

Tell, don't ask!

Its all about responsibility

Peter Stålbrand

Twitter: @liteolika

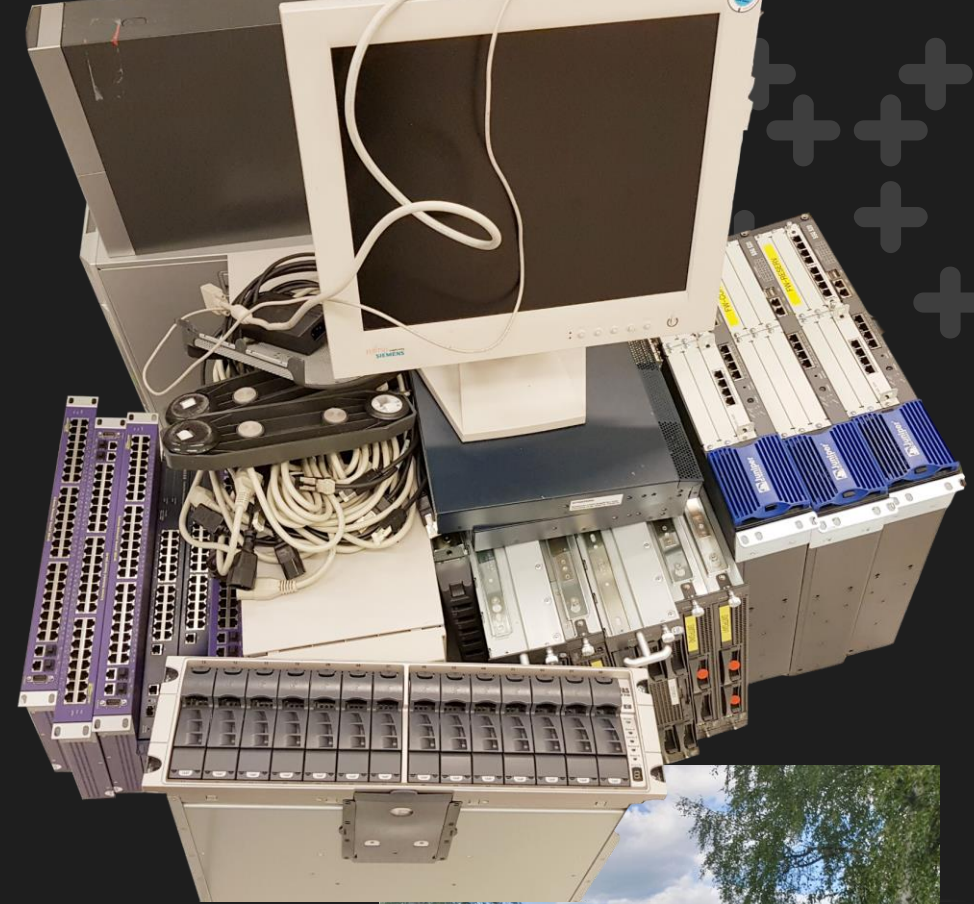
peter.stalbrand@activesolution.se

Blog: <https://peter.liteolika.se>



Who am i?

- Lives in Stockholm with my wife Anna
- Senior developer at Active Solution
- Previous
 - network, server- and storagespecialist
 - datacenter and security manager
- 20 years of developing with multiple languages and different platforms
- "Open sorcerer"



