Performance Tuning

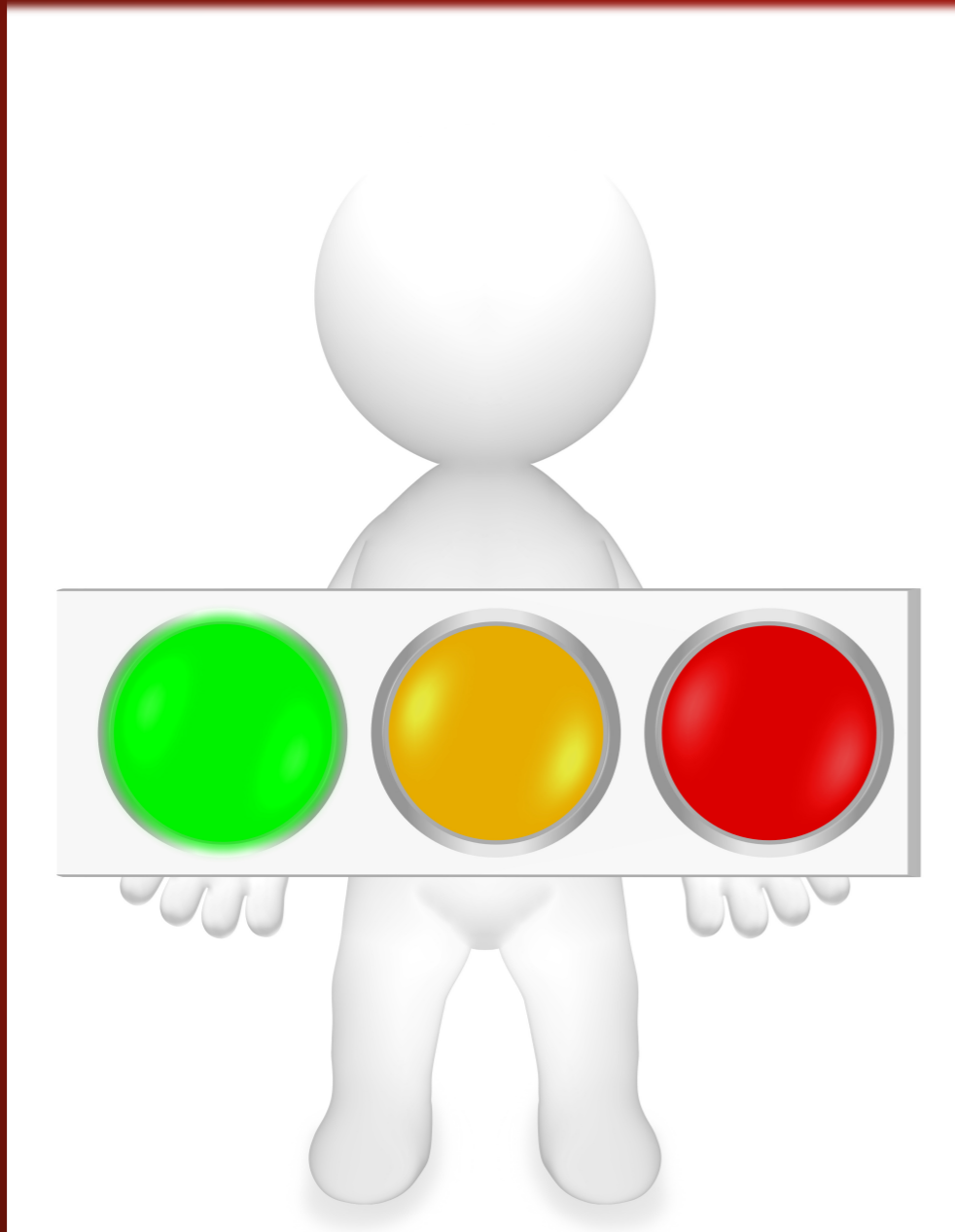## A General Approach To Creating Better Code

Better Code

- Bad vs. Ok vs. Good vs. Great

- Common Flow Pitfalls

- Common Logic Pitfalls

- The Curse of Complexity

- Memory and Loops

- Tools and Assistance

DEVELPRENEUR
BUILDING BETTER DEVELOPERS

Good or Better?

- Bare Minimum or Brute Force

- Some Clean Up

- Avoid Common Errors

- Design and Review

- Design and Evaluate

DEVELPRENEUR
BUILDING BETTER DEVELOPERS

Common Flow Pitfalls

- End too soon

- Extra loop (or one too few)

- Inaccessible Code

- Multiple paths/Incorrect One (Same names)

