni, aumentando l'efficienza del processo di testing e consentendo una copertura più ampia [24]. L'unione con un sistema di capture and replay

ha per esempio permesso a Final Fantasy VII Remake di automatizzare diversi test di controllo qualità. Il sistema registra l'input del controller e lo stato del gioco durante una sessione di gameplay e utilizza questi dati per riprodurre le azioni del giocatore, simulando scenari di gioco. Questa tecnica permette di testare il gioco in modo automatico, individuando bug che potrebbero verificarsi solo in determinate condizioni o dopo un lungo periodo di gioco. Inoltre, l'integrazione di algoritmi di esplorazione consente al sistema di testare aree del gioco al di fuori dei percorsi standard, aumentando ulteriormente la copertura del testing e individuando potenziali problemi in zone meno frequentate dai giocatori. Questo approccio ibrido, che combina il replay di azioni pre-registrate con l'esplorazione autonoma dell'ambiente, permette di raggiungere un livello di copertura e di efficienza del testing difficilmente ottenibile con il solo playtesting manuale [22].

L'automazione comprende numerosi aspetti, tra cui quelli legati alla gestione dei processi di sviluppo. Un esempio significativo è l'integrazione di strumenti come *Jira* (software di task tracking), *GitHub* (piattaforma di version control) e il motore di sviluppo, che consente l'esecuzione automatizzata di operazioni quali commit e push. Questa integrazione permette a tutti i membri del team di monitorare le modifiche in tempo reale: da un lato, figure come il Project Manager, che potrebbero interfacciarsi prevalentemente con *Jira*, e dall'altro sviluppatori e artisti, che potrebbero preferire focalizzarsi direttamente sull'engine e su *GitHub*, senza dover necessariamente interagire con strumenti di gestione del progetto [3]. Oltre alla gestione delle modifiche, si può implementare una pipeline CI/CD per garantire un flusso di sviluppo efficiente e continuo. Ad ogni commit nel repository, strumenti come GitHub Actions, Jenkins o GitLab CI/CD avviano automaticamente la compilazione del codice e una serie di test automatizzati. Se questi test vengono superati, la build viene distribuita in un ambiente di staging per ulteriori verifiche. In caso di Continuous Deployment, il codice valido viene automaticamente rilasciato in produzione, garantendo aggiornamenti frequenti e stabili senza necessitare di interventi manuali. CI/CD è un insieme di pratiche DevOps che automatizzano e ottimizzano il processo di sviluppo software. La Continuous Integration (CI) prevede l'integrazione frequente del codice in un repository centrale, con verifiche automatiche per individuare rapidamente errori e garantire un codice pulito. La Continuous Delivery (CD) estende la CI automatizzando la preparazione del codice per il rilascio, mentre la Continuous Deployment automatizza il rilascio stesso agli utenti finali. Implementare CI/CD riduce errori, velocizza lo sviluppo e migliora la qualità del software, assicurando rilasci frequenti e iterativi [41].

Per quanto riguarda gli approcci, è possibile riportare un esempio pratico di pipeline di testing automatico su *Unreal Engine* adottata dallo studio *People Can Fly (PCF)*, specializzato nello sviluppo di giochi AAA. In questo contesto, *Jenkins* (tool CI/CD che automatizza build, test e deployment nei videogiochi) viene utilizzato per avviare *Gauntlet* [6], che esegue *Unreal Engine* con una riga di comando specifica, analizza i risultati dei test e raccoglie artefatti. I dati generati vengono archiviati in *Grafana Loki* (per i log), *Google Drive* (per i dati di grandi dimensioni) e *PostgreSQL*. I risultati sono resi accessibili tramite *Grafana*, con grafici interattivi e screenshot sincronizzati con i video [43].

Nel corso dell'implementazione del processo di testing, sono emerse alcune considerazioni fondamentali:

- È preferibile creare test semplici, evitando eccessiva complessità;
- L'uso della posta elettronica per l'archiviazione dei dati risulta inefficace; è più opportuno adottare strumenti che garantiscano un accesso più immediato e condiviso;
- Disporre di un sistema interattivo per la visualizzazione dei dati è essenziale, poiché le e-mail non offrono tale funzionalità;
- L'avvio della scrittura dei test in una fase precoce del progetto risulta vantaggioso.

Le soluzioni implementate per ottimizzare il processo sono state le seguenti:

- Sviluppo di un'API personalizzata per l'aggregazione dei dati;
- Adattamento di *Gauntlet* all'infrastruttura esistente;
- Estensione dell'*UE Automation Framework*;
- Utilizzo di *Jenkins* come sistema di integrazione continua per l'esecuzione di *Gauntlet*;
- Archiviazione dei log in *Grafana Loki* e dei dati di grandi dimensioni in *Google Drive*;
- Accesso ai dati tramite un'interfaccia web dedicata.

Tool per l'automazione

Nel contesto dell'automazione dei test nel settore videoludico, la scelta degli strumenti da utilizzare dipende da diversi fattori, tra cui il motore di gioco, il livello di integrazione desiderato e le competenze tecniche del team. Esistono tool nativi, progettati specificamente per un determinato engine, e tool agnostici, che permettono maggiore flessibilità ma richiedono soluzioni di comunicazione esterne per interagire con il gioco [16].

Da un lato, strumenti come *Unity Test Framework* e *Unreal Engine Automation Framework* sono nativi per i rispettivi motori di gioco, il che significa che offrono un'integrazione più diretta e un accesso più profondo alle funzionalità interne dell'engine. Questo consente ai team di sviluppo di scrivere test utilizzando i linguaggi principali dell'ambiente, *C#* per *Unity* e *C++/Blueprints* per *Unreal Engine*, senza la necessità di implementare interfacce di comunicazione aggiuntive. Tuttavia, questa forte integrazione può anche rappresentare una limitazione per studi che lavorano su più motori di gioco o che vogliono un approccio più generalizzato ai test automatizzati.

Dall'altro lato, strumenti come *AltTester* e *GameDriver* sono agnostici, il che significa che non sono legati a un motore specifico. Questa caratteristica li rende particolarmente utili per team che desiderano un approccio più trasversale al testing, magari perché lavorano con più engine o devono garantire compatibilità su diverse piattaforme. Tuttavia, essendo strumenti esterni, questi tool utilizzano interfacce di rete per comunicare con il gioco, il che può introdurre limitazioni o complessità aggiuntive rispetto alle soluzioni integrate direttamente nell'engine. Inoltre, sebbene supportino linguaggi diffusi come *Python* e *C#*, il loro utilizzo potrebbe richiedere configurazioni più complesse rispetto ai framework nativi.

Oltre ai tool specifici per il testing nei motori di gioco, esistono soluzioni più generiche che si integrano nel processo di automazione e sviluppo software in senso più ampio. *Selenium*, ad esempio, è ampiamente utilizzato per testare interfacce web, il che lo rende particolarmente utile per giochi o applicazioni che hanno componenti web-based, come i launcher o i backend per la gestione degli utenti. *Jenkins*, invece, non è un tool di testing in sé, ma è fondamentale per l'integrazione continua (CI/CD), permettendo di eseguire automaticamente i test a ogni modifica del codice. Questo tipo di automazione è cruciale per garantire che ogni iterazione dello sviluppo sia stabile e non introduca nuove regressioni.

La scelta dello strumento più adatto dipende quindi dall'equilibrio tra accessibilità, flessibilità e livello di integrazione richiesto. Se un team lavora esclusivamente su *Unity* o *Unreal*, potrebbe beneficiare di strumenti nativi per una gestione più diretta e meno laboriosa del testing. Al contrario, se il focus è sulla portabilità o sull'uso di più motori di gioco, strumenti agnostici potrebbero essere la soluzione migliore, anche a costo di una configurazione più complessa. Infine, l'integrazione con sistemi di continuous integration, tramite tool come *Jenkins*, può migliorare ulteriormente il flusso di lavoro, garantendo che i test automatici siano una parte integrante dello sviluppo piuttosto che un processo separato.

Sfide Uniche del Testing dei Videogiochi

Il testing dei videogiochi presenta sfide uniche che lo differenziano significativamente dal testing del software tradizionale, rendendolo un processo complesso e articolato. Una delle principali difficoltà risiede nell'assenza di standard consolidati e di metodologie condivise, il che complica la possibilità di confrontare i risultati e valutare l'efficacia dei vari approcci utilizzati. Inoltre, i videogiochi sono esperienze interattive molto più complesse rispetto alle applicazioni tradizionali: devono mantenere i giocatori concentrati e coinvolti attraverso design creativi che introducono elementi come la tensione intenzionale e il fallimento accettato come parte dell'esperienza. Questi principi non sono presenti nei software tradizionali, dove il fallimento viene percepito come un problema da evitare [16].

Un aspetto distintivo del testing nei videogiochi è rappresentato dai comportamenti non deterministici. A differenza del software tradizionale, lo stesso input può generare output differenti a causa di fattori come la simulazione fisica, l'intelligenza artificiale dei nemici e la generazione procedurale di contenuti. Questa caratteristica rende difficoltosa la riproducibilità dei problemi e complica lo sviluppo di test case. Inoltre, l'essenza stessa dei videogiochi – il divertimento e il coinvolgimento – non può essere testata automaticamente. Il divertimento è una qualità soggettiva e difficile da misurare, richiedendo necessariamente l'intervento umano.

La complessità visiva dei videogiochi moderni costituisce un'altra sfida significativa. Grafica tridimensionale, animazioni avanzate ed effetti visivi richiedono strumenti specializzati per catturare e analizzare l'output grafico con precisione. Questa complessità visiva è ulteriormente aggravata dal fatto che, nei videogiochi, i sistemi sono altamente accoppiati e interdipendenti, come ad esempio gli inventari, che interagiscono con altre meccaniche del gioco. Questo accoppiamento rende

difficile isolare le parti del sistema senza influenzare il resto, a differenza dello sviluppo web dove i componenti sono spesso modulari e indipendenti [16].

Un'altra peculiarità del testing nei videogiochi è legata al concetto di tempo. Mentre le applicazioni web raramente includono timer, nei videogiochi il tempo reale e il tempo virtuale (come timer, cooldown e eventi a tempo limitato) sono elementi spesso presenti. Questi aspetti rendono complessa la pianificazione dei test, poiché devono tener conto di condizioni temporali dinamiche e variabili. Inoltre, la frammentazione degli strumenti e l'assenza di standardizzazione nel settore complicano ulteriormente il processo di testing. I principali motori di gioco, come *Unreal Engine*, *Unity* e *Godot*, non offrono strumenti unificati per il testing, obbligando gli sviluppatori a creare soluzioni personalizzate o a fare affidamento su test manuali [29].

Per superare queste sfide, è cruciale decidere con attenzione cosa automatizzare e cosa mantenere manuale. Nonostante i limiti dell'automazione, è evidente che concentrare gli sforzi sui test di integrazione, piuttosto che sui test unitari, può rivelarsi più efficace per individuare bug significativi. Tuttavia, l'industria dei videogiochi è ancora lontana da una soluzione universale: i game engine e le console continuano a presentare incompatibilità, e la maggior parte degli studi, soprattutto quelli di piccole dimensioni, preferisce affidarsi al testing manuale a causa della mancanza di competenze interne per sviluppare test automatizzati [16].

Progressi nel Testing Automatico dei Videogiochi

Nonostante le sfide, negli ultimi anni si sono registrati progressi significativi nel campo del testing automatico dei videogiochi. In particolare, c'è stato un crescente interesse per l'utilizzo di agenti autonomi, basati sull'intelligenza artificiale, per esplorare l'ambiente di gioco e testare varie modalità e situazioni di gioco [29].

Questi agenti possono essere addestrati per giocare al gioco come farebbe un essere umano, fornendo input al gioco e osservandone l'output. Possono essere utilizzati per testare una varietà di aspetti del gioco, tra cui:

- **Esplorazione del mondo di gioco:** gli agenti possono essere addestrati per esplorare in modo sistematico l'ambiente di gioco, verificando la presenza di aree inaccessibili o comportamenti anomali [17].

- **Testing di funzionalità specifiche:** gli agenti possono essere programmati per eseguire azioni specifiche, come completare missioni, interagire con oggetti o combattere contro nemici, verificando il corretto funzionamento delle meccaniche di gioco [19].
- **Valutazione del bilanciamento del gioco:** gli agenti possono simulare diversi stili di gioco e livelli di abilità, fornendo dati utili per valutare il bilanciamento del gioco e l'esperienza utente [5].
- **Testing delle prestazioni:** gli agenti possono essere utilizzati per simulare un elevato numero di giocatori o azioni simultanee, testando la capacità del gioco di gestire carichi di lavoro elevati [5].

L'introduzione di modelli avanzati consente ai bot non solo di esplorare il mondo virtuale alla ricerca di bug, ma anche di simulare il comportamento umano, riproducendo diverse tipologie di giocatori (casuali, esperti, aggressivi, ecc.). Questo aspetto è fondamentale per garantire una copertura di QA più ampia e realistica. Inoltre, i tester devono verificare situazioni poco comuni, come la possibilità di esplorare aree nascoste o interagire con elementi apparentemente insignificanti, attività che i moderni strumenti di IA possono gestire in modo efficace [39].

Il Reinforcement Learning (RL) è una delle tecniche più utilizzate per addestrare i bot, consentendo loro di apprendere attraverso il tentativo ed errore per ottimizzare le loro azioni. L'IA Planning, invece, aiuta a rendere i bot più autonomi, migliorando la loro capacità di pianificare decisioni e sequenze di azioni. Inoltre, l'industria sta sperimentando l'apprendimento non supervisionato, dove i bot vengono addestrati su grandi set di dati senza etichette precise. Questo ha portato alla creazione dei Foundation Models [42], modelli di IA su larga scala che possono adattarsi a diversi giochi e contesti, migliorando ulteriormente l'efficacia del QA [39].

Esistono anche altre tipologie di AI di supporto, come l'AI generativa applicata alla scrittura del codice. Questi strumenti possono suggerire intere linee o blocchi di codice, individuare e correggere automaticamente bug, accelerando così sia la programmazione che il debugging. Sono particolarmente utili per risolvere problemi comuni e ben documentati, ma tendono a essere meno efficaci per compiti più innovativi o specifici.

Differenti applicazioni dell'AI possiamo trovarle nel lavoro di Marini, Gisslén e Sestini [25], di Electronic Arts, che affronta il problema dell'identificazione della

modifica specifica al codice responsabile del fallimento di un test, una sfida particolarmente complessa quando le modifiche provengono da più sviluppatori. Per risolvere questa difficoltà, è stato sviluppato un metodo innovativo che utilizza i Large Language Models (LLM) per associare automaticamente i messaggi di errore alle modifiche del codice che li hanno generati, consentendo una significativa riduzione dei tempi di analisi e correzione dei bug. Questo approccio, basato su modelli come BERT, si integra con i framework di sviluppo esistenti, offrendo agli sviluppatori uno strumento efficace per il tracciamento e la risoluzione dei problemi. Un'indagine condotta sugli utenti ha evidenziato una riduzione dei tempi di investigazione dei problemi fino al 60%, dimostrando il potenziale di questa tecnologia nel migliorare l'efficienza nello sviluppo di videogiochi.

Un ulteriore esempio di applicazione dell'AI nel Quality Assurance videoludico è rappresentato dal sistema SUPERNOVA [38], sviluppato sempre da Electronic Arts, che automatizza la selezione dei test e la prevenzione dei difetti attraverso il Risk-Based Testing (RBT) e il Machine Learning. SUPERNOVA utilizza modelli probabilistici e di apprendimento automatico per selezionare in modo ottimale i test da eseguire, riducendo il numero di test non necessari e ottimizzando l'allocazione delle risorse QA. Il sistema è stato implementato su titoli AAA, riducendo le ore di testing del 55% e migliorando l'efficacia dell'identificazione dei bug. Inoltre, il sistema include un modulo di prevenzione dei difetti che sfrutta modelli di apprendimento semi-supervisionato per identificare il codice potenzialmente problematico prima che venga integrato nel progetto, raggiungendo una precisione del 71% e un recall del 77% nell'identificazione di commit potenzialmente difettosi.

Nonostante l'ampia disponibilità di strumenti e framework per il testing automatico dei videogiochi, gli sviluppatori rimangono ancora riluttanti ad adottare agenti automatici per il test dei giochi. Questa esitazione è probabilmente legata alla natura invasiva di molti approcci attuali, che tendono a interrompere il flusso di lavoro degli sviluppatori. Inoltre, una limitata competenza in materia di testing automatizzato potrebbe comportare la necessità di investire tempo e risorse economiche, spesso difficili da reperire o allocare [37]. Per incoraggiare una maggiore adozione del testing automatico dei videogiochi, è necessario sviluppare nuovi approcci di testing che non interferiscano con il flusso di lavoro degli sviluppatori.

Tuttavia, è fondamentale fornire del tempo ai membri del team per sviluppare competenze nel testing automatizzato e assicurare la stabilità dell'infrastruttura sottostante. Il team di *Sea of Thieves* [35] ha rappresentato un esempio concreto dell'applicazione di tecniche di automazione in un gioco di grandi dimensioni.

Utilizzando un sistema di test automatizzati basati su attori, il team ha migliorato significativamente l'efficienza dei test, riducendo i tempi di verifica da due settimane a un giorno e mezzo. Un altro risultato misurabile è stato il controllo efficace sul numero massimo di bug durante la produzione, mantenendo il numero di bug segnalati basso, con un massimo di 214 bug rispetto ai 3.000 in *Banjo Kazooie: Nuts and Bolts* [34]. Inoltre, i test automatizzati hanno contribuito a una riduzione del tempo di crunch per gli sviluppatori, consentendo orari di lavoro più regolari [20].

Le lezioni apprese dal team sono state fondamentali per il successo dei test automatizzati. Il consenso del team è stato cruciale, con il supporto da tutti i livelli dell'organizzazione. È stato sottolineato che il pragmatismo negli approcci ai test è essenziale, adattando i metodi per soddisfare le esigenze del team e i requisiti del progetto.

In generale il testing automatico offre vantaggi significativi che non dovrebbero essere sottovalutati. In primo luogo, garantisce maggiore accuratezza nei test ripetitivi, riducendo il rischio di errori umani. Inoltre, migliora la qualità della vita dei tester manuali, liberandoli da compiti monotoni e consentendo loro di concentrarsi su attività più creative e strategiche. L'automazione consente anche una copertura di test più ampia, sfruttando la velocità e la continuità del processo, inclusa la possibilità di eseguire test su più dispositivi in parallelo, aspetto cruciale in un'epoca di piattaforme frammentate. Infine, l'automazione riduce i costi nel lungo termine, grazie alla possibilità di riutilizzare l'infrastruttura e i casi di test con costi marginali. Integrare il testing automatizzato può quindi migliorare la qualità del prodotto, l'efficienza del team e il processo complessivo di sviluppo dei videogiochi [15].

Embedded QA

Oltre a migliorare gli strumenti e le tecniche di testing automatico, è anche importante educare gli sviluppatori di videogiochi sui vantaggi del testing automatico [23]. Molti sviluppatori non sono a conoscenza dei potenziali vantaggi del testing automatico o credono che sia troppo costoso o complesso da implementare. Questa percezione è in parte dovuta alla storica mancanza di riconoscimento del ruolo del QA come una carriera strutturata all'interno dell'industria videoludica.

Tuttavia, il QA sta progressivamente evolvendo da un semplice punto di ingresso nel settore a un percorso professionale sostenibile e di valore. Tradizionalmente

percepito come un trampolino di lancio per arrivare ad altri ruoli, il QA offre in realtà opportunità di crescita significative, grazie alla sua natura flessibile che consente di esplorare diversi aspetti dello sviluppo del gioco. I professionisti del QA, attraverso la curiosità e l'interesse verso le varie discipline, possono contribuire attivamente al processo creativo, influenzando la produzione e garantendo la qualità del prodotto finale. Il QA, agendo come 'moltiplicatore di forza', facilita la collaborazione tra i team, promuove la player advocacy e gestisce la qualità, assicurando che il prodotto soddisfi gli standard desiderati. In particolare, il Quality Management emerge come un elemento chiave, permettendo ai team di concentrarsi sulla creazione mentre il QA si assicura che gli standard qualitativi siano rispettati. Questo approccio implica una valutazione dei rischi e una gestione della qualità che impatta direttamente la creazione del prodotto. L'ownership di specifiche aree del progetto e la capacità di comunicare efficacemente l'impatto del proprio lavoro sono cruciali per avanzare nella carriera QA e per far riconoscere il valore di questa figura all'interno dell'industria. In definitiva, il QA non è solo un tester, ma un vero e proprio game developer che contribuisce attivamente alla creazione di esperienze di gioco di alta qualità [23].

L'Embedded QA, presentato da BioWare, rappresenta un modello di Quality Assurance in cui i tester sono integrati direttamente nei team di sviluppo, lavorando a stretto contatto con programmatori, designer e altri membri del team durante l'intero ciclo di vita del progetto [17]. Questo approccio si contrappone al modello tradizionale di blackbox testing, in cui il team QA opera in modo isolato, ricevendo il gioco solo nelle fasi finali di sviluppo e senza una conoscenza approfondita del codice o degli strumenti di sviluppo. Nell'Embedded QA, i tester hanno accesso al codice sorgente, agli strumenti di sviluppo e ai processi decisionali del team di sviluppo.

Questo approccio ha guadagnato particolare rilevanza in un settore dove la crescente natura pubblica dei giochi e le pressioni della competizione spingono i team a rivedere le pratiche tradizionali di Quality Assurance. La stretta collaborazione tra QA e sviluppo si traduce in molteplici vantaggi:

- **Individuare i bug in modo più efficace e tempestivo:** La familiarità con il codice e gli strumenti di sviluppo permette ai tester di comprendere meglio il funzionamento interno del gioco e di identificare potenziali problemi in una fase iniziale del processo di sviluppo.
- **Fornire un feedback più preciso e utile agli sviluppatori:** La comprensione approfondita del gioco e delle sue meccaniche consente ai tester

di fornire feedback più dettagliato e mirato agli sviluppatori, aiutandoli a risolvere i problemi in modo più efficiente.

- **Contribuire attivamente alla progettazione e allo sviluppo del gioco:** I tester Embedded QA possono partecipare alle discussioni di design, suggerire miglioramenti e contribuire attivamente alla creazione di un'esperienza di gioco di alta qualità.

Un esempio pratico dell'efficacia di questo approccio emerge dalle esperienze di team che hanno integrato il QA nel flusso di lavoro fin dalle prime fasi. Jesse Vigil, docente presso USC Games, sottolinea come questa strategia abbia portato a una maggiore consapevolezza del processo QA tra gli sviluppatori e a un miglioramento della collaborazione tra reparti. In particolare, la formazione e il coinvolgimento diretto dei tester ha permesso a piccoli team di affrontare con successo le sfide legate allo sviluppo per console, garantendo standard qualitativi elevati anche in assenza di risorse tipiche di grandi editori [21].

Metodi

Per rispondere alle domande di ricerca, è stato sviluppato un questionario come strumento principale. Per garantire la privacy, non sono stati raccolti indirizzi email né informazioni che possano ricondurre direttamente all'identità dei compilatori. In questo capitolo vengono descritti la struttura del questionario e le metodologie adottate per l'analisi dei risultati ottenuti.

Struttura del questionario

Sono stati utilizzati strumenti come Notion, Google Form e Google Sheet per la progettazione e l'organizzazione del questionario. Quest'ultimo è stato redatto e somministrato in inglese, mentre nell' Appendice A è stato tradotto e riportato in italiano.

Il questionario è stato suddiviso in diverse sezioni, tre delle quali dedicate specificamente alle domande di ricerca (RQ):

- Domande Generali sul campione di intervistati
- Differenze tra processi di QA automatizzati e manuali
- QA Team e il rapporto con gli altri reparti
- Sfide del QA e rapporto con l'AI

Domande Generali sul campione di intervistati

In questa sezione, le domande sono progettate principalmente per verificare l'idoneità a completare l'intero questionario. In particolare, nel caso in cui il partecipante dichiari che nella propria azienda non vengono effettuati test, viene indirizzato a una breve sezione aggiuntiva in cui si indaga sull'eventuale esperienza pregressa in un team di QA. Se il partecipante dichiara di aver avuto tale esperienza, potrà proseguire con il questionario; diversamente, il questionario verrà concluso.

Le restanti domande si concentrano sul ruolo professionale ricoperto dallo sviluppatore che sta completando il questionario, includendo informazioni sugli anni di esperienza lavorativa, e sull'azienda in cui opera, con un'analisi della dimensione

e della localizzazione. Inoltre, viene chiesto se l'azienda utilizza servizi di test in outsourcing.

Differenze tra processi di QA automatizzati e manuali

La sezione si apre indagando quale engine venga utilizzato e per quale piattaforma si sviluppa principalmente. Successivamente, si analizzano i tipi di test più comuni (ad esempio, Functional, UX, Regression) e si verifica se questi vengano eseguiti in modalità automatica, manuale o attraverso una combinazione di entrambe. Si approfondiscono poi le motivazioni alla base della scelta tra test manuali e automatici. La sezione esplora inoltre i tipi di test automatici implementati, il loro livello di integrazione, i software utilizzati e le differenze rispetto al testing manuale, considerando aspetti come efficienza, vantaggi e svantaggi di ciascun approccio. Infine, vengono poste domande sui software di bug tracking impiegati dall'azienda.

QA Team e il rapporto con gli altri reparti

Le domande presenti in questa sezione esplorano come il team di QA integri l'automazione all'interno dei processi di sviluppo di un videogioco e collabori con gli altri reparti per ottimizzare la qualità del prodotto finale. Nello specifico, si chiede come l'automazione si integri nella giornata lavorativa e in che modo i risultati vengano comunicati all'intero team. Altre domande si focalizzano su quando il team di QA entra nel ciclo produttivo e su quando, secondo l'intervistato, dovrebbe essere coinvolto. Infine, si indaga se venga adottato un piano di qualità e quanto è importante che un tester sappia anche programmare.

Sfide del QA e rapporto con l'AI

Quest'ultima sezione si concentra maggiormente sulle principali sfide e difficoltà affrontate dal team di QA nei processi di automazione e come percepiscono l'evoluzione del loro ruolo con l'introduzione di tecnologie come l'AI nell'industria dei videogiochi. Queste domande mirano a comprendere quali potrebbero essere alcuni cambiamenti futuri, con o senza l'implementazione dell'AI nel team di QA.

Infine, al termine delle risposte a tutte le sezioni, viene offerta l'opportunità di lasciare un commento o un suggerimento attraverso una domanda aperta.

Reclutamento dei Partecipanti

Essendo il questionario anonimo, è stata adottata una strategia di distribuzione su diversi canali per raggiungere il maggior numero possibile di persone coinvolte nello

sviluppo o nel testing di videogiochi. In particolare, sono stati utilizzati alcuni canali Discord, gruppi e community su LinkedIn, oltre a una lista di contatti fornita da Joseph Hobbs, che ha permesso di contattare professionisti tramite email, LinkedIn e Discord. Infine, è stato chiesto a contatti fidati dell'industria di compilare il questionario e di condividerlo con altri professionisti del settore.

Metodologie

Dopo aver contattato oltre 100 persone attraverso diversi canali, sono pervenute solamente 18 risposte. Successivamente, è stata effettuata una verifica su tutte le risposte, risultate coerenti lungo l'intero questionario. Tuttavia, due intervistati non hanno potuto proseguire nella compilazione poiché hanno fornito risposte negative a entrambi i criteri di selezione precedentemente indicati. Di conseguenza, il numero totale di risposte valide è risultato pari a 16, le quali sono state successivamente analizzate. L'analisi di ciascuna sezione del questionario è stata condotta sia dal punto di vista quantitativo che qualitativo, con il supporto di rappresentazioni grafiche per una migliore interpretazione dei dati.

Analisi Quantitativa RQ1

Nella presente sezione, l'analisi mira a esaminare le differenze tra aziende che adottano approcci di testing automatico, manuale o una combinazione di entrambi. Per raggiungere tale obiettivo, i dati raccolti sono stati inizialmente suddivisi in tre gruppi, corrispondenti alle modalità di testing dichiarate dagli intervistati nella domanda 3.4 (Automatico, Manuale o Entrambi). Successivamente, sono state esaminate le variazioni di frequenza relative alle variabili individuate nelle domande 1.5, 1.6, 3.3, 3.5, 3.11 e 3.12. Particolare attenzione è stata rivolta al confronto tra la dimensione aziendale (domanda 1.5) e le domande a scelta multipla della sezione 3 sopra citate. Parallelamente, è stata effettuata un'ulteriore suddivisione basata sulla localizzazione geografica dell'azienda, distinguendo tra sedi situate in Italia e all'estero (domanda 1.6).

Analisi Quantitativa RQ2

La seconda sezione del questionario mira a indagare il livello di integrazione del QA con gli altri reparti, l'importanza attribuita alla comunicazione e l'impiego dell'automazione per facilitare la collaborazione. Per l'analisi quantitativa, i risultati sono stati inizialmente suddivisi nei tre gruppi precedentemente definiti (testing automatico, manuale ed entrambi). Successivamente, è stata effettuata un'ulteriore suddivisione in base alla dimensione aziendale e alla posizione geografica,

seguendo lo stesso criterio adottato nella sezione precedente. Tuttavia, sono state effettuate ulteriori suddivisioni per approfondire la comprensione delle risposte. In particolare, le risposte sono state raggruppate per ruolo, individuando quattro categorie: QA, Producer, Project Manager e Altro (presi dalla domanda 1.2). Inoltre, è stata introdotta una classificazione basata sugli anni di esperienza: 0-1, 2-5, 6-10 e oltre 10 anni (presa dalla domanda 1.4). Per ogni gruppo sono state correlate le domande incluse in questa sezione, nello specifico le domande 4.3, 4.4, 4.5, 4.11 e 4.12. Per ciascuna risposta e per ogni gruppo, sono stati calcolati la media e la deviazione standard. Questi parametri, confrontati tra di loro, si sono rivelati strumenti preziosi sia per individuare aree di particolare interesse, sia per identificare le domande che hanno generato maggior disaccordo tra i partecipanti. Infine, per un'analisi più approfondita di questa sezione, sono state considerate anche le risposte alla domanda 2.1, relativa all'adozione del QA testing tramite compagnie esterne.

Analisi Quantitativa RQ3

Questa sezione finale è dedicata all'esplorazione delle sfide affrontate dai team di QA e del loro rapporto con tecnologie emergenti, come l'intelligenza artificiale. L'analisi quantitativa segue lo stesso approccio adottato nella sezione precedente: le domande sono state suddivise nei medesimi gruppi e correlate tra loro, calcolando media e deviazione standard. In particolare le domande prese in considerazione sono: 5.5, 5.6, 5.7, 5.10 e 5.11. Il confronto tra le medie e le deviazioni standard ha fornito spunti significativi sulle sfide future e sull'evoluzione del ruolo del QA nel contesto dell'innovazione tecnologica, mettendo in luce le percezioni dei team sull'impatto dell'automazione e dell'intelligenza artificiale.

Analisi Qualitativa

L'analisi qualitativa delle risposte aperte raccolte nel questionario, per ogni sezione, è stata condotta utilizzando la Grounded Theory (GT), un approccio induttivo che consente di derivare teorie direttamente dai dati. Questa metodologia è particolarmente adatta per indagare fenomeni complessi e ottenere una comprensione approfondita delle percezioni e delle esperienze dei partecipanti.

L'analisi è stata applicata attraverso l'uso di un Large Language Model (LLM) per supportare l'estrazione e l'organizzazione dei temi. In particolare, l'analisi è stata realizzata utilizzando ChatGPT-4, che è stato impiegato per assistere nella classificazione e strutturazione dei dati raccolti nel questionario. Oltre all'impiego dello strumento, tutte le risposte sono state revisionate manualmente per verificare

l'accuratezza e la coerenza dei dati prodotti dall'LLM.

L'elaborazione dei temi è avvenuta attraverso un processo strutturato in tre fasi principali, corrispondenti ai passaggi della Grounded Theory:

1. Codifica aperta – identificazione dei concetti chiave

È stato richiesto a ChatGPT di identificare i concetti principali emergenti dalle risposte, con dei prompt specifici:

- *"Analizza questo file secondo le regole della Grounded Theory e sapendo che la RQ1 è 'In che modo le aziende di sviluppo videoludico scelgono tra testing manuale e automatizzato, e quali sono i principali fattori che influenzano questa decisione?'. Estrapola i concetti chiave più ricorrenti."*
- *"Analizza questo file secondo le regole della Grounded Theory e sapendo che la RQ2 è 'In che modo il team di Quality Assurance integra l'automazione nei processi di sviluppo del gioco e collabora con altri reparti per ottimizzare la qualità del prodotto finale?'. Estrapola i concetti chiave più ricorrenti."*
- *"Analizza questo file secondo le regole della Grounded Theory e sapendo che la RQ3 è 'Quali sono le principali sfide affrontate dai team di Quality Assurance nei processi di automazione, e come percepiscono l'evoluzione del loro ruolo in relazione all'introduzione di tecnologie come l'intelligenza artificiale nel settore dei videogiochi?'. Estrapola i concetti chiave più ricorrenti."*

2. Codifica assiale – raggruppamento dei concetti in categorie

Dopo aver ottenuto i concetti chiave, è stato chiesto a ChatGPT di raggrupparli in categorie tematiche, evidenziando le relazioni tra di essi. Il prompt utilizzato è stato: *"Sulla base dei concetti chiave individuati, raggruppa in categorie principali secondo la Grounded Theory"*.

3. Codifica selettiva – identificazione dei temi centrali

Una volta ottenute le categorie, è stata condotta una sintesi per individuare i temi principali che rispondevano alla RQ1, alla RQ2 o alla RQ3. Per questo passaggio, il prompt utilizzato è stato: *"Considerando le categorie individuate, sintetizza i temi centrali che emergono dall'analisi"*.

Grazie alla flessibilità della GT, è stato possibile esplorare i temi centrali delle domande di ricerca, facendo emergere strutture interpretative coerenti che riflettono le opinioni e le esperienze dei professionisti del settore. Questo approccio ha permesso di garantire un'analisi rigorosa e sistematica, in grado di fornire una base solida per interpretare i risultati qualitativi in relazione agli obiettivi della ricerca.

Sintesi

Sintesi analisi quantitative:

Sezione	Metodologia	Domande
Analisi RQ1	<p>Suddivisione in gruppi:</p> <ul style="list-style-type: none"> • testing automatico, manuale, entrambi; • dimensione aziendale; • localizzazione geografica dell'azienda. <p>Ogni gruppo è stato messo in relazione con le domande della sezione e sono state analizzate frequenze e distribuzioni.</p>	1.5, 1.6, 3.3, 3.4, 3.5, 3.11, 3.12
Analisi RQ2	<p>Suddivisione in gruppi:</p> <ul style="list-style-type: none"> • testing automatico, manuale, entrambi; • Per ruolo (QA, Producer, Project Manager, Altro); • Anni di esperienza; • dimensione aziendale; • localizzazione geografica dell'azienda. <p>Ogni gruppo è stato messo in relazione con le domande della sezione e sono state calcolate e confrontate media e deviazione standard.</p>	1.2, 1.4, 1.5, 1.6, 2.1, 3.4, 4.3, 4.4, 4.5, 4.11, 4.12
Analisi RQ3	Identico a RQ2.	1.2, 1.4, 1.5, 1.6, 2.1, 3.4, 5.5, 5.6, 5.7, 5.10, 5.11

Tabella 3.1: Sintesi delle metodologie quantitative usate

Sintesi analisi qualitativa:

Fase	Descrizione
Supporto LLM	ChatGPT-4, con successivo controllo manuale
Fase 1: Codifica Aperta	Identificazione dei concetti chiave emersi dalle risposte. I prompt specifici per ogni RQ erano: <i>"Analizza questo file secondo le regole della Grounded Theory e sapendo che la RQ1 è 'In che modo le aziende di sviluppo videoludico scelgono tra testing manuale e automatizzato...'. Estrapola i concetti chiave."</i> <i>"Analizza questo file secondo le regole della Grounded Theory e sapendo che la RQ2 è 'In che modo il team di Quality Assurance integra l'automazione...'. Estrapola i concetti chiave."</i> <i>"Analizza questo file secondo le regole della Grounded Theory e sapendo che la RQ3 è 'Quali sono le principali sfide affrontate dai team di Quality Assurance...'. Estrapola i concetti chiave."</i>
Fase 2: Codifica Assiale	Raggruppamento dei concetti chiave in categorie tematiche, evidenziando le relazioni tra di essi. Il prompt era: <i>"Sulla base dei concetti chiave individuati, raggruppalì in categorie principali secondo la Grounded Theory."</i>
Fase 3: Codifica Selettiva	Sintesi delle categorie per individuare i temi centrali. Il prompt utilizzato era: <i>"Considerando le categorie individuate, sintetizza i temi centrali che emergono dall'analisi."</i>
Risultato Finale	L'approccio ha permesso di esplorare i temi centrali delle domande di ricerca, garantendo un'analisi rigorosa e sistematica dei risultati qualitativi.

Tabella 3.2: Sintesi della metodologia qualitativa usata

Risultati

In questo capitolo vengono illustrati i risultati del questionario, analizzati secondo i metodi descritti nel capitolo 3. Nei capitoli sono inclusi i grafici considerati più utili, mentre ulteriori grafici con le informazioni rimanenti sono disponibili nell'Appendice B.

Informazioni sugli intervistati

In merito alla modalità di scoperta del sondaggio, si osserva che la maggior parte dei partecipanti ha appreso dell'indagine tramite Discord e LinkedIn, con quest'ultimo che si conferma come un canale particolarmente efficace per raggiungere professionisti del settore. L'assenza di risposte provenienti da email indica che i canali più diretti e social sono risultati più fruttuosi per la diffusione del sondaggio. Le risposte nella categoria "Altro" suggeriscono un coinvolgimento diretto da parte del creatore del questionario, indicando che una rete di contatti diretti ha avuto un ruolo nella diffusione dell'indagine.

Sul fronte delle professioni, la maggior parte dei partecipanti ricopre ruoli di Producer o di QA, con una presenza significativa di figure tecniche e di leadership nei team QA, come QA Automation Engineer e QA Lead. Inoltre, alcuni dei Producer hanno anche maturato esperienze precedenti in ruoli legati al QA. È importante evidenziare che vi è un solo Project Manager e una figura ibrida che dichiara di lavorare principalmente come QA, ma anche come Producer, ed è stata inclusa nella categoria "Altro".

I dati relativi all'esperienza nell'industria videoludica mostrano che la maggior parte dei partecipanti ha tra i 6 e i 10 anni di esperienza, ma le fasce 2-5 anni e oltre 10 anni sono quasi equivalenti in termini di risposte, suggerendo una distribuzione complessivamente bilanciata tra professionisti di medio e lungo corso. Non sono emersi partecipanti con meno di 2 anni di esperienza, evidenziando una partecipazione concentrata su figure con una consolidata conoscenza del settore.

Per quanto riguarda le dimensioni aziendali, le risposte indicano una distribuzione piuttosto equilibrata tra aziende di piccole, medie e grandi dimensioni, con i team più piccoli in una leggera minoranza. Questo riflette una varietà di realtà aziendali

coinvolte nel sondaggio.

Le risposte relative alla localizzazione delle aziende mostrano una leggera prevalenza di realtà internazionali rispetto a quelle italiane, con 10 partecipanti che lavorano per aziende con sede all'estero e 8 per aziende italiane. Questo equilibrio consente di raccogliere prospettive diverse sulle pratiche di Quality Assurance, influenzate sia da contesti nazionali che da quelli globali. La presenza di aziende italiane tra i partecipanti risulta particolarmente rilevante per analizzare l'approccio al QA nel panorama locale, in confronto a metodologie e strutture organizzative adottate da aziende di altri paesi.

In termini di tecnologie, il motore di gioco Unity emerge come il più utilizzato tra i partecipanti, seguito da Unreal Engine, riflettendo la loro ampia diffusione nel panorama dello sviluppo videoludico. Analizzando il rapporto tra l'engine utilizzato e la dimensione dell'azienda (Fig. 4.1), si osserva che Unity è prevalentemente adottato da aziende di grandi dimensioni (100+ dipendenti), con una rappresentanza significativa anche tra quelle medie (20-100 dipendenti) e piccole (1-20 dipendenti). Unreal Engine, invece, si concentra principalmente nelle aziende di medie dimensioni, con una presenza marginale nelle aziende più grandi. L'uso di motori proprietari è prevalente nelle grandi aziende, mentre Godot, pur essendo meno diffuso, appare esclusivamente tra le aziende di medie dimensioni. Nonostante le recenti modifiche alle tariffe di Unity, che hanno suscitato preoccupazioni tra gli sviluppatori, i dati indicano che molte aziende continuano a utilizzare questo motore, suggerendo una certa fedeltà o dipendenza dalle sue funzionalità. Parallelamente, Unreal Engine sta guadagnando terreno, con un'adozione crescente anche tra le aziende di medie dimensioni, probabilmente attratte dalle sue avanzate capacità grafiche e dalla struttura di licenza flessibile [7].

