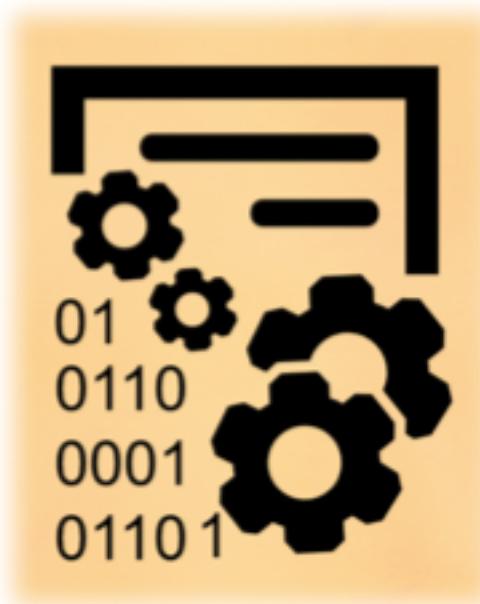


The Tale of Two Source-code Analysis Tools

Learning and experiences

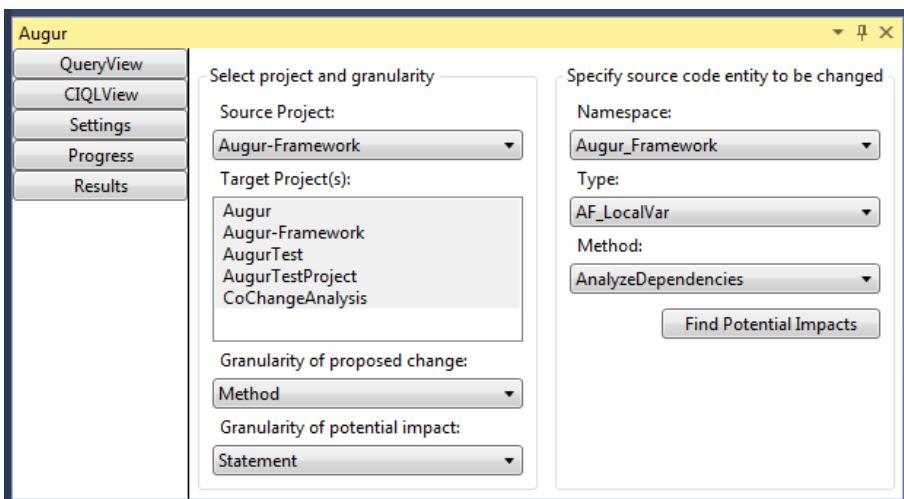


Tushar Sharma

Athens University of Economics and Business

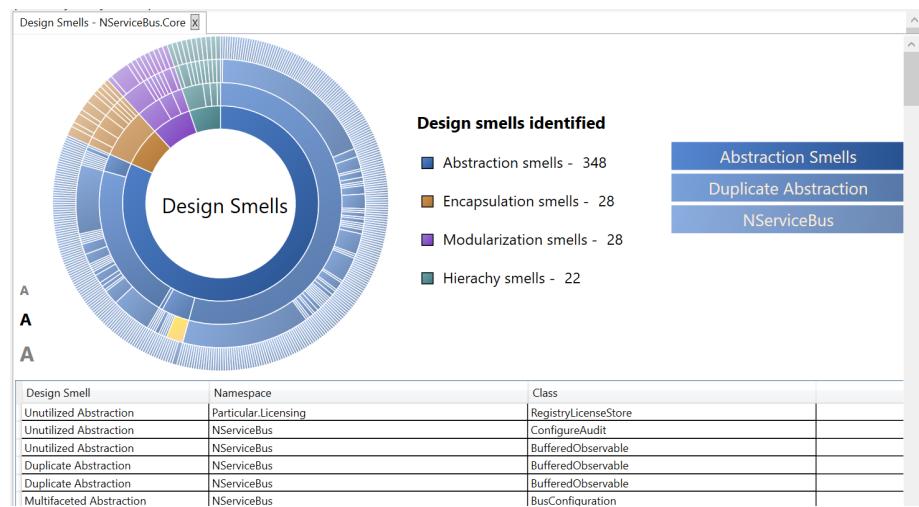
Funded by SENECA project under Marie-Skłodowska Curie Actions

Tools



Augur

A change impact analysis tool



Designite

A software design quality assessment tool

(<http://www.designite-tools.com>)

Tools

The screenshot shows the Augur software interface. On the left, a sidebar menu lists "Augur", "QueryView", "CQLView", "Settings", "Progress", and "Results". The main area is titled "Select project and granularity". It includes dropdown menus for "Source Project" (set to "Augur-Framework"), "Target Project(s)" (listing "Augur", "Augur-Framework", "AugurTest", "AugurTestProject", and "CoChangeAnalysis"), "Granularity of proposed change" (set to "Method"), and "Granularity of potential impact" (set to "Statement"). To the right, another panel titled "Specify source code entity to be changed" contains dropdowns for "Namespace" (set to "Augur_Framework"), "Type" (set to "AF_Localizar"), and "Method" (set to "AnalyzeDependencies"). A "Find Potential Impacts" button is at the bottom.

