

Fabio Dell'Onte

 fabiodellonte@gmail.com

 +39 329 748 8632

 Pesaro, Marche, Italia

 03/02/1982

 LinkedIn: <https://www.linkedin.com/in/fabiodellonte/>

 GitHub: <https://github.com/fabiodellonte>

Chi Sono

Sono una persona ottimista e amante della vita, che crede nell'equilibrio tra dedizione professionale e gioia di vivere. Amo le sfide mantenendo sempre un approccio positivo alla vita. Quando la situazione lo richiede, so essere estremamente determinato e concentrato, incanalando tutte le mie energie nel raggiungimento degli obiettivi. Credo fortemente nelle persone e nel loro potenziale, cercando sempre di far emergere il meglio in chi mi circonda. Soprattutto, sono profondamente legato alla verità, alla trasparenza e all'integrità in ogni aspetto della vita, considerando questi valori pilastri fondamentali sia delle relazioni personali che professionali. Appassionato di tecnologia e innovazione, con oltre 15 anni di esperienza nello sviluppo software e nella leadership di team. Durante la mia carriera, ho gestito con successo progetti complessi e guidato team ad alte prestazioni, concentrandomi sempre sulla consegna di soluzioni di qualità e sulla creazione di un ambiente collaborativo orientato alla crescita. Rimango operativo sul campo, spesso prototipando soluzioni, per poi di trasferire conoscenze e responsabilità al team.

Esperienza

Software Operations Manager · Biesse

Gen 2023 – Presente

- SDLC / Application Lifecycle Management (Agile DevOps, CI/CD)
- SCM: gestione della configurazione software
- Sviluppo protezione software
- Sviluppo infrastruttura Cloud
- Relazioni tecniche con fornitori
- Coordinamento team tecnico

Process and Test Manager · Biesse

Dic 2016 – Dic 2022

- Sviluppo processi ALM e automazione test
- Integrazione cloud per analisi dei dati utente
- Strumenti per infrastruttura software e distribuzione
- Gestione relazioni tecniche interne/esterne
- Coordinamento team tecnico

Software Team Leader · Biesse

Mar 2012 – Nov 2016

- Gestione tecnica software Cad/Cam (vetro e pietra)
- Sviluppo post processor e software per macchine lavorazione verticali
- Validazione software e QA
- Demo e presentazioni software nelle fiere del settore

Software Developer · Biesse

2007 – Feb 2012

- Integrazione librerie lavorazione 5 assi Moduleworks
- Generazione toolpath CAM e ottimizzazione lavorazioni
- Diagnostiche Cad/Cam
- Ottimizzazione prestazioni del codice

Competenze

Linguaggi: C#, C++, Python, TypeScript, PowerShell, NET Framework, STL, QT

DevOps & Strumenti: Windows, Linux, Azure, Azure DevOps, Jenkins, Git, GitHub, SVN, TFVC, Docker, YAML, SQL, JSON, XML

Specifiche: CAD/CAM, Windows Desktop Apps & Services, Remote Services, Troubleshooting, Problem Solving, Project Management, Process Automation, Web Apps, OpenGL

Analytics & Dati: Elastic Stack, Microsoft Power BI, Excel **Ambiente di Sviluppo:** Visual Studio 2022, VS Code, Cursor AI **AI & ML:** Ollama, OpenAI ChatGPT/API, Anthropic Claude, Meta LLaMA, Microsoft Phi

Formazione

Alma Mater Studiorum – Università di Bologna · Laurea Magistrale in Informatica

2005 – 2007 · Voto: 110/110 con lode

Tesi: *Tassellazione adattiva per superfici di suddivisione in XCSbd*

Università degli Studi di Urbino “Carlo Bo” · Laurea Triennale in Informatica Applicata

2001 – 2004 · Voto: 109/110

ITIS “Enrico Mattei” – Urbino · Elettronica e Telecomunicazioni

1996 – 2001 · Voto: 100/100

Lingue

- Italiano: madrelingua
- Inglese: intermedio

Interessi

Famiglia, Crescita personale e di team, Tecnologia, Automazione, AI, Running

Progetti chiave e servizi gestiti

Kaos Cloud Connectivity (Edge → Kanagawa → Flexy Frame) · 2023 – Presente

Contesto: Esporre in modo sicuro i dati macchina raccolti da **Monitoring Platform**; l'agent edge si registra al server Kaos (Kanagawa Cloud) e abilita l'app **Flexy Frame** in React per monitoraggio e comandi remoti.