Engine usati e dimensione dell'azienda

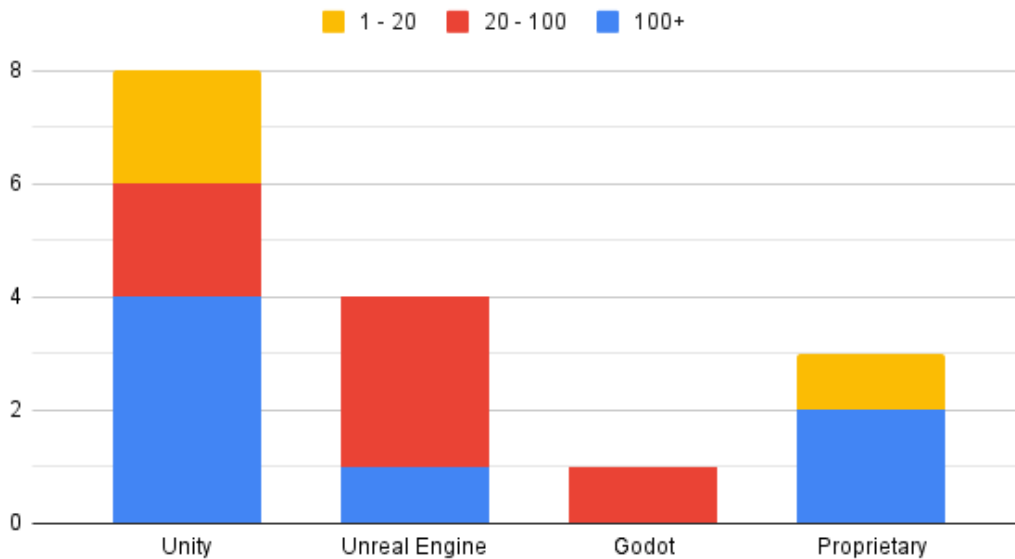


Figura 4.1: Grafico che incrocia le distribuzioni degli engine e della dimensione dell'azienda

Le risposte sulla piattaforma di sviluppo indicano una forte enfasi su PC e console, seguite da un interesse significativo per il mobile, riflettendo una diversificazione delle piattaforme di sviluppo. Tuttavia, confrontando questi dati con gli incassi globali dei primi tre quarti del 2024, che vedono il mobile rappresentare il 49% del mercato, console il 28% e PC il 23%, emerge un disallineamento: nonostante il mobile domini i ricavi, il numero di sviluppatori coinvolti su questa piattaforma risulta inferiore rispetto a PC e console [27]. Analizzando la distribuzione per dimensione aziendale (Fig. 4.2), si nota che nelle grandi aziende (100+ dipendenti) lo sviluppo è equamente distribuito tra PC, console e mobile, mentre nelle aziende di medie dimensioni (20-100 dipendenti) lo sviluppo su PC è leggermente predominante. Nelle aziende più piccole (1-20 dipendenti), lo sviluppo è distribuito in modo paritario tra le tre piattaforme. È stato inoltre segnalato un singolo caso di sviluppo per visori VR mobili, indicando un potenziale interesse per tecnologie emergenti e di nicchia.

Piattaforma più usata e dimensione dell'azienda

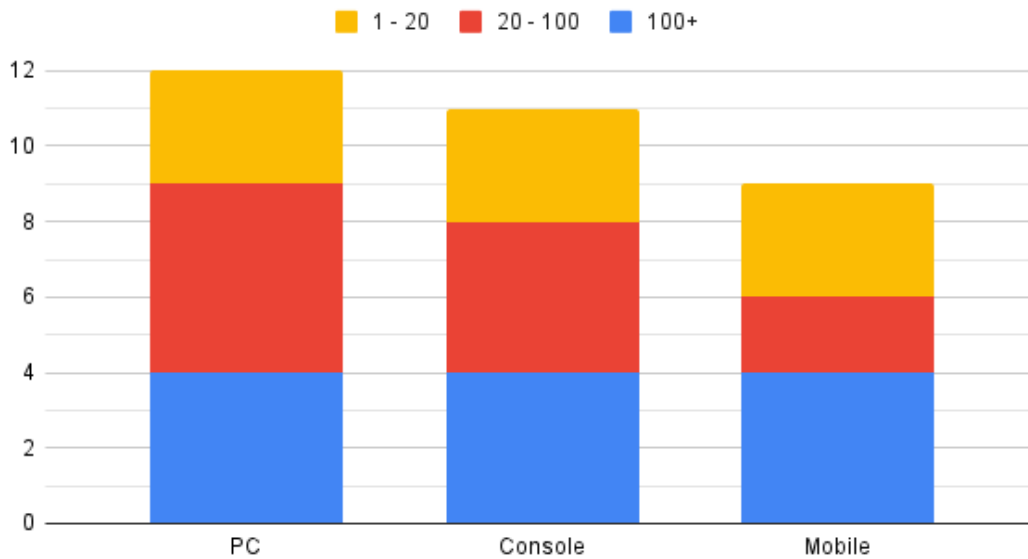


Figura 4.2: Grafico che incrocia le distribuzioni delle piattaforme e della dimensione dell'azienda

Per quanto riguarda la presenza di un team QA, la maggior parte dei partecipanti ha confermato che la propria azienda dispone di un team dedicato e la maggior parte dei partecipanti (12 su 18) conferma di svolgere attività di testing nelle proprie aziende, indicando che il testing continua a essere una parte rilevante dei processi di sviluppo. Dei 5 rispondenti che non dispongono di un team QA, 3 hanno comunque dichiarato di effettuare attività di testing. Alcuni hanno inoltre indicato modalità alternative di gestione, come l'outsourcing o l'assegnazione di compiti di QA a personale con altri ruoli. Un intervistato ha spiegato che la sua azienda non dispone di un team di QA a causa delle dimensioni ridotte del team. Tuttavia, secondo lui, questa motivazione sarebbe una scusa e ha suggerito che l'assenza del team QA sia più legata a fattori culturali che a reali limitazioni organizzative.

Infine, due risposte, delle 18 ricevute, non verranno incluse nelle analisi successive, poiché i rispettivi intervistati, dei Producer, pur lavorando in aziende con un team di QA, non si occupano direttamente di testing né hanno esperienze pregresse in ruoli legati al QA.

I grafici con tutte le distribuzioni sono presenti nell'Appendice B.

Invece qui sotto viene riportata una tabella delle informazioni importanti riportate

in questa sezione:

Ruolo	Dimensione	Anni	Italia	Team di QA	Testing	External QA	EX QA
Producer	1 - 20	10+	Yes	No	Yes		
Producer	20 - 100	10+	Yes	No	Yes		
Producer	20 - 100	6 - 10	No	No	No	Yes	Yes
QA	100+	2 - 5	No	Yes	Yes		
Producer	1 - 20	6 - 10	Yes	Yes	No	No	No
Project Manager	100+	2 - 5	No	No	No	Yes	Yes
QA	100+	2 - 5	No	Yes	Yes		
Mainly QA but I also work as a producer and have completed a course in project management	20 - 100	10+	Yes	Yes	Yes		
QA	100+	10+	No	Yes	No	Yes	Yes
QA	1 - 20	2 - 5	Yes	Yes	Yes		
Producer	100+	6 - 10	No	Yes	Yes		
Producer	100+	6 - 10	No	Yes	Yes		
Producer	20 - 100	2 - 5	Yes	Yes	No	Yes	No
Producer	100+	6 - 10	No	Yes	No	Yes	Yes
QA	20 - 100	6 - 10	Yes	Yes	Yes		
QA	20 - 100	2 - 5	Yes	Yes	Yes		
Producer	1 - 20	6 - 10	No	No	Yes		
QA	20 - 100	10+	No	Yes	Yes		

Tabella 4.1: Dati demografici

Risultati RQ1

La domanda 3.4, “Come vengono condotti i test nella tua azienda?”, è stata usata per suddividere i partecipanti in categorie, fornendo una base analitica per confrontare le risposte all’interno della sottosezione dedicata alla RQ1 e delle successive. Dai risultati emerge che una percentuale molto alta dei partecipanti effettua test sia manuali che automatizzati, mentre una piccola percentuale si limita ai soli test manuali. Nessuno, invece, utilizza esclusivamente test automatizzati. Questo ha portato alla classificazione dei partecipanti in due sole categorie: Manuale ed Entrambi.

Un aspetto interessante è che le tre risposte relative ai test manuali provengono da aziende con dimensioni diverse (1-20, 20-100 e 100+ dipendenti, Fig. 4.3), di cui solo una con sede in Italia e una priva di un team di QA dedicato, ma fa affidamento a QA esterno. Le motivazioni fornite da questi partecipanti per non adottare test automatizzati includono:

- “Troppo sforzo”
- “Le automazioni e gli strumenti automatizzati devono essere considerati durante la produzione. Tuttavia, se il tipo di gioco cambia frequentemente, spesso le automazioni non si adattano completamente. Nella nostra azienda, abbiamo scelto un approccio basato sull’intervento umano piuttosto che sull’automazione, a causa delle considerazioni sulle risorse e sui costi di sviluppo di questi strumenti.”
- “Team di QA troppo piccolo nella mia compagnia”

Tali risposte riflettono pienamente le tendenze generali nel settore [5].

Sono emerse, inoltre, due risposte incoerenti con la domanda, provenienti da partecipanti della categoria Entrambi:

- “Principalmente per cercare errori in grandi asset 3D”
- “Non sono responsabile del team. Il mio ruolo è più focalizzato sul project management e sui test manuali.”

Come un'azienda effettua i test in base alla sua dimensione

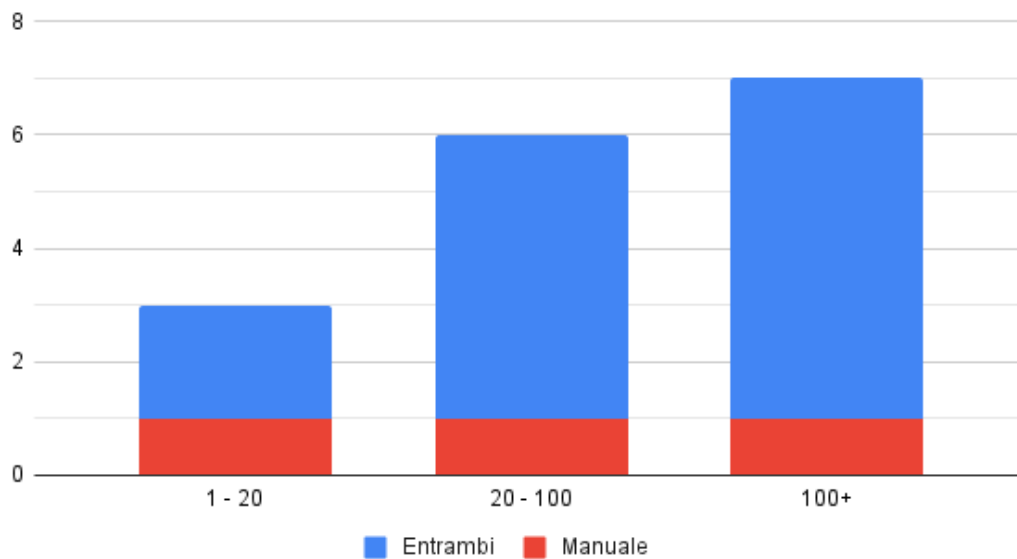


Figura 4.3: Come un'azienda effettua i test in base alla sua dimensione

Per quanto riguarda l'utilizzo di strumenti sviluppati internamente o di terze parti, la maggior parte dei partecipanti appartenenti alla categoria Entrambi fa uso di tool sviluppati internamente. Una piccola percentuale adotta un approccio misto, utilizzando sia strumenti interni che esterni, mentre nessuno si affida esclusivamente a tool esterni. Una risposta proveniente dalla categoria Manuale indica invece l'uso combinato di tool interni ed esterni. Un dato rilevante è che anche aziende di piccole dimensioni dimostrano una propensione a sviluppare strumenti interni, anziché affidarsi a soluzioni di terze parti (Fig. 4.4). Infine, non si rilevano differenze significative tra aziende italiane ed estere nell'adozione di tool sviluppati internamente.

Utilizzo di tool interni o esterni in base alla dimensione dell'azienda

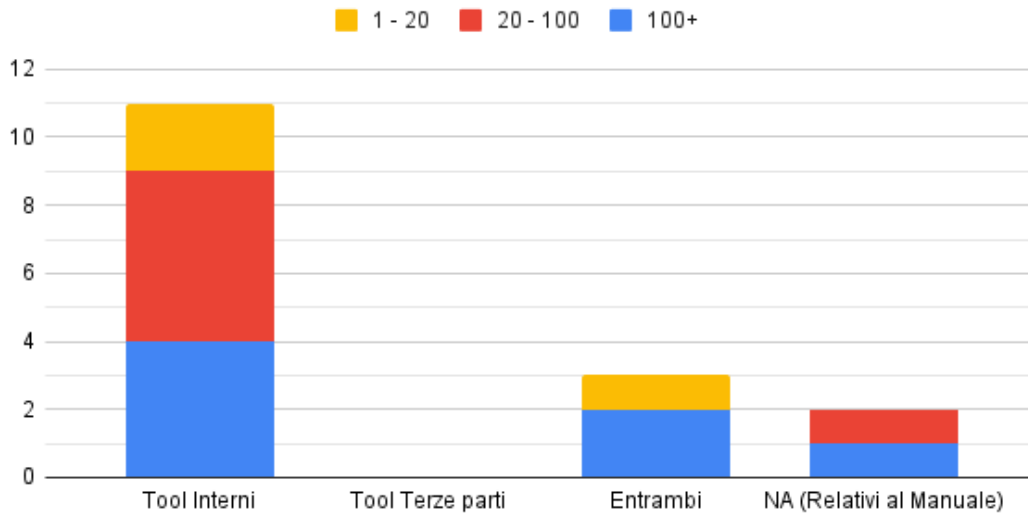


Figura 4.4: Utilizzo di tool interni o esterni in base alla dimensione dell'azienda

L'analisi delle risposte sui tipi di test più frequentemente eseguiti evidenzia alcune tendenze interessanti che riflettono le priorità e le strategie delle aziende nello sviluppo e nella qualità dei videogiochi. Il Functional Testing emerge come il più utilizzato, con tutti i partecipanti che lo hanno selezionato. Seguono il Regression Testing e il Performance Testing, indicati da molti partecipanti, a testimonianza dell'importanza di garantire stabilità e ottimizzazione, soprattutto in un contesto in cui i giochi devono soddisfare aspettative sempre più elevate in termini di fluidità e assenza di bug.

Anche test come il UX Testing e il Compliance Testing sono stati segnalati da una percentuale significativa di partecipanti, indicando una crescente attenzione verso l'esperienza utente e la conformità normativa, aspetti essenziali per il successo commerciale e il rispetto degli standard di mercato. Al contrario, il Localization Testing e il Compatibility Testing risultano meno diffusi, probabilmente perché la localizzazione di un gioco tende a essere affrontata nelle fasi finali dello sviluppo o, in alcuni casi, post-lancio, se le risorse di tempo e budget lo consentono. Di conseguenza, molti giochi vengono inizialmente sviluppati in lingua inglese. Per quanto riguarda il Compatibility Testing, la sua frequenza dipende dalla capacità

dell'azienda di sviluppare simultaneamente su più piattaforme, un aspetto che varia in base alle risorse e alle priorità del team di sviluppo.

Analizzando le risposte in relazione alle dimensioni aziendali, si osserva una distribuzione complessivamente simile ai dati totali. Le aziende di grandi dimensioni eseguono tutti i tipi di test, con una marcata prevalenza del Functional Testing e del Regression Testing. Anche le aziende di piccole dimensioni coprono l'intero spettro di test, ma presentano un valore significativamente basso per il Regression Testing. Questo potrebbe essere attribuito alla loro tendenza a concentrarsi su test considerati prioritari, come il Functional Testing o il Compliance Testing, o alla dimensione generalmente più contenuta del prodotto videoludico sviluppato. Le aziende di medie dimensioni, invece, mostrano un dato particolare: non effettuano Localization Testing, e il Compatibility Testing è il test meno frequente, in linea con la distribuzione generale delle risposte di tutti i partecipanti (Fig. 4.5).

Tipi di test in base a dimensione dell'azienda

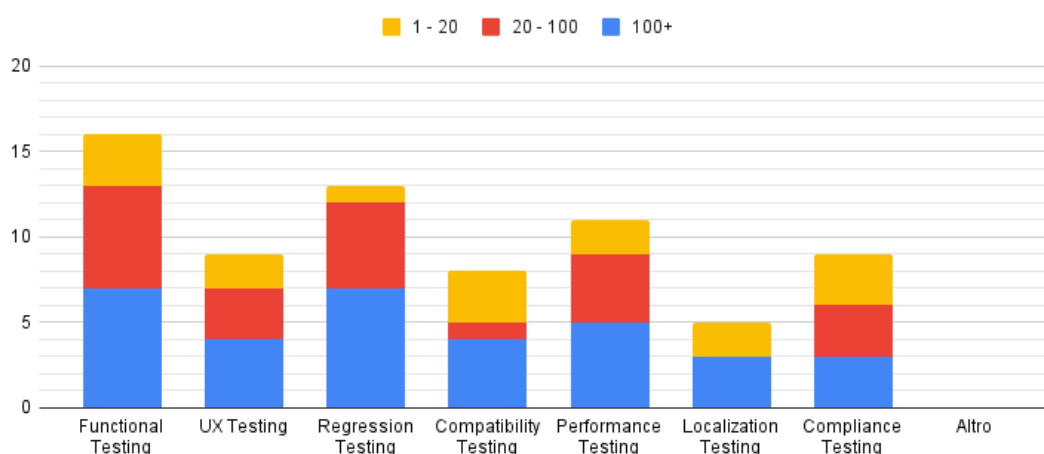


Figura 4.5: Tipi di test in base a dimensione dell'azienda

Dal punto di vista geografico, i partecipanti italiani segnalano una maggiore attenzione al Compliance Testing, mentre quelli esteri mostrano un utilizzo più ampio di Regression Testing e Performance Testing, escludendo il Functional Testing che rimane ovviamente il più utilizzato. Questa differenza potrebbe essere influenzata sia dalla dimensione aziendale, e quindi dalle risorse disponibili (tra i partecipanti, le aziende italiane sono 8 e appartengono principalmente alla categoria delle piccole o medie imprese), sia dalla natura del contratto stipulato con il publisher (Fig. 4.6).

Tipi di test in base alla posizione geografica dell'azienda

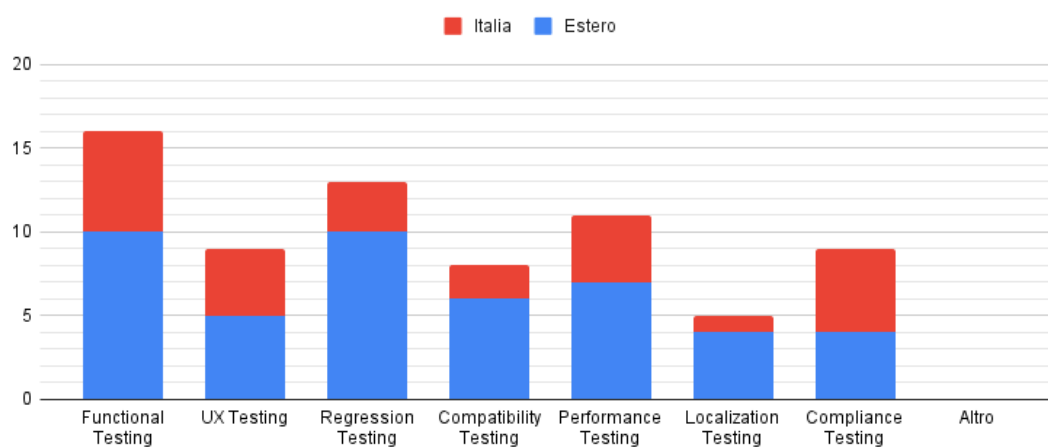


Figura 4.6: Tipi di test in base alla posizione geografica dell'azienda

Tipi di test in base a come l'azienda effettua i test

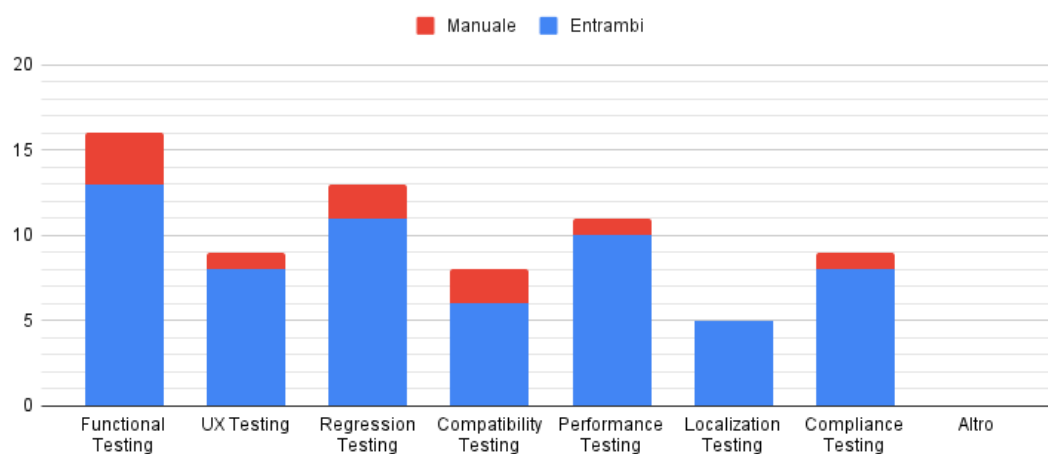


Figura 4.7: Tipi di test in base a come l'azienda effettua i test

Infine, è interessante osservare come le aziende appartenenti alla categoria Entrambi coprano tutti i tipi di test, risultando allineate con le tendenze generali emerse dalle risposte complessive. Al contrario, nella categoria Manuale non si registra alcuna risposta relativa al Localization Testing (Fig. 4.7).

Riguardo all'analisi di utilizzo di software di automazione possiamo notare come 8 partecipanti su 16 non hanno indicato alcun software specifico, ovviamente 3 dei quali fanno parte della categoria Manuale. Nelle restanti risposte, ovviamente facenti parte della categoria Entrambi, possiamo osservare che il Unity Test Framework è il più utilizzato con 3 risposte, seguito da Selenium e l'Unreal Engine Automation System, ciascuno con 1 risposta. Inoltre, 3 partecipanti hanno indicato l'uso di framework personalizzati:

- "Utilizziamo un framework di test sviluppato in proprio per il nostro motore proprietario."
- "Framework interno costruito intorno al concetto di test Game Loop di Google"
- "Framework personalizzato"

Dal punto di vista della dimensione aziendale, le aziende più grandi mostrano una maggiore propensione all'utilizzo di framework personalizzati e del Unity Test Framework, mentre nelle aziende di dimensioni più piccole, l'utilizzo di strumenti automatizzati è minimo, con una sola risposta che indica l'adozione di un framework personalizzato. Le aziende di medie dimensioni si distribuiscono invece su tutti e tre gli strumenti proposti nel questionario (Fig. 4.8).

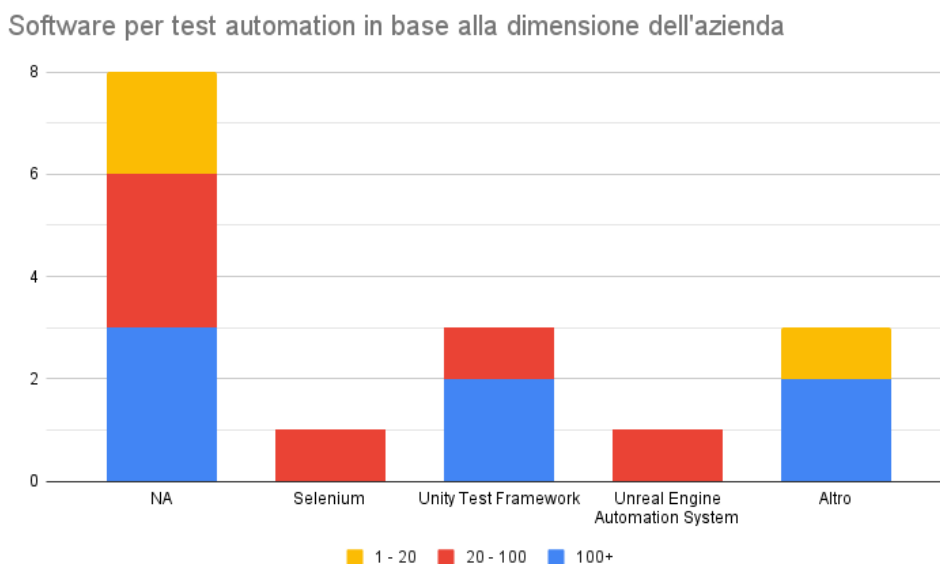


Figura 4.8: Software per test automation in base alla dimensione dell'azienda

Infine, per quanto riguarda la posizione geografica, le aziende italiane mostrano una preferenza per strumenti come Selenium e l'Unreal Engine Automation System, con una presenza limitata di framework personalizzati. Al contrario, le aziende estere evidenziano una maggiore adozione del Unity Test Framework e una tendenza più marcata all'uso di soluzioni su misura.

L'ultima domanda chiusa del questionario, relativa agli strumenti utilizzati per il bug tracking, evidenzia una chiara predominanza di Jira, indicato da 11 partecipanti su 16, confermandosi lo strumento più diffuso. Seguono, con numeri significativamente inferiori, Asana (2 risposte), Mantis (1 risposta), Trello (1 risposta). Inoltre, un partecipante ha specificato autonomamente l'utilizzo di ClickUp, uno strumento non incluso tra le opzioni proposte. Nessuno ha segnalato l'uso di Bugzilla o GitHub Issues.

Suddividendo i dati in base a come vengono effettuati i test, emerge che Jira è utilizzato in entrambe le categorie, con la categoria Manuale che utilizza solo questo strumento. Gli altri strumenti, come Asana, Mantis, Trello e ClickUp, sono stati riportati esclusivamente dai partecipanti che effettuano sia test manuali che automatizzati.

In relazione alle dimensioni aziendali, si osserva che Jira è lo strumento preferito da tutte le fasce dimensionali, con un'adozione particolarmente marcata nelle aziende di grandi dimensioni (7 risposte). Le aziende più piccole utilizzano anche strumenti alternativi come Trello, mentre le medie mostrano una maggiore varietà, includendo Asana, Mantis e ClickUp (Fig. 4.9).

Dal punto di vista geografico, le aziende estere rappresentano le principali utilizzatrici di Jira (10 risposte su 11 totali), mentre quelle italiane evidenziano una maggiore frammentazione, con l'adozione di strumenti come Asana, Mantis, Trello e ClickUp. Questi dati sottolineano una preferenza globale per Jira come standard, accanto a una maggiore varietà di scelte nelle realtà più piccole o locali.

Software per bug tracking in base alla dimensione dell'azienda

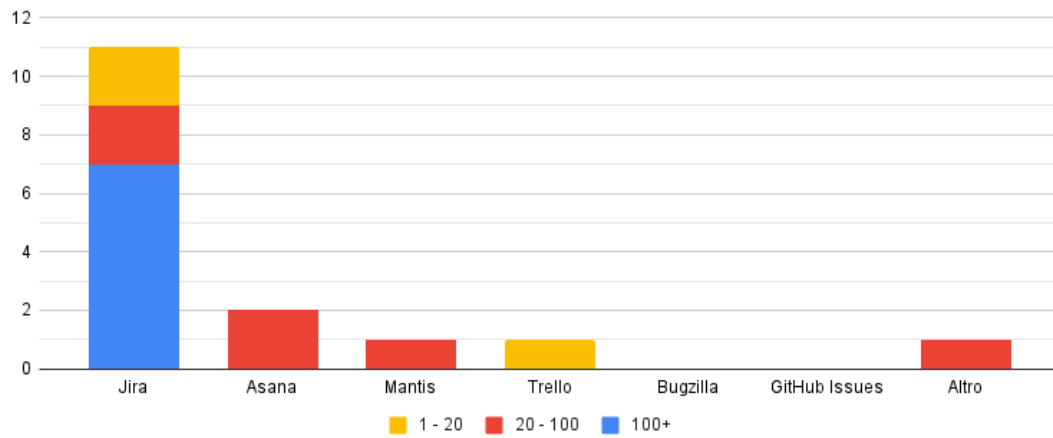


Figura 4.9: Software per bug tracking in base alla dimensione dell'azienda

Passando all'analisi delle risposte aperte, l'elaborazione dei temi è avvenuta attraverso un processo strutturato in tre fasi principali, corrispondenti ai passaggi della Grounded Theory, come precedentemente menzionato:

1. Codifica aperta – identificazione dei concetti chiave

Il file contenente le risposte aperte è stato analizzato per individuare parole e frasi ricorrenti.

Il prompt ha prodotto i risultati presenti nella figura 4.10.



Figura 4.10: Concetti chiave RQ1

2. Codifica assiale – raggruppamento dei concetti in categorie

L'analisi ha evidenziato le seguenti categorie:

- **Tipologie di Testing**

Questa categoria riguarda il tipo di test che le aziende eseguono e come vengono distribuiti tra manuale e automatizzato.

- Concetti: Automated tests, Manual testing, Performance testing, Regression and performance testing.

I test di performance e regressione sono spesso automatizzati per garantire ripetibilità e velocità, mentre alcuni test funzionali possono rimanere manuali.

- **Fattori Decisionali**

Questa categoria identifica i principali elementi che influenzano la scelta tra testing manuale e automatizzato.

- Concetti: Priority is, Can be automated, Both manual and automated.

Le aziende valutano la fattibilità dell'automazione in base alle priorità aziendali e alla natura del test.

- **Efficienza e Benefici dell'Automazione**

Qui emergono i benefici principali del testing automatizzato rispetto a quello manuale.

- Concetti: Automated tests are, Can run, Part of the process.

L'automazione viene scelta per la sua capacità di eseguire test ripetitivi rapidamente e senza errore umano.

- **Limitazioni e Sfide del Testing Manuale e Automatizzato**

Questa categoria raccoglie le principali limitazioni percepite nel testing manuale e automatizzato.

- Concetti: Manual testing is necessary, Test cases are, They are limited by.

L'automazione non è sempre possibile o conveniente, quindi il testing manuale resta essenziale in alcuni scenari.

3. Codifica selettiva – identificazione dei temi centrali

Sono stati evidenziati i seguenti temi:

1. Complementarietà tra Testing Manuale e Automatizzato
2. Fattori Decisionali: Priorità e Automazione Possibile
3. Benefici dell'Automazione: Velocità, Ripetibilità e Efficienza
4. Limitazioni del Testing Manuale e Automatizzato

Esplorando meglio le domande aperte emerge una distinzione significativa tra i test manuali e quelli automatizzati, entrambi fondamentali per il processo di Quality Assurance nello sviluppo videoludico. I test automatizzati sono ampiamente utilizzati per attività ripetitive e strutturate, come i test di performance, regressione, funzionali ed end-to-end. Questi test offrono un notevole vantaggio in termini di velocità ed efficienza, riducendo il carico di lavoro dei tester e consentendo un monitoraggio continuo durante lo sviluppo. Tale vantaggio in termini di risparmio temporale è stato confermato anche nel video intitolato “AI for Testing: The Development of Bots that Play Battlefield V” [19]. Tuttavia, i partecipanti hanno sottolineato che i test automatizzati non sono adatti per analizzare aspetti soggettivi come l'esperienza utente, la resa visiva della grafica o la qualità del suono. Questi elementi richiedono un giudizio umano più approfondito e vengono quindi

valutati in modo più efficace attraverso i test manuali, che permettono di cogliere sfumature e dettagli che l'automazione non può rilevare.

La scelta tra testing manuale e automatizzato è influenzata da diversi fattori, tra cui la priorità e la gravità dei bug riscontrati, la fase di sviluppo in cui si trova il progetto, ma anche costo, tempo e complessità tecnica. Ad esempio, i test automatizzati, nonostante la configurazione iniziale costosa, risultano particolarmente efficaci nelle prime fasi, quando l'obiettivo è identificare problemi critici come crash, blocchi di progressione o altre anomalie che possono compromettere lo sviluppo. Invece, durante le fasi di affinamento e playtesting, i test manuali assumono un ruolo centrale per garantire che il prodotto finale risponda agli standard qualitativi previsti e soddisfi le aspettative degli utenti finali.

Un aspetto rilevante emerso dalle risposte è il ruolo integrativo tra i due approcci. I test automatizzati non solo riducono il carico di lavoro sui tester manuali, ma forniscono anche dati e strumenti che possono essere utilizzati per migliorare l'efficacia dei test esplorativi. Questo approccio ibrido permette alle aziende di sfruttare al massimo i punti di forza di entrambi i metodi, massimizzando l'efficienza complessiva del processo di QA.

Tuttavia, l'affidabilità dei test automatizzati può essere compromessa dalla stabilità del gioco, in quanto questi strumenti richiedono un ambiente prevedibile e coerente per produrre risultati accurati. Nei progetti caratterizzati da frequenti modifiche al codice o instabilità, i test automatizzati possono generare falsi positivi o negativi, richiedendo un ulteriore intervento manuale per la verifica.

Infine, l'integrazione tra test manuali e automatizzati rappresenta una sfida cruciale per le aziende, ma anche un'opportunità strategica. Mentre i test automatizzati offrono efficienza, scalabilità e velocità, i test manuali continuano a essere indispensabili per garantire un'esperienza utente ottimale e una qualità artistica e creativa all'altezza delle aspettative del settore. Questo equilibrio tra automazione e manualità è un elemento chiave per affrontare le complessità del QA nello sviluppo dei videogiochi moderni.

Risultati RQ2

Passando all'analisi delle domande relative alla RQ2, queste sono state esaminate in relazione alle categorie precedentemente individuate, ossia Manuale ed Entrambi, con riferimento alle modalità di esecuzione dei test, alla dimensione aziendale e alla posizione geografica delle aziende coinvolte. Inoltre, l'analisi è stata arricchita con il confronto di due ulteriori categorie: il ruolo professionale dei rispondenti e gli anni trascorsi nell'industria. Come nella scorsa sezione, sono presenti sia domande aperte che chiuse; queste ultime, in particolare, seguono la scala Likert.

La comunicazione tra il team QA e gli altri team di sviluppo è considerata fondamentale dai partecipanti, con una media generale di 4,69 e una deviazione standard di 0,60, a indicare una forte convergenza di opinioni sull'importanza di questo aspetto.

Tra coloro che effettuano test manuali, si riscontra uniformità, con tutti che hanno attribuito il punteggio massimo, corrispondente al "completamente d'accordo". Per chi utilizza una combinazione di test manuali e automatizzati, la media si attesta leggermente più bassa, a 4,62, pur mantenendosi elevata. Questo dato sottolinea l'importanza di una comunicazione efficace tra i vari reparti per garantire il successo di un progetto.

In base alla dimensione aziendale, le realtà con oltre 100 dipendenti attribuiscono alla comunicazione il valore più alto (media 4,86). Anche le aziende di medie dimensioni mostrano valori elevati, mentre le aziende più piccole registrano una media inferiore (4,33) e una maggiore variabilità (deviazione standard 1,15) (Fig. 4.11). Dal punto di vista geografico, i partecipanti esteri attribuiscono una media di 4,9, rispetto al 4,33 dei rispondenti italiani.

Analizzando i ruoli, i professionisti QA evidenziano una percezione molto alta (media 4,71), seguiti da Producer (media 4,57), con una maggiore variabilità in quest'ultimo gruppo. Infine, l'esperienza nell'industria mostra valori più alti tra chi ha 2-10 anni di esperienza (media tra 4,8 e 4,83), mentre chi supera i 10 anni registra una media leggermente inferiore (4,4) e una maggiore dispersione.

È rilevante osservare la presenza di un generale atteggiamento positivo nei confronti della questione, sia tra i team dotati di un reparto QA interno, sia tra quelli che si avvalgono di risorse esterne o che, pur non avendo un team dedicato al QA,

Importanza comunicazione tra reparti in base alla dimensione dell'azienda

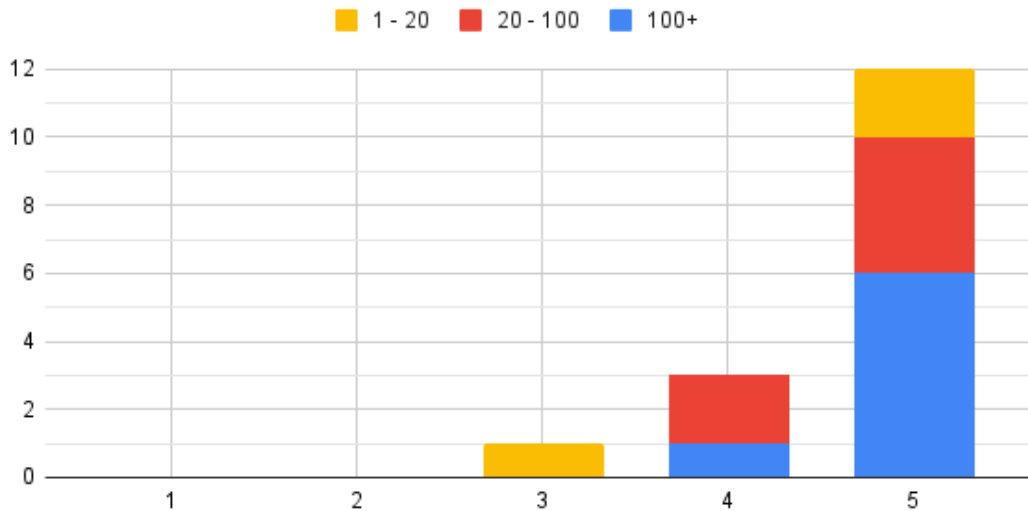


Figura 4.11: Importanza comunicazione tra reparti in base alla dimensione dell'azienda

svolgono comunque attività di testing. In tutti i casi, viene attribuita grande importanza alla comunicazione.

Invece per quanto riguarda la percezione dell'utilità degli strumenti di automazione nel facilitare la comunicazione tra il team di QA e altri reparti varia tra i partecipanti, con una media complessiva di 2,94 e una deviazione standard di 1,12, indicando un'opinione moderatamente negativa ma con significative differenze individuali. Le aziende che adottano sia il testing manuale che automatizzato mostrano una valutazione più alta (media 3,23) rispetto a quelle che si affidano esclusivamente al testing manuale (media 1,67).

Le differenze legate alla dimensione aziendale evidenziano che le aziende di piccole e medie dimensioni riportano una media di 3, ma con una differenza nella deviazione standard, rispettivamente 1,7 e 0,6. Invece, quelle più grandi registrano una media leggermente inferiore (2,86) e una deviazione standard di 1,35 (Fig. 4.12). Geograficamente, i partecipanti italiani percepiscono gli strumenti di automazione in modo più favorevole (media 3,33) rispetto ai partecipanti internazionali (media 2,7). Entrambi i gruppi, tuttavia, presentano una distribuzione delle risposte su quasi tutte le opzioni disponibili.

Parere su "I tool di automazione aiutano la comunicazione" in base alla dimensione dell'azienda

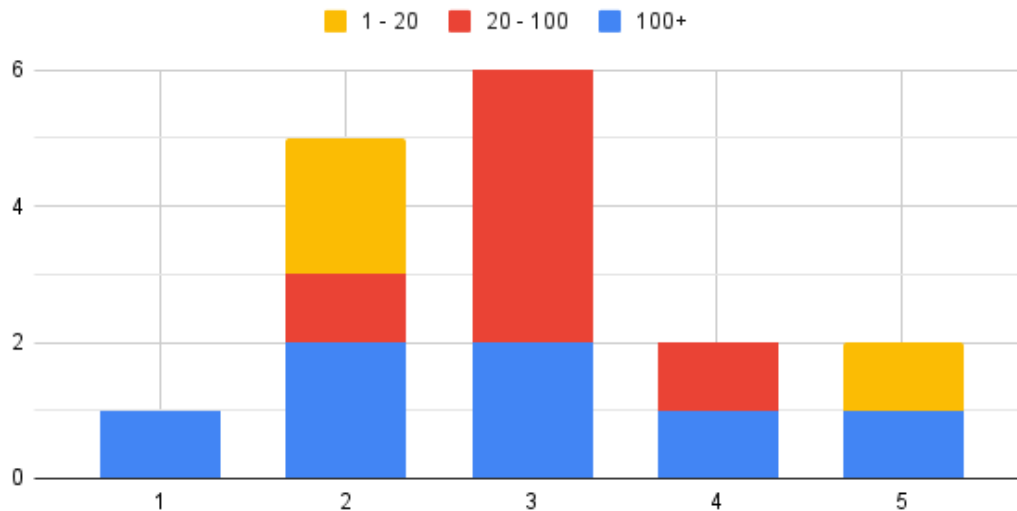


Figura 4.12: Parere su "I tool di automazione aiutano la comunicazione" in base alla dimensione dell'azienda

Dal punto di vista dei ruoli, i professionisti QA attribuiscono una media più alta (3,29) rispetto ai Producer (2,57). Infine, per quanto riguarda l'esperienza nell'industria, i partecipanti con oltre 10 anni di esperienza hanno una media di 3,8, mostrando una percezione più positiva, mentre quelli con 6-10 anni registrano una media inferiore (2,17), con risposte prevalentemente negative. I partecipanti con 2-5 anni di esperienza, invece, tendono a distribuire le risposte su valori centrali.

Si osserva che nei punteggi più alti, 4 e 5, siano presenti sia figure QA sia Producer, con una prevalenza delle prime. Questi includono individui con sia molta esperienza sia poca esperienza, oltre a un'azienda che utilizza servizi di QA esterno. Al contrario, le persone con esperienza intermedia tendono ad avere un'opinione negativa.

Per quanto riguarda l'importanza del feedback fornito dal team QA per il successo di un videogioco è generalmente riconosciuta dai partecipanti, con una media complessiva di 4,25 e una deviazione standard di 0,77, indicando una percezione positiva e relativamente omogenea. Le aziende che adottano una combinazione di

testing manuale e automatizzato registrano una media leggermente inferiore (4,23) rispetto a quelle che impiegano esclusivamente il testing manuale (4,33). Tuttavia, queste ultime presentano una variabilità maggiore, con una deviazione standard di 1,15 rispetto a 0,72.

Analizzando le dimensioni aziendali, le realtà più grandi attribuiscono un valore più alto (media 4,43) rispetto alle piccole aziende (media 3,67). Questo dato potrebbe indicare che, nelle aziende di dimensioni ridotte, i prodotti riescano a ottenere risultati anche senza il supporto di un team di QA (Fig. 4.13). Tale considerazione trova conferma nell'analisi delle risposte relative all'opzione 3 (centrale), in cui si osserva una maggiore presenza di piccole aziende e di Producer con un'esperienza superiore a 6 anni. Geograficamente, i partecipanti esteri mostrano una media più alta (4,4) rispetto a quelli italiani (4,0), con una maggiore variabilità nei contesti internazionali.