Augur

A change impact analysis tool

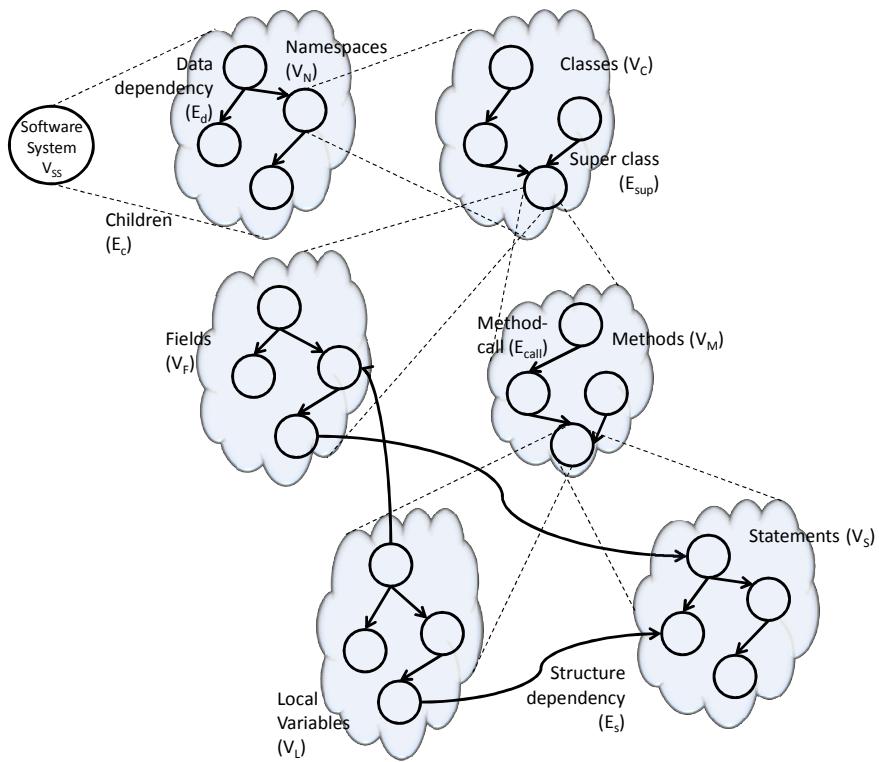


Designite

A software design quality assessment tool



Features



- **Change impact analysis with multiple granularity support**
 - Cutting across projects, namespaces, classes, methods, fields, and statements
- **Intra-granular queries**
 - Supporting a query where a change and the associated impact could be on different granularities



Features

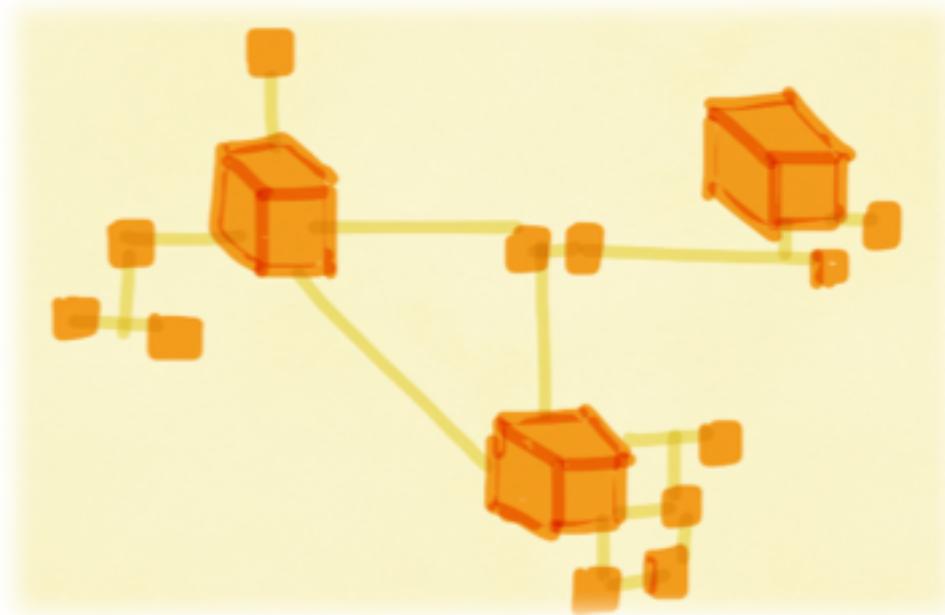
- **Change Impact Query Language (CIQL)**
 - For large scale batch querying – opening a new set of applications of CIA

*CIQL::get “<Granularity (Impact)>”
[within “<Scope>”]
[with “<Depth>”] where
“<Entity>” is
“<Granularity>”.*

