**EXPERIMENTATION IN SOFTWARE ENGINEERING**

Il primo capitolo del testo introduce l'ingegneria del software e la sua rilevanza, spiegando che questa disciplina si è sviluppata per risolvere le difficoltà incontrate durante lo sviluppo di software complessi. Vediamo in dettaglio i concetti chiave discussi, come i metodi di sviluppo, i modelli di processo e le metodologie di ricerca.

**1. Metodi di sviluppo del software**

L'ingegneria del software si avvale di diversi metodi per sviluppare prodotti software. I metodi principali includono:

* **Metodo Scientifico**: Osserva il mondo reale e costruisce modelli basati su queste osservazioni. Ad esempio, in ambito software, questo potrebbe includere la simulazione di un sistema per valutare la sua performance.
* **Metodo Ingegneristico**: Studia le soluzioni esistenti, propone cambiamenti, e poi valuta questi cambiamenti. Questo è tipicamente utilizzato nell'industria per migliorare processi e prodotti.
* **Metodo Empirico**: Propone un modello e lo valuta attraverso studi empirici, come studi di caso o esperimenti. Questo metodo è particolarmente utile nell'ingegneria del software poiché il comportamento umano ha un ruolo significativo.
* **Metodo Analitico**: Propone una teoria formale e la confronta con osservazioni empiriche. Questo metodo è più comune nelle aree formali dell'ingegneria e della scienza dei computer, come la teoria degli algoritmi.

**2. Modelli di processo del software**

Un modello di processo software descrive i passaggi e le attività necessarie per sviluppare un prodotto software. Alcuni dei modelli principali menzionati includono:

* **Modello a Cascata**: Un approccio lineare dove lo sviluppo procede in fasi successive, come requisiti, progettazione, implementazione, verifica e manutenzione. Ogni fase deve essere completata prima di passare alla successiva.
* **Sviluppo Incrementale**: Lo sviluppo avviene attraverso incrementi successivi, dove ogni incremento aggiunge funzionalità al sistema. Questo permette di rilasciare versioni funzionanti del software in modo iterativo.
* **Sviluppo Evolutivo**: Simile allo sviluppo incrementale, ma con un maggiore focus sull'adattamento del software in base al feedback degli utenti e ai cambiamenti nei requisiti.
* **Modello a Spirale**: Combina elementi del modello a cascata e dello sviluppo incrementale, enfatizzando la gestione del rischio. Ogni ciclo nella spirale rappresenta una fase dello sviluppo che include pianificazione, analisi del rischio, ingegneria e valutazione.
* **Metodi Agili**: Metodologie flessibili che enfatizzano lo sviluppo iterativo, il feedback continuo e la collaborazione tra team. I metodi agili, come Scrum e Extreme Programming (XP), tendono a ridurre la documentazione e a concentrarsi su un codice funzionante che viene sviluppato e migliorato continuamente.

**3. Miglioramento dei processi**

Le aziende sono continuamente impegnate nel miglioramento dei loro processi software per rimanere competitive. Il miglioramento del processo è necessario per produrre software di alta qualità a costi inferiori e con tempi di consegna ridotti. Alcuni modelli e paradigmi per il miglioramento dei processi includono:

* **Quality Improvement Paradigm (QIP)**: Un modello strutturato per il miglioramento continuo del processo software, definito da Basili. Include diversi passaggi per supportare un approccio sistematico e disciplinato.
* **Ciclo Plan/Do/Study/Act (PDSA)**: Un ciclo iterativo utilizzato per il miglioramento continuo dei processi. Include fasi di pianificazione, esecuzione, studio dei risultati e azione per implementare miglioramenti.
* **Capability Maturity Model (CMM)**: Un modello di maturità che valuta il livello di maturità dei processi software di un'organizzazione. CMM ha cinque livelli di maturità, ciascuno con aree chiave di processo che devono essere migliorate per passare al livello successivo.

**4. Metodologie di ricerca in ingegneria del software**

Il testo sottolinea l'importanza di adottare un approccio scientifico all'ingegneria del software, utilizzando metodi di ricerca validati per prendere decisioni informate e migliorare i processi di sviluppo. Le principali strategie di ricerca empirica includono:

* **Esperimenti Formali**: Esperimenti controllati che cercano di stabilire relazioni causali tra variabili. Questi esperimenti sono progettati per essere sistematici e ripetibili.
* **Studi di Caso**: Osservazione e analisi di progetti reali in contesti industriali per comprendere come i processi software funzionano in pratica. Gli studi di caso sono utili per studiare fenomeni complessi in ambienti reali.
* **Sondaggi e Interviste**: Raccolta di dati attraverso questionari e interviste per ottenere informazioni su pratiche, opinioni ed esperienze nel campo del software.

**5. Importanza degli studi empirici**

### Gli studi empirici sono cruciali per valutare l'efficacia delle metodologie di sviluppo, per validare nuove tecniche e strumenti, e per comprendere meglio le dinamiche che influenzano la produzione di software. Essi permettono di identificare relazioni tra variabili e di testare ipotesi che possono portare a miglioramenti significativi nei processi software. Il testo discute anche la difficoltà di condurre esperimenti "veri" con piena randomizzazione nel contesto del software, sottolineando l'importanza dei quasi-esperimenti, che possono comunque fornire risultati preziosi. ESEMPI 1. ****Metodi di sviluppo del software****

#### **Metodo Scientifico**

**Esempio**: Supponiamo di voler valutare l'efficienza di un nuovo algoritmo di compressione dei dati. Nel metodo scientifico, potremmo osservare diversi scenari di utilizzo di questo algoritmo, costruire un modello di simulazione del processo di compressione e misurare le prestazioni (ad esempio, la velocità e il tasso di compressione) in vari ambienti. Questo ci permette di capire meglio l'efficacia dell'algoritmo in condizioni diverse prima di implementarlo su larga scala.

#### **Metodo Ingegneristico**

**Esempio**: Una società sta utilizzando un sistema di gestione delle versioni del software, ma trova che ci sono frequenti conflitti di codice tra i team. Con il metodo ingegneristico, l'azienda potrebbe studiare l'attuale processo di gestione delle versioni, proporre modifiche come l'adozione di un nuovo flusso di lavoro o l'uso di strumenti di integrazione continua, e poi valutare l'efficacia di questi cambiamenti in termini di riduzione dei conflitti e miglioramento della collaborazione.

#### **Metodo Empirico**

**Esempio**: Immagina che un team di sviluppo voglia sapere se l'adozione della programmazione in coppia (pair programming) migliora la qualità del codice. Potrebbero organizzare un esperimento empirico in cui un gruppo di programmatori lavora in coppia mentre un altro gruppo lavora individualmente. Misurando il numero di difetti nel codice prodotto da entrambi i gruppi, il team potrebbe determinare se la programmazione in coppia porta a un codice di qualità superiore.

#### **Metodo Analitico**

**Esempio**: Consideriamo la progettazione di un algoritmo di ordinamento. Nel metodo analitico, potremmo sviluppare una teoria formale sull'efficienza dell'algoritmo (ad esempio, il numero di confronti necessari per ordinare una lista di dimensione n) e poi confrontare questa teoria con i risultati empirici ottenuti testando l'algoritmo su dati reali.

### 2. ****Modelli di processo del software****

#### **Modello a Cascata**

**Esempio**: Un team di sviluppo software che crea un'applicazione bancaria potrebbe utilizzare il modello a cascata. Prima definiscono tutti i requisiti (ad esempio, funzioni di login, trasferimento di fondi), poi progettano l'architettura del sistema, scrivono il codice, lo testano e infine rilasciano il prodotto finale. Il modello a cascata funziona bene in contesti in cui i requisiti sono stabili e ben compresi fin dall'inizio.

#### **Sviluppo Incrementale**

**Esempio**: Un team di sviluppo che lavora su un'applicazione di social media potrebbe adottare uno sviluppo incrementale, rilasciando prima una versione base con funzioni essenziali come la registrazione e la creazione di profili. Successivamente, potrebbero aggiungere nuove funzionalità come la messaggistica, le notifiche e la condivisione di foto in iterazioni successive.

#### **Sviluppo Evolutivo**

**Esempio**: Un team che lavora su un software di gestione delle risorse aziendali potrebbe iniziare con un prototipo che gestisce solo l'inventario. Basandosi sul feedback degli utenti, potrebbero evolvere il software aggiungendo funzionalità per la gestione delle risorse umane e delle finanze, adattandosi continuamente alle esigenze dell'azienda.

#### **Modello a Spirale**

**Esempio**: Una società che sviluppa un sistema di controllo per aerei potrebbe utilizzare il modello a spirale per gestire i rischi associati al progetto. In ogni ciclo della spirale, il team potrebbe identificare e analizzare i rischi (ad esempio, il rischio che un modulo software non soddisfi i requisiti di sicurezza), sviluppare una parte del sistema, testarla e quindi valutare i risultati prima di procedere al ciclo successivo.

#### **Metodi Agili**

**Esempio**: Un team che sviluppa un'app mobile per la gestione delle attività potrebbe adottare un metodo agile come Scrum. Ogni settimana (sprint), il team lavora su un insieme di funzionalità prioritarie, rilasciando una versione funzionante dell'app alla fine di ogni sprint. Il feedback degli utenti viene integrato rapidamente nel ciclo di sviluppo, consentendo al team di adattarsi rapidamente alle nuove esigenze o problemi.

### 3. ****Miglioramento dei processi****

#### **Quality Improvement Paradigm (QIP)**

**Esempio**: Un'azienda di sviluppo software si accorge che i test finali rivelano troppi difetti. Usando il QIP, il team potrebbe identificare questa area come critica, analizzare il processo di sviluppo e test attuale, e proporre miglioramenti come l'integrazione di test unitari automatici nelle prime fasi del ciclo di sviluppo.

#### **Ciclo Plan/Do/Study/Act (PDSA)**

**Esempio**: Immagina un team che desidera migliorare la velocità con cui risolve i bug. Nel ciclo PDSA, il team pianifica di adottare uno strumento di tracciamento dei bug più efficace (Plan), lo implementa (Do), studia i risultati per vedere se la risoluzione dei bug è migliorata (Study), e quindi decide se continuare con lo strumento o apportare ulteriori modifiche (Act).

#### **Capability Maturity Model (CMM)**

**Esempio**: Un'azienda di sviluppo software si trova al livello 2 di maturità del CMM, il che significa che ha processi software gestiti ma non standardizzati. Per avanzare al livello 3, l'azienda potrebbe concentrarsi sull'istituzione di processi standardizzati e documentati per tutte le attività di sviluppo, assicurando che le best practice vengano applicate uniformemente in tutti i progetti.

### 4. ****Metodologie di ricerca in ingegneria del software****

#### **Esperimenti Formali**

**Esempio**: Un ricercatore vuole testare l'efficacia di un nuovo strumento di refactoring del codice. In un esperimento formale, potrebbe far usare lo strumento a un gruppo di sviluppatori mentre un altro gruppo non lo utilizza. Il ricercatore misurerebbe poi la qualità del codice prodotto da entrambi i gruppi per vedere se c'è una differenza significativa.

#### **Studi di Caso**

**Esempio**: Un ricercatore potrebbe condurre uno studio di caso su un'azienda che ha implementato una nuova metodologia di sviluppo software. Lo studio osserva come la metodologia viene adottata, le sfide incontrate, e l'impatto sulle tempistiche di progetto e sulla qualità del software.

#### **Sondaggi e Interviste**

**Esempio**: Un accademico potrebbe condurre un sondaggio tra sviluppatori di software per raccogliere dati sulle pratiche di test utilizzate nel settore. Le interviste con sviluppatori senior potrebbero fornire approfondimenti su come e perché certe pratiche vengono preferite rispetto ad altre.

### 5. ****Importanza degli studi empirici****

Gli studi empirici sono fondamentali per rispondere a domande pratiche nel campo del software. Ad esempio:

* **Esempio**: Un'azienda potrebbe chiedersi se l'adozione della metodologia di sviluppo agile abbia effettivamente ridotto i tempi di rilascio rispetto al precedente approccio a cascata. Conducendo uno studio empirico, l'azienda potrebbe confrontare dati storici (tempi di consegna, numero di bug, soddisfazione dei clienti) prima e dopo l'adozione del metodo agile, fornendo una base oggettiva per valutare l'efficacia del cambiamento.

**Conclusione**

Il capitolo stabilisce le basi per comprendere l'ingegneria del software come una disciplina che richiede un approccio scientifico e disciplinato. Mette in luce la complessità dei processi software e la necessità di utilizzare studi empirici per migliorare continuamente la pratica dello sviluppo software.

**ESERCIZI**

**1.1. Perché gli esperimenti possono essere considerati come prototipi per i cambiamenti di processo?**

Gli esperimenti possono essere visti come prototipi per i cambiamenti di processo perché permettono di testare nuove idee o modifiche in un ambiente controllato prima di implementarle a livello completo. Proprio come un prototipo di prodotto consente di valutare la fattibilità e l'efficacia di un nuovo design, un esperimento su un processo permette di osservare gli effetti di un cambiamento senza compromettere l'intero sistema. Questo approccio riduce il rischio associato all'implementazione di nuove pratiche, fornendo dati empirici che possono guidare decisioni informate su quali cambiamenti adottare.

**1.2. Come possono essere utilizzati gli esperimenti nelle attività di miglioramento?**

Gli esperimenti possono essere utilizzati nelle attività di miglioramento per valutare l'efficacia di nuove tecniche, processi o strumenti prima di una loro adozione su larga scala. Ad esempio, un'azienda potrebbe condurre un esperimento per testare una nuova metodologia di test software su un progetto pilota. I risultati dell'esperimento (come la riduzione dei difetti o l'aumento della velocità di sviluppo) possono fornire evidenze concrete sull'efficacia della nuova metodologia, guidando così il processo decisionale per la sua implementazione in tutta l'organizzazione.

**1.3. Perché gli studi empirici sono importanti nell'ingegneria del software?**

Gli studi empirici sono importanti nell'ingegneria del software perché permettono di raccogliere dati oggettivi e misurabili sulle pratiche, metodologie e strumenti utilizzati nel settore. Poiché lo sviluppo software è fortemente influenzato da fattori umani e da variabilità contestuali, gli studi empirici offrono un mezzo per comprendere meglio queste dinamiche. Essi forniscono evidenze reali che possono essere utilizzate per migliorare i processi, validare teorie e supportare l'adozione di nuove tecniche, contribuendo così a una pratica più informata e basata su dati concreti.

**1.4. Quando è meglio utilizzare il metodo di ricerca empirico nell'ingegneria del software rispetto ai metodi scientifico, ingegneristico e analitico?**

Il metodo di ricerca empirico è particolarmente adatto nell'ingegneria del software quando si vuole comprendere e valutare il comportamento umano e le dinamiche organizzative, che non possono essere facilmente modellate o simulate. È preferibile utilizzarlo rispetto al metodo scientifico quando non è possibile fare affidamento su modelli puramente teorici o simulazioni; rispetto al metodo ingegneristico quando si vuole valutare empiricamente l'impatto delle modifiche in contesti reali; e rispetto al metodo analitico quando le variabili umane e comportamentali giocano un ruolo cruciale, rendendo difficile l'applicazione di teorie formali.

**1.5. In quali tre strategie sono divisi i metodi empirici?**

I metodi empirici nell'ingegneria del software sono generalmente suddivisi in tre strategie principali:

1. **Esperimenti**: Esperimenti formali e controllati che cercano di stabilire relazioni causali tra variabili.
2. **Studi di Caso**: Osservazione e analisi approfondita di un singolo progetto o organizzazione per comprendere come i processi funzionano in contesti reali.
3. **Sondaggi e Interviste**: Raccolta di dati attraverso questionari e interviste per ottenere informazioni sulle pratiche, opinioni ed esperienze di un ampio numero di partecipanti.

### CAPITOLO 2 – STRATEGIE EMPIRICHE Paradigmi di Ricerca: Qualitativa vs. Quantitativa

Esistono due paradigmi principali nella ricerca empirica, ciascuno con approcci distinti:

1. **Ricerca Esplorativa (Qualitativa):**
   * **Obiettivo:** Esplorare fenomeni in contesti naturali, lasciando che i risultati emergano dalle osservazioni.
   * **Caratteristiche:** Richiede un disegno di ricerca flessibile, capace di adattarsi ai cambiamenti durante lo studio.
   * **Metodo:** Basata su dati qualitativi, la ricerca qualitativa è spesso utilizzata per comprendere le cause e le spiegazioni che i soggetti dello studio attribuiscono ai fenomeni osservati.
   * **Approccio Induttivo:** Cerca di interpretare il fenomeno attraverso le spiegazioni offerte dai partecipanti, focalizzandosi sulla loro percezione del problema.
   * **Esempio:** Immagina di voler capire perché gli studenti di una scuola preferiscono un particolare metodo di insegnamento rispetto ad altri. Un ricercatore qualitativo potrebbe condurre interviste approfondite con gli studenti per esplorare le loro opinioni e percezioni. Attraverso queste interviste, potrebbero emergere temi comuni, come la sensazione di maggiore coinvolgimento o la chiarezza delle spiegazioni offerte da quel metodo specifico.
   * **Applicazioni:** Ideale per studiare credenze, comprensioni e prospettive multiple, ed è utile per capire perché si ottengono certi risultati in studi quantitativi.
2. **Ricerca Esplicativa (Quantitativa):**
   * **Obiettivo:** Quantificare le relazioni o confrontare gruppi diversi per identificare relazioni causa-effetto.
   * **Caratteristiche:** Richiede un disegno di ricerca fisso, in cui le variabili e i fattori sono stabiliti prima dell'inizio dello studio.
   * **Metodo:** Basata su dati quantitativi, la ricerca quantitativa è adatta per testare l'effetto di manipolazioni o interventi.
   * **Vantaggi:** Permette confronti chiari e analisi statistiche rigorose. È tipicamente utilizzata in esperimenti controllati.
   * **Esempio:** Supponiamo che un ricercatore voglia verificare se un nuovo metodo di insegnamento aumenta i punteggi degli esami degli studenti. In questo caso, potrebbe dividere gli studenti in due gruppi: uno che utilizza il nuovo metodo e uno che segue il metodo tradizionale. Dopo un periodo definito, il ricercatore confronterà i punteggi degli esami dei due gruppi per vedere se c'è una differenza significativa. Questo è un classico esempio di ricerca quantitativa, in cui si cerca di stabilire una relazione causa-effetto.
   * **Applicazioni:** Adatta per studi che cercano di determinare quanto un intervento o un trattamento influisca su un determinato risultato.

**Complementarietà tra Approcci:**

* **Integrare Qualitativo e Quantitativo:** I due approcci non sono in competizione, ma sono complementari. Ad esempio, un esperimento quantitativo potrebbe quantificare l'efficacia di un metodo, mentre uno studio qualitativo potrebbe esplorare le ragioni dietro eventuali differenze tra i gruppi studiati.
  + **Esempio Integrato:** Immagina di aver scoperto attraverso un esperimento quantitativo che un nuovo programma di allenamento per atleti migliora significativamente le loro prestazioni. Tuttavia, non è chiaro perché questo programma sia così efficace. Un successivo studio qualitativo, che coinvolge interviste con gli atleti, potrebbe rivelare che il programma li motiva di più grazie alla varietà degli esercizi o alla maggiore personalizzazione rispetto ad altri programmi.

**Obiettivi del Capitolo:**

Il testo si propone di:

1. **Introduzione alle Strategie Empiriche:** Fornire una panoramica delle diverse strategie di ricerca empirica.
2. **Aspetti Importanti:** Sottolineare aspetti chiave relativi alle strategie empiriche, come la replica degli esperimenti e l’aggregazione degli studi.
3. **Applicazioni Pratiche:** Dimostrare come queste strategie possono essere applicate nel contesto del trasferimento tecnologico e dei programmi di miglioramento.

**Dettagli Specifici:**

* **Strategie Empiriche:** Le sezioni 2.1-2.5 offrono un'analisi delle diverse strategie empiriche, come sondaggi, studi di caso ed esperimenti, e ne confrontano l’efficacia in diversi contesti.
  + **Esempio di Studio di Caso:** Un'azienda potrebbe utilizzare uno studio di caso per capire come una particolare tecnologia è stata implementata con successo in un'altra azienda simile. Attraverso interviste e osservazioni, il ricercatore raccoglie dati qualitativi che forniscono una comprensione approfondita del processo.
* **Replicazione e Teoria:** Vengono discussi l'importanza della replicazione degli esperimenti (per verificare la validità dei risultati) e il ruolo delle teorie nella guida degli studi empirici.
  + **Esempio di Replicazione:** Se un esperimento ha dimostrato che una nuova strategia di marketing aumenta le vendite del 20%, un altro ricercatore potrebbe replicare l'esperimento in un contesto diverso per vedere se i risultati sono coerenti. La replica è cruciale per confermare l'affidabilità dei risultati iniziali.
* **Aggregazione degli Studi:** L'importanza di aggregare i risultati di diversi studi per ottenere una visione complessiva più solida.
  + **Esempio di Meta-Analisi:** Un ricercatore potrebbe aggregare i risultati di vari studi che esaminano l'efficacia di una particolare terapia psicologica, ottenendo così una panoramica più robusta dell'efficacia complessiva di quella terapia.

**Applicazioni nella Tecnologia e nel Miglioramento:**

Il capitolo chiude con una discussione su come queste strategie di ricerca possono essere integrate nei processi di trasferimento tecnologico e nei programmi di miglioramento, sottolineando l'importanza di un approccio basato su prove empiriche per guidare decisioni e innovazioni.

* **Esempio Applicativo:** Supponiamo che un'azienda tecnologica stia cercando di migliorare il proprio processo di sviluppo software. Potrebbe utilizzare la ricerca quantitativa per misurare l'impatto di nuove pratiche di codifica sui tempi di rilascio del software e, allo stesso tempo, condurre studi qualitativi per comprendere come i programmatori percepiscono e adottano queste nuove pratiche. Questi risultati combinati potrebbero guidare l'azienda a implementare miglioramenti che non solo riducono i tempi, ma che sono anche ben accettati dal team.

### ****Panoramica delle Strategie Empiriche****

Nel campo della ricerca empirica, specialmente nell'ingegneria del software, esistono tre principali strategie di indagine che possono essere adottate in base all'obiettivo della valutazione e alle condizioni dell'indagine stessa:

1. **Survey (Sondaggio)**
2. **Case Study (Studio di Caso)**
3. **Experiment (Esperimento)**

Ognuna di queste strategie ha caratteristiche distintive, vantaggi e applicazioni specifiche. Di seguito, esploreremo ciascuna di esse in dettaglio.

