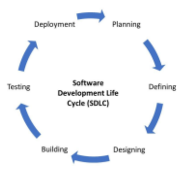
finxter Book: **Simplicity - The Finer Art of Creating Software**

Complexity

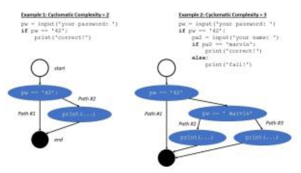
"A whole, made up of parts—difficult to analyze, understand, or explain".

Complexity appears in
- Project Lifecycle
- Code Development
- Algorithmic Theory
- Processes
- Social Networks
- Learning & Your Daily Life

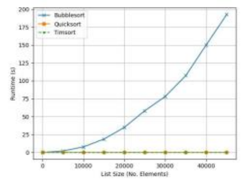
Project Lifecycle



Cyclomatic Complexity



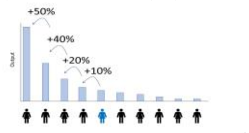
Runtime Complexity

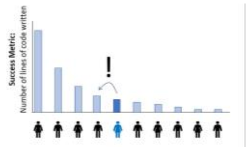


→ Complexity reduces productivity and focus. It'll consume your precious time. Keep it simple!

80/20 Principle

Majority of effects come from the minority of causes.

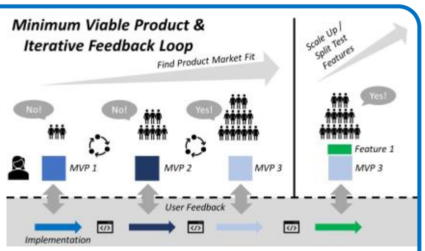




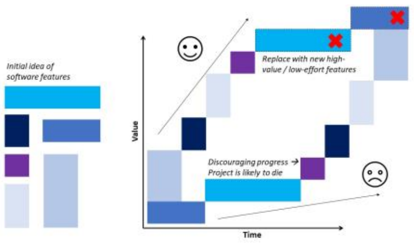
Pareto Tips
1. Figure out your success metrics.
2. Figure out your big goals in life.
3. Look for ways to achieve the same things with fewer resources.
4. Reflect on your own successes
5. Reflect on your own failures
6. Read more books in your industry.
7. Spend much of your time improving and tweaking existing products
8. Smile.
9. Don't do things that reduce value

Maximize Success Metric:
#lines of code written

Minimum Viable Product (MVP)



A minimum viable product in the software sense is code that is stripped from all features to focus on the core functionality.


Initial idea of software features
Replace with new high-value / low-effort features
Value
Discouraging progress → Project is likely to die
Time

How to MVP?
- Formulate hypothesis
- Omit needless features
- Split test to validate each new feature
- Focus on product-market fit
- Seek high-value and low-cost features

Clean Code Principles

1. You Ain't Going to Need It
2. The Principle of Least Surprise
3. Don't Repeat Yourself
4. **Code For People Not Machines**
5. Stand on the Shoulders of Giants
6. Use the Right Names
7. Single-Responsibility Principle
8. Use Comments
9. Avoid Unnecessary Comments
10. Be Consistent
11. Test
12. Think in Big Pictures
13. Only Talk to Your Friends
14. Refactor
15. Don't Overengineer
16. Don't Overuse Indentation
17. Small is Beautiful
18. Use Metrics
19. Boy Scout Rule: Leave Camp Cleaner Than You Found It

Unix Philosophy

1. Simple's Better Than Complex
2. **Small is Beautiful (*Again*)**
3. Make Each Program Do One Thing Well
4. Build a Prototype First
5. Portability Over Efficiency
6. Store Data in Flat Text Files
7. Use Software Leverage
8. Avoid Captive User Interfaces
9. **Program = Filter**
10. Worse is Better
11. Clean > Clever Code
12. **Design *Connected* Programs**
13. Make Your Code Robust
14. Repair What You Can — But Fail Early and Noisily
15. Write Programs to Write Programs

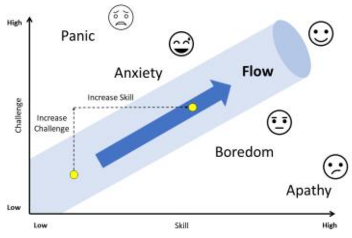
Premature Optimization

"Programmers waste enormous amounts of time thinking about […] the speed of noncritical parts of their programs. We should forget about small efficiencies, say about 97 % of the time: premature optimization is the root of all evil." – **Donald Knuth**

Performance Tuning 101
1. Measure, then improve
2. Focus on the slow 20%
3. Algorithmic optimization wins
4. All hail to the cache
5. Solve an easier problem version
6. Know when to stop

Flow

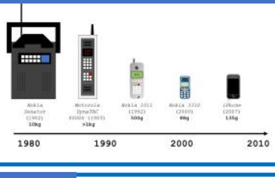
"… the source code of ultimate human performance" – ***Kotler***



How to Achieve Flow? (1) clear goals, (2) immediate feedback, and (3) balance opportunity & capacity.

Flow Tips for Coders
1. Always work on an explicit practical code project
2. Work on fun projects that fulfill your purpose
3. Perform from your strengths
4. Big chunks of coding time
5. Reduce distractions: smartphone + social
6. Sleep a lot, eat healthily, read quality books, and exercise → garbage in, garbage out!

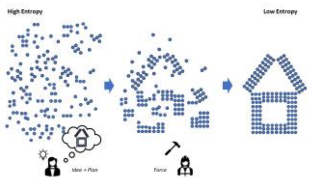
Less Is More in Design



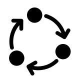
How to Simplify Design?
1. Use whitespace
2. Remove design elements
3. Remove features
4. Reduce variation of fonts, font types, colors
5. Be consistent across UIs

Focus

You can take raw resources and move them from a state of high entropy into a state of low entropy—using *focused effort towards the attainment of a greater plan*.



3-Step Approach of Efficient Software Creation
1. Plan your code
2. Apply focused effort to make it real.
3. Seek feedback



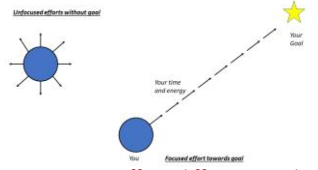


Figure: Same effort, different result.

*Subscribe to the **11x FREE** Python Cheat Sheet Course:*
https://blog.finxter.com/python-cheat-sheets/