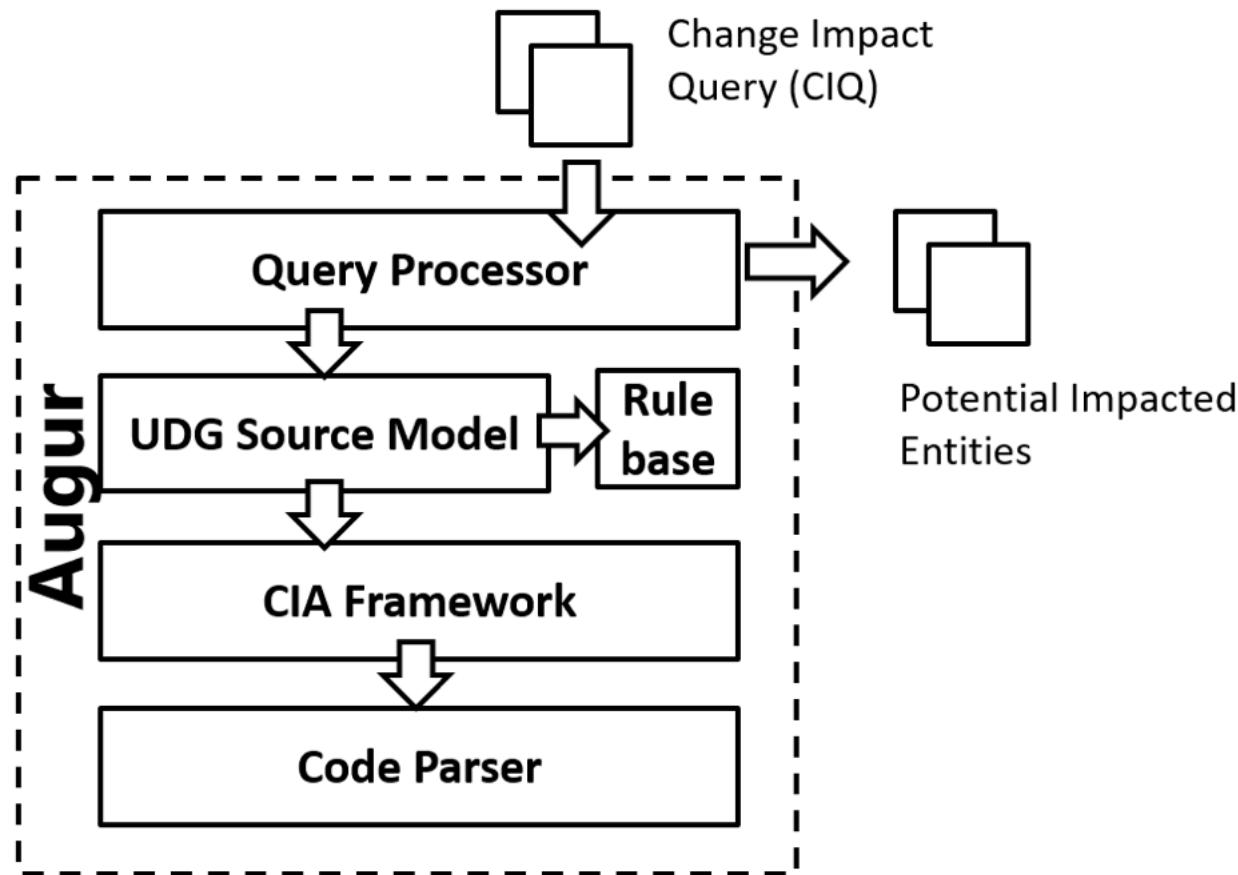


Features

- **Support for extended dependencies**
 - Data
 - Control
 - Semantic
 - Environment



Architecture

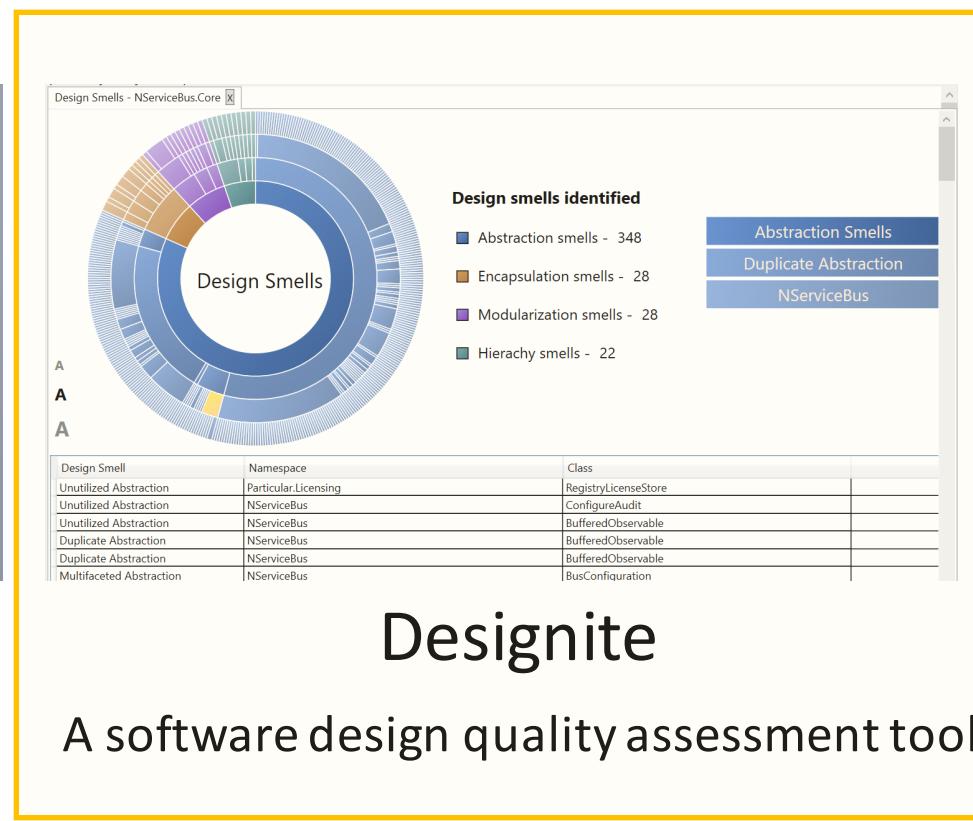


Tools

The screenshot shows the Augur software interface. On the left, a sidebar menu includes: Augur, QueryView, CQLView, Settings, Progress, and Results. The main area has two sections: 'Select project and granularity' and 'Specify source code entity to be changed'. In the first section, 'Source Project' is set to 'Augur-Framework' and 'Target Project(s)' list 'Augur', 'Augur-Framework', 'AugurTest', 'AugurTestProject', and 'CoChangeAnalysis'. In the second section, 'Namespace' is 'Augur_Framework', 'Type' is 'AF_Localizar', and 'Method' is 'AnalyzeDependencies'. A 'Find Potential Impacts' button is at the bottom.

