

Fabio Dell'Onte

✉️ fabiodellonte@gmail.com

📞 +39 329 748 8632

📍 Pesaro, Marche, Italy

🎂 02 Mar 1982

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

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

About Me

I'm an optimistic, life-loving person who believes in keeping a healthy balance between professional dedication and the joy of living.

I welcome challenges while maintaining a consistently positive outlook. When the situation demands it, I become laser-focused and determined, channeling all my energy into meeting objectives.

I have a deep faith in people and their potential, always aiming to bring out the best in those around me.

Above all, I'm strongly committed to truth, transparency, and integrity in every aspect of life—values I see as pillars of both personal and professional relationships.

Passionate about technology and innovation, I have 15+ years of experience in software development and team leadership.

Throughout my career I've successfully managed complex projects and led high-performance teams, always focusing on delivering quality solutions and fostering a collaborative, growth-oriented environment.

I remain hands-on, often prototyping solutions before transferring knowledge and responsibility to the team.

Experience

Software Operations Manager · Biesse

Jan 2023 – Present

- SDLC / Application Lifecycle Management (Agile, DevOps, CI/CD)
- Software configuration management (SCM)
- Software-protection development
- Cloud-infrastructure development
- Technical relationships with suppliers

- Technical-team coordination

Process and Test Manager · Biesse

Dec 2016 – Dec 2022

- ALM-process development and test automation
- Cloud integration for user-data analytics
- Tooling for software infrastructure and distribution
- Internal / external technical relationship management
- Technical-team coordination

Software Team Leader · Biesse

Mar 2012 – Nov 2016

- Technical lead for CAD/CAM software (glass & stone)
- Post-processor and vertical-machining software development
- Software validation and QA
- Software demos and presentations at industry trade fairs

Software Developer · Biesse

2007 – Feb 2012

- Integration of ModuleWorks 5-axis machining libraries
- CAM toolpath generation and optimisation
- CAD/CAM diagnostics
- Code-performance tuning

Skills

Languages: C#, C++, C, Python, TypeScript, PowerShell, .NET Framework, STL, Qt

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

Focus Areas: CAD/CAM, Windows desktop apps & services, remote services, troubleshooting, problem solving, project management, process automation, web apps, OpenGL

Data & Analytics: Elastic Stack, Microsoft Power BI, Excel

IDEs: Visual Studio 2022, VS Code, Cursor AI

AI & ML: Ollama, OpenAI ChatGPT/API, Anthropic Claude, Meta LLaMA, Microsoft Phi

Education

Alma Mater Studiorum – University of Bologna · MSc in Computer Science
2005 – 2007 · Grade: 110/110 cum laude
Thesis: *Adaptive Tessellation for Subdivision Surfaces in XCSubd*

University of Urbino “Carlo Bo” · BSc in Applied Computer Science
2001 – 2004 · Grade: 109/110

ITIS “Enrico Mattei” – Urbino · Electronics & Telecommunications Diploma
1996 – 2001 · Grade: 100/100

Languages

- Italian: native
 - English: intermediate
-

Interests

Family, personal & team growth, technology, automation, AI, running

Key Projects and Managed Services

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

Context: Securely expose machine data collected by the **Monitoring Platform**; the edge agent registers with the Kaos server (Kanagawa Cloud) and enables the **Flexy Frame** React app for remote monitoring and commands.

Role: **Cloud Solution Architect & Lead Developer** – micro-service design, TypeScript/Node coding, edge↔cloud integration, CI/CD.

Architecture:

- **Non-containerised micro-services** deployed on Linux servers in the internal datacentre.
- Internet exposure via **Nginx reverse proxy / HTTPS**.
- **WebSocket gateway (IRIS)** for real-time streaming.

Actions:

- Developed REST + WSS APIs in **Node.js/TypeScript**; secure handshake edge-agent → token → dedicated channel.

- Built **Flexy Frame** (React) front-end for live KPIs and remote commands.
- GitHub Actions pipeline for continuous build, test, and deployment on the Kanagawa cluster.
- **Elastic APM** instrumentation + uptime/performance dashboards.
Results: One-click machine onboarding • 100% traceability of remote sessions.
Stack: Node.js · TypeScript · React · WebSocket · Nginx · IRIS protocol · Elastic APM · GitHub Actions · VS Code / Cursor

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

Context: Full-stack telemetry for 6+ applications: logs, performance, usage analytics streamed in real time to an on-prem Elastic cluster.