#### **1. Survey (Sondaggio)**

* **Definizione:** Un sondaggio è un sistema per raccogliere informazioni da o su persone allo scopo di descrivere, confrontare o spiegare la loro conoscenza, atteggiamenti e comportamenti.
* **Caratteristiche Principali:**
  + **Retrospettività:** Spesso eseguiti in modo retrospettivo, ad esempio, dopo che un particolare strumento o tecnica è stato utilizzato per un certo periodo.
  + **Metodi di Raccolta Dati:** Utilizza principalmente interviste o questionari per raccogliere dati qualitativi o quantitativi.
  + **Campionamento:** Si basa sulla selezione di un campione rappresentativo della popolazione di interesse.
  + **Analisi:** I risultati vengono analizzati per derivare conclusioni descrittive ed esplicative, che vengono poi generalizzate all'intera popolazione.
* **Esempio:** Immagina di voler capire la soddisfazione degli utenti riguardo a una nuova applicazione mobile. Potresti inviare un questionario a un campione di utenti chiedendo loro di valutare vari aspetti dell'app, come l'usabilità, le funzionalità e la stabilità. Le risposte raccolte possono poi essere analizzate per determinare aree di forza e debolezza dell'applicazione.

#### **2. Case Study (Studio di Caso)**

* **Definizione:** Uno studio di caso nell'ingegneria del software è un'indagine empirica che si basa su molteplici fonti di evidenza per investigare una singola istanza [singolo caso di studio] (o un piccolo numero di istanze [gruppo di casi di studio]) di un fenomeno contemporaneo nell'ingegneria del software all'interno del suo contesto reale, specialmente quando il confine tra fenomeno e contesto non può essere chiaramente definito.
* **Caratteristiche Principali:**
  + **Focalizzazione su Progetti Specifici:** Viene utilizzato per ricercare progetti, attività o incarichi specifici.
  + **Raccolta Dati Continua:** I dati vengono raccolti per uno scopo specifico durante tutto il periodo dello studio.
  + **Analisi Statistica:** Basandosi sulla raccolta dati, è possibile effettuare analisi statistiche, come la regressione lineare o l'analisi delle componenti principali.
  + **Livello di Controllo:** Ha un livello di controllo inferiore rispetto a un esperimento; è principalmente un metodo osservazionale.
* **Esempio:** Supponiamo che un'azienda abbia implementato una nuova metodologia di sviluppo agile. Uno studio di caso potrebbe coinvolgere l'osservazione del team durante diversi sprint, interviste con i membri del team e l'analisi delle metriche di performance (come la velocità o il numero di bug risolti) per capire come la nuova metodologia influenzi il processo di sviluppo.

#### **3. Experiment (Esperimento)**

* **Definizione:** Un esperimento (o esperimento controllato) nell'ingegneria del software è un'indagine empirica che manipola un fattore o variabile dell'ambiente studiato. Basandosi sulla randomizzazione, diversi trattamenti vengono applicati a o da diversi soggetti, mantenendo costanti le altre variabili, e misurando gli effetti sulle variabili di risultato. Negli esperimenti orientati all'uomo, gli individui applicano diversi trattamenti agli oggetti, mentre negli esperimenti orientati alla tecnologia, diversi trattamenti tecnici vengono applicati a diversi oggetti.
* **Caratteristiche Principali:**
  + **Ambiente di Laboratorio:** Generalmente condotti in un ambiente controllato che permette un alto livello di controllo sulle variabili.
  + **Randomizzazione:** I soggetti vengono assegnati a diversi trattamenti in modo casuale.
  + **Manipolazione delle Variabili:** Una o più variabili vengono manipolate mentre le altre vengono mantenute costanti.
  + **Misurazione e Analisi:** Gli effetti della manipolazione vengono misurati e analizzati statisticamente per determinare l'efficacia dei trattamenti.
* **Quasi-Esperimenti:** In situazioni in cui non è possibile assegnare trattamenti ai soggetti in modo casuale (ad esempio, a causa di limitazioni etiche o pratiche), si utilizzano i quasi-esperimenti, dove l'assegnazione dei trattamenti emerge dalle caratteristiche intrinseche dei soggetti o degli oggetti.
* **Esempio:** Un ricercatore vuole testare l'efficacia di un nuovo algoritmo di compressione dei dati. Potrebbe progettare un esperimento in cui diversi file di dimensioni e contenuti simili vengono compressi utilizzando sia l'algoritmo tradizionale che quello nuovo. Misurando il tempo di compressione e la dimensione del file compresso, il ricercatore può determinare se il nuovo algoritmo offre vantaggi significativi.

### ****Confronto tra le Strategie****

| **Strategia** | **Tipo di Disegno** | **Dati Raccolti** |
| --- | --- | --- |
| Survey | Fisso | Qualitativi e/o Quantitativi |
| Case Study | Flessibile | Qualitativi e/o Quantitativi |
| Experiment | Fisso | Principalmente Quantitativi |

* **Disegno Fisso vs. Flessibile:**
  + **Disegno Fisso:** Implica che la struttura dello studio è predefinita e non cambia durante l'indagine. Tipico dei sondaggi e degli esperimenti.
  + **Disegno Flessibile:** Permette adattamenti e modifiche durante lo studio in base alle osservazioni e ai risultati emergenti. Caratteristico degli studi di caso.
* **Tipo di Dati:**
  + **Qualitativi:** Dati descrittivi che forniscono approfondimenti sulle percezioni, opinioni e motivazioni dei partecipanti.
  + **Quantitativi:** Dati numerici che possono essere misurati e analizzati statisticamente.

### ****Applicazioni e Considerazioni Pratiche****

* **Sondaggi:** Utilizzati ampiamente nelle scienze sociali per raccogliere dati sulle opinioni e i comportamenti di una popolazione. Ad esempio, prevedere l'esito di un'elezione attraverso sondaggi d'opinione.
* **Studi di Caso:** Ideali per approfondire la comprensione di fenomeni complessi in contesti reali. Ad esempio, analizzare come una specifica azienda ha implementato con successo una nuova tecnologia.
* **Esperimenti:** Forniscono il massimo controllo sulle variabili e permettono di stabilire relazioni causa-effetto. Ad esempio, testare se una nuova metodologia di sviluppo software riduce effettivamente il numero di bug rispetto alla metodologia tradizionale.

### ****Integrando le Strategie****

Spesso, le strategie possono essere combinate per ottenere una comprensione più completa di un fenomeno. Ad esempio:

* **Prima un Sondaggio, poi uno Studio di Caso:** Un sondaggio può identificare aziende che hanno implementato con successo una certa tecnologia. Successivamente, uno studio di caso su una di queste aziende può fornire approfondimenti dettagliati sul processo di implementazione.
* **Esperimento seguito da Sondaggio:** Dopo aver condotto un esperimento per testare una nuova interfaccia utente, un sondaggio può essere utilizzato per raccogliere feedback dagli utenti sulle loro esperienze soggettive.

### ****2.3 Case Studies****

Uno **studio di caso** è una strategia di ricerca empirica che mira a investigare un singolo fenomeno o entità nel suo contesto reale, entro uno specifico periodo di tempo. Questo tipo di studio è particolarmente utile quando il fenomeno studiato è difficile da separare chiaramente dal suo ambiente circostante. Il ricercatore raccoglie informazioni dettagliate su un singolo progetto o evento durante un periodo prolungato, utilizzando diverse procedure di raccolta dati e prospettive di analisi.

**Esempio:** Supponiamo che un'azienda voglia valutare l'efficacia di un nuovo strumento di progettazione software. Un ricercatore potrebbe condurre uno studio di caso su un progetto specifico in cui lo strumento viene utilizzato, osservando e raccogliendo dati durante tutto il ciclo di vita del progetto per valutare l'impatto dello strumento sulle prestazioni del team e sulla qualità del prodotto finale.

#### **Caratteristiche degli Studi di Caso**

* **Flessibilità:** A differenza degli esperimenti, che sono studi controllati con variabili manipolate intenzionalmente, gli studi di caso si concentrano sull'osservazione delle variabili in una situazione tipica, senza la necessità di isolare o manipolare fattori specifici.
* **Realismo:** Gli studi di caso sono spesso più realistici e facili da pianificare rispetto agli esperimenti, poiché si svolgono in contesti naturali. Tuttavia, questo approccio presenta la difficoltà di generalizzare i risultati, poiché le conclusioni sono legate al contesto specifico studiato.

**Esempio:** Se un'azienda introduce un nuovo processo di sviluppo software, potrebbe essere difficile vedere subito i risultati di questo cambiamento. Un caso di studio potrebbe richiedere l'osservazione dell'intero ciclo di vita del progetto e l'analisi dei risultati solo dopo la consegna del prodotto finale per valutare l'impatto del cambiamento.

#### **Tipologie di Studi di Caso**

Gli studi di caso possono essere utilizzati per confrontare diverse metodologie o approcci. Esistono diverse modalità per organizzare uno studio di caso, secondo Kitchenham et al. [97]:

1. **Confronto con una Base di Riferimento Aziendale:** I risultati ottenuti con un nuovo metodo possono essere confrontati con dati storici raccolti da progetti standard dell'azienda. Ad esempio, si può confrontare la produttività o il tasso di difetti prima e dopo l'introduzione di una nuova metodologia.
2. **Confronto con un Progetto Simile (Sister Project):** Un progetto parallelo che utilizza il metodo tradizionale può essere confrontato con quello che utilizza il nuovo metodo. Entrambi i progetti dovrebbero avere caratteristiche simili per rendere il confronto valido.
3. **Applicazione a Componenti Specifiche:** Il nuovo metodo può essere applicato casualmente a specifici componenti di un prodotto, mentre altri componenti seguono il metodo tradizionale. Questo approccio si avvicina a un esperimento, ma non è un vero esperimento poiché i progetti non sono selezionati casualmente.

#### **Fattori Confondenti e Altri Aspetti**

Negli studi di caso, è fondamentale ridurre al minimo gli effetti dei fattori confondenti, ovvero quei fattori che rendono difficile distinguere gli effetti di due variabili. Ad esempio, se si osserva un miglioramento delle prestazioni con un nuovo strumento, potrebbe essere difficile determinare se il miglioramento sia dovuto allo strumento stesso o all'esperienza dell'utente.

**Esempio:** Se uno studio di caso mostra che un nuovo strumento aumenta la produttività, potrebbe essere complicato stabilire se l'aumento sia dovuto all'efficacia dello strumento o all'abilità e all'esperienza degli sviluppatori che lo utilizzano.

#### **Vantaggi e Svantaggi degli Studi di Caso**

* **Vantaggi:**
  + Possono catturare aspetti che un esperimento non può visualizzare, come la complessità, l'imprevedibilità e la dinamica di un progetto reale.
  + Sono più semplici da pianificare e implementare rispetto agli esperimenti, e spesso offrono un'immagine più realistica del contesto di studio.
* **Svantaggi:**
  + È difficile generalizzare i risultati, poiché sono specifici al contesto studiato.
  + I ricercatori non hanno il controllo completo della situazione, il che significa che i cambiamenti imprevedibili possono complicare l'interpretazione dei risultati.

**Esempio di Problema:** In un piccolo studio di caso, il problema principale potrebbe essere legato alla tecnica di sviluppo software in esame. Tuttavia, in uno studio di caso più grande, il problema potrebbe derivare dalla gestione di un team di persone più numeroso, con maggiori difficoltà di comunicazione e coordinazione.

### ****2.4 Experiments****

Gli **esperimenti** sono condotti quando si desidera avere un controllo preciso sulla situazione e manipolare direttamente e sistematicamente il comportamento delle variabili in studio. Gli esperimenti tipicamente includono più trattamenti per confrontare i risultati ottenuti.

**Esempio:** Supponiamo di voler confrontare due metodi di ispezione del codice. Se è possibile controllare chi utilizza un metodo e chi utilizza l'altro, nonché dove e quando vengono applicati, allora è possibile condurre un esperimento. Questo tipo di manipolazione può avvenire in un ambiente offline, come un laboratorio, o online, in un contesto reale sul campo.

#### **Tipologie di Esperimenti**

* **Esperimenti Umani vs. Esperimenti Tecnologici:**
  + **Esperimenti Umani:** Gli esseri umani applicano diversi trattamenti agli oggetti, come due metodi di ispezione applicati a due frammenti di codice. Tuttavia, questo tipo di esperimento ha un minor controllo rispetto a quelli tecnologici, poiché il comportamento umano può variare.
  + **Esperimenti Tecnologici:** Strumenti o tecnologie diverse vengono applicati agli oggetti, come due strumenti di generazione di casi di test applicati allo stesso programma. Gli strumenti sono generalmente più prevedibili e meno influenzati da bias, rendendo questi esperimenti più controllati.

**Esempio:** In un esperimento umano, due gruppi di sviluppatori potrebbero essere incaricati di testare lo stesso software utilizzando due strumenti diversi. Nel caso di un esperimento tecnologico, si potrebbe applicare lo stesso test di generazione di casi su due sistemi diversi per confrontarne l'efficacia.

### ****Confronto tra Studi di Caso ed Esperimenti****

La principale differenza tra **studi di caso** ed **esperimenti** risiede nel **livello di controllo** che il ricercatore ha sul contesto e sulle variabili in gioco.

#### **Studi di Caso**

* **Contesto Reale:** Gli studi di caso vengono condotti in contesti reali, in cui il fenomeno o l'entità studiata è osservata nel suo ambiente naturale. Qui, il ricercatore non cerca di manipolare o controllare in modo rigoroso le variabili del contesto. Invece, si concentra sull'osservazione di come il fenomeno si manifesta nel contesto specifico, spesso senza la possibilità di isolare completamente le variabili che potrebbero influenzare i risultati.
  + **Esempio:** Se un'azienda introduce un nuovo metodo di gestione del progetto, uno studio di caso potrebbe seguire un singolo progetto dall'inizio alla fine per osservare come il nuovo metodo influisce sulle dinamiche del team, sulla produttività e sulla qualità del prodotto. Tuttavia, in questo contesto reale, molti altri fattori (come l'esperienza dei membri del team o la complessità del progetto) potrebbero influenzare i risultati, e il ricercatore ha un controllo limitato su questi fattori.

#### **Esperimenti**

* **Contesto Controllato:** Gli esperimenti, d'altra parte, vengono progettati per avere un controllo molto più rigoroso sulle variabili di interesse. Il ricercatore può manipolare una o più variabili indipendenti e osservare gli effetti di queste manipolazioni su una o più variabili dipendenti, mentre tutte le altre variabili rimangono costanti. Questo controllo preciso permette di isolare gli effetti della variabile manipolata, rendendo più facile stabilire relazioni di causa-effetto.
  + **Esempio:** In un esperimento, un ricercatore potrebbe voler confrontare l'efficacia di due tecniche di test software. Per fare questo, potrebbe selezionare due gruppi di sviluppatori con livelli di esperienza simili e assegnare loro lo stesso compito, ma utilizzando tecniche di test diverse. Il ricercatore può controllare variabili come il tempo a disposizione, il tipo di software da testare e l'ambiente di lavoro per assicurarsi che l'unica differenza significativa tra i gruppi sia la tecnica di test utilizzata. In questo modo, qualsiasi differenza nei risultati può essere attribuita direttamente alla tecnica di test piuttosto che ad altri fattori.

#### **Riassunto della Differenza**

* **Studi di Caso:** Offrono una visione dettagliata di un fenomeno in un contesto reale, ma con limitato controllo sulle variabili, il che rende più difficile isolare e identificare le cause precise di ciò che si osserva. Sono ideali per capire come un fenomeno si manifesta in situazioni complesse e dinamiche.
* **Esperimenti:** Consentono un controllo rigoroso sulle variabili, permettendo di manipolare specifici fattori e osservare gli effetti di queste manipolazioni. Sono ideali per testare ipotesi specifiche e stabilire relazioni di causa-effetto in un ambiente controllato.

In sintesi, mentre gli studi di caso sono più adatti a esplorare fenomeni complessi in ambienti naturali, gli esperimenti sono più efficaci per testare ipotesi in condizioni controllate, minimizzando l'influenza di fattori esterni.

#### **Caratteristiche degli Esperimenti**

Gli esperimenti sono appropriati per investigare diversi aspetti, tra cui:

* **Conferma di Teorie:** Testare teorie esistenti per verificarne la validità.
* **Conferma del Senso Comune:** Verificare se le concezioni comuni delle persone sono accurate.
* **Esplorazione di Relazioni:** Testare se una certa relazione tra variabili è valida.
* **Valutazione dell'Accuratezza di Modelli:** Verificare se l'accuratezza di certi modelli è come previsto.
* **Validazione di Misure:** Assicurarsi che una misura rifletta correttamente ciò che dovrebbe misurare.

**Esempio:** Un esperimento potrebbe essere condotto per verificare se un nuovo metodo di sviluppo riduce effettivamente i difetti nel software rispetto al metodo tradizionale, confermando o confutando una teoria esistente.

#### **Processo dell'Esperimento**

Condurre un esperimento coinvolge diverse fasi:

1. **Scoping:** Definire l'ambito dell'esperimento, identificando obiettivi e variabili.
2. **Pianificazione:** Dettagliare come l'esperimento sarà condotto, compresa la selezione dei soggetti e la definizione dei trattamenti.
3. **Operazione:** Eseguire l'esperimento, applicando i trattamenti e raccogliendo dati.
4. **Analisi e Interpretazione:** Analizzare i dati raccolti per interpretare i risultati.
5. **Presentazione e Packaging:** Presentare i risultati in modo comprensibile e utilizzabile per la comunità scientifica o industriale.

### ****Esempio Completo del Processo di un Esperimento****

Immaginiamo che un'azienda di sviluppo software voglia condurre un esperimento per valutare l'efficacia di un nuovo strumento di refactoring automatico del codice rispetto al metodo manuale tradizionale. L'obiettivo è capire se l'uso dello strumento migliora la qualità del codice e riduce i tempi di sviluppo.

#### **1. Scoping (Definizione dell'Ambito)**

* **Obiettivi:**
  + Valutare l'efficacia del nuovo strumento di refactoring.
  + Confrontare i risultati in termini di qualità del codice e tempo impiegato rispetto al metodo manuale.
* **Variabili:**
  + **Variabile Indipendente:** Metodo di refactoring (nuovo strumento vs. metodo manuale).
  + **Variabili Dipendenti:** Qualità del codice (misurata tramite il numero di bug trovati nei test successivi) e tempo di sviluppo (misurato in ore impiegate).
* **Ambito:** L'esperimento sarà limitato a un modulo specifico di un'applicazione esistente, in cui il refactoring è necessario.

#### **2. Pianificazione**

* **Selezione dei Soggetti:**
  + Due gruppi di sviluppatori con esperienza simile verranno selezionati. Il primo gruppo utilizzerà il nuovo strumento di refactoring, mentre il secondo gruppo continuerà con il metodo manuale.
* **Definizione dei Trattamenti:**
  + **Gruppo A:** Utilizzerà il nuovo strumento di refactoring automatico.
  + **Gruppo B:** Utilizzerà il metodo manuale tradizionale per eseguire il refactoring.
* **Assegnazione dei Soggetti:** I soggetti verranno assegnati ai gruppi in modo casuale per evitare bias.
* **Controllo delle Variabili:** Assicurarsi che tutti i soggetti utilizzino lo stesso ambiente di sviluppo e che lavorino sullo stesso modulo del codice.

#### **3. Operazione**

* **Esecuzione dell'Esperimento:**
  + Entrambi i gruppi lavoreranno sul refactoring del modulo assegnato. Il gruppo A utilizzerà il nuovo strumento, mentre il gruppo B utilizzerà il metodo manuale.
* **Raccolta dei Dati:**
  + Registrare il tempo impiegato da ciascun gruppo per completare il refactoring.
  + Eseguire test automatici sul codice refactorato per contare il numero di bug trovati.

#### **4. Analisi e Interpretazione**

* **Analisi dei Dati:**
  + Confrontare il tempo medio di completamento del refactoring tra i due gruppi.
  + Confrontare il numero medio di bug trovati nel codice refactorato.
  + Utilizzare test statistici (come un test t di Student) per determinare se le differenze osservate tra i due gruppi sono statisticamente significative.
* **Interpretazione dei Risultati:**
  + Se il gruppo A (strumento automatico) impiega significativamente meno tempo e produce meno bug rispetto al gruppo B (metodo manuale), si può concludere che il nuovo strumento è più efficace.

#### **5. Presentazione e Packaging**

* **Presentazione dei Risultati:**
  + Preparare un report dettagliato che descriva l'obiettivo dell'esperimento, la metodologia utilizzata, i risultati ottenuti e le conclusioni tratte.
  + Includere grafici e tabelle che illustrano i confronti tra i gruppi.
* **Condivisione dei Risultati:**
  + Presentare i risultati al management dell'azienda per supportare la decisione sull'adozione del nuovo strumento.
  + Pubblicare i risultati in una conferenza di ingegneria del software o in un journal per condividere i risultati con la comunità scientifica e industriale.

### ****2.5 Confronto tra Strategie Empiriche****

La scelta della strategia di ricerca empirica dipende da vari prerequisiti dell'indagine, come lo scopo, le risorse disponibili e il tipo di dati che si desidera raccogliere. Di seguito viene fornito un confronto tra le principali strategie empiriche — sondaggi, studi di caso ed esperimenti — basato su diversi fattori chiave.

#### **1. Controllo sull'Esecuzione**

* **Survey:** Il controllo sull'esecuzione è limitato. Nei sondaggi, i dati vengono raccolti tramite questionari o interviste, ma il ricercatore non ha il controllo sull'ambiente in cui si svolge la raccolta dei dati. Ad esempio, se un partecipante decide di non rispondere o di interrompere la partecipazione, il ricercatore non può intervenire.
* **Case Study:** Anche negli studi di caso, il ricercatore ha un controllo limitato. Poiché i dati vengono raccolti in un contesto reale durante l'esecuzione di un progetto, eventi esterni come la sospensione del progetto da parte della direzione possono interrompere lo studio.
* **Experiment:** Gli esperimenti offrono il massimo controllo sull'esecuzione. Il ricercatore può manipolare le variabili indipendenti, decidere come e quando raccogliere i dati e controllare l'ambiente in cui l'esperimento viene condotto.

#### **2. Controllo delle Misurazioni**

* **Survey:** Il controllo sulle misurazioni è basso. Nei sondaggi, il ricercatore può solo raccogliere opinioni o percezioni dei partecipanti e non può includere misurazioni dirette di variabili come la volatilità dei requisiti.
* **Case Study:** Negli studi di caso, il ricercatore ha un maggiore controllo sulle misurazioni, potendo scegliere quali dati raccogliere e come monitorare le variabili nel contesto reale.
* **Experiment:** Negli esperimenti, il ricercatore ha pieno controllo sulle misurazioni. Può decidere esattamente quali variabili misurare e come farlo durante l'esperimento.

#### **3. Costo dell'Indagine**

* **Survey:** I sondaggi hanno il costo più basso tra le strategie. Richiedono meno risorse, poiché spesso consistono nell'invio di questionari o interviste, senza la necessità di un ambiente controllato o di attrezzature specializzate.
* **Case Study:** Gli studi di caso hanno un costo medio. Richiedono più tempo e risorse rispetto ai sondaggi, poiché il ricercatore deve raccogliere dati dettagliati su un progetto o fenomeno specifico nel contesto reale.
* **Experiment:** Gli esperimenti sono i più costosi, poiché richiedono la creazione di un ambiente controllato, l'assegnazione di soggetti e la gestione di variabili multiple.