Augur

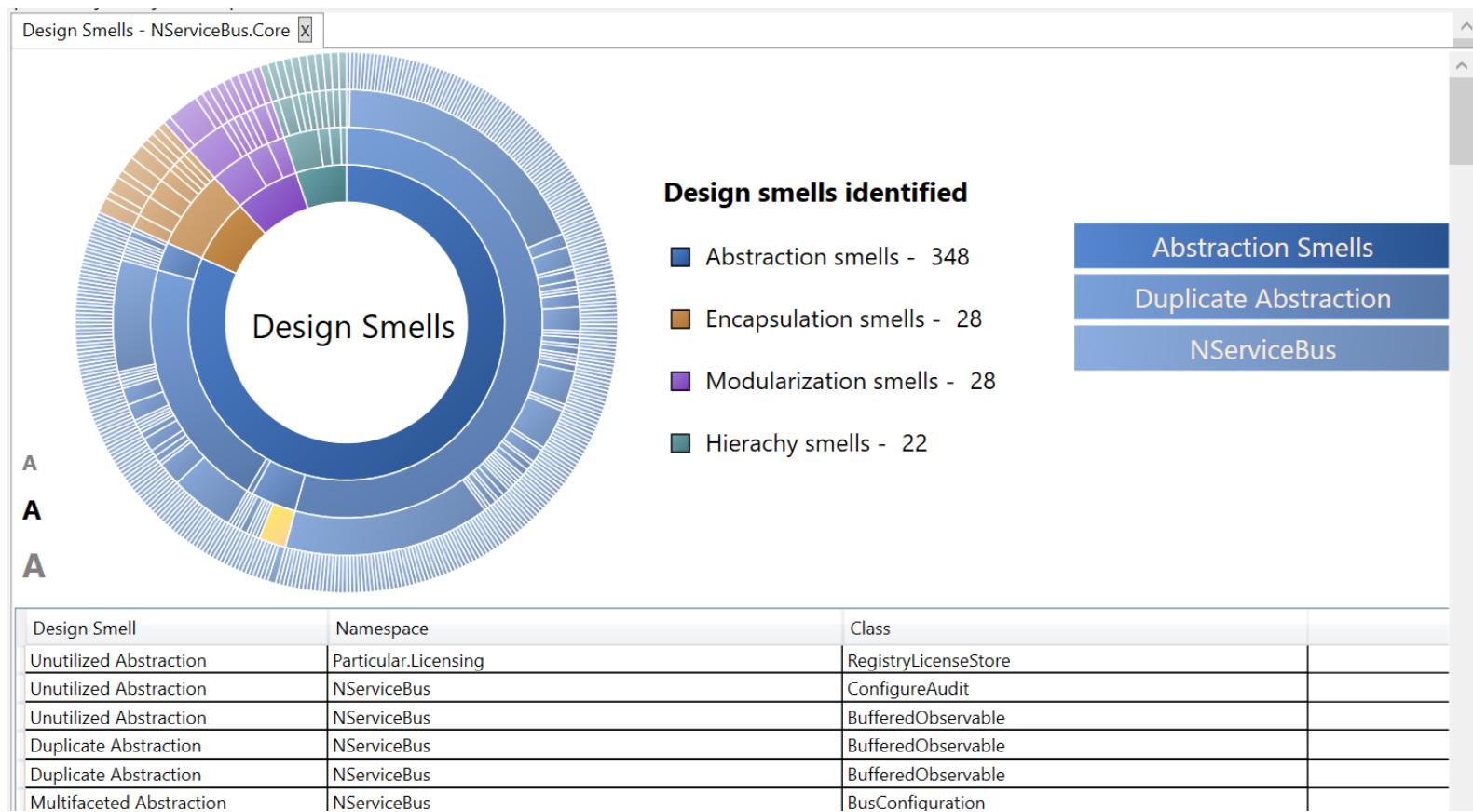
A change impact analysis tool





Features

- Supports detection of 19 design smells and 11 implementation smells



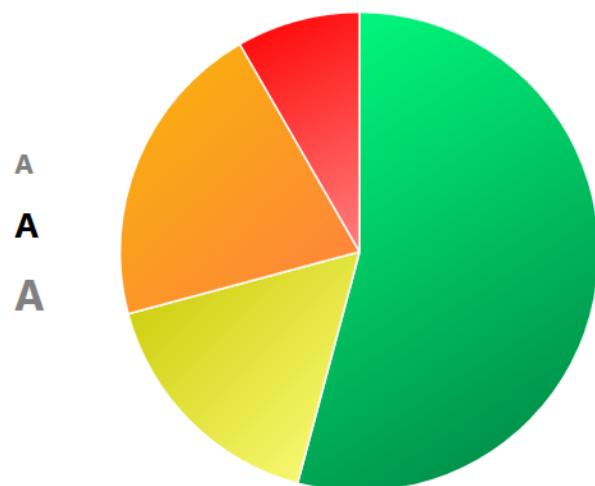


Features

- Supports computation of various metrics with custom thresholds

WMC - Weighted Method per Class

Distribution of types based on metric thresholds



- Types honoring the metric threshold - 13
- Types slightly above the metric threshold - 4
- Types quite above the metric threshold - 5
- Types dangerously above the metric threshold - 2

Reset thresholds

Green threshold	<input type="range"/>	80.00
Yellow threshold	<input type="range"/>	100.00
Orange threshold	<input type="range"/>	150.00

