Case Study — Velocity Digital Agency

How Velocity turned “Closed Won” into same-day project starts (and saved 317 hours in 90 days)

> “We stopped ‘starting projects.’ Now projects start themselves.”
> — Ops Lead, Velocity (internal recap)

The moment that exposed the real problem

Velocity didn’t have a sales problem.

They had a **handoff problem**.

Sales would close a deal in HubSpot.

Everyone would celebrate.

Then… the silence.

Not because anyone was lazy.
Because the next steps lived inside people’s heads.

And anything that lives inside people’s heads eventually gets dropped.

The business

Client: Velocity Digital Agency
Industry: Full-service digital marketing agency
Deal volume: 12–15 new deals per month (HubSpot)
Team reality: fast sales, overloaded delivery

Velocity was growing.
But growth was quietly creating a new kind of debt.

Coordination debt.

Every new client added another round of manual setup.

And manual setup never stays small.

It multiplies.

The old way (what happened after Closed Won)

Here's what "delivery onboarding" really meant:

- Someone copies company details from HubSpot into a spreadsheet "so ops can see it."
- Someone creates a project page in Notion.
- Someone tries to remember what tasks should exist for that service.
- Someone sends a welcome email (late) because they didn't know it was their job.
- Someone pings Slack: "Did anyone set up the workspace?"
- Someone hopes nothing was missed.

That "someone" changed depending on the week.

So the outcome changed too.

Sometimes it went smoothly.

Sometimes it didn't.

But every time, it cost time.

2.5 hours per client.

At 12–15 deals per month, that's **30–37 hours/month** of pure coordination.

And that was only the visible cost.

The real cost (the part nobody puts on a timesheet)

The time hurt.

But the lag hurt more.

Deals would close on Friday.

Projects would start Tuesday.

And in that gap:

- Clients wondered if they made a mistake.
- PMs started the relationship on the back foot.
- Sales had to "check in" just to keep the client warm.

- The delivery team opened Monday already behind.

Velocity wasn't losing deals.

They were losing *momentum.*

And momentum is what makes a client feel confident.

The goal

Velocity didn't ask for "automation."

They asked for something simpler:

"When a deal closes, can the project just... start?"

Same day.

Same structure.

Same checklist.

Same notifications.

Every time.

No heroics.

No memory required.

The solution (FlowForge Systems)

We built a complete **HubSpot → n8n → Notion** sales-to-delivery automation system, with an AI context layer.

Core trigger

When a deal moves to **Closed Won** in HubSpot:

1. HubSpot sends a webhook
2. n8n runs a workflow (32 nodes)
3. Notion becomes the delivery OS (databases updated/created)
4. The team + client get notified

5. All communications are logged

The part most “automation” projects skip

Velocity didn’t need a single integration.

They needed the **mechanism**:

- Deduplication (no duplicate clients or mismatched records)
- Routing logic (what service type? what checklist?)
- Reliable database creation
- AI summarization to turn messy deal notes into usable project context
- Logging so everything is traceable later

This is what makes automation trustworthy.

Not “it ran once.”

But “it runs correctly every time.”

What we built (technical implementation)

1) HubSpot trigger

- **HubSpot webhook** fires when deal status changes to **Closed Won**.

2) Orchestration in n8n (32 nodes)

- Validates payloads
- Checks if the client already exists (matching algorithm)
- Prevents duplicates
- Routes by service type
- Handles retries and error paths
- Sends notifications and logs outcomes

3) Delivery OS in Notion (4 databases)

Clients Database

- Name
- Domain
- Website

- HubSpot Company ID
- Status
- Relation: Projects

Projects Database

- Client relation
- HubSpot Deal ID
- Deal Amount
- Status
- Start Date
- Project Type

Tasks Database

- Task Name
- Project relation
- Status
- Priority
- Assignee
- Due Date

Communications Log

- Subject
- Project relation
- Client relation
- Type
- Recipient
- Date Sent

4) AI context layer (Claude/GPT)

- Reads deal notes and relevant fields
- Produces a concise project summary:
    - what was sold
    - what matters most
    - key deadlines or constraints
    - risks or dependencies

5) Notifications & comms

- Automated welcome email sent to the client

- Slack message to **#new-projects**
- Communications logged to the correct client + project

The before/after (what changed operationally)

Before

Closed Won created a new job:

> "Start the project."

That job required:

- attention
- time
- memory
- coordination
- multiple tools
- multiple people

So it was always delayed.

After

Closed Won created an outcome:

> "Project is ready."

Same day.

Same structure.

Same checklist.

Same notifications.

No waiting for someone to "have time."

Results after 90 days (measured, not guessed)

- **Zero manual project setup** (was 2.5 hours per client)
- **127 clients processed** through the system

- **317 hours saved** in 3 months
- Project start time: **same day as deal close** (was 2–4 days)
- Client satisfaction scores **increased 23%**
- Operations team freed to focus on delivery (not coordination)
- Agency scaled from **12 → 19 deals/month** without hiring coordinators

> “Our Fridays used to create Monday problems. Now a deal closes and the team already knows what to do.”
> — Founder, Velocity (post-deployment)

Timeline, investment, ROI

- **Timeline:** 3 weeks (kickoff → deployment)
- **Investment:** $4,500
- **ROI:** 7.2 weeks

The ROI calculation was straightforward:

- Remove 2.5 hours/client
- Multiply by monthly clients
- Multiply by ops hourly cost
- Add the invisible gains: faster starts, better client confidence, fewer internal interruptions

Velocity didn’t just save time.

They removed a recurring operational failure point.

Why this worked (the actual lesson)

Velocity didn’t need more discipline.

They needed a system where discipline is built-in.

Because the problem was never “people not trying hard enough.”

The problem was:

- Too many tools
- Too many steps
- Too much manual transfer of context
- Too much reliance on memory

We made the mechanism visible.

We made it automatic.

And now, the moment revenue is booked… delivery begins.

What this looks like for you

If any of these sound familiar:

- Your deals close… and delivery “starts later”
- Tasks get rebuilt from memory
- Clients wait while your team “sets things up”
- Your ops team lives inside Slack pings
- One coordinator holds the process together

Then you don’t have an onboarding process.

You have a bottleneck.

FlowForge removes that bottleneck.

So your team stops scrambling after Closed Won.

And starts delivering—immediately.