



Organizing some key points and ways that adilas is different. We came up with four layers. They are:

1. Product & Platform Differences
2. System Architecture Differences
3. Conceptual & Philosophical Differences
4. Company & Cultural Differences

Layer 1: Product & Platform Differences

Unified Operational Platform

- All-in-one system
- Systemized approach
- Operations lead accounting
- Mixes CRM, POS, ERP, Scheduling, Inventory, Accounting, Reporting, etc.
- Month-to-month subscription
- No long-term contracts
- Non-industry specific

Flexibility & Adaptability

- Built-in customization
- Custom code on top of the platform
- Permissions, settings, and templates
- Match customer processes
- Speak their language
- Able to adapt out of the box

Searchability & Data Access

- All Data Is Live And Searchable
- Business intelligence from everyday use
- Deep historical access
- Confidence in past data
- Peace of mind
- Paperless office
- Photos, documents, media, content

Documentation & Knowledge Capture

- Scan and store anything
- Digital storytelling
- Historical record keeping
- Objects and data over time
- Life cycles
- Media-rich records

Layer 2: System Architecture Differences

Operations First Architecture

- Operations lead accounting
- Different advanced accounting methods
- Data assembly line
- Transaction-first design
- Accounting becomes a byproduct of operations

- Most systems start with accounting and build outward. Adilas starts with business activity and lets accounting emerge naturally.

Connected Data Model

- Main Players
- Flex Grid
- Relationships between everything
- Cross-functional data connections
- Sub inventory
- Multiple engines
- This is the "city instead of buildings" concept. Most software creates isolated modules. Adilas creates relationships.

Results-Based Design

- Adilas Formula
- $\text{Mix}(\text{Functions} + \text{Players} + \text{Concepts}) = \text{Results}$
- Desired outcomes focus
- Systems thinking
- Core concepts
- This is unusual because most software is feature-centered. Adilas is result-centered.

Time-Based Architecture

- Time, as a category, often gets overlooked
- Elements of Time
- Overcoming batching
- Closing time gaps

- Objects over time
- Life cycles
- Historical tracking
- Event progression
- Many systems store data. Adilas stores the story of the data over time.

Multi-Level Visibility

- Bird's eye view to micro detail
- Flex zones
- Flex bubbles
- World building
- Layers of visibility
- People can zoom in or zoom out. Most software forces a single viewpoint. Adilas allows multiple levels of observation.

Layer 3: Conceptual & Philosophical Differences

World Building

- World building
- Ecosystem thinking
- Building operational worlds
- Connected structures
- Expandable environments
- Most software manages tasks. Adilas builds worlds.

Reality Simulation

- Physical actions are simulated into digital actions
- Reality first and software second
- Digital storytelling
- Business modeling
- Process modeling
- Concepts over code – we believe that the concepts are 10x to 100x more valuable than the code

Bridging Gaps

- Bridging operations and accounting
- Bridging people and data
- Bridging systems
- Bridging time gaps
- Bridging knowledge gaps

Systems Thinking

- Everything affects everything else
- Connected outcomes
- Relationships matter
- Cause and effect visibility
- Most software solves isolated problems. Adilas exposes underlying systems.

Layer 4: Company & Cultural Differences

Built By Dreamers

- "I wish it could do..."
- Listening to ideas
- User-inspired growth
- Idea farming
- Dreaming has been part of Adilas from the beginning.
- We are dreamers and encourage you to dream as well – you dream it up, we'll help you wire it up!

The Journey

- 20+ years of development
- Next logical step
- Patience
- Organic evolution
- Build and break
- Ship A → Ship B – new product vision and plans
- Unlike many startups, Adilas wasn't built around a funding event. It evolved through thousands of real operational problems.

Challenging Tradition

- Counter logic
- Challenging tradition
- Different accounting approaches
- Different architecture
- Different assumptions

- When asked about competition, one of the co-founders stated, "Tradition is our biggest competition".

Function Over Fashion

- Practicality
- Real-world use
- Utility over appearance
- Built for work

Protection By Sharing

- Sharing ideas
- Teaching concepts
- Open discussion
- Community building
- IP through expansion rather than secrecy