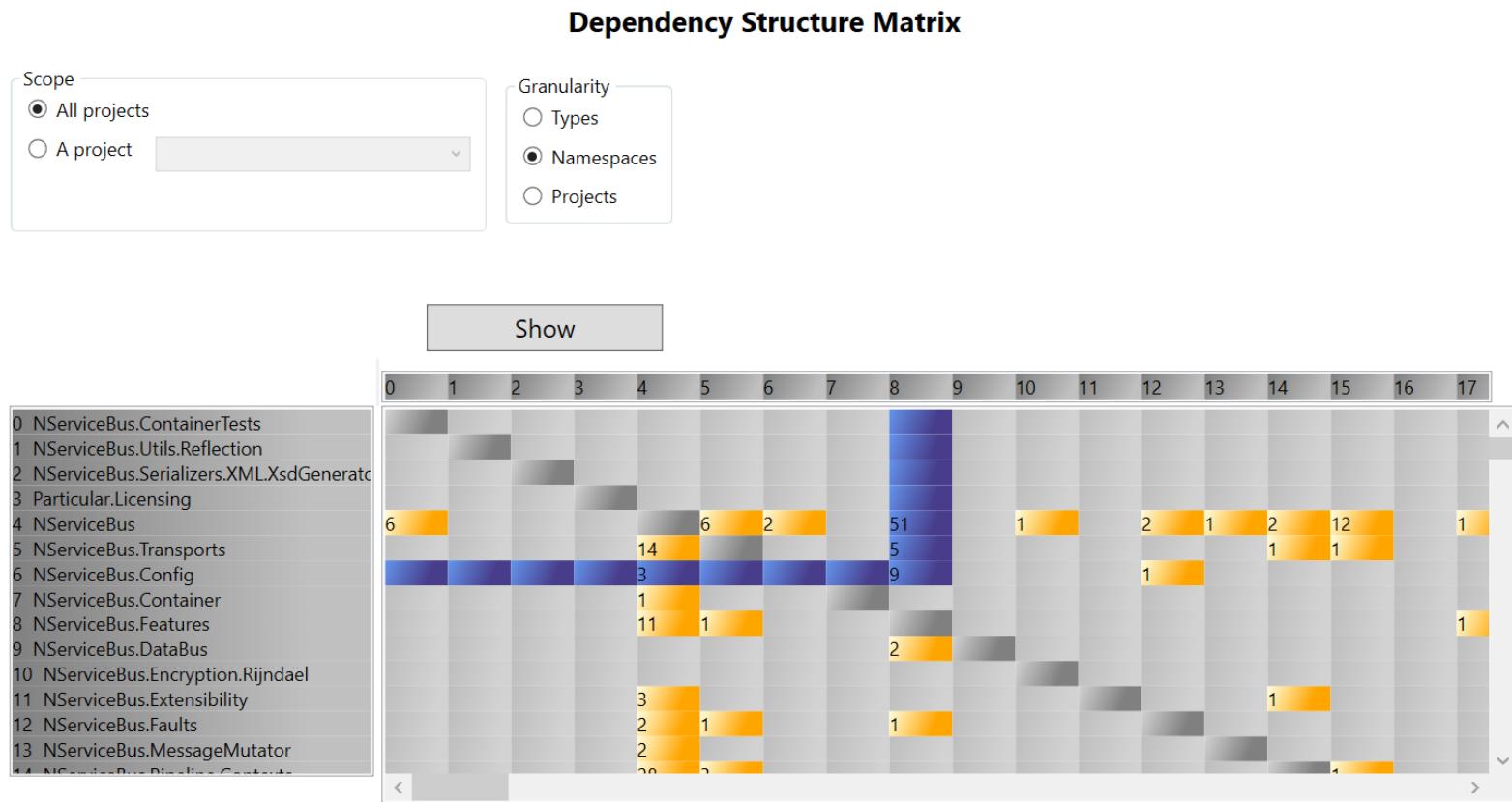
Select metric

- DIT Fan-in Fan-out LCOM LOC NC NOF NOM NOPF NOPM NOP WMC



Features

- Provides Dependency Structure Matrix

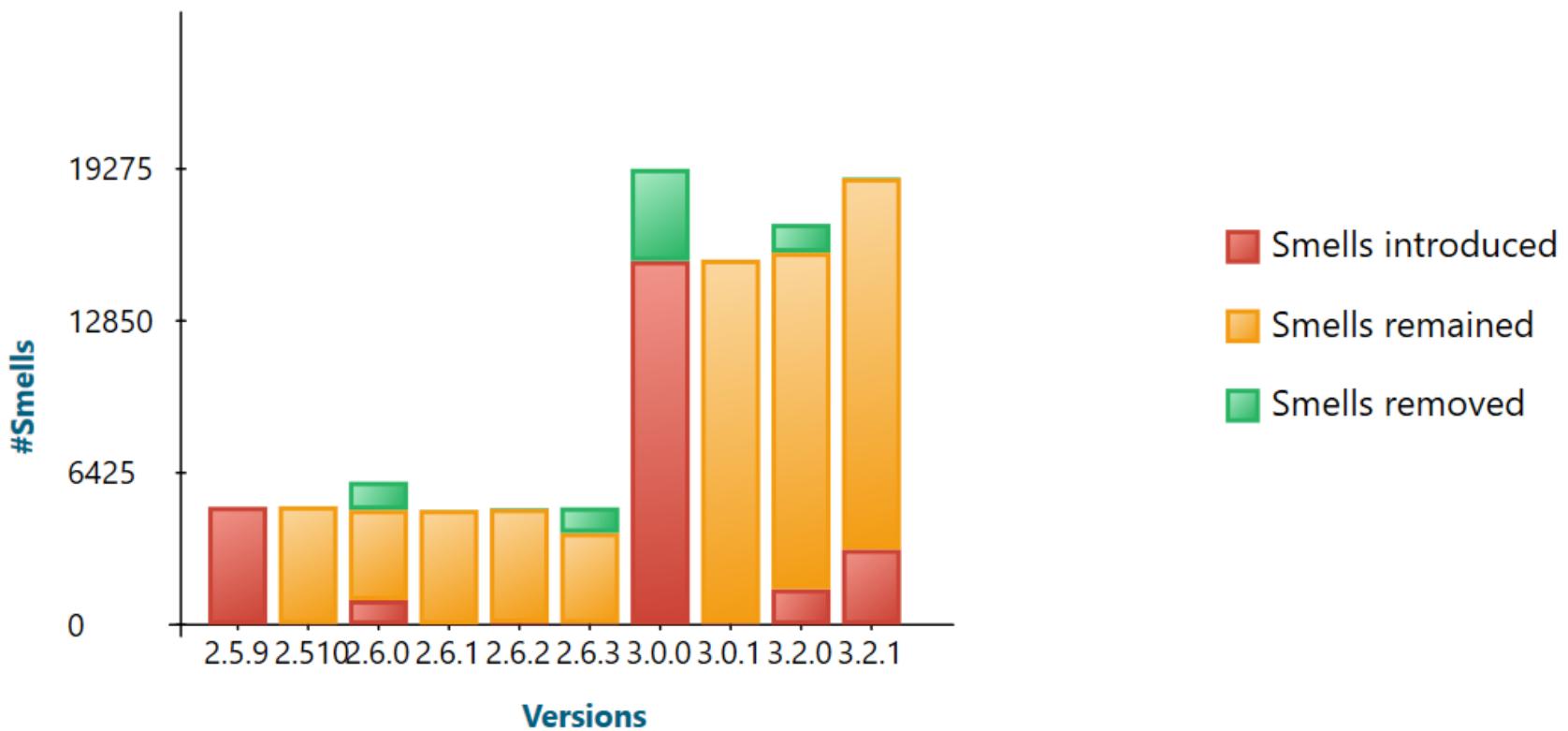




Features

