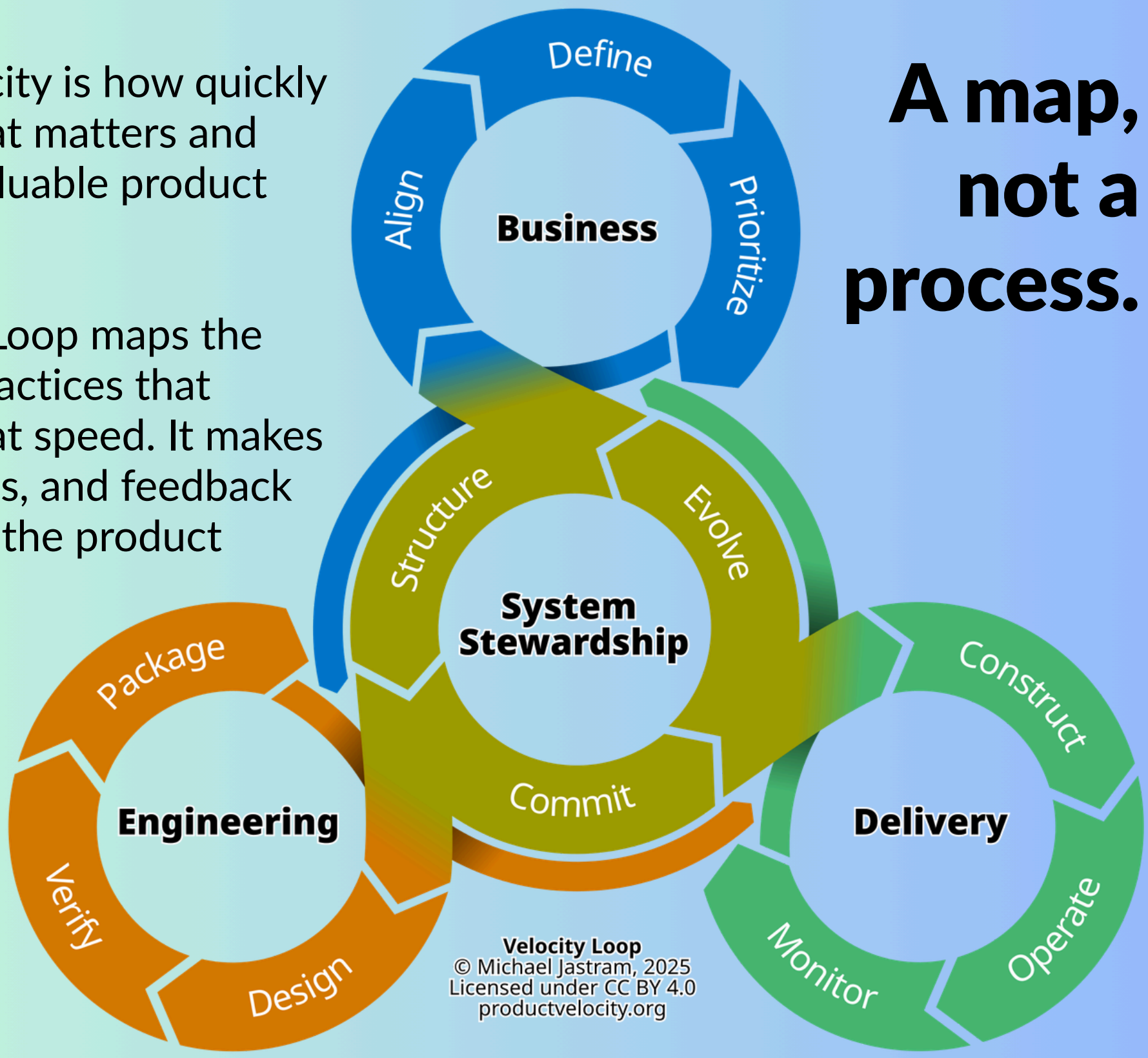
# The Velocity Loop

A system-level model for understanding and improving Product Velocity

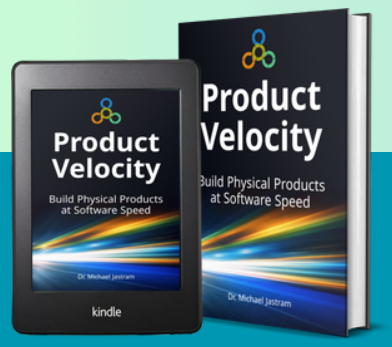
Product Velocity is how quickly you learn what matters and turn it into valuable product change.

The Velocity Loop maps the interacting practices that determine that speed. It makes flow, decisions, and feedback visible across the product system.