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

Actions:

- C# service installed via WiX bootstrap that instantiates a plug-in agent (IoC container) at app start-up.
- Collects logs, DB queries, CPU/RAM counters, Windows events, and action tracking; ships via **Filebeat** to **Elasticsearch**.
- **Kibana** dashboards for crash trends, health, and usage KPIs; ‘Kanagawa Cloud’ gateway for selective external streaming.
- Agent auto-update from **Azure Blob Storage**; REST polling channel for remote commands (debug/patch).
Results: 24×7 observability on 6+ products • faster bug diagnosis • zero-touch update roll-outs.
Stack: C#/.NET · IoC plug-in · Filebeat · Elasticsearch · Kibana · Azure Blob Storage · Azure DevOps

GenAI & LLM Experimentation Suite · 2024 – Present

Context: Bring generative AI into the engineering software ecosystem, from Kaos infrastructure to R&D.

Role: AI Integrator (PoC) – model integration, proofs of concept, knowledge sharing.

Highlights:

- Integrated **OpenAI GPT-3.5/4** and self-hosted models (**Ollama**, **DeepSeek**, **Claude**, **Gemini**) into Kaos channels.
- Experiments in **fine-tuning**, **RAG**, and prompt-engineering for internal agents.
- Rapid development with **FastAPI** / **Node** + CI/CD deploy; demo front-end **React (Flexy Frame)** on Netlify.
- Daily use of **Cursor AI** and GitHub Copilot.

Results: Standard pipeline for custom models; GenAI sandbox adopted by Dev & QA teams; foundation for support chatbot and internal code-assist.

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

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

Context: Protect IP and manage licensing across the company's software portfolio; transition from Gemalto HASP dongles to steganographic (PNG) software licences.

Role: License & Security Lead – KeyProgrammer maintenance, licensing policy, LKLib evolution.

Actions:

- Maintained KeyProgrammer tool (C#) for HASP key generation on internal VPBs; user & policy management.
- 2019: integrated external PNG-steganography library → developed reusable LKLib (C#, multilayer obfuscation).
- Ported licensing API to 6 + products; plug-in module (HW/SW) with runtime choice.
- Automated licence provisioning via DevOps pipelines and usage dashboard.
- Handled key support & renewals, migrated users from dongles to soft licences.

Results: 60% of software migrated to SW licence • dongle costs down • online activation < 10 s • enhanced IP protection.

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

Energy Monitoring System (OPC UA) · 2024 – Present

Context: Collect and store machine energy-consumption metrics via OPC UA for continuous analysis and efficiency optimisation.

Role: Lead Developer – design, C++/Qt coding, OPC-UA integration.

Actions:

- Implemented OPC UA poller acquiring energy KPIs and persisting them in a historical store.
- Qt UI on Windows for live charts and periodic queries (day / week / month).
- Hot-reloadable JSON config for machine settings and alert thresholds.
- Local service logging and status dashboard.

Results: Continuous consumption monitoring • historical data foundation for efficiency analysis.

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

Test Automation Platform & VM Farm Orchestrator · 2015 – Present

Context: End-to-end test automation for all core software products, integrating execution and code-coverage into CI/CD pipelines.

Role: Test Automation Architect & Lead – framework conception, tool development, pipeline governance.

Actions:

1. Integration Test (ISO G-Code textual diff)
2. UI Record & Playback for WPF apps
3. Code-Coverage (dotCover, VS Coverage)

- Orchestrated workload balancing on a VM farm: agent distributes code queues from pipeline jobs.
- Test-queue management: priorities, retry, HTML report with coverage metrics pushed into CI.
- Instrumented C++/C# and uploaded reports to DevOps dashboards.
Results: 50 % regression tests automated • coverage rising steadily year-on-year.
Stack: C#/NET · C++ · WPF · JetBrains dotCover · Visual Studio Coverage · Azure DevOps Pipelines · PowerShell

Diagnostic Center · 2021 – 2022

Context: Reduce manual troubleshooting for the Service team: local tool executing health checks (registry, files, DB, ...) via per-app plug-ins.

Role: Concept Owner & Requirements Lead – idea, spec gathering, development oversight.

Actions:

- Designed modular architecture: each team provides a C# plug-in with a series of tests.
- Standardised check templates (registry key, file presence, DB config).
- Guided dev team on UI and parallel-execution engine.
- Documentation & training for Service rollout.

Results: Automated troubleshooting • decreased ticket-resolution times.