#### **4. Facilità di Replicazione**

* **Survey:** I sondaggi sono relativamente facili da replicare. Poiché coinvolgono principalmente la raccolta di dati attraverso questionari, possono essere ripetuti facilmente con diversi campioni della popolazione.
* **Case Study:** Gli studi di caso sono difficili da replicare. Poiché sono strettamente legati a un contesto specifico, replicare uno studio di caso in un altro contesto potrebbe non produrre gli stessi risultati.
* **Experiment:** Gli esperimenti, se ben progettati, sono relativamente facili da replicare. Poiché il ricercatore ha un alto grado di controllo sulle variabili, è possibile ripetere l'esperimento in condizioni simili per verificare i risultati.

#### **Studi Longitudinali vs. Replicazioni**

* **Studi Longitudinali:** Uno studio longitudinale è un'indagine condotta sullo stesso gruppo di soggetti nel tempo. Questo tipo di studio è utile per osservare cambiamenti o sviluppi nel corso del tempo. Ad esempio, un sondaggio può essere ripetuto ogni anno per monitorare i cambiamenti nelle opinioni degli sviluppatori su una nuova tecnologia.
* **Replicazioni:** La replicazione di uno studio implica la ripetizione dell'indagine con nuovi soggetti o in un contesto diverso per verificare la validità dei risultati originali. Una replicazione è considerata "vera" se riesce a replicare sia il design che i risultati dello studio originale.

### ****Conclusione****

La scelta della strategia empirica dipende dalle specifiche esigenze dell'indagine. Se è necessario un controllo rigoroso e la possibilità di manipolare le variabili, un esperimento è la scelta migliore. Se invece si desidera esplorare un fenomeno in un contesto reale, uno studio di caso potrebbe essere più appropriato. Per raccogliere opinioni o dati descrittivi da un ampio campione, un sondaggio è la soluzione più economica ed efficiente.

Il confine tra le diverse strategie non è sempre netto; ad esempio, uno studio di caso comparativo potrebbe essere considerato un quasi-esperimento in un contesto industriale. La scelta finale dovrebbe essere guidata dagli obiettivi dell'indagine, dalle risorse disponibili e dal tipo di dati richiesti.

### ****2.6 Replicazioni****

La **replicazione** di un esperimento consiste nel ripetere un'indagine sotto condizioni simili, spesso variando la popolazione dei soggetti. Questo processo aiuta a determinare quanto si può confidare nei risultati dell'esperimento originale. L'idea è che, se i soggetti sono rappresentativi di una popolazione più ampia e l'ipotesi di randomizzazione è corretta, le repliche all'interno di questa popolazione dovrebbero produrre gli stessi risultati del primo esperimento. Tuttavia, se i risultati non coincidono, significa che non tutti i fattori che influenzano l'esito sono stati adeguatamente considerati nel design originale dell'esperimento.

Anche se è possibile misurare una certa variabile o replicare un esperimento, a volte ciò può essere difficile o troppo costoso. Esistono diversi tipi di repliche:

#### **Tipi di Replicazioni**

1. **Repliche Ristrette (Close Replications):**
   * **Definizione:** Queste repliche seguono le procedure originali il più fedelmente possibile. Spesso sono chiamate anche repliche esatte.
   * **Vantaggio:** Mantenendo sotto controllo i fattori noti, si rafforza la fiducia nei risultati ottenuti.
   * **Svantaggio:** Le repliche ristrette a volte richiedono che gli stessi ricercatori conducano lo studio, poiché possiedono conoscenze implicite sulle procedure dell'esperimento che difficilmente possono essere documentate. Questo può portare a un rischio significativo di **bias dell'esperimentatore**, ovvero l'influenza inconscia del ricercatore sui risultati.
2. **Repliche Differenziate (Differentiated Replications):**
   * **Definizione:** Queste repliche studiano le stesse domande di ricerca, ma utilizzando procedure sperimentali diverse o variando deliberatamente una o più condizioni principali dell'esperimento.
   * **Vantaggio:** Permettono studi più esplorativi e, se le differenze nei fattori e nelle impostazioni sono ben documentate e analizzate, possono fornire una comprensione più approfondita dei fenomeni studiati.
   * **Fattori da Considerare:** Quando si conducono repliche differenziate, è importante documentare e analizzare attentamente i seguenti aspetti:
     + **Sito:** Dove viene condotto l'esperimento.
     + **Sperimentatori:** Chi conduce l'esperimento.
     + **Design dell'Esperimento:** Come è stato progettato l'esperimento.
     + **Strumenti:** Forme e materiali utilizzati.
     + **Variabili Misurate:** Quali variabili sono state raccolte.
     + **Soggetti:** Chi partecipa all'esperimento.

#### **Classificazioni Avanzate di Replicazioni (Secondo Basili et al.)**

Basili e colleghi hanno proposto una classificazione più dettagliata delle repliche, che include:

* **Repliche Ristrette:** Sinonimo di repliche esatte o close replications.
* **Repliche che Variano le Variabili Intrinseche allo Studio:** Ad esempio, cambiando la popolazione dei soggetti ma mantenendo costante il design dell'esperimento.
* **Repliche che Variano le Variabili Intrinseche al Focus dello Studio:** Ad esempio, cambiando l'obiettivo principale dell'indagine.
* **Repliche che Variano le Variabili di Contesto:** Cambiando l'ambiente in cui viene valutata la soluzione.
* **Repliche che Variano le Modalità di Esecuzione dell'Esperimento:** Modificando come l'esperimento viene condotto, pur mantenendo lo stesso obiettivo.
* **Repliche che Estendono la Teoria:** Utilizzando l'esperimento per ampliare o modificare le teorie esistenti.

#### **Considerazioni sulla Replicazione**

* **Repliche Ristrette:** Mentre offrono il vantaggio di controllare i fattori noti, spesso non riescono a catturare tutta la complessità di un esperimento di ingegneria del software, dove molti fattori possono variare.
* **Repliche Differenziate:** Possono essere più utili per scoprire nuove informazioni, soprattutto se le differenze nei fattori sono attentamente documentate e analizzate. Queste repliche possono contribuire a verificare l'ipotesi originale in contesti diversi, piuttosto che limitarsi a replicare esattamente il design sperimentale originale.

Alcuni studiosi sostengono che, nell'ingegneria del software, le repliche differenziate siano preferibili alle repliche ristrette, poiché forniscono maggiori informazioni sulla validità delle ipotesi in diverse condizioni e contesti.

In conclusione, la replicazione è un processo cruciale per verificare la solidità dei risultati ottenuti in un esperimento, e la scelta tra una replica ristretta o differenziata dipende dall'obiettivo specifico della ricerca e dalla natura dell'esperimento originale.

### ****2.7 Teoria nell'Ingegneria del Software****

Le teorie sono fondamentali per la comprensione e l'interpretazione dei fenomeni in qualsiasi disciplina scientifica. Esse offrono spiegazioni basate su concetti di base e meccanismi sottostanti, fornendo un'importante controparte alla conoscenza di tendenze temporanee e manifestazioni superficiali. Nell'ingegneria del software, le teorie possono essere utilizzate per generare, confermare e ampliare la nostra comprensione dei processi e delle pratiche, ma, come osservato da Hannay et al. in una revisione sistematica della letteratura sugli esperimenti di ingegneria del software (1993-2002), l'uso delle teorie in questo campo è ancora scarso.

#### **Teorie e "Leggi" in Ingegneria del Software**

Endres e Rombach hanno identificato una lista di 50 risultati che definiscono come "leggi", una nozione mutuata dalle scienze naturali che descrive fenomeni ripetibili. Molte di queste "leggi" sono generali e non specifiche per l'ingegneria del software, come ad esempio "ci vogliono 5.000 ore per trasformare un principiante in un esperto". Nella loro struttura, le teorie spiegano queste "leggi", le ipotesi propongono spiegazioni temporanee sul perché un fenomeno si comporta in un certo modo, e le congetture sono supposizioni sul fenomeno stesso.

Zendler, invece, propone una "teoria preliminare dell'ingegneria del software", composta da tre ipotesi fondamentali, sei ipotesi centrali e quattro ipotesi elementari, organizzate in una gerarchia che va dal più astratto al più concreto, tutte derivanti da risultati di studi sperimentali.

#### **Tipi di Teorie (Secondo Gregor)**

Gregor descrive cinque tipi generali di teorie, che possono essere adattate al contesto dell'ingegneria del software:

1. **Analisi:** Descrive l'oggetto di studio, includendo tassonomie, classificazioni e ontologie.
2. **Spiegazione:** Offre una spiegazione su perché accade qualcosa.
3. **Predizione:** Mira a prevedere cosa accadrà, spesso utilizzando modelli matematici o probabilistici.
4. **Spiegazione e Predizione:** Combina spiegazione e predizione, tipicamente definita come una "teoria empiricamente basata".
5. **Progettazione e Azione:** Descrive come fare qualcosa, spesso sotto forma di scienza del design (es. metodologia di sviluppo).

C'è un dibattito sul fatto che questa ultima categoria possa essere considerata una teoria vera e propria.

#### **Quadro per le Teorie nell'Ingegneria del Software (Secondo Sjøberg et al.)**

Sjøberg et al. propongono un quadro teorico per l'ingegneria del software che serve a strutturare e organizzare le teorie in questo campo, suddividendo le componenti principali della teoria in quattro parti fondamentali. Ecco una spiegazione più dettagliata di ciascuna di queste parti:

**1. Costrutti**

* **Cosa sono:** I costrutti rappresentano le entità fondamentali che la teoria cerca di descrivere, spiegare o predire. Sono i concetti chiave su cui la teoria si basa.
* **Esempio:** In una teoria sullo sviluppo del software, i costrutti potrebbero includere elementi come "team di sviluppo", "tecnologie di programmazione", "metodologie di gestione del progetto", e "qualità del software". Questi sono i "mattoni" con cui la teoria costruisce le sue spiegazioni.

**2. Proposizioni**

* **Cosa sono:** Le proposizioni sono le relazioni che vengono ipotizzate tra i costrutti. Esse descrivono come i diversi costrutti sono collegati tra loro e quali effetti hanno l'uno sull'altro.
* **Esempio:** Una proposizione potrebbe affermare che "l'uso di metodologie agili (costrutto) migliora la qualità del software (costrutto) riducendo il numero di bug rilevati dopo il rilascio". Qui, la proposizione stabilisce una relazione tra l'uso di una metodologia specifica e l'outcome sulla qualità del software.

**3. Spiegazioni**

* **Cosa sono:** Le spiegazioni forniscono il ragionamento che sta alla base delle proposizioni. Derivano da osservazioni empiriche o da un ragionamento logico che collega i costrutti tra loro. Essenzialmente, le spiegazioni cercano di rispondere alla domanda "perché?" dietro le relazioni proposte.
* **Esempio:** Riprendendo la proposizione precedente, la spiegazione potrebbe essere che "le metodologie agili promuovono una maggiore comunicazione e iterazioni frequenti, che permettono di identificare e risolvere i problemi prima che diventino bug nel prodotto finale". Questo spiega perché l'uso delle metodologie agili potrebbe migliorare la qualità del software.

**4. Ambito (Scope)**

* **Cosa è:** L'ambito definisce i limiti entro i quali la teoria è applicabile. Specifica le circostanze, i contesti e le condizioni in cui la teoria è considerata valida.
* **Archetipi:** Sjøberg et al. propongono che l'ambito possa essere espresso in termini di quattro archetipi:
  + **Attore:** L'entità che partecipa o è coinvolta, come un individuo, un team, un progetto, un'organizzazione o un'intera industria.
  + **Tecnologia:** Gli strumenti, le tecniche, i metodi o i linguaggi utilizzati, come un modello di processo o uno strumento di sviluppo.
  + **Attività:** Le azioni eseguite, come pianificare, creare, modificare o analizzare un sistema software.
  + **Sistema Software:** Il sistema stesso, che può essere classificato secondo dimensioni come la dimensione, la complessità, il dominio applicativo (es. amministrativo, embedded, tempo reale) ecc.
* **Esempio:** Se una teoria afferma che "l'uso di metodologie agili migliora la qualità del software", l'ambito potrebbe specificare che questo è vero per "progetti software di piccole e medie dimensioni in ambienti di sviluppo collaborativi", indicando chiaramente dove e quando la teoria è applicabile.

### ****Archetipi per l'Ambito delle Teorie****

| **Classe Archetipica** | **Sottoclassi** |
| --- | --- |
| Attore | Individuo, team, progetto, organizzazione o industria |
| Tecnologia | Modello di processo, metodo, tecnica, strumento o linguaggio |
| Attività | Pianificare, creare, modificare o analizzare (un sistema software) |
| Sistema Software | Classificabile lungo varie dimensioni, come dimensione, complessità, dominio applicativo, ecc. |

### ****Importanza delle Teorie nell'Ingegneria del Software****

Le teorie sono essenziali per concettualizzare e comunicare la conoscenza all'interno di un campo di ricerca. Sono utili per aggregare la ricerca esistente, impostare studi di replicazione e comunicare con i praticanti durante il processo decisionale, che si tratti di scelte strategiche di tecnologia o decisioni di progetto basate su modelli predittivi.

### ****Sfide e Prospettive Future****

Nonostante la loro importanza teorica, i sistemi di teoria proposti finora hanno avuto un impatto limitato nel campo dell'ingegneria del software. Tuttavia, lo sviluppo di teorie solide è fondamentale per far evolvere l'ingegneria del software in un campo scientifico maturo. Per questo motivo, è essenziale continuare a sviluppare e affinare le teorie nel contesto dell'ingegneria del software, per migliorare la nostra comprensione e la nostra capacità di comunicare e applicare le conoscenze acquisite.

### ****2.8 Aggregazione delle Evidenze dagli Studi Empirici****

Man mano che aumenta il numero di studi empirici, emerge la necessità di aggregare le evidenze provenienti da molteplici indagini, come gli studi di replicazione. Questo processo è essenziale per due motivi principali:

1. **Costruire su Conoscenze Esistenti:** La ricerca dovrebbe sempre partire dalle conoscenze già acquisite, integrando e ampliando ciò che è stato precedentemente scoperto.
2. **Rispondere a Domande Complesse:** Spesso, più studi empirici insieme possono fornire risposte a domande che non sono state sufficientemente affrontate da studi singoli. La raccolta e la sintesi delle evidenze empiriche devono comunque rispettare gli standard scientifici.

#### **Revisione Sistematiche della Letteratura**

Le **revisioni sistematiche della letteratura** sono uno strumento utilizzato per raccogliere e sintetizzare le evidenze empiriche provenienti da diverse fonti. Kitchenham e Charters definiscono queste revisioni come "uno studio secondario che utilizza una metodologia ben definita per identificare, analizzare e interpretare tutte le evidenze disponibili relative a una specifica domanda di ricerca in modo imparziale e (in una certa misura) ripetibile" . Gli studi empirici individuati sono chiamati **studi primari**, mentre la revisione sistematica della letteratura è definita **studio secondario**.

Una revisione sistematica della letteratura inizia con una specifica domanda di ricerca, simile a quella di un singolo studio empirico. Ad esempio, una domanda potrebbe essere: "La tecnologia/metodo A è migliore di B?" .

#### **Ricerca e Raccolta degli Studi**

La ricerca degli studi empirici viene effettuata attraverso query di database, nonché consultando riviste, atti di conferenze e letteratura grigia (ad esempio, rapporti tecnici) basata su parole chiave specifiche. Una tecnica comunemente usata è lo "snowballing", che consiste nel seguire i riferimenti di un documento per trovare altri articoli rilevanti. Lo snowballing può essere:

* **Backward Snowballing:** Seguendo la lista dei riferimenti di un documento.
* **Forward Snowballing:** Cercando articoli che citano il documento rilevante.

#### **Studi di Mappatura**

Se la domanda di ricerca è più generale o il campo è meno esplorato, si può condurre uno **studio di mappatura** (o **scoping study**). Gli studi di mappatura hanno domande di ricerca più ampie e mirano a identificare lo stato della pratica o della ricerca su un determinato argomento, spesso individuando le tendenze di ricerca . A causa del loro scopo più ampio, le procedure di ricerca e classificazione sono meno rigorose e hanno caratteristiche più qualitative rispetto alle revisioni sistematiche.

#### **Sintesi e Aggregazione delle Evidenze**

Una volta raccolto un insieme di studi empirici su un argomento, si passa alla **sintesi o aggregazione** delle evidenze. Le sintesi basate su metodi statistici sono chiamate **meta-analisi**. Esempi di meta-analisi nell'ingegneria del software includono studi sui metodi di rilevamento dei difetti, sulle metodologie agili e sulla programmazione in coppia.

Se le procedure di meta-analisi non sono applicabili, si utilizza una **sintesi descrittiva**, che può includere visualizzazioni, tabulazioni e statistiche descrittive dei dati raccolti . Più la domanda di ricerca è ampia, più è probabile che vengano utilizzati metodi qualitativi per la sintesi. Cruzes e Dybå presentano una panoramica dei metodi di sintesi qualitativa .

#### **Crescita dell'Interesse per le Revisioni Sistematiche**

L'interesse per le revisioni sistematiche della letteratura nell'ingegneria del software è cresciuto notevolmente durante il primo decennio del XXI secolo. Kitchenham et al. riportano che sono state pubblicate 53 revisioni sistematiche uniche tra il 2004 e il 2008 . Oltre alla sintesi dei risultati empirici, queste revisioni hanno anche portato all'identificazione di proposte di miglioramento per la segnalazione degli studi empirici e per i database in cui sono conservati.

Le revisioni sistematiche della letteratura sono ulteriormente approfondite nel Capitolo 4.

**2.9 L'Empirismo nel Contesto dell'Ingegneria del Software**

**Perché Condurre Studi Empirici nell'Ingegneria del Software?**

Gli studi empirici, in particolare gli esperimenti e altri studi quantitativi, sono fondamentali nell'ingegneria del software per ottenere risultati oggettivi e statisticamente significativi. Questi risultati sono cruciali per comprendere, controllare, prevedere e migliorare il processo di sviluppo software. Gli studi empirici forniscono informazioni essenziali per il processo decisionale nelle organizzazioni che mirano al miglioramento continuo.

Prima di introdurre nuove tecniche, metodi o strumenti di lavoro, è preferibile effettuare una valutazione empirica dei vantaggi e degli svantaggi di tali cambiamenti. Questo approccio riduce il rischio e aumenta le probabilità di successo dell'implementazione di nuove pratiche.

**Requisiti Fondamentali per il Successo nello Sviluppo Software**

Per avere successo nello sviluppo software, è necessario soddisfare alcuni requisiti di base:

1. **Comprendere il Processo e il Prodotto Software.**
2. **Definire le Qualità del Processo e del Prodotto.**
3. **Valutare Successi e Fallimenti.**
4. **Fornire Feedback Informativo per il Controllo dei Progetti.**
5. **Apprendere dall'Esperienza.**
6. **Impacchettare e Riutilizzare l'Esperienza Rilevante.**

Gli studi empirici supportano il raggiungimento di questi requisiti e sono particolarmente utili nel contesto della ricerca accademica e industriale nell'ingegneria del software, così come nelle organizzazioni che cercano un miglioramento continuo. Un esempio di approccio sistematico all'apprendimento organizzativo è il **"Experience Factory"** proposto da Basili, che si basa sul **Quality Improvement Paradigm**. Questo approccio include anche un meccanismo per definire e valutare obiettivi operativi utilizzando la metodologia **Goal/Question/Metric (GQM)**.

**Valutazione Empirica delle Modifiche ai Processi**

Le organizzazioni che cercano il miglioramento devono valutare l'impatto delle modifiche ai processi (ad esempio, l'introduzione di un nuovo metodo o strumento) prima di implementarle. Gli studi empirici sono fondamentali per ottenere informazioni oggettive e quantificabili sugli effetti di tali modifiche. Le strategie empiriche, come descritto nelle sezioni precedenti, includono sondaggi, studi di caso ed esperimenti, e ciascuna ha il suo posto nel processo di valutazione.

**Quadro per la Valutazione delle Modifiche ai Processi**

Nella valutazione delle modifiche ai processi, le strategie empiriche possono essere utilizzate in tre diversi contesti di ricerca:

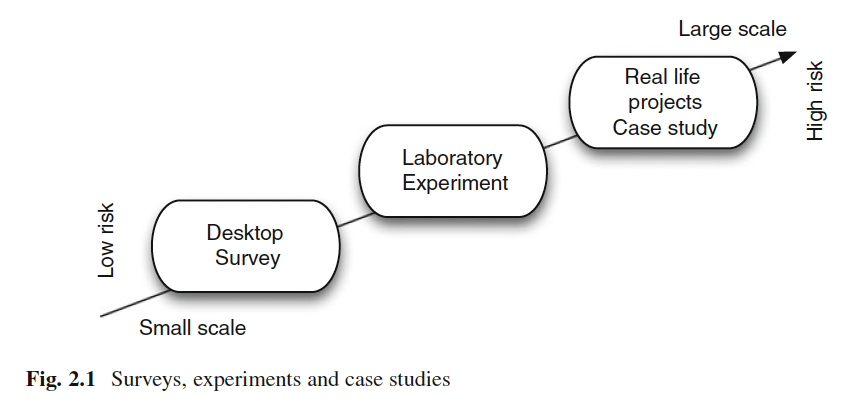
1. **Desktop:** La proposta di cambiamento viene valutata offline, senza eseguire il processo modificato. In questo contesto, è possibile condurre sondaggi, ad esempio, attraverso interviste o studi di letteratura.
2. **Laboratorio:** La proposta di cambiamento viene valutata in un ambiente di laboratorio offline (in vitro), dove un esperimento viene condotto e una parte limitata del processo viene eseguita in modo controllato.
3. **Realtà:** La proposta di cambiamento viene valutata in una situazione di sviluppo reale (in vivo), come nei progetti pilota. In questo contesto, spesso è troppo costoso condurre esperimenti controllati, quindi gli studi di caso sono generalmente più appropriati.

**Scelta della Strategia Empirica**

La scelta della strategia empirica dipende dal contesto, dai costi e dai rischi coinvolti. Ad esempio, testare un nuovo metodo di progettazione in un progetto su larga scala è più rischioso rispetto a uno studio di laboratorio o desktop, poiché il fallimento potrebbe compromettere la qualità del prodotto finale. Gli esperimenti e gli studi di caso sono spesso più costosi rispetto alle valutazioni desktop, poiché coinvolgono l'esecuzione del processo di sviluppo. Tuttavia, la scelta della strategia deve essere fatta caso per caso, considerando quale approccio sia più efficace in base ai costi e ai rischi specifici.