Parere su "Il feedback del team QA è essenziale per il successo di un videogioco" in base alla dimensione dell'azienda

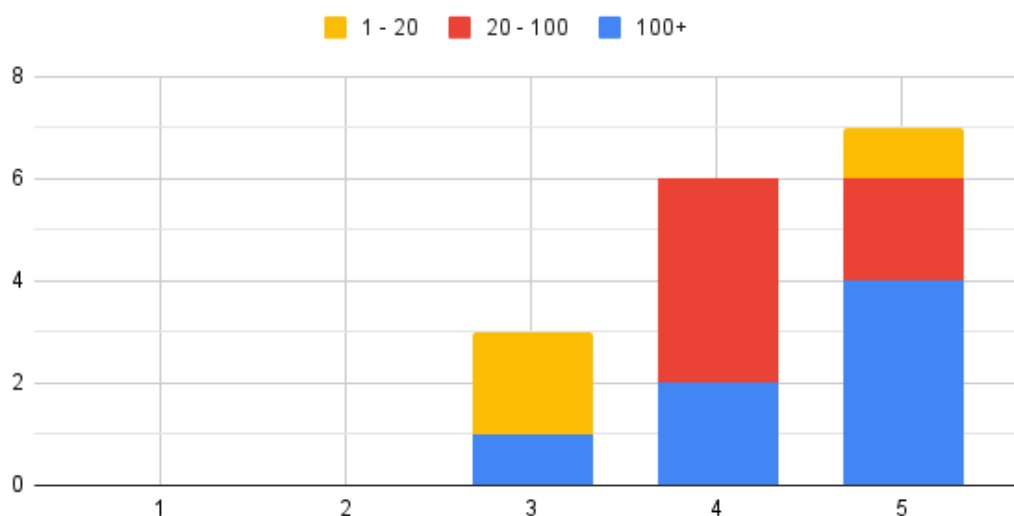


Figura 4.13: Parere su "Il feedback del team QA è essenziale per il successo di un videogioco" in base alla dimensione dell'azienda

Dal punto di vista dei ruoli, i professionisti QA valutano il feedback come particolarmente importante (media 4,43), mentre i Producer presentano una maggiore dispersione (media 4,0, deviazione standard 1), riflettendo una gamma più ampia

di percezioni. In termini di esperienza, i partecipanti con 2-5 anni nell'industria mostrano la media più alta (4,4) prevalentemente composta da figure QA. Seguono i partecipanti con oltre 10 anni di esperienza, che presentano una media di 4,2, anch'essi in maggioranza QA, e quelli con 6-10 anni di esperienza, con una media di 4,17.

L'integrazione del team di QA nelle fasi iniziali dello sviluppo di un videogioco è valutata in modo generalmente moderato dai partecipanti, con una media complessiva di 2,69 e una deviazione standard di 1,14, suggerendo una variabilità significativa nelle esperienze. Le risposte, infatti, sono distribuite su tutte le opzioni disponibili. Le aziende che utilizzano il testing manuale e quelle che adottano una combinazione di testing manuale e automatizzato mostrano risultati molto simili, rispettivamente con una media di 2,67 e 2,69, ma una leggera maggiore variabilità è osservata nelle aziende che utilizzano entrambi i metodi (deviazione standard di 1,18).

Le aziende più piccole attribuiscono la minore importanza all'integrazione del QA all'inizio del processo (media 1,67), mentre le aziende di medie (media 3) e grandi dimensioni (media 2,86) mostrano un maggiore riconoscimento, sebbene nessuna di queste categorie esprima un pieno accordo con l'affermazione presente nella domanda (Fig. 4.14). Geograficamente, le risposte internazionali indicano un'integrazione più alta (media 3,1) rispetto a quelle italiane (media 2), con una maggiore variabilità nelle prime, mentre le risposte italiane sono più concentrate su valori bassi.

Dal punto di vista dei ruoli, i professionisti QA, con una media di 2,86 e deviazione standard di 1,2, tendono a concentrarsi maggiormente sul non integrare il team di QA fin dall'inizio. I Producer, con una media di 2,43, presentano invece una distribuzione più uniforme su quasi tutti i valori (deviazione standard di 1,3). È importante sottolineare che nei valori più bassi sono presenti le aziende che non dispongono di un team di QA e che svolgono attività di testing utilizzando approcci alternativi (Fig. 4.15). In base all'esperienza, i partecipanti tendono a concentrarsi sui valori centrali, con 6-10 anni nell'industria, mostrano la media più alta (3), seguiti dai meno esperti (2-5 anni, media 2,6) e dai più esperti (10+ anni, media 2,4). Questi risultati indicano che l'integrazione del QA nelle prime fasi è ancora una sfida significativa per molte aziende, con notevoli margini di miglioramento.

Integrazione team QA all'inizio dello sviluppo in base alla dimensione dell'azienda

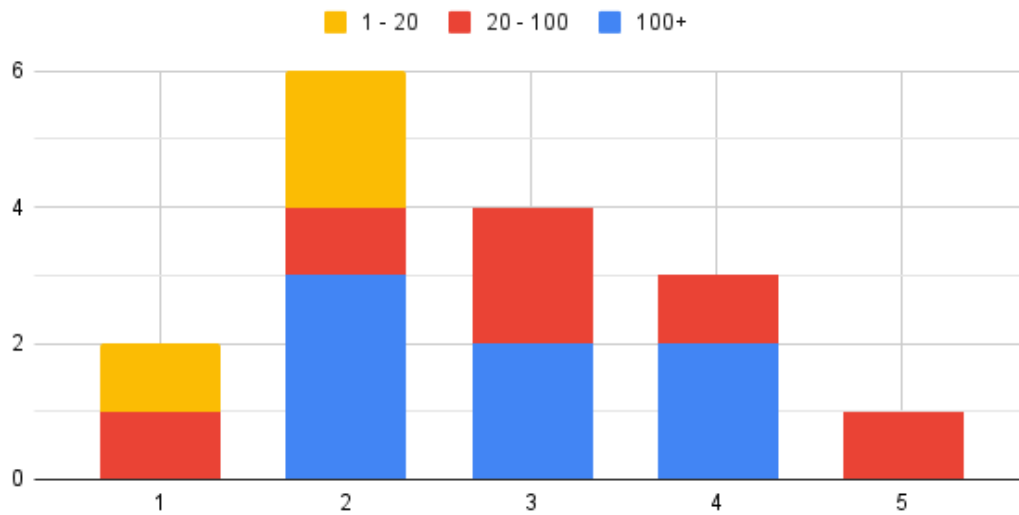


Figura 4.14: Integrazione team QA all'inizio dello sviluppo in base alla dimensione dell'azienda

Parere integrazione team QA all'inizio dello sviluppo in base al Ruolo

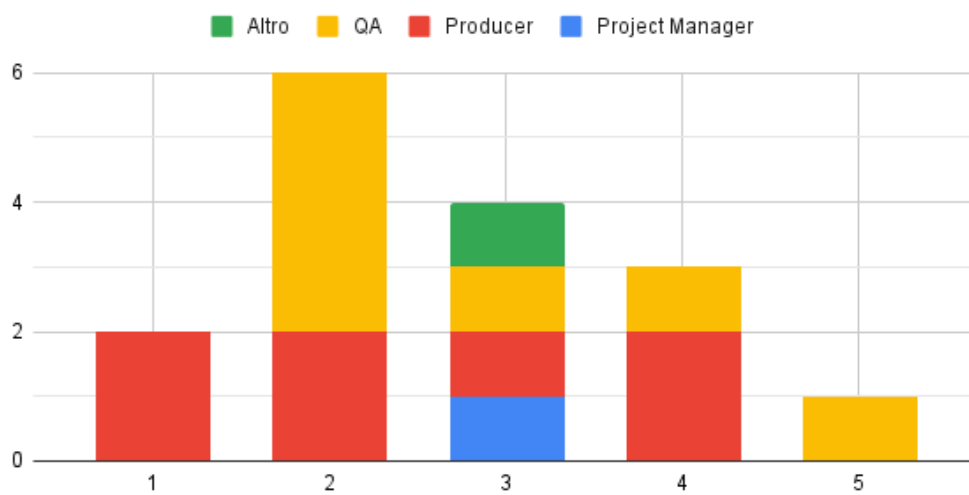


Figura 4.15: Parere integrazione team QA all'inizio dello sviluppo in base al Ruolo

Le risposte alla domanda sull'utilità per un QA tester di possedere competenze di programmazione evidenziano una valutazione generalmente positiva (media 3,44, deviazione standard 1,21), ma con differenze significative tra i partecipanti in base a vari fattori. Le aziende che adottano sia il testing manuale che automatizzato attribuiscono un punteggio più elevato (media 3,85) rispetto a quelle che utilizzano esclusivamente il testing manuale (media 1,67).

Per quanto riguarda la dimensione aziendale, le piccole e medie imprese riportano una media più alta (3,83), indicando che in contesti con risorse limitate, la versatilità dei tester è particolarmente apprezzata. Tuttavia, è importante notare che nelle piccole aziende la deviazione standard è molto alta (2,1), con le risposte distribuite tra due valori positivi e una sul "totale disaccordo". Al contrario, nelle aziende più grandi, la media si riduce a 3,14, probabilmente a causa di una maggiore specializzazione dei ruoli (Fig. 4.16). Geograficamente, i partecipanti italiani considerano queste competenze più utili (media 3,83) rispetto ai partecipanti esteri (media 3,2). Questo potrebbe essere attribuito anche alle dimensioni aziendali, poiché tra le risposte italiane si riscontrano esclusivamente aziende di dimensioni medie e piccole.

Parere sull' "Utilità del QA tester nel programmare" in base alla dimensione dell'azienda

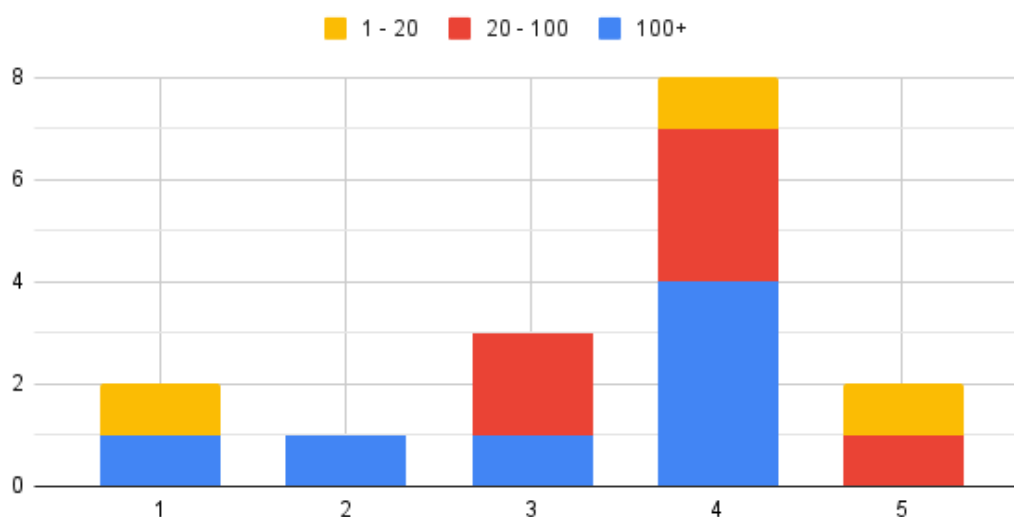


Figura 4.16: Parere sull' "Utilità del QA tester nel programmare" in base alla dimensione dell'azienda

Dal punto di vista dei ruoli, i membri della QA attribuiscono un punteggio superiore (media 3,57) rispetto ai Producer (media 3,14), suggerendo che i tester percepiscono un beneficio diretto nell'applicare queste competenze (Fig. 4.17).

Parere sull' "Utilità del QA tester nel programmare" in base al Ruolo

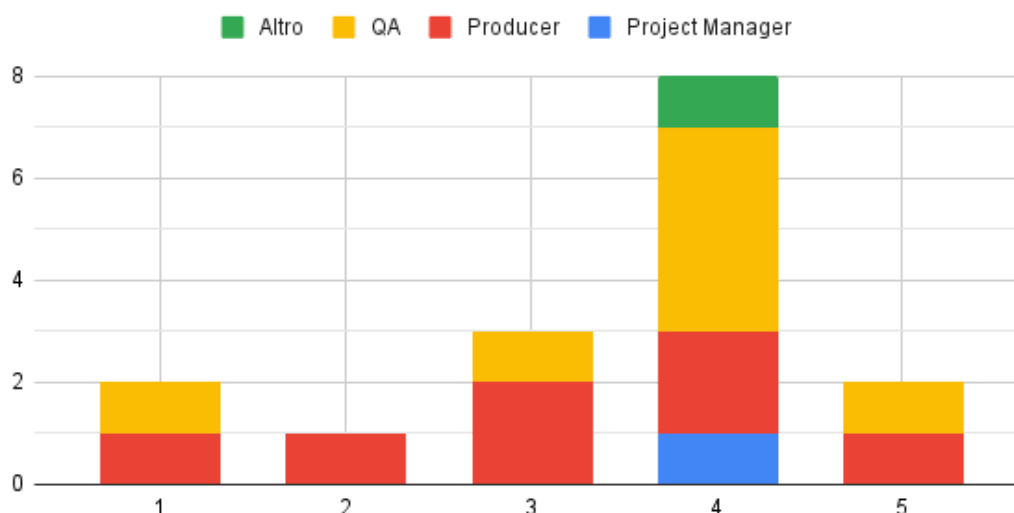


Figura 4.17: Parere sull' "Utilità del QA tester nel programmare" in base al Ruolo

Infine, l'esperienza nell'industria mostra che i partecipanti con più di 10 anni di esperienza attribuiscono il punteggio più alto (media 4,00), mentre i professionisti con meno esperienza registrano medie leggermente inferiori (3,4 per 2-5 anni, 3,0 per 6-10 anni). Questo dato suggerisce che una maggiore esperienza permette di apprezzare meglio l'impatto delle competenze di programmazione nel migliorare l'efficienza e la collaborazione nei processi di QA.

Prima di procedere con l'analisi delle domande aperte, è significativo osservare come solo un numero esiguo di aziende adottano un Quality Plan, con cinque risposte affermative rispetto a undici negative. Analizzando le risposte secondo le categorie utilizzate per l'intera indagine, emerge che tra le aziende che adottano un Quality Plan vi sono sia quelle che impiegano test manuali sia quelle che combinano test manuali e automatizzati, con una prevalenza di queste ultime. Inoltre, i rispondenti che confermano l'adozione di un Quality Plan appartengono a tutte e tre le fasce di esperienza professionale considerate, operano in aziende di dimensioni

differenti e sono presenti sia in Italia che all'estero.

Per quanto riguarda le motivazioni che portano a fare un Quality Plan le uniche risposte sono:

- “Cicli di QA e ciclo di correzione” - fornita da un rispondente che svolge test solamente manuali;
- “C’è una persona dedicata che si concentra maggiormente sulla qualità del gioco, mentre altri si assicurano che funzioni meccanicamente come previsto” - fornita da un rispondente che svolge sia test solamente manuali che automatici.

Purtroppo queste risposte non spiegano meglio come avviene un Quality Plan.

Per quanto riguarda le risposte negative possiamo osservare che alcuni sottolineano come la mancanza di coinvolgimento nelle fasi iniziali dello sviluppo costringa il QA a operare in modo più reattivo, limitando la possibilità di pianificare in anticipo. Viene inoltre evidenziato che problemi di comunicazione, scarsa organizzazione e assenza di documentazione adeguata, uniti alla difficoltà del project management nel chiudere le feature e definire chiaramente l’ambito del progetto, impediscono l’implementazione di un Quality Plan fino alle ultime fasi dello sviluppo.

Diverse risposte suggeriscono che il Quality Plan non viene adottato perché non rientra nei processi standard o perché le priorità della leadership non sono allineate con la necessità di una pianificazione strutturata della qualità. In altri casi, i progetti vengono considerati gestibili anche senza un piano di qualità formale, poiché si ritiene che il testing intensivo non sia necessario. Un altro punto di vista riguarda l’esperienza accumulata dai membri senior del team, la quale consente di operare senza la necessità di documentare formalmente le pratiche di QA. Infine, alcuni rispondenti non forniscono una motivazione chiara o dichiarano semplicemente di non adottare un Quality Plan.

L’analisi delle risposte aperte relative alla RQ2 è stata condotta seguendo lo stesso approccio usato per la RQ1.

1. Codifica aperta – identificazione dei concetti chiave

Il file contenente le risposte aperte è stato analizzato per individuare parole e frasi ricorrenti.

Il prompt ha prodotto i risultati presenti nella figura 4.18.



Figura 4.18: Concetti chiave RQ2

2. Codifica assiale – raggruppamento dei concetti in categorie

L'analisi ha evidenziato le seguenti categorie:

- **Integrazione dell'automazione nei processi di sviluppo**

- Concetti: Production, Development, Phase, Time, Project.

Questi termini descrivono l'automazione come parte del ciclo di sviluppo del gioco, sottolineando il suo impatto sui tempi e sulle fasi del progetto.

- **Ruolo del team di QA**

- Concetti: Team, Teams, Involved.

Indicano la centralità del team di Quality Assurance nel processo e il loro coinvolgimento nella collaborazione interfunzionale.

- **Collaborazione tra reparti**

- Concetti: Other, Depends, Also.

Riflettono l'interazione tra il QA e altri reparti, evidenziando la necessità di dipendenze e coordinamento.

- **Efficacia dell'automazione**

- Concetti: Tests, Results.

Focalizzano sull'obiettivo dell'automazione nel QA, ovvero migliorare i test e ottenere risultati più affidabili.

- **Impatto sul prodotto finale**

- Concetti: Game.

Indica che l'automazione non è solo un miglioramento interno ai processi, ma ha effetti diretti sulla qualità del videogioco finale.

3. **Codifica selettiva** – identificazione dei temi centrali

Sono stati evidenziati i seguenti temi:

1. L'automazione come parte integrante del ciclo di sviluppo
2. Il ruolo strategico del team di QA e la collaborazione interfunzionale
3. L'efficacia dell'automazione nel migliorare la qualità del prodotto finale

Dopo aver utilizzato lo strumento e letto tutte le risposte, è possibile approfondire ulteriormente i temi trattati:

L'impatto dell'automazione sul lavoro quotidiano dei tester varia a seconda del livello di integrazione nei processi di sviluppo. Dalle risposte emergono due principali scenari:

- **Automazione integrata nella pipeline:** In alcuni casi, i test automatici vengono eseguiti a ogni iterazione dello sviluppo, permettendo ai QA tester di concentrarsi su test più complessi o su attività di validazione manuale.
- **Assenza di automazione o impatto trascurabile:** Alcuni team non utilizzano l'automazione in modo significativo, affidandosi ancora a metodi tradizionali. In alcuni casi, questa scelta è dettata dalla mancanza di strumenti o dalla preferenza per il testing manuale.

Alcuni partecipanti hanno anche evidenziato vantaggi specifici derivanti dall'automazione, tra cui:

- Maggiore stabilità delle build, con versioni aggiornate e testate automaticamente prima del rilascio.
- Disponibilità di strumenti di visualizzazione avanzati per il monitoraggio delle performance e della posizione delle risorse di gioco.
- Riduzione del tempo dedicato a test ripetitivi, permettendo ai tester di concentrarsi su aspetti più critici del gameplay.

Successivamente è stato evidenziato che la comunicazione dei risultati dei test automatizzati ai diversi reparti avviene attraverso diverse strategie, che possono essere classificate in:

- **Integrazione nei sistemi CI/CD:** In molte aziende, i test automatici sono integrati nei processi di Continuous Integration, permettendo agli sviluppatori di visualizzare immediatamente i risultati in caso di fallimento.
- **Report automatici e notifiche:** Alcuni team ricevono notifiche istantanee sugli errori critici, report via email o aggiornamenti direttamente nei launcher interni.
- **Gestione manuale dei risultati:** In alcuni casi, i risultati devono essere elaborati e comunicati manualmente dai tester attraverso report dettagliati o strumenti di tracking come Jira.
- **Assenza di comunicazione strutturata:** Alcuni team non danno priorità all'analisi dei risultati dei test automatizzati, rendendo inefficace il loro utilizzo per migliorare la qualità complessiva del prodotto.

L'uso di strumenti come Jira e l'integrazione con l'intelligenza artificiale per la generazione di report visivi emergono come pratiche comuni tra le aziende che danno maggiore importanza all'automazione nei processi di QA.

In generale, la comunicazione tra il team di QA e gli altri team di sviluppo viene gestita in modi diversi a seconda dell'organizzazione e delle pratiche adottate dai singoli gruppi di lavoro. Alcuni evidenziano l'importanza dell'integrazione diretta del QA all'interno dei team di sviluppo, permettendo così un'interazione continua e una collaborazione più efficace, per evitare che informazioni importanti vadano

perse. Altri sottolineano il ruolo fondamentale delle riunioni, come quelle di design e sviluppo, nonché degli incontri di gestione del progetto, a cui il QA partecipa attivamente per rimanere sempre allineato con le altre figure coinvolte.

Diversi rispondenti mettono in evidenza l'uso di strumenti di comunicazione come Slack, Discord e Jira per mantenere un flusso informativo costante, mentre altri ritengono che meeting regolari, report di stato e pratiche strutturate come daily stand-up e retrospettive siano essenziali per garantire il coordinamento. Alcuni sottolineano l'importanza di trattare il QA come una parte integrante della pianificazione tecnica e produttiva, coinvolgendolo nella definizione della roadmap e nelle discussioni sulla capacità operativa del team.

C'è anche chi osserva che la dimensione del team influisce sulla qualità della comunicazione: gruppi più piccoli facilitano un'interazione più fluida, ma ciò non elimina del tutto il rischio di perdita di dati nel flusso di lavoro.

Infine, un aspetto chiave che influisce sull'efficacia dell'integrazione dell'automazione è il momento in cui il team QA entra a far parte del processo di sviluppo. Le risposte evidenziano tre principali approcci:

- **QA presente sin dalle prime fasi:** Alcune aziende coinvolgono il QA fin dalla fase di pre-produzione o prototipazione, permettendo di identificare criticità fin dalle fasi iniziali e sviluppare test automatizzati in parallelo allo sviluppo del gioco.
- **QA introdotto durante la produzione:** In molti casi, il QA inizia a lavorare quando la produzione è già avanzata, limitando le possibilità di influenzare il design e l'architettura del gioco in ottica di testabilità.
- **QA coinvolto tardivamente:** In alcuni scenari, il QA entra nel processo solo nelle fasi finali, limitando drasticamente l'efficacia dell'automazione e aumentando il rischio di problemi critici non individuati in tempo.

Diversi partecipanti hanno sottolineato che un'integrazione precoce del QA sarebbe ideale per garantire un miglior controllo della qualità e un'efficace implementazione dell'automazione.

Risultati RQ3

Le risposte relative alle domande della sezione RQ3 sono state analizzate tramite le stesse identiche categorie della sezione precedente, ossia Manuale ed Entrambi, con riferimento alle modalità di esecuzione dei test, la dimensione aziendale, la posizione geografica delle aziende coinvolte, il ruolo professionale dei rispondenti e gli anni trascorsi nell'industria.

Le risposte alla domanda sull'efficacia dell'automazione nel ridurre il numero di bug non rilevati durante lo sviluppo mostrano una percezione complessivamente positiva (media 3,81, deviazione standard 0,91), con una prevalenza di punteggi tra 3 e 5 (Fig. 4.19). Le aziende che utilizzano sia il testing manuale che quello automatizzato attribuiscono maggiore fiducia all'automazione (media 3,92) rispetto a quelle che adottano esclusivamente il testing manuale (media 3,33). Sebbene la differenza non sia particolarmente marcata, questo dato, insieme a risposte precedenti, suggerisce una crescente apertura verso l'automazione da parte di queste ultime.

Frequenza su l'automazione può ridurre il numero di bug non rilevati durante lo sviluppo

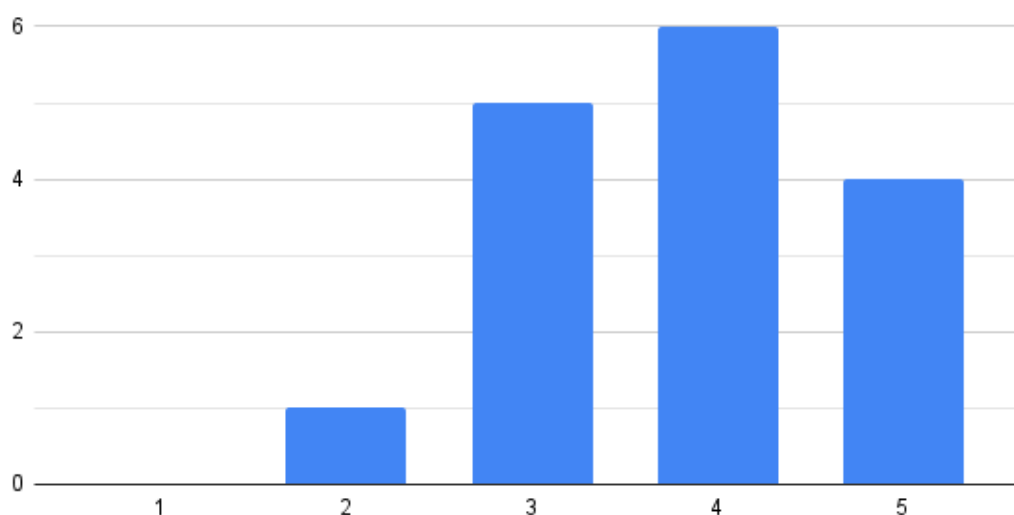


Figura 4.19: Frequenza su l'automazione può ridurre il numero di bug non rilevati durante lo sviluppo

Analizzando la dimensione delle aziende, quelle più piccole (1-20 dipendenti) presentano il punteggio medio più alto (4,33). Aziende di dimensioni maggiori (100+ dipendenti) confermano comunque una valutazione positiva (media 4,14), mentre quelle di medie dimensioni (20-100 dipendenti) mostrano una percezione meno entusiasta (media 3,16). Geograficamente, i partecipanti esteri esprimono maggiore fiducia nell'automazione (media 4,00) rispetto ai partecipanti italiani (media 3,5), suggerendo una possibile apertura maggiore all'automazione.

Dal punto di vista dei ruoli, i membri del QA mostrano la maggiore convinzione dell'utilità dell'automazione (media 3,86), seguiti dai Producer (media 3,71). Infine, l'esperienza nell'industria videoludica influisce sulla percezione: i partecipanti con 2-5 anni e oltre 10 anni di esperienza registrano medie elevate (rispettivamente 4,00 e 3,8).

È interessante analizzare il profilo della persona che ha assegnato il voto più basso (2). Si tratta di un Producer con oltre 10 anni di esperienza, che lavora in Italia in un'azienda di medie dimensioni priva di un team dedicato al QA, ma che utilizza comunque sia il testing manuale che quello automatizzato. Questo dato, unito a una risposta precedente, suggerisce che la percezione negativa dell'utilità dell'automazione potrebbe essere influenzata da fattori culturali e dall'esperienza maturata in contesti specifici. La mancanza di un team di QA strutturato potrebbe inoltre contribuire a una visione meno favorevole dell'efficacia dell'automazione nel raggiungere l'efficienza desiderata. Sebbene si tratti di un singolo caso e non sia sufficiente per trarre conclusioni definitive, il risultato rappresenta un elemento interessante.

Le risposte alla domanda sull'impatto degli strumenti di automazione nella riduzione dei tempi di sviluppo e rilascio dei videogiochi evidenziano una percezione generalmente positiva, con una media complessiva di 3,56 (deviazione standard 1,03). Tuttavia, il giudizio varia sensibilmente in base alle caratteristiche dei partecipanti e delle loro aziende. Le aziende che utilizzano sia il testing manuale che quello automatizzato esprimono un'opinione più favorevole (media 3,69) rispetto a quelle che adottano solo il testing manuale (media 3,00), confermando che l'integrazione dell'automazione può essere percepita come un vantaggio strategico.

Le piccole aziende (1-20 dipendenti) mostrano il punteggio medio più alto (media 4,33), suggerendo che l'automazione potrebbe essere particolarmente utile in contesti con risorse limitate. Al contrario, le aziende di medie dimensioni (20-100 dipendenti) e grandi dimensioni (100+ dipendenti) registrano medie leggermente

inferiori (rispettivamente 3,33 e 3,43) e si assestano su un sentimento neutro. Geograficamente, i partecipanti italiani valutano l'automazione in modo più positivo (media 3,83) rispetto ai partecipanti esteri (media 3,40).

Tra i ruoli, i professionisti del QA mostrano una maggiore fiducia nell'automazione (media 3,86), mentre i Producer evidenziano una percezione meno favorevole (media 3,14). Quest'ultima categoria registra un picco di risposte negative, probabilmente dovuto alla difficoltà di integrare l'automazione in processi già consolidati. L'introduzione dell'automazione, infatti, può essere percepita come un fattore che incrementa i tempi di sviluppo e rilascio, un aspetto particolarmente critico dal punto di vista dei Producer, che tendono a privilegiare l'efficienza a breve termine. Nonostante ciò, si rileva comunque una certa apertura verso l'automazione, che potrebbe crescere con l'evolversi delle pratiche e l'ottimizzazione dei processi (Fig. 4.20).

L'automazione aiuta a ridurre i tempi di sviluppo e rilascio in base al Ruolo

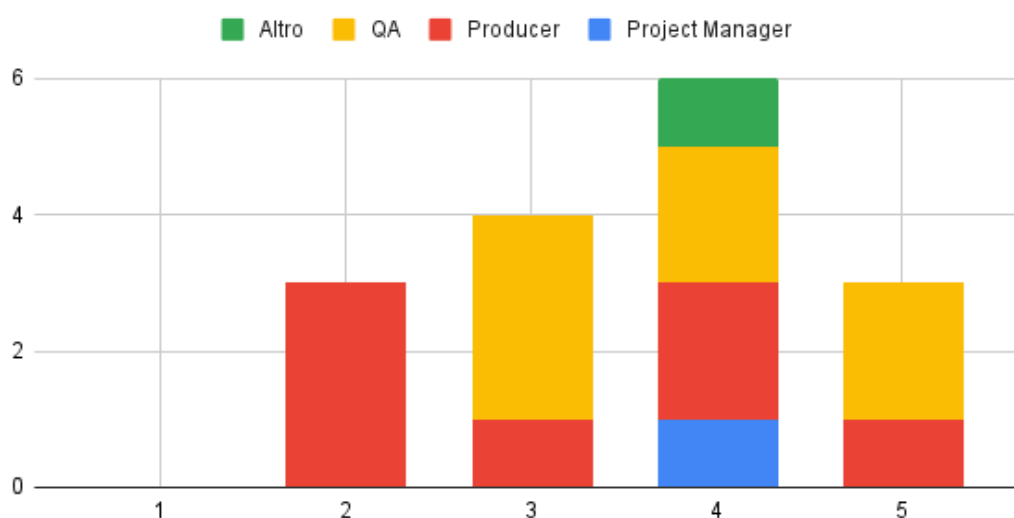


Figura 4.20: L'automazione aiuta a ridurre i tempi di sviluppo e rilascio in base al Ruolo

Infine, per quanto riguarda l'esperienza nell'industria, i partecipanti con 2-5 anni di esperienza presentano la media più alta (3,8), seguiti da quelli con oltre 10 anni (3,6). Questo suggerisce che sia i professionisti meno esperti che quelli più veterani riconoscono i benefici dell'automazione, sebbene in modo diverso, rispetto a chi ha un'esperienza intermedia (6-10 anni, media 3,33).

Le risposte alla domanda sull'impatto futuro dell'intelligenza artificiale nel testing dei videogiochi mostrano una percezione tendenzialmente negativa ma diversificata, con una media complessiva di 3,19 (deviazione standard 1,22). La distribuzione riflette un'ampia varietà di opinioni, con un numero significativo di partecipanti che esprime scetticismo (7 risposte per il punteggio 2) e un discreto gruppo che ritiene il cambiamento significativo (4 risposte per il punteggio 4 e 3 per il punteggio 5). Le aziende che utilizzano solo testing manuale mostrano un'opinione leggermente più favorevole (media 3,33) rispetto a quelle che combinano testing manuale e automatizzato (media 3,15).

Dal punto di vista dimensionale, le piccole aziende (1-20 dipendenti) dimostrano il maggiore entusiasmo (media 4,33). Questo entusiasmo può essere attribuito alla maggiore flessibilità organizzativa e alla possibilità di integrare più facilmente strumenti innovativi, come l'IA, nella loro pipeline di lavoro. Le grandi aziende (100+ dipendenti), invece, registrano il punteggio medio più basso (media 2,71), suggerendo maggiore cautela o una percezione di difficoltà nell'integrare nuove tecnologie su larga scala. Le aziende di medie dimensioni registrano una media di 3,17, con risposte distribuite lungo quasi tutto lo spettro dei valori (Fig. 4.21). Geograficamente, i partecipanti italiani hanno una percezione più positiva (media 3,5) rispetto a quelli esteri (media 3).

L'AI cambierà il modo in cui verranno testati i videogiochi in base alla dimensione dell'azienda

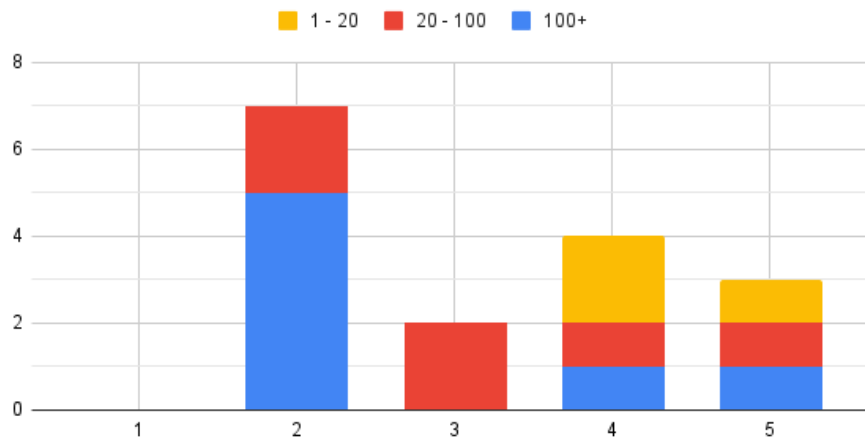


Figura 4.21: L'AI cambierà il modo in cui verranno testati i videogiochi in base alla dimensione dell'azienda

Tra i ruoli, i Producer esprimono una valutazione più alta (media 3,29) rispetto ai QA (media 3). Tuttavia, non emerge un consenso unanime tra i vari ruoli. Infine, i professionisti con oltre 10 anni di esperienza mostrano il maggior ottimismo (media 4), mentre i partecipanti con 2-5 anni di esperienza sono più scettici (media 2,6). Questo divario potrebbe riflettere una maggiore capacità, da parte dei professionisti più esperti, di cogliere i vantaggi concreti delle nuove tecnologie e di comprendere come queste possano supportare i team di sviluppo.

I rispondenti hanno fornito ulteriori motivazioni relative alle risposte date a questa domanda. Le opinioni si possono racchiudere nei seguenti temi:

- **Automazione e riduzione del personale QA:** Molti compiti noiosi e ripetitivi non richiederanno più l'intervento umano. Il ruolo del QA diventerà più orientato alla regolazione, gestione e analisi dei dati prodotti dagli agenti automatizzati che eseguono i test. Tuttavia, questo comporta dei rischi: la disciplina si ridurrà in termini di numero di addetti e aumenterà la possibilità di trascurare problemi individuabili con metodi manuali.
- **Limitazioni dell'IA nel testing:** L'IA non può ragionare su cosa sia un bug o meno, soprattutto considerando la varietà dei giochi. Alcune tecniche basate sull'IA possono essere d'aiuto, ma non rivoluzioneranno il settore. Per fare un confronto, nel testing web l'IA non ha ancora sostituito gli strumenti esistenti, quindi è difficile credere che lo farà prima nel settore dei videogiochi.
- **Qualità del testing:** L'IA potrebbe portare a test di scarsa qualità, falsi bug segnalati, tempi di sviluppo più lunghi e un generale calo della qualità del prodotto. L'assistenza dell'IA spesso implica la necessità di correggere e ricontrollare il suo lavoro, riducendo l'effettivo risparmio di tempo. Tuttavia, con un buon framework di automazione e una documentazione adeguata, in futuro l'IA potrebbe essere utilizzata per creare test automatizzati.
- **Automazione di test specifici, grafici e di gameplay:** L'IA potrebbe automatizzare test più soggettivi o difficili da codificare, come gli aspetti visivi e audio, nonché i controlli sulle risorse grafiche. Sono stati sviluppati modelli di apprendimento automatico per testare i sistemi di combattimento nei videogiochi, ma il loro costante riaddestramento, dovuto ai continui cambiamenti nel gioco, ha reso l'investimento poco conveniente. Tuttavia, alcuni test ripetitivi e non tecnici, come verificare che non sia possibile uscire dalla mappa o assicurarsi che si possa raggiungere determinati punti nel gioco, potrebbero essere efficacemente gestiti da un'IA.

Le risposte alla domanda sull'efficacia delle nuove tecnologie, come l'intelligenza artificiale, nel migliorare i processi di testing automatizzato mostrano un livello di ottimismo moderato, con una media complessiva di 3,31 (deviazione standard 1,14). La distribuzione evidenzia una prevalenza di valutazioni medie o alte (6 risposte per il punteggio 4 e 2 per il punteggio 5), bilanciate da un minor numero di opinioni più scettiche (4 risposte tra 1 e 2). Il risultato è particolarmente interessante se confrontato con la domanda precedente. I rispondenti, infatti, tendono a ritenere che l'IA non rivoluzionerà il modo di testare i giochi, ma che il suo valore risiederà principalmente nel migliorare l'efficienza dei test automatizzati. Questo suggerisce una visione pragmatica dell'adozione dell'IA: più che un cambiamento radicale, i partecipanti si aspettano un'evoluzione incrementale, focalizzata sull'ottimizzazione dei processi esistenti. Tale percezione potrebbe riflettere sia un certo scetticismo verso il potenziale trasformativo dell'IA, sia una consapevolezza delle sue attuali limitazioni nell'ambito del game testing.

Le aziende che utilizzano esclusivamente testing manuale hanno espresso un'opinione leggermente più positiva (media 3,67) rispetto a quelle che integrano sia testing manuale che automatizzato (media 3,23).

Tra le dimensioni aziendali, le piccole aziende (1-20 dipendenti) e quelle medie (20-100 dipendenti) sono le più ottimiste (media 4,33 e 3,83), in linea con le risposte alla domanda precedente. Al contrario, le grandi aziende (100+ dipendenti) sono più caute (media 2,43) e un picco di risposte concentrate sul valore 2 (Fig. 4.22). Geograficamente, i partecipanti italiani mostrano una maggiore fiducia nell'efficacia delle nuove tecnologie, con una media di 4. Al contrario, i colleghi esteri registrano una media inferiore, pari a 2,9, con un picco sul valore 4, ma una distribuzione più ampia su tutto lo spettro dei punteggi.

I professionisti del QA registrano una media di 3,14, mentre i Producer si mostrano leggermente più ottimisti, con una media di 3,29. Entrambe le categorie presentano una distribuzione delle risposte su tutto lo spettro dei valori, riflettendo opinioni diversificate all'interno di ciascun ruolo. Infine, l'esperienza nell'industria evidenzia un pattern chiaro: i partecipanti con oltre 10 anni di esperienza sono i più positivi (media 4,2), mentre i meno esperti (2-5 anni) mostrano maggiore scetticismo (media 2,8). Questi valori sono completamente allineati con la domanda precedente.

Quanto l'AI può migliorare l'efficacia del testing automatizzato in base alla dimensione dell'azienda

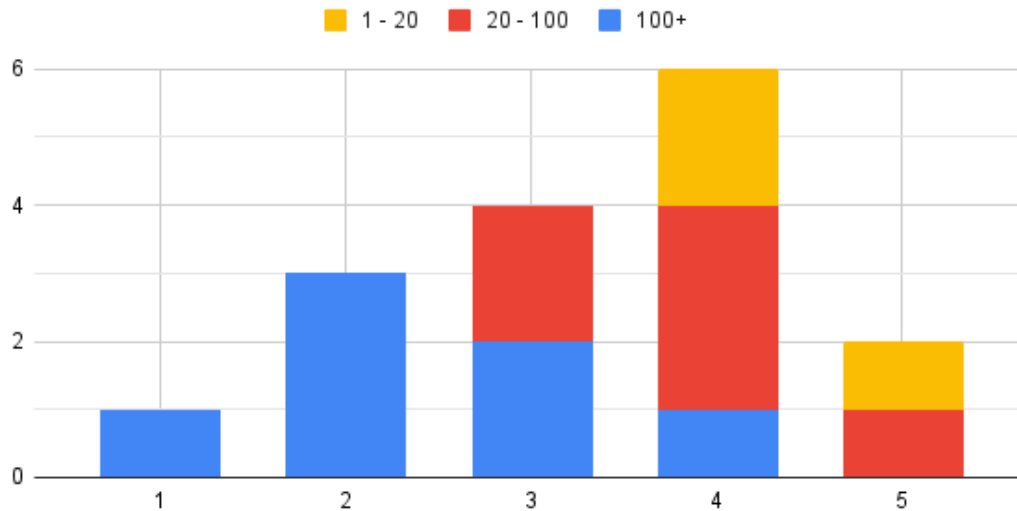


Figura 4.22: Quanto l'AI può migliorare l'efficacia del testing automatizzato in base alla dimensione dell'azienda

Le risposte alla domanda sull'impatto dell'IA nel migliorare la collaborazione tra il team di QA e gli sviluppatori indicano un generale scetticismo, con una media complessiva di 2,38 (deviazione standard 1,02) (Fig. 4.23). Mettendo questo risultato in relazione con le altre domande, emerge chiaramente come l'IA venga percepita principalmente come uno strumento utile per il testing, ma non per favorire o migliorare il rapporto tra sviluppatori e team di QA. Le aziende che adottano esclusivamente testing manuale mostrano una media di 3, superiore a quelle che utilizzano entrambi i tipi di testing (2,23).

Per quanto riguarda la dimensione aziendale, le piccole imprese (1-20 dipendenti) sono più ottimiste (media 3,67), in linea con le risposte delle domande precedenti. Le medie imprese (20-100 dipendenti) mostrano il livello di fiducia più basso (media 1,83), che risulta inferiore rispetto alle domande precedenti. Le grandi aziende (100+ dipendenti), infine, mantengono una percezione negativa in linea con le domande precedenti, registrando una media di 2,29. Dal punto di vista geografico, non emergono differenze significative: i partecipanti italiani riportano una media di 2,33, simile a quella dei partecipanti esteri (2,4).