Velocity Loop
© Michael Jastram, 2025
Licensed under CC BY 4.0
productvelocity.org

**A map, not a process.**




www.productvelocity.org

Book by Michael Jastram

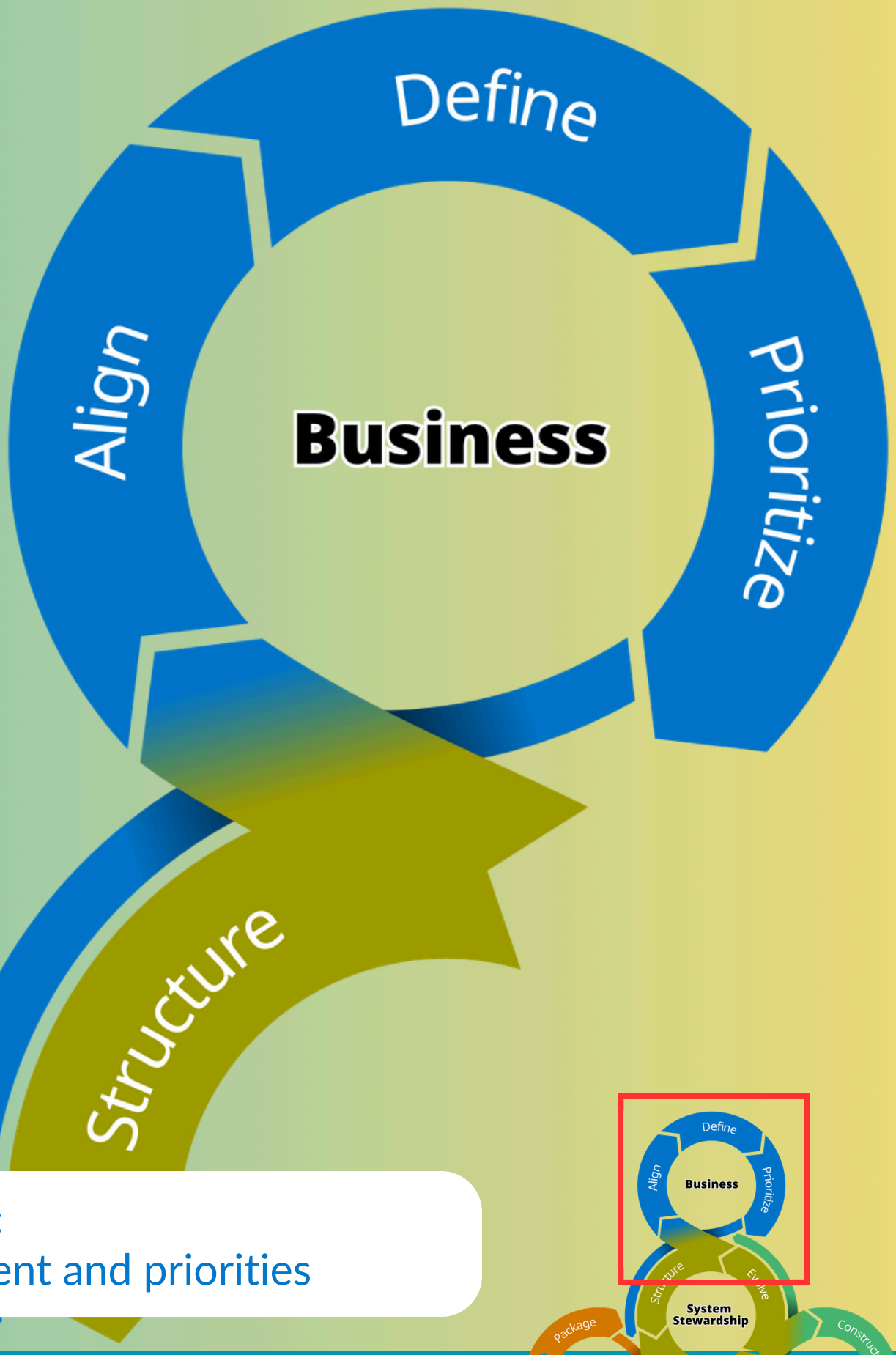
# Business: Decide What Matters

Translate stakeholder value into direction and priorities that can evolve.

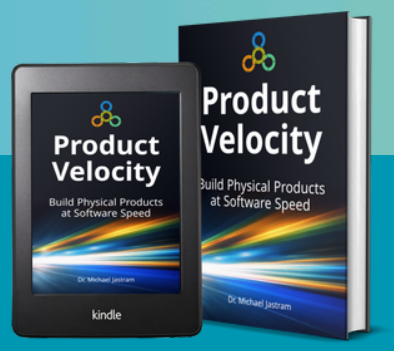
- **Align:** Build a shared understanding of whose needs matter and what creates value. Reconcile competing interests through explicit trade-offs.

