Test Driven Development

SOLID Principles and code smells

Rune Holen / Bouvet

SOLID principles:

Single Responsibility

Open-Closed

Liskov substitution

Interface segregation

Dependency Inversion

Code smells

Avoiding different code smells, increases readability and decreases chance of unexpected effects (errors).

Dispensables

Lazy class – A class, which does too little. Possibly placehoder?

Data class – Special case of lazy class – only having data, no behaviour

Speculative Generality – unused class, method, field or parameter.

Comments – better than commenting what is done - create a method, which is named accordingly

Dead Code – Deleted code has no bugs and significantly improves readability

Duplicated Code – Disturbs readability and forces changes to be performed several times

Code smells

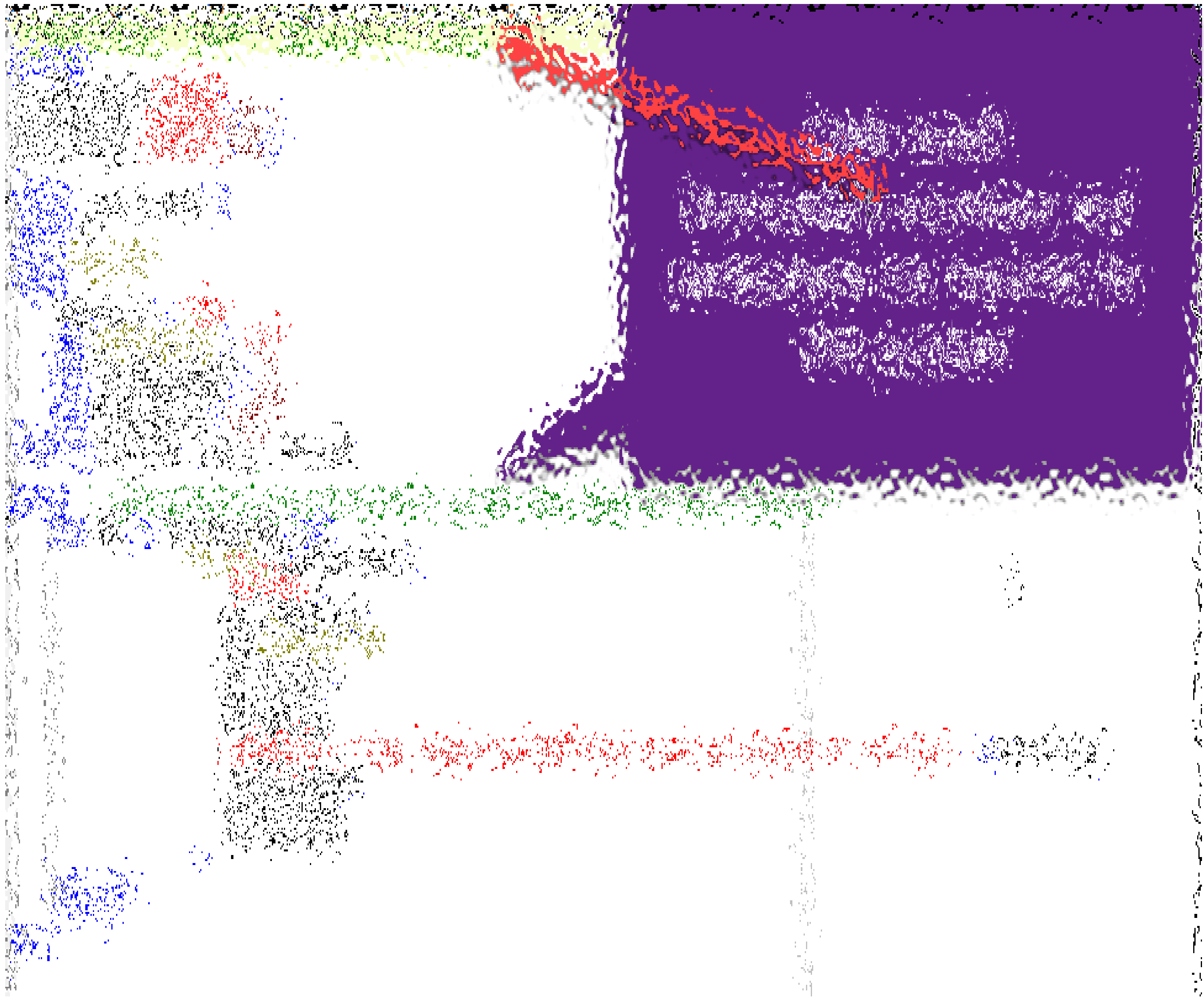
Dispensables - continued

Script example – amend deals in one strategy. This could be simplified by creating a new method with name: UpdateM2mOnFutures(deal_reference);

All parameters required by signature of pk_deal.DummyAmendDeal() can be encapsulated in simpler function with appropriate name.