**Approccio Incrementale e Metodologia di Supporto**

Indipendentemente dalla strategia scelta, è consigliabile iniziare con studi su piccola scala e poi espandere lo studio man mano che aumenta la conoscenza e diminuisce il rischio. Inoltre, è necessario un supporto metodologico per lavorare con il miglioramento, raccogliere dati e archiviare le informazioni. Questi aspetti sono ulteriormente discussi nelle sezioni successive.

****

### ****Quality Improvement Paradigm (QIP)****

Il **Quality Improvement Paradigm (QIP)** è uno schema generale di miglioramento progettato specificamente per il settore del software. Questo paradigma è simile al ciclo **Plan/Do/Study/Act (PDSA)**, utilizzato per il miglioramento continuo, e si articola in sei fasi fondamentali.

#### **Fasi del Quality Improvement Paradigm**

1. **Characterize (Caratterizzare):**
   * **Obiettivo:** Comprendere l'ambiente in cui l'organizzazione opera utilizzando modelli disponibili, dati empirici, intuizioni e altre risorse. Durante questa fase, si stabiliscono i **baselines** (punti di riferimento) per i processi aziendali esistenti, identificando la loro criticità.
   * **Esempio:** In un'organizzazione di sviluppo software, questa fase potrebbe comportare la raccolta di dati sulle pratiche attuali di gestione del progetto, sui tassi di difetto e sui tempi di consegna per capire dove sono necessari miglioramenti.
2. **Set Goals (Stabilire Obiettivi):**
   * **Obiettivo:** Basandosi sulla caratterizzazione iniziale e sulle capacità strategiche rilevanti per l'organizzazione, si definiscono obiettivi quantificabili per il successo del progetto e il miglioramento delle prestazioni organizzative. Le aspettative ragionevoli vengono definite utilizzando i baselines stabiliti nella fase di caratterizzazione.
   * **Esempio:** Se l'analisi dei dati mostra che i ritardi nei progetti sono un problema critico, l'obiettivo potrebbe essere quello di ridurre i tempi di consegna del 20% entro il prossimo anno.
3. **Choose Process (Scegliere il Processo):**
   * **Obiettivo:** In base alla caratterizzazione dell'ambiente e agli obiettivi stabiliti, si scelgono i processi appropriati per il miglioramento, insieme ai metodi e agli strumenti di supporto, assicurandosi che siano coerenti con gli obiettivi prefissati.
   * **Esempio:** Se l'obiettivo è ridurre i difetti nel codice, l'organizzazione potrebbe decidere di implementare una nuova metodologia di revisione del codice o adottare strumenti di automazione per i test.
4. **Execute (Eseguire):**
   * **Obiettivo:** Si procede con lo sviluppo del prodotto, fornendo feedback sul progetto basato sui dati raccolti riguardanti il raggiungimento degli obiettivi. Durante questa fase, è cruciale monitorare le prestazioni rispetto agli obiettivi stabiliti e apportare eventuali correzioni.
   * **Esempio:** Durante la fase di esecuzione, si raccolgono dati su quante volte sono stati identificati e corretti i difetti tramite la nuova metodologia di revisione del codice.
5. **Analyze (Analizzare):**
   * **Obiettivo:** Al termine di ogni progetto, i dati e le informazioni raccolte vengono analizzati per valutare le pratiche correnti, identificare problemi, registrare risultati e formulare raccomandazioni per i futuri miglioramenti dei progetti.
   * **Esempio:** L'analisi potrebbe rivelare che la nuova metodologia di revisione del codice ha effettivamente ridotto i difetti, ma ha anche aumentato i tempi di revisione. Si potrebbe raccomandare una formazione aggiuntiva per bilanciare la qualità con l'efficienza.
6. **Package (Impacchettare):**
   * **Obiettivo:** Consolidare l'esperienza acquisita sotto forma di nuovi modelli, aggiornati e raffinati, o altre forme di conoscenza strutturata derivanti da questo e da precedenti progetti.
   * **Esempio:** L'organizzazione potrebbe documentare le best practices emerse dall'uso della nuova metodologia di revisione del codice, creando linee guida e modelli da riutilizzare in progetti futuri.

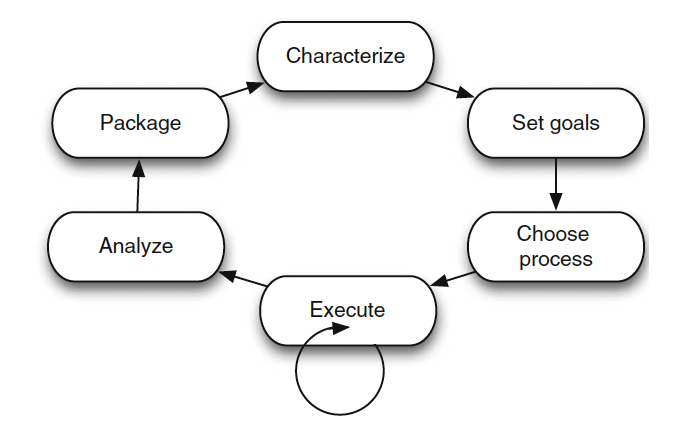
#### **Cicli di Feedback nel QIP**

Il QIP implementa due cicli di feedback principali:

1. **Project Feedback Cycle (Ciclo di Controllo del Progetto):**
   * **Descrizione:** Questo ciclo fornisce feedback continuo al progetto durante la fase di esecuzione. Indipendentemente dagli obiettivi dell'organizzazione, è essenziale che il progetto pilota utilizzi le risorse nel miglior modo possibile. Indicatori quantitativi a livello di progetto e di attività sono utili per prevenire e risolvere problemi.
   * **Esempio:** Durante lo sviluppo di un progetto, il team potrebbe monitorare la quantità di difetti rilevati e risolti in ogni iterazione, utilizzando questi dati per migliorare le prestazioni nelle fasi successive.
2. **Corporate Feedback Cycle (Ciclo di Capitalizzazione):**
   * **Descrizione:** Questo ciclo fornisce un feedback all'organizzazione nel suo complesso. Serve a fornire informazioni analitiche sulle prestazioni del progetto al termine dello stesso, confrontando i dati del progetto con gli standard dell'organizzazione e analizzando le concordanze e le discrepanze. L'esperienza riutilizzabile viene accumulata in una forma che sia utile e applicabile ad altri progetti.
   * **Esempio:** Al termine di un progetto, l'organizzazione potrebbe confrontare i risultati raggiunti con quelli attesi, utilizzando l'analisi per migliorare i processi e diffondere le best practices emerse.

### ****Conclusione****

Il Quality Improvement Paradigm (QIP) offre un approccio strutturato e ciclico per migliorare continuamente la qualità dei processi e dei prodotti software. Attraverso una combinazione di caratterizzazione dell'ambiente, definizione degli obiettivi, scelta dei processi, esecuzione, analisi e impacchettamento delle esperienze, il QIP aiuta le organizzazioni a ottenere risultati misurabili e a migliorare costantemente le loro pratiche di sviluppo software.

****

**Experience Factory**

Il concetto di **Experience Factory** è un elemento chiave del Quality Improvement Paradigm (QIP). Si basa sull'idea che il miglioramento dello sviluppo software richiede un apprendimento continuo. L'Experience Factory ha lo scopo di raccogliere, analizzare e rendere riutilizzabili le esperienze acquisite nei vari progetti di sviluppo software.

**Concetto di Base**

L'Experience Factory si configura come un'organizzazione separata dalla **Project Organization**. Mentre la Project Organization è focalizzata sulla produzione e manutenzione del software, l'Experience Factory si occupa dell'analisi e della sintesi delle esperienze acquisite durante i progetti. Queste esperienze vengono confezionate sotto forma di modelli di esperienza, che possono essere informali, formali o schematici. Questi modelli vengono poi archiviati in un repository chiamato **experience base**.

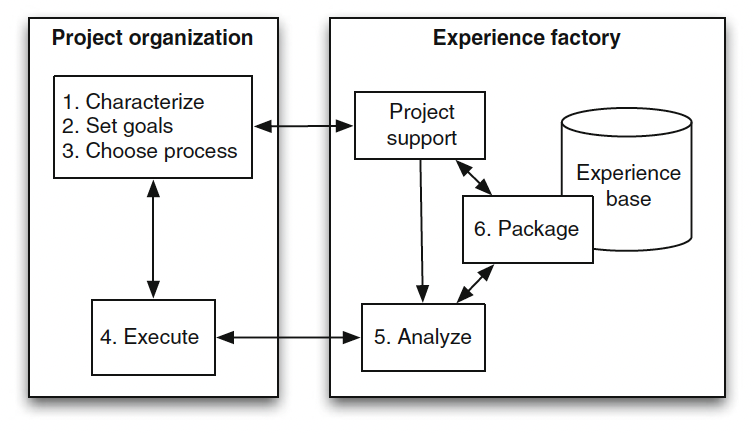
**Funzionamento dell'Experience Factory**

1. **Raccolta delle Informazioni:**
   * La Project Organization fornisce all'Experience Factory informazioni dettagliate sui progetti, comprese le caratteristiche dell'ambiente, i dati di sviluppo, l'uso delle risorse, i record di qualità e le informazioni sui processi.
   * Inoltre, la Project Organization fornisce feedback sulle prestazioni effettive dei modelli utilizzati nel progetto.
2. **Elaborazione e Confezionamento dell'Esperienza:**
   * L'Experience Factory analizza le informazioni ricevute e confeziona le esperienze sotto forma di modelli e misure utili. Questi modelli possono riguardare processi, prodotti e altre forme di conoscenza, e sono creati attraverso l'interazione con persone, documenti e supporti automatizzati.
   * L'obiettivo è rendere queste esperienze facilmente accessibili e modificabili per il riutilizzo in progetti futuri.
3. **Feedback e Supporto ai Progetti:**
   * L'Experience Factory fornisce feedback diretto ai progetti, insieme a obiettivi e modelli personalizzati sulla base di progetti simili già affrontati.
   * Offre anche baselines, strumenti, lezioni apprese e dati specifici per il progetto in corso.

**Vantaggi dell'Experience Factory**

L'Experience Factory crea un ciclo di apprendimento continuo all'interno dell'organizzazione, dove le esperienze passate vengono catturate, analizzate e riutilizzate per migliorare i processi futuri. Questo ciclo permette all'organizzazione di:

* **Ridurre i Rischi:** Valutare nuove tecnologie o processi in modo offline, come in esperimenti controllati o studi di caso, per evitare rischi elevati.
* **Migliorare Continuamente:** Apprendere dalle esperienze passate e implementare miglioramenti nei processi di sviluppo.
* **Ottimizzare l'Uso delle Risorse:** Fornire supporto ai progetti in corso basato su dati empirici e modelli collaudati, migliorando così l'efficienza e la qualità del software prodotto.

****

**Goal/Question/Metric (GQM) Method**

Il metodo **Goal/Question/Metric (GQM)** è un approccio strutturato per misurare in modo mirato e significativo all'interno di un'organizzazione. Si basa su tre principi fondamentali:

1. **Specificare gli Obiettivi:** L'organizzazione deve definire chiaramente gli obiettivi per sé stessa e per i suoi progetti.
2. **Collegare gli Obiettivi ai Dati:** Gli obiettivi devono essere collegati ai dati che li definiscono in modo operativo, garantendo che la misurazione sia pertinente e utile.
3. **Fornire un Quadro per l'Interpretazione:** Deve esistere un quadro strutturato che permetta di interpretare i dati raccolti in relazione agli obiettivi dichiarati.

**Struttura del Modello di Misurazione GQM**

Il risultato dell'applicazione del metodo GQM è un modello di misurazione che si focalizza su un insieme specifico di questioni e un insieme di regole per l'interpretazione dei dati di misurazione. Questo modello di misurazione ha tre livelli gerarchici:

1. **Livello Concettuale (Goal - Obiettivo):**
   * **Descrizione:** Un obiettivo viene definito per un oggetto (che può essere un prodotto, un processo o una risorsa) per varie ragioni, tenendo conto di diversi modelli di qualità, da diversi punti di vista e in relazione a un ambiente specifico.
   * **Esempio:** Un obiettivo potrebbe essere migliorare la qualità del codice in un progetto software riducendo il numero di bug rilevati durante i test.
2. **Livello Operativo (Question - Domanda):**
   * **Descrizione:** Una serie di domande viene utilizzata per caratterizzare come l'ottenimento di un obiettivo specifico sarà valutato. Queste domande cercano di caratterizzare gli oggetti di misurazione rispetto a un aspetto qualitativo selezionato e di determinarne la qualità dal punto di vista selezionato.
   * **Esempio:** Una domanda potrebbe essere: "Quanti bug critici sono stati rilevati nei test di integrazione?" Questa domanda aiuta a capire in che misura è stato raggiunto l'obiettivo di migliorare la qualità del codice.
3. **Livello Quantitativo (Metric - Metrica):**
   * **Descrizione:** Un insieme di dati è associato a ogni domanda per rispondere in modo quantitativo, sia oggettivamente che soggettivamente.
   * **Esempio:** Le metriche associate alla domanda potrebbero includere il numero di bug rilevati, il numero di bug risolti e il tempo medio di risoluzione dei bug.

**Processo di Applicazione del Metodo GQM**

Il processo di applicazione del metodo GQM inizia con la formulazione degli obiettivi, che è un passo critico per il successo dell'approccio. Gli obiettivi vengono formulati sulla base di:

1. **Politiche e Strategie dell'Organizzazione:** Le linee guida strategiche definiscono le priorità per l'organizzazione.
2. **Descrizioni di Processi e Prodotti:** Una comprensione dettagliata dei processi e dei prodotti esistenti aiuta a stabilire obiettivi realistici e rilevanti.
3. **Modelli Organizzativi:** I modelli che descrivono l'organizzazione forniscono il contesto per formulare obiettivi che siano coerenti con la struttura e la cultura aziendale.

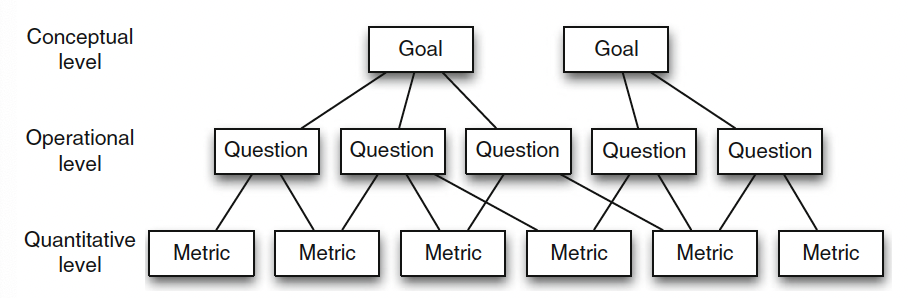
Una volta formulati gli obiettivi, si sviluppano le domande basate su questi obiettivi. Le domande sono poi associate a metriche appropriate che forniranno le risposte in modo quantitativo.

**Utilizzo Pratico del Metodo GQM**

Il metodo GQM è utilizzato per migliorare i processi basati su misurazioni e può essere applicato a una vasta gamma di contesti all'interno dell'ingegneria del software. Le linee guida pratiche su come utilizzare il GQM per il miglioramento dei processi basati sulle misurazioni sono fornite da Briand et al. e da van Solingen e Berghout.

**Conclusione**

Il metodo GQM è uno strumento potente per garantire che le misurazioni all'interno di un'organizzazione siano allineate con gli obiettivi strategici e operativi. La sua struttura gerarchica consente di passare dagli obiettivi generali a domande specifiche e poi a metriche dettagliate, facilitando l'interpretazione dei dati raccolti e il loro utilizzo per il miglioramento continuo dei processi e dei prodotti.



### ****2.10 Trasferimento Tecnologico Basato su Evidenze Empiriche****

Il trasferimento tecnologico basato su studi empirici non solo ha valore in sé, ma rappresenta anche un'opportunità per la collaborazione tra mondo accademico e industria. Nell'ingegneria del software, che è una disciplina di ricerca applicata, è cruciale affrontare problemi rilevanti per l'industria, e non limitarsi a ricerche puramente accademiche. La collaborazione tra università e industria facilita il trasferimento di conoscenze e l'applicazione di nuovi metodi, tecnologie e strumenti dal contesto accademico a quello industriale.

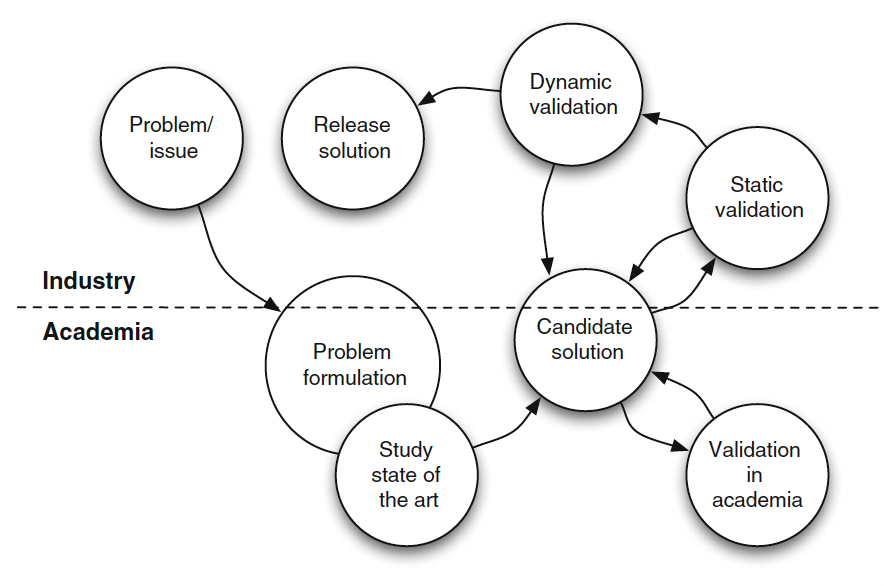
### ****Modello di Trasferimento Tecnologico di Gorschek et al.****

Gorschek et al. hanno documentato un modello per il trasferimento tecnologico basato su un'esperienza collaborativa a lungo termine. Questo modello comprende sette fasi principali, che illustrano come diverse strategie empiriche, in particolare gli esperimenti, possano essere utilizzate per miglioramenti guidati empiricamente. Di seguito vengono descritte le sette fasi:

1. **Identificazione del Problema Industriale:**
   * **Descrizione:** Il primo passo è identificare le sfide reali in un contesto industriale specifico, il che implica la presenza del ricercatore presso il partner industriale. Questa fase può essere svolta utilizzando sondaggi o interviste.
   * **Obiettivo:** Catturare le sfide che possono essere formulate come problemi di ricerca, evitando che il ricercatore diventi un semplice consulente per problemi a breve termine.
2. **Formulazione del Problema:**
   * **Descrizione:** Una volta identificata la sfida, essa deve essere formulata come un problema di ricerca con domande specifiche. Se vengono identificate più sfide, è necessario dare priorità a quelle da affrontare. Inoltre, è importante identificare un referente all'interno dell'azienda che funga da "campione" per la collaborazione di ricerca.
   * **Obiettivo:** Garantire che il problema sia ben definito e che esista un supporto interno all'azienda per la ricerca.
3. **Sviluppo di una Soluzione Candidata:**
   * **Descrizione:** Sulla base degli approcci esistenti e delle esigenze specifiche, viene sviluppata una soluzione candidata, che può includere l'adattamento ai processi, metodi, tecnologie e strumenti attualmente utilizzati dall'azienda.
   * **Obiettivo:** Collaborare strettamente con il partner industriale per assicurarsi che la soluzione sia applicabile e possa essere personalizzata per l'industria.
4. **Validazione Accademica:**
   * **Descrizione:** Una prima validazione della soluzione proposta viene condotta in un ambiente accademico per minimizzare i rischi. Questa validazione può essere eseguita sotto forma di esperimenti o studi di caso con studenti o rappresentanti dell'industria.
   * **Obiettivo:** Identificare eventuali difetti evidenti nella soluzione e proporre miglioramenti, garantendo che la migliore soluzione possibile sia disponibile per l'implementazione industriale.
5. **Validazione Statica:**
   * **Descrizione:** I rappresentanti dell'industria valutano la soluzione candidata offline, attraverso presentazioni, interviste o workshop. Questo passaggio è cruciale per sensibilizzare l'organizzazione alla nuova soluzione e ridurre la resistenza al cambiamento.
   * **Obiettivo:** Apportare modifiche alla soluzione in base al feedback e prepararla per una validazione dinamica.
6. **Validazione Dinamica:**
   * **Descrizione:** Una volta superata la validazione statica, si passa alla validazione dinamica, preferibilmente attraverso una valutazione pilota. Questa fase può essere eseguita in un progetto, un sotto-progetto o una parte specifica di un sistema, monitorando da vicino l'applicazione della nuova soluzione.
   * **Obiettivo:** Valutare l'efficacia della soluzione in un contesto reale e apportare ulteriori miglioramenti se necessario.
7. **Rilascio della Soluzione:**
   * **Descrizione:** La soluzione generica deve essere personalizzata per ogni situazione unica. È essenziale garantire che la soluzione venga trasferita correttamente all'interno dell'organizzazione, con supporto sufficiente in termini di descrizioni, formazione e strumenti.
   * **Obiettivo:** Assicurare l'integrazione della nuova soluzione nell'organizzazione e studiarne l'uso su scala più ampia attraverso ulteriori studi empirici.

### ****Conclusioni****

Il modello di trasferimento tecnologico basato su evidenze empiriche mostra come le strategie empiriche possano essere applicate per supportare il trasferimento di nuovi risultati di ricerca dall'identificazione dei bisogni fino all'uso industriale effettivo. È importante notare che, mentre i rappresentanti industriali sono principalmente interessati alla personalizzazione della soluzione per il loro ambiente, i ricercatori sono focalizzati sulla valutazione della soluzione generica. Tuttavia, entrambe le parti traggono vantaggio dalla collaborazione: l'industria riceve una soluzione pratica a una sfida identificata, mentre i ricercatori possono testare e validare i loro risultati in un ambiente industriale reale.

****

### ****2.11 Etica nella Sperimentazione****

Ogni attività di ricerca empirica che coinvolge soggetti umani deve tenere in considerazione aspetti etici. Mentre alcuni di questi aspetti sono regolati da leggi nazionali, altri non lo sono affatto. La ricerca condotta da Andrews e Pradhan ha identificato problematiche etiche nell'ingegneria del software e ha evidenziato come le politiche esistenti siano spesso insufficienti. Allo stesso modo, Hall e Flynn hanno condotto un sondaggio sulla pratica etica nel Regno Unito, scoprendo una preoccupante mancanza di consapevolezza, un problema che non sembra limitato solo a quel paese.

