

# Rebuild or Refactor

**Talal El Afchal**

`talal.elaafchal@gmail.com`

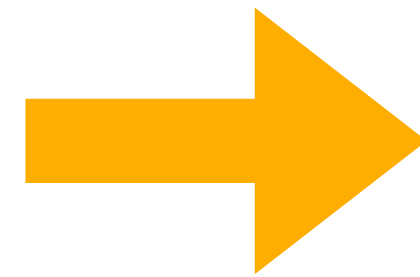
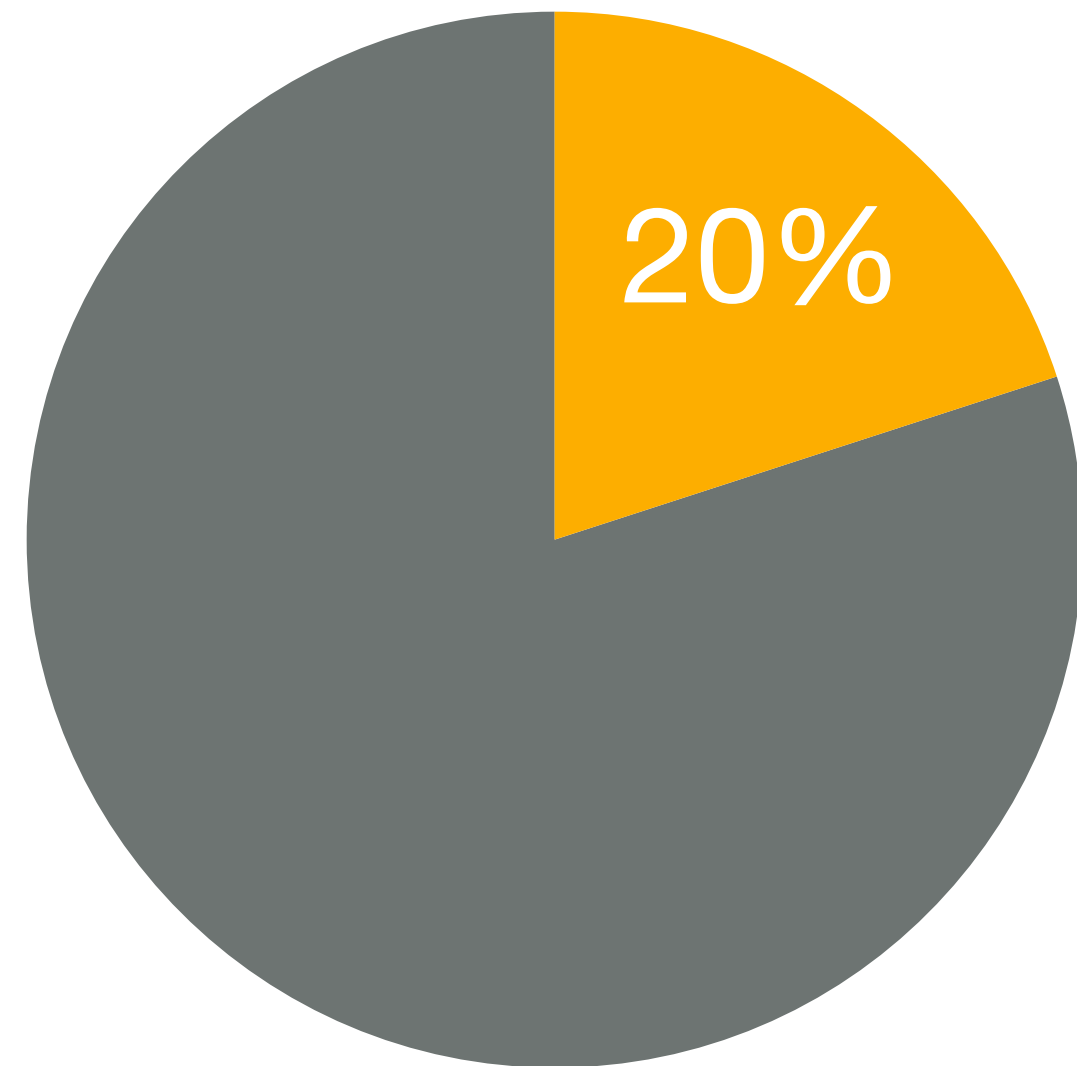
**Should we Refactor or Rebuild ?**