Frequenza su l'AI può migliorare la collaborazione tra il team di QA e gli sviluppatori

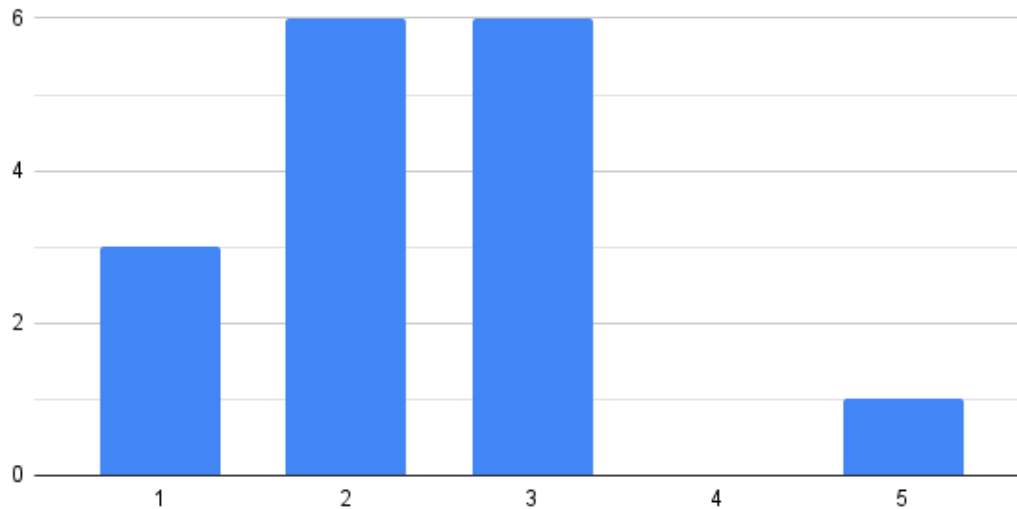


Figura 4.23: Frequenza su l'AI può migliorare la collaborazione tra il team di QA e gli sviluppatori

Tuttavia, i ruoli professionali offrono una prospettiva interessante: i QA (media 2,29) esprimono valutazioni lievemente inferiori rispetto ai Producer (media 2,57). Chi ha oltre 10 anni di esperienza è leggermente più fiducioso (media 2,6), nonostante le loro risposte siano distribuite su tutti i valori. Al contrario, i meno esperti (2-5 anni) e i partecipanti con 6-10 anni riportano valori inferiori (media rispettivamente di 2,4 e 2,17). La distribuzione su tutti i valori suggerisce che, indipendentemente dall'esperienza, le opinioni rimangono diversificate e influenzate da fattori personali e organizzativi.

Infine è possibile affermare, anche tramite l'analisi delle domande aperte, che l'intelligenza artificiale è percepita come un'opportunità con un potenziale significativo per il QA, ma la sua adozione solleva diverse questioni. Se da un lato può migliorare l'efficienza e la copertura dei test, dall'altro la sua implementazione richiede competenze specifiche che non tutti i team possiedono. Inoltre, vi è una generale incertezza su quanto l'AI possa effettivamente sostituire il lavoro umano nei processi di QA più complessi, in particolare per la valutazione di aspetti estetici e sensoriali.

Passando all'analisi delle risposte aperte relative alla RQ3 è stata condotta seguendo lo stesso approccio usato per la RQ1 e per la RQ2.

1. Codifica aperta – identificazione dei concetti chiave

Il file contenente le risposte aperte è stato analizzato per individuare parole e frasi ricorrenti.

Il prompt ha prodotto i risultati presenti nella figura 4.24.

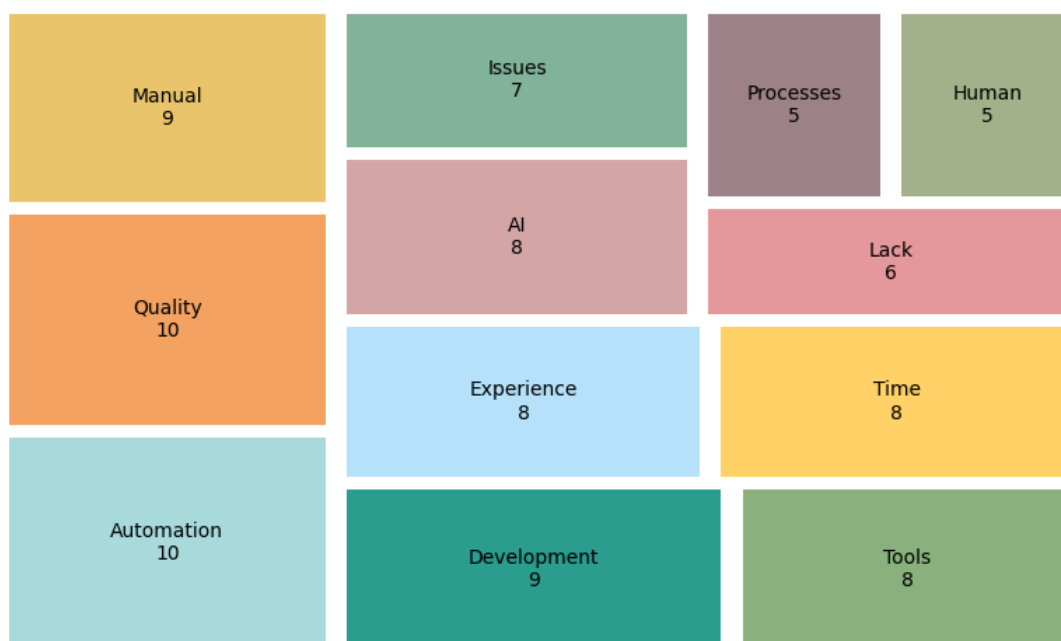


Figura 4.24: Concetti chiave RQ3

2. Codifica assiale – raggruppamento dei concetti in categorie

L'analisi ha evidenziato le seguenti categorie:

- **Sfide nell'automazione del QA**
 - Concetti: Automation, Manual, Testing, Tools, Issues, Lack.

Questa categoria raccoglie gli aspetti critici della transizione dall'approccio manuale all'automazione, comprese le problematiche e le difficoltà nell'adozione di strumenti automatizzati.

- **Impatto sull'organizzazione del lavoro e sui tempi**

- Concetti: Time, Work, More, Very, Up.

Riguarda il modo in cui l'automazione influisce sulla gestione del lavoro quotidiano e sulle tempistiche dei processi di QA.

- **Qualità e controllo nei processi di sviluppo**

- Concetti: Quality, Development, Experience.

Questa categoria si concentra su come l'automazione impatti la qualità del prodotto e il controllo nel ciclo di sviluppo.

- **Evoluzione tecnologica e impatto dell'AI**

- Concetti: AI, Tools, Games, Game.

Riguarda il ruolo delle nuove tecnologie, in particolare l'intelligenza artificiale, e il loro impatto nel settore QA per i videogiochi.

- **Difficoltà e barriere nel cambiamento**

- Concetti: Lack, Issues.

Identifica le principali difficoltà percepite dal team QA, come la mancanza di risorse o la resistenza al cambiamento.

3. Codifica selettiva – identificazione dei temi centrali

Sono stati evidenziati i seguenti temi:

1. Transizione dall'approccio manuale all'automazione: resistenze e criticità
2. Impatto dell'automazione sulla gestione del lavoro e delle tempistiche
3. Il ruolo dell'AI nel QA: opportunità e incertezze
4. Qualità e controllo nel ciclo di sviluppo
5. Difficoltà nell'adozione di nuove tecnologie e barriere al cambiamento

Dopo aver utilizzato lo strumento e letto tutte le risposte, è possibile approfondire ulteriormente i temi trattati:

Un primo elemento riguarda il bilanciamento tra automazione e testing manuale, la cui dicotomia, già evidenziata nelle sezioni precedenti, continua a emergere.

Sebbene l'automazione sia ritenuta essenziale per ottimizzare i tempi di esecuzione e ridurre errori umani nei test ripetitivi, molte attività del QA non sono facilmente automatizzabili. In particolare, aspetti come il game feel, la qualità dell'esperienza utente e la valutazione dell'equilibrio del gameplay richiedono ancora un intervento umano. Il testing manuale continua quindi a essere percepito come indispensabile, soprattutto per la sua capacità di individuare problemi che vanno oltre la mera esecuzione di script predefiniti.

Un ulteriore tema sono le limitazioni degli strumenti di automazione attualmente disponibili. La frammentazione del settore e l'assenza di standard condivisi rendono complessa l'integrazione di tool efficaci, costringendo molti team a sviluppare soluzioni interne o a combinare più strumenti con livelli di compatibilità variabili. Inoltre, diversi partecipanti segnalano che gli attuali software di automazione tendono a essere più adatti a test funzionali piuttosto che a valutazioni qualitative, limitandone l'efficacia nel contesto videoludico.

Un tema è rappresentato dai vincoli organizzativi e dalle pressioni temporali. Molti team di QA operano sotto scadenze strette, il che riduce la possibilità di sperimentare nuove soluzioni di automazione o di investire in formazione avanzata. La mancanza di tempo per una corretta implementazione e manutenzione dei test automatizzati può limitare i benefici stessi dell'automazione, portando alcuni team a preferire metodi di testing più tradizionali e immediatamente efficaci.

Infine, una problematica ricorrente nelle risposte è la natura dinamica dello sviluppo dei videogiochi e l'impatto che i cambiamenti continui hanno sulla vita del QA. Il costante aggiornamento del codice e delle funzionalità può rendere rapidamente obsolete molte automazioni, costringendo i tester a rivedere frequentemente gli script e a intervenire manualmente su situazioni impreviste. Questo fenomeno complica ulteriormente il lavoro del QA, che deve adattarsi a una pipeline di sviluppo spesso frammentata e soggetta a modifiche dell'ultimo minuto. Tale instabilità influisce anche sulla comunicazione tra i team, con i tester che spesso ricevono informazioni incomplete o tardive sulle modifiche in corso, rendendo più difficile una pianificazione efficace dei test. In alcuni casi, questa mancanza di coordinazione porta a situazioni in cui gli stessi bug vengono testati più volte o, al contrario, in cui alcune criticità passano inosservate fino a fasi avanzate dello sviluppo.

Oltre agli aspetti tecnici e organizzativi, un altro elemento critico emerso dall'analisi riguarda il ruolo stesso del QA all'interno del processo di sviluppo. Molti

professionisti del settore percepiscono una scarsa valorizzazione del loro lavoro, sia in termini di riconoscimento economico che di rispetto all'interno del team. Il QA è spesso considerato un ruolo di supporto piuttosto che una parte integrante dello sviluppo del videogioco, nonostante la sua importanza nel garantire la qualità del prodotto finale. Questa percezione influisce sulla motivazione dei tester e, in alcuni casi, sulla loro permanenza nel settore. Una maggiore considerazione del QA, attraverso migliori condizioni economiche, un'integrazione più strutturata nei processi decisionali e una comunicazione più fluida con gli altri reparti, potrebbe contribuire a migliorare significativamente l'efficacia e l'efficienza del testing.

Nel complesso, l'analisi evidenzia, come negli altri sottocapitoli, che l'automazione è vista come una risorsa strategica per il QA, ma non come una soluzione universale. L'approccio emergente sembra orientarsi verso un modello ibrido, in cui l'automazione supporta il testing manuale piuttosto che sostituirlo, con un ruolo crescente delle nuove tecnologie a condizione che siano adattabili alle specificità dello sviluppo videoludico. Tuttavia, affinché l'automazione possa avere un impatto significativo, è necessario affrontare le problematiche legate alla gestione del cambiamento, alla comunicazione tra i vari reparti e al riconoscimento del QA come una componente essenziale del team di sviluppo, sia in termini di rispetto professionale che di adeguata retribuzione.

Risultati Qualitativi

In questo sottocapitolo vengono riassunti i principali argomenti trattati nei risultati qualitativi, che vengono anche integrati e confrontati tra le diverse RQ mediante grafici e tabelle. In particolare, si segue la suddivisione delle codifiche: Aperta, Assiale e Selettiva.

Codifica Aperta

Qui vengono riportati tutti i concetti chiave:

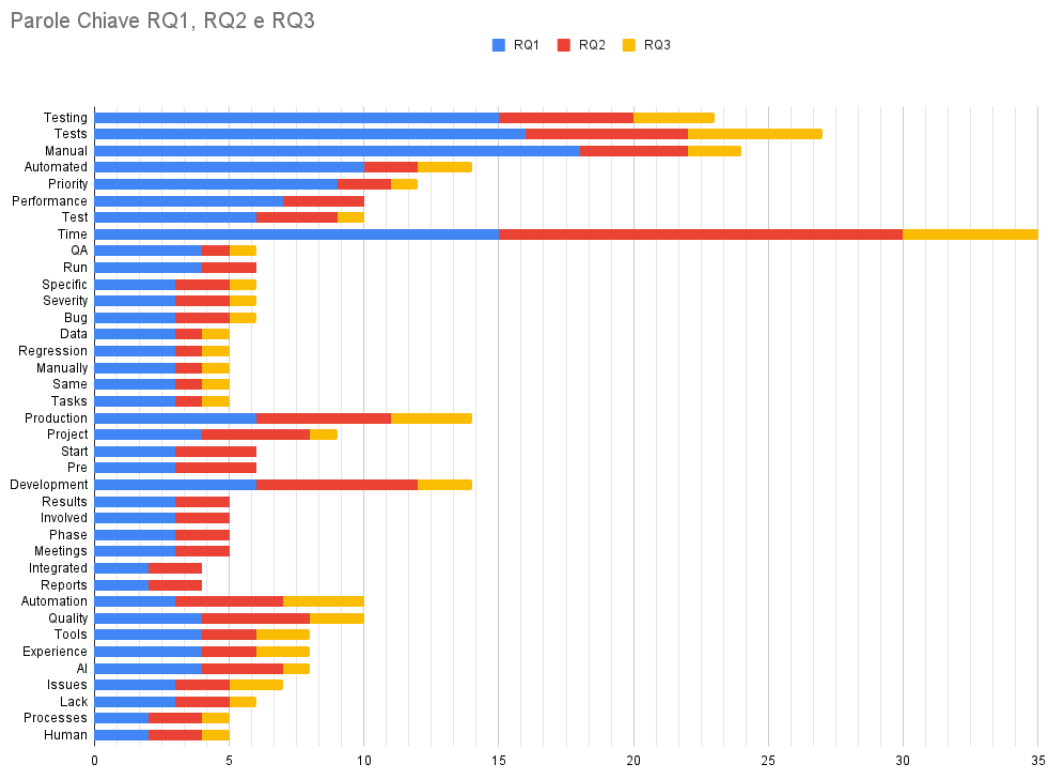


Figura 4.25: Parole Chiave RQ1, RQ2 e RQ3

Parola	RQ1	RQ2	RQ3	Occorrenze
Testing	15	5	3	23
Tests	16	6	5	27
Manual	18	4	2	24
Automated	10	2	2	14
Priority	9	2	1	12
Performance	7	3	0	10
Test	6	3	1	10
Time	15	15	5	35
QA	4	1	1	6
Run	4	2	0	6
Specific	3	2	1	6
Severity	3	2	1	6
Bug	3	2	1	6
Data	3	1	1	5
Regression	3	1	1	5
Manually	3	1	1	5
Same	3	1	1	5
Tasks	3	1	1	5
Production	6	5	3	14
Project	4	4	1	9
Start	3	3	0	6
Pre	3	3	0	6
Development	6	6	2	14
Results	3	2	0	5
Involved	3	2	0	5
Phase	3	2	0	5
Meetings	3	2	0	5
Integrated	2	2	0	4
Reports	2	2	0	4
Automation	3	4	3	10
Quality	4	4	2	10
Tools	4	2	2	8
Experience	4	2	2	8
AI	4	3	1	8
Issues	3	2	2	7
Lack	3	2	1	6
Processes	2	2	1	5
Human	2	2	1	5

Codifica Assiale

In questa sezione vengono presentate tutte le categorie derivanti dai concetti chiave, che sono state anche integrate nelle altre RQ correlate:

Argomento	RQ1	RQ2	RQ3
Tipologie di Testing	✓		
Fattori Decisionali	✓		
Efficienza e Benefici dell'Automazione	✓	✓	
Limitazioni e Sfide del Testing Manuale e Automatizzato	✓	✓	✓
Integrazione dell'automazione nei processi di sviluppo		✓	
Ruolo del team di QA		✓	
Collaborazione tra reparti		✓	
Efficacia dell'automazione	✓	✓	
Impatto sul prodotto finale		✓	✓
Sfide nell'automazione del QA			✓
Impatto sull'organizzazione del lavoro e sui tempi		✓	✓
Qualità e controllo nei processi di sviluppo	✓		✓
Evoluzione tecnologica e impatto dell'AI			✓
Difficoltà e barriere nel cambiamento			✓

Codifica Selettiva

Infine, in questa sezione vengono presentati tutti i temi derivati dai concetti chiave e dalle categorie, che sono stati anche integrati nelle altre RQ correlate:

Temi	RQ1	RQ2	RQ3
Complementarietà tra Testing Manuale e Automatizzato	✓		
Fattori Decisionali: Priorità e Automazione Possibile	✓		✓
Benefici dell'Automazione: Velocità, Ripetibilità e Efficienza	✓		
Limitazioni del Testing Manuale e Automatizzato	✓	✓	✓
L'automazione come parte integrante del ciclo di sviluppo		✓	
Il ruolo strategico del team di QA e la collaborazione interfunzionale		✓	
L'efficacia dell'automazione nel migliorare la qualità del prodotto finale	✓	✓	✓
Transizione dall'approccio manuale all'automazione: resistenze e criticità	✓	✓	✓
Impatto dell'automazione sulla gestione del lavoro e delle tempistiche	✓	✓	✓
Il ruolo dell'AI nel QA: opportunità e incertezze			✓
Qualità e controllo nel ciclo di sviluppo		✓	✓
Difficoltà nell'adozione di nuove tecnologie e barriere al cambiamento		✓	✓

Discussioni

L'obiettivo di questo capitolo è analizzare e integrare tutte le informazioni raccolte fino a questo punto, al fine di fornire risposte complete e approfondite alle domande di ricerca iniziali.

Automazione E Manuale

La domanda 3.4, “Come vengono condotti i test nella tua azienda?”, ha rappresentato un elemento cruciale per la segmentazione dei partecipanti e ha permesso di identificare due categorie principali: "Manuale" ed "Entrambi". I risultati emersi dalla sezione dedicata alla RQ1 evidenziano un prevalente utilizzo combinato di test manuali e automatizzati, con una significativa minoranza che adotta esclusivamente il testing manuale. L'assenza di risposte che segnalano l'uso esclusivo di test automatizzati riflette una tendenza ben consolidata nel settore, dove l'automazione non è vista come una panacea, ma piuttosto come una risorsa complementare al testing manuale.

Per quanto riguarda i tipi di test, la predominanza del Functional Testing tra tutti i partecipanti conferma la centralità di questo test per il mantenimento della stabilità e funzionalità del gioco, mentre la combinazione di Regression e Performance Testing dimostra l'importanza di garantire la qualità anche sotto condizioni di stress e in presenza di modifiche al codice. Al contrario, la minor frequenza del Localization Testing e del Compatibility Testing è legata sia alla fase di sviluppo in cui questi test vengono condotti, sia alla limitata capacità delle piccole aziende di supportare molteplici piattaforme simultaneamente. È importante notare che i test maggiormente automatizzati sono quelli di Performance, Regression, Functional ed End-to-End. Questo evidenzia come le attività più ripetitive siano oggetto di automazione, con l'obiettivo di ridurre il carico di lavoro dei tester, permettendo loro di concentrarsi su aspetti più soggettivi, come l'esperienza utente.

L'adozione di strumenti per il bug tracking evidenzia un altro aspetto distintivo tra le aziende, con Jira che si conferma come lo strumento preferito dalla maggioranza, ma con altre soluzioni come Asana e Mantis che emergono soprattutto nelle realtà più piccole e localizzate, in particolare in Italia. Questo dato riflette una differenza nell'approccio tra aziende più strutturate, che adottano soluzioni più consolidate e

standardizzate, e quelle più piccole, che possono orientarsi su strumenti più accessibili e flessibili. Inoltre, la priorità dei bug nei processi di QA è determinata da una serie di criteri che variano in base al contesto di sviluppo. Il fattore principale è la severità del problema, con i bug più critici—come crash, blocchi della progressione o errori che impediscono la generazione delle build—che vengono risolti con la massima urgenza. Seguono i problemi che compromettono l’usabilità del gioco, l’esperienza dell’utente o il raggiungimento degli obiettivi di sviluppo. Altri fattori chiave includono la frequenza di riproduzione, l’impatto su milestone e release, e la fase di sviluppo in cui il bug viene riscontrato. Alcuni studi adottano sistemi di automazione per assegnare la priorità in base a parametri predefiniti, mentre la decisione finale spesso coinvolge figure chiave del team di produzione. Infine, i problemi estetici o di qualità della vita, pur essendo meno urgenti, restano rilevanti per la rifinitura del prodotto finale. È interessante analizzare quali software di bug tracking vengono utilizzati e i criteri con cui i bug vengono valutati, poiché questi aspetti sono fondamentali per determinare l’integrazione di automazioni nel ciclo di sviluppo. Dalle risposte al questionario si rileva che la percezione generale riguardo all’efficacia dell’automazione nel ridurre il numero di bug non rilevati è positiva. Molti dei software menzionati offrono funzionalità di automazione, come la possibilità di collegare un sistema di versioning per aprire o aggiornare automaticamente un ticket in caso di bug [3]. Questo approccio consente di ottimizzare tempi e risorse cognitive non solo per il team di QA, ma per tutti gli sviluppatori coinvolti.

Per rispondere alla RQ1: "In che modo le aziende di sviluppo videoludico scelgono tra testing manuale e automatizzato, e quali sono i principali fattori che influenzano questa decisione?", i risultati suggeriscono che l’automazione non sostituisce il testing manuale, ma si integra nei processi di sviluppo, contribuendo a ottimizzarne l’efficienza. Le motivazioni alla base della preferenza per il testing manuale in alcune aziende, nonostante la presenza di tecnologie automatizzate, variano a seconda delle dimensioni aziendali, della natura del progetto e della disponibilità di risorse. Altre motivazioni risiedono in elementi come il gameplay e l’esperienza utente, due ambiti estremamente complessi da automatizzare, in cui la presenza di una persona fisica resta essenziale. I partecipanti hanno segnalato che l’automazione richiede un notevole investimento di tempo e risorse, oltre a richiedere aggiornamenti costanti per mantenere la sua efficacia, soprattutto quando i giochi cambiano frequentemente. Le risposte suggeriscono che le piccole e medie aziende, o quelle con team di QA ridotti, preferiscono concentrarsi su test manuali a causa della flessibilità richiesta da progetti in continuo cambiamento, e della difficoltà nel giustificare l’investimento in strumenti automatizzati. Nelle fasi iniziali dello

sviluppo videoludico, i cambiamenti possono essere numerosi. Poiché un videogioco è un prodotto creativo, iterativo e poco standardizzato, salvo alcune eccezioni, risulta complesso identificare fin da subito gli elementi chiave e di successo. Un altro fattore emerso è la scarsa conoscenza dell'automazione, che spesso genera scetticismo nei suoi confronti. Questo atteggiamento è comprensibile considerando l'evoluzione del mercato e le sfide che lo caratterizzano. I giochi AAA richiedono in media circa cinque anni di sviluppo, comportando costi elevati e il rischio di dover affrontare cambiamenti nei trend di mercato. Al contrario, gli studi indie e AA hanno cicli di sviluppo più brevi, risultando generalmente più sostenibili [27]. Di conseguenza, la decisione di investire risorse, tempo e denaro nell'automazione dipende fortemente dalla necessità di bilanciare i tempi di sviluppo e la sostenibilità dell'azienda o della startup.

In tutte le risposte dei partecipanti è emerso un riferimento ai vantaggi dell'automazione. Tra questi, spiccano una maggiore stabilità delle build, grazie a versioni costantemente aggiornate e testate prima del lancio, e strumenti avanzati di visualizzazione per monitorare le performance e la posizione delle risorse di gioco. Inoltre, come già menzionato, l'automazione consente di ridurre significativamente il tempo dedicato ai test ripetitivi, permettendo ai tester di concentrarsi su aspetti più critici del gameplay.

Concludendo, l'automazione, pur offrendo vantaggi in termini di velocità, scalabilità ed efficienza, non può sostituire il testing manuale, che resta fondamentale per garantire la qualità dell'esperienza utente e la coerenza visiva e sonora del gioco. Questo equilibrio tra automazione e manualità emerge come un elemento chiave nella strategia di QA adottata dalle aziende, a dimostrazione che la sinergia tra i due approcci è spesso la via più efficace per garantire un prodotto finale di alta qualità, che soddisfi le aspettative sia del team di sviluppo che degli utenti finali.

Comunicazione e Integrazione

Dall'analisi delle risposte fornite dai partecipanti al questionario emerge un quadro chiaro sulle modalità di integrazione dell'automazione nei processi di Quality Assurance e sulla collaborazione del team QA con gli altri reparti per ottimizzare la qualità del prodotto finale.

Innanzitutto, la comunicazione tra il team QA e gli altri reparti è percepita come un elemento chiave per il successo del progetto, con un elevato grado di consenso tra i partecipanti. Questa tendenza si riscontra in modo uniforme, indipendentemente dalle metodologie di testing adottate, dalle dimensioni aziendali e dall'area geografica.

In generale, il feedback fornito dal team QA è ampiamente riconosciuto come cruciale per il successo del videogioco, sebbene vi siano alcune variazioni legate alla dimensione aziendale e al ruolo professionale. Le aziende più grandi attribuiscono un valore più alto al feedback del QA rispetto alle piccole aziende, dove i risultati positivi possono essere raggiunti anche senza un team QA strutturato.

Sebbene sia considerata essenziale, la modalità di comunicazione varia a seconda dell'azienda e della struttura del team. Ad esempio, strumenti come Slack, Discord o Jira sono considerati efficaci, così come l'adozione di Daily Stand-up e Retrospective, due riunioni tipiche dello sviluppo Agile che facilitano la comunicazione sia all'interno dei team che tra di essi. Il Daily Stand-up è una riunione giornaliera di circa 15 minuti in cui il team si allinea sugli sviluppi recenti, pianifica le attività imminenti e identifica eventuali ostacoli, rispondendo a domande come: "Cosa ho fatto ieri?", "Cosa farò oggi?" e "Ci sono impedimenti?". La Retrospective, invece, si svolge al termine di ogni Sprint e offre al team l'opportunità di riflettere sul lavoro appena concluso, valutando ciò che è andato bene, ciò che potrebbe essere migliorato e definendo azioni concrete per ottimizzare i processi negli Sprint successivi. Questi incontri promuovono la comunicazione, la trasparenza e il miglioramento continuo all'interno del team [32].

Per quanto riguarda l'automazione. La percezione dell'utilità degli strumenti automatizzati nel facilitare la comunicazione tra il QA e gli altri reparti è generalmente negativa, con significative differenze tra coloro che hanno esperienza con il testing automatizzato e coloro che si affidano esclusivamente al testing manuale. Le aziende che già adottano una combinazione di testing manuale e automatizzato tendono ad avere una visione più positiva, mentre le realtà più piccole o prive di

un reparto QA dedicato manifestano più scetticismo. Inoltre, i partecipanti con maggiore esperienza nell'industria sembrano riconoscere maggiormente il valore dell'automazione rispetto a coloro con un'esperienza intermedia. Le automazioni più comuni in questo ambito includono i report automatici inviati via email o le notifiche dei software in caso di errori critici. Inoltre, vi sono le automazioni che possono essere create tramite strumenti come Jira e GitHub, come già menzionato nel sottocapitolo precedente.

Le risposte raccolte nell'ultima sezione del questionario, che permetteva ai partecipanti di aggiungere ulteriori considerazioni, confermano che l'aspetto umano riveste ancora un ruolo fondamentale in questo lavoro. Parlare con uno sviluppatore (di qualsiasi reparto) e comprendere il suo linguaggio è un aspetto da non sottovalutare. Inoltre, un altro punto importante riguarda l'importanza delle competenze di programmazione per un QA tester, su cui le opinioni sono per lo più favorevoli. È sempre meglio ricevere un feedback negativo da un collega piuttosto che da una macchina in modo freddo e metodico. Inoltre, sebbene alcuni bug possano risultare divertenti e altri frustranti, rappresentano comunque un'occasione per rafforzare il legame all'interno del team.

Un altro aspetto della comunicazione dipende dal momento in cui il team di QA entra nel processo di produzione e dal team in cui viene inserito. L'integrazione del team QA nelle fasi iniziali dello sviluppo si rivela un aspetto ancora problematico per molte aziende, con un consenso limitato sulla sua effettiva importanza. I risultati suggeriscono che l'automazione non viene ancora percepita come uno strumento determinante per un coinvolgimento anticipato del QA, e che tale pratica risulti più diffusa nelle aziende più strutturate e con una maggiore esperienza nell'automatizzazione del testing.

Invece, secondo alcuni partecipanti, è fondamentale mantenere un'interazione continua tra il team di QA e gli altri membri del team per garantire una collaborazione efficace e prevenire la perdita di informazioni. Sebbene l'Embedded QA [17] non venga citato esplicitamente, è a questo concetto che i partecipanti si riferiscono, come già discusso nel capitolo 2. Avere almeno una persona dedicata al QA per ogni team di sviluppo permetterebbe di individuare i bug in modo più rapido ed efficace, fornire feedback più precisi e utili agli sviluppatori, e contribuire attivamente alla progettazione e allo sviluppo del gioco. Questi aspetti favoriscono anche un maggiore senso di appartenenza al progetto per i membri del team di QA, evitando che vengano visti come un elemento marginale, come spesso accade [23]. Dalle risposte alle domande aperte è emerso che molti professionisti del

settore si sentono poco valorizzati economicamente e non adeguatamente rispettati all'interno del team, venendo considerati solo come un supporto tecnico. Questa percezione influisce negativamente sul loro coinvolgimento emotivo e sulla volontà di rimanere nel settore. Il coinvolgimento emotivo è un fattore chiave per il successo di un prodotto creativo.

L'Embedded QA risulta particolarmente vantaggioso nel contesto del testing automatico poiché offre ai tester l'opportunità di sviluppare strumenti di testing più efficaci e mirati, grazie alla loro conoscenza diretta del codice sorgente e degli strumenti di sviluppo. Questo approccio consente di creare test automatici più precisi, capaci di individuare bug specifici e di analizzare scenari di gioco complessi. Inoltre, la stretta collaborazione tra i team di QA e sviluppo favorisce una perfetta integrazione del testing automatico nel flusso di lavoro, rendendolo meno invasivo e più efficiente. Questo approccio potrebbe anche contribuire alla creazione di Quality Plan efficaci, che però, al momento, molti partecipanti non sviluppano a causa di un coinvolgimento insufficiente nelle fasi iniziali, di una scarsa comunicazione e organizzazione, e anche perché non fanno parte dei processi considerati standard.

In conclusione, i risultati indicano che l'integrazione dell'automazione nei processi di Quality Assurance e la collaborazione con altri reparti sono aspetti ancora in evoluzione nel settore dello sviluppo videoludico. Mentre la comunicazione tra QA e altri team è universalmente considerata essenziale, l'adozione dell'automazione e il coinvolgimento del QA nelle fasi iniziali del processo di sviluppo presentano ancora margini di miglioramento. Inoltre, il valore attribuito alle competenze di programmazione evidenzia la necessità di una maggiore specializzazione per i QA tester, soprattutto in ambienti che fanno largo uso di strumenti automatizzati. Infine, l'adozione dell'Embedded QA si dimostra una strategia altamente vantaggiosa, in quanto permette di integrare il team di QA fin dalle prime fasi di sviluppo. Questo approccio consente ai tester di collaborare direttamente con gli sviluppatori, favorendo una comprensione più profonda del codice e delle dinamiche di gioco. Inoltre, il QA può individuare i bug in modo più rapido ed efficace, contribuire attivamente alla progettazione e garantire una qualità del prodotto finale superiore. L'integrazione del QA nel team di sviluppo non solo migliora la tempestività dei feedback e la precisione dei test, ma promuove anche un senso di appartenenza che stimola il coinvolgimento e il miglioramento continuo del prodotto. Quindi, con la conoscenza diretta dei processi e del codice, il team di QA può sviluppare automazioni più mirate ed efficaci, ottimizzando ulteriormente il flusso di lavoro e riducendo il carico di lavoro manuale.

Sfide e AI

Nei sottocapitoli precedenti sono state menzionate diverse sfide affrontate dal team di QA e dall'automazione. Qui, invece, viene presentata una panoramica più completa, basata sull'analisi di tutte le risposte raccolte nel questionario.

L'automazione nei processi di Quality Assurance rappresenta una risorsa sempre più utilizzata nello sviluppo dei videogiochi, ma la sua implementazione incontra numerose sfide che ne limitano l'efficacia. La complessità e la varietà proprie dei videogiochi rendono il testing automatizzato meno efficace rispetto a quello manuale in molte situazioni. I videogiochi non sono semplici applicazioni software: devono garantire libertà, varietà e interazione costante con il giocatore, elementi difficili da replicare in uno scenario puramente automatizzato.

Uno dei principali ostacoli è rappresentato dalla frammentazione e dalla mancanza di standardizzazione nel settore. Ogni gioco ha un proprio motore, una propria architettura e strumenti di sviluppo differenti, rendendo complessa la creazione di soluzioni di test automatizzati universali. Spesso, i team QA si trovano a dover lavorare con strumenti limitati o poco avanzati, mentre l'automazione richiede competenze tecniche specifiche e un significativo investimento in termini di tempo e risorse. Se l'automazione non è prevista sin dalle prime fasi di sviluppo, integrarla in un secondo momento diventa una sfida notevole e spesso non porta ai risultati sperati.

Un altro problema rilevante riguarda la scalabilità e i tempi di esecuzione dei test. Con giochi sempre più grandi e complessi, riuscire a coprire ogni aspetto attraverso l'automazione diventa sempre più difficile. I test automatizzati possono impiegare molto tempo per restituire risultati e, in alcuni casi, producono dati incoerenti, obbligando comunque a una revisione manuale. Inoltre, il cosiddetto "flakiness" dei test automatizzati, ovvero test che producono risultati inconsistenti senza modifiche al codice o all'ambiente di esecuzione, porta a situazioni in cui gli esiti possono variare tra diverse esecuzioni, rendendo difficile individuare con certezza un problema.

L'aspetto visivo e sonoro di un videogioco rappresenta un ulteriore limite per l'automazione. Valutare la qualità della grafica, l'impatto sonoro o il bilanciamento di un gioco richiede ancora l'occhio umano, poiché questi aspetti sono spesso soggettivi e difficilmente quantificabili da un algoritmo. Similmente, l'esperienza utente e la percezione della fluidità del gameplay rimangono difficili da testare senza il

coinvolgimento diretto di persone reali. Anche la simulazione di sessioni di gioco realistiche rappresenta una sfida, poiché riprodurre il comportamento di un giocatore umano in modo credibile richiede una complessità computazionale molto elevata.

La comunicazione tra il team QA e gli altri reparti di sviluppo emerge come un altro punto critico. Spesso il lavoro del QA non viene adeguatamente compreso o valorizzato, portando a difficoltà nella prioritizzazione dei bug e nella gestione delle segnalazioni. Le modifiche frequenti e non documentate, tipiche di molti progetti di sviluppo, rendono ancora più complicata la gestione del testing automatizzato, che richiede aggiornamenti costanti per restare efficace. In alcuni ambienti di lavoro, il QA è percepito come un ruolo subordinato rispetto ad altri team, creando tensioni interne e ostacolando la collaborazione.

Le pressioni legate alle tempistiche di sviluppo rappresentano un ulteriore ostacolo per l'adozione dell'automazione. I team QA devono spesso bilanciare la necessità di test approfonditi con il rispetto di scadenze strette, soprattutto nel caso di progetti ad alto budget. La necessità di rilasciare aggiornamenti frequenti e la crescente complessità dei giochi moderni rendono difficile implementare strategie di automazione che possano seguire il ritmo rapido dello sviluppo.

Un ultimo argomento emerso dal questionario è l'intelligenza artificiale. La quale nel testing dei videogiochi viene accolta con maggiore scetticismo rispetto all'automazione tradizionale. La percezione complessiva è che l'IA possa portare vantaggi incrementali piuttosto che rivoluzionare il settore. Alcuni rispondenti ritengono che l'IA possa migliorare i processi esistenti, ma non sostituire completamente il testing manuale.

Un tema centrale emerso nelle risposte è il timore che l'automazione e l'IA possano ridurre il numero di professionisti QA necessari. Molti partecipanti ritengono che il ruolo del QA si sposterà sempre più verso la gestione e l'analisi dei dati generati dagli strumenti automatizzati, con un impatto potenzialmente negativo sull'occupazione. Tuttavia, viene sottolineato che l'IA presenta ancora limitazioni significative, in particolare nell'identificazione di bug soggettivi e nella valutazione della qualità complessiva di un videogioco. Sebbene alcuni test possano essere automatizzati, come il controllo delle risorse grafiche e la verifica delle collisioni, l'intelligenza artificiale non è ancora in grado di sostituire il giudizio umano in ambiti soggettivi, come il game feel e l'esperienza utente.

Nonostante il generale scetticismo emerso dal questionario riguardo all'adozione dell'intelligenza artificiale nel QA, diversi sviluppi recenti dimostrano che l'IA sta già trovando applicazioni concrete nel settore. Come discusso in precedenza, agenti autonomi basati su intelligenza artificiale vengono impiegati per esplorare ambienti di gioco, testare funzionalità specifiche e persino valutare il bilanciamento del gameplay attraverso tecniche di Reinforcement Learning e IA Planning. Queste tecniche possono essere implementate sia dallo stesso studio di sviluppo o appoggiarsi a servizi esterni [39]

Secondo quanto riportato da un intervistato, l'intelligenza artificiale dovrebbe supportare l'esecuzione di compiti particolarmente tediosi, come ad esempio il miglioramento della scrittura delle task. Inoltre, strumenti in fase di sviluppo, come quelli esplorati da EA [25], mirano a identificare il commit responsabile di un bug, riducendo significativamente i tempi di ricerca del problema. Un ulteriore ambito di applicazione potrebbe essere l'uso dell'IA per uniformare il bug reporting, integrando informazioni provenienti da video, immagini e descrizioni testuali, facilitando così la comunicazione tra il team di QA e il team di sviluppo.

Nonostante le sfide, le risposte mostrano un atteggiamento pragmatico nei confronti del futuro dell'automazione. Più che una rivoluzione, l'automazione e l'IA sono viste come strumenti di miglioramento incrementale, che possono ottimizzare i processi di QA senza sostituire completamente il lavoro umano. L'integrazione dell'IA potrebbe rivelarsi particolarmente utile in test di tipo grafico e di gameplay, laddove sia necessario verificare aspetti visivi o la correttezza di elementi dinamici nel gioco. Tuttavia, il settore sembra ancora lontano da una trasformazione radicale, e l'esperienza umana continua a essere considerata un elemento fondamentale per garantire la qualità dei videogiochi.

Processo automatizzato di QA nei videogiochi

Considerando sia le sfide e i problemi riscontrati, sia le necessità e i vantaggi derivanti dall'automazione nel testing, si propone un processo di testing per il videogioco. L'obiettivo è l'ottimizzazione del processo di Quality Assurance nello sviluppo di videogiochi, attraverso un bilanciamento efficace tra automazione e testing manuale. In particolare, si intende identificare quali processi possano essere automatizzati, ad esempio quelli più ripetitivi. Allo stesso tempo, si mira a definire quali attività debbano rimanere manuali, come l'esecuzione di test case specifici o la valutazione del game feel, aspetti che richiedono l'intervento umano per garantire un'esperienza di gioco ottimale. In questo contesto, si mira a valorizzare il ruolo del team di QA senza appesantire il workflow di sviluppo. L'approccio si basa sull'integrazione del concetto di Embedded QA, sfruttando sistemi di CI/CD e strumenti avanzati per il testing automatizzato, al fine di migliorare l'efficienza e la qualità del prodotto finale.

I principali fattori del processo proposto includono il budget, la dimensione del team e quella del gioco. Di conseguenza, si torna alla classificazione degli studi menzionata nel Capitolo 2, distinguendo tra studi indipendenti e quelli categorizzati come AA o AAA. In particolare, gli studi indie possono essere autofinanziati (evenienza sempre più rara), finanziati tramite crowdfunding, supportati da un publisher o bandi pubblici. Tutti gli studi, in teoria, possono avere un contratto con un publisher, il quale può offrire vari servizi, tra cui il supporto al QA, sia interno che esterno. Ad esempio, qui è possibile trovare un contratto tipico con il publisher Raw Fury [12].

Pertanto, è necessario distinguere tra due scenari molto probabili:

- Studi indie con un contratto con un publisher che fornisce QA esterno o meno.
- Studi AA o AAA, che hanno accesso sia a un QA interno sia, potenzialmente, a un QA esterno.

Alla luce di queste premesse, si consideri il caso di uno studio di sviluppo specializzato in giochi single player (escludendo volutamente i titoli multiplayer, in quanto richiederebbero un'analisi più approfondita sulla tipologia di test automatizzabili). Lo studio ha già stipulato un contratto con un publisher, ma non dispone di un team QA esterno dedicato. Ciò implica che il progetto abbia già superato con successo la fase di pre-produzione e la validazione delle meccaniche di gioco, passaggi

fondamentali per ottenere l'approvazione del publisher.

Nonostante ciò, viene comunque prevista una brevissima fase di preproduzione, durante la quale vengono sviluppati i concept e si svolgono riunioni di design. A questi incontri partecipa anche il responsabile del QA, che a seconda della dimensione dell'azienda può essere il QA Manager, il producer o il project manager. L'obiettivo è quello di prevenire eventuali problemi di design e definire le strategie di test più efficaci. In questa fase si decide quali aspetti del testing verranno automatizzati e quali, invece, saranno eseguiti manualmente. Sebbene si cerchi di adottare un approccio il più generale possibile per garantire un workflow efficiente, sarà comunque necessario prendere decisioni specifiche in merito ai test da implementare, come avviene in ogni progetto. I test che verranno sicuramente automatizzati includono: Unit, Functional, Integration, Regression, Performance ed End-to-end.

In parallelo, vengono avviati i primi test unitari su elementi già abbastanza stabili e poco soggetti a cambiamenti. Si utilizzano framework di testing specifici per l'engine scelto (ad esempio, Unity Test Framework o Unreal Automation Tool), con il supporto di strumenti basati su AI per la scrittura del codice. Un sistema di CI/CD, come un qualsiasi servizio di automazione delle build (es. GitHub Actions, Jenkins), esegue automaticamente questi test ogni volta che viene effettuata una modifica nel branch dedicato alla creazione di una build di gioco. I risultati possono essere archiviati su una piattaforma cloud accessibile al team. Inoltre, è fondamentale automatizzare lo strumento di gestione delle task o del monitoraggio dei bug (es. Jira), integrandolo con la propria piattaforma di controllo versione scelta (es. GitHub), scegliendo la soluzione più adatta in base alle esigenze del team.