S.O.L.I.D

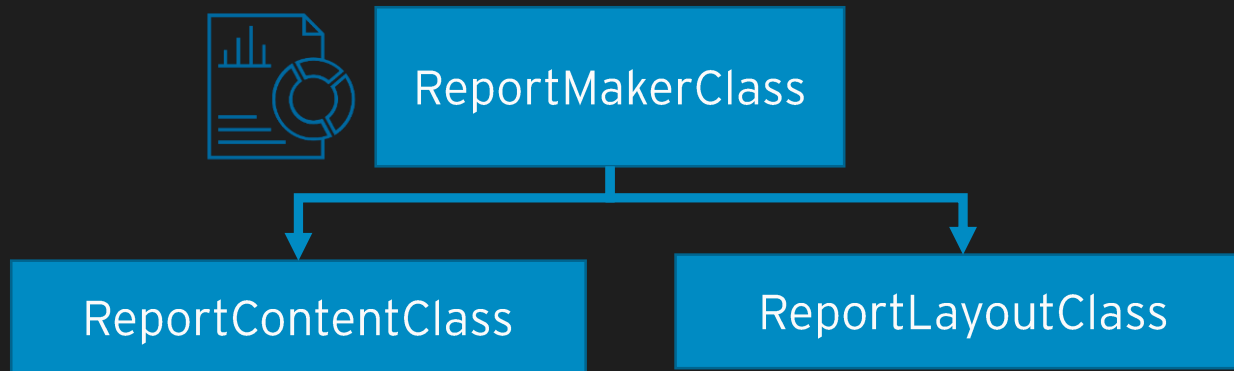


- Single Responsibility Principle (SRP)
 - A class should have one, and only one reason to change
- Open-Closed Principle (OCP)
 - Open for extension, closed for modification
- Liskovs Substitution Principle (LSP)
 - Derived classes must be substitutable for their base classes
- Interface Segregation Principle (ISP)
 - Make fine grained interfaces that are client specific
- Dependency Inversion Principle (DIP)
 - Depend on abstractions, not on concretions

Single Responsibility Principle

“A class should have one, and only one reason to change”

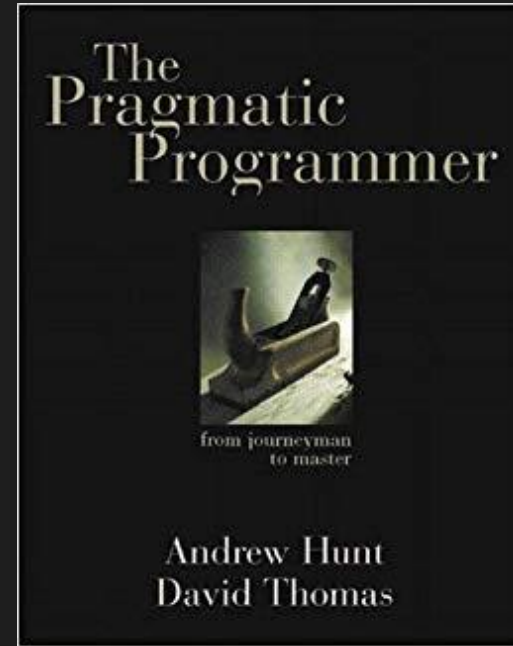
What does “reason to change” mean?