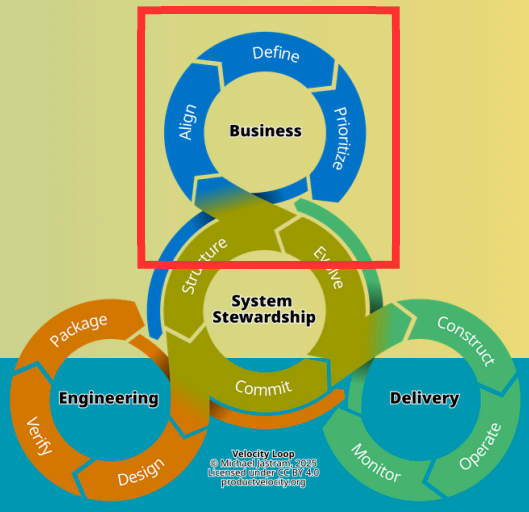
- **Define:** Turn this into value intent, constraints, and success criteria. Don’t prescribe the solution.

- **Prioritize:** Decide what matters most now. Set relative importance and sequencing so teams can act without constant approval.



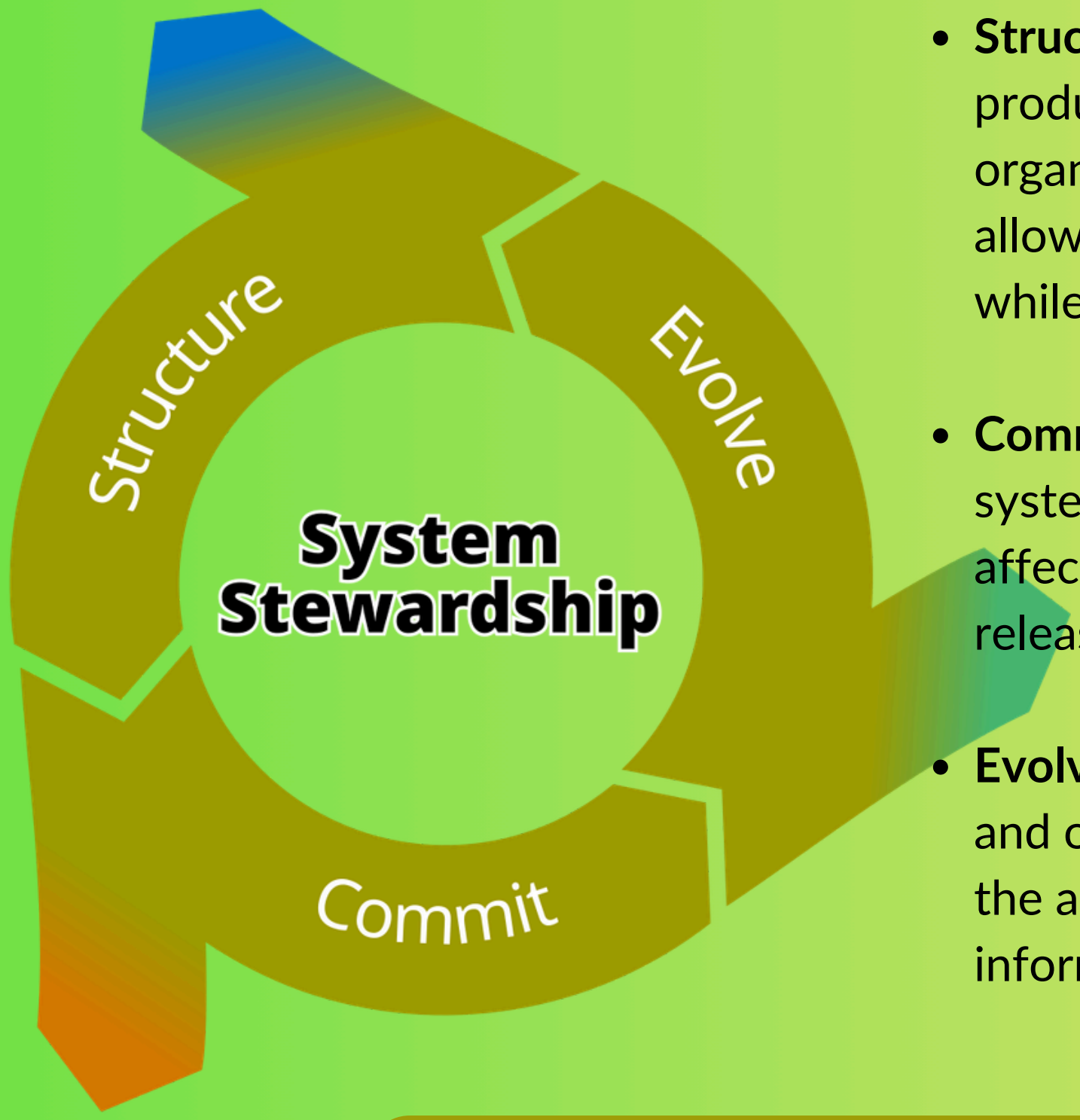

**Key Output:**
Evolving intent and priorities



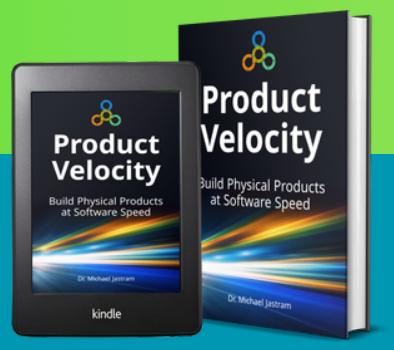
# System Stewardship: Coherence and Evolution

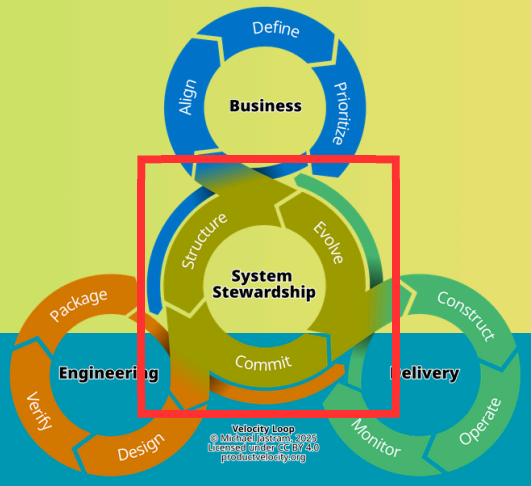
Maintain the shared structure that allows the product to change without losing system integrity.




