0-to-hero

08/10 Mentors <> 08/10 Sessions

CODESTOYEVSKY

Vugar Gasimov

Software developer

Helmes

Relocate to Estonia

DevHunt

https://www.workinestonia.com/
https://studyinestonia.ee/en

Bolt

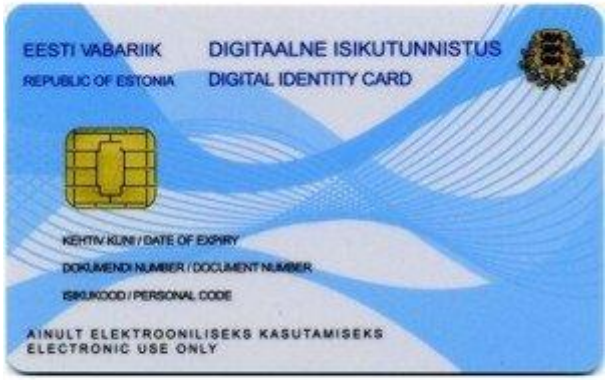
pipedrive

TransferWise

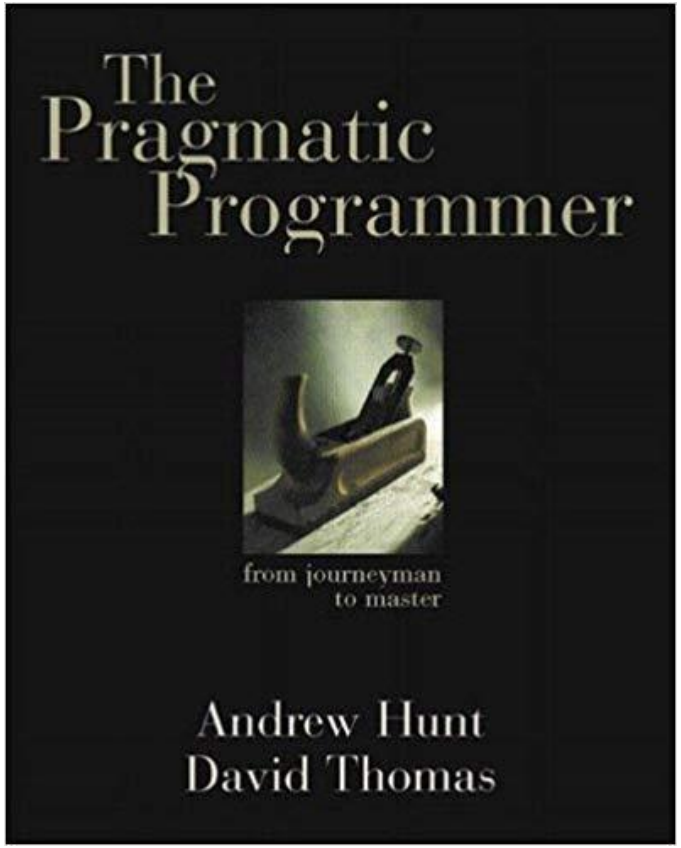
monese

The Most Digitally Savvy Country on Earth

E-Residency

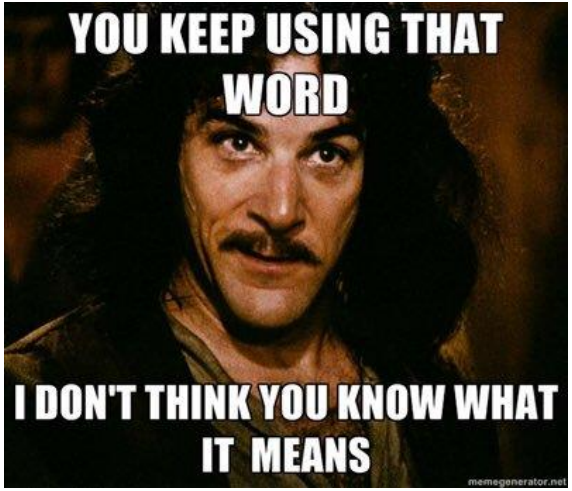

EESTI VABARIIK
REPUBLIC OF ESTONIA
DIGITAALNE ISIKUTUNNISTUS
DIGITAL IDENTITY CARD
KEHTIV KUNI / DATE OF EXPIRY
DOKUMENDI NUMBER / DOCUMENT NUMBER
ISIKUKOOD / PERSONAL CODE
AINULT ELEKTROONILISEKS KASUTAMISEKS
ELECTRONIC USE ONLY

The Pragmatic Programmer: From Journeyman to Master



Pragmatic (adjective) dictionary meaning

Dealing with things sensibly and realistically in a way that is based on practical rather than theoretical considerations.