Method coupling premise

--> Reduce amount of parameters to relevant scope by creating new wrapper function with fewer parameters..

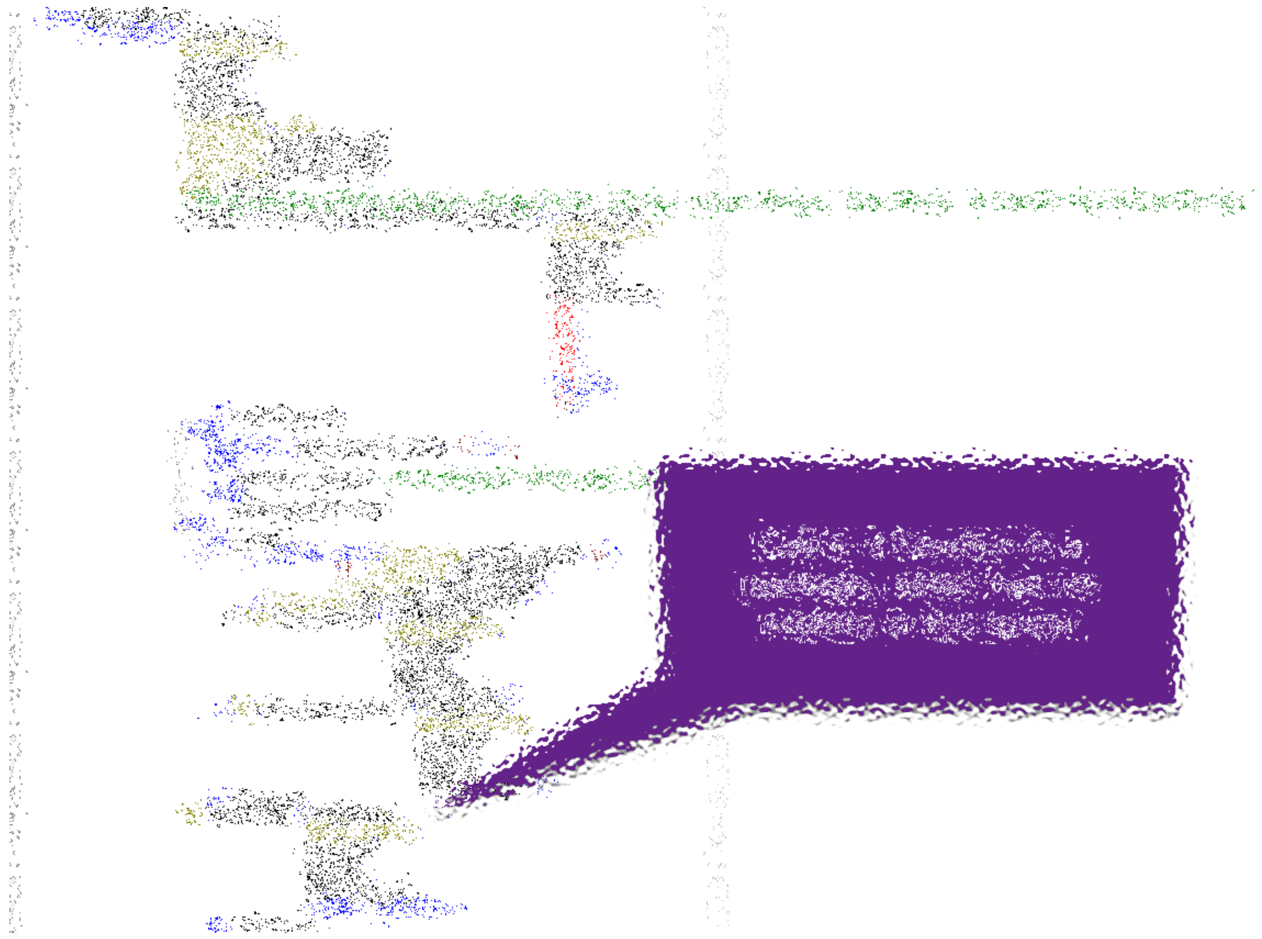


Code smells

Couplers

Feature Envy – Don’t use features or fields/variables, which belong to a different scope