- **Structure:** Establish and maintain product architecture and organizational boundaries that allow largely independent work while preserving coherence.
- **Commit:** Make and preserve system-level decisions that affect multiple domains, teams, releases, or lifecycle stages.
- **Evolve:** Incorporate engineering and operational evidence into the architecture and use it to inform future decisions.

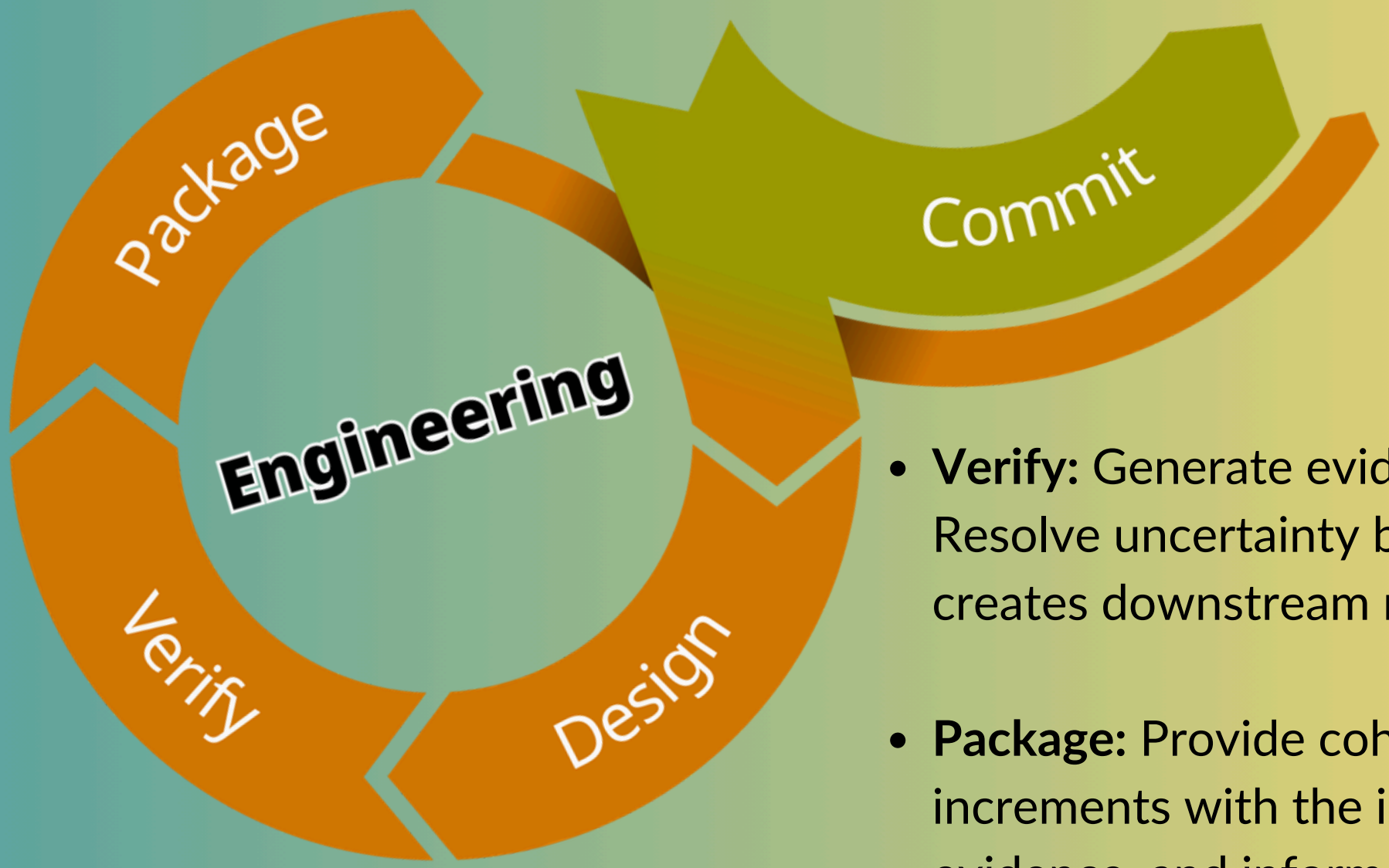
**Key Output:**
Shared structure, explicit commitments, system knowledge



