

“Responsible Customization: Stewarding a Shared Platform Like Rock RMS”

1. Opening Thought: With Great Power...

- Rock RMS gives us incredible flexibility—far more than most platforms.
 - But with that comes responsibility: to others, to the platform, and to the community (like my mom said, you tend to only pick-up your friends bad habits).
 - We are stewards, not owners. What we build lives beyond us.
-

2. The Reality of Limitations

- Every platform has limits—that’s not failure, it’s reality.
 - **Living within those limits** is often wiser than stretching or breaking them.
 - Short-term workarounds cause long-term pain.
-

3. Tempting but Dangerous Practices

- Injecting JS to override core behavior.
 - Hiding required UI with CSS instead of using settings or security.
 - Calling internal code that is not meant for public use.
 - Editing Rock core files.
 - Modifying the core database schema/indexes.
-

4. Why These Happen

- Business urgency.
- “It’s how we’ve always done it.”
- Internal pressure from staff or leadership.
- A belief there’s no other way.

But Rock offers a suite of customization options that, when used well, provide power without pain.

5. Healthy, Supported Customization Paths

- **Lava** for dynamic content and logic.
- **Block settings** for configuration and behavior tuning.
- **Workflows** to manage automation and business logic.
- **Plugins** for safe, upgradable extensions.
- **Helix/Lava Apps** for interactive, scalable custom features.

You often *can* do it the right way. And if you can't, you can request that it be added.

6. The Path to Product Change

- Rock is a responsive platform. Feature requests are heard.
 - But they go further when made in the spirit of shared benefit.
 - Use what's available. Then ask clearly for what's missing.
-

7. Working in the Spirit of the Platform

- Not every design choice will make sense immediately.
 - Many Rock features are designed with **long-term goals** or **future features** in mind.
 - Customizations should respect that direction—rather than fight it.
 - Often bad practices come from not fully understanding our tools.
-

8. Opinion vs. Architecture

- Many “must-have” requests are **opinionated**, not essential.
- These often come from internal clients or ministry teams.
- It's our job to **lead**, not just serve.

Leadership means coaching, mentoring, and **casting vision** before requests arrive.

9. Casting a Vision for Consistency

- When staff “demand hacks” or “refuse to use it unless X,” it often reveals a lack of:
 - Vision
 - Strategy
 - Technical leadership
- It’s tempting to blame leadership, but **the buck should stop with us.**
- If we aren’t leading clearly, we’ll always be reacting chaotically. If we’re sharing our vision in response to their request we are not proactively casting vision.

Start with a vision like:

Foundation Vision Statement

Because we desire to minister to our people in the most meaningful and effective way possible, we believe it is essential to truly know them. That level of personalization and care can only be achieved through a system that serves as a single source of truth—one platform that unifies data, insights, and ministry efforts.

To pursue this vision, we are committed to:

1. **Choosing integrated solutions** within the system, even when external tools may offer slightly better features but cannot share data seamlessly.
2. **Prioritizing holistic, unified strategies** over fragmented, niche alternatives that create silos and hinder long-term ministry impact.
3. **Investing in long-term stewardship** by building with consistency, scalability, and sustainability in mind.
4. **Leading with clarity**, coaching internal teams toward shared vision and understanding, so decisions are made with alignment and purpose.

This approach may require sacrifice and restraint, but it positions us to serve more effectively, respond more intelligently, and love more personally—because our technology strategy supports, rather than distracts from, our mission.

This is often best sold through stories.

1. At NewSpring Church, a family found peace after their daughter’s passing because a volunteer had recorded a note in the shared system about a recent salvation conversation.

That simple act ensured the family knew their daughter had accepted Christ—and gave them comfort in knowing she is in heaven.

2. At Church of the Highlands, a man was stopped from taking his own life because a follow-up process was faithfully carried out. The system triggered timely outreach, connecting with him at the exact moment he needed it most—saving his life.

These stories could not have happened without a central source of truth—a shared system that unified information and action. They were made possible because a few individuals chose to give up a personal preference or a small benefit in favor of a greater good: the opportunity to truly minister in life-changing moments.

When we hear it framed this way, that small thing our colleague insisted they couldn't live without suddenly feels trivial in comparison. This is certainly the case when they hear the vision ahead of their need and can buy into the vision before they have a lusting for that bright shiny object.

10. Counting the Real Cost

- Quick fixes today become tech debt tomorrow.
- They cost:
 - Time
 - Trust
 - Stability
- And they often fall on someone else—your future team, or another church.

Stewardship means considering the **total cost of ownership**, not just the immediate benefit.

11. Closing Challenge

- Every change we make is a legacy.
- Let's not just be clever administrator—let's be wise leaders.
- Be the voice casting a clear, consistent, and faithful digital vision.

- Let's be the master of our toolsets.