Durante la fase iniziale di produzione viene sviluppato il core gameplay insieme alle funzionalità principali. In questo contesto, si applicano i principi dell'embedded QA, integrando uno o più membri del QA all'interno del team di sviluppo, che include programmatori, designer e altre figure chiave. Inoltre, vengono organizzate riunioni settimanali per identificare i problemi, stabilire le priorità e monitorare i progressi. I test funzionali, in questa fase, devono essere prevalentemente automatizzati e costantemente monitorati per rilevare eventuali variazioni. È possibile avvalersi anche di strumenti specifici, come AltTester o GameDriver, per ottimizzare il processo di testing.

Nelle fasi avanzate dello sviluppo, l'attenzione si sposta sull'ottimizzazione delle

prestazioni e sul bilanciamento del gameplay. In questo contesto, vengono implementati test end-to-end automatizzati che simulano il comportamento dell'utente durante l'intera esperienza di gioco, permettendo di monitorare in tempo reale le performance del sistema. Questi test sono progettati per rilevare automaticamente eventuali anomalie, come cali di FPS o memory leak, eseguendo il gioco in scenari reali e identificando eventuali colli di bottiglia nelle risorse di sistema. Inoltre, il monitoraggio delle performance è integrato con strumenti di visualizzazione avanzata, come grafici interattivi, che permettono di osservare e analizzare facilmente l'andamento dei parametri critici (come l'uso della CPU, della memoria e della GPU) durante il test, facilitando l'individuazione di aree da ottimizzare. Questi grafici possono essere implementati sia all'interno del motore di gioco che tramite strumenti esterni. Parallelamente, è fondamentale automatizzare i test di regressione per garantire che le nuove implementazioni non compromettano funzionalità precedenti. Questi test, eseguiti continuamente durante il ciclo di sviluppo, sono essenziali per assicurare la stabilità del prodotto, e l'uso di grafici e visualizzazioni dei risultati consente di avere una panoramica chiara e immediata dello stato del sistema, identificando rapidamente eventuali regressioni o problemi emersi. In questo modo, l'osservazione visiva dei dati diventa cruciale per una gestione efficace delle performance e della qualità complessiva del gioco.

In questa fase, possono essere adottate due tecniche di testing avanzate:

- L'AI-driven testing: si basa su bot automatici che esplorano il gioco, eseguono azioni casuali o predefinite e cercano bug in modo autonomo. Questi bot simulano il comportamento dei giocatori e testano scenari che potrebbero sfuggire ai test manuali o automatizzati tradizionali. Il bot può essere basato su regole (script predefiniti) o AI avanzata (reinforcement learning, machine learning).
- Test basati su Capture & Replay (C&R): prevedono la registrazione delle interazioni dei giocatori, che vengono poi tradotte in script di test ripetibili. Gli approcci C&R possono variare in base al metodo di generazione degli script, sfruttando API e framework di automazione esistenti per rieseguire le interazioni all'interno del gioco oppure utilizzando altre tecniche, come coordinate o computer vision, per simulare l'interazione in tempo reale.

Tra la produzione della build Alpha e Beta, in cui il gioco è quasi completo, è essenziale eseguire playtest manuali, sia interni che esterni. Questi test sono fondamentali per valutare aspetti che non possono essere verificati automaticamente, come il *game feel* del gameplay. Inoltre, è consigliabile continuare a sfruttare strumenti di C&R e monitorare le prestazioni per garantire un'esperienza di gioco

ottimale.

Infine, dopo il lancio del gioco, il QA continua a svolgere un ruolo chiave nel monitoraggio della stabilità del prodotto. Gli aggiornamenti e le patch vengono testati in ambienti separati per evitare che nuovi problemi emergano dopo la distribuzione. L'automazione dei regression test assicura che ogni modifica non comprometta il funzionamento del gioco. Inoltre, grazie all'analisi dei dati raccolti dai giocatori, è possibile individuare problematiche che potrebbero non essere emerse in precedenza.

L'implementazione di questa pipeline di lavoro può offrire i seguenti benefici produttivi:

- **Maggiore efficienza:** Riduzione dei tempi di testing grazie all'esecuzione automatica e continua dei test.
- **Migliore rilevamento di bug:** L'AI-driven testing consente di individuare scenari di errore complessi e difficili da replicare manualmente. Inoltre, l'automazione dei test di regressione e funzionali facilita l'identificazione tempestiva dei bug, migliorando l'efficienza del processo di sviluppo.
- **Collaborazione più efficace:** L'integrazione dell'automazione con strumenti di gestione delle task semplifica la comunicazione tra il team QA e gli altri reparti, grazie anche all'approccio dell'Embedded QA.
- **Ridurre i costi a lungo termine e Scalabilità:** Eseguire test automatizzati in modo costante consente di riutilizzare parte degli script anche in progetti futuri, adattandoli facilmente a nuove esigenze.
- **Riduzione del carico di lavoro ripetitivo:** L'automazione dei test di regressione, test funzionali e test di performance permette al QA di concentrarsi su aspetti più complessi, come il game feel e l'esperienza utente, che non possono essere testati automaticamente. Cercando così di abbattere stress e crunch.
- **Migliore controllo sulla qualità del prodotto:** L'automazione continua tramite la CI/CD pipeline consente di identificare bug in modo tempestivo. Inoltre introdurre il QA fin dalle fasi iniziali permette di pensare alla qualità del prodotto fin dall'inizio della produzione.
- **Maggiore visibilità e riconoscimento del ruolo del QA:** integrare il QA fin dall'inizio della produzione, integrarlo all'interno del team di sviluppo

permette di renderlo parte attiva e strategica del processo produttivo e non più come una fase "finale" del ciclo di sviluppo.

In sintesi, è fondamentale automatizzare i seguenti test:

Unit Test	Eseguiti su elementi stabili e poco soggetti a cambiamenti, tramite framework di testing integrati nei motori di gioco.
Functional Test	Automatizzati per verificare la corretta implementazione delle funzionalità principali del gameplay.
Regression Test	Automatizzati per assicurare che nuove implementazioni non compromettano funzionalità precedenti.
AI-driven Testing	Tramite bot automatici che eseguono azioni casuali o predefinite per rilevare bug nascosti. Questi bot possono basarsi su regole statiche o sfruttare algoritmi di machine learning.
Performance Test	Eseguiti per monitorare cali di FPS o memory leak.
Integration Test	Automatizzati per verificare la corretta comunicazione tra i diversi moduli del gioco.
End-to-end Test	Simulano l'intero flusso di gioco, dall'avvio alla chiusura, per identificare problemi di esperienza utente o di stabilità generale.

Tabella 5.1: Test da automatizzare nel processo di sviluppo

Infine, viene riportato uno schema riassuntivo del processo:

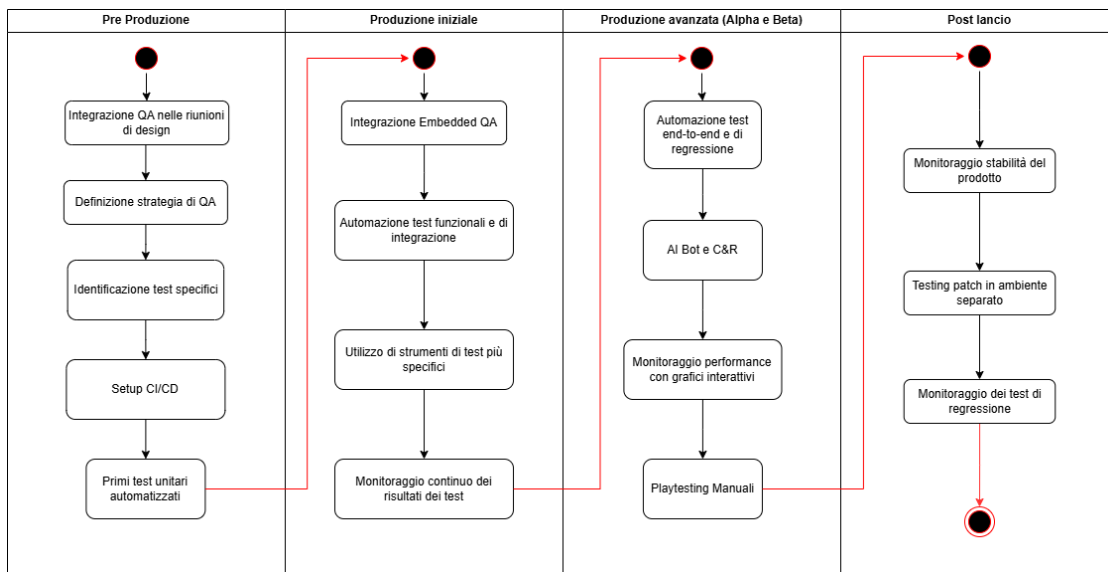


Figura 5.1: Processo automatizzato di QA

Conclusioni

L'analisi condotta in questa ricerca ha evidenziato come l'automazione stia trasformando i processi di Quality Assurance nell'industria videoludica, senza tuttavia sostituire completamente il testing manuale. I dati raccolti confermano che le aziende preferiscono un approccio ibrido, in cui i test automatizzati sono impiegati per attività ripetitive e di regressione, mentre il testing manuale rimane essenziale per garantire la qualità dell'esperienza utente e per individuare problematiche non facilmente rilevabili dagli strumenti automatici. I risultati ottenuti evidenziano che, sebbene l'automazione offra vantaggi significativi in termini di efficienza e ripetibilità dei test, la scelta tra testing manuale e automatizzato dipende da diversi fattori, tra cui la dimensione dello studio, le risorse disponibili e la tipologia del progetto sviluppato.

L'integrazione dell'automazione nei processi di QA si è rivelata un elemento chiave per ottimizzare i flussi di lavoro e migliorare la qualità complessiva del prodotto finale. Tuttavia, le sfide legate all'implementazione di strumenti automatizzati, alla formazione del personale e alla comunicazione tra reparti restano ancora presenti. Un elemento chiave emerso dallo studio è il ruolo del QA all'interno del ciclo di sviluppo del gioco. Nonostante l'importanza delle attività di testing, il settore soffre ancora di una percezione del QA come posizione entry-level, con conseguenti difficoltà nel trattenere talenti esperti e una frequente rotazione del personale. È evidente la necessità di una maggiore valorizzazione della figura del tester, spesso relegata a un ruolo marginale rispetto ad altri dipartimenti di sviluppo. Questo fenomeno impatta direttamente la qualità del prodotto finale e suggerisce la necessità di una maggiore professionalizzazione del settore.

Un altro aspetto rilevante riguarda l'introduzione dell'intelligenza artificiale nel testing, che viene accolta con un atteggiamento misto. Sebbene alcuni professionisti riconoscano il potenziale dell'IA nel migliorare l'efficienza del QA, esiste ancora un forte scetticismo. In particolare, l'IA viene vista come uno strumento di supporto piuttosto che una soluzione autonoma.

La ricerca ha inoltre sottolineato come ci sia una tendenza a volere il QA sempre più integrato nei processi produttivi attraverso strategie di Embedded QA e CI/CD, permettendo un monitoraggio continuo della qualità del gioco durante

tutto il ciclo di sviluppo.

Alla luce di questi risultati, questa ricerca suggerisce alcune direzioni per ottimizzare i processi di QA nel settore videoludico. Da un lato, è necessario promuovere un'integrazione più strutturata dell'automazione, sfruttando i suoi vantaggi in termini di efficienza e ripetibilità, senza trascurare il valore aggiunto del testing umano. Dall'altro, è fondamentale migliorare la percezione e il riconoscimento del ruolo del QA, investendo in formazione e creando percorsi di carriera più solidi per i professionisti del settore.

Infine, per approfondire ulteriormente la tematica dell'automazione nel QA videoludico, sono possibili diversi sviluppi futuri:

- **Espandere il campione di ricerca:** Per ottenere dati più dettagliati e rappresentativi, sarebbe utile estendere il questionario a un numero maggiore di professionisti del settore o affinare la selezione degli intervistati per coinvolgere figure con ruoli specifici nel QA, attraverso interviste più mirate e approfondite.
- **Osservazione diretta nei processi produttivi:** Un passo successivo potrebbe essere quello di svolgere un periodo di osservazione all'interno di un'azienda videoludica, per analizzare direttamente come vengono applicati i processi di QA e in che modo l'automazione viene effettivamente utilizzata nel ciclo di sviluppo.
- **Sviluppo di sistemi AI per supportare il QA:** L'intelligenza artificiale potrebbe essere esplorata come strumento per automatizzare la creazione di task di QA in base alle caratteristiche del progetto. Un sistema AI potrebbe analizzare la documentazione di sviluppo e l'engine fin dall'inizio del progetto, per suggerire e creare test specifici da eseguire, migliorando ulteriormente l'efficacia dell'automazione.

Queste prospettive aprono la strada a nuove possibilità per migliorare l'efficienza e la precisione del QA nei videogiochi, contribuendo all'evoluzione del settore verso metodologie di testing sempre più integrate e sofisticate. L'obiettivo finale è quello di raggiungere un equilibrio efficace tra innovazione tecnologica e competenze umane, garantendo un alto livello di qualità nei videogiochi senza sacrificare la creatività e il controllo umano sui processi di testing.

Bibliografia e Sitografia

- [1] Giuseppe Carrabba Andrea Alosi. Sviluppare un videogioco: cosa fa e quanto è importante il QA Manager? <https://www.everyeye.it/articoli/speciale-sviluppare-videogioco-cosa-importante-qa-manager-63279.html>, 2024. Ultimo accesso: 24 febbraio 2025.
- [2] Brittany Alva. What makes a AAA game a AAA game? <https://store.epicgames.com/it/news/what-makes-a-aaa-game-a-aaa-game>, 2023. Consultato il 16 dicembre 2024.
- [3] Atlassian. How to integrate Jira Software and GitHub | The Developer's Edge | Atlassian. https://youtu.be/N-RZjp4og28?si=VO_9WNVR80DQ0xa2, 2023. Ultimo accesso: 3 febbraio 2025.
- [4] Team Cherry. Hollow Knight - Release Trailer. <https://youtu.be/UA02urG23S4?si=KHUnvq9ZJkmVg0x1>, 2017. Ultimo accesso: 22 febbraio 2025.
- [5] Jaehyung Cho. Bughunting on a Budget: Exploring Quality Assurance Practices and Tools for Indie Game Developers. Master's thesis, University of Alberta, 2022.
- [6] Unreal Engine Documentation. Gauntlet Automation Framework. <https://dev.epicgames.com/documentation/en-us/unreal-engine/gauntlet-automation-framework-in-unreal-engine>. Ultimo accesso: 26 febbraio 2025.
- [7] Francesco Fossetti e Marco Mottura. Il disastro di Unity & Death Stranding su iPhone. <https://open.spotify.com/episode/5stXnEys0AW0qTQgyzuKDV?si=b468609123ef4784>, Settembre 2023. Podcast, Gong! - Il podcast di RoundTwo, 209.
- [8] Francesco Fossetti e Marco Mottura. Il fallimento di Concord & Itsuno lascia Capcom - Episodio 325. <https://open.spotify.com/episode/6ExX8TddBwa20N3nrCNYXV?si=89f35d73556544c1>, Settembre 2024. Podcast, Gong! - Il podcast di RoundTwo, 325.
- [9] Charles P. Schultz e Robert Denton Bryant. *Game Tesin: all in one, Third Edition*. Mercury Learning and Information, 2016.

- [10] Aleksandar Tesic QA Engineer. QA ENGINEER IN THE VIDEO GAMES INDUSTRY. <https://nordeus.com/qa-engineer-in-the-video-games-industry/>, 2022. Ultimo accesso: 24 febbraio 2025.
- [11] Tracy Fullerton. *Game Design Workshop*. AK Peters/CRC Press, 2018.
- [12] Raw Fury. Why we are publishing Raw Furys publishing agreement. <https://rawfury.com/why-we-are-publishing-raw-furys-publishing-agreement/>. Ultimo accesso: 5 marzo 2025.
- [13] GamesIndustry.biz. Layoffs, n.d. Accessed: 2025-03-12.
- [14] Rob Fahey gamesindustry.biz. Embracing disaster. <https://www.gamesindustry.biz/embracing-disaster-opinion>. Ultimo accesso: 12 marzo 2025.
- [15] Romain Gauthier. 5 good reasons to invest in automated game testing. <https://gameconductor.net/blog/5-good-reasons-to-invest-in-automated-game-testing>, 2021. Ultimo accesso: 4 gennaio 2025.
- [16] Romain Gauthier. The challenges of test automation in video games. <https://gameconductor.net/blog/the-challenges-of-test-automation-in-video-games>, 2024. Ultimo accesso: 18 dicembre 2024.
- [17] GDC. BioWare Embedded QA: Your Support Class. <https://www.youtube.com/watch?v=RMMofotYwAQ>, 2019. Ultimo accesso: 16 dicembre 2024.
- [18] GDC. A QA Perspective on Live Game Production. <https://www.youtube.com/watch?v=cMxNTsD-DaQ>, 2021. Ultimo accesso: 28 dicembre 2024.
- [19] GDC. AI for Testing: The Development of Bots that Play Battlefield V. https://youtu.be/s1JOSbUR6KE?si=7hv0_OUKlvmqoeU, 2021. Ultimo accesso: 6 dicembre 2024.
- [20] GDC. Automated Testing of Gameplay Features in 'Sea of Thieves'. <https://youtu.be/X673t0i8pU8?si=qEUZ4HdfcYM-DThS>, 2021. Ultimo accesso: 5 gennaio 2025.
- [21] GDC. Don't You Dare Say Showstopper: Bootstrapping Your Own QA Team. <https://youtu.be/p53chylGfok?si=91r6UyrDWEaQJqHL>, 2021. Ultimo accesso: 5 gennaio 2025.
- [22] GDC. 'Final Fantasy VII' Remake: Automating Quality Assurance and the Tools for the Future. https://youtu.be/L2bJ4E_4zN8?si=-Pv5PrAWaMbQM5tc, 2024. Ultimo accesso: 28 dicembre 2024.
- [23] GDC. QA Career Path: Evolution from an Industry Entry Point. <https://www.youtube.com/watch?v=0hp-aEEupBg>, 2024. Ultimo accesso: 16

dicembre 2024.

- [24] Serenella Manzi. How to Measure Game Testing: a Survey of Coverage Metrics and an Implementation on the iv4XR Framework. Master's thesis, Politecnico di Torino, 2023/2024.
- [25] Leonardo Marini, Linus Gisslén, and Alessandro Sestini. Leveraging Large Language Models for Efficient Failure Analysis in Game Development, 2024.
- [26] Maxim Mozgovoy and Evgeny Pyshkin. A Comprehensive Approach to Quality Assurance in a Mobile Game Project. In *Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia*, CEE-SECR '18, New York, NY, USA, 2018. Association for Computing Machinery.
- [27] Newzoo. Newzoo's Global Games Market Report 2024. <https://newzoo.com/resources/trend-reports/newzoos-global-games-market-report-2024-free-version>, 2024. Ultimo accesso: 17 gennaio 2025.
- [28] Nintendo of America Maddy Makes Games. Celeste – Nintendo Switch Trailer. <https://youtu.be/iofYDsA2yqg?si=uYtov5KTbCxNSdjH>, 2017. Ultimo accesso: 22 febbraio 2025.
- [29] Cristiano Politowski, Yann-Gaël Guéhéneuc, and Fabio Petrillo. Towards automated video game testing: still a long way to go. In *Proceedings of the 6th International ICSE Workshop on Games and Software Engineering: Engineering Fun, Inspiration, and Motivation*, GAS '22, page 37–43, New York, NY, USA, 2022. Association for Computing Machinery.
- [30] Cristiano Politowski, Fabio Petrillo, and Yann-Gaël Guéhéneuc. A Survey of Video Game Testing. In *2021 IEEE/ACM International Conference on Automation of Software Test (AST)*, pages 90–99, 2021.
- [31] IGN Kojima Productions. Kojima Productions' Death Stranding Reveal Trailer - E3 2016. <https://youtu.be/i2nuHEGhwiv?si=KcVdJGq3k9mLKyrT>, 2016. Ultimo accesso: 22 febbraio 2025.
- [32] Dan Radigan. Che cos'è una riunione stand-up e suggerimenti per organizzarne una. <https://www.atlassian.com/it/agile/scrum/standups>. Ultimo accesso: 4 febbraio 2025.
- [33] Rido Ramadan and Bayu Hendradjaya. Development of game testing method for measuring game quality. In *2014 International Conference on Data and Software Engineering (ICODSE)*, pages 1–6, 2014.
- [34] IGN Rare. Banjo-Kazooie: Nuts & Bolts Xbox 360 Trailer - First. <https://youtu.be/QLWLd09GLWM?si=JiH3c0UKz5RreNp1>, 2011. Ultimo accesso: 22 febbraio 2025.
- [35] Xbox Rare. Sea of Thieves E3 Announce Trailer. https://youtu.be/1z48qvGsA_0?si=I6SHpfcIPibSH2ei, 2015. Ultimo accesso: 22 febbraio

- 2025.
- [36] Cyberpunk 2077 CD PROJEKT RED. Cyberpunk 2077 – E3 2018 trailer ufficiale. https://youtu.be/8X2kIfS6fb8?si=1Iwp98qcI__tAIpt, 2018. Ultimo accesso: 22 febbraio 2025.
 - [37] Mattia Riola. Test automation in video game development: Literature review and Sound testing implementation. Master’s thesis, Politecnico di Torino, 2022/2023.
 - [38] Alexander Senchenko, Naomi Patterson, Hamman Samuel, and Dan Ispir. Supernova: Automating test selection and defect prevention in aaa video games using risk based testing and machine learning. In *2022 IEEE Conference on Software Testing, Verification and Validation (ICST)*, page 345–354. IEEE, April 2022.
 - [39] Julian Togelius. Improving QA Game Testing with Evolved AI. <https://www.gamedeveloper.com/programming/improving-qa-game-testing-with-evolved-ai>. Ultimo accesso: 3 marzo 2025.
 - [40] More Perfect Union. Video Game Testers Demand End to Exploitation, 14-Hour Days, And Poverty Wages. <https://youtu.be/T5WkRtJX1EM?si=VMoDLwTWtdSJFziP>, 2022. Ultimo accesso: 23 febbraio 2025.
 - [41] Unity. What is CI/CD? <https://unity.com/topics/what-is-ci-cd>. Ultimo accesso: 9 marzo 2025.
 - [42] Stanford University. Introducing the Center for Research on Foundation Models (CRFM). <https://hai.stanford.edu/news/introducing-center-research-foundation-models-crfm>. Ultimo accesso: 3 marzo 2025.
 - [43] Game Industry Conference Piotr Zalewski. Automating game testing in Unreal Engine - Piotr Zalewski || People Can Fly ||. https://youtu.be/erNW3r9uRH0?si=12CXxZVz_y_nG0e-, 2022. Ultimo accesso: 26 febbraio 2025.

Appendice A Questionario

Il questionario è stato somministrato in inglese e viene di seguito riportato in italiano. Esso è suddiviso in sei sezioni principali. La prima è dedicata alle domande demografiche. La seconda ha lo scopo di verificare se l'intervistato rappresenta un campione adeguato per l'analisi. La terza, la quarta e la quinta sezione sono relative rispettivamente alla RQ1, RQ2 e RQ3. L'ultima sezione è composta da una singola domanda aperta, che consente di fornire eventuali commenti.

Le tipologie di domande presenti sono:

- Aperte
- A scelta multipla (contrassegnate con il simbolo del cerchio)
- A selezione multipla (contrassegnate con il simbolo del quadrato)
- A risposta binaria (contrassegnate con il simbolo del cerchio)
- A scala Likert (vanno da 1 a 5, dove 1 è "Per niente" e 5 è "Completamente")

Infine le domande con l'asterisco sono obbligatorie.

Domande demografiche

1.1 Come hai scoperto questo sondaggio?*

- Discord
- LinkedIn
- Mail
- Altro:

1.2 Qual è il tuo ruolo nella compagnia di videogiochi per cui lavori?*

- QA
- Producer

Project Manager

Altro:

1.3 Specifica il tuo ruolo nel team di QA in "Altro":* (Se non fai parte di un team di QA, seleziona NA)

NA

Altro:

1.4 Da quanti anni lavori nell'industria dei videogiochi?*

0 - 1

2 - 5

6 - 10

10+

1.5 Quanti dipendenti ha la compagnia di videogiochi per cui lavori?*

1 - 20

20 - 100

100+

1.6 La compagnia per cui lavori ha sede in Italia?*

Sì

No

1.7 C'è un team di QA nella compagnia per cui lavori attualmente?* (Rispondi "Sì" se fai parte del team di QA)

Sì

No

1.8 Se hai risposto "No" alla domanda precedente, potresti spiegare perché?

1.9 Continuate comunque a svolgere attività di testing nella vostra compagnia?*

- Sì (Va in automatico alla sezione A.3)
- No (Va in automatico alla sezione A.2)

Domande per valutare l'utilizzo di QA esterno o l'esperienza passata con attività di QA

2.1 Il tuo team si affida a compagnie esterne per il testing di Quality Assurance?*

2.2 Hai ricoperto un ruolo in un team di QA in passato?*

Confronto tra QA automatizzata e manuale

Queste domande esaminano le differenze tra i processi di QA automatizzati e manuali nell'individuare bug e problemi di gameplay.

3.1 Qual è il motore più comunemente utilizzato nella tua azienda?*

- Unreal Engine
- Unity
- Godot
- Proprietario

3.2 Per quale piattaforma sviluppa principalmente la tua azienda?*

- PC
- Console
- Mobile
- Altro:

3.3 Quali tipi di test vengono eseguiti più frequentemente? *

- Functional Testing
- UX Testing
- Regression Testing
- Compatibility Testing
- Performance Testing
- Localization Testing
- Compliance Testing
- Altro:

3.4 Come vengono condotti i test nella tua azienda? *

- Automatico
- Manuale
- Entrambi

3.5 La tua azienda utilizza strumenti di testing sviluppati internamente o di terze parti?* (Se hai risposto Manuale o per altri motivi, seleziona "NA")

- NA
- Utilizziamo strumenti di testing sviluppati internamente
- Utilizziamo strumenti di testing disponibili sul mercato
- Entrambi

3.6 Perché non eseguite test automatizzati?* (Se utilizzi test automatizzati, seleziona "NA")

- NA
- Altro:

3.7 Quali tipi di test sono più frequentemente automatizzati nei tuoi progetti di sviluppo videoludico?* (Se hai risposto Manuale o per altri motivi, seleziona "NA")

- NA
- Altro:

3.8 Come si integrano i test automatizzati con altre metodologie di QA (es. play-testing, testing manuale)?* (Se hai risposto Manuale o per altri motivi, seleziona "NA")

- NA
- Altro:

3.9 Come si confrontano i test automatizzati con i test manuali in termini di efficienza ed efficacia?* (Se hai risposto Manuale o per altri motivi, seleziona "NA")

- NA
- Altro:

3.10 Quali vantaggi hai riscontrato nell'utilizzo dell'automazione nei processi di QA rispetto a un approccio manuale?* (Se hai risposto Manuale o per altri motivi, seleziona "NA")

- NA
- Altro:

3.11 Quale tra i seguenti software per l'automazione dei test utilizzi?* (Se hai risposto Manual o per altri motivi, seleziona "NA")

- NA
- Selenium
- Unity Test Framework
- Unreal Engine Automation System
- Altro:

3.12 Quali strumenti software utilizzati per il bug tracking?*

- Jira
- Asana
- Mantis
- Trello
- Bugzilla
- GitHub
- Issues
- Altro:

3.13 Come viene determinata la priorità dei bug e quali criteri vengono utilizzati?*

Automazione e collaborazione nel QA

Queste domande indagano come il team di Quality Assurance integra l'automazione nei processi di sviluppo del gioco e collabora con altri reparti per ottimizzare la qualità del prodotto finale.

4.1 In che modo l'automazione influenza il lavoro quotidiano dei tester QA all'interno del team?*

4.2 Come raccoglie e comunica il team di QA i risultati dei test automatizzati agli altri reparti?*

4.3 Su una scala da 1 a 5, quanto ritieni importante la comunicazione tra il team di QA e gli altri team di sviluppo?*

4.4 Quanto sei d'accordo, su una scala da 1 a 5, con l'affermazione: "Il feedback del team di QA è essenziale per il successo di un videogioco"?*

4.5 Su una scala da 1 a 5, quanto è integrato il team di QA (incluso il lead o il project manager) nelle prime fasi del processo di sviluppo del gioco?*

4.6 Quando inizia a lavorare il team di QA (incluso il lead o il project manager) all'interno del ciclo di produzione della tua azienda?*

4.7 Secondo te, quando dovrebbe iniziare a lavorare il team di QA (incluso il lead o il project manager) all'interno del ciclo di produzione per ottenere i maggiori benefici?*

4.8 Adottate un Piano di Qualità nel vostro ciclo di produzione?*

- o Si
- o No

4.9 Specifica come avviene o perché non viene fatto:

4.10 Come viene garantita una comunicazione efficace tra il team di QA e gli altri team di sviluppo?*

4.11 Su una scala da 1 a 5, quanto potrebbero facilitare la comunicazione tra il team di QA e gli altri reparti gli strumenti di automazione?*

4.12 Su una scala da 1 a 5, quanto è utile che un tester QA sappia anche programmare (ad esempio, per correggere direttamente i bug)?*

Sfide e AI nel QA

Queste domande esplorano le principali sfide che i team di QA affrontano nei processi di automazione e come percepiscono l'evoluzione del loro ruolo con l'introduzione di tecnologie come l'intelligenza artificiale nell'industria dei videogiochi.

5.1 Quali sono le difficoltà nell'applicare il testing automatizzato ai videogiochi, e in quali casi il testing manuale rimane indispensabile?*

5.2 Quali sono le principali sfide nel campo della Quality Assurance?*

5.3 Quali aspetti dell'automazione risultano più difficili da implementare per il team di QA?*

5.4 Che cambiamenti vorresti vedere nei test per i videogiochi?*

5.5 Su una scala da 1 a 5, credi che l'automazione possa ridurre il numero di bug non rilevati durante lo sviluppo?*

5.6 Su una scala da 1 a 5, come gli strumenti di automazione aiutano a ridurre i tempi di sviluppo e rilascio dei videogiochi?*

5.7 Su una scala da 1 a 5, quanto pensi che tecnologie come l'intelligenza artificiale cambieranno il modo in cui i videogiochi vengono testati nei prossimi anni?*

5.8 (Relativo alla domanda precedente) Specifica come:

5.9 Quali sono le principali sfide nell'adottare tecnologie AI per l'automazione dei test?*

5.10 Su una scala da 1 a 5, quanto pensi che le nuove tecnologie, come l'AI, possano migliorare l'efficacia del testing automatizzato?*

5.11 Su una scala da 1 a 5, quanto pensi che l'uso dell'AI possa migliorare la collaborazione tra il team di QA e gli sviluppatori?*

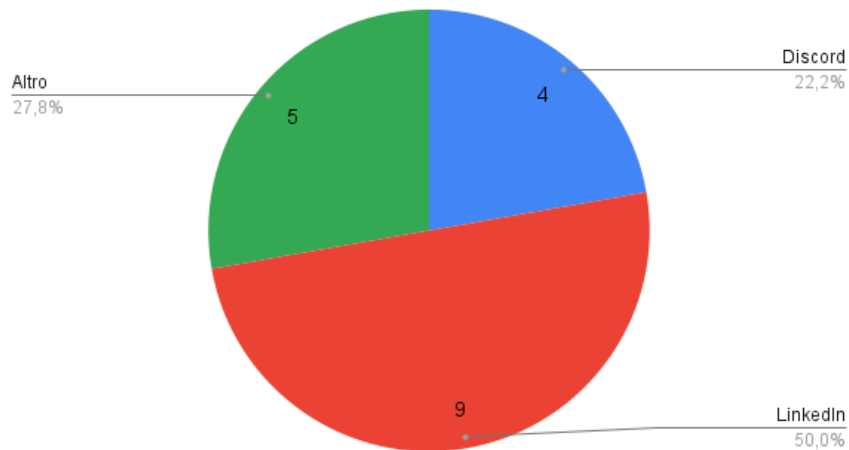
Suggerimenti

6.1 Se hai commenti o suggerimenti aggiuntivi, sentiti libero di condividerli qui:

Appendice B Grafici

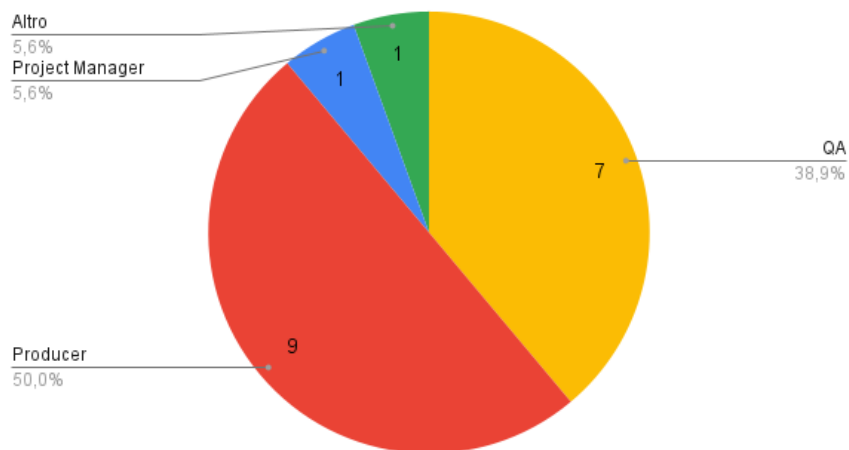
B.1 Grafici Demografici

Come i partecipanti hanno scoperto il sondaggio



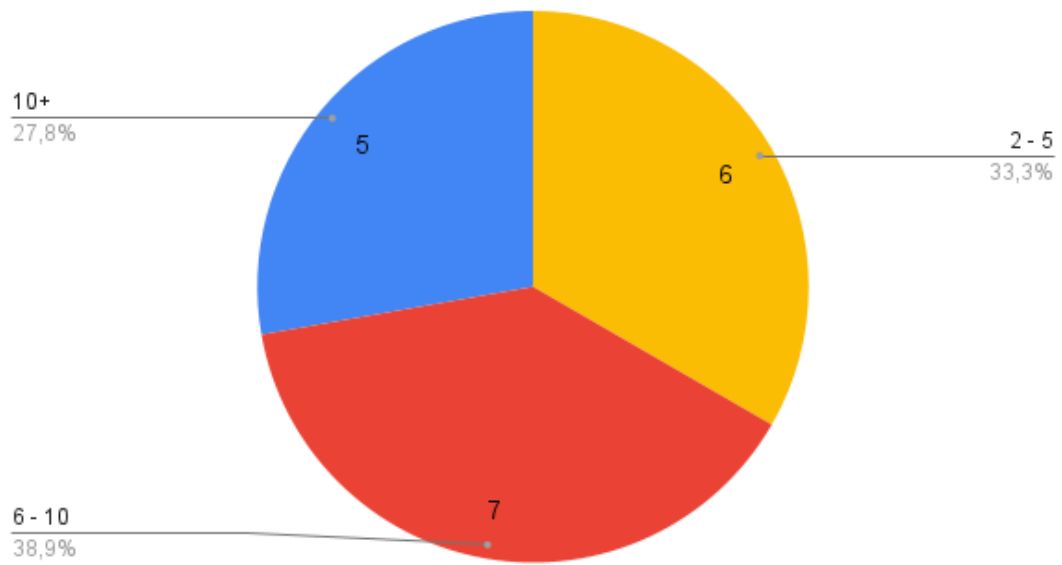
Come i partecipanti hanno scoperto il sondaggio

Ruoli dei partecipanti



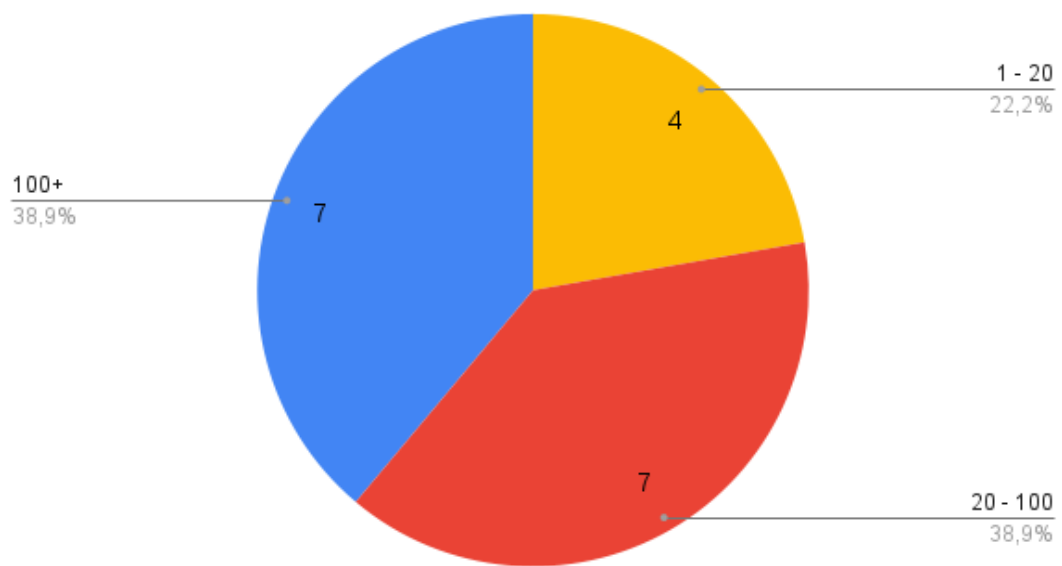
Ruoli dei partecipanti

Anni nell'industria dei partecipanti



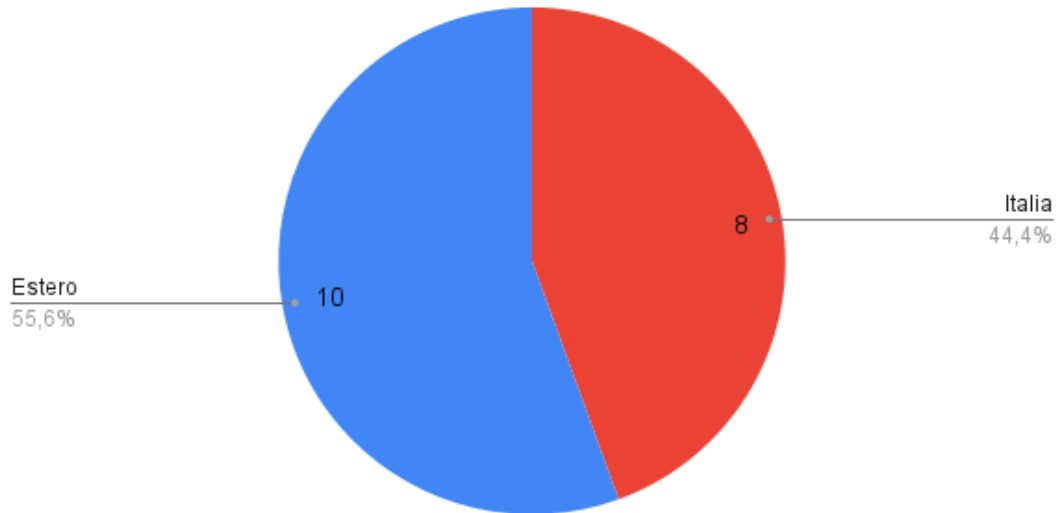
Anni nell'industria dei partecipanti

Dimensione dell'azienda dei partecipanti



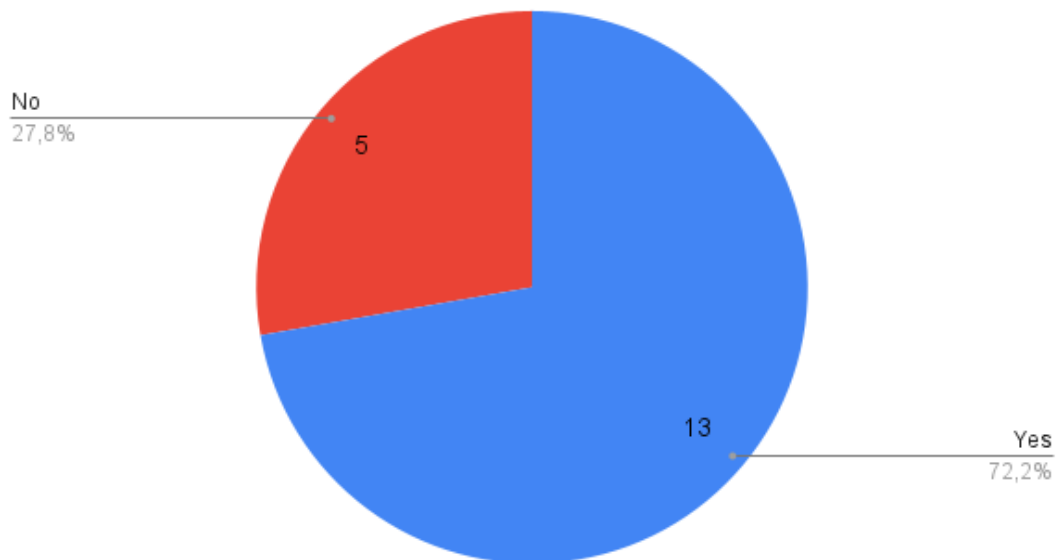
Dimensione dell'azienda dei partecipanti

Distribuzione dei partecipanti in base alla sede dell'azienda in Italia



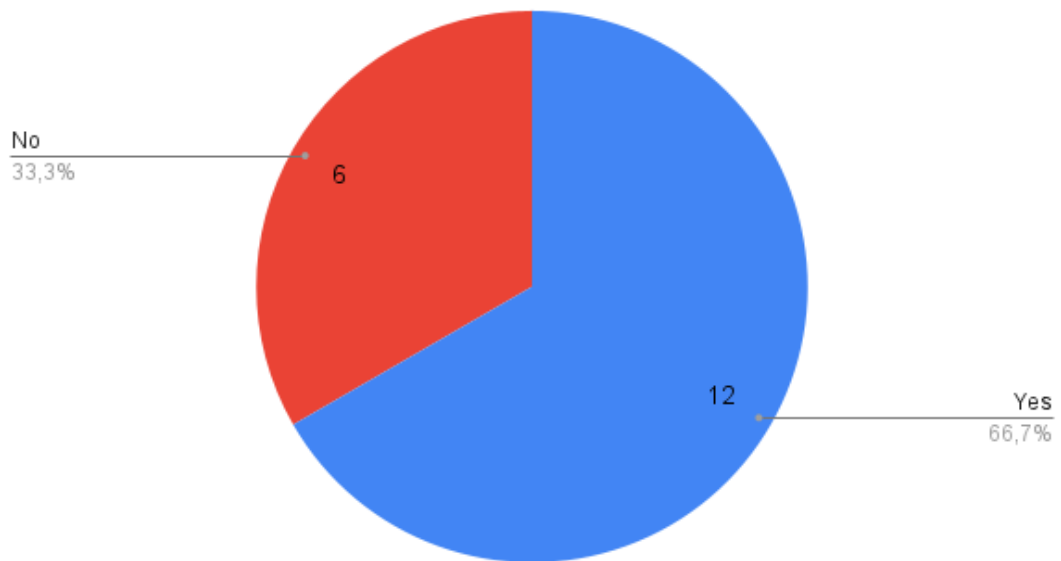
Distribuzione dei partecipanti in base alla sede dell'azienda in Italia