- Performs Trend Analysis

Trend Analysis



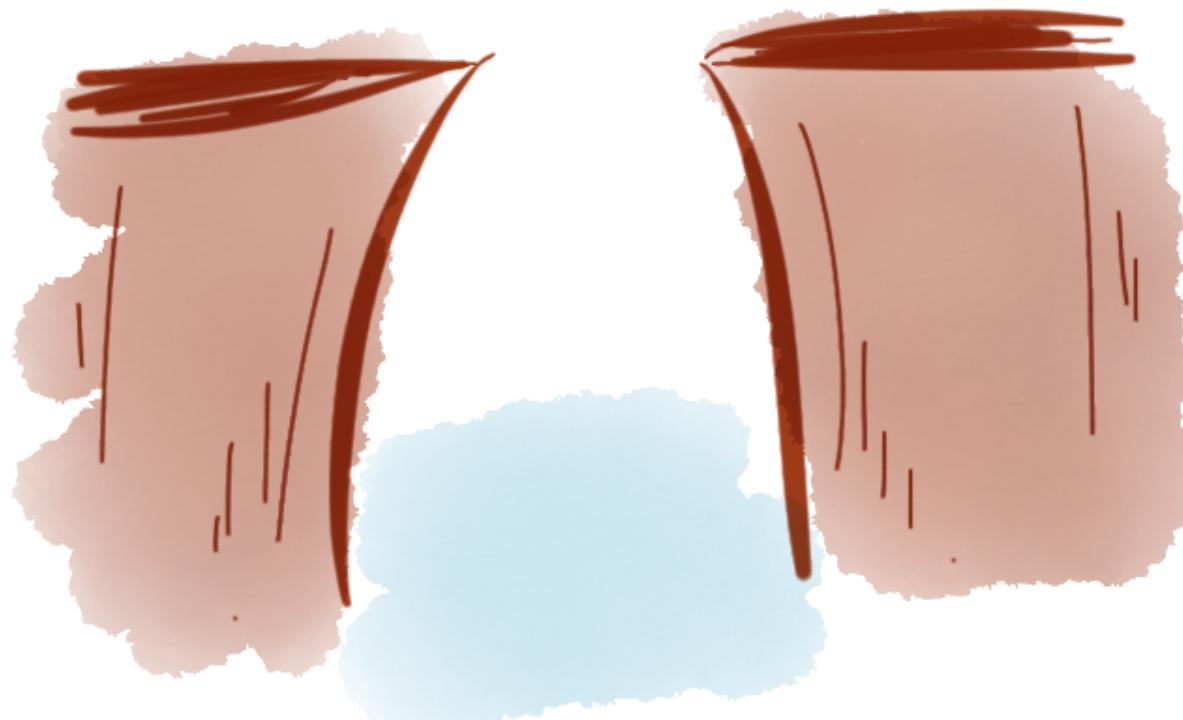


Learning and experiences

The Big Gap

between Academics and Industry

Proposing a new research program in a corporate research organization is not easy!