Ruolo: Cloud Solution Architect & Lead Developer – design micro-servizi, coding TypeScript/Node, integrazione edge ↔ cloud, CI/CD.

Architettura:

- **Micro-servizi non containerizzati** deployati su server Linux nel datacenter interno.
- Pubblicazione verso Internet tramite **reverse proxy Nginx** / HTTPS.
- Gateway WebSocket (IRIS) per streaming realtime.

Azioni:

- Sviluppate API REST + WSS in **Node.js/TypeScript**; handshake sicuro edge-agent → token → canale dedicato.
- Front-end **Flexy Frame** (React) per KPI live e comandi remoti.
- Pipeline GitHub Actions per build, test e deploy continuo su cluster Kanagawa.

- Strumentazione **Elastic APM** + dashboard uptime/perf.

Risultati: onboarding macchina one-click • tracciabilità 100 % sessioni remote.

Stack: Node.js · TypeScript · React · WebSocket · Nginx · IRIS protocol · Elastic APM · GitHub Actions · VS Code / Cursor

Monitoring Platform (Plug-in IoC Agent + Elastic) · 2019 – Presente

Contesto: Telemetria full-stack per 6 + applicazioni: log, prestazioni, usage analytics inviati in tempo reale a cluster Elastic on-prem.

Ruolo: Principal Architect & Lead Developer – design modulare, coding C#, setup Elastic, CI/CD.

Azioni:

- Servizio **C#** installato via bootstrap WiX che, all'avvio dell'app, istanzia un **agent** con plug-in specifico (IoC container).
- Raccolta log, query DB, counter CPU/RAM, eventi Windows e action-tracking; invio dati via **Filebeat** a **Elasticsearch**.
- Dashboards **Kibana** per crash trend, health e KPI d'uso; gateway 'Kanagawa Cloud' per streaming esterno selezionato.
- Auto-update agent da **Azure Blob Storage**; canale polling REST per comandi remoti (debug/patch).

Risultati: osservabilità 24×7 su 6 + prodotti • diagnosi bug accelerata • rollout aggiornamenti zero-touch.

Stack: C#/.NET · IoC Plug-in · Filebeat · Elasticsearch · Kibana · Azure Blob Storage · Azure DevOps

GenAI & LLM Experimentation Suite · 2024 – Presente

Contesto: Portare l'Intelligenza Artificiale generativa nell'ecosistema dei software del reparto di ingegneria, dall'infrastruttura Kaos all'R&D.

Ruolo: AI Integrator PoC – integrazione modelli, PoC, knowledge-sharing.

Highlights:

- Integrato **OpenAI GPT-3.5/4** e modelli self-hosted (**Ollama**, **DeepSeek**, **Claude**, **Gemini**) su canali Kaos.
- Test di **fine-tuning**, **RAG** e prompt-engineering per agenti interni.
- Sviluppo rapido con **FastAPI / Node** + deploy CI/CD; demo frontend **React (Flexy Frame)**, Netlify.
- Uso quotidiano di **Cursor AI**, **GitHub Copilot** **Risultati:** pipeline standard per modelli custom; sandbox GenAI adottata da team Dev & QA; base per chatbot di supporto e code-assist aziendale.

Stack: OpenAI · Ollama · FastAPI · Node.js · React · Azure DevOps / GitHub Actions ·

License Protection Suite (HASP HW → Steganography SW) · 2012 – Presente

Contesto: Garantire protezione IP e gestione licenze per l'intero portafoglio software dell'azienda; da dongle Gemalto HASP a licenza software steganografica (PNG).

Ruolo: License & Security Lead – manutenzione KeyProgrammer, policy licenze, evoluzione libreria LKLib.