### ****Principi Chiave dell'Etica nella Sperimentazione****

Singer e Vinson hanno identificato quattro principi chiave che dovrebbero guidare la condotta degli studi empirici:

1. **Consenso Informato:**
   * I soggetti devono dare il loro consenso informato per partecipare, il che implica che dovrebbero avere accesso a tutte le informazioni rilevanti sullo studio prima di prendere una decisione. Questa decisione deve essere esplicita e libera, senza pressioni implicite da parte di manager, professori, ecc.
2. **Valore Scientifico:**
   * Lo studio deve avere un valore scientifico sufficiente per giustificare i rischi a cui i soggetti sono esposti, anche se questi rischi sono minimi.
3. **Riservatezza:**
   * I ricercatori devono prendere tutte le misure possibili per mantenere la riservatezza dei dati e delle informazioni sensibili, anche quando questo entra in conflitto con gli interessi di pubblicazione.
4. **Beneficenza:**
   * I benefici dello studio devono superare i rischi, non solo per i singoli soggetti, ma anche per i gruppi di soggetti e le organizzazioni coinvolte.

### ****Linee Guida Pratiche****

Questi principi si traducono in linee guida pratiche relative alla pianificazione, conduzione e reporting di uno studio sperimentale.

#### **Revisione Etica**

In alcuni paesi, la legislazione richiede una revisione etica per gli studi che coinvolgono soggetti umani. Questi studi devono seguire procedure specifiche e fornire documentazione per essere approvati da un Comitato Etico (ERB). Sebbene queste procedure siano derivate principalmente dalle esigenze della ricerca biomedica, non sono sempre adeguatamente adattate alle necessità dell'ingegneria del software.

#### **Consenso Informato**

La base per uno studio empirico orientato agli esseri umani è che i soggetti partecipano volontariamente e hanno abbastanza informazioni per decidere se partecipare o meno. I soggetti devono poter ritirarsi dallo studio in qualsiasi momento, senza subire penalità. Il consenso dovrebbe essere dato per iscritto e includere elementi come il titolo del progetto, le informazioni di contatto, i rischi e i benefici, la riservatezza e il diritto di ritirarsi.

#### **Riservatezza**

I soggetti devono essere certi che le informazioni che condividono con i ricercatori rimarranno confidenziali. La riservatezza riguarda tre aspetti principali:

* **Privacy dei Dati:** Accesso limitato ai dati, ad esempio tramite protezione con password e crittografia.
* **Anonimato dei Dati:** Mantenere separate le identità dei soggetti dai dati raccolti.
* **Anonimato della Partecipazione:** Mantenere segreta la decisione di partecipare.

#### **Risultati Sensibili**

I risultati di uno studio empirico possono essere sensibili per diversi stakeholder. I ricercatori devono essere pronti a gestire situazioni in cui i risultati possono avere implicazioni delicate per i soggetti, gli sponsor o gli stessi ricercatori. Ciò può includere l'adozione di procedure per garantire l'indipendenza dei risultati, come la confidenzialità dei dati e il diritto alla pubblicazione indipendente.

#### **Induzione e Feedback**

Per motivare la partecipazione, possono essere offerti incentivi, come l'esperienza acquisita o compensi monetari. Tuttavia, questi incentivi devono essere bilanciati per garantire che il consenso sia volontario e non forzato. Inoltre, per mantenere relazioni di lungo termine con i soggetti e garantire la fiducia, è importante fornire feedback sui risultati e sulle analisi dello studio.

### ****Conclusione sull'Etica****

Singer e Vinson hanno sollecitato lo sviluppo di un codice etico specifico per l'ingegneria del software empirica. Sebbene non esista ancora un codice etico formale, le linee guida proposte da Vinson e Singer rappresentano un passo importante. Lo sviluppo di linee guida etiche concrete e personalizzate per la ricerca empirica nell'ingegneria del software beneficerebbe sia i soggetti coinvolti, proteggendoli, sia lo sviluppo del campo di ricerca stesso.

**ESERCIZI**

**2.1. Qual è la differenza tra ricerca qualitativa e quantitativa?**

* **Ricerca qualitativa**: Si concentra sull'interpretazione e la comprensione dei fenomeni attraverso l'analisi di dati non numerici, come interviste, osservazioni, e testi. È spesso usata per esplorare i motivi, le opinioni e le esperienze soggettive delle persone.

**Esempio**: In un'indagine qualitativa su come gli sviluppatori percepiscono un nuovo strumento di sviluppo software, i dati potrebbero essere raccolti attraverso interviste approfondite, e l'obiettivo sarebbe capire le loro esperienze e opinioni dettagliate.

* **Ricerca quantitativa**: Si basa sulla raccolta e analisi di dati numerici per identificare tendenze, relazioni e modelli. Questo tipo di ricerca è utile per quantificare variabili e per testare ipotesi specifiche.

**Esempio**: Un esperimento quantitativo potrebbe misurare il numero di bug risolti utilizzando due diversi strumenti di debugging, per determinare quale strumento è più efficace.

**2.2. Cos'è un sondaggio? Fai esempi di diversi tipi di sondaggi nell'ingegneria del software.**

Un **sondaggio** è un metodo di raccolta di informazioni da un gruppo di persone, progettato per descrivere, confrontare o spiegare le loro conoscenze, atteggiamenti e comportamenti. È spesso utilizzato in studi retrospettivi per valutare l'efficacia di una tecnica o strumento.

**Esempi di sondaggi nell'ingegneria del software**:

* **Sondaggio descrittivo**: Un sondaggio per determinare quanti sviluppatori in un'azienda utilizzano una specifica metodologia di sviluppo software.
* **Sondaggio esplorativo**: Un sondaggio condotto prima dell'adozione di una nuova tecnica per capire le aspettative degli sviluppatori riguardo a quella tecnica.
* **Sondaggio esplicativo**: Un sondaggio per capire perché alcuni sviluppatori preferiscono un ambiente di sviluppo rispetto a un altro, esaminando le relazioni tra le preferenze e altre variabili come l'esperienza.

**2.3. Qual è il ruolo delle repliche e delle revisioni sistematiche della letteratura nella costruzione della conoscenza empirica?**

* **Repliche**: Le repliche consistono nel ripetere un esperimento o uno studio per verificare se i risultati ottenuti sono validi e riproducibili in contesti diversi o con soggetti diversi. Le repliche rafforzano la fiducia nei risultati empirici e aiutano a identificare variabili non considerate nel design originale.
* **Revisioni sistematiche della letteratura**: Queste revisioni raccolgono e sintetizzano le evidenze empiriche da più studi per rispondere a domande di ricerca specifiche. Le revisioni sistematiche aiutano a costruire una base di conoscenza solida e coerente, identificando trend e gap nella ricerca esistente.

**Esempio**: Una meta-analisi che combina i risultati di diversi studi empirici sull'efficacia delle metodologie agili nell'ingegneria del software fornisce una visione più completa rispetto a singoli studi isolati.

**2.4. Come può essere combinata l'Experience Factory con il metodo Goal/Question/Metric e studi empirici in un contesto di trasferimento tecnologico?**

* **Experience Factory**: È un'organizzazione che supporta lo sviluppo del prodotto raccogliendo, analizzando e sintetizzando le esperienze acquisite nei progetti per creare modelli di esperienza riutilizzabili.
* **Metodo Goal/Question/Metric (GQM)**: Questo metodo struttura la misurazione definendo obiettivi (Goal), formulando domande (Question) per valutare il raggiungimento degli obiettivi, e associando metriche (Metric) per rispondere a tali domande.
* **Combinazione in un contesto di trasferimento tecnologico**: L'Experience Factory può utilizzare il metodo GQM per strutturare la raccolta e l'analisi delle esperienze durante i progetti di trasferimento tecnologico. Gli studi empirici, come esperimenti e studi di caso, possono essere utilizzati per valutare l'efficacia delle nuove tecnologie e delle modifiche di processo. I risultati vengono poi analizzati e memorizzati nell'Experience Factory, facilitando il miglioramento continuo e il trasferimento delle conoscenze ad altri progetti.

**Esempio**: Se un'azienda introduce un nuovo strumento di gestione dei requisiti, l'Experience Factory potrebbe utilizzare il GQM per definire obiettivi specifici per migliorare la gestione dei requisiti, raccogliere dati empirici durante l'implementazione dello strumento, e analizzare i risultati per migliorare ulteriormente il processo.

**2.5. Quali sono i principi etici chiave da osservare durante la conduzione di esperimenti?**

1. **Consenso informato**: I partecipanti devono dare il loro consenso volontario e informato prima di partecipare allo studio. Devono comprendere pienamente lo scopo dello studio, i rischi e i benefici, e devono poter ritirarsi in qualsiasi momento senza conseguenze.
2. **Valore scientifico dello studio**: Lo studio deve avere un valore scientifico tale da giustificare i rischi per i partecipanti, anche se minimi.
3. **Riservatezza**: I dati personali e le informazioni sensibili raccolte durante lo studio devono essere mantenuti confidenziali. Questo include la protezione della privacy dei dati, l'anonimato dei partecipanti, e la riservatezza della partecipazione stessa.
4. **Valutazione dei rischi e benefici**: I benefici derivanti dallo studio devono superare i rischi per i partecipanti, non solo individualmente ma anche collettivamente.
5. **Inducement**: Gli incentivi offerti ai partecipanti non devono essere tali da compromettere la volontarietà della loro partecipazione.

**Esempio**: Durante un esperimento in un'azienda, i ricercatori devono assicurarsi che i dipendenti partecipino volontariamente e che i loro dati siano trattati in modo da proteggere la loro privacy e riservatezza.

### ****Capitolo 3: Misurazioni****

La misurazione nel software è cruciale per consentire il controllo dei progetti, dei prodotti e dei processi. Come affermato da DeMarco, "Non puoi controllare ciò che non puoi misurare". La misurazione è anche una parte centrale negli studi empirici, che sono utilizzati per indagare gli effetti di input specifici sull'oggetto di studio. Per controllare lo studio e osservare gli effetti, dobbiamo essere in grado di misurare sia gli input per descrivere ciò che causa l'effetto sull'output, sia l'output stesso. Senza misurazioni, non è possibile avere il controllo desiderato, e quindi uno studio empirico non può essere condotto.

### ****Concetti di Base****

Una misura è una mappatura dall'attributo di un'entità a un valore di misurazione, di solito un valore numerico. Le entità sono oggetti che possiamo osservare nel mondo reale. L'obiettivo della mappatura degli attributi in un valore di misurazione è quello di caratterizzare e manipolare gli attributi in modo formale. Una delle caratteristiche di base di una misura è quindi che deve preservare le osservazioni empiriche dell'attributo.

Quando utilizziamo una misura negli studi empirici, dobbiamo essere certi che la misura sia valida. Per essere valida, la misura non deve violare le proprietà essenziali dell'attributo che misura e deve essere una corretta caratterizzazione matematica dell'attributo.

#### **Validità delle Misure**

* **Validità Analitica:** La misura deve essere in grado di catturare accuratamente e in modo affidabile l'elemento di interesse.
* **Validità Empirica:** Riguarda quanto bene, ad esempio, un punteggio si correla con qualcosa misurato in un altro contesto.

#### **Effetto della Dimensione**

L'effetto della dimensione è un modo semplice per quantificare la differenza tra due gruppi. Questo è particolarmente importante negli esperimenti, poiché può essere possibile mostrare una differenza statisticamente significativa tra due gruppi, ma potrebbe non essere significativa dal punto di vista pratico.

### ****Tipi di Scale di Misurazione****

La mappatura da un attributo a un valore di misurazione può essere effettuata in molti modi diversi, e ogni mappatura di un attributo è una scala. Ad esempio, se l'attributo è la lunghezza di un oggetto, possiamo misurarlo in metri, centimetri o pollici, ognuno dei quali è una scala diversa della misura della lunghezza.

Se vogliamo trasformare una misura in un'altra scala e questa trasformazione preserva la relazione tra gli oggetti, si chiama **trasformazione ammissibile**. Le affermazioni che rimangono vere anche dopo una riscalatura sono chiamate **significative**, mentre quelle che non lo sono diventano **insignificanti**.

### ****Classificazione delle Misure****

Le misure possono essere classificate in due modi:

1. **Misure Dirette o Indirette:**
   * **Dirette:** Quando misuriamo direttamente un attributo.
   * **Indirette:** Quando la misura deriva da un'altra misurazione.
2. **Misure Oggettive o Soggettive:**
   * **Oggettive:** Basate su fatti osservabili e misurabili.
   * **Soggettive:** Basate su percezioni o giudizi individuali.

### ****Conclusione****

La teoria della misurazione è fondamentale per garantire che gli studi empirici siano validi e che i risultati siano significativi. Comprendere le diverse scale di misurazione, insieme alla validità delle misure, è essenziale per condurre esperimenti e ricerche nell'ingegneria del software in modo efficace.

### ****3.1.1 Tipi di Scale****

I tipi di scale di misurazione più comuni sono i seguenti:

#### **1. Scala Nominale**

* **Descrizione:** La scala nominale è la meno potente tra i tipi di scale. Mappa semplicemente l'attributo di un'entità in un nome o simbolo. Questa mappatura può essere vista come una classificazione delle entità in base all'attributo.
* **Trasformazioni Possibili:** Le trasformazioni ammissibili per le scale nominali sono quelle che preservano il fatto che le entità possono essere mappate solo uno a uno.
* **Esempi:** Classificazione, etichettatura e tipizzazione dei difetti.

#### **2. Scala Ordinale**

* **Descrizione:** La scala ordinale classifica le entità secondo un criterio di ordinamento ed è quindi più potente della scala nominale. Esempi di criteri di ordinamento sono "maggiore di", "migliore di" e "più complesso".
* **Trasformazioni Possibili:** Le trasformazioni ammissibili per la scala ordinale sono quelle che preservano l'ordine delle entità, ossia M′=F(M), dove M′ e M sono misure diverse sullo stesso attributo e F è una funzione monotona crescente.
* **Esempi:** Voti scolastici e complessità del software.

#### **3. Scala a Intervalli**

* **Descrizione:** La scala a intervalli è utilizzata quando la differenza tra due misure è significativa, ma non il valore in sé. Questa scala ordina i valori allo stesso modo della scala ordinale, ma include una nozione di "distanza relativa" tra due entità, rendendola quindi più potente della scala ordinale.
* **Trasformazioni Possibili:** Le trasformazioni ammissibili con questo tipo di scala sono quelle in cui le misure sono una combinazione lineare l'una dell'altra, ossia M′=αM+β, dove M′ e M sono misure diverse sullo stesso attributo.
* **Esempi:** La temperatura misurata in gradi Celsius o Fahrenheit.

#### **4. Scala di Rapporto**

* **Descrizione:** Se esiste un valore zero significativo e il rapporto tra due misure ha senso, si può usare una scala di rapporto. Questa scala è la più potente poiché permette di fare affermazioni significative sul rapporto tra le misure.
* **Trasformazioni Possibili:** Le trasformazioni ammissibili sono quelle che hanno lo stesso zero e le scale differiscono solo per un fattore, ossia M′=αM, dove M′ e M sono misure diverse sullo stesso attributo.
* **Esempi:** Lunghezza, temperatura misurata in Kelvin e durata di una fase di sviluppo.

### ****Relazione con la Ricerca Qualitativa e Quantitativa****

Le scale di misurazione sono correlate alla ricerca qualitativa e quantitativa. In generale:

* **Ricerca Qualitativa:** Si occupa di misurazioni sulle scale nominali e ordinali.
* **Ricerca Quantitativa:** Tratta misurazioni sulle scale a intervalli e di rapporto.

Questa distinzione influisce anche sui tipi di statistiche che possono essere utilizzate sulle misure, come verrà discusso ulteriormente nel Capitolo 10.

### ****3.1.2 Misure Oggettive e Soggettive****

Le misure possono essere classificate in due categorie principali: **oggettive** e **soggettive**.

#### **Misure Oggettive**

* **Descrizione:** Una misura è oggettiva quando il valore misurato non dipende da un giudizio soggettivo e si basa unicamente sull'oggetto misurato. Le misure oggettive sono riproducibili: possono essere ripetute più volte da diversi ricercatori e produrranno lo stesso valore (entro il margine di errore di misurazione).
* **Esempi:**
  + **Linee di Codice (LOC):** Il numero di linee di codice in un programma è una misura oggettiva. Chiunque conti le linee di codice di un programma dovrebbe ottenere lo stesso numero.
  + **Data di Consegna:** La data in cui un progetto software viene consegnato è un'altra misura oggettiva; è un fatto concreto che può essere verificato senza interpretazione.

#### **Misure Soggettive**

* **Descrizione:** Una misura è soggettiva quando il valore misurato dipende non solo dall'oggetto, ma anche dal punto di vista di chi effettua la misurazione. Qui, la persona che misura contribuisce con un giudizio, e la misura può variare se viene effettuata di nuovo, magari da un'altra persona o in un altro contesto.
* **Esempi:**
  + **Competenze del Personale:** Valutare le competenze di un programmatore è soggettivo perché dipende dal giudizio di chi valuta, che può basarsi su vari criteri (esperienza, conoscenza, performance).
  + **Usabilità:** La percezione dell'usabilità di un software può variare da persona a persona, rendendo questa una misura soggettiva.

Le misure soggettive sono sempre potenzialmente soggette a bias, ovvero distorsioni introdotte dalla percezione e dal giudizio umano.

### ****3.1.3 Misure Dirette o Indirette****

Le misure possono essere ulteriormente classificate in **dirette** e **indirette**, a seconda che l'attributo d'interesse possa essere misurato direttamente o debba essere derivato da altre misure.

#### **Misure Dirette**

* **Descrizione:** Una misura diretta è quella che può essere misurata direttamente, senza la necessità di ricorrere ad altre misurazioni.
* **Esempi:**
  + **Linee di Codice (LOC):** Come già menzionato, il conteggio delle linee di codice in un programma è una misura diretta.
  + **Numero di Difetti:** Il numero di difetti trovati durante i test è un'altra misura diretta; si basa sul conteggio effettivo dei difetti rilevati.

#### **Misure Indirette**

* **Descrizione:** Una misura indiretta si ottiene derivando il valore da altre misure dirette. Queste misure sono utili quando l'attributo d'interesse non può essere misurato direttamente.
* **Esempi:**
  + **Densità di Difetti:** Questo è un esempio di misura indiretta, calcolata come il numero di difetti diviso per il numero di linee di codice.
  + **Produttività dei Programmatori:** Un altro esempio è la produttività dei programmatori, che può essere calcolata dividendo il numero di linee di codice prodotte per lo sforzo (ad esempio, ore di lavoro) impiegato.

### ****Conclusione****

La distinzione tra misure oggettive e soggettive, così come tra misure dirette e indirette, è fondamentale per comprendere come raccogliere e interpretare i dati nei progetti di ingegneria del software. Le misure oggettive e dirette sono generalmente preferite per la loro riproducibilità e affidabilità, mentre le misure soggettive e indirette richiedono una maggiore attenzione per evitare bias e interpretazioni errate.

### ****3.2 Misurazioni nell'Ingegneria del Software****

Gli oggetti di interesse nell'ingegneria del software possono essere suddivisi in tre classi principali:

1. **Processo:**
   * **Descrizione:** Il processo descrive le attività necessarie per produrre il software.
   * **Esempio:** Attività di test.
   * **Misure:**
     + **Attributo Interno:** Sforzo (tempo o risorse impiegate nel processo).
     + **Attributo Esterno:** Costo (costo totale del processo di test).
2. **Prodotto:**
   * **Descrizione:** I prodotti sono gli artefatti, i deliverable o i documenti che risultano da un'attività di processo.
   * **Esempio:** Codice sorgente.
   * **Misure:**
     + **Attributo Interno:** Dimensione (ad esempio, linee di codice).
     + **Attributo Esterno:** Affidabilità (quanto è probabile che il software funzioni senza errori).
3. **Risorse:**
   * **Descrizione:** Le risorse sono oggetti come il personale, l'hardware o il software necessari per un'attività di processo.
   * **Esempio:** Personale.
   * **Misure:**
     + **Attributo Interno:** Età (età media del personale).
     + **Attributo Esterno:** Produttività (quanto codice o quante funzionalità produce il team in un certo periodo di tempo).

#### **Attributi Interni ed Esterni**

In ogni classe, si fa una distinzione tra **attributi interni** ed **esterni**:

* **Attributi Interni:** Misurabili direttamente in termini dell'oggetto stesso.
* **Attributi Esterni:** Misurabili solo rispetto a come l'oggetto si relaziona con altri oggetti.

Spesso, gli ingegneri del software vogliono fare affermazioni su un attributo esterno di un oggetto, che però di solito è una misura indiretta e deve essere derivata da attributi interni (misure dirette). Ad esempio, la **produttività** del personale (un attributo esterno) potrebbe essere misurata indirettamente tramite il numero di linee di codice prodotte (un attributo interno).

#### **Sfide nelle Misurazioni dell'Ingegneria del Software**

Le misurazioni nell'ingegneria del software differiscono dalle misurazioni in altri domini, come la fisica, dove gli attributi e le modalità di misurazione sono spesso più chiari e universalmente accettati. Nell'ingegneria del software, può essere difficile definire un attributo in modo che tutti siano d'accordo su come misurarlo.

Inoltre, è difficile dimostrare che le misure siano altro che tipi di scala nominale o ordinale, il che limita l'uso di metodi di analisi statistica più potenti che richiedono scale di intervalli o di rapporto. La validazione delle misure indirette è più complessa poiché sia le misure dirette che i modelli utilizzati per derivare la misura esterna devono essere validati.

#### **Analisi Statistica e Tipi di Scala**

La scelta dei tipi di scala nelle misurazioni è importante perché influenza i metodi di analisi statistica che possono essere utilizzati. Sebbene l'analisi statistica formale dipenda dal tipo di scala, molti metodi statistici sono robusti rispetto a distorsioni non lineari della scala degli intervalli, a patto che le distorsioni non siano troppo estreme. Questo significa che, con attenzione e consapevolezza dei rischi, è possibile utilizzare metodi statistici più potenti anche se non si può dimostrare che si dispone di scale di intervallo o di rapporto.

Briand et al. sostengono che possiamo usare metodi di analisi statistica più potenti anche se non possiamo provare di avere scale di intervallo o di rapporto. Questo approccio può produrre risultati utili senza dover ricorrere a campioni molto grandi, che altrimenti sarebbero necessari.

### ****Conclusione****

Le misurazioni nell'ingegneria del software sono complesse e richiedono una comprensione approfondita sia degli attributi interni che esterni degli oggetti di interesse. Sebbene ci siano limitazioni intrinseche nelle misurazioni software rispetto ad altri campi, un'analisi attenta e l'uso giudizioso di strumenti statistici possono fornire preziose intuizioni per migliorare i processi, i prodotti e le risorse nell'ingegneria del software.

### ****3.3 Misurazioni nella Pratica****

Nella pratica, le metriche sono definite dal ricercatore e poi raccolte durante la fase operativa dello studio empirico. La modalità di raccolta delle metriche è cruciale e dovrebbe idealmente richiedere uno sforzo minimo da parte dei soggetti coinvolti nello studio. In molti esperimenti, i soggetti compilano moduli per fornire i dati necessari, ma è possibile anche definire sistemi di strumentazione in cui i dati vengono raccolti automaticamente, ad esempio, dall'ambiente di sviluppo. Lethbridge et al. discutono diverse tecniche generali per la raccolta dei dati.

