DR-Tools

A tool quality suite to help the
developers to maintain health and code evolution

drtools.site

Who am I?

glacerda@wildtech.com.br
@guilhermeslac

- ✓ MSc and Computer Science PhD Student (UFRGS)
- ✓ Graduate and Postgraduate Lecturer (Unisinos)
- ✓ Associate Consultant at Wildtech
- ✓ Agile Methods Pioneer in Brazil
- ✓ XP-RS/GUMA Co-founder
- ✓ ScrumAlliance , IASA, SBC, and ACM Member

WILDTECH

Why medicine metaphor?

DR-Tools

32 Heuristics

drtools.site

Heuristics - Summary

1. **Context that provides general information about Project dimensions**
   Indicative of better filtering the information using the '--top X' option to help to understand (Classification: SMALL, MEDIUM and LARGE)

2. **Consider the average number of classes per namespace**
   Indicative that classes are not evenly distributed

3. **Evaluate the average number of SLOCs per class**
   Indicative of very large classes

4. **Observe the average distribution of methods by classes**
   Indicative of many features by class

5. **Consider the average complexity by class**
   Indicative of the classes complexity

DR-Tools

Heuristics - Namespaces

6. **Observe the distribution of classes by namespace**

   If a namespace has many classes (high NOC), it can be indicative of 'promiscuous package'

7. **Evaluate the distribution of abstract types (abstract classes, interfaces) by namespaces**

   Indicative for extension and reuse

8. **Evaluate the relationship of the NOC and NAC metrics for the namespace**

   A huge difference between them can indicate a poor distribution between abstract types and concrete types

DR-Tools

Heuristics – Types (1)

9. **Evaluate metrics beyond the SLOC**
   WMC, DEPS (DEP and I-DEP) and NOM/NPM are good indications of how the class is doing

10. **Class with high NOA and NOM, but low WMC**
    It can be indicative of POJO (Plain Old Java Object)

11. **High SLOC, but without many methods (low NOM/NPM)**
    It may be indicative of 'long methods'

12. **High SLOC and WMC, but without many methods (low NOM/NPM)**
    It can be indicative of 'complex class'

13. **High NOM/NPM can be indicative of class with many responsibilities**
    Indicates low cohesion and possibly 'god class'

DR-Tools

Heuristics – Types (2)

14. **High NOM/NPM and low NOA can be indicative of class with many responsibilities**
    It can be indicative of a 'controller' class

15. **High NOM and low NPM may indicate that the methods have been divided**
    Indicative of private/protected/default methods

16. **High NOA can be indicative of class with many responsibilities**
    It may be indicative of low cohesion, making maintenance difficult

17. **High DEP and low I-DEP can indicate a class with many external dependencies**
    Dependencies on external APIs (frameworks, libs)

18. **High I-DEP (and therefore high DEP), can indicate a class with many project class dependencies**
    High coupling incidence

DR-Tools

Heuristics – Methods

19. **High PARAM may be indicative of a method with low cohesion**
 Possibly it is a 'long method'

20. **High CYCLO and low MLOC can be a 'complex method'**
 Indicative of complexity, legibility and understanding problem

21. **High NBD can be a 'complex/long method'**
 Indicative of complexity, legibility and understanding problem

22. **High CALLS may indicate high coupling**
 Indicative of problem of several dependencies

23. **High MLOC, CYCLO, CALLS, and NBD is a strong indicator of more than one problem**
 It can be indicative of a 'complex/long method'

DR-Tools

Heuristics – Coupling (1)

24. **Avoid cyclical dependencies**
Make changes complex and generate 'total build syndrome'

25. **High CA may indicate that the namespace is stable**
If a type changes, possibly it will cause any type which depends on it to be changed

26. **High CE may indicate that the namespace is unstable**
The incidence of change in other namespaces that this namespace depends on will cause it to change

27. **I indicates the instability of the namespace**
I=0 indicates maximum stability of the namespace; I=1 indicates maximum instability of the namespace

DR-Tools

Heuristics – Coupling (2)

28. **If I=0, it indicates that CA>0 and CE=0, indicates total stability**

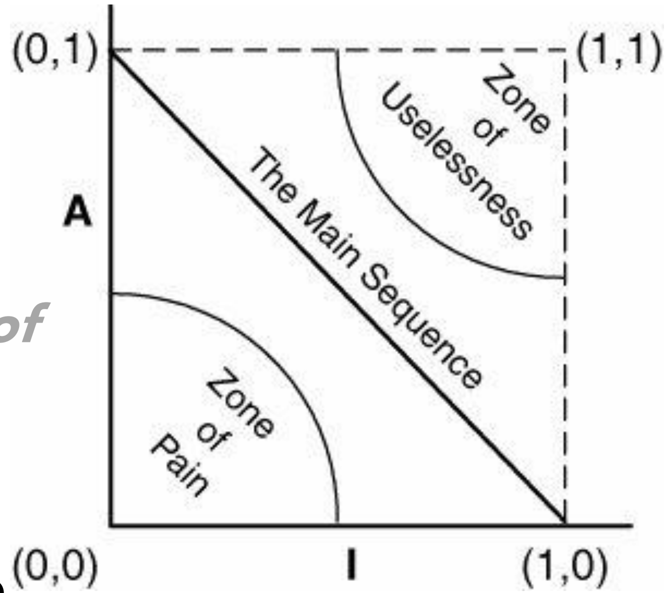
It is responsible and independent. Dependent namespaces make it difficult to change and have no dependency on others that can force the change

29. **A indicates the namespace degree of abstraction**

A=0, namespace has no abstract types; A=1, namespace only has abstract types

DR-Tools

Heuristics – Coupling (3)



30. **Consider namespaces that are in exclusion zones**

Zone of Pain (namespaces with I and A close to 0) and Zone of Uselessness (namespaces with I and A close to 1)

31. **Namespace located next to the main sequence indicates that it is neither abstract nor too unstable**

D value (between 0 and 1) that will indicate the position in the main sequence

32. **D indicates how far a namespace is from the main sequence**

D close to 0 indicates proximity to the main sequence; D close to 1, indicates the distance from the main sequence. These values (closer to 1) can indicate when a namespace is maintainable and less sensitive to changes

DR-Tools

DR-Tools

A tool quality suite to help the developers to maintain health and code evolution

drtools.site