Azioni:

- Manutenuto tool **KeyProgrammer** (C#) per generazione chiavi HASP su VPB interne; gestione utenti & policy.
- 2019: integrata libreria esterna steganografia PNG → sviluppata **LKLib** riusabile (C#, obfuscation multilayer).
- Porting API licensing in 6 + prodotti; modulo plug-in (HW/SW) con scelta runtime.
- Automatizzato provisioning licenze via pipeline DevOps e dashboard usage.
- Gestito supporto e rinnovi chiavi, migrazione utenze da dongle a licenza soft.

Risultati: 60 % software migrato a licenza SW • ridotti costi dongle • attivazioni online < 10 s • protezione IP potenziata.

Stack: C#/.NET · Gemalto HASP SDK · PNG Steganography Lib · Obfuscation · Azure DevOps

Energy Monitoring System (OPC UA) · 2024 – Presente

Contesto: Raccolta e storicizzazione dei consumi energetici delle macchine sulle quali è installto il software tramite protocollo **OPC UA**, per analisi continua e ottimizzazione efficienza.

Ruolo: Lead Developer – design, coding C++/Qt, integrazione con modulo OPC-UA esistente.

Azioni:

- Implementato poller OPC UA che acquisisce KPI energetici e li persiste in archivio storico.
- UI Qt su Windows per grafici live e query periodiche (giorno / settimana / mese).
- Gestito configurazione macchine e soglie alert in file JSON ricaricabili a caldo.
- Logging e status dashboard per service in locale.

Risultati: monitoraggio continuo dei consumi • base dati storica per analisi efficienza.

Stack: C++ · Qt · OPC UA SDK · Windows · JSON config

Test Automation Platform & VM Farm Orchestrator · 2015 – Presente

Contesto: Automatizzare test end-to-end per tutti i prodotti software Core dell'azienda, integrando esecuzione e code-coverage nelle pipeline CI/CD.

Ruolo: Test Automation Architect & Lead – ideazione framework, sviluppo tool, governance pipeline.

Azioni:

- Creato tool **C#** con 3 modalità:
 - i. **Integration Test** (diff testo ISO GCode)
 - ii. **UI Record & Playback** per app **WPF**
 - iii. **Code-Coverage** (dotCover, VS Coverage)
- Orchestrato bilanciamento carico su farm di VM: agente distribuisce code-queue dai job pipeline.
- Gestione coda test: priorità, retry, report HTML con metriche coverage inviato in CI.

- Strumentata instrumentazione C++/C# e upload report su dashboard DevOps.
Risultati: test regressione 50 % automatizzati • coverage crescita costante anno-su-anno.
Stack: C#/ .NET • C++ • WPF • JetBrains dotCover • Visual Studio Code Coverage • Azure DevOps Pipelines • PowerShell

Diagnostic Center · 2021 – 2022

Contesto: Ridurre il troubleshooting manuale per il Service: tool locale che esegue check di funzionamento (registry, file, DB, ecc.) tramite plug-in per ciascuna applicazione.

Ruolo: **Concept Owner & Requirements Lead** – ideazione, raccolta specifiche, supervisione sviluppo.

Azioni:

- Definita architettura modulare: ogni team integra un plug-in **C#** con serie di test (1 → n).
- Standardizzati template di check (registry key, file presence, config DB).
- Guidato dev team nella UI e nel motore di esecuzione parallela.
- Documentazione e training Service per installazione/uso in campo.

Risultati: troubleshooting automatizzato • ridotti tempi risoluzione ticket.

Stack: C#/ .NET • Plug-in DLL • Windows Registry/File/DB APIs • PowerShell (installer)

Release Planning & Orchestration Service · 2017 – Presente

Contesto: Coordinare rilasci multi-prodotto con branch strategy dedicata e Team Test, garantendo calendario condiviso e release notes ufficiali.

Ruolo: **Release Coordinator & Automation Lead** – pianificazione, reminder, branch-management, pubblicazione.

Azioni:

- Creato query e dashboard **Azure DevOps** per calendario rilasci; reminder automatici fine-sviluppo a dev & QA.
- Gestito chiusura branch feature → release, creazione branch hotfix e duplicazione test suite per ogni ramo.
- Orchestrata esecuzione test con Team Test, validazione build e firma release.
- Automatizzata generazione & pubblicazione **Release Notes** (Markdown → Git).
- Sincronizzato team (Dev, QA, PM) via mail + Teams; governance permessi su pipeline e repos.