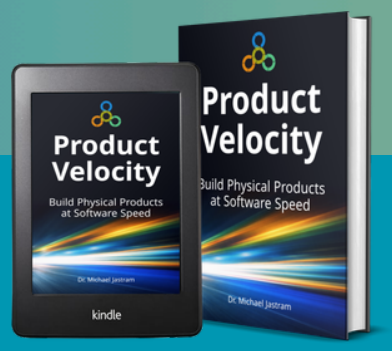
# Engineering: Turn Intent Into Something That Works

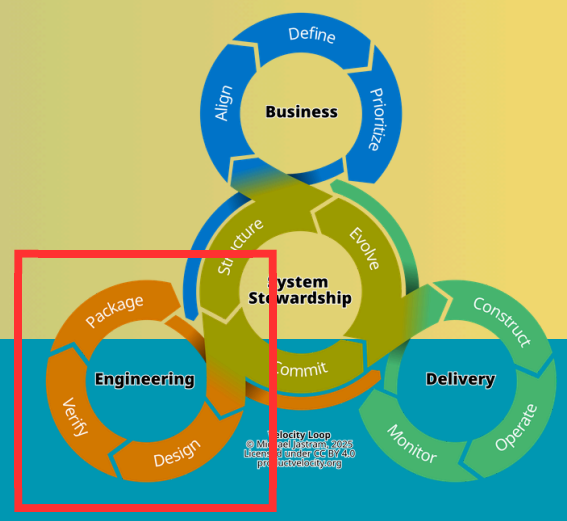
Create verified product increments while keeping work able to move through the system.




- **Verify:** Generate evidence early. Resolve uncertainty before it creates downstream rework.
- **Package:** Provide coherent increments with the interfaces, evidence, and information needed for integration and delivery.
- **Design:** Develop solutions within agreed boundaries, interfaces, and constraints. Keep implementation decisions close to the work.