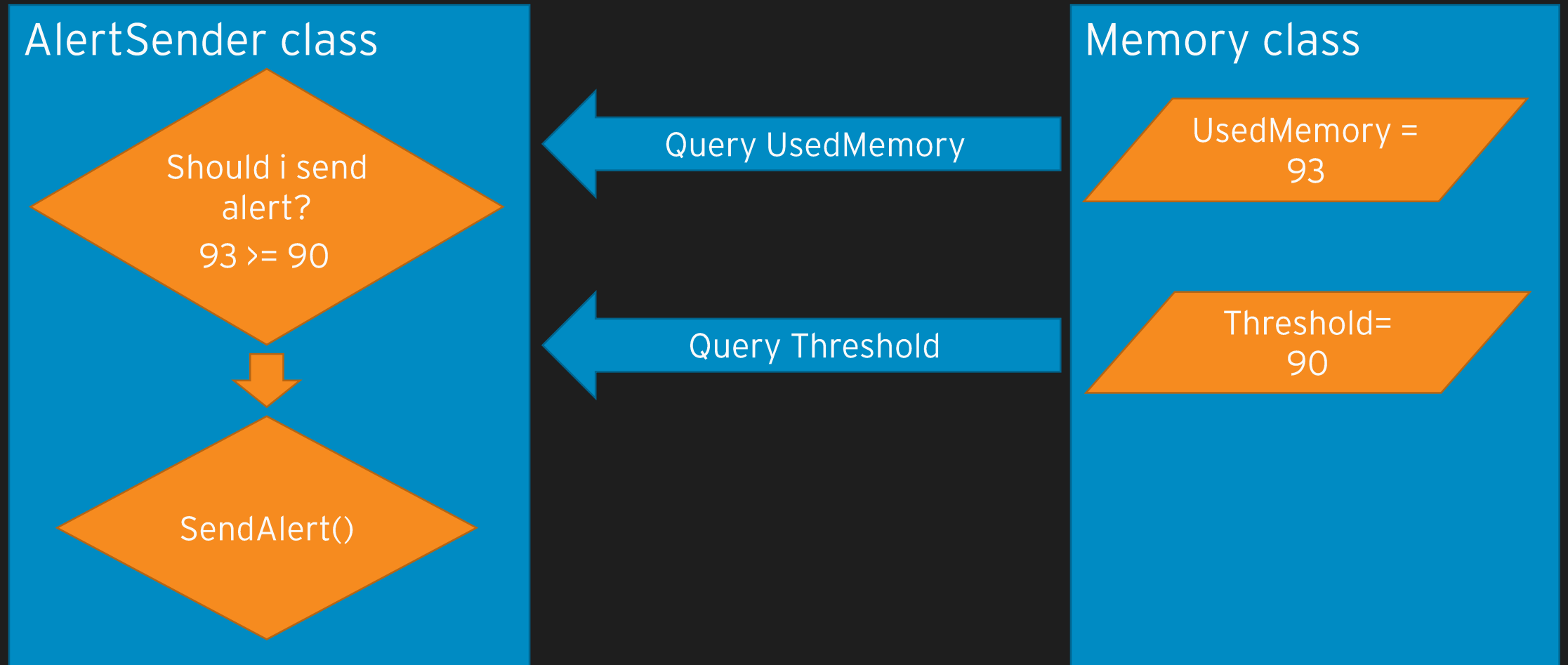
Robert C. Martin (Uncle Bob)

"Tell don't ask"