Risultati: > 95 % release on-time.

Stack: Azure DevOps Queries & Dashboards • Git Branch Policies • PowerShell • ADO Test Plans • Markdown Release Notes

License Provisioning & Cost-Control Suite · 2021 – Presente

Contesto: 15 + software strategici (IDE, DevOps, productivity) usati da > 250 utenti interni.

Ruolo: Service Owner – budgeting, rinnovi contratti, ottimizzazione costi.

Azioni:

- Accentrato approvvigionamento licenze e vault credenziali.
 - Implementato modello previsionale budget basato su consumo reale.
- Risultati:** TCO –17 % • onboarding –50 % • visibilità licenze 100 %.
- Stack:** Power BI • Azure DevOps • GitHub API • PowerShell • Elastic Stack

VM & Physical Lab Infrastructure · 2019 – Presente

Contesto: 220+ VM 20+ PC fisici dedicati a test automatici/manuali, build-farm CI e server per servizi esposti all'esterno.

Ruolo: Infrastructure Owner – capacity planning, monitoraggio, security hardening.

Azioni:

- Standardizzato immagini e provisioning con script **PowerShell**.
- Implementato cluster build farm (AzureDevOps + agents autoscaling, jenkins).
- Gestito patching e backup incrementali.

Risultati: uptime > 99,9 %

Stack: VMware / VSphere · Windows & Linux · Jenkins · PowerShell, Remote Desktop Manager

SCM & Repository Governance Suite · 2016 – Presente

Contesto: Gestione centralizzata di Azure DevOps (on-prem & SaaS), GitHub Enterprise, SVN on-prem, Jenkins, TFVC/Git – > 250 utenti interni + fornitori esterni.

Ruolo: SCM Service Owner – policy di commit/branch, controllo permessi, server & licenze, coordinamento vendor.

Azioni:

- Definite e applicate branch-policy standard (PR mandatory, code-review, build-gate).
- Manutenzione proattiva server (patch, backup, monitor) → **alta affidabilità** per build & release.
- Gestione contratti/licenze e onboarding fornitori esterni.

Risultati: uptime > 99,9 % • accesso repo 100 % audit-compliant • servizio high-touch per 250+ utenti.

Stack: Azure DevOps on-prem/SaaS · GitHub Enterprise · SVN · Jenkins · PowerShell · Elastic Stack

SDLC Process Automation & Work-Item Governance · 2018 – Presente

Contesto: Allineare il processo di sviluppo software alle direttive corporate e di reparto su 20 + progetti Azure DevOps, tramite Work Item personalizzati e integrazione con servizi aziendali.

Ruolo: SDLC Process Architect – design campi/regole WI, integrazioni, formazione team, gestione security & access.

Azioni:

- Personalizzati campi, stati e regole di Work Item per più project collection.
- Integrato Work Item con Service Desk e Power BI (reporting KPI qualità).
- Redatto e diffuso linee-guida di processo su wiki, formati 200 + dev/QA.

- Sviluppato tool di linting Work Item e notifiche compliance via API.

- Gestito livelli di accesso e permessi per team interni ed esterni.

Risultati: adozione processo 100 % • governance audit-ready.

Stack: Azure DevOps API • PowerShell • REST • Wiki Markdown • Power BI

Power BI DevOps Software Development Dashboards Suite · 2022 – Presente

Contesto: Necessità di insight unificati su backlog, release e produttività team; dati estratti da Azure DevOps e resi disponibili a più funzioni aziendali.

Ruolo: BI Implementer & Stakeholder Liaison – sviluppo report, versioning Git, governance permessi.

Azioni:

- Creato connettori analytics **OData** → Power BI per work-item, build & release data.
- Versionato i report in **Git**; pipeline CI per pubblicazione automatica su workspace Power BI.
- Definito modello di permessi per audience (Dev, QA, Governance, Release) con row-level security.
- Pubblicate dashboard su KPI produttività team, lead-time, PR aging, release calendar.

Risultati: decision-making data-driven per 5+ reparti • ridotto excel manuali (–90 h/mese) • adozione > 80 % team.