**Key Output:**
Verified, integration-ready increments

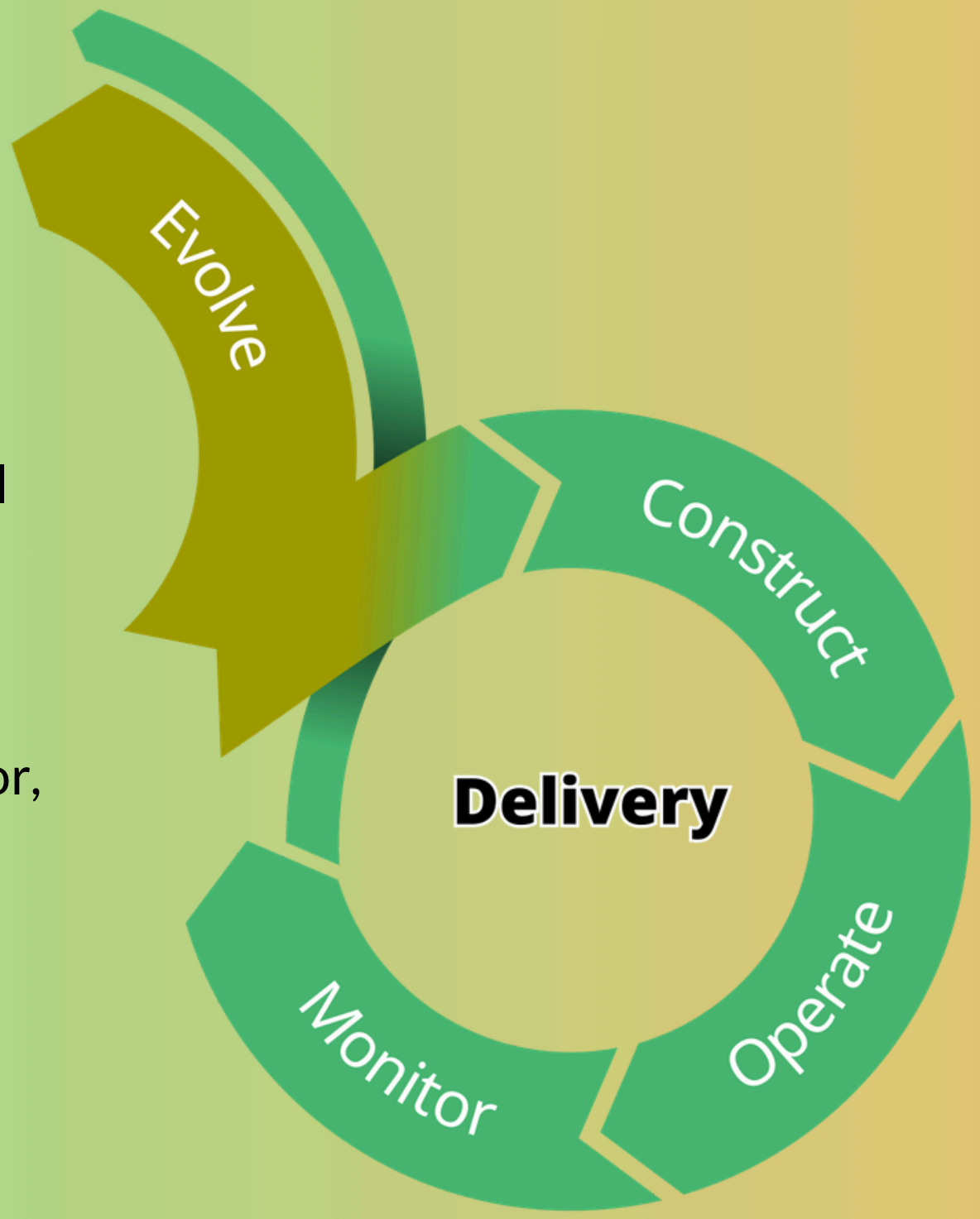


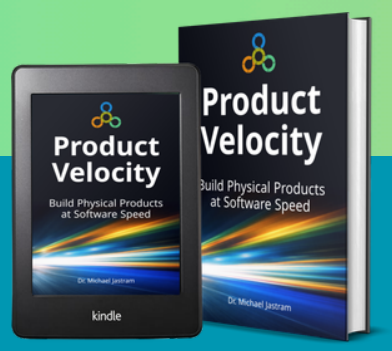
# Delivery: Make Value Usable

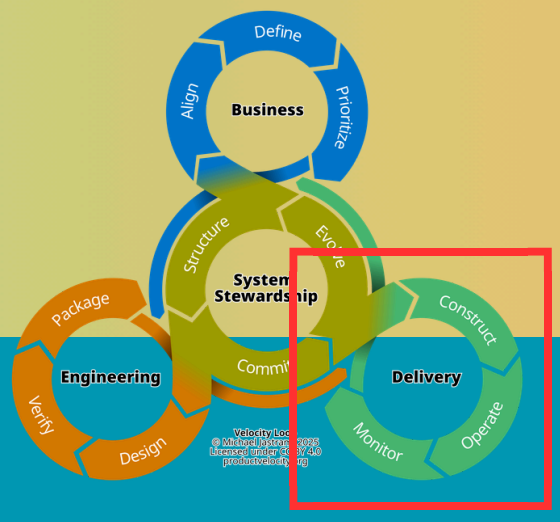
Turn engineered increments into real product instances and learn from this.

- **Construct:** Integrate, build, manufacture, and deploy. Make the transition from prototype to repeatable delivery explicit.
- **Operate:** Keep the product useful through operation, service, maintenance, and updates.
- **Monitor:** Capture quality, behavior, usage, failures, and production data. Feed evidence back to the people and decisions that can act on it.