### ****Qualità delle Metriche Raccolte****

Poiché le metriche raccolte costituiscono la base per l'analisi successiva, la loro qualità è fondamentale. È essenziale comprendere esattamente che tipo di metriche vengono raccolte, a quale tipo di scala appartengono e che tipo di distribuzione rappresentano, in particolare se i dati sono distribuiti normalmente o meno.

#### **Distribuzione dei Dati**

La distribuzione dei dati può essere analizzata attraverso statistiche descrittive. Ad esempio, i dati possono essere rappresentati graficamente o analizzati con altre tecniche per determinare la normalità della distribuzione. Questo aspetto è ulteriormente approfondito nel Capitolo 10.

#### **Definizione delle Metriche**

La definizione delle metriche può influire significativamente sulla loro capacità di riflettere ciò che interessa al ricercatore. Ad esempio, Kitchenham et al. hanno confrontato due modi di visualizzare la produttività e hanno dimostrato che un grafico a dispersione che mostra lo sforzo rispetto alla dimensione del progetto fornisce informazioni migliori rispetto a un grafico che mostra la produttività nel tempo.

Un consiglio generale è di evitare l'uso di metriche costruite dal rapporto di due misure indipendenti, a meno che non si comprenda appieno l'implicazione della misura stessa.

### ****Garanzia di Qualità dei Dati Raccolti****

Durante l'esecuzione dello studio, è importante assicurarsi che i dati raccolti siano corretti. Ciò significa che il ricercatore dovrebbe applicare procedure di garanzia della qualità durante l'esperimento, ad esempio, rivedendo come i soggetti compilano i moduli, controllando la coerenza tra i diversi valori, ecc. La validazione dei dati è ulteriormente discussa nel Capitolo 8.

### ****Ruolo dell'Inventore o Proprietario della Metodologia****

Un fattore importante riguarda chi è l'inventore o proprietario degli aspetti indagati in un esperimento. Idealmente, qualcuno diverso dall'inventore di nuovi metodi dovrebbe valutare tali metodi negli esperimenti e in altre ricerche, come raccomandato da Kitchenham et al. Questo perché l'inventore di un metodo ha naturalmente l'interesse che il metodo performi bene, e c'è sempre il rischio che il ricercatore, consapevolmente o inconsapevolmente, selezioni metriche favorevoli per il metodo investigato. Inoltre, se i soggetti sanno che il ricercatore è l'inventore del metodo investigato, ciò potrebbe influire sulle loro prestazioni. Se gli esperimenti vengono condotti studiando i propri metodi, il design e la selezione delle metriche dovrebbero essere rivisti da ricercatori esterni.

### ****Conclusione****

La raccolta di metriche durante studi empirici nell'ingegneria del software richiede un'attenzione particolare alla qualità e alla validità dei dati raccolti. Le metriche devono essere ben definite, e la loro raccolta e analisi devono essere condotte con rigore per garantire che i risultati siano affidabili e significativi. Inoltre, per evitare bias, è preferibile che la valutazione di nuove metodologie sia eseguita da ricercatori indipendenti, assicurando così un'analisi obiettiva e imparziale.

**ESERCIZI**

### ****3.1. Cosa sono misura, misurazione e metrica e come sono correlate?****

* **Misurazione:** La misurazione è il processo di assegnazione di numeri o simboli agli attributi di entità nel mondo reale secondo regole chiaramente definite. È l'atto di quantificare le caratteristiche di un oggetto o di un processo.
* **Misura:** Una misura è il valore numerico specifico o il simbolo che viene assegnato a un attributo come risultato del processo di misurazione. Caratterizza l'attributo dell'entità osservata.
* **Metrica:** Il termine "metrica" è utilizzato in due contesti nell'ingegneria del software. Può riferirsi al campo della misurazione stesso o a una misura specifica, come "linee di codice" (LOC), che è una metrica utilizzata per quantificare la dimensione di un programma software.

**Relazione:** Le metriche sono tipi specifici di misure che quantificano un attributo, e la misurazione è il processo di ottenimento di queste misure. In sintesi, la misurazione è il processo, la misura è il risultato, e la metrica è il tipo specifico di misura utilizzato per valutare determinati attributi.

### ****3.2. Quali sono i quattro principali tipi di scale di misurazione?****

I quattro principali tipi di scale di misurazione sono:

1. **Scala Nominale:** La forma più semplice di scala di misurazione che categorizza i dati senza alcun ordine o classificazione. Viene utilizzata per scopi di classificazione (es. tipi di difetti, linguaggi di programmazione).
2. **Scala Ordinale:** Questa scala fornisce una classifica o un ordine tra le entità in base a un particolare attributo, ma le differenze tra i ranghi non sono quantificate (es. sistemi di valutazione, classifiche).
3. **Scala a Intervalli:** Questa scala consente di misurare le differenze tra i valori, ma non ha un vero punto zero, il che significa che i rapporti non sono significativi (es. temperatura in gradi Celsius o Fahrenheit).
4. **Scala di Rapporto:** Il tipo di scala più potente, che ha un vero punto zero e consente la misurazione di quantità assolute, permettendo confronti significativi tra i rapporti (es. lunghezza, peso, durata del tempo).

### ****3.3. Quali sono le differenze tra una misura diretta e una misura indiretta?****

* **Misura Diretta:** Una misura diretta quantifica un attributo direttamente senza bisogno di misurare altri attributi. La misurazione è semplice e non dipende da ulteriori dati o calcoli (es. linee di codice, numero di difetti trovati).
* **Misura Indiretta:** Una misura indiretta viene derivata da una o più misure dirette. Comporta la combinazione o la manipolazione di altri attributi misurati per produrre un valore per un attributo che non può essere misurato direttamente (es. densità dei difetti, produttività dei programmatori).

**Differenza:** La differenza principale è che le misure dirette forniscono una quantificazione diretta di un attributo, mentre le misure indirette sono derivate elaborando o combinando altre misurazioni dirette.

### ****3.4. In quali tre classi sono suddivise le misurazioni nell'ingegneria del software?****

Le misurazioni nell'ingegneria del software sono suddivise in tre classi:

1. **Processo:** Queste misurazioni si riferiscono alle attività coinvolte nella produzione del software (es. sforzo richiesto, tempo impiegato per i test).
2. **Prodotto:** Queste misurazioni si riferiscono agli artefatti o ai deliverable prodotti dal processo software (es. linee di codice, affidabilità del software).
3. **Risorse:** Queste misurazioni si riferiscono agli input necessari per il processo software, inclusi risorse umane, hardware e software (es. produttività dei programmatori, utilizzo dell'hardware).

### ****3.5. Cosa sono gli attributi interni ed esterni e come sono correlati principalmente alle misure dirette e indirette?****

* **Attributi Interni:** Questi sono caratteristiche di un oggetto che possono essere misurate direttamente all'interno dell'oggetto stesso, senza riferimento ad altre entità. Sono tipicamente associati a misure dirette (es. numero di linee di codice in un prodotto software).
* **Attributi Esterni:** Questi sono caratteristiche che possono essere misurate solo considerando la relazione dell'oggetto con altri oggetti o il suo effetto su altre entità. Sono generalmente misurati indirettamente (es. affidabilità del software, che può dipendere dalla frequenza dei difetti rispetto agli scenari d'uso).

**Relazione:** Gli attributi interni sono tipicamente correlati a misure dirette perché possono essere quantificati direttamente senza riferimento esterno. Gli attributi esterni, al contrario, sono più spesso correlati a misure indirette perché richiedono confronto, contesto o derivazione da altre misurazioni.

### ****Capitolo 4: Revisioni Sistematiche della Letteratura****

Le revisioni sistematiche della letteratura (Systematic Literature Reviews, SLR) vengono condotte per "identificare, analizzare e interpretare tutte le evidenze disponibili relative a una specifica domanda di ricerca" [96]. L'obiettivo di queste revisioni è fornire un quadro completo, esaustivo e valido delle evidenze esistenti. Per raggiungere questo obiettivo, l'identificazione, l'analisi e l'interpretazione devono essere eseguite in modo scientifico e rigoroso. Kitchenham e Charters hanno adattato linee guida per le revisioni sistematiche della letteratura, ispirandosi principalmente dalla medicina, le hanno valutate [24] e aggiornate di conseguenza [96]. Queste linee guida, strutturate secondo un processo in tre fasi per pianificare, condurre e riportare la revisione, sono riassunte di seguito.

### ****4.1 Pianificazione della Revisione****

La pianificazione di una revisione sistematica della letteratura include diverse azioni fondamentali:

#### **Identificazione della necessità di una revisione**

La necessità di una revisione sistematica nasce da un ricercatore che vuole comprendere lo stato dell'arte in un'area specifica, o da professionisti che desiderano utilizzare evidenze empiriche per le loro decisioni strategiche o attività di miglioramento. Se esistono già revisioni sistematiche della letteratura nel campo d'interesse, queste dovrebbero essere valutate per quanto riguarda il loro scopo e qualità, per determinare se sono sufficienti a soddisfare le esigenze attuali di una revisione. Una revisione sistematica della letteratura può essere vista come un metodo di ricerca per eseguire una revisione della letteratura.

#### **Definizione delle domande di ricerca**

L'area della revisione sistematica e le specifiche domande di ricerca stabiliscono il focus per l'identificazione degli studi primari, l'estrazione dei dati dagli studi e l'analisi. Pertanto, le domande di ricerca devono essere ben ponderate e formulate con cura. Gli aspetti da considerare nella formulazione delle domande di ricerca includono [96]:

* **Popolazione:** Qual è il gruppo di persone, programmi o aziende di interesse per la revisione?
* **Intervento:** Quale tecnologia, strumento o procedura viene studiato nell'intervento applicato nello studio empirico?
* **Confronto:** Qual è il trattamento di controllo con cui l'intervento viene confrontato? In particolare, è fondamentale definire un intervento "placebo", poiché "non utilizzare l'intervento" spesso non è una scelta valida in ingegneria del software.
* **Risultati:** I risultati dell'esperimento non dovrebbero essere solo statisticamente significativi, ma anche rilevanti dal punto di vista pratico. Ad esempio, potrebbe non essere interessante un risultato migliore del 10% in un certo aspetto se richiede il doppio del tempo.
* **Contesto:** Deve essere definito il contesto dello studio, che rappresenta una visione estesa della popolazione, includendo se è stato condotto in ambito accademico o industriale, in quale segmento industriale, e anche gli incentivi per i soggetti coinvolti [78, 132].
* **Progetti sperimentali:** È necessario definire i disegni sperimentali da includere nella domanda di ricerca.

Staples e Niazi raccomandano che l'ambito di una revisione sistematica della letteratura sia limitato da domande di ricerca chiare e ristrette per evitare studi ingestibili [166].

#### **Sviluppo di un protocollo di revisione**

Il protocollo di revisione definisce le procedure per la revisione sistematica della letteratura e funge anche da registro per la conduzione della revisione stessa. Pertanto, è un documento "vivente" di importanza sia per la conduzione pratica della revisione, sia per la sua validità. Kitchenham e Charters propongono che il protocollo di revisione copra i seguenti elementi [96]:

* Contesto e motivazione
* Domande di ricerca
* Strategia di ricerca per gli studi primari
* Criteri di selezione degli studi
* Procedure di selezione degli studi
* Checklist e procedure per la valutazione della qualità degli studi
* Strategia di estrazione dei dati
* Sintesi dei dati estratti
* Strategia di diffusione dei risultati
* Cronoprogramma del progetto

Il protocollo dovrebbe preferibilmente essere rivisto da pari per garantirne la coerenza e la validità. L'esperienza nelle revisioni sistematiche della letteratura sottolinea l'importanza di uno studio preliminare per aiutare a delimitare le domande di ricerca, oltre a mantenere una certa flessibilità nel modificare le domande di ricerca durante lo sviluppo del protocollo, man mano che il problema oggetto di studio diventa più chiaro [24].

### ****4.2 Conduzione della Revisione****

Condurre la revisione significa mettere in pratica il protocollo di revisione. Questa fase include diverse attività cruciali:

#### **Identificazione della Ricerca**

L'attività principale in questo passaggio consiste nello specificare le stringhe di ricerca e applicarle ai database. Tuttavia, include anche ricerche manuali su riviste e atti di conferenze, nonché la consultazione di siti web di ricercatori o l'invio di domande a questi ultimi. La ricerca sistematica di studi primari basata su riferimenti a e da altri studi è chiamata "snowballing" (effetto valanga) [145].

La strategia di ricerca è un compromesso tra trovare tutti gli studi primari rilevanti e non ottenere un numero schiacciante di falsi positivi, che devono essere esclusi manualmente [43]. Un falso positivo è un risultato che appare positivamente rilevante, ma che si rivela non esserlo e quindi deve essere rimosso. La stringa di ricerca viene sviluppata in base all'area da coprire e alle domande di ricerca. Utilizzare più database è una necessità per coprire tutta la letteratura rilevante, ma questo genera anche duplicati che devono essere identificati e rimossi. Alla fine, bisogna accettare che gli articoli trovati siano un campione della popolazione di tutti gli articoli su un argomento specifico. La questione chiave è che il campione rappresenti effettivamente la popolazione di interesse.

Gli studi primari pubblicati tendono ad avere un bias di pubblicazione, il che significa che (in un certo senso) i risultati positivi hanno maggiori probabilità di essere pubblicati rispetto ai risultati negativi. Pertanto, anche la letteratura grigia, come rapporti tecnici, tesi, pubblicazioni rifiutate e lavori in corso, dovrebbe essere ricercata [96].

I risultati della ricerca e un registro delle azioni intraprese dovrebbero essere memorizzati, preferibilmente utilizzando un sistema di gestione delle referenze.

#### **Selezione degli Studi Primari**

La base per la selezione degli studi primari sono i criteri di inclusione ed esclusione. Questi criteri dovrebbero essere sviluppati in anticipo per evitare bias. Tuttavia, potrebbero dover essere aggiustati durante il corso della selezione, poiché tutti gli aspetti di inclusione ed esclusione non sono evidenti nella fase di pianificazione.

Il set identificato di studi candidati viene processato in relazione ai criteri di selezione. Per alcuni studi, è sufficiente leggere il titolo o l'abstract per giudicare l'articolo, mentre altri articoli richiedono un'analisi più approfondita, ad esempio, della metodologia o delle conclusioni per determinare il loro stato. Gli abstract strutturati [30] possono aiutare nel processo di selezione.

Poiché il processo di selezione è una questione di giudizio, anche con criteri di selezione ben definiti, è consigliabile che due o più ricercatori valutino ogni articolo, o almeno un campione casuale degli articoli. Quindi, l'accordo tra valutatori può essere misurato utilizzando la statistica Cohen Kappa [36] e riportato come parte della valutazione della qualità della revisione sistematica della letteratura. Tuttavia, va notato che è possibile ottenere una statistica Cohen Kappa relativamente alta a causa del fatto che molti articoli trovati nella ricerca automatica sono facilmente esclusi dai ricercatori durante la valutazione manuale. Pertanto, può essere importante condurre la valutazione in più fasi, ad esempio iniziando rimuovendo quegli articoli che sono ovviamente non pertinenti, anche se trovati nella ricerca.

#### **Valutazione della Qualità degli Studi**

Valutare la qualità degli studi primari è importante, soprattutto quando gli studi riportano risultati contraddittori. La qualità degli studi primari può essere utilizzata per analizzare la causa dei risultati contrastanti o per ponderare l'importanza dei singoli studi durante la sintesi dei risultati.

Non esiste una definizione universalmente accettata e applicabile di "qualità dello studio". I tentativi di mappare i criteri di qualità dalla medicina non si sono adattati all'intervallo di qualità degli studi di ingegneria del software [47].

Il mezzo più pratico per la valutazione della qualità sono le checklist, anche se il loro supporto empirico può essere debole. Uno studio di Kitchenham et al. ha anche dimostrato che sono necessari almeno tre revisori per effettuare una valutazione valida [105]. Le checklist utilizzate nella valutazione della qualità degli studi empirici sono disponibili nella letteratura sull'ingegneria del software empirica [96, 105, 145].

La valutazione della qualità può portare all'esclusione di alcuni studi primari, se la qualità dello studio fa parte dei criteri di selezione. È inoltre importante notare che la qualità degli studi primari dovrebbe essere valutata, non la qualità del reporting. Tuttavia, è spesso difficile giudicare la qualità di uno studio se viene riportato in modo carente. Potrebbe essere necessario contattare gli autori per trovare o chiarire informazioni mancanti nei report.

#### **Estrazione e Monitoraggio dei Dati**

Una volta deciso l'elenco degli studi primari, i dati degli studi primari vengono estratti. Un modulo di estrazione dati è progettato per raccogliere le informazioni necessarie dai report degli studi primari. Se i dati della valutazione della qualità vengono utilizzati per la selezione degli studi, il modulo di estrazione è suddiviso in due parti: una per i dati di qualità, che viene compilata durante la valutazione della qualità, e una per i dati dello studio, da compilare durante l'estrazione dei dati.

Il modulo di estrazione dati è progettato in base alle domande di ricerca. Per la sintesi puramente meta-analitica, i dati sono un insieme di valori numerici, che rappresentano il numero di soggetti, le caratteristiche degli oggetti, gli effetti del trattamento, gli intervalli di confidenza, ecc. Per insiemi meno omogenei di studi, devono essere incluse descrizioni più qualitative degli studi primari. Oltre ai dati grezzi, per ogni studio primario vengono registrati il nome del revisore, la data di estrazione dei dati e i dettagli di pubblicazione.

Il modulo di estrazione dati dovrebbe essere testato prima di essere applicato all'intero set di studi primari. Se possibile, l'estrazione dei dati dovrebbe essere eseguita indipendentemente da due ricercatori, almeno per un campione di studi, al fine di valutare la qualità della procedura di estrazione.

Se uno studio primario è pubblicato in più di un articolo, ad esempio, se un articolo di conferenza viene esteso a una versione di rivista, dovrebbe essere conteggiato solo un caso come studio primario. Di solito, si preferisce la versione della rivista, poiché è la più completa, ma entrambe le versioni possono essere utilizzate nell'estrazione dei dati. I rapporti tecnici di supporto o le comunicazioni con gli autori possono anche servire come fonti di dati per l'estrazione.

#### **Sintesi dei Dati**

La forma più avanzata di sintesi dei dati è la meta-analisi. Questa si riferisce a metodi statistici applicati per analizzare i risultati di diversi studi indipendenti. La meta-analisi presuppone che gli studi sintetizzati siano omogenei, o che la causa dell'eterogeneità sia ben conosciuta [135]. Una meta-analisi confronta le dimensioni dell'effetto e i valori p per valutare il risultato sintetizzato. È principalmente applicabile agli esperimenti replicati, se presenti, a causa del requisito di omogeneità.

In sintesi, gli studi da includere in una meta-analisi devono [135]:

* Essere dello stesso tipo, ad esempio, esperimenti formali
* Avere la stessa ipotesi di test
* Avere le stesse misure dei costrutti di trattamento ed effetto
* Riportare gli stessi fattori esplicativi

Le procedure di meta-analisi coinvolgono tre passaggi principali [135]:

1. Decidere quali studi includere nella meta-analisi.
2. Estrarre la dimensione dell'effetto dal report dello studio primario o stimarla se non è pubblicata.
3. Combinare le dimensioni dell'effetto degli studi primari per stimare e testare l'effetto combinato.

La meta-analisi è una tecnica utilizzata per sintetizzare i risultati di diversi studi indipendenti che hanno esaminato lo stesso fenomeno o hanno testato la stessa ipotesi. Questa tecnica è particolarmente utile per ottenere una visione d'insieme e per trarre conclusioni più solide dai dati disponibili. Le procedure della meta-analisi si articolano in tre passaggi principali:

### 1. ****Decidere quali studi includere nella meta-analisi****

* **Scopo:** Identificare quali studi sono rilevanti e appropriati per essere inclusi nella meta-analisi.
* **Criteri di inclusione:** Si devono definire chiaramente i criteri per selezionare gli studi, basati su vari fattori come il tipo di studio (ad esempio, esperimenti controllati), l'ipotesi testata, le misure utilizzate, e la qualità dello studio.
* **Esclusione:** Studi che non soddisfano i criteri prestabiliti vengono esclusi. Questo passaggio è cruciale per assicurarsi che la meta-analisi confronti studi simili, riducendo la variabilità che potrebbe confondere i risultati.

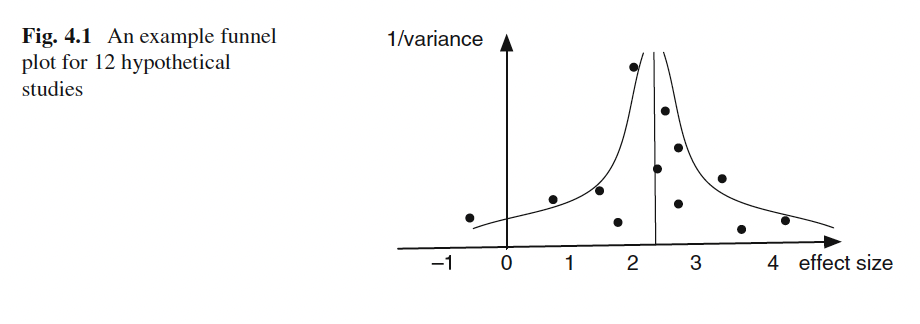
### 2. ****Estrarre la dimensione dell'effetto dal report dello studio primario o stimarla se non è pubblicata****

* **Dimensione dell'effetto:** La dimensione dell'effetto (effect size) è una misura quantitativa della forza di un fenomeno osservato. Esempi comuni di dimensioni dell'effetto includono la differenza media standardizzata, il rapporto di rischio, e l'odds ratio.
* **Estrazione:** Per ciascuno studio selezionato, si estraggono le informazioni necessarie per calcolare la dimensione dell'effetto. Se la dimensione dell'effetto non è riportata direttamente nello studio, può essere necessario calcolarla a partire dai dati forniti (come medie, deviazioni standard, o tassi di incidenza).
* **Uniformità:** Tutte le dimensioni dell'effetto devono essere calcolate o convertite in un formato uniforme per poter essere confrontate e combinate.

