

Sustainable Software Engineering: Advances in Research and Practice

**Approfondimento di Architettura del Software
Università degli Studi di Milano-Bicocca**

**Matteo Breganni 869549
Samuele Toniazzo 918624**



Introduzione

I **sistemi software** sono diffusi in molte aree delle nostre vite. La loro capacità di adattarsi e rispondere a nuove esigenze ha trasformato il software in un elemento cruciale per il successo delle aziende, rendendo evidente la necessità di sviluppare **software di alta qualità** che possa durare nel tempo.

In questa presentazione vengono esaminate le tematiche presenti nell'articolo di ricerca "**Sustainable software engineering: Reflections on advances in research and practice**", che riesamina i precedenti argomenti e assunzioni fatte nel precedente paper, pubblicato 5 anni prima, in luce degli avanzamenti in questo campo di ricerca dal 2018. Ciò offre l'opportunità di acquisire nuove prospettive sulle direzioni relative alla sostenibilità del software, sia in ambito accademico sia industriale, con particolare attenzione all'architettura del software.

Vogliamo quindi identificare le **tendenze emergenti**, i **progressi** compiuti (definizioni, architetture di riferimento, metriche...), le **sfide ancora aperte** e infine dare una **visione per il futuro** del software sostenibile.

Metodologia della Ricerca

Il metodo usato per condurre l'analisi è quello del "**forward snowballing**", un approccio sistematico per tracciare l'evoluzione di un argomento di ricerca attraverso l'analisi di tutti gli studi che citano l'articolo originale.

Il punto di partenza dell'analisi è stato un **articolo** di riferimento di Venters et al. pubblicato nel 2018. Il metodo dello snowballing ha previsto l'identificazione di tutti gli **studi successivi** che hanno citato questo articolo, consentendo di mappare come le idee e i concetti originali si sono evoluti e diffusi nel tempo all'interno della comunità scientifica.

Tra aprile 2018 e giugno 2023 sono state individuate in totale **234 studi**. Sono stati quindi definiti criteri di inclusione ed esclusione per selezionare solo gli studi pertinenti e dal totale solo **102** sono risultati pertinenti in base ai criteri di selezione.

Criteri di inclusione:

- Lo studio doveva **citare** esplicitamente l'articolo di Venters et al. del 2018.
- Lo studio doveva focalizzarsi sull'**architettura del software** o sull'ingegneria del software.
- Lo studio doveva essere correlato ad almeno una dimensione della **sostenibilità**.
- Lo studio doveva essere sottoposto a **peer-review**.

Criteri di esclusione:

- Duplicati dello stesso studio.
- Studi non scritti in lingua inglese.
- Studi con testo completo non pubblicamente accessibile.
- Studi con meno di 4 pagine.
- Studi non scientifici e non sottoposti a peer-review.

Gli studi sono stati mappati ulteriormente in quattro categorie: **definizione di sostenibilità, architetture di riferimento, metriche ed educazione**.

Definizione di Sostenibilità del Software

Prima del 2018

Non esisteva una **definizione** condivisa di sostenibilità del software, analogamente a quanto accade per altri concetti nell'ingegneria del software, come il concetto di architettura del software. Il termine era generalmente associato alla capacità di un sistema di "**durare**" nel tempo, ma questa definizione applicata alla sostenibilità risultavano troppo generiche per applicazioni tecniche. Venters et al. proponevano una definizione più strutturata, considerandola un "**requisito non funzionale composito**" misurabile attraverso: **estensibilità, interoperabilità, manutenibilità, portabilità, scalabilità e usabilità**. Tuttavia, nessuna delle definizioni proposte suggeriva metriche idonee per misurare la sostenibilità di un sistema software.

Dopo il 2018

La **mancanza di consenso** sulla definizione di sostenibilità del software persiste. Alcuni definiscono la sostenibilità come "capacità di durare", altri la collegano a una o più "qualità del software" o alle "dimensioni della sostenibilità" (ecologica, tecnica, economica, sociale e individuale) sostenendo che la sostenibilità sia un concetto multisistemico e stratificato, applicabile sia al processo di sviluppo del software sia agli artefatti software risultanti.