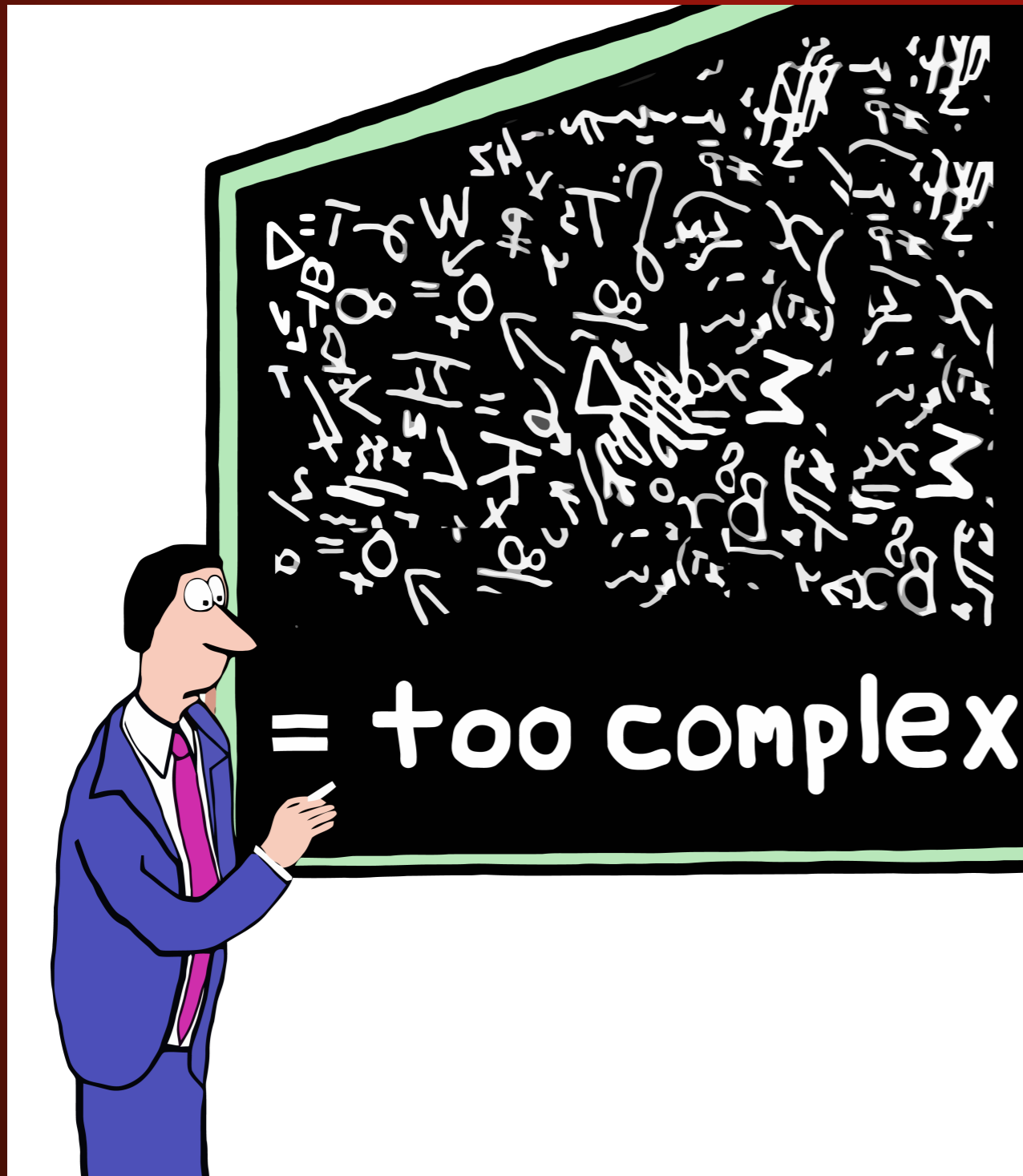- Failure to increment/decrement or exit cleanly

Common Logic Pitfalls

- Equals vs. Assignment

- Off By One

- Complex Boolean logic

- User can not possibly get to this state

- Resource Usage and Black boxes

- Side Effects

DEVELPRENEUR
BUILDING BETTER DEVELOPERS

Curse of Complexity

- Regular additions with addressing technical debt

- "Cool" features of a language

- Cadillac vs. Bicycle

- Showing Off

- Magic Numbers

DEVELPRENEUR
BUILDING BETTER DEVELOPERS

Addressing Complexity

- Code Reviews (personal and team)

- Regular Improvement Cycles

- Rough Draft vs. Simplification

- Think before adoption/Wrong templates

- Full compliance not always necessary

DEVELPRENEUR
BUILDING BETTER DEVELOPERS

Memory and Loops

- Close what you open.

- Simplify Loops wherever possible.

- Do the above even if it is annoying or requires odd declarations.

- Store or use only what you need.

- Think about types and storage.

DEVELPRENEUR
BUILDING BETTER DEVELOPERS

Tools and Assistance

- Performance meters

- Memory and OS level tracking

- Static analysis tools

- Team members and code reviews

- Multiple test environments (dev vs. user)

DEVELPRENEUR
BUILDING BETTER DEVELOPERS

Final Thoughts

- Design and Follow Through.

- Test with a reasonable and even heavy loads.

- Always check off by ones, empty and full states.

- Learn from your mistakes.

- Questions? Comments?

DEVELPRENEUR
BUILDING BETTER DEVELOPERS

What We Learned

- Performance tuning your code is part of best practices.

- You can develop habits to write better code.

- Use an 80/20 approach and you can solve a lot of problems.

- Take advantage of tools and other resources.

Thank You!

I appreciate your time and would love to discuss any of this further. You can send questions, comments and suggestions through any of these methods.

- info@develpreneur.com

- https://develpreneur.com/contact-us

- @develpreneur

- https://www.facebook.com/Develpreneur

Our goal is making every developer better.

DEVELPRENEUR
BUILDING BETTER DEVELOPERS