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

Release Planning & Orchestration Service · 2017 – Present

Context: Coordinate multi-product releases with dedicated branch strategy and Test Team, ensuring shared calendar and official release notes.

Role: Release Coordinator & Automation Lead – planning, reminders, branch management, publishing.

Actions:

- Built Azure DevOps queries & dashboards for release calendar; automatic end-of-dev reminders to Dev & QA.
- Managed branch flow: feature → release, hotfix creation, duplicated test suite per branch.
- Orchestrated test execution with Test Team, build validation, and release signing.
- Automated **Release Notes** generation & publication (Markdown → Git).
- Synced Dev, QA, PM via email + Teams; permission governance on pipelines & repos.

Results: > 95 % releases on time.

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

License Provisioning & Cost-Control Suite · 2021 – Present

Context: 15 + strategic software tools (IDEs, DevOps, productivity) used by > 250 internal users.

Role: Service Owner – budgeting, contract renewals, cost optimisation.

Actions:

- Centralised licence procurement and credential vault.
- Implemented budget-forecast model based on actual consumption.

Results: TCO –17 % • onboarding time –50 % • 100 % licence visibility.

Stack: Power BI · Azure DevOps · GitHub API · PowerShell · Elastic Stack

VM & Physical Lab Infrastructure · 2019 – Present

Context: 220 + VMs and 20 + physical PCs for automated/manual tests, CI build farm, and externally exposed services.

Role: Infrastructure Owner – capacity planning, monitoring, security hardening.

Actions:

- Standardised images & provisioning with **PowerShell** scripts.
- Built a cluster build farm (Azure DevOps agents auto-scaling, Jenkins).
- Managed patching and incremental backups.

Results: Uptime > 99.9 %.

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

SCM & Repository Governance Suite · 2016 – Present

Context: Central governance of Azure DevOps (on-prem & SaaS), GitHub Enterprise, SVN, Jenkins, TFVC/Git – > 250 internal users + external suppliers.

Role: SCM Service Owner – commit/branch policies, permission control, servers & licences, vendor coordination.

Actions:

- Defined & enforced standard branch policies (mandatory PR, code review, build gate).
- Proactive server maintenance (patches, backups, monitoring) → **high availability** for build & release.
- Managed contracts/licences and external-supplier onboarding.

Results: Uptime > 99.9 % • 100 % audit-compliant repo access • high-touch service for 250 + users.

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

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

Context: Align software-development process with corporate & departmental directives across 20 + Azure DevOps projects via customised Work Items and integration with corporate services.

Role: SDLC Process Architect – WI fields/rules design, integrations, team training, security & access

management.

Actions:

- Customised WI fields, states, and rules for multiple project collections.
- Integrated Work Items with Service Desk and Power BI (quality-KPI reporting).
- Authored process guidelines on wiki, trained 200 + dev/QA staff.
- Developed Work Item-lint tool and compliance notifications via API.
- Managed access levels and permissions for internal and external teams.

Results: 100 % process adoption • audit-ready governance.

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

Power BI DevOps Software-Development Dashboards Suite · 2022 – Present

Context: Need for unified insight on backlog, releases, and team productivity; data extracted from Azure DevOps and shared across business functions.

Role: BI Implementer & Stakeholder Liaison – report development, Git versioning, permission governance.

Actions:

- Built OData analytics connectors → Power BI for work-item, build & release data.
- Versioned reports in Git; CI pipeline for automatic publishing to Power BI workspace.
- Defined permission model for audiences (Dev, QA, Governance, Release) with row-level security.
- Published dashboards on team productivity KPIs, lead time, PR aging, release calendar.

Results: Data-driven decision-making for 5 + departments • manual Excel reports reduced (– 90 h/month) • > 80 % team adoption.

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

DevOps Integration Bridge (Salesforce – Teamcenter – Azure DevOps) · 2018 – Present

Context: Real-time synchronisation of Problem Reports (PR) between Salesforce, PLM Teamcenter, and Azure DevOps Work Items; previously manual alignment (~ 15 min/PR).

Role: Integration Engineer & Service Owner – design, C# development, deploy & API-permission management.

Actions:

- Developed C#/.NET micro-service converting and publishing PRs via filesystem protocol to the three systems.
- Implemented resilient retry, field mapping, and WI ↔ PLM link management.
- CI/CD pipeline for build and rollout on Windows Server host.