**Key Output:**
Operational product instances and real-world evidence



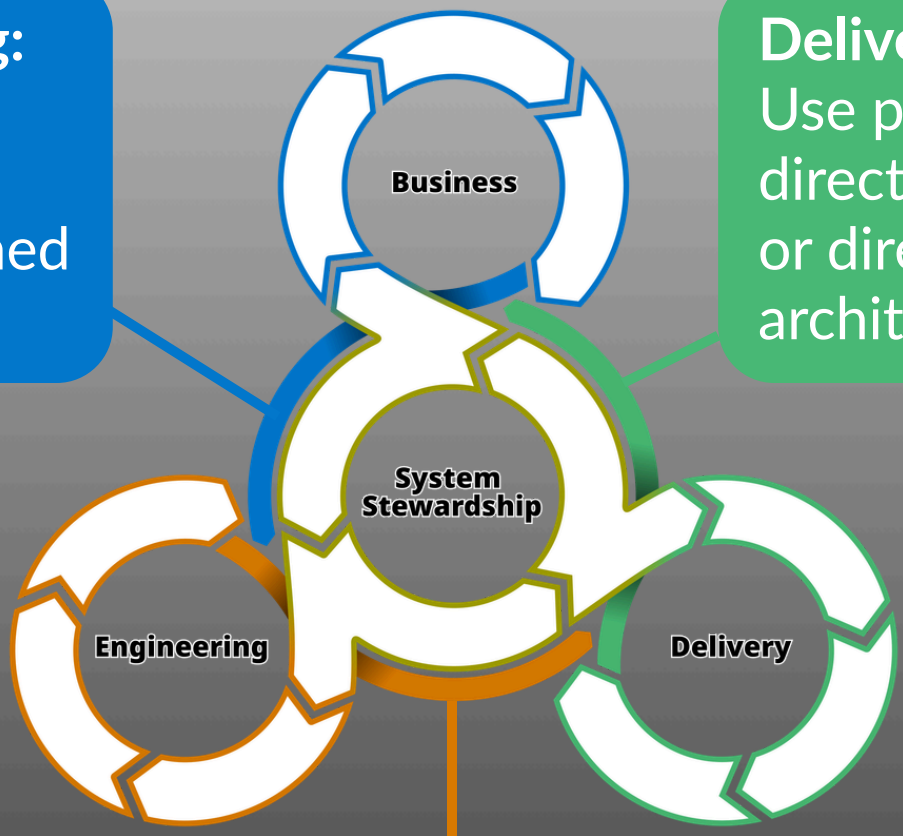
# Bypass: Take the Direct Path

System Stewardship does not need to mediate every interaction.

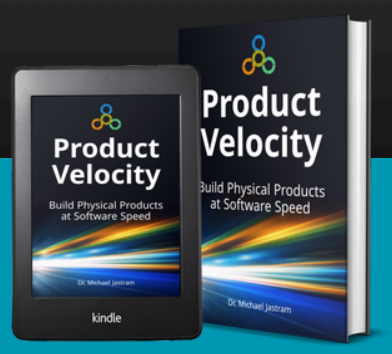
The practice loops can interact directly when existing architecture, interfaces, and commitments provide enough guidance.

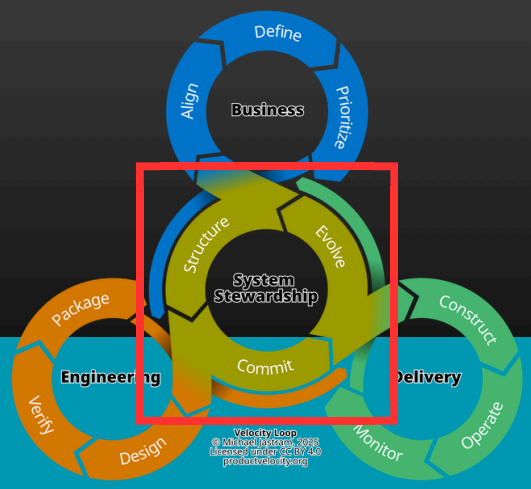
**Business ↔ Engineering:**
Communicate directly when a change can be handled within established system boundaries.

**Delivery ↔ Business:**
Use production evidence directly when it changes value or direction without requiring architectural intervention.