Termini come "**manutenibilità**", "**sostenibilità**" e "**approccio sostenibile**" in questo contesto vengono spesso utilizzati senza definizioni chiare, generando possibili fraintendimenti.

In contrasto per Venters il software sostenibile è «**esplicitamente progettato per la manutenibilità continua e l'evolubilità senza incorrere in un debito tecnico proibitivo e un impatto negativo sulle dimensioni della sostenibilità**». Si riconosce la necessità di allineare questa definizione con metriche software specifiche (es. dipendenza media dei componenti, costo di propagazione, debito strutturale, linee di codice duplicate, complessità ciclomatica, complessità cognitiva oltre alle metriche relative alla cronologia delle modifiche)

Definizione di Sostenibilità del Software

Considerazioni finali

Attualmente, mancano ancora parametri e **KPI (indicatori chiave di prestazione)** solidi per determinare quanto un sistema o un'architettura siano davvero sostenibili.

Nonostante la disponibilità di molte metriche per valutare aspetti come la manutenibilità e la complessità del codice, non esiste una combinazione di tali metriche efficiente per derivare valori di riferimento per la sostenibilità.

La combinazione ottimale delle metriche ,precedentemente elencate, per ottenere valori di riferimento relativi alla sostenibilità rappresenta ancora una sfida e un'area di ricerca aperta. Questo problema è ulteriormente aggravato dalla mancanza di **modelli di costo** in grado di catturare le decisioni progettuali critiche che portano alla definizione di un'architettura software.

Come conseguenza, la ricerca sulla sostenibilità del software sta solo scalfendo la superficie per fornire indicatori per la stima della sostenibilità. Esistono però approcci già consolidati, come gli **indicatori del Global Reporting Initiative** e approcci di singole nazioni, ad esempio il **Well-being of Future Generations Act in Galles**, che possono fornire una base per sviluppare metriche più robuste.

In breve c'è una necessità critica di:

- Sviluppare **metriche di sostenibilità più precise**.
- Definire meglio **modelli di costo** per le decisioni progettuali.
- Integrare la sostenibilità tecnica con le altre dimensioni della sostenibilità.

Architetture di Riferimento Sostenibili

Prima del 2018

Le **architetture software** sono state riconosciute come la base di ogni sistema che debba soddisfare requisiti di qualità, fra cui **manutenibilità** ed **estensibilità**, fattori fondamentali per favorire la realizzazione di sistemi sostenibili. Sostenibilità che dovrebbe essere presa in considerazione già in fase di progettazione. All'epoca, l'attenzione era rivolta principalmente a questi due aspetti e mancava un consenso condiviso su cosa fosse esattamente la **sostenibilità dell'architettura** e su come misurarla.

Si evidenziava che due fenomeni, il **drift**, ovvero il progressivo allontanamento tra l'implementazione del sistema e il suo design originario, e l'**erosione architettuale**, ossia il degrado della struttura e della coerenza dell'architettura a causa di scelte tecniche rapide, potevano degradare la sostenibilità. Entrambi i fenomeni contribuivano infatti al **debito tecnico**, il costo cumulativo di decisioni progettuali subottimali che richiedono interventi futuri. Si suggerivano possibili soluzioni come la gestione del debito tecnico architettuale, la misurazione della sostenibilità del design e decisioni di design sostenibili. Diverse soluzioni vennero proposte tra cui la **gestione del debito tecnico architettuale**, la **misurazione della sostenibilità del design** e l'adozione di **decisioni di design sostenibili**.

Architetture di Riferimento Sostenibili

Dopo il 2018

La ricerca sulle architetture di riferimento sostenibili si è ampliata, prendendo in considerazione anche altre **dimensioni della sostenibilità: ambientali, sociali, economiche e individuali**.

Ne è emersa l'urgenza di sviluppare **architetture di riferimento condivise** per sistemi che abbiano un impatto significativo su dimensioni della sostenibilità più ampie, con progressi riscontrati in settori come agricoltura, e-health e smart cities.

