Developer

✅ Developer Process Guidelines

📄 Agile Development Approach – Team Viasocket

This is how we build—fast, focused, and together. Follow this agile flow every sprint to keep the team in sync and our product moving forward with quality and speed.

🏁 1. Finalizing the Sprint

- The sprint is finalized after a collaborative discussion between the **Sprint Manager** and the developers.
- Once finalized, the **Sprint Document** becomes the **only source of truth** for work.
- Tasks must be picked only from the sprint doc.

⛔ 2. No Mid-Sprint Task Swapping

- Tasks outside the sprint are **not allowed**.
- If something critical comes up:
    - Discuss it with the **Sprint Manager**.
    - If approved, it will be officially added to the sprint.

🔍 3. Pick Task → Finalize Approach → Log Entry

- Once a developer picks a task:

1. **Define and finalize the approach** before starting development.

2. Create an entry in the **Agile Arc Table** with approach doc:
    - Approach doc collection
    - **Approach Doc should contain:**
        - Task title
        - Why it's needed
        - Your approach
        - Impact areas
        - Developer name

- This creates alignment across devs, QA, and productp.

⏱️ 4. ETA Estimation

- Add your **ETA (Estimated Time of Arrival)** for each picked task after approach finalization.

- This helps track overall sprint progress and spot bottlenecks early.

⚙️ 5. Dev → Prod: Own the Flow

- Developers are responsible for:
    - Completing the task
    - Pushing the task from **dev to staging to production**
    - Ensuring the feature is fully tested and ready for prod

---

🧾 6. Post-Completion QA Handoff

After deployment:

- Developers must document:
    - **Task summary**
    - **Impact areas**
    - Any special QA instructions
- This enables QA to test quickly and accurately.

📢 7. Clear Communication

- Daily stand-ups must reflect real-time task status.
- Use task comments and channels to communicate blockers or changes.
- Share context — don’t leave the next person guessing.

✅ 8. Definition of Done

A task is marked “Done” only when:

- Code is merged
- Feature is deployed to production
- Agile Arc table is filled
- QA documentation is complete
- QA is notified and verification is underway

🌀 9. Sprint Retrospective

- Everyone participates in retros:
    - What went well
    - What needs improvement
    - Any ideas to optimize the next sprint
- Your insights shape how we work — share openly.

🚨 10. Adherence to Process

To keep the team aligned and our workflow smooth, **following this Agile process is mandatory** for all developers.

- 📌 **First violation** of the process will result in a **formal warning**.
- 📌 **Second violation** will lead to a **half-day salary deduction**.

We're a high-ownership team — this process exists to help us deliver better, faster, and with fewer blockers. Let's commit to it together.

Let's stick to this framework so we can stay lean, ship fast, and build better every sprint. 🧠🛠️

QA

✅ QA Process Guidelines

📖 Introduction to QA

QA plays a critical role in delivering a stable, reliable product. It ensures that every change is properly planned, reviewed, tested, and documented before it reaches the end user—upholding both product quality and team.

📌 Feature Testing Process

🔍 1. Card Verification

Card Available?

- ❌ **No Card** → Reject the feature
    - If merged → Revert the PR
- ✅ **Yes** → Continue to next step

Card Must Contain:

- 📄 **Approach Documentation** with:
    - Task Details
    - Why is it needed?
    - Approach
    - Impact Areas
    - Pull Request URL
- ⏱️ **ETA** (Estimated Time of Arrival)

If any of these are missing:

- ❌ Reject the feature
- 🔁 If merged → Revert the PR

🔗 2. Pull Request and Branch Validation

- ✅ **Branch name in PR** must match branch name in **DbDash card**
  - ❌ If mismatch → Reject PR
  - 🔁 If merged → Revert the PR
- ✅ **At least one developer approval** on the PR
  - ❌ If not present → Reject PR
  - 🔁 If merged → Revert the PR

🧪 3. Local Testing

Proceed **only if all above conditions are met**.

Testing Result:

- ✅ Passes → Approve to merge on **Dev**
- 🐞 Bug found →
  - ❌ Reject PR
  - 📝 Comment the issue on the PR
  - 🔁 Send PR back to developer
  - ⚙️ Create automation for this process

🧪 4. Testing on Dev Environment

After merge on Dev:

- Start Dev environment testing

If Bug Found:

- 🐞 Report it in **DbDash**
    - Set **Priority**
    - Assign to **Developer**

If Feature Passes on Dev:

- ✅ Mark status as **QA-Approved-Dev**

🐞 Bug Process

📥 1. Bug Reporting

When a bug is found:

- Create an **entry in Agile Arc table** with:
    - Detailed Explanation
    - 📎 Attachment (Screenshot, Log, etc.)
    - Assignee
    - Priority
    - Type = Bug

🛠️ 2. Bug Resolution Validation

Once developer resolves the bug, validate the card:

Card Must Contain:

- ⏱️ ETA
- 🧠 RCA:
    - What caused the bug
    - How it was fixed
- 🧩 Impact Areas (optional)

If missing required details:

- ❌ Reject the PR
- 🔁 If merged → Revert the PR

🧪 3. Bug Testing

- Conduct **local tests** (if needed)

Testing Result:

- ❌ If issue still exists → Reject card and PR
    - 🔁 If merged → Revert the PR
- ✅ If resolved → Approve on **Dev**