Stack: Power BI • Azure DevOps OData • Git • Excel

DevOps Integration Bridge (Salesforce - Teamcenter - Azure DevOps) · 2018 – Presente

Contesto: Sincronizzare automaticamente le Problem Report (PR) fra Salesforce, PLM Teamcenter e Work Item Azure DevOps; in precedenza l'allineamento era manuale (≈ 15 min/PR).

Ruolo: Integration Engineer & Service Owner – design, sviluppo C#, gestione deploy e permessi API.

Azioni:

- Sviluppato micro-servizio **C#/.NET** che converte e pubblica PR via protocollo su filesystem verso i tre sistemi.
- Implementato retry resiliente, mapping campi e gestione link work-item ↔ PLM.
- Pipeline CI/CD per build e rollout su host Windows Server.

Risultati: 3 PR/giorno sincronizzate **in tempo reale** • **+187 h/anno risparmiate** (15 min → 0 s) • allineamento dati 100 % fra CRM, PLM, DevOps.

Stack: C#/.NET • REST/SOAP API • Azure DevOps API • Salesforce API • Teamcenter SOA • Elastic Stack

TimeBridge (Azure DevOps → Oracle Time & Labor) · 2020 – Presente

Contesto: Necessità di trasferire automaticamente le ore lavorate nei Work Item Azure DevOps verso Oracle Time & Labor; prima il team (> 250 persone) impiegava ≈ 10 min/settimana a testa per l'inserimento manuale.

Ruolo: Product Owner & Maintainer – raccolta requisiti, project-management, manutenzione

evolutiva.

Azioni:

- Guidato lo sviluppatore principale (team interno) e definito backlog funzionale.
- Implementato micro-servizio **C#/.NET**: polling Azure DevOps API, trasformazione dati, push su Oracle OTL.
- Gestito deployment, monitor e gestione OAuth/API keys.
- Introdotta validazione automatica ore e report di esito via email.

Risultati: 20 min/utente/settimana eliminati → **≈ 2 000 h/anno risparmiate** • inserimenti error-free • adozione 100 % team.

Stack: C#/.NET · Azure DevOps REST API · Oracle PowerShell · SQL

Process-Rules Validator (Azure DevOps Work Items) · 2016 – Presente

Contesto: Garantire la conformità al processo Scrum su 2 progetti Azure DevOps attraverso 30 + regole automatiche (AreaPath, Sprint, Business Value, linkage, test-coverage, ecc.).

Ruolo: Lead Developer & Maintainer – raccolta requisiti, architettura plug-in, governance permessi e mailing.

Azioni:

- Sviluppato servizio **C#/.NET** schedulato nightly: esegue validazioni, aggrega risultati per Team.
- Architettura aperta: nuove regole pluggabili via JSON & DLL senza downtime.
- Report inviati da via mail con elenco issue e link WI; RLS personalizzata per 10 + team.
- Gestite eccezioni: reply-mail → auto-ticket al Team Operations.

Risultati: verifica 100 % Work Item ogni notte • correzioni proattive → – errori di processo • risparmiate ore di audit manuale.

Stack: C#/.NET · Azure DevOps REST API · SMTP · JSON plug-ins

Release Notes Portal (GitHub Pages) · 2022 – Presente

Contesto: Rendere pubbliche e ricercabili le note di rilascio di 6 software principali; automatizzare l'estrazione del campo *Release Notes* dai Work Item Azure DevOps e la pubblicazione su GitHub Pages.

Ruolo: Solution Owner & Lead Developer – raccolta requisiti, integrazione DevOps, mentoring junior dev.

Azioni:

- Scritto job **C#/.NET** che legge Work Item (ADO API), genera markdown e fa push su repo GitHub Pages.
- Implementato autenticazione GitHub con PAT rotating e gestione branch/pull-request.
- Abilitato feedback utenti via **GitHub Issues** collegati alle release note.
- Integrata traduzione automatica (Google Translation API) e **full-text search** JS sul sito.

- Guidato sviluppatore junior nella UI Vue/Bootstrap e nel CI/CD con GitHub Actions.
Risultati: note di rilascio sempre aggiornate (build nightly) • visibilità cross-team • gestione feedback centralizzata.
Stack: C#/.NET • Azure DevOps REST API • GitHub Pages & Actions • Google Translate API

Internal Package Manager (NuGet-like) • 2016 – Presente

Contesto: Necessità di un repository binari versionato per 280 + moduli ($\approx 1,4$ M LOC) a supporto di 6 + software core; doveva semplificare dipendenze, branching e build installer.

