# Eisenhower Matrix for Software Developers

Your Code Quality & Velocity Dashboard

For guidance and digital tool, visit: https://www.eisenhowermatrix.com/templates/eisenhower-matrix-for-software-developers/

|  |  |
| --- | --- |
| **Hot Fix**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Real Engineering**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Automate/Delegate**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Delete/Ignore**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# How to Use This as a Developer

Your value isn't in closing tickets—it's in solving problems elegantly. This matrix helps you balance urgency with code quality.

# The Developer's Decision Tree

if (production\_broken) return 'Q1';
if (improves\_codebase || builds\_skills) return 'Q2';
if (someone\_else\_can\_do) return 'Q3';
else return 'Q4'; // probably delete

# Remember

• Technical debt is a Q2 priority, not Q4
• Learning time is work time
• Clean code is faster in the long run
• Meetings aren't your real work
• Ship early, iterate often

# The Developer's 3-Question Algorithm

Quick decision tree for any task:

## Question 1: Is production broken or will the build fail?

**If YES:** It's URGENT → Continue to Q2

**If NO:** It's NOT URGENT → Continue to Q2

## Question 2: Will this improve code quality, performance, or my skills long-term?

**If YES:** It's IMPORTANT → Place based on urgency

**If NO:** It's NOT IMPORTANT → Place based on urgency

## Question 3: Can this be automated, scripted, or handled by someone else?

**If YES:** Consider DELEGATING/AUTOMATING (especially if in Q3)

**If NO:** You need to code it yourself

# Common Software-developers Traps to Avoid

## The Shiny Object Syndrome

Constantly switching to the newest framework or tool instead of mastering what you have.

**Solution:** New tech is Q2 learning time, not Q1 production code. Learn it, evaluate it, then decide.

## The Perfection Paralysis

Endlessly refactoring code that already works well enough.

**Solution:** Ship at 80% and iterate. Perfect code that never ships helps no one.

## The Hero Complex

Being the only one who understands critical parts of the codebase.

**Solution:** Document and share knowledge. Bus factor of 1 is a risk, not a badge of honor.

## The Yes Developer

Accepting every feature request, bug fix, and 'quick favor' that comes your way.

**Solution:** Your job is to deliver value, not to make everyone happy. Learn to say 'not now' or 'here's the trade-off.'

# The Developer's Daily Standup with Your Matrix

10 minutes to optimize your algorithm for the day

## Morning (7 minutes)

1. Check build status and error logs

2. Review PR comments and blockers

3. Sort Jira tickets into quadrants

4. Choose ONE Q2 task to complete

5. Block 2+ hours of focus time

## Afternoon (3 minutes)

1. Commit and push completed work

2. Update ticket status

3. Note any tech debt created

4. Plan tomorrow's Q2 task

5. Close unnecessary tabs (yes, all 47)