A Pragmatic Philosophy

Think

- Ponder the problem before jumping in
- Research
- Plan
- Discuss
- Estimate
- Don’t be a slave to formal methods

Care about your craft

- Take pride in your work
- If you don’t enjoy your profession, you’ll never be great in it
- If you do enjoy your work, you’ll also enjoy the process of getting better at it (mostly!)
- Learning never ends!

Provide options, don’t lame excuses

Some things can’t be done on time, others can’t be done at all

- Don’t focus on what cannot be done
- Concentrate on what can be done instead
- Find solutions to problems

In “The War Against the Chtorr,” David Gerrold distinguishes between “guests” and “hosts”

- A “host” does whatever needs to be done
- A “guest” figures that it’s someone else’s job

Don’t live with broken windows

If something is wrong, fix it

- If you can’t get to it immediately, at least mark it for future work
- Take some action to prevent further damage
- Small problems accumulate and turn your code into crap

Software entropy

Entropy: Amount of disorder in system.

Law of thermodynamics: Entropy tends to increase in system. Similar thing happens in software system, in result software rot happens. How to stop it ?

- A spoonful of wine in a barrel of sewage gives you a barrel of sewage
- A spoonful of sewage in a barrel of wine gives you a barrel of sewage

Take responsibility

- One of the cornerstones of the pragmatic philosophy is the idea of taking responsibility for yourself and your actions in terms of your career advancement, your project, and your day-to-day work.

- A Pragmatic Programmer takes charge of his or her own career, and isn't afraid to admit ignorance or error.

- It's not the most pleasant aspect of programming, to be sure, but it will happen—even on the best of projects.

- Despite thorough testing, good documentation, and solid automation, things go wrong. Deliveries are late. Unforeseen technical problems come up.

Be a Catalyst for Change

You can't force change on people. Instead, show them how the future might be and help them participate in creating it.

Remember the Big Picture

Don't get so engrossed in the details that you forget to check what's happening around you.

Don't be like the frog. Keep an eye on the big picture. Constantly review what's happening around you, not just what you personally are doing

Invest regularly in your knowledge portfolio

- Learn at least one new language every year
- Read a technical book each quarter
- Read nontechnical books, too
- Take classes
- Participate in local user groups
- Experiment with different environments
- Stay current
- Get wired


Helmes

Coding

- Participate in open-source
- 10,000 hours rule (~416 days)
- Modify and learn from other people's code
- Teach

Egoless programming

- Software belongs to the organization, not the individual
- You don’t “own” your code
- The idea is that if you have an emotional interest in your code, you will resist criticism
- Don’t take criticism of your program as personal criticism

Basic tools

- Keep knowledge in plain text
- Use the power of command shells
- Use single editor well
- Always use source code control
- Learn branching strategies
- Don’t fear code generation
- Know your tools well, try to use shortcuts
- Use Bug, issue, project tracking tools
- Databases
- Use dual monitors
- Take care of your posture
- CI/CD

Bend, Or Break

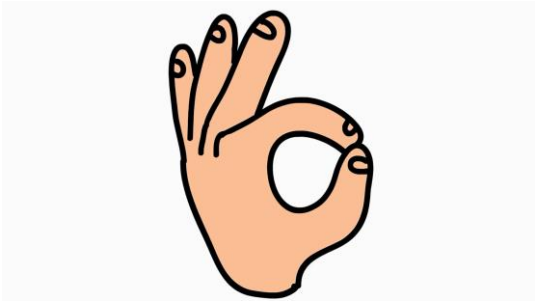
- Minimize coupling
- Configure, don’t integrate
- Separate views from models
- Make it easy to reuse
- Reversibility: There are no final decision!
- Domain languages: Program close to the domain language
- Estimate to avoid surprises
- Learn OOP, SOLID, DRY, YANGI principles
- Learn design patterns

When you should refactor

- Duplication
- Non-Orthogonal design
- Outdated knowledge – Things change, code needs to keep up
- Don’t repeat yourself
- Write -500 lines of code
- Refactor for reuse

Any program can be improved

- There are few, if any, perfect programs
- Always be ready to rewrite your program to make it better
- But know when to stop!
- Rewriting and re-designing your program is sometimes called refactoring
- Perfect is enemy of good
- **80% of IT Software Projects FAIL**