Si è inoltre riconosciuta l'importanza di **aggiornamenti continui** per mantenere le architetture allineate al dominio di riferimento e garantire che restino utili nel tempo. Esempi di **architetture di riferimento tecnicamente sostenibili** sono, tra le altre, **AUTOSAR, AXMEDIS e ARC-IT**

Tutte e tre, nate rispettivamente 20, 27 e 15 anni fa, presentano dalla nascita aggiornamenti periodici, moduli aggiunti riallocati o rimossi, supporto continuo e rilascio di nuove versioni e sono supportate da consorzi di partner strategici come BMW e Stellantis.

Anche fornitori di servizi IT come Google, Amazon e Microsoft hanno cominciato a integrare la sostenibilità nei loro framework architetturali, ma con una limitazione principalmente limitata alla **riduzione delle emissioni di gas serra** nei carichi di lavoro cloud.

Metriche e Obiettivi di Sostenibilità

Prima il 2018

Esistevano una vasta gamma **di metriche software** per valutare aspetti come la manutenibilità e la complessità del codice, ma mancava la combinazione giusta di queste metriche e nella definizione di indicatori espliciti per misurare la sostenibilità. Nonostante la disponibilità di numerosi attributi di qualità (stabilità modularità, scalabilità ecc..) esisteva una difficoltà nel misurarli con la combinazione giusta delle metriche esistenti. Alcune metriche interpretano la sostenibilità valutando gli effetti a cascata delle decisioni di progettazione, altre valutano gli aspetti "verdi" del sistema. Le seconde sono più semplici da definire poiché si analizzano dati da sensori o statistiche hardware. Venne introdotto in questo contesto il modello **GREENSOFT** che esplicava il concetto di "green and sustainable software" per regolamentare l'uso responsabile di risorse ed energia nell'ICT.

Dopo il 2018

Nonostante da anni si usino numerose metriche per migliorare la manutenibilità di sistemi e architetture, rimane l'esigenza di avere indicatori di sostenibilità adeguati. Gli studi di **Carrillo et al** hanno cominciato ad analizzare **l'effetto a cascata delle decisioni di progettazione architetturale**, mostrando le relazioni tra sostenibilità e metriche. Vengono proposti tre intervalli di instabilità per valutare come le decisioni stabili si collegano tra loro e influenzano la sostenibilità dell'architettura. Gli stessi autori nel comprendere la complessità delle reti di decisioni, analizzano la **complessità delle topologie dei grafi** di decisione valutando il modello su due casi di studio e alcuni progetti evidenziando la relazione con la sostenibilità tecnica.

Per stimare la sostenibilità del software, le metriche relative alla **manutenibilità** del software rappresentano una buona opzione, le quali indicano la qualità di un sistema. Costi di manutenzione elevati possono essere indice di una bassa sostenibilità.

Metriche e Obiettivi di Sostenibilità

Nuove direzioni

Sono stati sviluppati **strumenti di analisi del debito tecnico** che, tramite le metriche adeguate, aiutano a correggere difetti di qualità (*design smell* e *architectural smell*) che impattano negativamente sulla sostenibilità dell'architettura.

È stata introdotta una maggiore attenzione verso gli **indicatori ambientali** come il **consumo energetico** i quali però sono estremamente **dipendenti dal dominio** e dal contesto specifico.

Studi recenti hanno iniziato a considerare la relazione diretta tra **manutenibilità** del software e consumo energetico. La ricerca ha investigato come release sottoposte **refactoring** o processi di manutenzione possano portare a **risparmi energetici**, misurandone il consumo mediante il **FEETINGS**. Ciò ha portato a stabilire una connessione diretta tra la sostenibilità tecnica e quella ambientale

Altri lavori valutano aspetti "green" e i consumi energetici dei data center che, con il passare degli anni e con una sempre più alta domanda, diventano sempre più energivore.

Diverse organizzazioni hanno proposto delle metriche e degli indicatori per stimare la sostenibilità delle loro iniziative secondo parametri come: consumo energetico, gestione dell'acqua o trasporti efficienti.