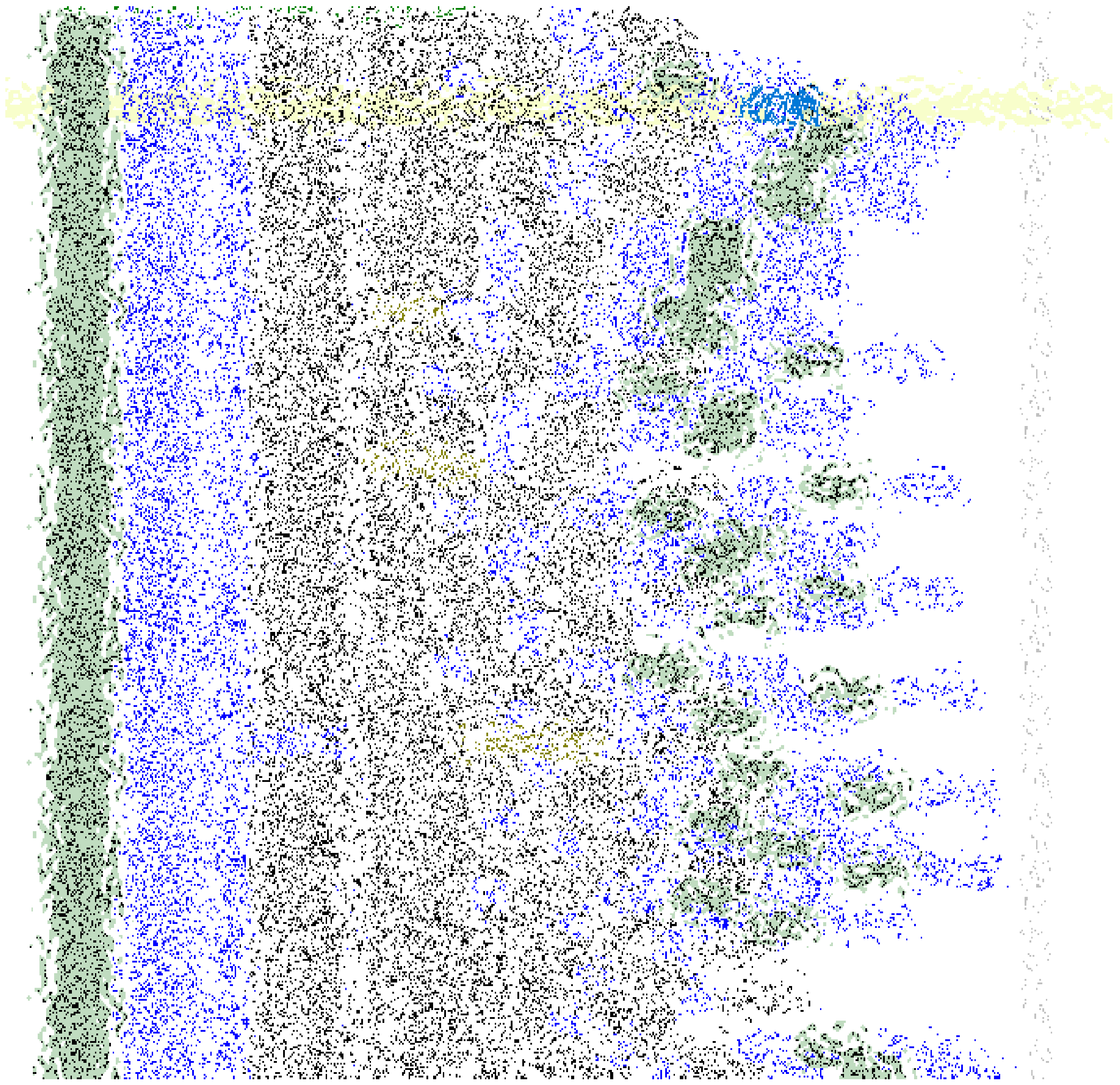
Inappropriate Intimacy – special case of feature envy when a method/class uses fields/methods from other classes.



