

Descrizione progetto di ricerca
*Sistemi, strumenti, e metodologie per il
deployment e l'aggiornamento di software
industriale nel cloud-edge continuum*

Tutor: Prof. Danilo Pianini

May 29, 2023

1 Collocazione

I sistemi software possiedono un proprio ciclo di vita, che inizia con la definizione dei requisiti iniziali e la progettazione, per terminare con il ritiro operativo. Storicamente, le fasi di definizione dei requisiti e progettazione sono quelle che hanno ricevuto più attenzione da parte della comunità scientifica, con approcci dedicati alla comprensione del dominio [4], alla progettazione architetturale [9, 10], a quella di dettaglio [5, 6], finendo con la codifica. Questo modello di sviluppo ha cominciato però a mostrare limiti quando il software ha cominciato ad essere riconosciuto come un asset in continua evoluzione, che ha bisogno di essere mantenuto e aggiornato nel tempo in uno stato di “endless development” [7]. Per affrontare questi scenari, sempre più comuni con la diffusione di sistemi software distribuiti e di software come servizio, sono stati elaborati approcci di sviluppo che favoriscono la capacità di reagire a requisiti in costante cambiamento e di produrre velocemente miglioramenti incrementali del software [8]. Un aspetto cruciale di questi approcci è la capacità di poter distribuire in modo rapido e controllato le nuove versioni del software (deployment) [12, 3, 11]. Questi approcci, su cui sono attivi diversi filoni di ricerca, saranno sempre più importanti nel prossimo futuro con il passaggio ad architetture eterogenee che rientrano nel paradigma del cloud-edge continuum [1].

Uno dei nodi critici di questi approcci è la loro applicabilità a sistemi che vanno ad interfacciarsi con sistemi fisici, e, in particolare, con macchine industriali. Queste ultime sono sempre più spesso dotate di capacità di

calcolo e di connettività [2]; conseguentemente, larga parte delle loro capacità è demandato a software apposito, capace di interfacciarsi sia con l'esterno che con gli azionatori elettro-meccanici. Questo software ha vincoli rilevanti in termini di piattaforma hardware, nonché in termini di affidabilità e sicurezza; cionondimeno, e, anzi, proprio per questo, è necessario che sia aggiornato in modo controllato, con possibilità di aggiornamento di più macchine di linea (che collaborano fra loro per il raggiungimento degli obiettivi di business) in modo transazionale, con recupero dello stato precedente in caso di fallimento o di regressione rispetto al software precedente. Purtroppo, le tecniche tipicamente impiegate per software che non hanno interfacciamento direttamente con hardware industriale non sono, di norma, portabili in questo speciale contesto, che richiede quindi un approccio dedicato.

2 Obiettivi dell'assegno di ricerca

L'attività di ricerca proposta si pone come obiettivo principale la definizione di una metodologia per la gestione dell'aggiornamento di sistemi software manifatturieri, e, più specificatamente, come esempio operativo, applicabile a macchine industriali connesse e integrate nel cloud-edge continuum per la sezionatura e la lavorazione di legno e plastica. In particolare, la ricerca ha come obiettivo la definizione di un processo automatico (o semi-automatico) che consenta di aggiornare le versioni del software installate sulle macchine in modo sicuro e controllato. Tali metodologie dovranno tener conto delle caratteristiche pre-esistenti dei sistemi industriali obiettivo, e dovranno inoltre essere in grado di adattarsi a sistemi eterogenei, in cui le macchine possono avere configurazioni anche largamente differenti, e possono essere installate in ambienti eterogenei. Il risultato della ricerca sarà lo studio, e la conseguente produzione, di un insieme di metodologie in grado di poter essere applicate in modo trasparente alla maggior parte dei sistemi consolidati, senza richiedere modifiche sostanziali al software pre-esistente. Ciascuna alternativa verrà correlata di un'analisi dei costi e dei benefici, in modo da poter valutare la sua applicabilità in diversi scenari, in aggiunta a un insieme di pre-requisiti necessari per poter essere applicata. Questo progetto si avvarrà della collaborazione di SCM S.p.A. e rientra nell'ambito del Bando per la promozione di progetti di investimento in attuazione dell'art. della LR 14/2014 dell'Emilia-Romagna – DGR 1098/2022 per il progetto tipologia B) presentato da SCM Group S.p.A. (PG/2022/1032336), CUP: E69J22007520009.

3 Piano di Formazione

Il piano di formazione prevede i seguenti punti:

1. Studio delle metodologie di deployment e aggiornamento di sistemi software presenti allo stato dell'arte.
2. Produzione di tecniche di deployment che possano essere applicate a sistemi software di macchine industriali, in particolare per sistemi manifatturieri, in cui è necessario che il software sia installato su più macchine eterogenee.
3. Analisi dei costi e benefici di ciascuna alternativa prodotta.
4. Applicazione delle tecniche prodotte ad un caso di studio per validarne i benefici.

References

- [1] Luciano Baresi, Danilo Filgueira Mendonça, Martin Garriga, Sam Guinea, and Giovanni Quattrocchi. A unified model for the mobile-edge-cloud continuum. *ACM Trans. Internet Techn.*, 19(2):29:1–29:21, 2019.
- [2] Piers Barrios, Christophe Danjou, and Benoît Eynard. Literature review and methodological framework for integration of iot and PLM in manufacturing industry. *Comput. Ind.*, 140:103688, 2022.
- [3] Michael de Jong, Arie van Deursen, and Anthony Cleve. Zero-downtime SQL database schema evolution for continuous deployment. In *39th IEEE/ACM International Conference on Software Engineering: Software Engineering in Practice Track, ICSE-SEIP 2017, Buenos Aires, Argentina, May 20-28, 2017*, pages 143–152. IEEE Computer Society, 2017.
- [4] Eric Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley, 2004.
- [5] Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1 edition, 1994.
- [6] Jeremy Gibbons. *Design Patterns as Higher-Order Datatype-Generic Programs*. 2006.

- [7] Fumihiko IKUINE and Hideki FUJITA. Endless development is the best quality assurance. *Annals of Business Administrative Science*, 12(5):251–263, October 2013.
- [8] Adarsh Kumar Kakar and Ashish Kakar. Have the agile principles endured? an empirical investigation post 20th anniversary of the agile manifesto (2001). In Tung X. Bui, editor, *56th Hawaii International Conference on System Sciences, HICSS 2023, Maui, Hawaii, USA, January 3-6, 2023*, pages 6496–6505. ScholarSpace, 2023.
- [9] Mohamad Kassab, Manuel Mazzara, JooYoung Lee, and Giancarlo Succi. Software architectural patterns in practice: an empirical study. *Innov. Syst. Softw. Eng.*, 14(4):263–271, 2018.
- [10] Zhongyi Lu, Declan T. Delaney, and David Lillis. A survey on microservices trust models for open systems. *IEEE Access*, 11:28840–28855, 2023.
- [11] Chaitanya K. Rudrabhatla. Comparison of zero downtime based deployment techniques in public cloud infrastructure. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. IEEE, October 2020.
- [12] Eddy Truyen, Bert Lagaisse, Wouter Joosen, Arnout Hoebreckx, and Cédric De Dycker. Flexible migration in blue-green deployments within a fixed cost. In *Proceedings of the 6th International Workshop on Container Technologies and Container Clouds, WOC@Middleware 2020, Delft, The Netherlands, December 07-11, 2020*, pages 13–18. ACM, 2020.