The Big Gap

between Academics and Industry



Learning (as a researcher)

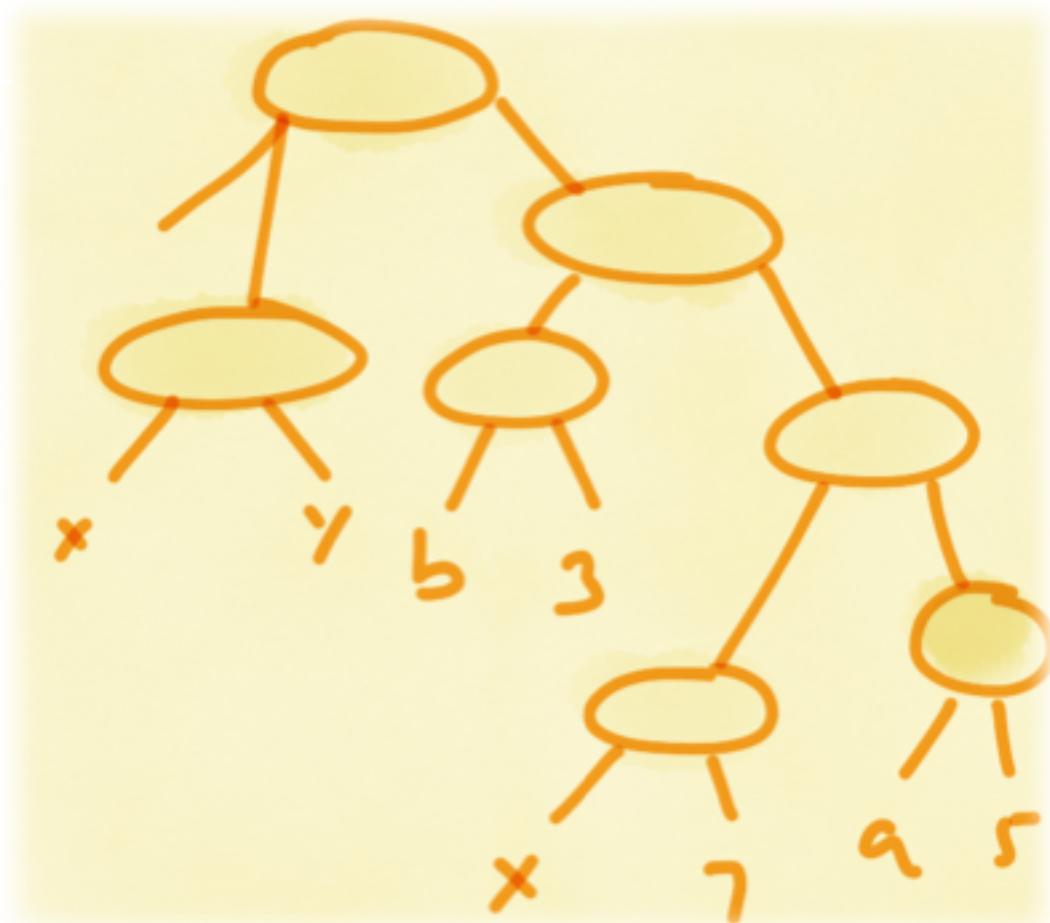
- Make sure the availability of artifacts and their broader applicability

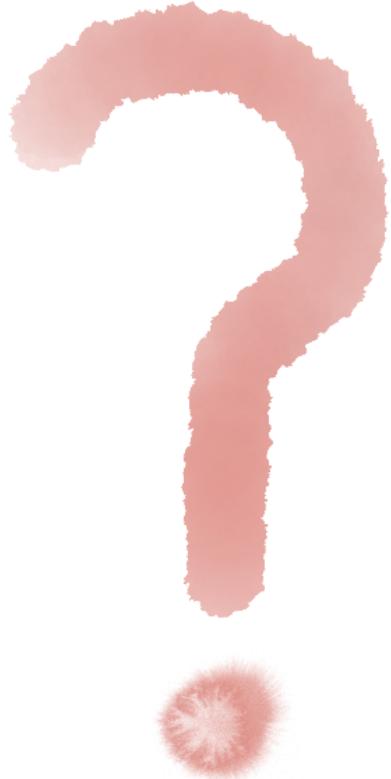


Parsing mechanism

Various options for collecting source code information

- String manipulation
- Reflection
- AST
- Byte code analysis





Which AST library?

CSParser

MS Roslyn

#Recognize!

NRefactory

Metaspec



Parsing mechanism

Selection criteria

- License
- Features
- Cost
- Community support
- Future proof-ness



Architecture



Plug-in
or
Independent application



Console application

Options

- Use conditional compilation
(using ConditionalAttribute)
- Duplicate the code-base
- Perform architecture refactoring





Console application



Experience:

- Architecture refactoring is expensive but effective!
- Support for architecture refactoring within IDEs is not sufficient



Extensibility

Smell detection logic must be extensible
i.e. new rules can be added without any change in source code analysis logic and user interface

Learning:

- The role of appropriate design is important



Information dissemination

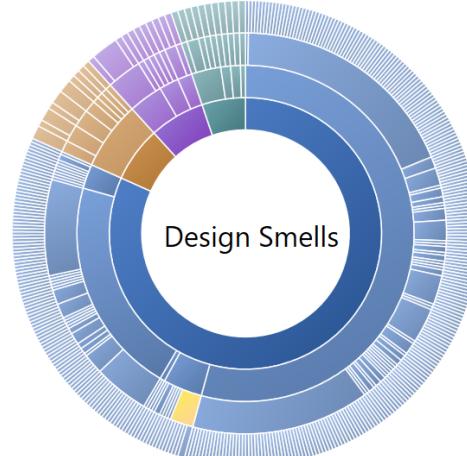
Producing useful information is desirable;
presenting it well to the user is the extra mile.