Team di QA presente in azienda



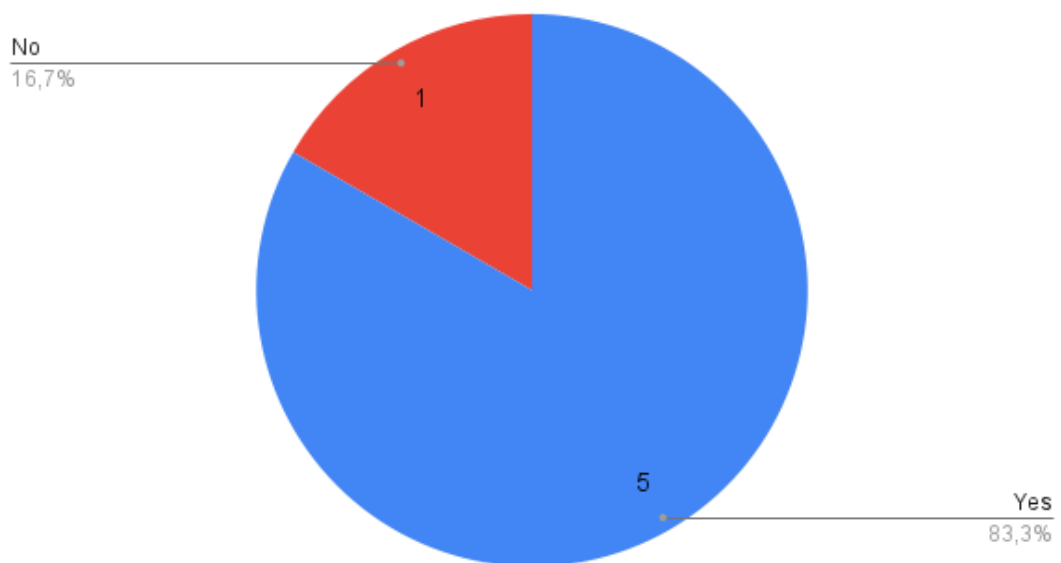
Team di QA presente in azienda

Partecipanti che eseguono comunque test nelle loro aziende



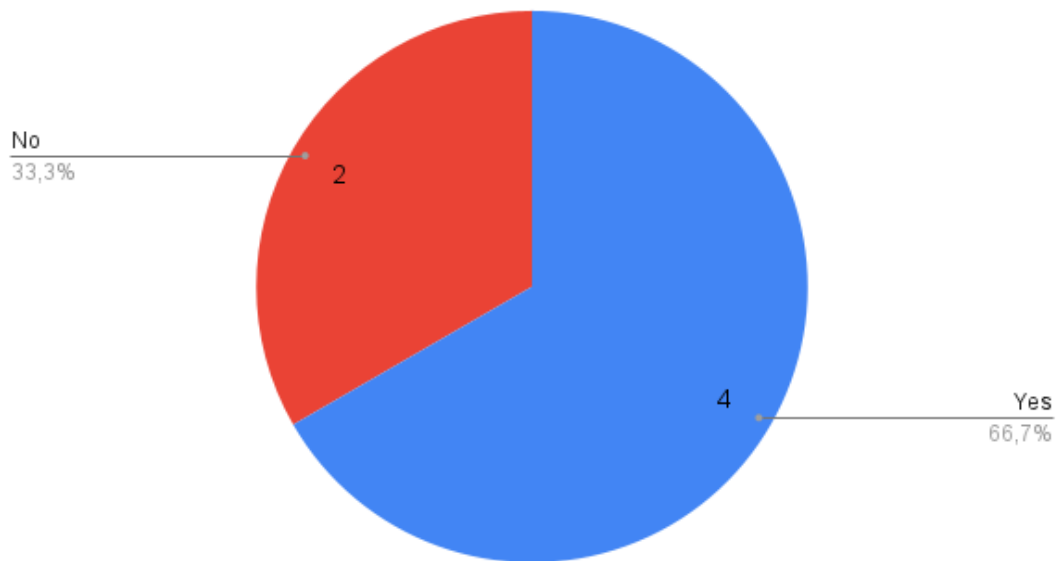
Partecipanti che eseguono comunque test nelle loro aziende

Aziende che si affidano a del QA esterno



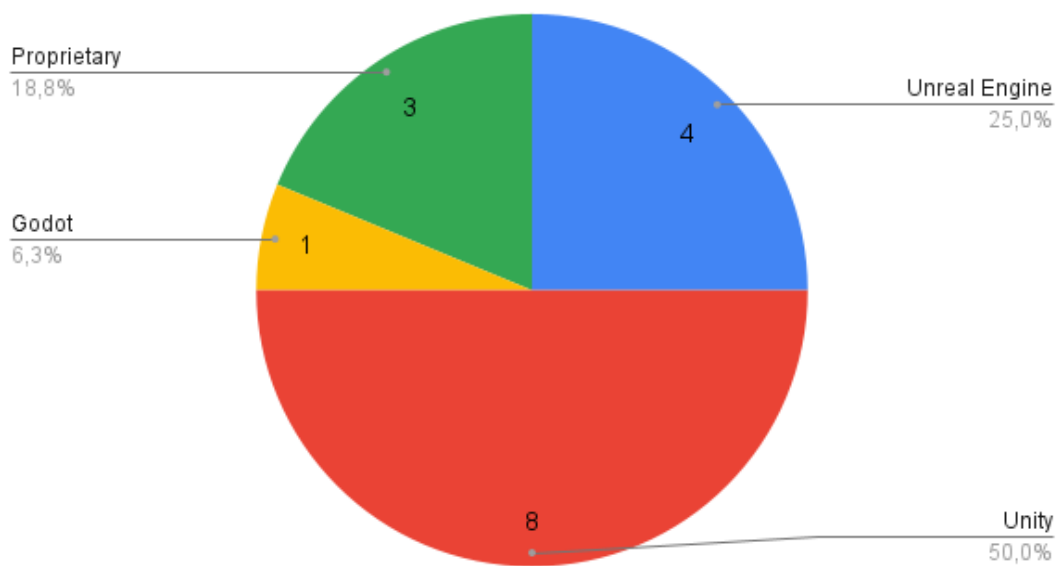
Aziende che si affidano a del QA esterno

Partecipanti che hanno assunto un ruolo di QA in passato



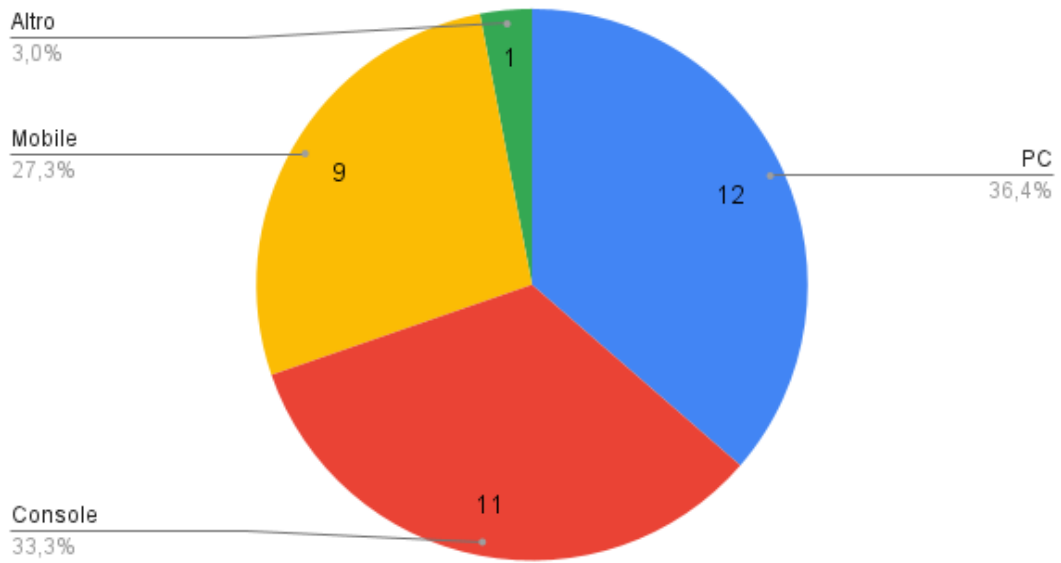
Partecipanti che hanno assunto un ruolo di QA in passato

Engine più usato dai partecipanti



Engine più usato dai partecipanti

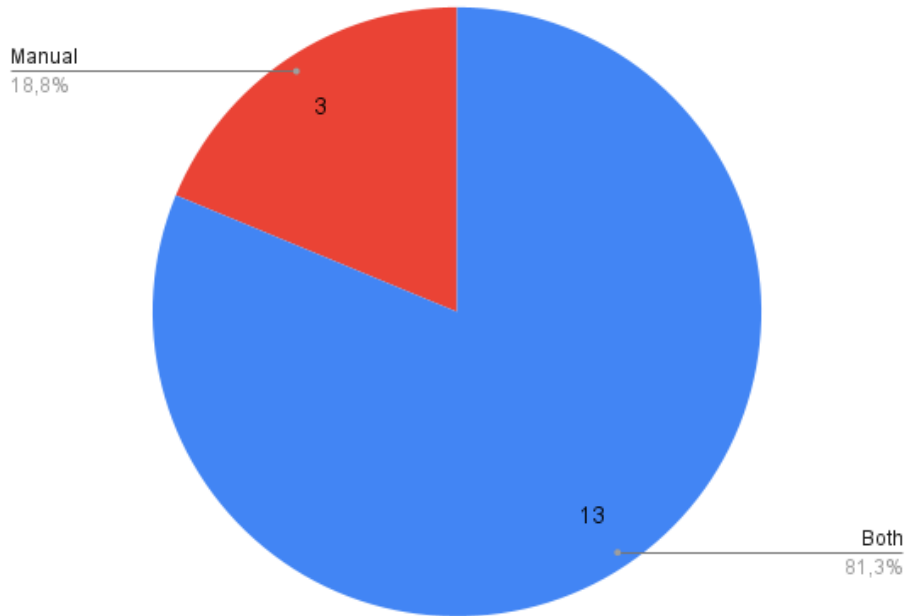
Piattaforma sulla quale sviluppare più usata dai partecipanti



Piattaforma sulla quale sviluppare più usata dai partecipanti

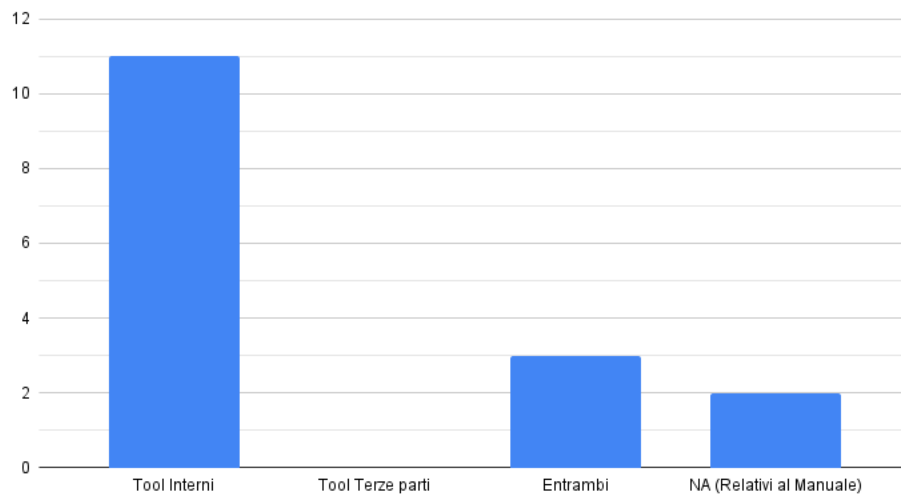
B.2 Grafici RQ1

Frequenza di come i test vengono condotti in azienda



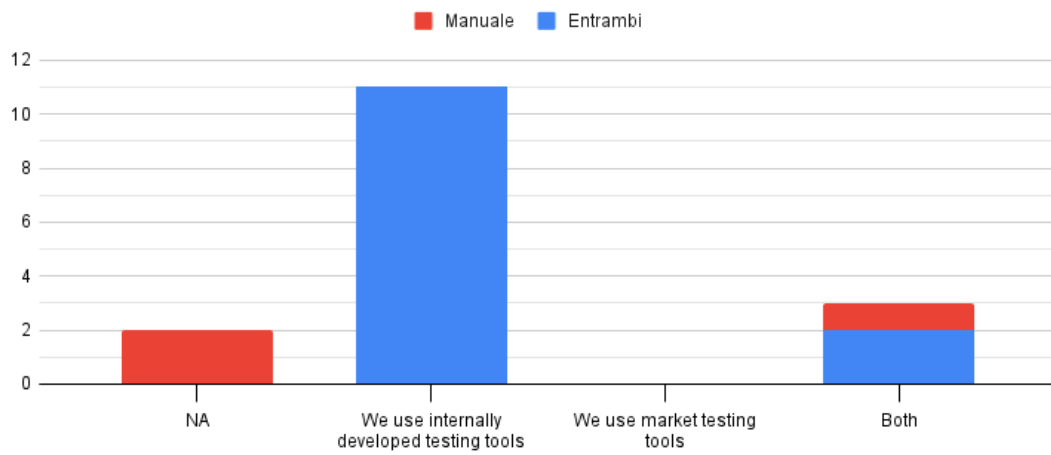
Frequenza di come i test vengono condotti in azienda

Frequenza tool interni o esterni



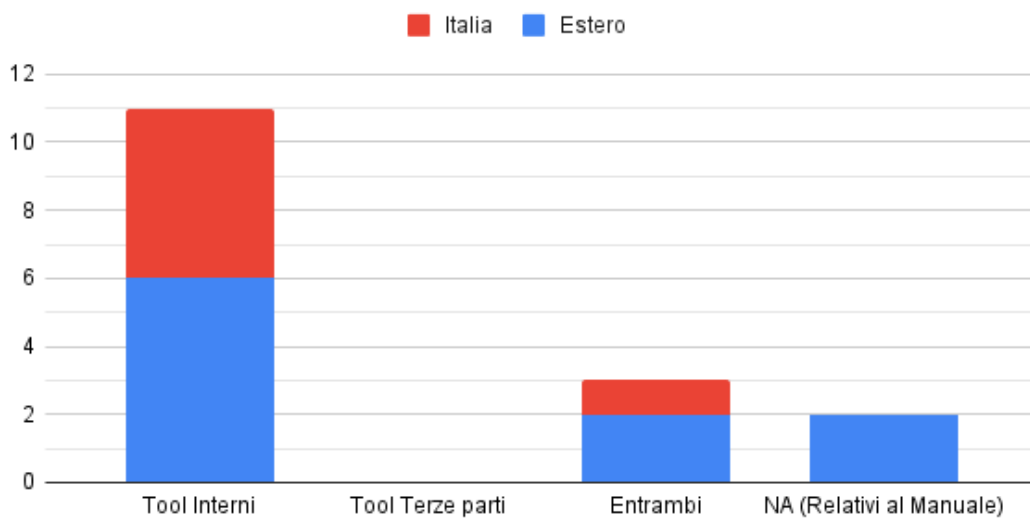
Frequenza tool interni o esterni

Come l'azienda effettua i test in base a se usa tool interni o esterni



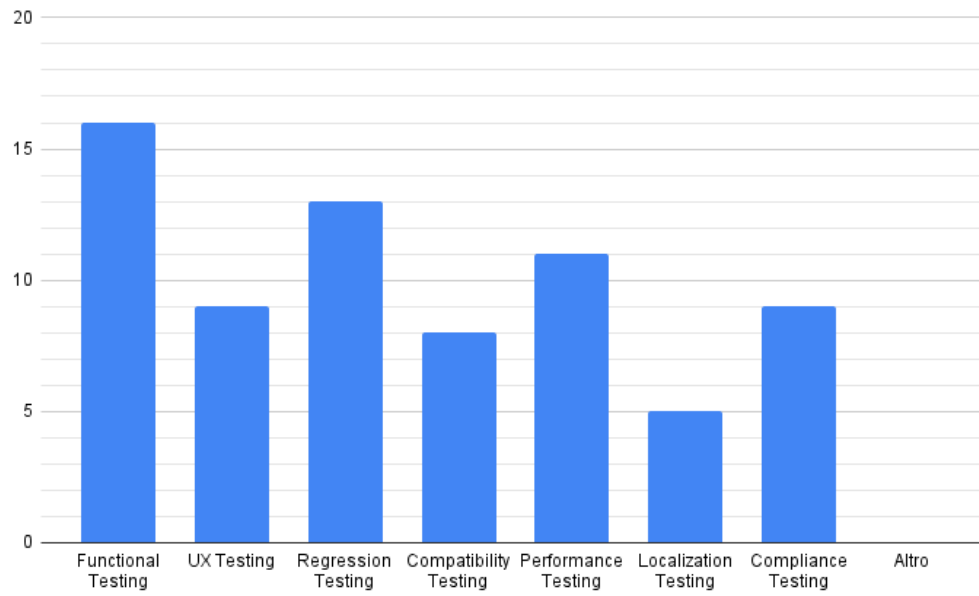
Come l'azienda effettua i test in base a se usa tool interni o esterni

Utilizzo di tool interni o esterni in base alla posizione geografica dell'azienda



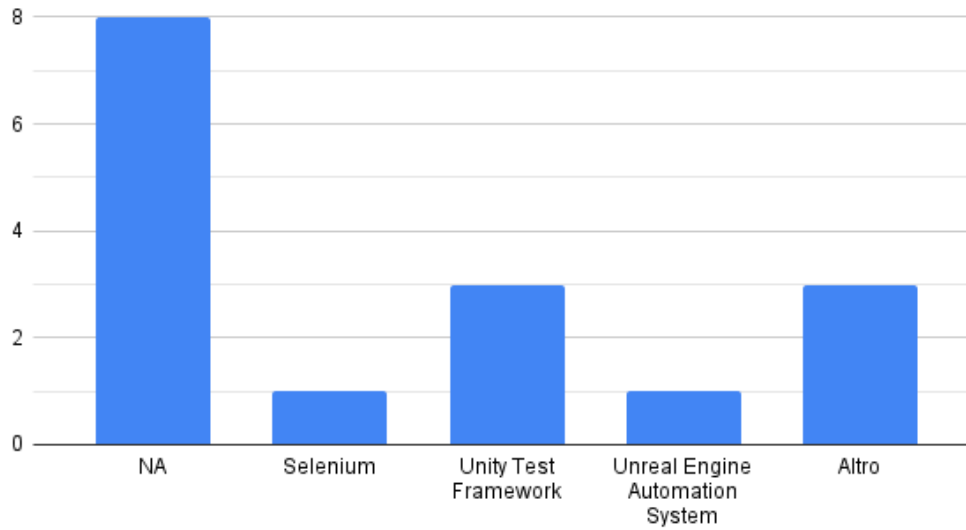
Utilizzo di tool interni o esterni in base alla posizione geografica dell'azienda

Frequenza tipi di test



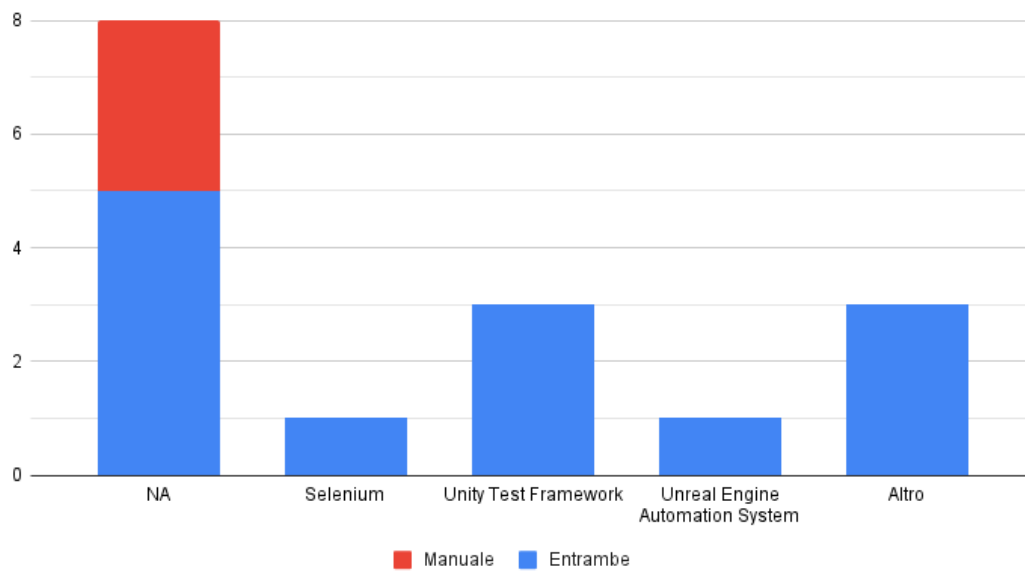
Frequenza tipi di test

Frequenza software di automazione



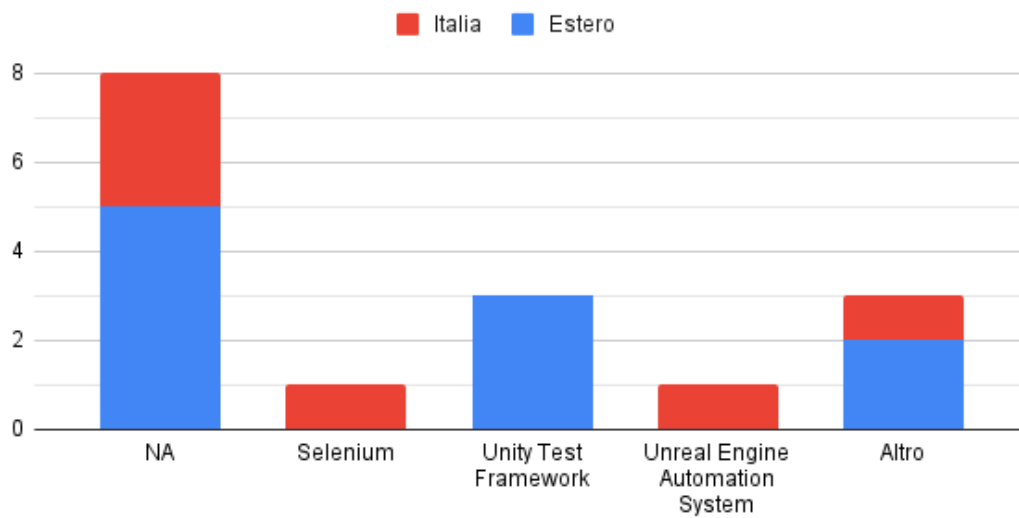
Frequenza software di automazione

Software per test automation in base a come l'azienda effettua i test



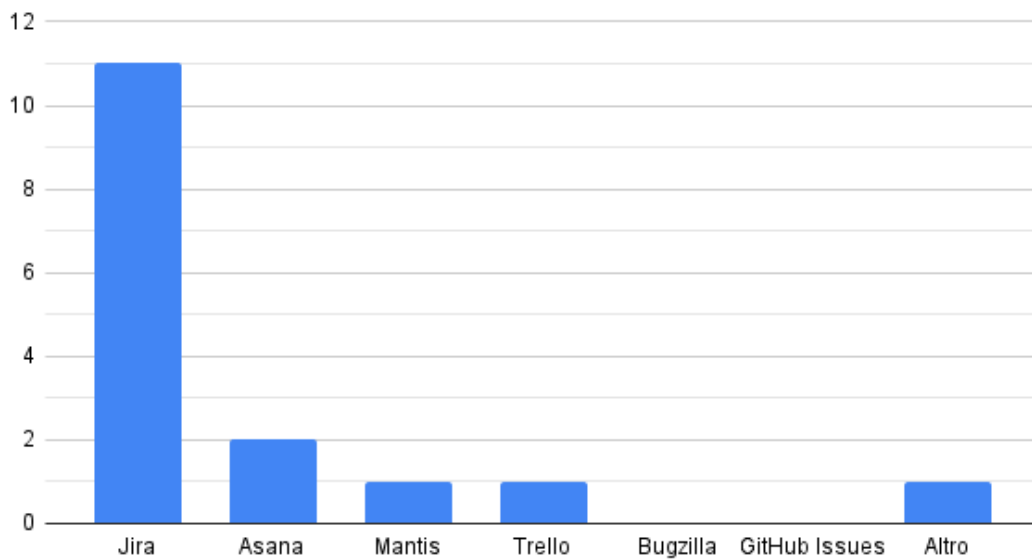
Software per test automation in base a come l'azienda effettua i test

Software di automazione in base alla posizione geografica dell'azienda



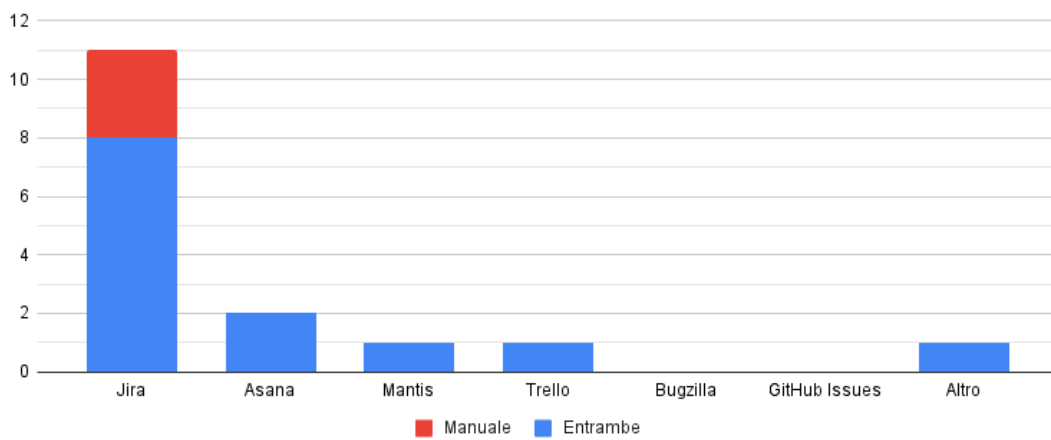
Software di automazione in base alla posizione geografica dell'azienda

Frequenza software di bug tracking



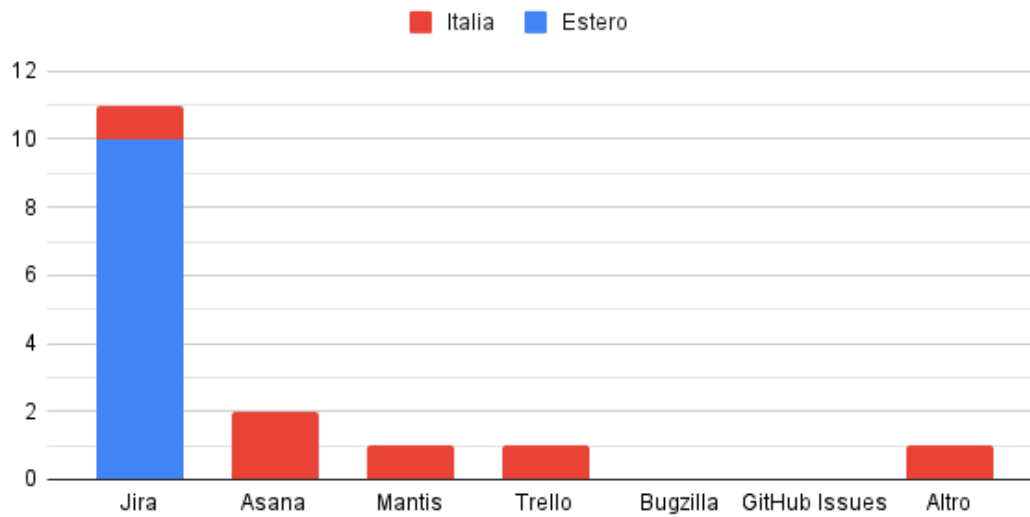
Frequenza software di bug tracking

Software per bug tracking in base a come l'azienda effettua i test



Software per bug tracking in base a come l'azienda effettua i test

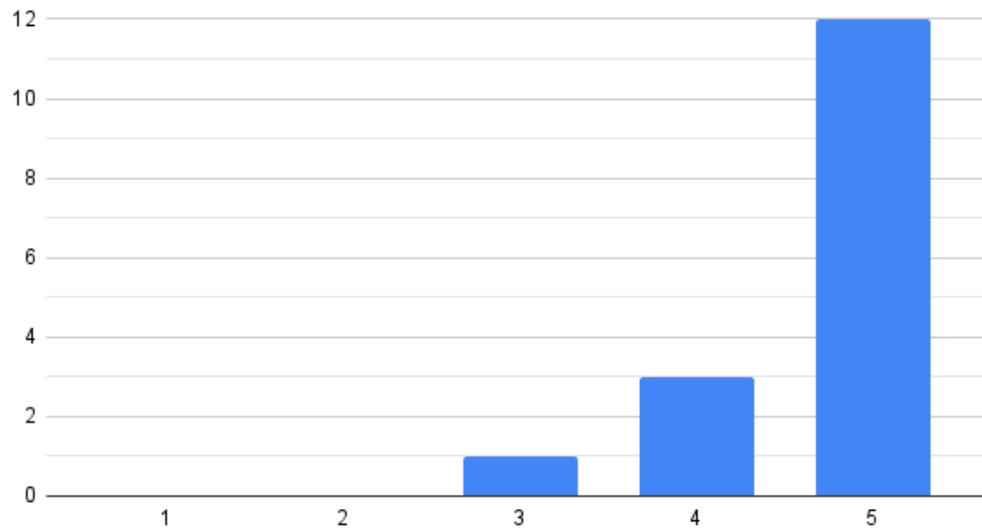
Software per bug tracking in base alla posizione geografica dell'azienda



Software per bug tracking in base alla posizione geografica dell'azienda

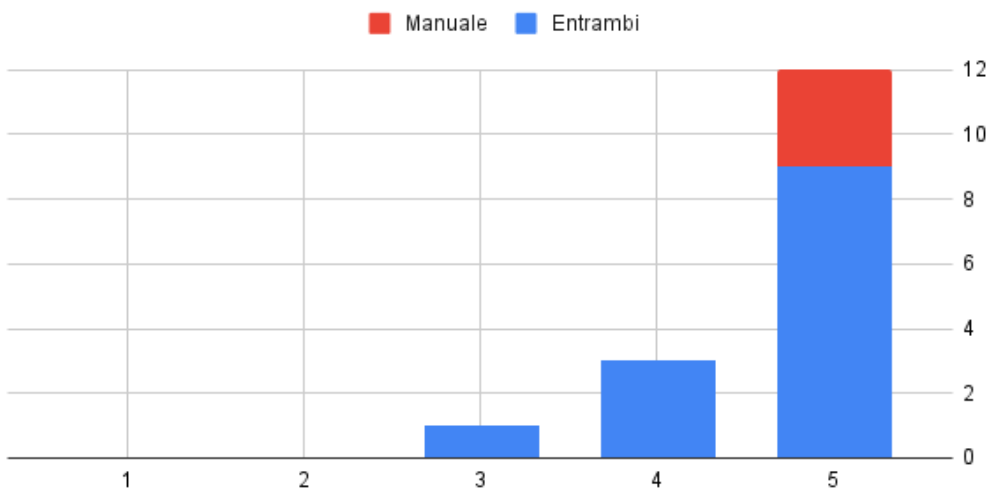
B.3 Grafici RQ2

Frequenza sull'importanza della comunicazione tra reparti



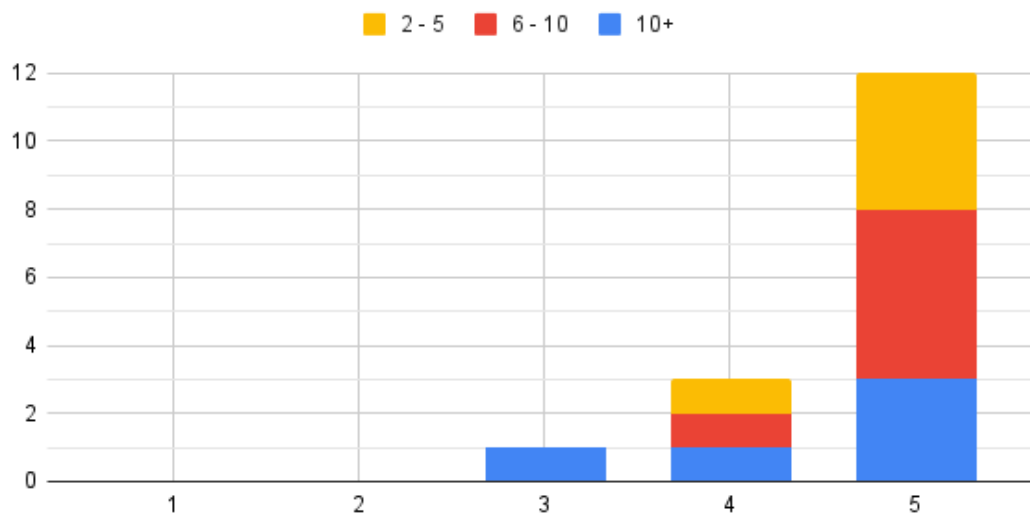
Frequenza sull'importanza della comunicazione tra reparti

Importanza comunicazione tra reparti in base a come l'azienda effettua i test



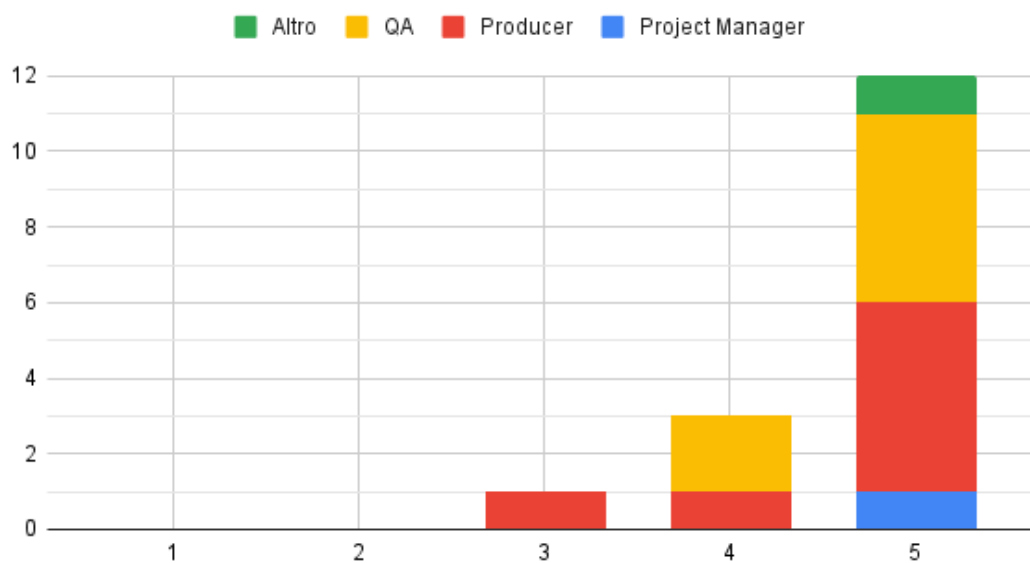
Importanza comunicazione tra reparti in base a come l'azienda effettua i test

Importanza comunicazione tra reparti in base agli anni di esperienza



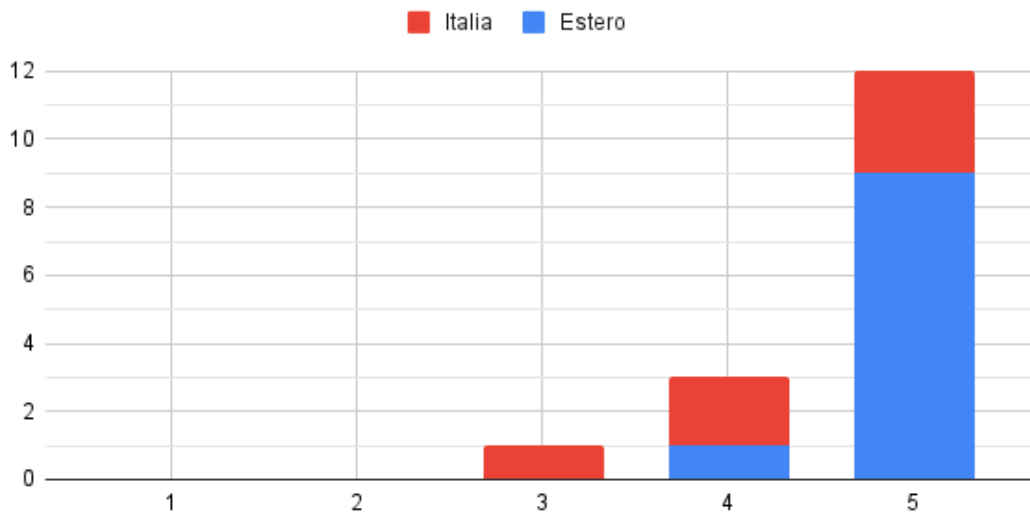
Importanza comunicazione tra reparti in base agli anni di esperienza

Importanza comunicazione tra reparti in base al Ruolo



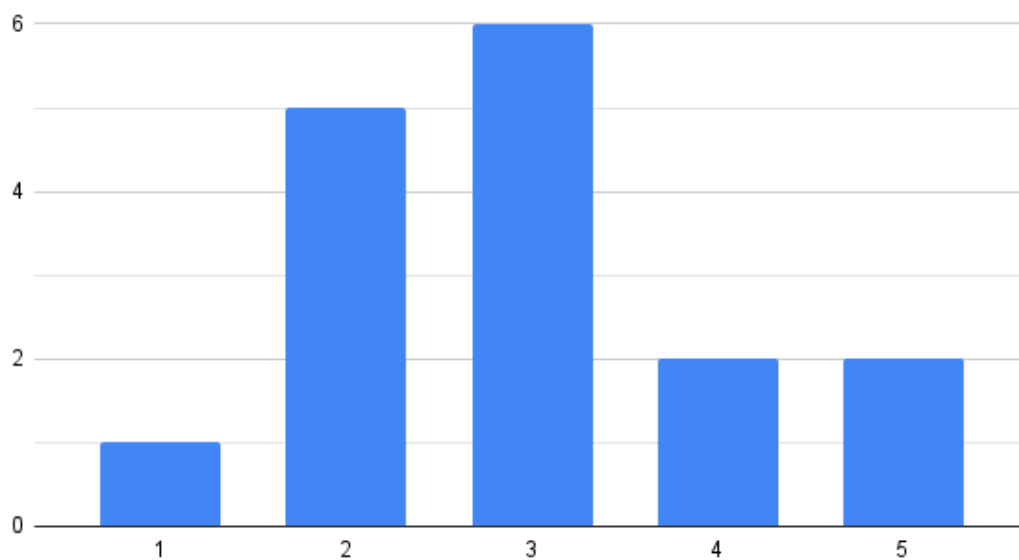
Importanza comunicazione tra reparti in base al Ruolo

Importanza comunicazione tra reparti in base alla posizione geografica dell'azienda



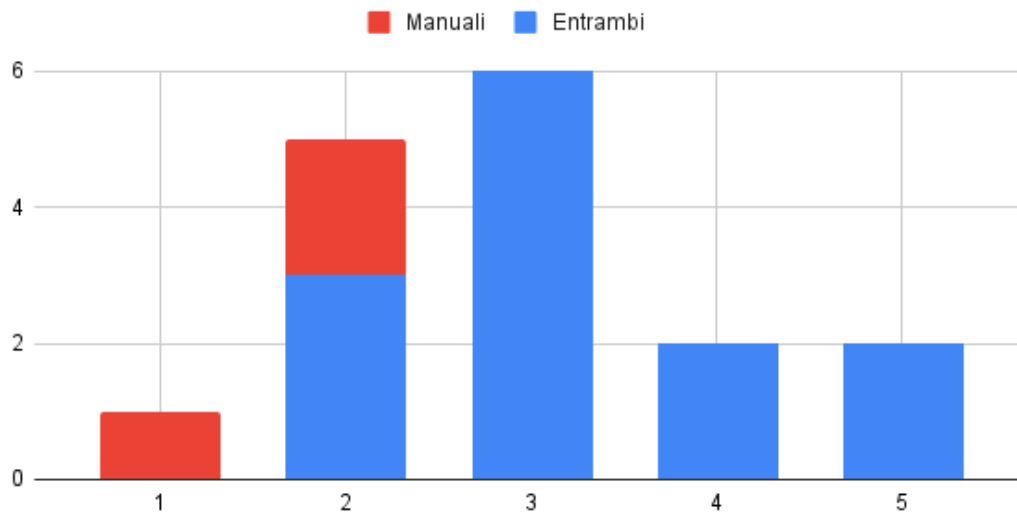
Importanza comunicazione tra reparti in base agli anni di esperienza

Frequenza su "I tool di automazione aiutano la comunicazione"