### 3. ****Combinare le dimensioni dell'effetto degli studi primari per stimare e testare l'effetto combinato****

* **Combinazione:** Le dimensioni dell'effetto estratte dagli studi individuali vengono combinate per ottenere una dimensione dell'effetto aggregata. Questo processo implica l'uso di tecniche statistiche che possono variare a seconda che gli studi siano omogenei (cioè simili) o eterogenei (cioè diversi).
* **Modello a effetti fissi o casuali:**
  + **Effetti fissi:** Si utilizza quando si assume che tutti gli studi misurino lo stesso effetto.
  + **Effetti casuali:** Si utilizza quando si assume che gli studi misurino effetti diversi ma correlati.
* **Stima dell'effetto combinato:** Questo passaggio produce una stima generale della dimensione dell'effetto, che rappresenta la media ponderata delle dimensioni dell'effetto degli studi individuali. Si può anche testare se l'effetto combinato è statisticamente significativo.
* **Analisi della variabilità:** Se esiste una significativa eterogeneità tra gli studi (cioè se le dimensioni dell'effetto variano molto tra uno studio e l'altro), questo può essere esplorato per capire quali fattori influenzano tali variazioni.

Oltre alle procedure di selezione degli studi primari presentate sopra, la meta-analisi dovrebbe includere un'analisi del bias di pubblicazione. Tali metodi includono il funnel plot, in cui le dimensioni dell'effetto osservate vengono tracciate rispetto a una misura della dimensione dello studio, ad esempio l'inverso della varianza o un'altra misura di dispersione. I punti dati dovrebbero disperdersi attorno a un modello a "imbuto" se il set di studi primari è completo. Le lacune nell'imbuto indicano che alcuni studi non sono stati pubblicati o trovati [135].



La dimensione dell'effetto è un indicatore indipendente dall'unità o dalla scala utilizzata in ciascuno degli studi primari. Dipende dal tipo di studio, ma potrebbe tipicamente essere la differenza tra i valori medi di ciascun trattamento. Questa misura deve essere normalizzata per consentire confronti con altre scale, ossia divisa per la deviazione standard combinata [135].

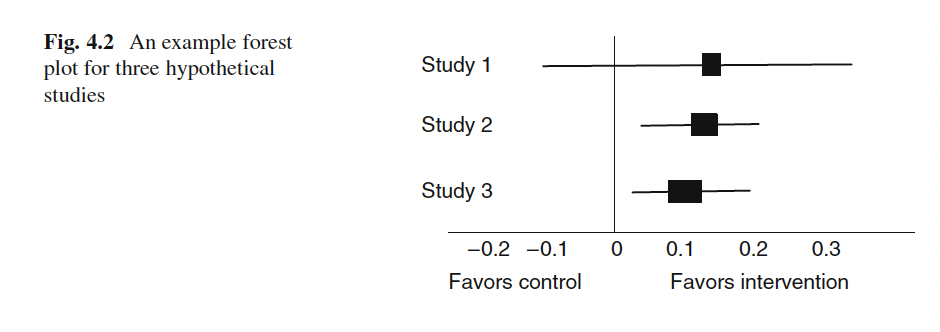
L'analisi presuppone omogeneità tra gli studi e viene quindi eseguita con un modello a effetti fissi. La meta-analisi stima la dimensione dell'effetto reale calcolando un valore medio delle dimensioni dell'effetto degli studi individuali, che sono medie a loro volta. Esistono test per identificare l'eterogeneità, come il test Q e il test del Rapporto di Verosimiglianza, che dovrebbero essere applicati per garantire che le condizioni del modello siano soddisfatte [135].

Per i dati eterogenei, esiste un modello a effetti casuali, che consente variabilità dovuta a un fattore sconosciuto, che influenza le dimensioni dell'effetto per gli studi primari. Questo modello fornisce stime sia per l'errore di campionamento, come il modello a effetti fissi, sia per la variabilità nelle sottopopolazioni eterogenee.

Metodi meno formali per la sintesi dei dati includono la sintesi descrittiva o narrativa. Questi metodi tabulano i dati dagli studi primari in un modo che illumina la domanda di ricerca. Come requisito minimo sui dati tabulati, Kitchenham e Charters propongono che vengano presentati i seguenti elementi [96]:

* Dimensione del campione per ciascun intervento
* Stime delle dimensioni dell'effetto per ciascun intervento con errori standard per ciascun effetto
* Differenza tra i valori medi per ciascun intervento e l'intervallo di confidenza per la differenza
* Unità utilizzate per misurare l'effetto

I risultati statistici possono essere visualizzati utilizzando forest plots. Un forest plot presenta le medie e le varianze della differenza tra trattamenti per ciascuno studio.



Sintetizzare studi eterogenei e studi a metodi misti richiede approcci qualitativi. Cruzes e Dyba [39] hanno esaminato studi secondari in ingegneria del software, che includevano la sintesi delle evidenze empiriche. Hanno identificato diversi metodi di sintesi, molti dalla medicina, di cui sette metodi sono stati utilizzati in ingegneria del software. Questi metodi sono brevemente introdotti di seguito. Per maggiori dettagli, fare riferimento a Cruzes e Dyba [39] e alle relative referenze.

* **Analisi Tematica:** Un metodo che mira a identificare, analizzare e riportare modelli o temi negli studi primari. Al minimo, organizza e presenta i dati in modo dettagliato e interpreta vari aspetti del tema sotto studio.
* **Sintesi Narrativa:** Racconta una "storia" che origina dalle evidenze primarie. Le evidenze grezze e le interpretazioni vengono strutturate, utilizzando ad esempio la tabulazione dei dati, raggruppamenti e cluster, o il conteggio dei voti come strumento descrittivo. La sintesi narrativa può essere applicata a studi con dati qualitativi o quantitativi, o combinazioni di essi.
* **Analisi Comparativa:** Un metodo mirato ad analizzare connessioni causali complesse. Utilizza la logica booleana per spiegare le relazioni tra causa ed effetto negli studi primari. L'analisi elenca condizioni necessarie e sufficienti in ciascuno degli studi primari e trae conclusioni dalla presenza/assenza di variabili indipendenti in ciascuno degli studi.
* **Metodo del Survey dei Casi:** Originariamente definito per studi di casi, ma può essere applicato anche a esperimenti eterogenei. Aggrega la ricerca esistente applicando uno strumento di survey di domande specifiche a ciascuno studio primario. I dati dal survey sono quantitativi, e quindi l'aggregazione viene eseguita utilizzando metodi statistici.
* **Meta-Etnografia:** Traduce studi l'uno nell'altro e sintetizza le traduzioni in concetti che vanno oltre i singoli studi. Le interpretazioni e le spiegazioni negli studi primari sono trattate come dati nello studio meta-etnografico.
* **Meta-Analisi:** Come menzionato sopra, basata su metodi statistici per integrare dati quantitativi da diversi casi.
* **Analisi del Campo:** Mira a fornire una panoramica della ricerca in un campo, piuttosto che sintetizzare i risultati della ricerca. Queste analisi sono anche chiamate studi di mappatura.

Indipendentemente dal metodo di sintesi, dovrebbe essere condotta un'analisi della sensibilità per analizzare se i risultati sono coerenti tra diversi sottoinsiemi di studi.

### ****4.3 Reporting the Review****

Quando si scrive un report su una revisione sistematica della letteratura, è fondamentale considerare il pubblico di riferimento, poiché la forma e il contenuto del report devono essere adattati alle esigenze di chi lo leggerà. In particolare, se l'obiettivo della revisione è influenzare i praticanti, il formato del report deve essere pensato per essere accessibile e utile a loro.

Kitchenham e Charters suggeriscono diversi modi per diffondere i risultati della revisione sistematica, a seconda del pubblico:

1. **Riviste e riviste orientate ai praticanti:** Pubblicare articoli che siano facilmente comprensibili e direttamente applicabili ai professionisti nel campo.
2. **Comunicati stampa per la stampa popolare e specializzata:** Utilizzare i media per raggiungere un pubblico più ampio, inclusi coloro che potrebbero non essere direttamente coinvolti nella ricerca accademica.
3. **Opuscoli di sintesi:** Creare brevi documenti riassuntivi che forniscano una panoramica dei risultati principali della revisione.
4. **Poster:** Utilizzare poster per comunicare i risultati in conferenze o altri eventi pubblici.
5. **Pagine web:** Creare siti o pagine web che presentino i risultati in modo accessibile e interattivo.
6. **Comunicazione diretta con le entità interessate:** Inviare i risultati direttamente ai gruppi o alle organizzazioni che potrebbero beneficiare delle scoperte.

Per il pubblico accademico, è essenziale un report dettagliato delle procedure adottate nello studio. Questo permette ad altri ricercatori di valutare la qualità della revisione sistematica. Il report dovrebbe includere eventuali modifiche al protocollo di studio, elenchi completi degli studi primari inclusi ed esclusi, i dati relativi alla loro classificazione, nonché i dati grezzi derivati da ciascuno degli studi primari. Se il report principale non può includere tutti i dettagli a causa di limitazioni di spazio, è consigliabile pubblicare un rapporto tecnico di supporto online.

### ****4.4 Mapping Studies****

Se la domanda di ricerca per la revisione della letteratura è più ampia, o se il campo di studio è meno esplorato, si può optare per uno **studio di mappatura** anziché per una revisione sistematica della letteratura. Uno studio di mappatura, a volte chiamato **scoping study**, esplora un campo più ampio di ricerca per ottenere una panoramica dello stato dell'arte o della pratica su un determinato argomento.

Uno studio di mappatura segue un processo simile a quello delle revisioni sistematiche della letteratura, ma con criteri di inclusione/esclusione e qualità diversi. A causa del suo scopo più ampio e del tipo variabile di studi inclusi, i dati raccolti e la sintesi tendono ad essere più qualitativi rispetto a quelli delle revisioni sistematiche. Tuttavia, per garantire che uno studio di mappatura sia utile e rilevante, è importante che l'analisi vada oltre le semplici statistiche descrittive, collegando le tendenze e le osservazioni ai bisogni del mondo reale.

Kitchenham et al. hanno fornito una sintesi delle principali caratteristiche degli studi di mappatura rispetto alle revisioni sistematiche della letteratura, evidenziando le differenze nelle procedure e nei risultati attesi.

### ****4.5 Example Reviews****

Kitchenham et al. riportano che tra il 2004 e il 2008 sono state pubblicate 53 revisioni sistematiche della letteratura uniche nel campo dell'ingegneria del software. Essi concludono che il numero di revisioni sistematiche pubblicate è in crescita, e che la qualità delle revisioni tende ad aumentare. Tuttavia, c'è ancora una grande variabilità tra coloro che sono consapevoli e utilizzano linee guida sistematiche per la conduzione delle revisioni e coloro che non si riferiscono a nessuna linea guida.

Ad esempio, Sjøberg et al. hanno esaminato gli studi sperimentali condotti nell'ingegneria del software tra il 1993 e il 2002, trovando che solo l'1,9% degli articoli presentava esperimenti. Questi esperimenti erano concentrati principalmente in due categorie: il ciclo di vita del software/ingegneria e metodi/tecniche, a causa del numero relativamente elevato di esperimenti sulle tecniche di ispezione e sul design orientato agli oggetti.

Hannay et al. hanno condotto una meta-analisi sull'efficacia del pair programming, analizzando dati provenienti da 18 studi primari e presentando le loro scoperte tramite forest plot, una tecnica di visualizzazione dei dati che mostra le differenze nei risultati tra i vari studi.

| **Elementi della SLR** | **Studio di mappatura sistematica** | **Revisione sistematica della letteratura** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| **Obiettivi** | Classificazione e analisi tematica della letteratura su un argomento di ingegneria del software | Identificazione delle migliori pratiche riguardo a procedure, tecnologie, metodi o strumenti specifici attraverso l'aggregazione delle informazioni provenienti da studi comparativi |

|  |  |  |
| --- | --- | --- |
| **Domanda di ricerca** | Generica – legata alle tendenze della ricerca. Del tipo: quali ricercatori, quanta attività, che tipo di studi, ecc. | Specifica – legata ai risultati degli studi empirici. Del tipo: la tecnologia/metodo A è migliore o peggiore di B? |

|  |  |  |
| --- | --- | --- |
| **Processo di ricerca** | Definito dall'area tematica | Definito dalla domanda di ricerca che identifica le tecnologie specifiche investigate |

|  |  |  |
| --- | --- | --- |
| **Ambito** | Ampio – tutti i documenti relativi a un'area tematica sono inclusi, ma vengono raccolti solo dati di classificazione su questi | Focalizzato – solo documenti empirici relativi a una specifica domanda di ricerca sono inclusi e de1ttagliate informazioni sui risultati della ricerca sono estratte da ciascun documento |

|  |  |  |
| --- | --- | --- |
| **Requisiti della strategia di ricerca** | Spesso meno stringente se sono di interesse solo le tendenze della ricerca; ad esempio, gli autori possono cercare solo un set mirato di pubblicazioni, limitarsi agli articoli di riviste o limitarsi a una o due biblioteche digitali | Estremamente stringente – tutti gli studi rilevanti devono essere trovati. Solitamente, i team di revisione sistematica della letteratura devono usare tecniche oltre la semplice ricerca nelle fonti dati, come consultare le referenze negli studi primari identificati e/o contattare i ricercatori nel campo per scoprire se stanno intraprendendo nuove ricerche nell'area |

|  |  |  |
| --- | --- | --- |
| **Valutazione della qualità** | Non essenziale. Anche complicata dalla natura inclusiva della ricerca che può includere studi teorici così come studi empirici di tutti i tipi, rendendo complicata la valutazione della qualità degli studi primari | Importante per garantire che i risultati siano basati su prove di alta qualità |

|  |  |  |
| --- | --- | --- |
| **Risultati** | Un insieme di documenti relativi a un'area tematica categorizzati in una varietà di dimensioni e conteggi del numero di documenti in varie categorie | I risultati degli studi primari sono aggregati per rispondere alla specifica domanda di ricerca, possibilmente con qualificatori (ad esempio, i risultati si applicano solo ai principianti) |

**ESERCIZI**

### ****4.1. Qual è la differenza tra una revisione sistematica della letteratura e una revisione della letteratura più generale?****

Una **revisione sistematica della letteratura (SLR)** differisce da una revisione della letteratura generale per il suo approccio strutturato e rigoroso. Una SLR mira a **identificare, analizzare e interpretare tutte le prove disponibili** relative a una specifica domanda di ricerca in modo **completo e imparziale**. Segue un protocollo ben definito, con passaggi specifici come pianificazione, conduzione e redazione del report, per garantire l'affidabilità e la validità del processo di revisione.

Al contrario, una **revisione della letteratura generale** è solitamente **più narrativa e meno strutturata**. Potrebbe non seguire un protocollo predefinito, e la selezione della letteratura potrebbe essere soggettiva, con il rischio di introdurre bias. Le revisioni della letteratura generale forniscono spesso una panoramica di un campo, sintetizzando le conoscenze esistenti senza il rigore sistematico applicato nelle SLR.

### ****4.2. Quali strategie di ricerca esistono per gli studi primari?****

Esistono diverse strategie di ricerca per identificare gli studi primari in una revisione sistematica della letteratura:

1. **Ricerche nei database:** Applicare stringhe di ricerca a database elettronici (ad esempio, IEEE Xplore, Scopus, PubMed) per recuperare studi rilevanti.
2. **Ricerche manuali:** Cercare manualmente in riviste chiave, atti di conferenze e altre pubblicazioni pertinenti.
3. **Snowballing:** Cercare studi sistematicamente seguendo le referenze di uno studio (snowballing all'indietro) o guardando agli studi più recenti che citano lo studio originale (snowballing in avanti).
4. **Letteratura grigia:** Cercare fonti non peer-reviewed come rapporti tecnici, tesi e lavori non pubblicati per ridurre il bias di pubblicazione.
5. **Siti web di ricercatori:** Visitare i siti web dei ricercatori chiave nel campo o contattarli direttamente per studi rilevanti non pubblicati o in corso.

### ****4.3. Perché due ricercatori dovrebbero eseguire alcuni degli stessi passaggi in una revisione sistematica della letteratura?****

È importante che **due o più ricercatori** eseguano alcuni degli stessi passaggi in una SLR per garantire **affidabilità e ridurre il bias**. Quando più ricercatori valutano gli studi o eseguono l'estrazione dei dati:

1. **Affidabilità tra valutatori:** L'accordo tra i ricercatori può essere misurato (ad esempio, utilizzando la statistica Kappa di Cohen) per valutare la coerenza e l'obiettività del processo di selezione o estrazione dei dati.
2. **Riduzione del bias:** Riduce il rischio che i bias personali influenzino l'inclusione/esclusione degli studi o l'interpretazione dei dati.
3. **Validità:** Aumenta la validità e la credibilità della revisione, poiché eventuali disaccordi possono essere discussi e risolti, portando a conclusioni più solide.

### ****4.4. Quali requisiti devono soddisfare gli studi primari per essere inclusi in una meta-analisi?****

Per essere inclusi in una **meta-analisi**, uno studio primario deve generalmente soddisfare i seguenti requisiti:

1. **Omogeneità:** Gli studi devono essere simili per tipo, ad esempio tutti esperimenti formali.
2. **Ipotesi coerente:** Gli studi devono testare la stessa ipotesi o domanda di ricerca.
3. **Misure comparabili:** Gli studi devono utilizzare le stesse o compatibili misure dei costrutti di trattamento ed effetto.
4. **Report delle dimensioni dell'effetto:** Idealmente, gli studi primari dovrebbero riportare le dimensioni dell'effetto, o queste dovrebbero essere stimabili dai dati forniti, per consentire la sintesi quantitativa.
5. **Fattori esplicativi:** Gli studi devono riportare gli stessi o simili fattori esplicativi, permettendo il confronto tra gli studi.

### ****4.5. Quali sono le principali differenze tra uno studio di revisione sistematica e uno studio di mappatura?****

Le principali differenze tra una **SLR** e uno **studio di mappatura** sono:

1. **Scopo:**
   * **SLR:** Si concentra sul rispondere a specifiche domande di ricerca ristrette identificando, analizzando e sintetizzando tutti gli studi rilevanti.
   * **Studio di mappatura:** Fornisce una panoramica del panorama della ricerca in un'area più ampia, identificando lacune, tendenze e cluster di ricerca.
2. **Sintesi dei dati:**
   * **SLR:** Generalmente coinvolge una sintesi dettagliata, che può includere meta-analisi o sintesi qualitativa.
   * **Studio di mappatura:** Tende a essere più descrittivo, spesso utilizzando visualizzazioni come mappe o grafici per mostrare tendenze e distribuzioni.
3. **Criteri di inclusione:**
   * **SLR:** Utilizza criteri di inclusione/esclusione rigorosi per garantire la rilevanza e la qualità degli studi.
   * **Studio di mappatura:** Può avere criteri più ampi, includendo una varietà più ampia di tipi di studi, per catturare una visione completa del campo di ricerca.
4. **Output:**
   * **SLR:** Mira a fornire risposte definitive a specifiche domande di ricerca, spesso con implicazioni pratiche.
   * **Studio di mappatura:** Mira a fornire una mappa del campo di ricerca, mostrando ciò che è stato studiato e identificando aree che necessitano di ulteriori ricerche.

The Unfairness of Popularity Bias in Recommendation

Il documento si concentra sul problema del **bias di popolarità** nei sistemi di raccomandazione, un fenomeno in cui gli articoli popolari (quelli che ricevono più valutazioni) tendono a essere raccomandati molto più frequentemente rispetto a quelli meno popolari, lasciando questi ultimi sottorappresentati. Tradizionalmente, la ricerca si è concentrata su come affrontare questo problema aumentando il numero di articoli meno popolari raccomandati o migliorando la copertura complessiva del catalogo. Tuttavia, questo studio adotta una prospettiva diversa, focalizzandosi sull'esperienza degli utenti.

### ****Obiettivi della Ricerca****

Il lavoro si propone di esplorare come il bias di popolarità influisca sulle raccomandazioni ricevute dagli utenti rispetto alle loro aspettative. Viene proposta una suddivisione degli utenti in tre gruppi, basati sul loro interesse verso gli articoli popolari:

* **Niche**: Utenti con un basso interesse per gli articoli popolari.
* **Diverse**: Utenti con un interesse equilibrato tra articoli popolari e non.
* **Blockbuster-focused**: Utenti con un alto interesse per gli articoli popolari.

### ****Risultati****

L'analisi empirica, condotta su un dataset di film (MovieLens 1M), rivela che molti algoritmi di raccomandazione tendono a concentrarsi eccessivamente sugli articoli popolari, anche quando gli utenti mostrano interesse per articoli meno popolari. Questo porta a una **disparità di bias estrema**, dove le raccomandazioni non rispecchiano le reali preferenze degli utenti, soprattutto per quelli del gruppo "Niche", che sono i più penalizzati.

### ****Implicazioni****

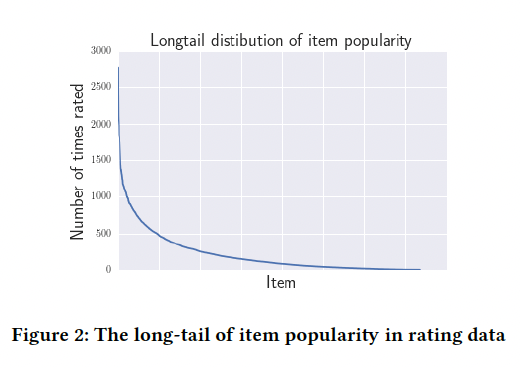
Gli utenti con un minore interesse per gli articoli popolari, spesso più attivi nel sistema (hanno valutato più articoli), sono identificati come stakeholder importanti. Tuttavia, i sistemi di raccomandazione non soddisfano adeguatamente le loro aspettative. Questo suggerisce la necessità di sviluppare algoritmi di raccomandazione che considerino meglio le preferenze individuali, evitando di sovraesporre articoli popolari a scapito della varietà e dell'innovazione.

### ****Domande di Ricerca****

Il documento cerca di rispondere a due domande principali:

1. **RQ1**: Quanto sono interessati gli individui o i gruppi di utenti agli articoli popolari?
2. **RQ2**: Come il bias di popolarità degli algoritmi impatta utenti con diversi livelli di interesse per gli articoli popolari?

L'obiettivo finale è capire come questi bias influenzano le esperienze degli utenti e suggerire miglioramenti per rendere i sistemi di raccomandazione più equi e rappresentativi delle reali preferenze degli utenti.

****

La sezione "Related Work" del documento discute i precedenti studi riguardanti il problema del **bias di popolarità** nei sistemi di raccomandazione e le sfide che questo bias comporta. Ecco una sintesi dei temi principali trattati:

### ****Popolarità e Accuratezza delle Raccomandazioni****

Diversi ricercatori hanno esplorato come la distribuzione sbilanciata dei dati, tipicamente caratterizzata da una "long-tail" (ovvero una grande quantità di elementi meno popolari), influisca sull'accuratezza complessiva delle raccomandazioni [6, 8, 17]. Alcuni studi hanno proposto algoritmi progettati per mitigare questo bias, cercando di dare maggiore visibilità agli elementi meno popolari e migliorare la copertura del catalogo [3–5, 14].