Ruolo: Platform Supervisor & Maintainer – definizione policy, direzione dev principale, governance e manutenzione evolutiva.

Azioni:

- Specificato architettura e naming-convention **Semantic Versioning** per pacchetti interni.
- Automazione full-stack (C#/.NET): push binari, creazione branch release, generazione componenti e bundle di deploy.
- Pipeline CI/CD per build installer (MSI/EXE) con tagging automatico.
- Gestione permessi granulari e audit log su repository binari.
- Documentazione e training per team Dev & QA (280 moduli migrati).

Risultati: adozione 100 % moduli interni • dipendenze risolte in < 2 s • ridotti errori di build correlati a versioning.

Stack: c++, C#/.NET • NuGet Protocol • Git/TFVC • Azure DevOps Pipelines • PowerShell

Installer Governance Suite • 2008 – Presente

Contesto: Uniformare la distribuzione di 6 + linee software su Windows (client & server) con standard di installazione coerenti, rollback sicuri e firma digitale.

Ruolo: Installer Governance Lead – linee guida, template MSI/EXE, supervisione policy, manutenzione continua.

Azioni:

- Definite best-practice e template **WiX Toolset** per nuovi installer; migrazione graduale da InstallShield legacy.
- Libreria script **PowerShell** per prerequisiti, servizi Windows e registrazione componenti.
- Gestito pipeline CI/CD: build MSI, creazione auto-extracting ZIP/EXE, firma e pubblicazione.
- Revisioni codice installer e coaching team prodotto sulle policy di packaging.
- Supporto e fix post-release → tracking issue e miglioramento continuo.

Risultati: standardizzazione 90 % installer portfolio • incidenti installazione ridotti • time-to-release allineato al ciclo DevOps.

Stack: WiX Toolset • InstallShield (legacy) • C# • MSI • Auto-extracting EXE • PowerShell • Windows Services

Cloud Installer (VS-Installer-like) • 2019 – Presente

Contesto: Downloader/launcher centralizzato per 5 software core; gestione versioni, notifiche update e retry su errori di download (SFTP → locale).

Ruolo: **Solution Co-Owner & QA Lead** – raccolta requisiti, supervisione sviluppo collaboratore, definizione test automatici.

Azioni:

- Architettura **C# bootstrap WiX** (≈ 7 MB) + plug-in per ciascun prodotto (EULA, validazione licenza, opzioni installazione).
- Pipeline release → upload binari su archivio SFTP, metadati su DB Azure; pubblicazione automatica via DevOps.
- Implementato meccanismo checksum & retry → download sicuro anche su reti instabili.
- Notifiche in-app su nuove versioni con changelog e one-click install.
- Suite test automatici (install / uninstall / rollback) integrata in CI.

Risultati: roll-out centralizzato • tempo setup ridotto (installazione guidata unica) • adozione in corso nei team (5 prodotti già migrati).

Stack: C#/.NET · WiX Toolset Bootstrapper · SFTP · Azure SQL · Azure DevOps Pipelines

Software Support L3 Service Desk · 2017 – 2023

Contesto: Garantire assistenza di terzo livello al Service on-site su tematiche software, coordinando escalation urgenti e documentazione di troubleshooting.

Ruolo: **L3 Support Team Lead** – gestione team (3 persone), rapporto con Service & Dev.

Azioni:

- Istituita **shared mailbox** con regole di ingaggio e criteri automatici di assegnazione ticket.
- Definito triage oggettivo (fast lane vs. normal) e priorità SLA per urgenze cliente.
- Gestito reperibilità, ferie e load-balancing delle richieste; monitor chiamate Teams.
- Documentato troubleshooting ricorrente su wiki → filtro verso team sviluppo, riducendo escalation non necessarie.

Risultati: escalations a Dev -40 % • Service più autonomo sul campo.