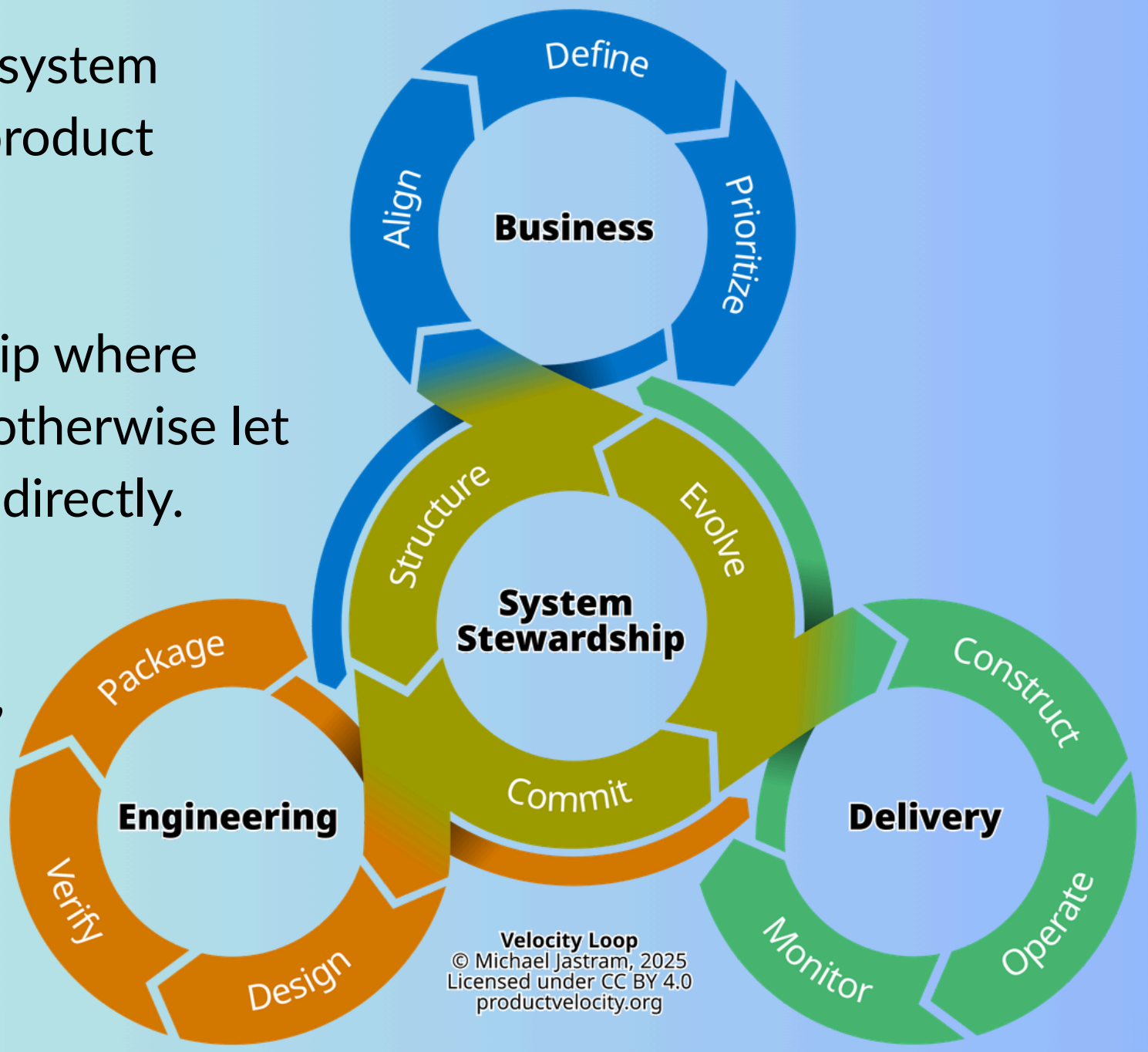

**Engineering ↔ Delivery:**
Move verified increments directly toward integration, production, or deployment when no new system-level decision is required.

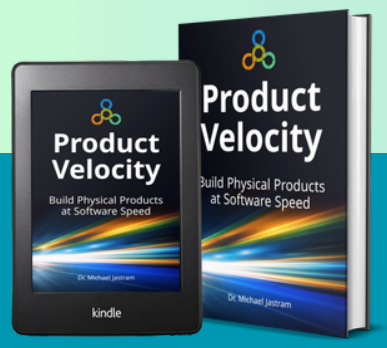


# Why the Velocity Loop Matters

- Optimize the whole loop, not the speed of individual functions.
- Where pain appears is not necessarily where the underlying constraint originates.
- Keep architecture and system decisions alive as the product evolves.
- Use System Stewardship where coherence requires it; otherwise let practice loops interact directly.
- Diagnose where work waits, decisions queue, feedback stops, or unnecessary coordi-nation slows the loop.




www.productvelocity.org

Book by Michael Jastram