### ****Aspettative degli Utenti e Equità nelle Raccomandazioni****

Questo lavoro si distingue per il focus sull'equità delle raccomandazioni rispetto alle aspettative degli utenti. Gli autori vogliono capire come il bias di popolarità influisca sulle raccomandazioni, in modo che queste possano discostarsi dalle effettive preferenze degli utenti, specialmente in termini di quanti elementi popolari o non popolari si aspettano di vedere nella loro lista di raccomandazioni. La deviazione dalle aspettative viene considerata ingiusta perché causata dal bias nei dati di input e dai bias algoritmici presenti in alcuni sistemi di raccomandazione.

### ****Calibrazione e Distorsione nei Generi****

Un lavoro simile a quello presentato è quello di [19], che introduce il concetto di **calibrazione**: le raccomandazioni dovrebbero essere coerenti con lo spettro di elementi che un utente ha valutato. Ad esempio, se un utente ha valutato il 70% di film d'azione e il 30% di film romantici, ci si aspetta che le raccomandazioni seguano lo stesso schema. Tuttavia, il lavoro presentato si differenzia poiché non utilizza informazioni sui contenuti per analizzare come gli algoritmi sono calibrati. Invece, gli autori considerano che la distorsione di genere nelle raccomandazioni potrebbe essere causata da diversi livelli di popolarità tra i generi.

### ****Equità nei Sistemi di Raccomandazione****

L'attenzione alla **fairness** nei sistemi di raccomandazione è aumentata negli ultimi anni [13, 20]. Alcuni studi si concentrano sulla rimozione della discriminazione algoritmica contro utenti appartenenti a specifiche demografie [21], oppure sull'assicurare che elementi di diverse categorie (ad esempio, articoli meno popolari o appartenenti a diversi fornitori) ricevano una giusta esposizione nelle raccomandazioni [9, 15]. La definizione di equità adottata in questo lavoro è più vicina alla "calibration fairness" [19], dove si afferma che gli utenti dovrebbero ricevere raccomandazioni in linea con le loro aspettative basate sugli articoli che hanno valutato.

### ****Bias di Popolarità nei Diversi Algoritmi****

Infine, Jennach et al. [12] hanno confrontato diversi algoritmi di raccomandazione in termini di accuratezza e bias di popolarità, osservando che alcuni algoritmi tendono a concentrarsi più su articoli popolari rispetto ad altri. In questo lavoro, gli autori sono interessati principalmente a esaminare il bias di popolarità dal punto di vista delle aspettative degli utenti, piuttosto che solo dall'accuratezza complessiva delle raccomandazioni.

In sintesi, questa sezione mette in luce come il lavoro presentato si inserisca nel contesto delle ricerche esistenti, contribuendo con una nuova prospettiva focalizzata sull'equità delle raccomandazioni rispetto alle aspettative degli utenti, e fornendo una spiegazione di come il bias di popolarità possa distorcere le raccomandazioni in modo non allineato alle preferenze individuali.

La sezione "Popularity Bias in Data" esplora come il bias di popolarità nei dati di valutazione influenzi le raccomandazioni fornite dai sistemi di raccomandazione. Questo studio utilizza il dataset MovieLens 1M, che contiene oltre un milione di valutazioni anonime su circa 3.900 film fornite da oltre 6.000 utenti.

### ****Distribuzione Long-Tail nei Dati****

Il dataset MovieLens 1M mostra una distribuzione long-tail della popolarità degli articoli, dove pochi articoli popolari ricevono la maggior parte delle valutazioni, mentre la maggior parte degli altri articoli rimane relativamente inosservata. Questo squilibrio nella distribuzione porta gli algoritmi di raccomandazione a ereditare e spesso amplificare il bias di popolarità, raccomandando più frequentemente gli articoli già popolari. Questo meccanismo rafforza un ciclo in cui gli articoli popolari diventano sempre più popolari, mentre quelli meno noti ricevono sempre meno attenzione.

### ****Interesse degli Utenti per gli Articoli Popolari****

La sezione si concentra sull'analisi dell'interesse degli utenti per gli articoli popolari. Non tutti gli utenti sono interessati principalmente agli articoli popolari; molti utenti cercano invece raccomandazioni che includano anche articoli meno popolari. Gli autori sostengono che un buon sistema di raccomandazione dovrebbe essere in grado di soddisfare le aspettative di tutti gli utenti, non solo di quelli interessati agli articoli popolari. Ignorare le esigenze di quegli utenti che preferiscono articoli meno popolari potrebbe portare a una perdita di interesse verso il sistema da parte di questi utenti.

### ****Propensione degli Utenti verso Articoli Popolari****

Gli autori analizzano il grado di interesse degli utenti verso gli articoli popolari, mostrando che oltre 4.000 utenti nel dataset MovieLens 1M hanno almeno il 20% di articoli non popolari nei loro profili. Anche gli altri 2.000 utenti mostrano un certo interesse per gli articoli non popolari, con una proporzione di articoli popolari nei loro profili che varia tra l'80% e il 100%. Questo indica che la maggior parte degli utenti ha un mix di interessi che include anche articoli meno popolari.

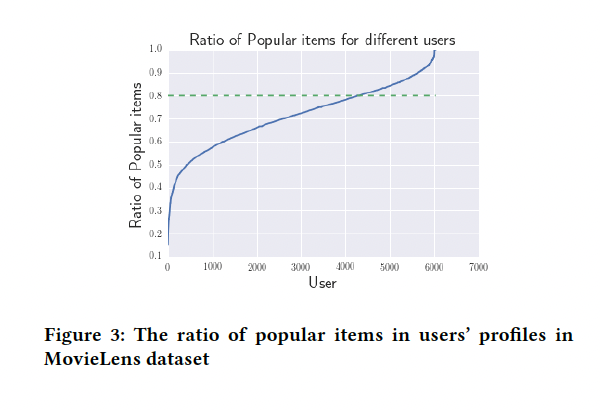
### ****Dimensione del Profilo dell'Utente e Bias di Popolarità****

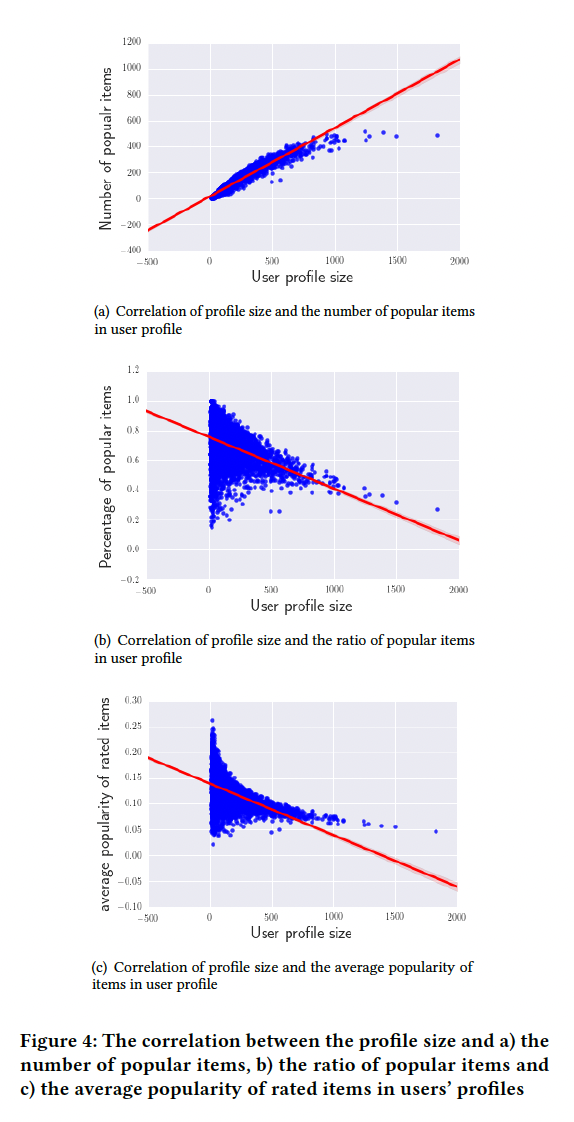
Viene esaminata la relazione tra la dimensione del profilo dell'utente (il numero di articoli valutati) e la presenza di articoli popolari nel profilo. Gli autori osservano che esiste una forte correlazione positiva tra la dimensione del profilo e il numero assoluto di articoli popolari, il che è prevedibile: con un numero maggiore di valutazioni, aumenta la probabilità che nel profilo compaiano articoli popolari.

Tuttavia, quando si considera la proporzione di articoli popolari nel profilo rispetto alla dimensione del profilo, emerge una correlazione negativa: più grande è il profilo, minore è la proporzione di articoli popolari. Questo perché, una volta esauriti gli articoli popolari, l'utente deve iniziare a valutare articoli meno conosciuti. Inoltre, gli autori mostrano che la popolarità media degli articoli valutati da un utente diminuisce all'aumentare delle dimensioni del profilo, suggerendo che gli utenti con profili più grandi tendono a esplorare e valutare un numero maggiore di articoli meno popolari.

### ****Implicazioni per i Sistemi di Raccomandazione****

Questi risultati evidenziano che gli utenti con una minore propensione a valutare articoli popolari tendono a interagire di più con il sistema, fornendo più valutazioni. Questi utenti, quindi, hanno un impatto più significativo sulle prestazioni del sistema di raccomandazione e le loro esigenze non dovrebbero essere trascurate. Un sistema di raccomandazione efficace dovrebbe essere in grado di riconoscere e soddisfare le diverse aspettative degli utenti, inclusi quelli che preferiscono articoli meno popolari, per garantire un'esperienza soddisfacente e inclusiva per tutti.

****

****

Nella sezione "Different Groups of Users in Terms of Propensity for Popularity," gli autori definiscono tre gruppi distinti di utenti basati sulla proporzione di articoli popolari presenti nei loro profili, ovvero sul loro interesse per gli articoli popolari. Questi gruppi sono definiti come segue:

### ****1. Niche Users (N)****

* **Definizione:** Gli utenti che appartengono al 20% inferiore in termini di proporzione di articoli popolari nel loro profilo.
* **Caratteristiche:** Più della metà del loro profilo è composto da articoli non popolari (long-tail). Questi utenti mostrano un interesse significativo per gli articoli meno conosciuti.

### ****2. Blockbuster-focused Users (B)****

* **Definizione:** Gli utenti che appartengono al 20% superiore in termini di proporzione di articoli popolari nel loro profilo.
* **Caratteristiche:** In media, più dell'85% degli articoli nel loro profilo sono popolari. Questi utenti tendono a preferire i successi più conosciuti e mainstream.

### ****3. Diverse Users (D)****

* **Definizione:** Gli utenti che non rientrano né nel gruppo Niche né nel gruppo Blockbuster-focused. Comprendono il restante 60% degli utenti.
* **Caratteristiche:** Questi utenti hanno una combinazione di interessi che includono sia articoli popolari che non popolari.

### ****Distribuzione dei Gruppi nel Dataset****

* **Figura 5-a:** Mostra la distribuzione della popolazione di utenti nel dataset MovieLens 1M. La maggioranza degli utenti rientra nel gruppo Diverse, con oltre 3.500 utenti. I gruppi Niche e Diverse combinati rappresentano più di 4.500 utenti, ossia circa il 75% della popolazione degli utenti.

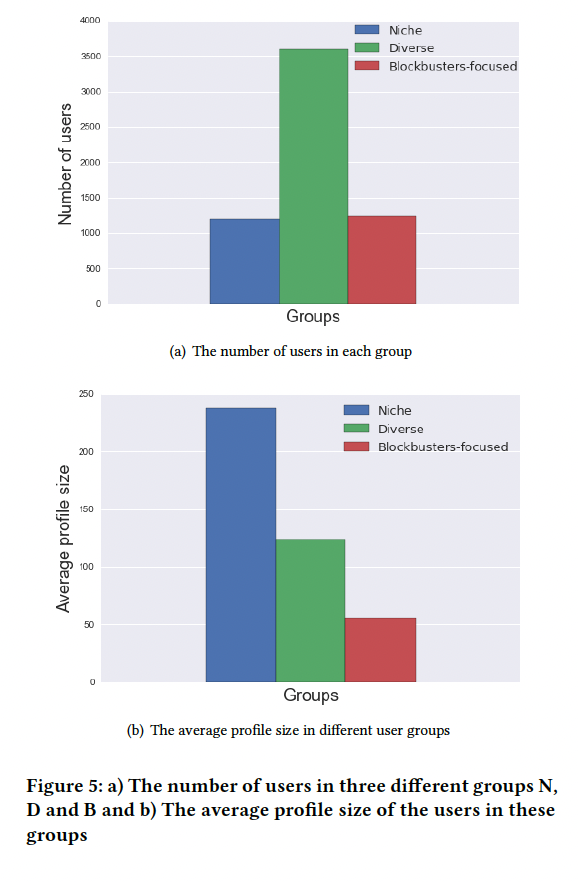
### ****Dimensione Media del Profilo****

* **Figura 5-b:** Mostra la dimensione media del profilo per ciascun gruppo di utenti. Coerentemente con le analisi precedenti, gli utenti Niche hanno i profili più grandi, seguiti dagli utenti Diverse. Il gruppo Blockbuster-focused ha invece la dimensione media del profilo più piccola.

### ****Considerazioni****

Questa categorizzazione evidenzia come gli utenti non siano omogenei nelle loro preferenze e nelle loro interazioni con il sistema di raccomandazione. In particolare, gli utenti Niche, che mostrano un interesse marcato per gli articoli meno popolari, tendono ad avere profili più ampi, suggerendo un'interazione più attiva con il sistema. Questo potrebbe indicare che gli utenti Niche potrebbero avere un impatto più significativo sulla performance del sistema e che le loro esigenze dovrebbero essere prese in considerazione in modo particolare quando si progettano algoritmi di raccomandazione.

L'analisi di questi gruppi di utenti offre una comprensione più approfondita delle diverse aspettative che i sistemi di raccomandazione devono affrontare, evidenziando la necessità di algoritmi che possano bilanciare efficacemente le raccomandazioni per soddisfare le esigenze di ciascun gruppo.

****

La sezione "Algorithmic Propagation of Popularity Bias" esplora come diversi algoritmi di raccomandazione propagano il bias di popolarità presente nei dati di valutazione. Gli autori analizzano il comportamento di vari algoritmi senza considerare inizialmente le differenze tra i gruppi di utenti, ma successivamente esaminano come questi algoritmi si comportano rispetto alle aspettative di ciascun gruppo di utenti (Niche, Diverse, Blockbuster-focused).

### ****Propagazione Generale del Bias di Popolarità****

Gli autori hanno testato diversi algoritmi, tra cui User KNN, Item KNN, SVD++, e Biased Matrix Factorization (MF), oltre a due algoritmi semplici (Most popular e Random). Gli algoritmi sono stati calibrati per ottenere una precisione simile (circa 0.1) al fine di permettere un confronto equo del loro comportamento rispetto al bias di popolarità.

* **Risultati Generali:**
  + **Correlazione tra valutazioni e raccomandazioni:** I risultati mostrano che, tranne che per l'algoritmo Random, esiste una forte correlazione tra il numero di volte che un elemento è stato valutato e il numero di volte che viene raccomandato. L'algoritmo Most popular, come previsto, mostra la correlazione più forte, seguito da User KNN e Item KNN.
  + **Algoritmi meno correlati:** Biased MF non mostra una correlazione significativa tra le valutazioni e le raccomandazioni, ma il suo scatter plot è molto sparso, indicando che un numero limitato di elementi viene raccomandato frequentemente, mentre molti altri sono raccomandati raramente.
  + **SVD++:** Mostra una correlazione positiva tra il numero di volte che un elemento è stato valutato e raccomandato, ma meno forte rispetto a KNN.

### ****Bias di Popolarità nelle Raccomandazioni per i Diversi Gruppi di Utenti****

Successivamente, gli autori analizzano come ciascun algoritmo mantiene il rapporto tra articoli popolari e non popolari nelle raccomandazioni rispetto a ciò che gli utenti si aspettano.

* **Figura 7:** Mostra che per tutti gli algoritmi, eccetto il Random, la proporzione di articoli popolari nelle raccomandazioni è significativamente aumentata rispetto alle aspettative degli utenti, confermando come questi algoritmi impongano un bias di popolarità.
  + **Impatto sui Niche Users:** Questi utenti sperimentano la discrepanza più grande tra le loro aspettative e le raccomandazioni ricevute.
  + **SVD++:** È l'algoritmo che sembra mantenere una proporzione di articoli popolari nelle raccomandazioni più vicina a quella attesa dagli utenti.

### ****Group Average Popularity (GAP)****

Gli autori introducono una metrica chiamata **Group Average Popularity (GAP)**, che misura la popolarità media degli articoli nei profili degli utenti di un certo gruppo o nelle loro raccomandazioni.

* **ΔGAP:** Viene definita come la differenza tra la GAP nelle raccomandazioni e la GAP nei profili degli utenti. Un ΔGAP pari a 0 indica una rappresentazione equa delle preferenze degli utenti rispetto alla popolarità degli articoli nelle raccomandazioni.
* **Risultati ΔGAP:** Most Popular impone il cambiamento più elevato in GAP, seguito da Item KNN. L'algoritmo Random mostra un ΔGAP negativo, indicando che le sue raccomandazioni includono articoli meno popolari rispetto ai profili degli utenti.

### ****Conclusioni****

Gli autori concludono che il bias di popolarità nei dati di valutazione viene ulteriormente propagato dagli algoritmi di raccomandazione, spesso in modo ingiusto, specialmente per gli utenti con minore interesse per gli articoli popolari (Niche Users). Questi utenti ricevono raccomandazioni che sono molto più orientate verso articoli popolari di quanto desiderato.

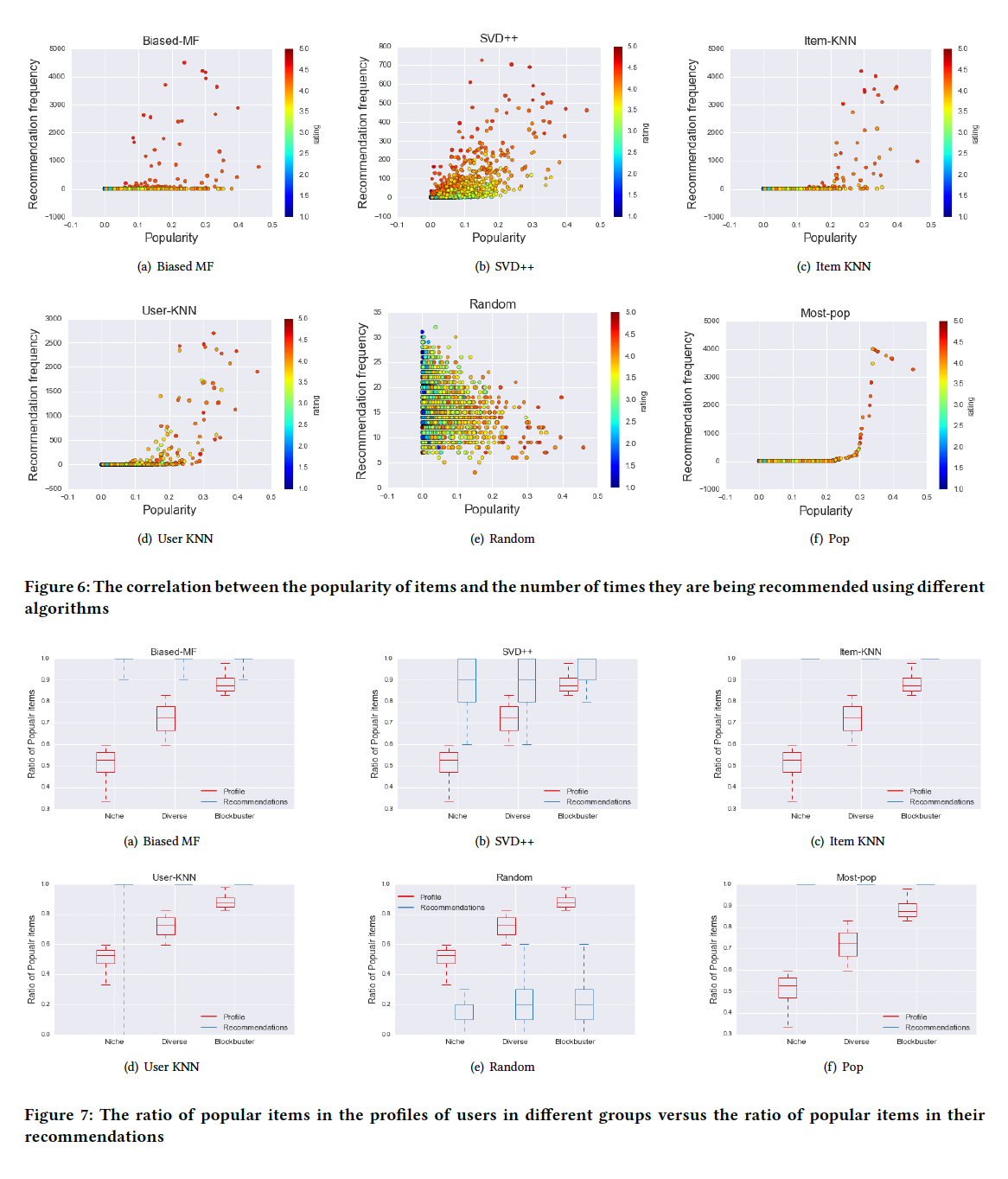
### ****Discussione e Conclusioni Finali****

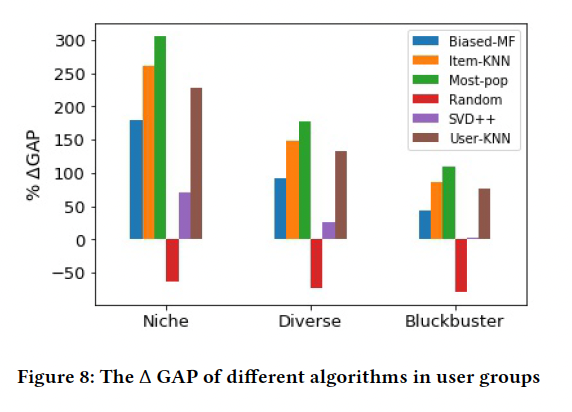
* **RQ1:** Non tutti gli utenti hanno lo stesso livello di interesse per gli articoli popolari. Molti utenti si aspettano di ricevere raccomandazioni che includano anche articoli meno popolari.
* **RQ2:** Gli algoritmi di raccomandazione sono risultati estremamente ingiusti nei confronti degli utenti con scarso interesse per gli articoli popolari, fornendo loro quasi esclusivamente raccomandazioni di articoli popolari.

### ****Lavori Futuri****

Gli autori propongono di estendere l'analisi ad altri dataset e di esplorare ulteriori misure di soddisfazione per i diversi gruppi di utenti, come la rilevanza e la diversità delle raccomandazioni.

In sintesi, questa sezione evidenzia come i diversi algoritmi di raccomandazione trattano il bias di popolarità e come questo bias influisce in modo diverso sui vari gruppi di utenti, sottolineando la necessità di algoritmi più equi che possano meglio rispondere alle diverse aspettative degli utenti.

****

****