Stack: Microsoft 365 Shared Mailbox · Microsoft Teams · Azure DevOps (Boards/Wiki) · PowerShell reporting

CAD/CAM Toolpath Optimisation & Simulation · 2007 – 2015

Contesto: Sviluppo CAM in C/C++ per lavorazioni vetro e pietra; ottimizzazione percorsi utensile e tempi macchina nei prodotti Cad/Cam dell'azienda.

Ruolo: **C/C++ Developer & Demo Lead** – coding core, profiling prestazioni, demo per clienti e fiere.

Azioni:

- Integrato libreria **ModuleWorks 5-Axis** per strategie multiasse avanzate.
- Scritte routine di ottimizzazione supporti/bloccaggio pezzi e smooth-path (arc, spline).
- Profiling con **AQtime** e JetBrains profiler → tuning hot-path CPU/mem.

- Implementata simulazione traiettorie in **OpenGL**.
- Gestite demo live e raccolta feedback durante fiere internazionali.
Risultati: tempi lavorazione –15 % • ridotto jerk macchina • qualità superficie migliorata (minori ripassi).
Stack: C/C++ • ModuleWorks 5X • OpenGL • AQtime • JetBrains Profiler

Documentation & Localization Workflow · 2021 – presente

Contesto: Centralizzare redazione manualistica e traduzioni (UI strings + manuali) per i software del reparto di ingegneria, garantendo coerenza terminologica e tempi di pubblicazione.

Ruolo: Documentation & Localization Coordinator – pianificazione release doc, gestione traduzioni, sviluppo tool MT.

Azioni:

- Accentrata manualistica su un unico redattore tecnico; definito piano roll-out e milestone di consegna per ogni release.
- **Integrato nei Work Item** (Feature/Bug) un flag “Needs Translation/Manual” → dev indica requisito, trigger per ufficio lingue.
- Coordinato flusso traduzioni con ufficio linguistico (stringhe UI + PDF manuali).
- Sviluppato tool di **traduzione automatica** con glossario proprietario per fallback in attesa della versione ufficiale.
- Allineati sprint Dev ↔ doc tramite board DevOps; versioning doc in Git per PR e code-review.
Risultati: time-to-publish manuali –30 % • coerenza terminologia ↑ • nessun ritardo release dovuto a doc.
Stack: Git • Azure DevOps (Boards/Wiki) • MT Engine + glossario • PowerShell automation

Manual Testing & Validation Governance · 2020 – Presente

Contesto: Organizzare validazioni funzionali su 6 + prodotti usando Test Plans / Test Cases di Azure DevOps; garantire checklist ripetibili e metriche qualità build.

Ruolo: Test Team Supervisor & Mentor – coordinamento QA interni/esterni, linee guida, reporting qualità.

Azioni:

- Definite naming-convention e template bilingue (IT/EN) per Test Case facilmente condivisibili.
- Classificate configurazioni macchina e ambienti (OS, driver, HW) → mapping one-to-many con Test Suites.
- Automatizzato report qualità build (pass/fail, blocker) inviato a Dev & PM ogni sprint.
- Mentoring e crescita del team QA (skill matrix, sessioni pair-testing).
- Gestito backlog bug da validazione manuale ↔ sviluppo; integrazione con Work Item e dashboard.

Risultati: copertura test manuali in crescita costante • time-to-feedback ↓ **Stack:** Azure DevOps Test Plans • Test Suites & Configurations • Power BI Reports • Microsoft Teams

VM & Security Governance Service · 2021 – Presente

Contesto: Gestione autorizzazioni software/AD, certificati, password vault e health-check VM per l'infrastruttura Dev & Build .

Ruolo: DevOps & Security Owner – policy accesso, monitoraggio spazio volumi AD, firma eseguibili.

Azioni:

- Monitor VM (CPU/Disk) + alert mail/Teams; script PowerShell on-prem.
- Gestito rinnovo certificati e rotazione password vault condiviso.
- Pipeline Azure DevOps per build agent + firma automatica EXE/MSI (signtool).
- Report capacità volumi AD con forecast trend.

Risultati: uptime VM > 99,9 % • incidenti storage evitati • processo firma zero-touch in CI/CD.

Stack: PowerShell · Azure DevOps Pipelines · AD DS · Windows CertSrv · signtool · treesize