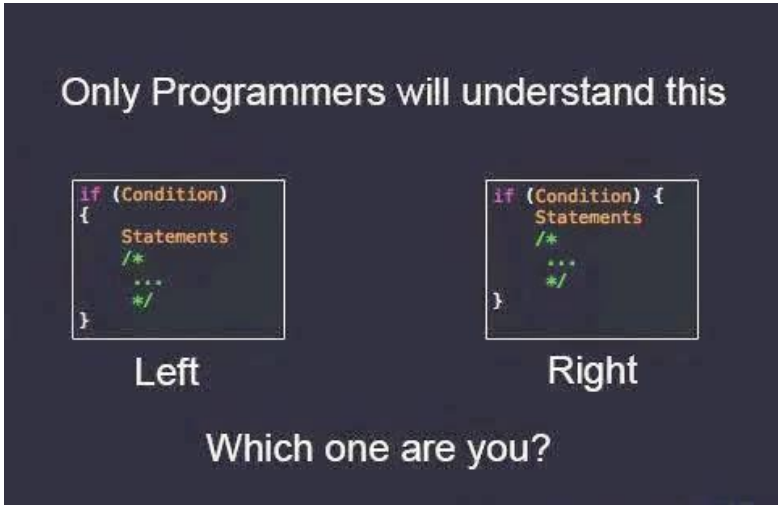
Test Early, Test Often

- Unit tests
- Integration tests
- Performance tests
- Usability tests
- Nobody does enough

“SELECT” is not broken!!!

Opinionated Programming

- Naming conventions and coding standards
- Use **Editor Config**
- Good naming
- Use enumeration
- If vs. Select Case
- Be m master debugger
- Use exception
- Comment effectively
- Shy code
- Test coverage



Write

- English* is just programming language
- Documentation
- Communication

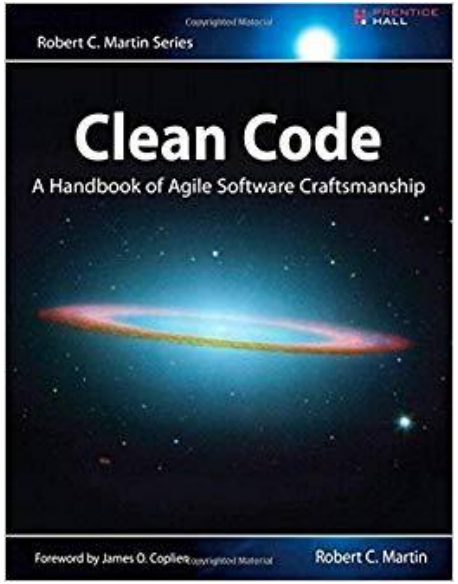
It's both what you say and the way you say it

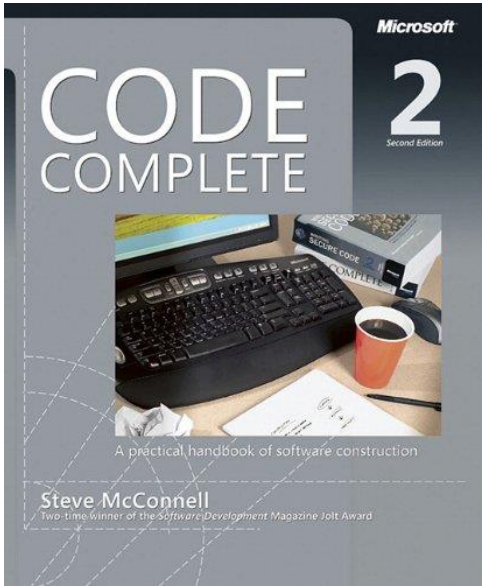
- Communication is important—more important than programming itself
  - How you present your program usually matters more than how good your program is
    - Make it look good
    - Use spell checkers
    - Learn something about good design
  - Good communication (and documentation) is essential for working in a team
  - Every piece of e-mail is a permanent record

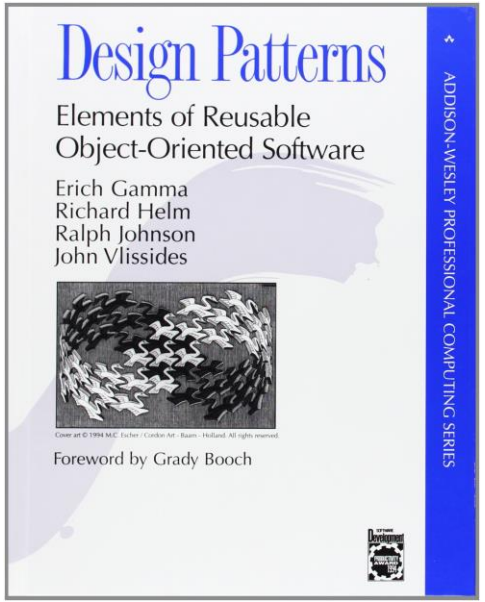
What makes a Pragmatic programmer?

1. Easy adopter / fast adapter
2. Inquisitive – You tend to ask questions
3. Critical thinker – You rarely takes the things as given
4. Realistic – This gives you a good feel for how difficult things are
5. Jack of all trades – You try to be familiar with a broad range of techs and environments

Books to read







END