Mancano però delle metriche precise per misurare la **sostenibilità nelle dimensioni individuali e sociali**, ed è necessario comprendere come le azioni di sostenibilità tecnica e ambientale possano aiutare le persone.

Rimane tutt'oggi una sfida **definire KPI per stimare la sostenibilità** nella sua dimensione ambientale.

Formazione e Competenze sulla Sostenibilità

Prima il 2018

Prima del 2018 nell'istruzione universitaria in informatica mancava un'educazione specifica sul **software sostenibile**, sia a livello accademico che nella formazione per la ricerca.

Nonostante ciò, si era osservato l'inizio di importanti iniziative di riforma educativa nelle competenze digitali, di gestione dei dati e computazionali, viste in un'ottica sociale più ampia. Sono emerse e si sono ampliate iniziative importanti come **Software Carpentry** e il **Software Sustainability Institute**, che si proponevano di promuovere pratiche di sviluppo del software più sostenibili e miravano a fornire competenze di base per la gestione del software nella ricerca.

Formazione e Competenze sulla Sostenibilità

Dopo il 2018

La pandemia di COVID-19, ha portato a un cambiamento significativo nel modo in cui l'educazione veniva erogata portando a diverse sfide sociali e difficoltà legate alla salute mentale. Con il ritorno in presenza si sono osservate differenze negli studenti in particolare nell'ambito dell'informatica in cui emergono preoccupazioni per la ridotta interazione tra di loro. Questo potrebbe influire negativamente sulla loro carriera.

Varie istituzioni hanno dato sempre più peso verso l'integrazione della sostenibilità nei programmi di studio, in linea con gli **Obiettivi di Sviluppo Sostenibile** delle Nazioni Unite.

L'ingegneria è chiamata a contribuire in modo significativo al raggiungimento degli SDG ma pochi corsi di studio di informatica, matematica o fisica includono la sostenibilità nei loro programmi, preferendo un focus maggiore sulle competenze tecniche e sulla teoria classica, piuttosto che sui concetti di sostenibilità.

Gli studi in questo periodo evidenziano la necessità di competenze in una vasta gamma di aree, tra cui:

- **Concetti fondamentali di sostenibilità.**
- **Pensiero sistemico:** analizzare le interconnessioni nei sistemi.
- **Soft skills:** capacità di comunicazione e collaborazione.
- **Sostenibilità tecnica:** conoscenza dei principi di progettazione e sviluppo del software sostenibile
- **Costruzione del business case per la sostenibilità:** capacità di dimostrare il valore economico delle pratiche sostenibili.
- **Impatto della sostenibilità:** comprensione dell'impatto delle scelte di progettazione sull'ambiente e sulla società.
- **Etica:** consapevolezza delle implicazioni etiche e sociali del software.

Formazione e Competenze sulla Sostenibilità

Sfide future

Una delle principali sfide è riuscire a **creare spazio per la formazione sulla sostenibilità** all'interno dei corsi di molte università. Iniziative come il **ACM/IEEE** per i curricula di informatica offrono un'opportunità per ripensare l'insegnamento, integrando concetti fondamentali di sostenibilità e nuovi temi rilevanti.

Esiste un'opportunità per incorporare **l'attivismo** nel campo dell'educazione informatica che potrebbe portare gli studenti a diventare portatori di cambiamento attraverso il loro lavoro.

Prospettive Future

Tendenze nell'Architettura del Software Sostenibile

Sebbene la progettazione di architetture software è un processo consolidato, ma la misurazione della consistenza architetturale e le strutture che erodono la sostenibilità rimangono delle rimangono sfide aperte.

Una delle sfide principali è quella di **comprendere meglio le interazioni con le diverse dimensioni della sostenibilità** sull'architettura del software. Questo significa andare oltre la sola sostenibilità tecnica e considerare anche le dimensioni ambientali, sociali, economiche e individuali. Bisogna quindi progettare **architetture software che siano resilienti** in un periodo di rapidi cambiamenti, come la transizione verso un'economia globale sempre più lontana dai combustibili fossili.