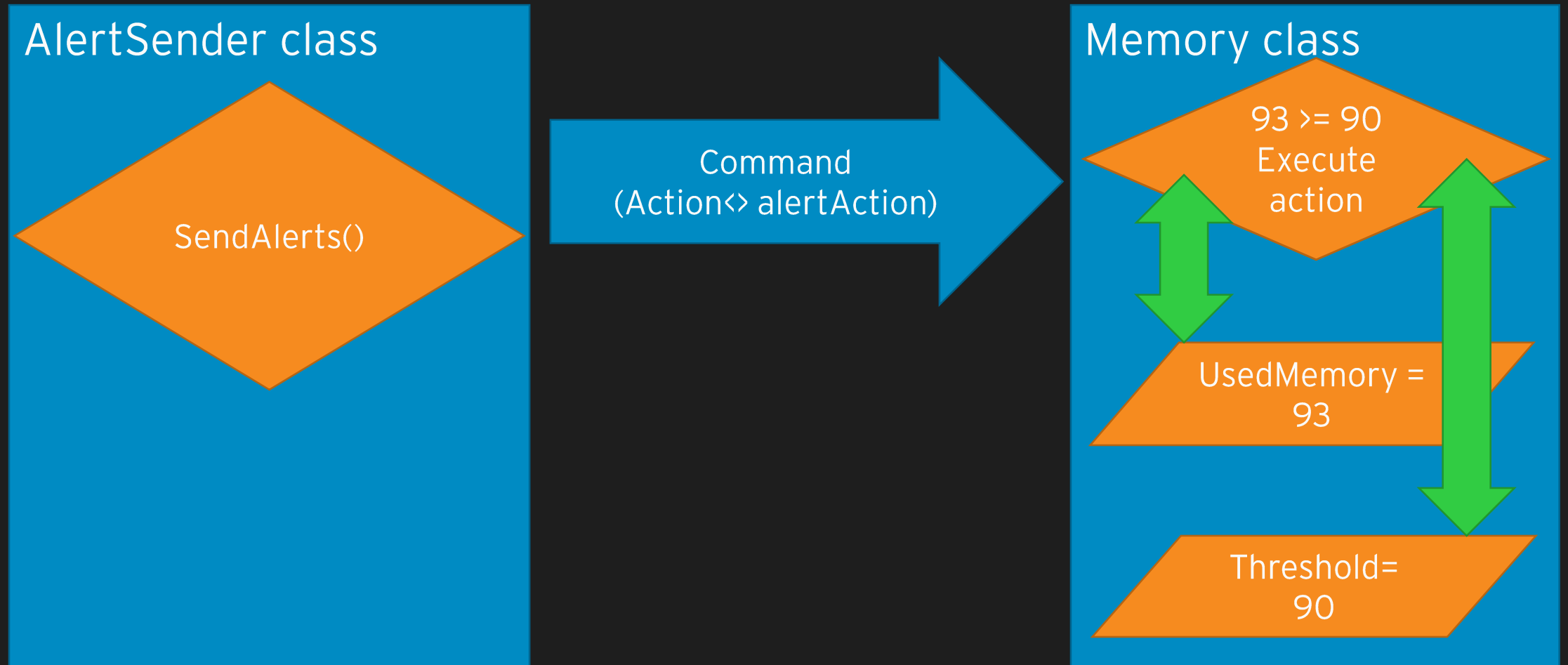
- "The pragmatic programmer" by Andy Hunt and Dave Thomas
- Command objects to do things
- Do not query objects and then tell them to do things based on the query results



Violation of TDA



Proper TDA



Avoid testing pitfalls



Testing a class with too much responsibility may:

- Open a database connection
- Query for data
- Calculate memory usage
- Transform into DTO

Demo

Just tell me.

Principle trade-offs and cohesion

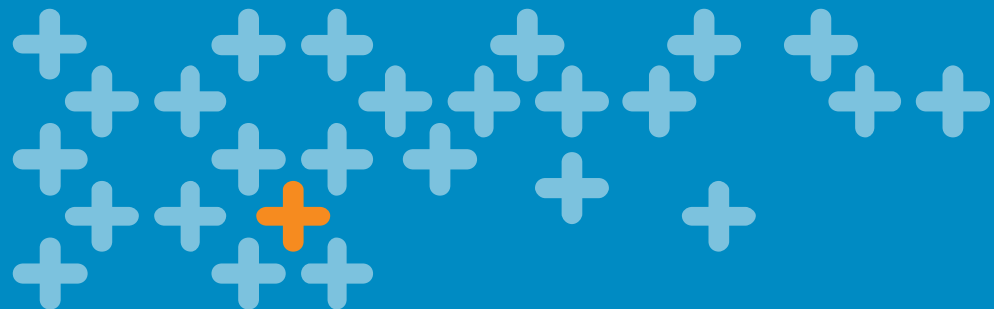


- TDA often implements DRY (Don't repeat yourself)
- LOD (Law Of Demeter)
 - Loose coupling
 - Limited knowledge about other units
- Don't be a "GetterEradicor"

Hungry for more?



- If you want to learn more about the SOLID principles, check out Chris Klug's talk from TechEd at Channel 9.
- The Clean Code Blog – By Robert C. Martin
<https://blog.cleancoder.com/>
- Book - "The Pragmatic Programmer"
20th Anniversary Edition
<https://pragprog.com/>
- Slides and code
<https://github.com/Liteolika/SpeakerSessions>



Thank you!

Go create beautiful, responsible code

Peter Stålbrand

Twitter: @liteolika

peter.stalbrand@activesolution.se

Blog: <https://peter.liteolika.se>