Results: 3 PR/day synchronised instantly • +187 h/year saved • 100 % data alignment among 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 – Present

Context: Automatically transfer hours logged in Azure DevOps Work Items to Oracle Time & Labor; previously 250 + people spent ≈ 10 min/week each on manual entry.

Role: Product Owner & Maintainer – requirements gathering, project management, continuous maintenance.

Actions:

- Led primary developer (internal team) and defined functional backlog.
- Implemented C#/.NET micro-service: polls Azure DevOps API, transforms data, pushes to Oracle OTL.
- Managed deployment, monitoring, and OAuth/API keys.
- Introduced automatic hour validation and emailed result reports.

Results: 20 min/user/week eliminated → ≈ 2000 h/year saved • error-free entries • 100% team adoption.

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

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

Context: Ensure Scrum-process compliance on 2 Azure DevOps projects via 30 + automatic rules (AreaPath, Sprint, Business Value, linkage, test coverage, etc.).

Role: Lead Developer & Maintainer – requirements gathering, plug-in architecture, permission & mailing governance.

Actions:

- Developed nightly-scheduled C#/.NET service: runs validations, aggregates results per team.
- Open architecture: new rules pluggable via JSON & DLL without downtime.
- Reports emailed with issue list & WI links; personalised RLS for 10 + teams.
- Exception handling: reply-mail → auto-ticket to Operations Team.

Results: 100% Work Items checked nightly • proactive corrections → fewer process errors • manual audit hours saved.

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

Release Notes Portal (GitHub Pages) · 2022 – Present

Context: Make release notes of 6 main software products public and searchable; automate extraction of *Release Notes* field from Azure DevOps Work Items and publication on GitHub Pages.

Role: Solution Owner & Lead Developer – requirements collection, DevOps integration, junior-dev mentoring.

Actions:

- Wrote C#/.NET job that reads Work Items (ADO API), generates markdown, and pushes to GitHub Pages repo.
- Implemented GitHub authentication with rotating PAT and branch/pull-request management.
- Enabled user feedback via **GitHub Issues** linked to release notes.
- Integrated automatic translation (Google Translation API) and **full-text search JS** on the site.
- Guided junior dev on Vue/Bootstrap UI and CI/CD with GitHub Actions.

Results: Release notes always up to date (nightly build) • cross-team visibility • centralised feedback management.

Stack: C#/.NET · Azure DevOps REST API · GitHub Pages & Actions · Google Translate API

Internal Package Manager (NuGet-like) · 2016 – Present

Context: Versioned binary repository for 280 + modules (~ 1.4 M LOC) supporting 6 + core software products; simplify dependencies, branching, and installer builds.

Role: Platform Supervisor & Maintainer – policy definition, lead dev direction, governance, evolutionary maintenance.

Actions:

- Specified architecture & **Semantic Versioning** naming convention for internal packages.
- Full-stack automation (C#/.NET): binary push, release branch creation, component & deploy-bundle generation.
- CI/CD pipeline for MSI/EXE installer builds with automatic tagging.
- Granular permissions and audit log on binary repository.
- Documentation & training for Dev & QA teams (280 modules migrated).

Results: 100% internal-module adoption • dependencies resolved in < 2 s • fewer build errors related to versioning.

Stack: C++ · C#/.NET · NuGet protocol · Git/TFVC · Azure DevOps Pipelines · PowerShell

Installer Governance Suite · 2008 – Present

Context: Standardise distribution of 6 + software lines on Windows (client & server) with coherent install standards, safe rollback, and digital signing.

Role: Installer Governance Lead – guidelines, MSI/EXE templates, policy supervision, continuous maintenance.

Actions:

- Defined best practices and **WiX Toolset** templates for new installers; gradual migration from legacy InstallShield.
- PowerShell script library for prerequisites, Windows services, and component registration.
- Managed CI/CD pipeline: MSI build, auto-extracting ZIP/EXE creation, signing, and publication.
- Installer-code reviews and product-team coaching on packaging policies.

- Post-release support & fixes → issue tracking and continuous improvement.

Results: 90% installer portfolio standardised • fewer installation incidents • time-to-release aligned with DevOps cycle.

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

Cloud Installer (VS-Installer-like) · 2019 – Present

Context: Central downloader/launcher for 5 core software products; version management, update notifications, and retry on download errors (SFTP → local).

Role: Solution Co-Owner & QA Lead – requirements collection, collaborator supervision, automated-test definition.