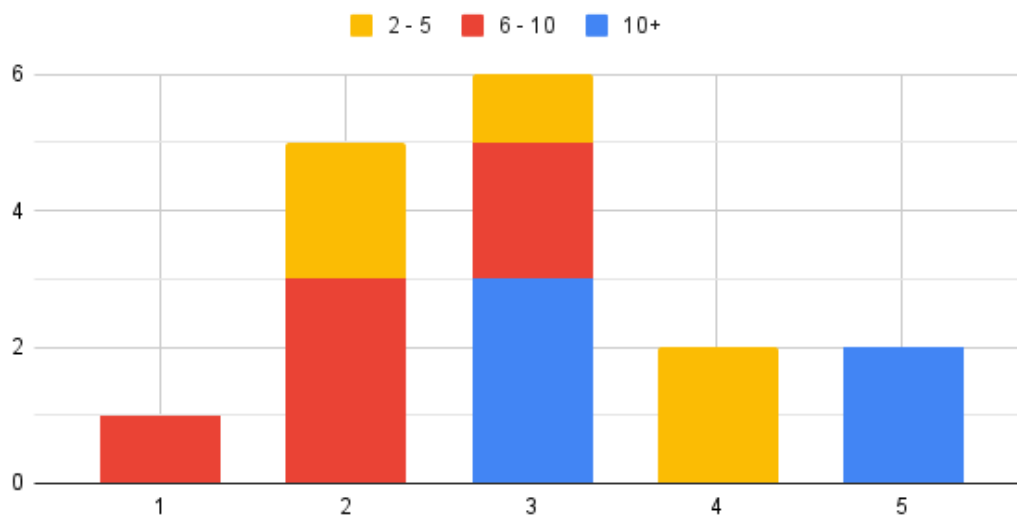
Frequenza su "I tool di automazione aiutano la comunicazione"

Parere su "I tool di automazione aiutano la comunicazione" in base a come l'azienda effettua i test



Parere su "I tool di automazione aiutano la comunicazione" in base a come l'azienda effettua i test

Parere su "I tool di automazione aiutano la comunicazione" in base agli anni di esperienza



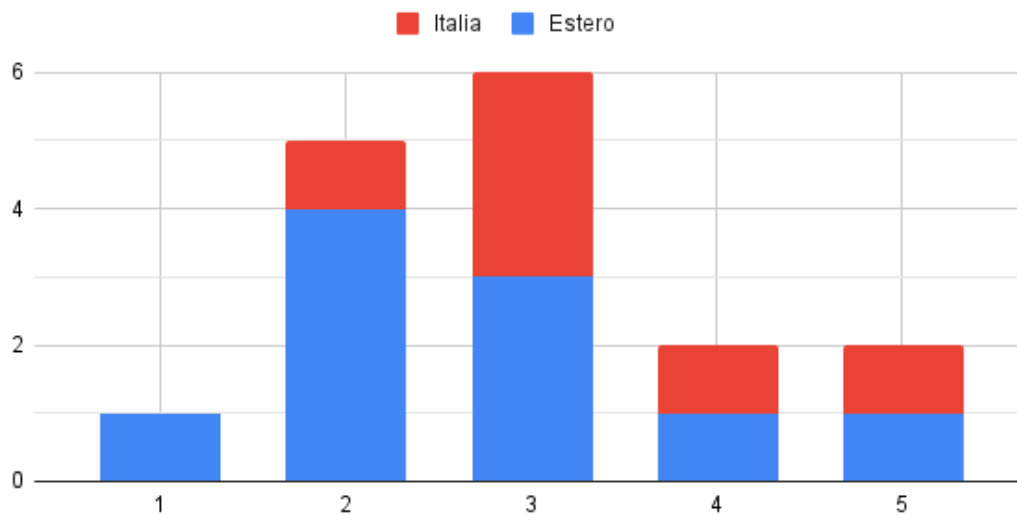
Parere su "I tool di automazione aiutano la comunicazione" in base agli anni di esperienza

Parere su "I tool di automazione aiutano la comunicazione" in base al Ruolo



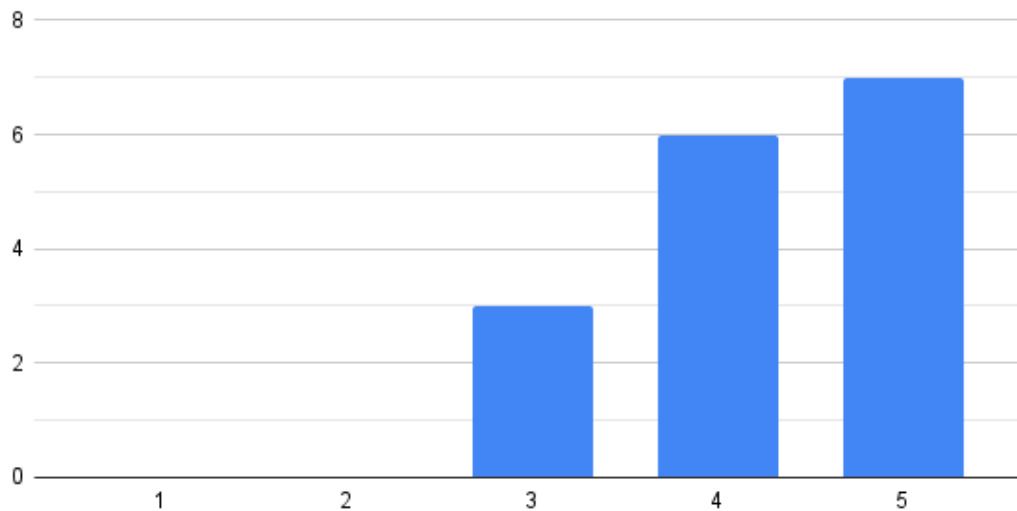
Parere su "I tool di automazione aiutano la comunicazione" in base al Ruolo

Parere su "I tool di automazione aiutano la comunicazione" in base alla posizione geografica dell'azienda



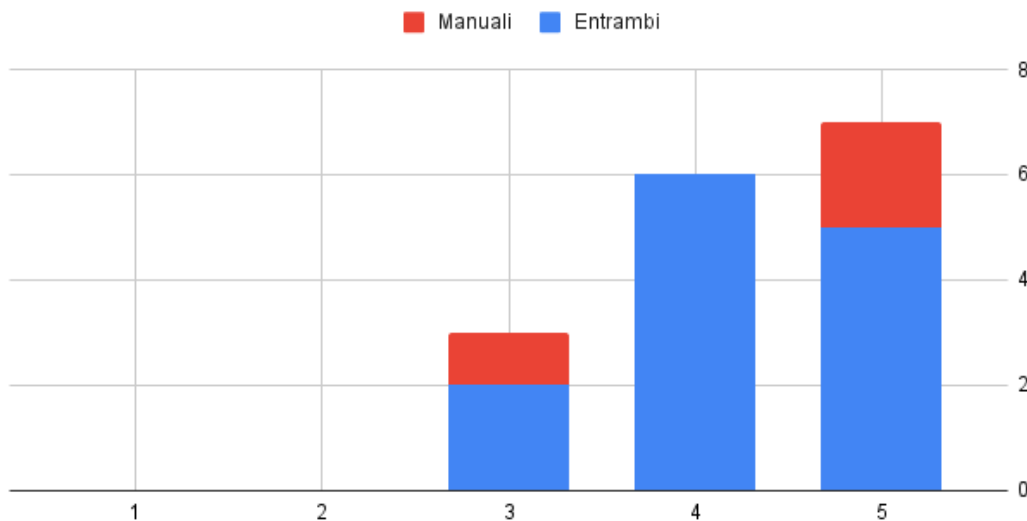
Parere su "I tool di automazione aiutano la comunicazione" in base alla posizione geografica dell'azienda

Frequenza su "Il feedback del team QA è essenziale per il successo di un videogioco"



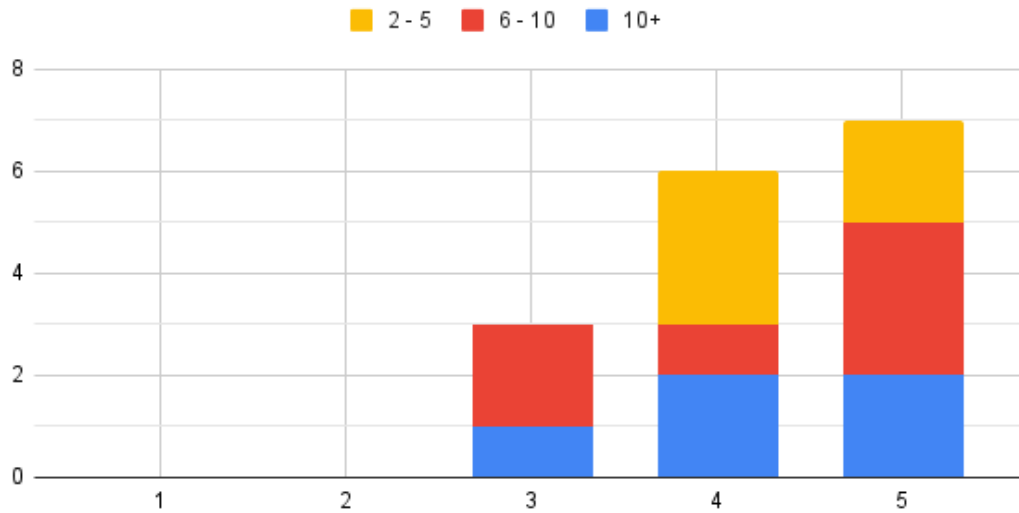
Frequenza su "Il feedback del team QA è essenziale per il successo di un videogioco"

Parere su "Il feedback del team QA è essenziale per il successo di un videogioco" in base a come l'azienda effettua i test



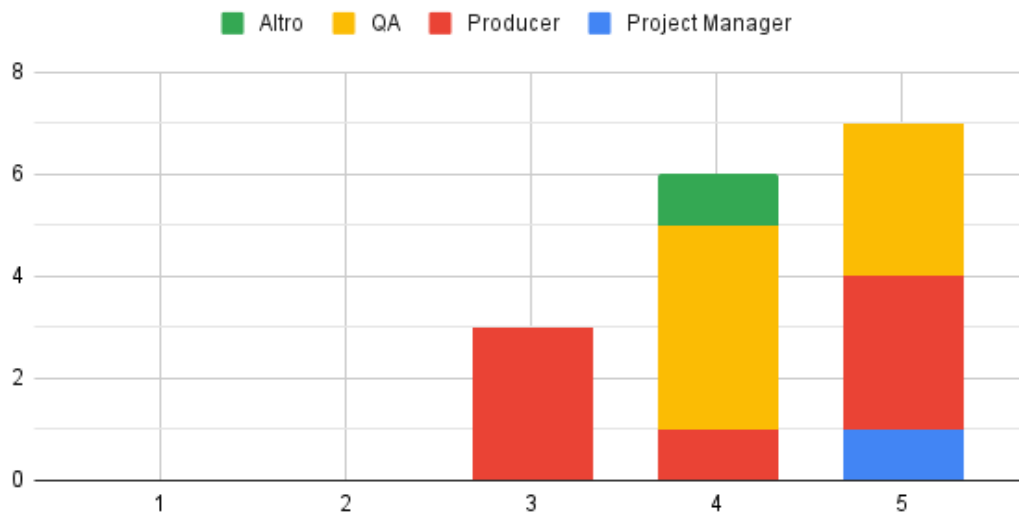
Parere su "Il feedback del team QA è essenziale per il successo di un videogioco" in base a come l'azienda effettua i test

Parere su "Il feedback del team QA è essenziale per il successo di un videogioco" in base agli anni di esperienza



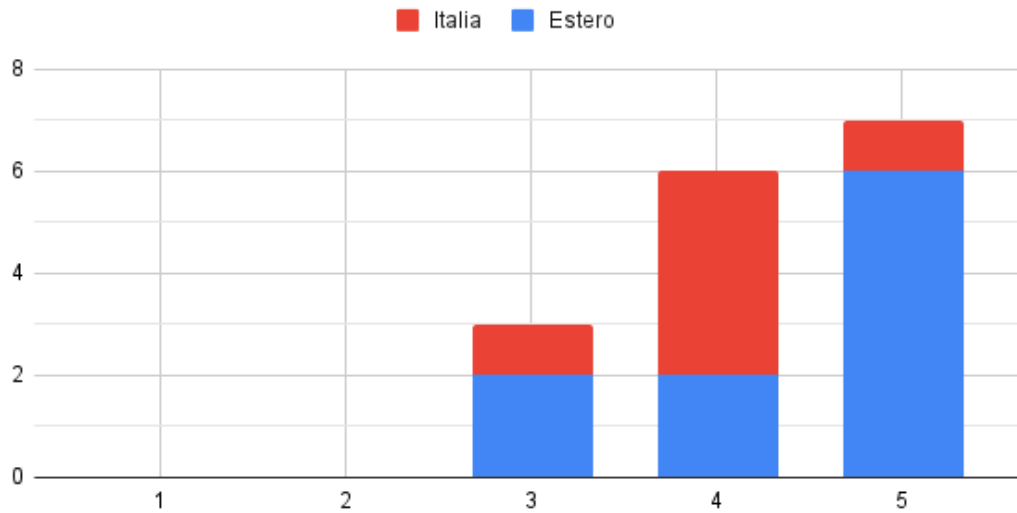
Parere su "Il feedback del team QA è essenziale per il successo di un videogioco" in base agli anni di esperienza

Parere su "Il feedback del team QA è essenziale per il successo di un videogioco" in base al Ruolo



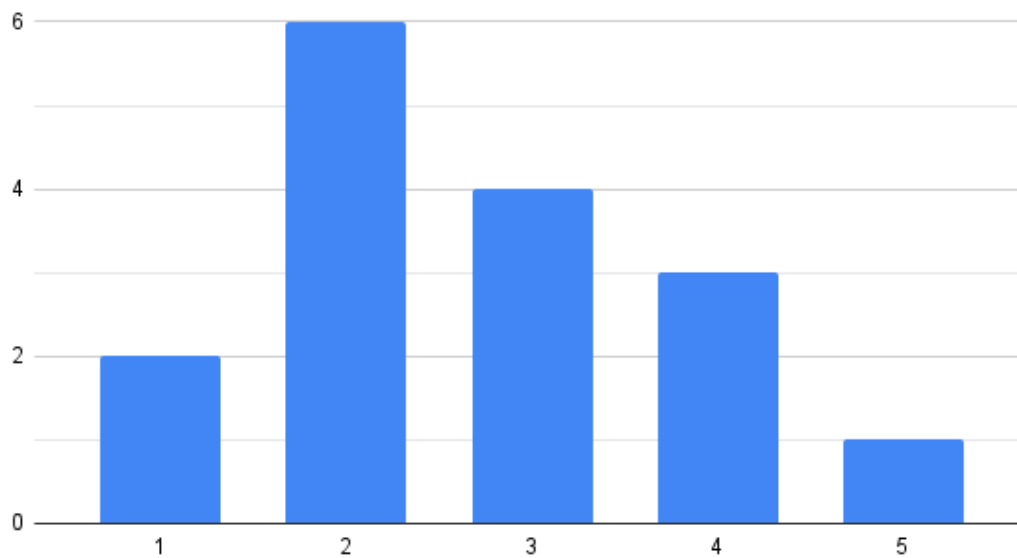
Parere su "Il feedback del team QA è essenziale per il successo di un videogioco" in base al Ruolo

Parere su "Il feedback del team QA è essenziale per il successo di un videogioco" in base alla posizione geografica



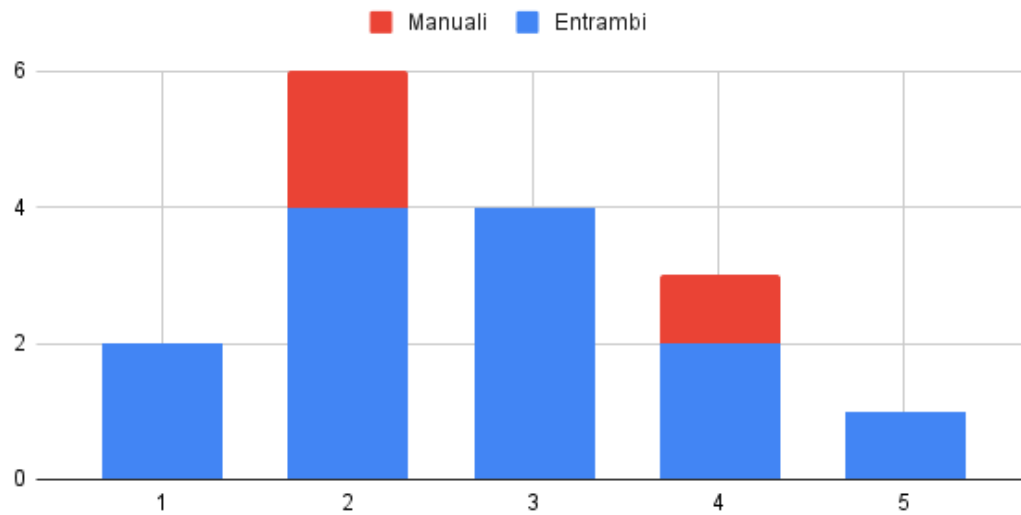
Parere su "Il feedback del team QA è essenziale per il successo di un videogioco" in base alla posizione geografica dell'azienda

Frequenza integrazione team QA all'inizio dello sviluppo



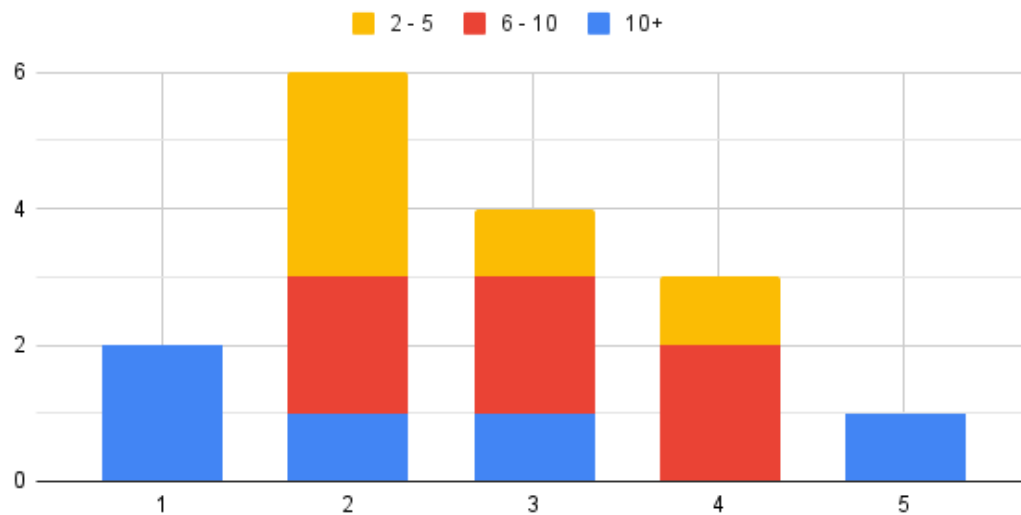
Frequenza integrazione team QA all'inizio dello sviluppo

Integrazione team QA all'inizio dello sviluppo in base a come l'azienda effettua i test



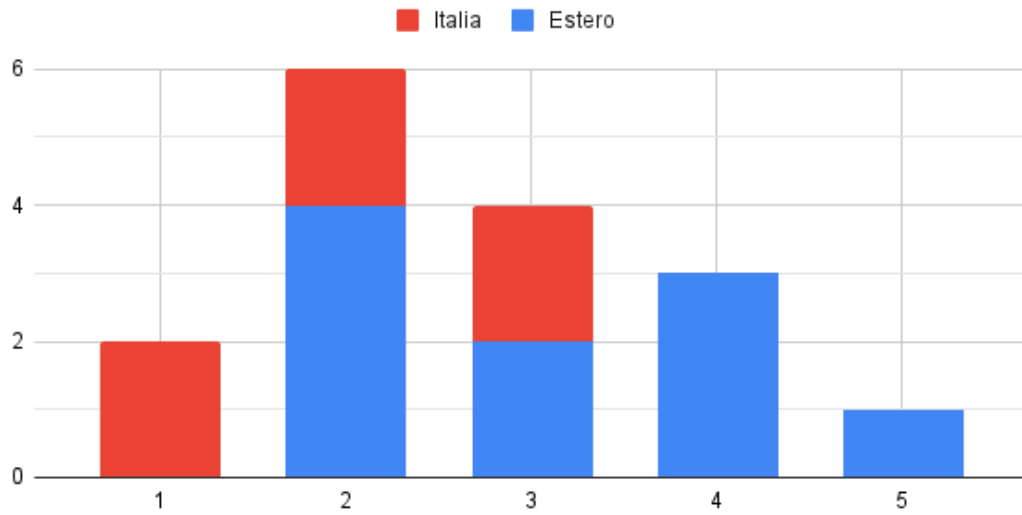
Integrazione team QA all'inizio dello sviluppo in base a come l'azienda effettua i test

Integrazione team QA all'inizio dello sviluppo in base agli anni di esperienza



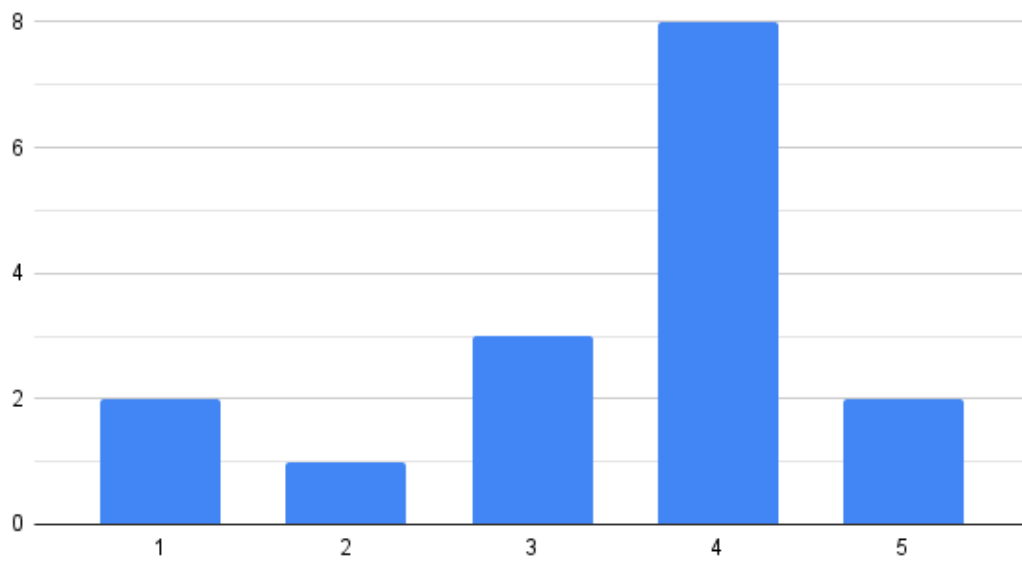
Integrazione team QA all'inizio dello sviluppo in base agli anni di esperienza

Integrazione team QA all'inizio dello sviluppo in base alla posizione geografica dell'azienda



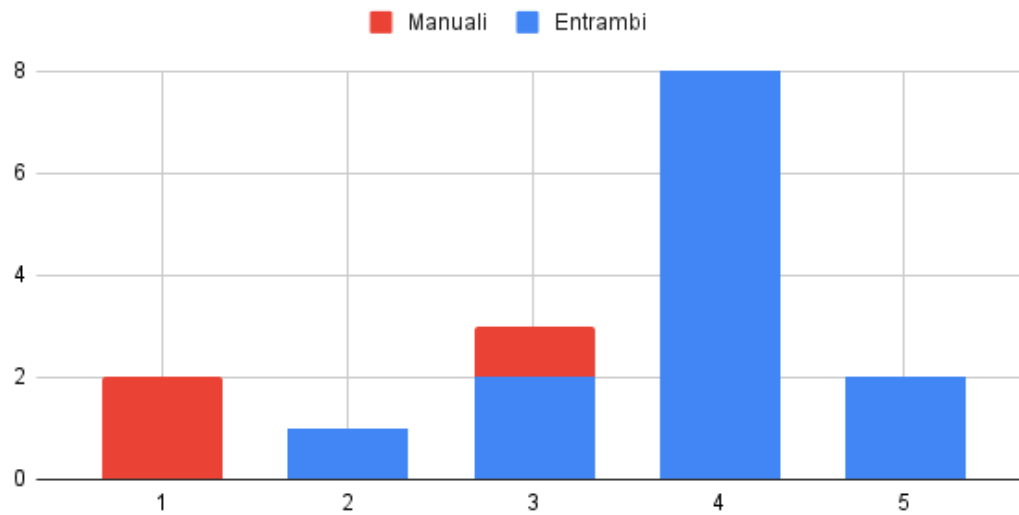
Integrazione team QA all'inizio dello sviluppo in base alla posizione geografica dell'azienda

Frequenza sull' "Utilità del QA tester nel programmare"



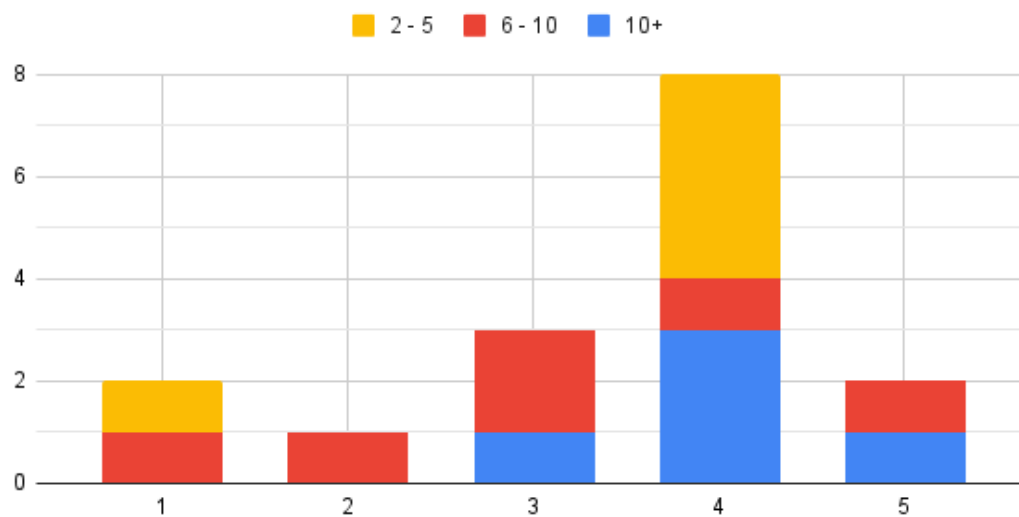
Frequenza sull' "Utilità del QA tester nel programmare"

Parere sull' "Utilità del QA tester nel programmare" in base a come l'azienda effettua i test



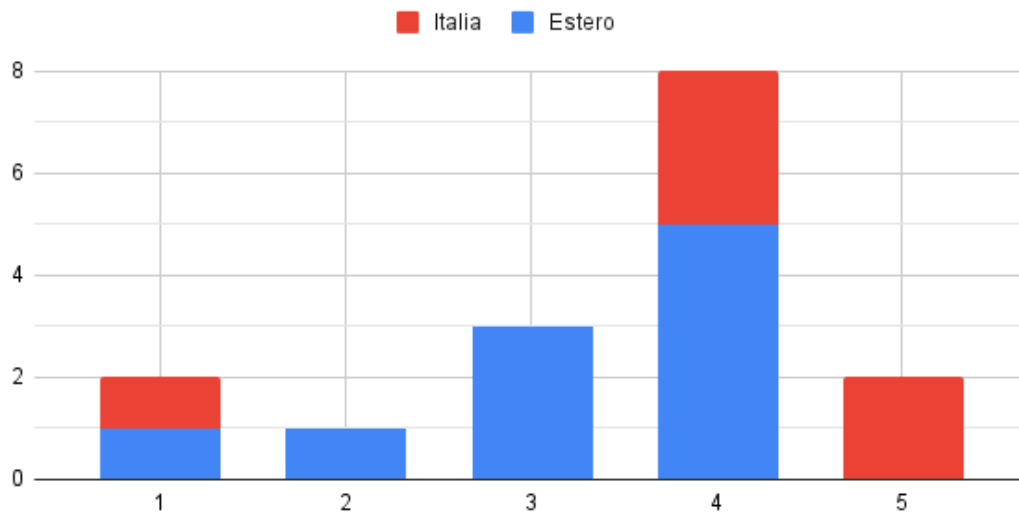
Parere sull' "Utilità del QA tester nel programmare" in base a come l'azienda effettua i test

Parere sull' "Utilità del QA tester nel programmare" in base agli anni di esperienza



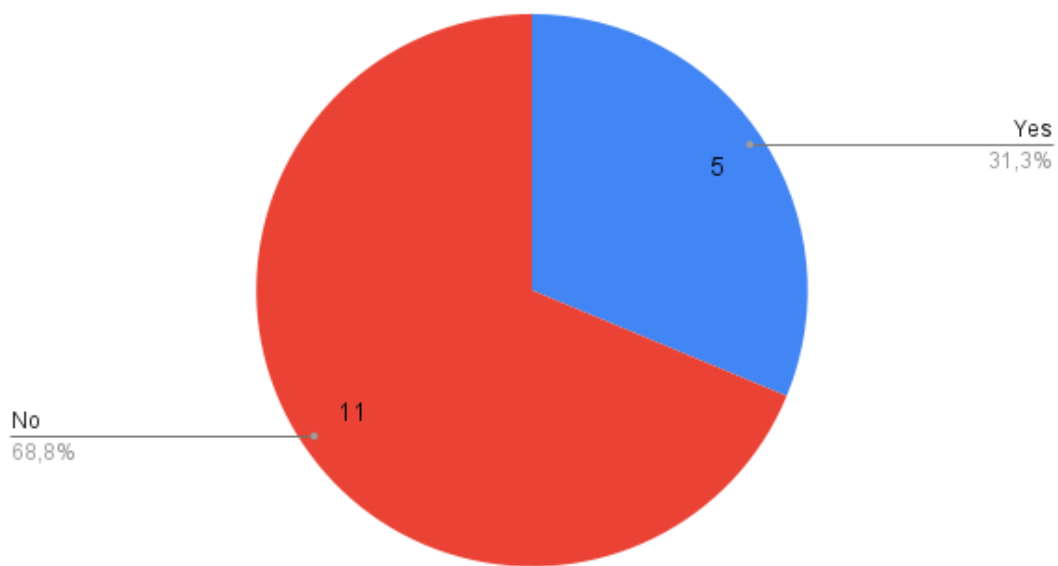
Parere sull' "Utilità del QA tester nel programmare" in base agli anni di esperienza

Parere sull' "Utilità del QA tester nel programmare" in base alla posizione geografica dell'azienda



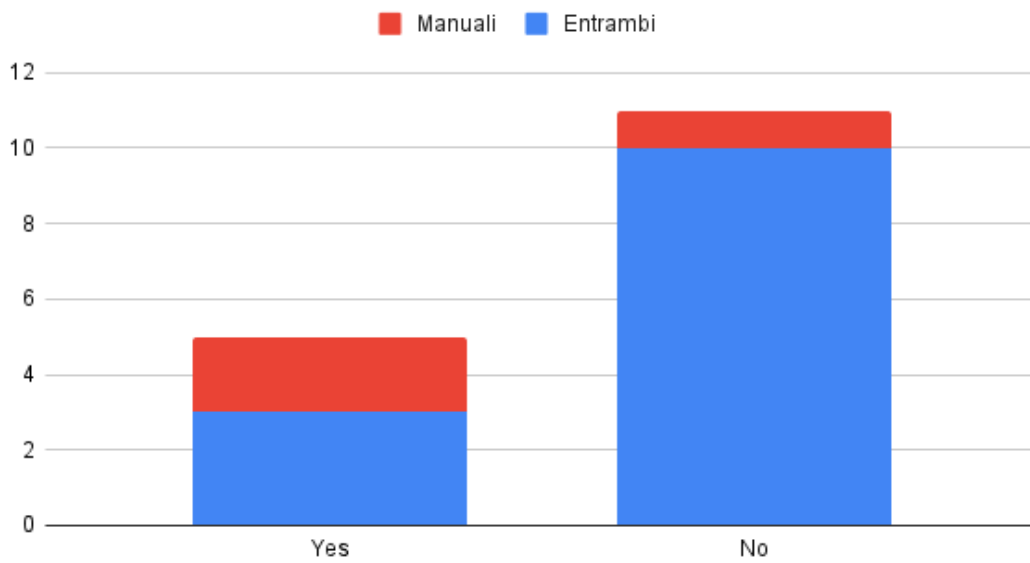
Parere sull' "Utilità del QA tester nel programmare" in base alla posizione geografica dell'azienda

Frequenza Quality Plan



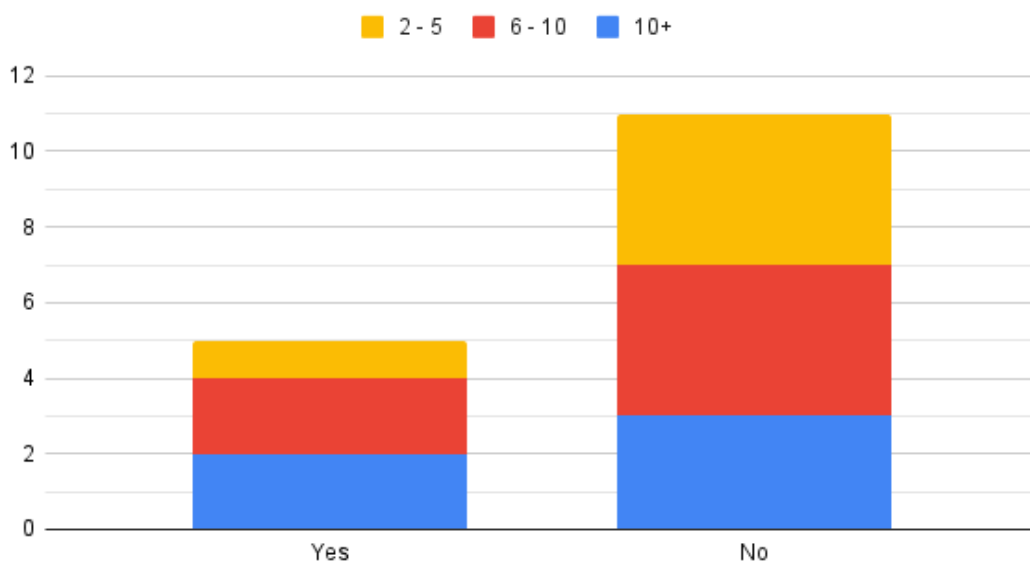
Frequenza Quality Plan

Quality Plan in base a come l'azienda effettua i test



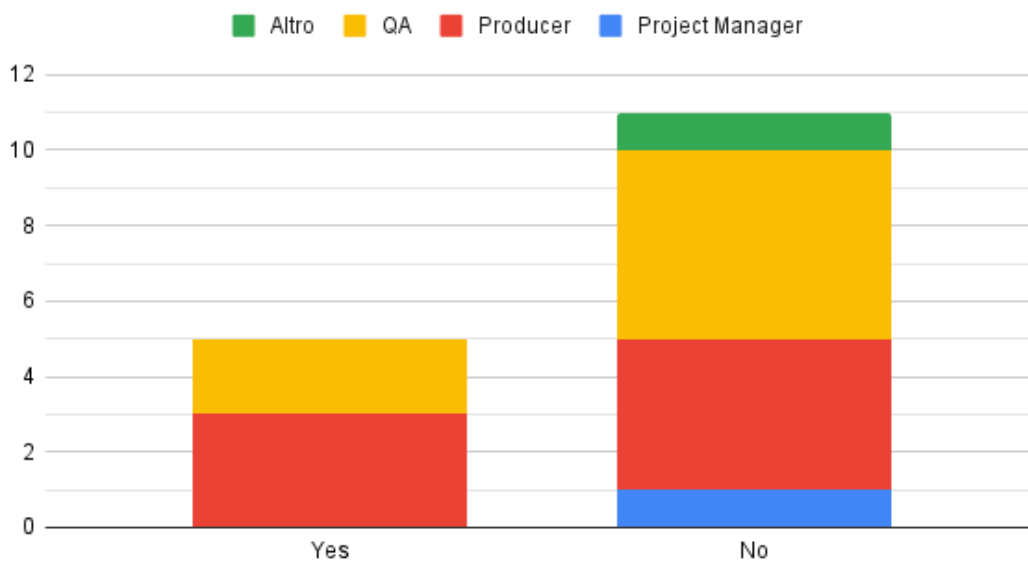
Quality Plan in base a come l'azienda effettua i test

Quality Plan in base agli anni di esperienza



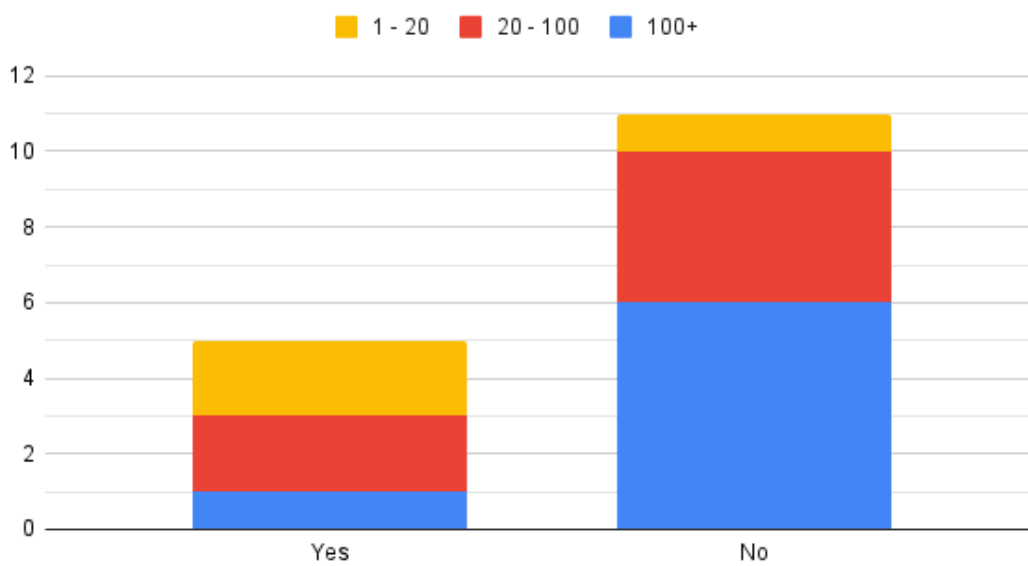
Quality Plan in base agli anni di esperienza

Quality Plan in base al Ruolo



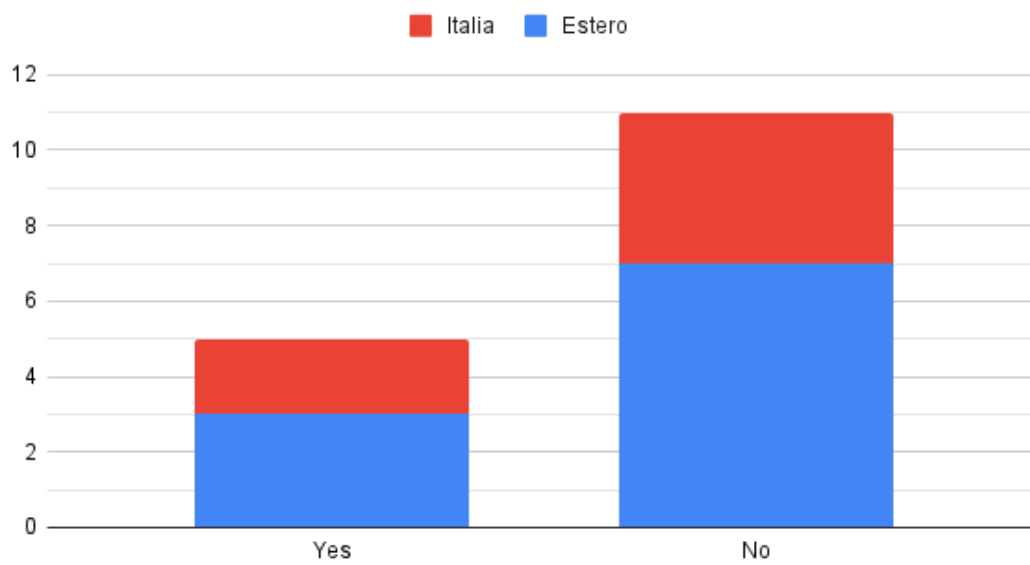
Quality Plan in base al Ruolo

Quality Plan in base alla dimensione dell'azienda



Quality Plan in base alla dimensione dell'azienda

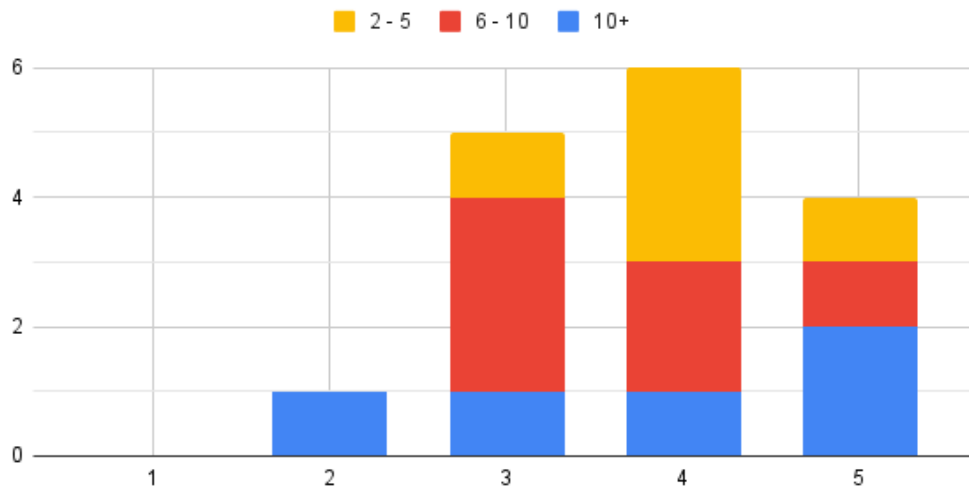
Quality Plan in base alla posizione geografica dell'azienda



Quality Plan in base alla posizione geografica dell'azienda

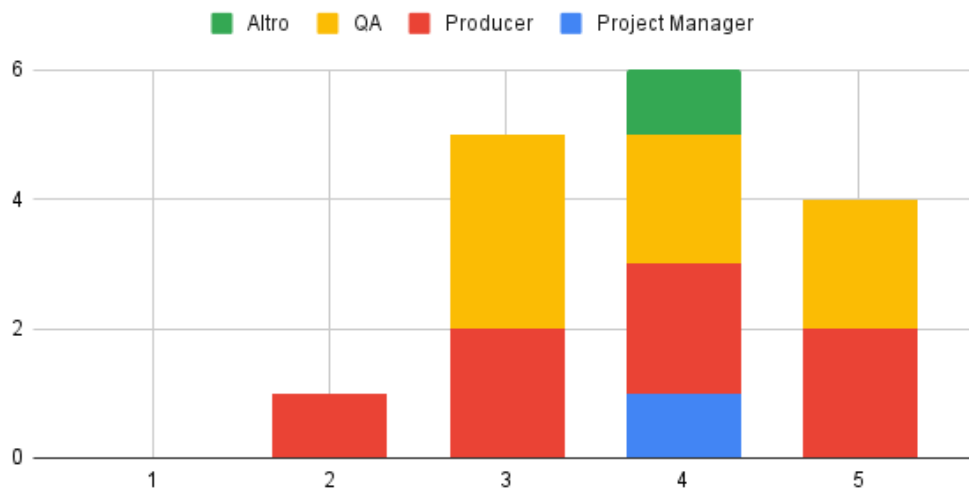
B.4 Grafici RQ3

L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base agli anni di esperienza