Le architetture software devono essere progettate per **supportare obiettivi di sostenibilità** nelle aree per cui vengono create, il che potrebbe richiedere il coinvolgimento di una gamma più ampia di stakeholder.

Si evidenzia inoltre come, tranne alcune consolidate come AUTOSAR e ARC-IT, molte architetture non sopravvivono oltre la loro prima versione. L'obiettivo è quindi quello di migliorarne la longevità migliorando la ricerca sulla **valutazione degli impatti di sostenibilità delle architetture di riferimento**.

Prospettive Future

Tendenze nelle Metriche di Sostenibilità

Gli avanzamenti nelle metriche di sostenibilità in diversi domini applicativi sono stati diversi negli ultimi anni, ma si sono comunque individuate alcune lacune affrontabili in future ricerche:

- Si riconosce la necessità di definire e sviluppare **indicatori di sostenibilità tecnica di alto livello** basati su metriche di codice e architettura. Questi indicatori devono fornire informazioni sulla capacità di un sistema di durare nel tempo in relazione ai costi di manutenzione, andando oltre le metriche di manutenibilità classiche.
- Si incoraggia la ricerca su **modelli predittivi** che possano anticipare le richieste inaspettate di risorse, analizzando il consumo e adattandolo ai picchi di domanda. Questo aiuterebbe a migliorare ulteriormente la **sostenibilità ambientale**.
- È necessario sviluppare **KPI per stimare la sostenibilità** in vari ambiti (clima, trasporti, acqua, oceani), dato che la maggior parte delle misure attualmente esistenti sono focalizzate sulla sostenibilità in software, energia e la gestione dei rifiuti. Questi nuovi KPI aiuterebbero le organizzazioni ad avviare iniziative sostenibili.
- Esiste ancora una significativa mancanza di metriche per misurare la **sostenibilità individuale e comunitaria** nelle dimensioni tecniche e ambientali. Bisogna collegare queste dimensioni per capire come la sostenibilità possa beneficiare le persone.

Prospettive Future

Tendenze nell'educazione e nelle competenze per la sostenibilità

Analizzando le offerte di lavoro, le competenze richieste per gli specialisti in sostenibilità includono **competenze analitiche, comunicative, di leadership e problem-solving**. In questo contesto c'è una grande necessità di integrare la sostenibilità in tutti i corsi di **informatica**, includendo lo sviluppo di queste competenze. È necessario bilanciare la formazione di **generalisti** (con una conoscenza sia dell'informatica che della sostenibilità) e di **specialisti** (con competenze in aree come l'architettura del software e l'infrastruttura del software sostenibile).

Si evidenzia il rischio di **greenwashing** e di dichiarazioni aziendali che non sono supportate da azioni concrete. Per questo, è importante che la formazione includa aspetti di responsabilità etica.

La **progettazione del software** è un elemento chiave per lo sviluppo di software sostenibile. Una progettazione sub-ottimale può portare a complessità, debito tecnico, code smells e aumento del rischio di entropia del software, con conseguente aumento dei **costi di manutenzione ed evoluzione**.

Il ragionamento a livello architetturale porta a miglioramenti visibili nella progettazione e nello sviluppo del software, con conseguente aumento in termini di qualità e sostenibilità.

C'è assolutamente bisogno di **strumenti innovativi** per gli ambienti emergenti e dinamici di oggi, dove i sistemi software sono progettati per una continua evolvibilità e l'adattabilità continua senza incorrere in un debito architetturale e tecnico proibitivo.

Conclusioni

- L'articolo evidenzia come la sostenibilità del software sia un **campo complesso**, caratterizzato da diverse sfide e aree che richiedono ulteriore ricerca e sviluppo.
- Le architetture del software sono fondamentali per la sostenibilità e devono essere **progettate per l'evolvibilità continua**, senza accumulare un debito tecnico proibitivo.
- La sfida principale è misurare la sostenibilità come una **qualità chiave del software**, combinando metriche appropriate e collegando le diverse dimensioni di sostenibilità.
- Il futuro dell'ingegneria del software sostenibile richiede un **approccio multidisciplinare**, che includa una visione etica e un impegno per la formazione e la **ricerca**.