Code smells
Couplers – continued

Feature Envy

Call to function in BO package means this view is depending on this package (which we are going to retire!) and the schema, where view is defined requires GRANT privilege to be able to grant read access to this foreign function to users of view.



Code smells

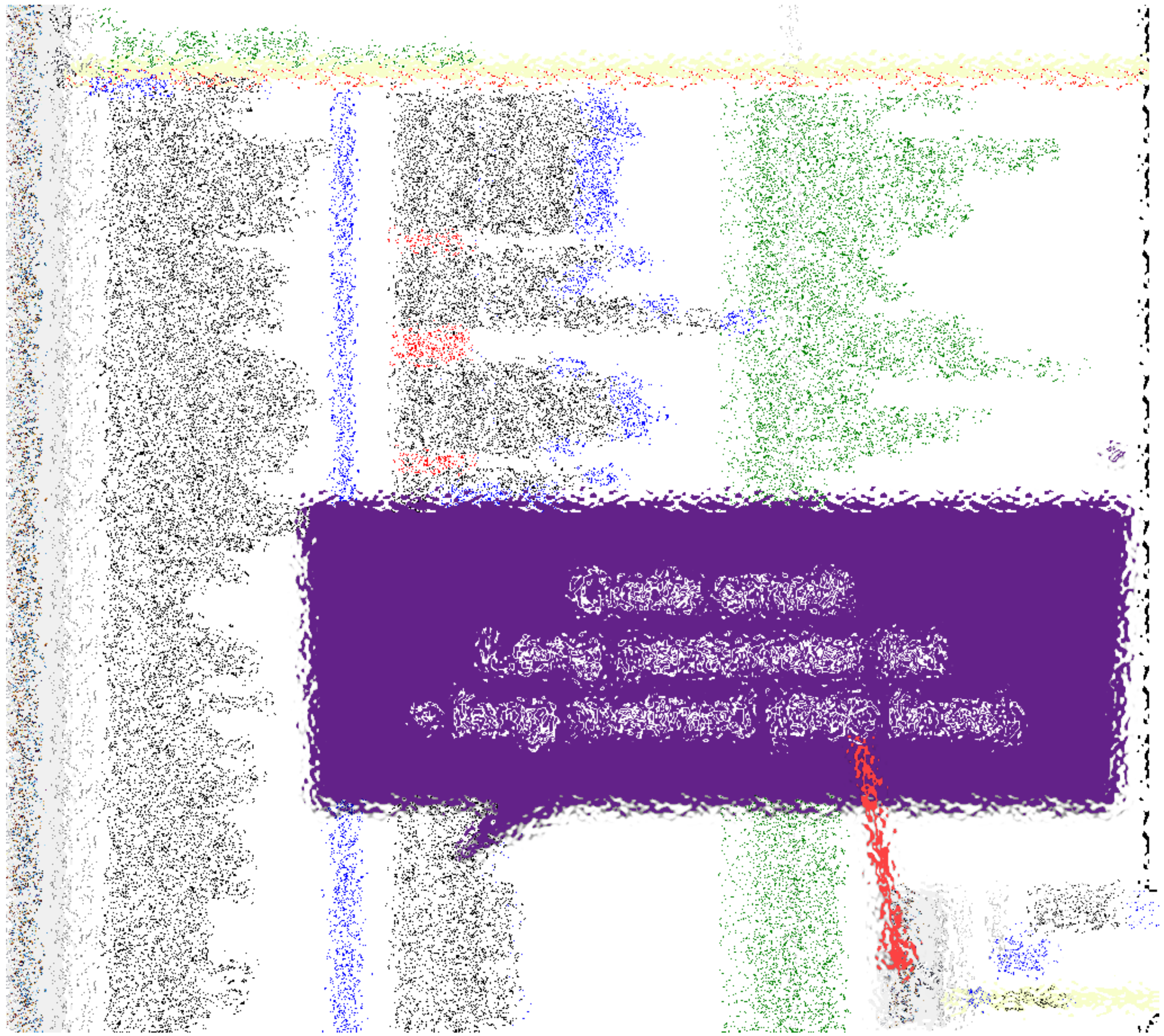
Bloaters:

Long method – hard to read and risk of side effects. Use shorter, specific methods

Large class – One responsibility. Don't try to fix everything in one class.

Long parameter list – makes it hard to invoke method correctly

Long method with very long parameter list makes it very hard to use



Questions?

rune.holen@bouvet.no