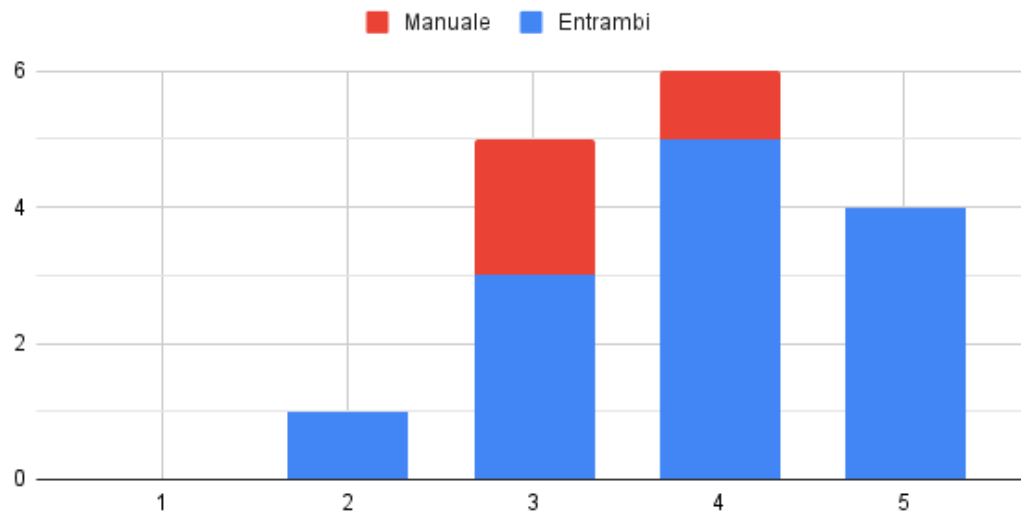
L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base agli anni di esperienza

L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base al Ruolo



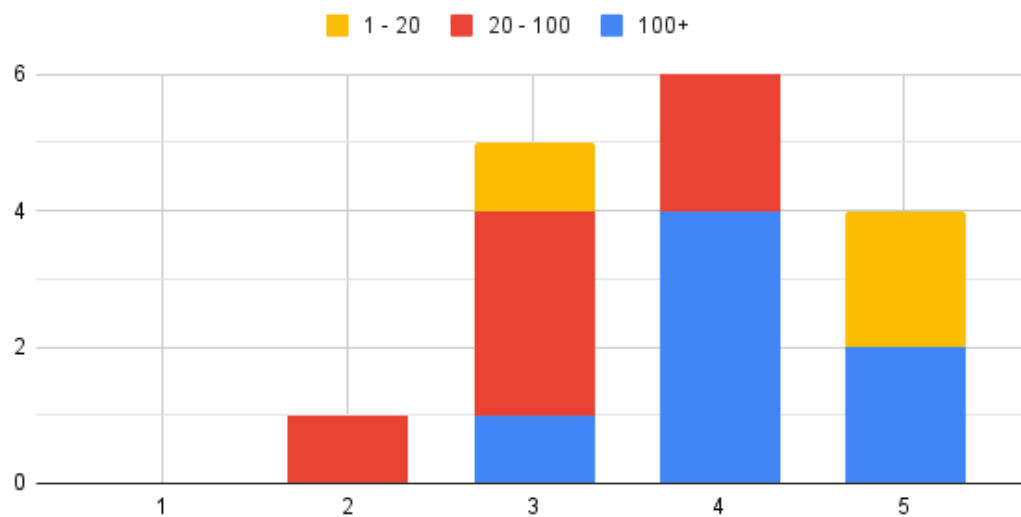
L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base al Ruolo

L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base al tipo di test



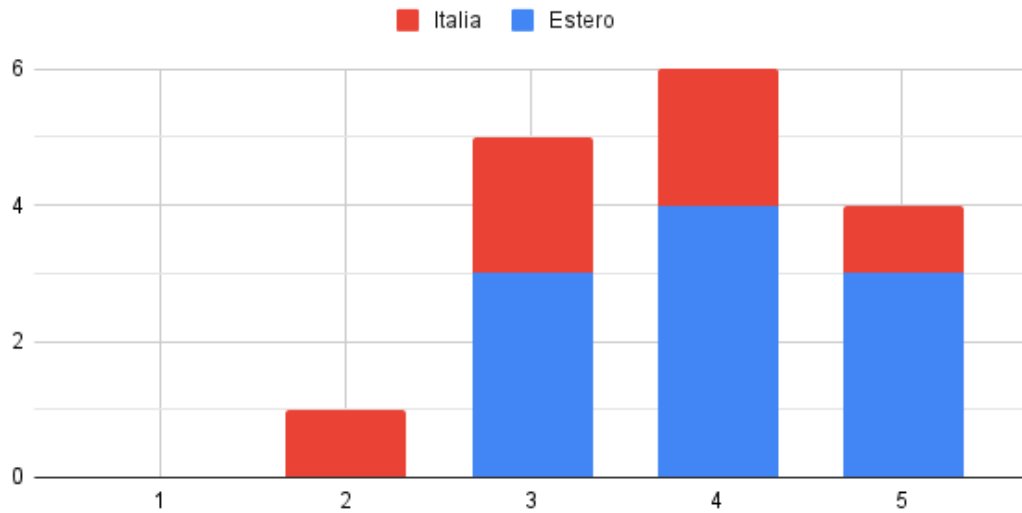
L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base al tipo di test

L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base alla dimensione dell'azienda



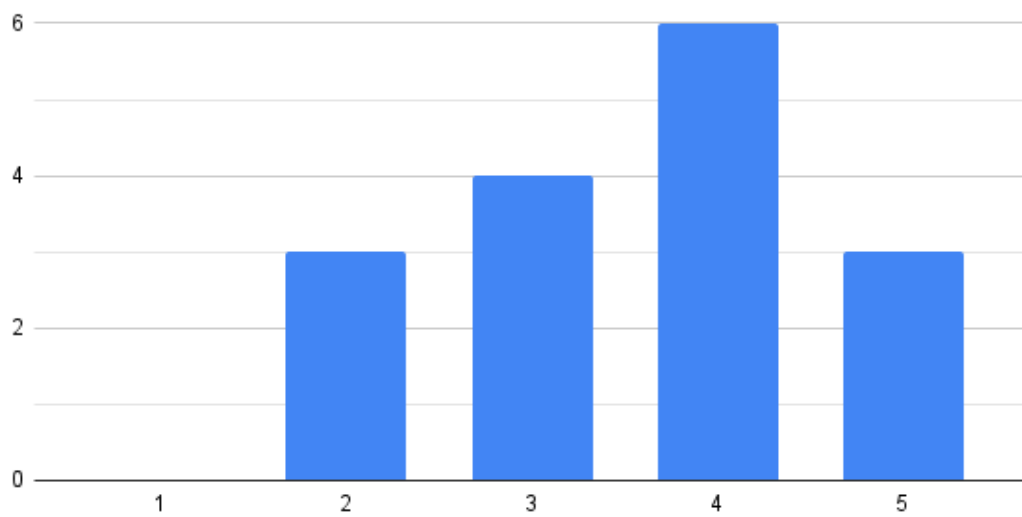
L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base alla dimensione dell'azienda

L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base alla posizione geografica dell'azienda



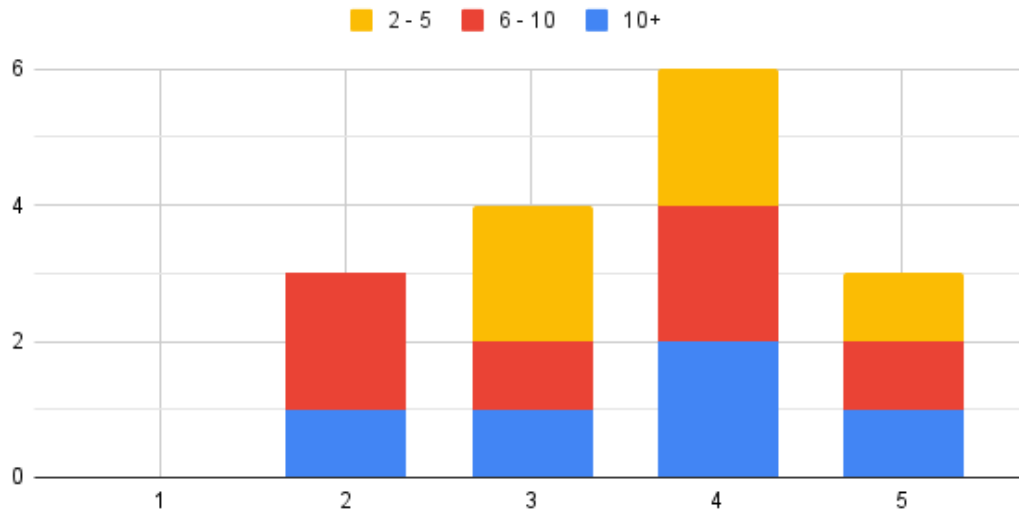
L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base alla posizione geografica dell'azienda

Frequenza su l'automazione aiuta a ridurre i tempi di sviluppo e rilascio



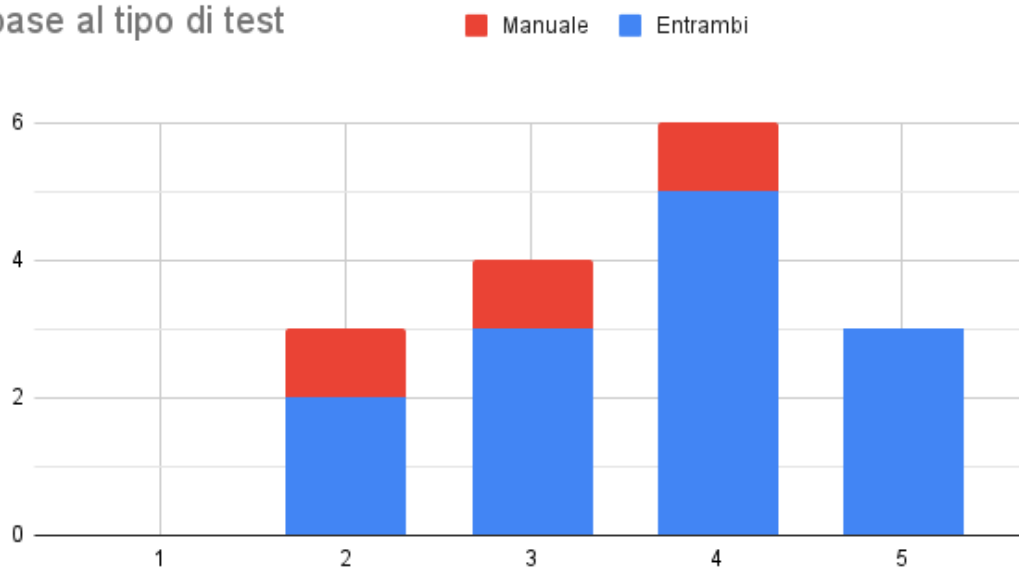
Frequenza su l'automazione aiuta a ridurre i tempi di sviluppo e rilascio

L'automazione aiuta a ridurre i tempi di sviluppo e rilascio in base agli anni di esperienza



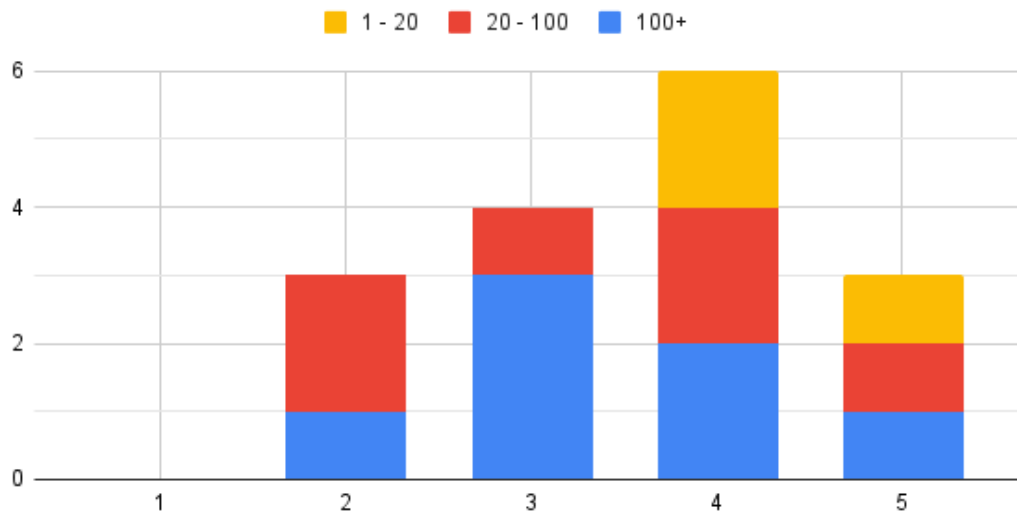
L'automazione può ridurre il numero di bug non rilevati durante lo sviluppo in base agli anni di esperienza

L'automazione aiuta a ridurre i tempi di sviluppo e rilascio in base al tipo di test



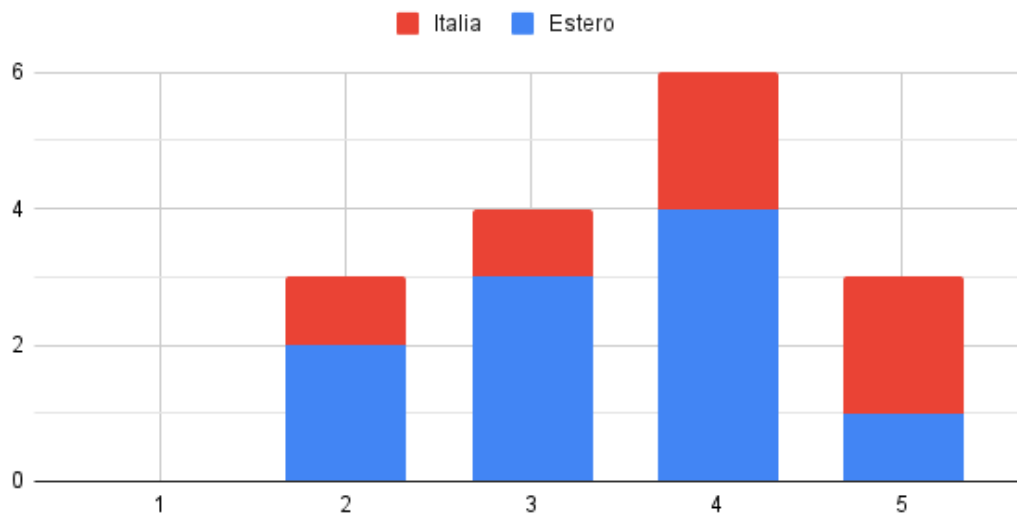
L'automazione aiuta a ridurre i tempi di sviluppo e rilascio in base al tipo di test

L'automazione aiuta a ridurre i tempi di sviluppo e rilascio in base alla dimensione dell'azienda



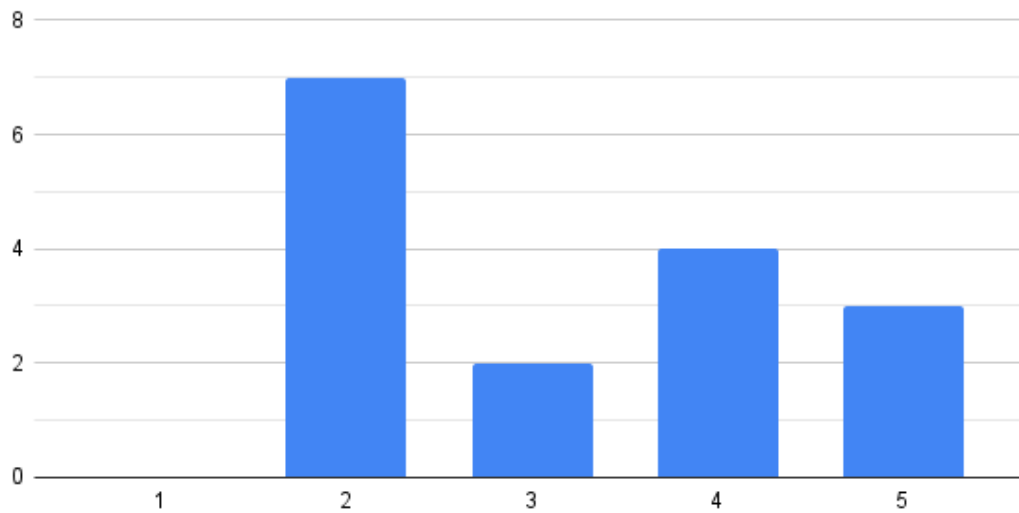
L'automazione aiuta a ridurre i tempi di sviluppo e rilascio in base alla dimensione dell'azienda

L'automazione aiuta a ridurre i tempi di sviluppo e rilascio in base alla posizione geografica dell'azienda



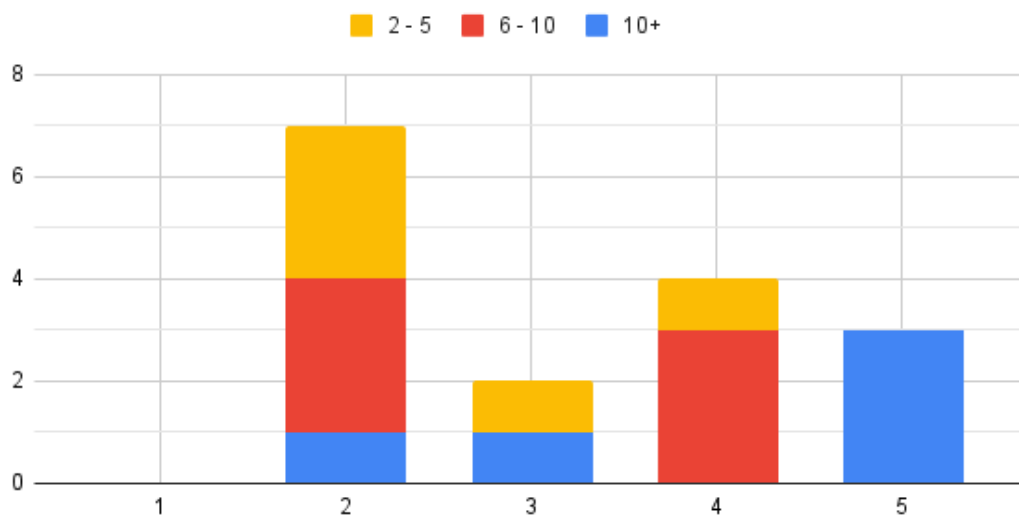
L'automazione aiuta a ridurre i tempi di sviluppo e rilascio in base alla posizione geografica dell'azienda

Frequenza su l'AI cambierà il modo in cui verranno testati i videogiochi



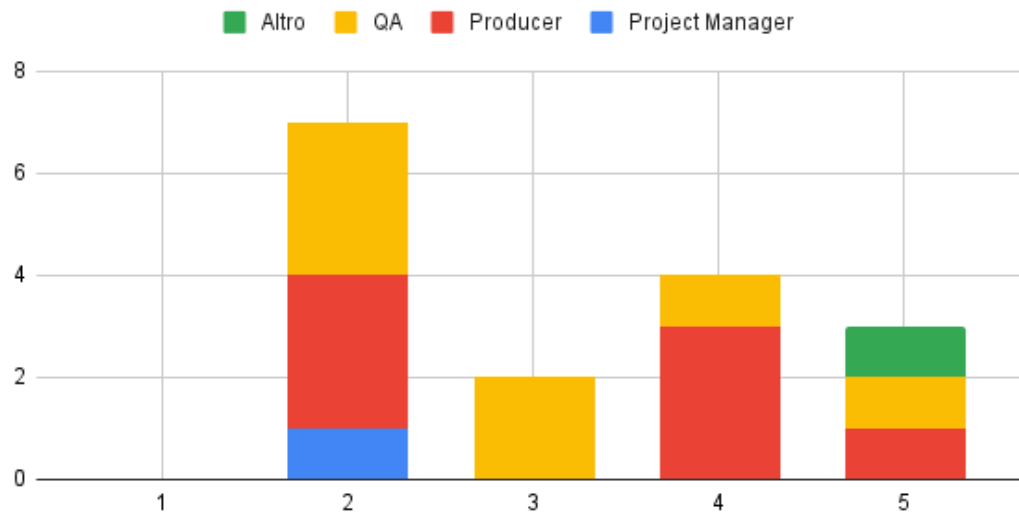
Frequenza su l'AI cambierà il modo in cui verranno testati i videogiochi

L'AI cambierà il modo in cui verranno testati i videogiochi in base agli anni di esperienza



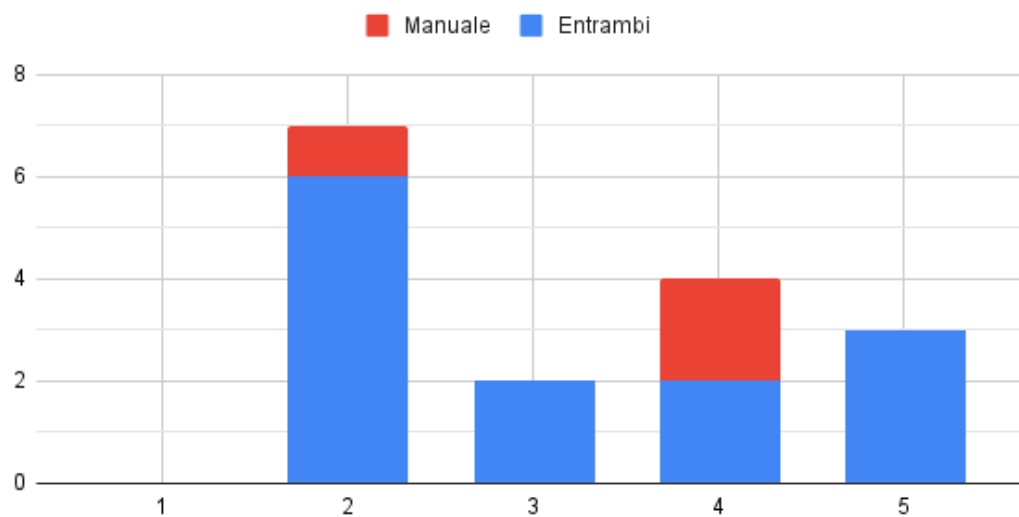
L'AI cambierà il modo in cui verranno testati i videogiochi in base agli anni di esperienza

L'AI cambierà il modo in cui verranno testati i videogiochi in base al Ruolo



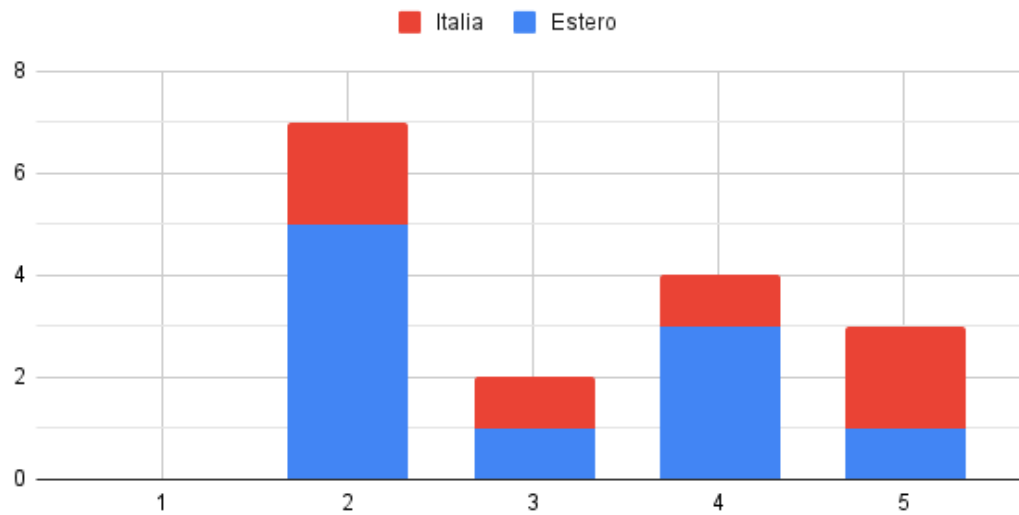
L'AI cambierà il modo in cui verranno testati i videogiochi in base al Ruolo

L'AI cambierà il modo in cui verranno testati i videogiochi in base al tipo di test



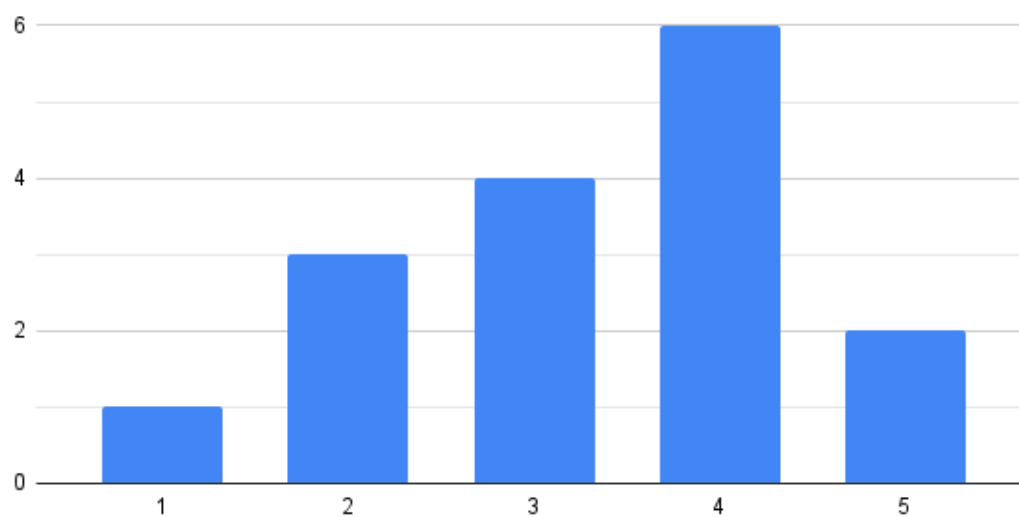
L'AI cambierà il modo in cui verranno testati i videogiochi in base al tipo di test

L'AI cambierà il modo in cui verranno testati i videogiochi in base alla posizione geografica dell'azienda



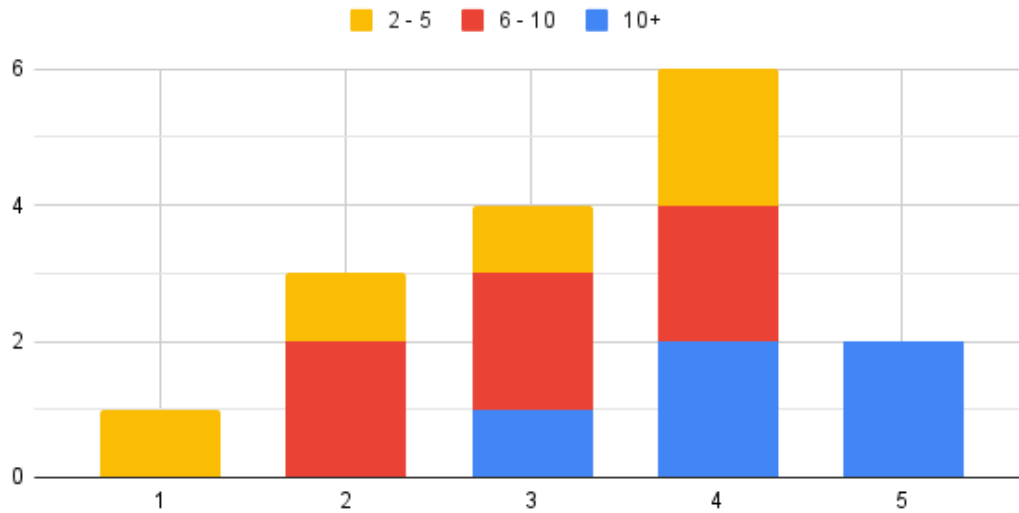
L'AI cambierà il modo in cui verranno testati i videogiochi in base alla posizione geografica dell'azienda

Frequenza su quanto l'AI può migliorare l'efficacia del testing automatizzato



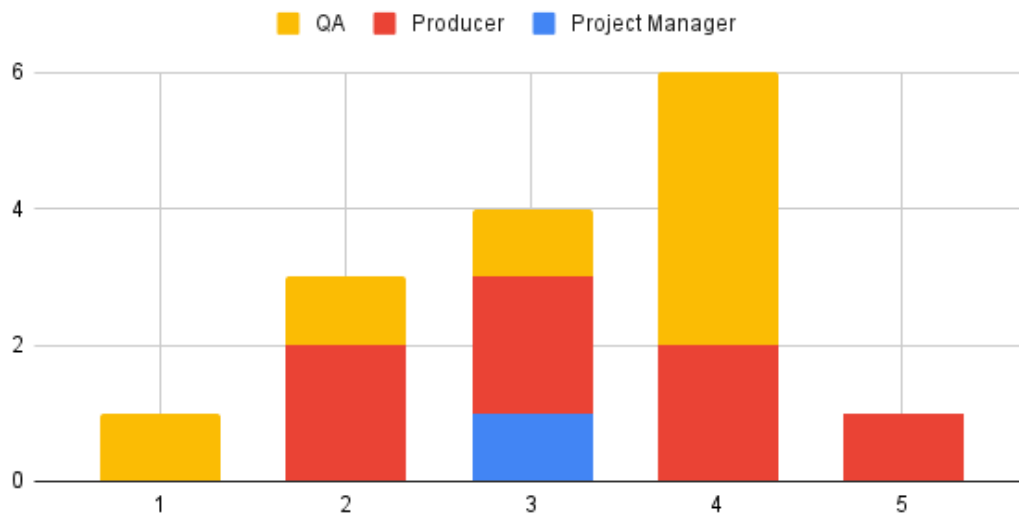
Frequenza su quanto l'AI può migliorare l'efficacia del testing automatizzato

Quanto l'AI può migliorare l'efficacia del testing automatizzato in base agli anni di esperienza



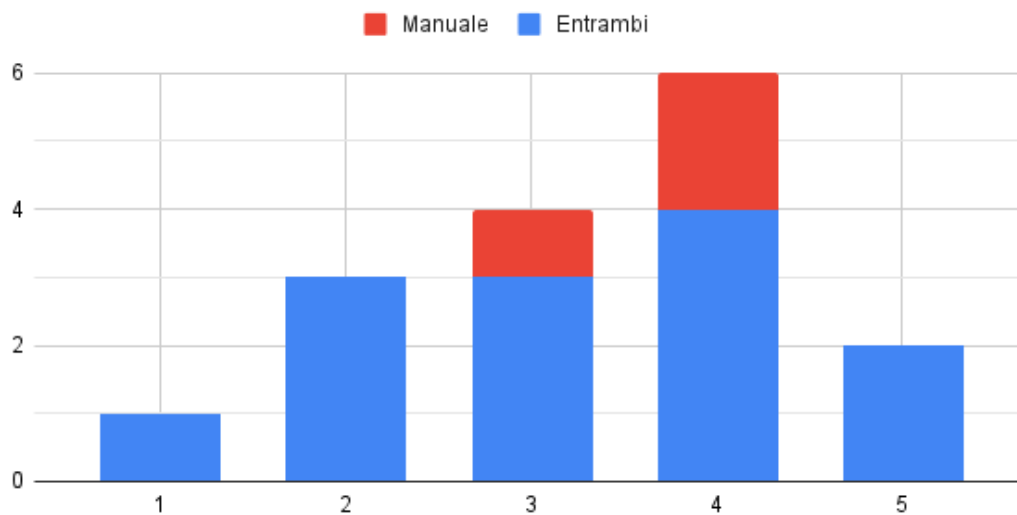
Quanto l'AI potrebbe migliorare l'efficacia del testing automatizzato in base agli anni di esperienza

Quanto l'AI può migliorare l'efficacia del testing automatizzato in base al Ruolo



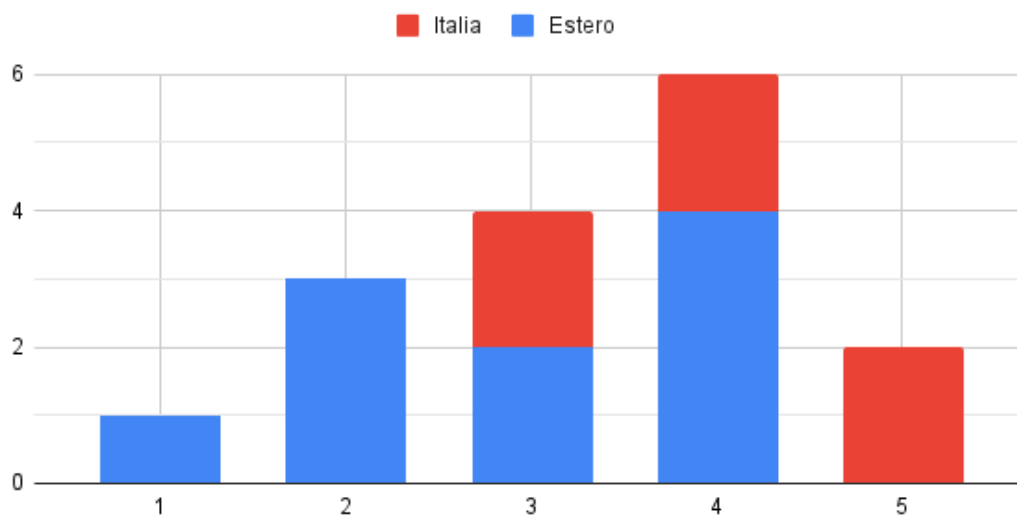
Quanto l'AI potrebbe migliorare l'efficacia del testing automatizzato in base al Ruolo

Quanto l'AI può migliorare l'efficacia del testing automatizzato in base al tipo di test



Quanto l'AI potrebbe migliorare l'efficacia del testing automatizzato in base al tipo di test

Quanto l'AI può migliorare l'efficacia del testing automatizzato in base alla posizione geografica dell'azienda



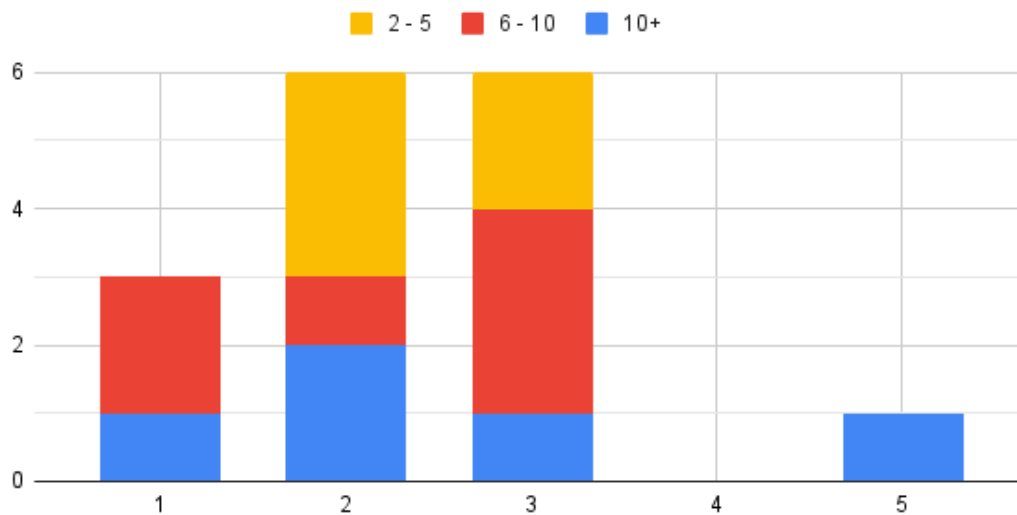
Quanto l'AI potrebbe migliorare l'efficacia del testing automatizzato in base alla posizione geografica dell'azienda

L'AI può migliorare la collaborazione tra il team di QA e gli sviluppatori in base al Ruolo



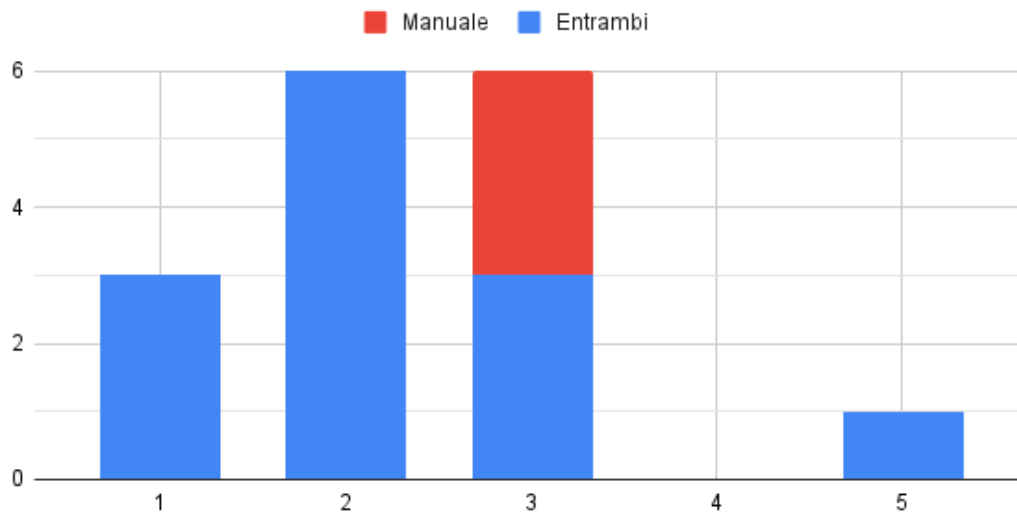
L'AI potrebbe migliorare la collaborazione tra il team di QA e gli sviluppatori in base al Ruolo

L'AI può migliorare la collaborazione tra il team di QA e gli sviluppatori in base agli anni di esperienza



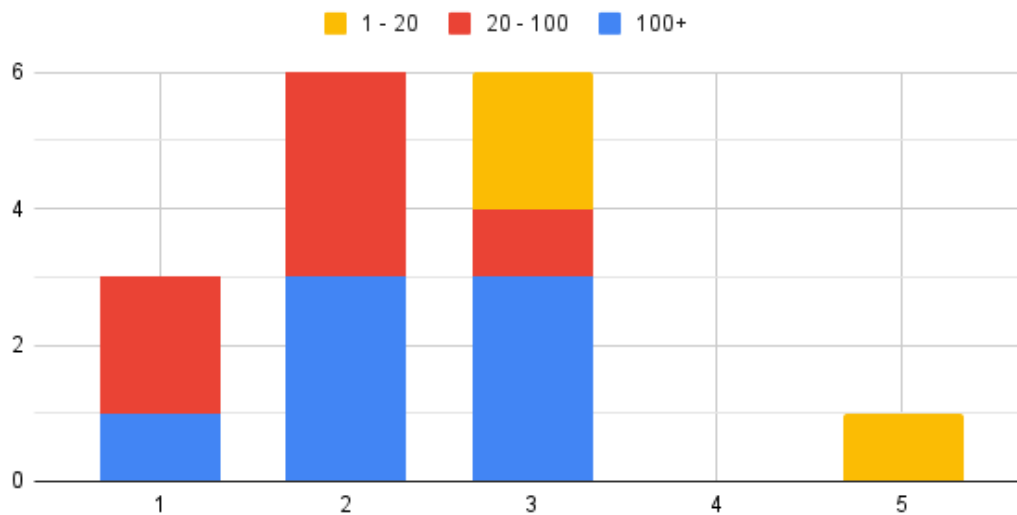
L'AI potrebbe migliorare la collaborazione tra il team di QA e gli sviluppatori in base agli anni di esperienza

L'AI può migliorare la collaborazione tra il team di QA e gli sviluppatori in base al tipo di test



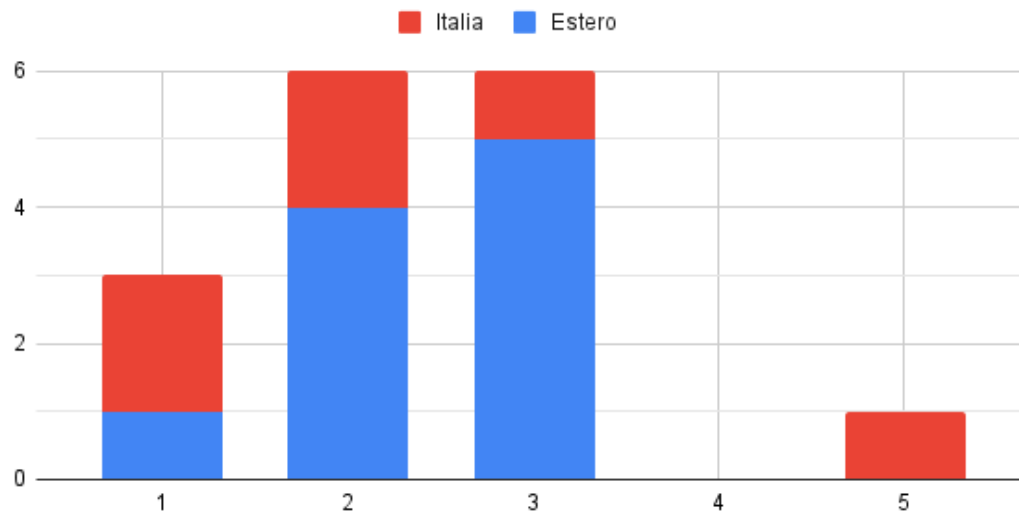
L'AI potrebbe migliorare la collaborazione tra il team di QA e gli sviluppatori in base al tipo di test

L'AI può migliorare la collaborazione tra il team di QA e gli sviluppatori in base alla dimensione dell'azienda



L'AI potrebbe migliorare la collaborazione tra il team di QA e gli sviluppatori in base alla dimensione dell'azienda

L'AI può migliorare la collaborazione tra il team di QA e gli sviluppatori in base alla posizione geografica dell'azienda



L'AI potrebbe migliorare la collaborazione tra il team di QA e gli sviluppatori in base alla posizione geografica dell'azienda