# Refactor

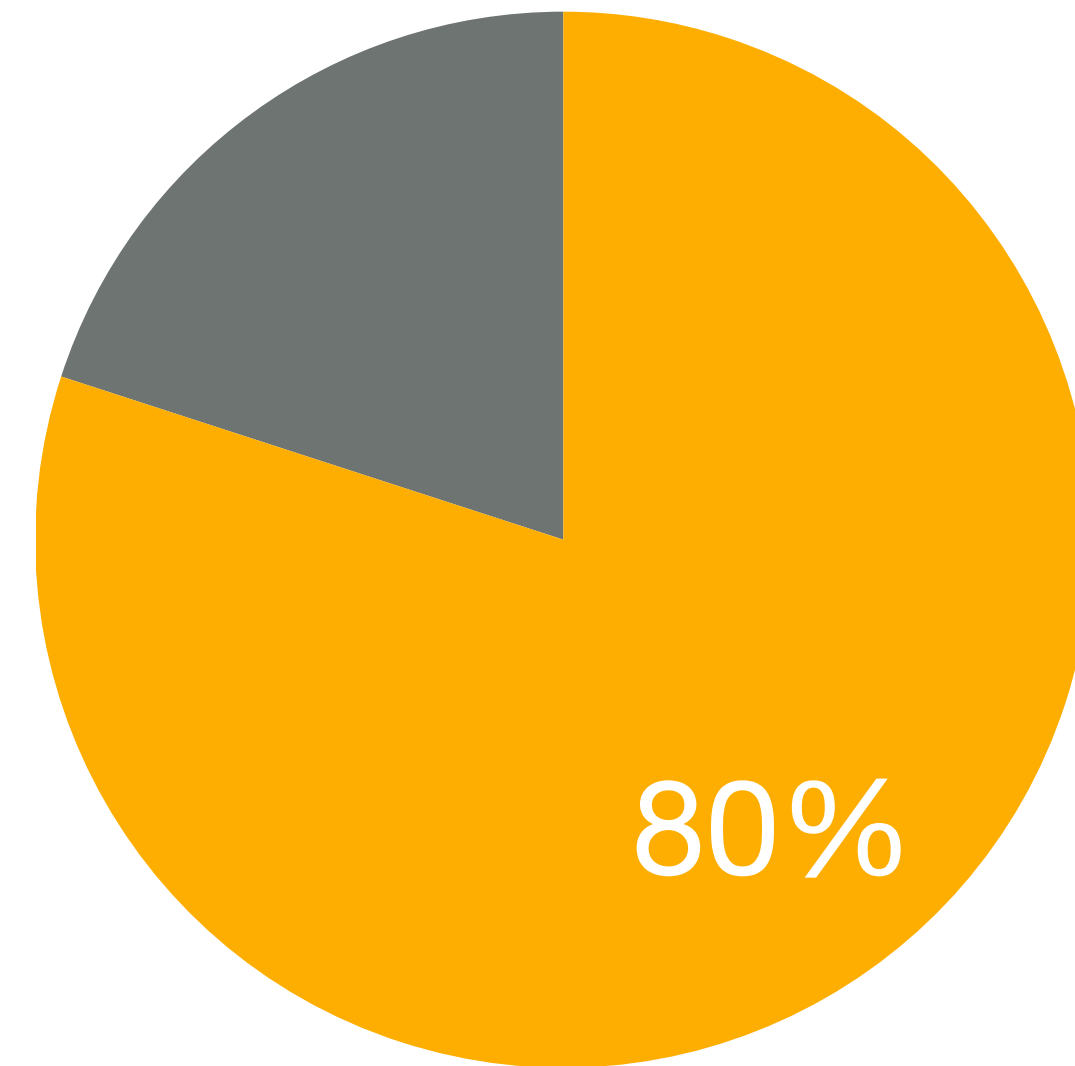
- Code smells expert
- Object Calisthenics rules
- Solid principles
- Design Pattern

# Refactor

**Effort**



**Results**



**Refactoring 80-20 rule**



# Refactor



**It's all green then refactor**





# Trading App



The screenshot shows a login screen with the text 'Welcome' and 'Insert your credentials.' Below this are two input fields labeled 'UserID' and 'Password'. A 'Continue' button is positioned below the input fields. At the bottom, there is a link for 'Having problems? Help' and a highlighted button that says 'Want to try? Create demo user'.

**Should we Refactor or Rebuild ?**

# Rebuild

- Start clean and blame the previous developers
- Allows new developers to instantly implement code they are comfortable with
- Deliver new value



# Rebuilding software

**“ the single worst strategic mistake that any software company can make “**



**Joel Spolsky**

<https://www.joelonsoftware.com/>

# Rebuilding Risks-Disadvantages

- A rewrite is more difficult than a write
- Leads to losing some of the legacy features
- Building a new system as poorly as the old system
- Rebuilding usually takes longer than estimated

**Should we Refactor or Rebuild ?**

**A complete rewrite of an existing application  
should be the last choice**



**Thank you**

[talal.elafchal@gmail.com](mailto:talal.elafchal@gmail.com)

# References

- <https://www.joelonsoftware.com/2000/04/06/things-you-should-never-do-part-i/>
- <https://entrision.com/blog/rebuild-versus-refactor/>
- <https://www.industriallogic.com/blog/rewrites-hazardous/>
- <https://www.bennorthrop.com/rewrite-or-refactor-book/chapter-2-the-risks-of-rewrites.php>
- <https://medium.com/adobetech/we-decided-to-rewrite-our-software-you-wont-believe-what-happened-next-dd03574f6654>