Actions:

- **C# WiX bootstrap** architecture (~ 7 MB) + plug-in per product (EULA, licence validation, install options).
- Release pipeline → binary upload to SFTP archive, metadata to Azure DB; automatic publication via DevOps.
- Checksum & retry mechanism → safe downloads even on unstable networks.
- In-app notifications on new versions with changelog and one-click install.
- Automated test suite (install / uninstall / rollback) integrated into CI.

Results: Centralised rollout • reduced setup time (single guided installer) • adoption in progress (5 products migrated).

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

Software Support L3 Service Desk · 2017 – 2023

Context: Provide third-level support to on-site Service for software issues, coordinating urgent escalations and documenting troubleshooting.

Role: L3 Support Team Lead – managed team (3 people), liaison with Service & Dev.

Actions:

- Established **shared mailbox** with engagement rules and automatic ticket assignment.
- Defined objective triage (fast lane vs normal) and SLA priorities for customer urgency.
- Managed on-call roster, holidays, and load balancing; monitored Teams calls.
- Documented recurring troubleshooting on wiki → reduced unnecessary dev escalations.

Results: Dev escalations –40% • more autonomous Service in the field.

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

CAD/CAM Toolpath Optimisation & Simulation · 2007 – 2015

Context: CAM development in C/C++ for glass & stone machining; toolpath optimisation and cycle-time reduction in company CAD/CAM products.

Role: C/C++ Developer & Demo Lead – core coding, performance profiling, demos for customers & trade fairs.

Actions:

- Integrated **ModuleWorks** 5-Axis library for advanced multi-axis strategies.
- Wrote routines for support/fixture optimisation and smooth-path (arc, spline).
- Profiling with **AQtime** and JetBrains profiler → tuned CPU/memory hot paths.
- Implemented trajectory simulation in **OpenGL**.
- Managed live demos and feedback collection at international fairs.

Results: –15 % machining time • reduced machine jerk • improved surface quality (fewer re-passes).

Stack: C/C++ · ModuleWorks 5X · OpenGL · AQtime · JetBrains Profiler

Documentation & Localization Workflow · 2021 – Present

Context: Centralise documentation and translations (UI strings + manuals) for engineering software, ensuring terminology consistency and publication tempo.

Role: Documentation & Localization Coordinator – doc-release planning, translation management, MT-tool development.

Actions:

- Centralised manuals under one technical writer; rollout plan & delivery milestones per release.
- Added “Needs Translation/Manual” flag to Work Items → dev indicates requirement, triggers languages office.
- Coordinated translation flow with language office (UI strings + PDF manuals).
- Developed **machine-translation tool** with proprietary glossary as interim fallback.
- Synced Dev sprints with docs via DevOps board; doc versioning in Git for PR & code review.

Results: –30 % time-to-publish manuals • ↑ terminology consistency • no release delays due to docs.

Stack: Git · Azure DevOps (Boards/Wiki) · MT engine + glossary · PowerShell automation

Manual Testing & Validation Governance · 2020 – Present

Context: Organise functional validations on 6+ products using Azure DevOps Test Plans/Cases; ensure repeatable checklists and build-quality metrics.

Role: Test Team Supervisor & Mentor – coordination of internal/external QA, guidelines, quality reporting.

Actions:

- Defined bilingual naming conventions & templates (IT/EN) for easily shareable Test Cases.
- Classified machine configurations & environments (OS, drivers, HW) → one-to-many mapping with Test Suites.
- Automated build-quality report (pass/fail, blockers) emailed to Dev & PM each sprint.

- Mentored QA team growth (skill matrix, pair-testing sessions).
- Managed bug backlog from manual validation ↔ development; integrated with Work Items and dashboards.
Results: Manual-test coverage steadily increasing • time-to-feedback ↓
Stack: Azure DevOps Test Plans · Test Suites & Configurations · Power BI Reports · Microsoft Teams

VM & Security Governance Service · 2021 – Present

Context: Manage software/AD authorisations, certificates, password vault, and VM health checks for Dev & Build infrastructure.

Role: DevOps & Security Owner – access policies, AD-volume monitoring, executable signing.

Actions:

- Monitored VMs (CPU/Disk) + mail/Teams alerts; on-prem PowerShell scripts.
- Managed certificate renewals and shared-vault password rotation.
- Azure DevOps pipeline for build agents + automatic EXE/MSI signing (signtool).
- Volume-capacity reports with forecast trend.

Results: VM uptime > 99.9 % • storage incidents avoided • zero-touch signing process in CI/CD.

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