Designite - A Software Design Quality Assessment Tool

Project

- NServiceBus
- NServiceBus.MessageInterfaces.Tests
- NServiceBus.ObjectBuilder.CastleWindsor
- NServiceBus.ObjectBuilder.StructureMap
- NServiceBus.ObjectBuilder.Autofac
- NServiceBus.ObjectBuilder.Unity
- ObjectBuilder.Tests
- NServiceBus.ObjectBuilder.Spring
- NServiceBus.ObjectBuilder.Ninject
- NServiceBus.SagaPersisters.InMemory.Tests
- XsdGenerator
- NServiceBus.Logging.Tests
- NServiceBus.PowerShell
- LicenseInstaller
- ReturnToSourceQueue
- NServiceBus.Core
 - Metrics
 - Class-level Metrics
 - Method-level Metrics
 - Design Smells
 - Abstraction Design Smells
 - Encapsulation Design Smells
 - Modularization Design Smells
 - Hierarchy Design Smells
- NServiceBus.GatewayPersister.NHibernate
- NServiceBus.Persistence.NHibernate.Tests
- NServiceBus.NHibernate
- NServiceBus.SagaPersisters.NHibernate.Tests
- NServiceBus.Unicast.Subscriptions.NHibernate
- NServiceBus.TimeoutPersisters.NHibernate
- NServiceBus.Hosting.Tests
- NServiceBus.Host
- NServiceBus.Tests

Design Smells - NServiceBus.Core



Design smells identified

Abstraction smells - 348

Encapsulation smells - 28

Modularization smells - 28

Hierarchy smells - 22

Abstraction Smells

Duplicate Abstraction

NServiceBus

Broken Modularization:

This smell arises when data and/or methods that ideally sh abstractions.

Cause of the smell:

The tool detected the smell in this class because it contains declared in this class: `LicenseType`, `MaxThroughputPerSecond`

Design Smell Namespace

Design Smell	Namespace	Class
Unutilized Abstraction	Particular.Licensing	RegistryLicenseStore
Unutilized Abstraction	NServiceBus	ConfigureAudit
Unutilized Abstraction	NServiceBus	BufferedObservable
Duplicate Abstraction	NServiceBus	BufferedObservable
Duplicate Abstraction	NServiceBus	BufferedObservable
Multifaceted Abstraction	NServiceBus	BusConfiguration

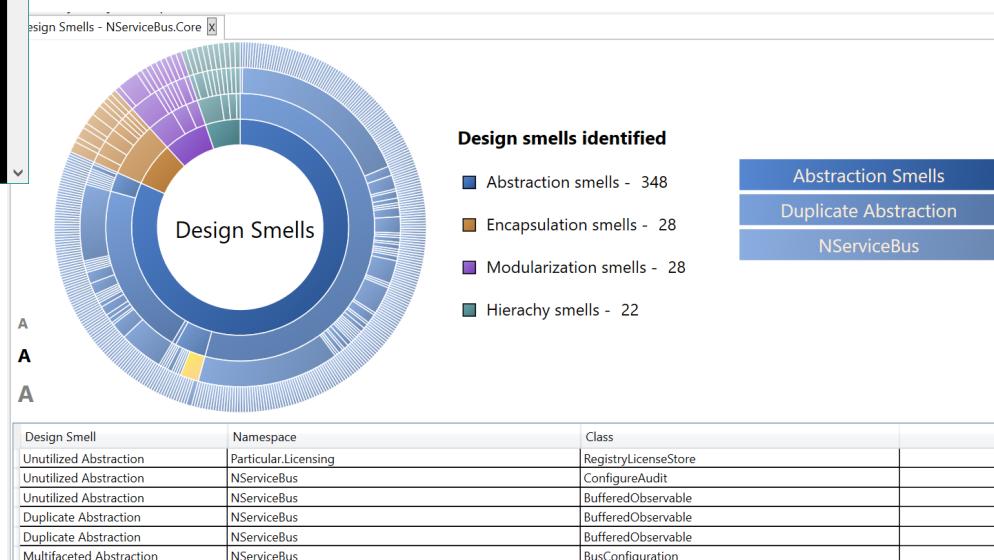


Information dissemination

Different types of users, different requirements.

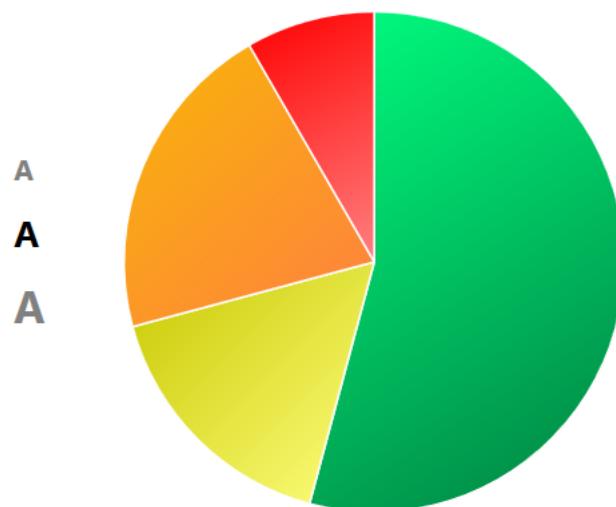
```
C:\Windows\System32\cmd.exe
C:\Program Files (x86)\Designite\Designite>DesigniteConsole.exe
Usage: DesigniteConsole <InputArg> <Option> <ExportPath>
  where <InputArg> is either -
    Solution file name with path in double quotes, or
    Batch file name with path in double quotes.
  and <Option> is either -
    -E      to export analysis results in an Excel sheet, or
    -X      to export analysis results in an XML file
    -C      to export analysis results in CSV files
  and <ExportPath> refers to the export file name with path in double quotes
for Excel or XML format. For CSV format, specify the folder path.

C:\Program Files (x86)\Designite\Designite>
```





Paying attention to user requirements



Distribution of types based on metric thresholds

- Types honoring the metric threshold - 13
- Types slightly above the metric threshold - 4
- Types quite above the metric threshold - 5
- Types dangerously above the metric threshold - 2

Reset thresholds

Green threshold	<input type="range"/>	80.00
Yellow threshold	<input type="range"/>	100.00
Orange threshold	<input type="range"/>	150.00

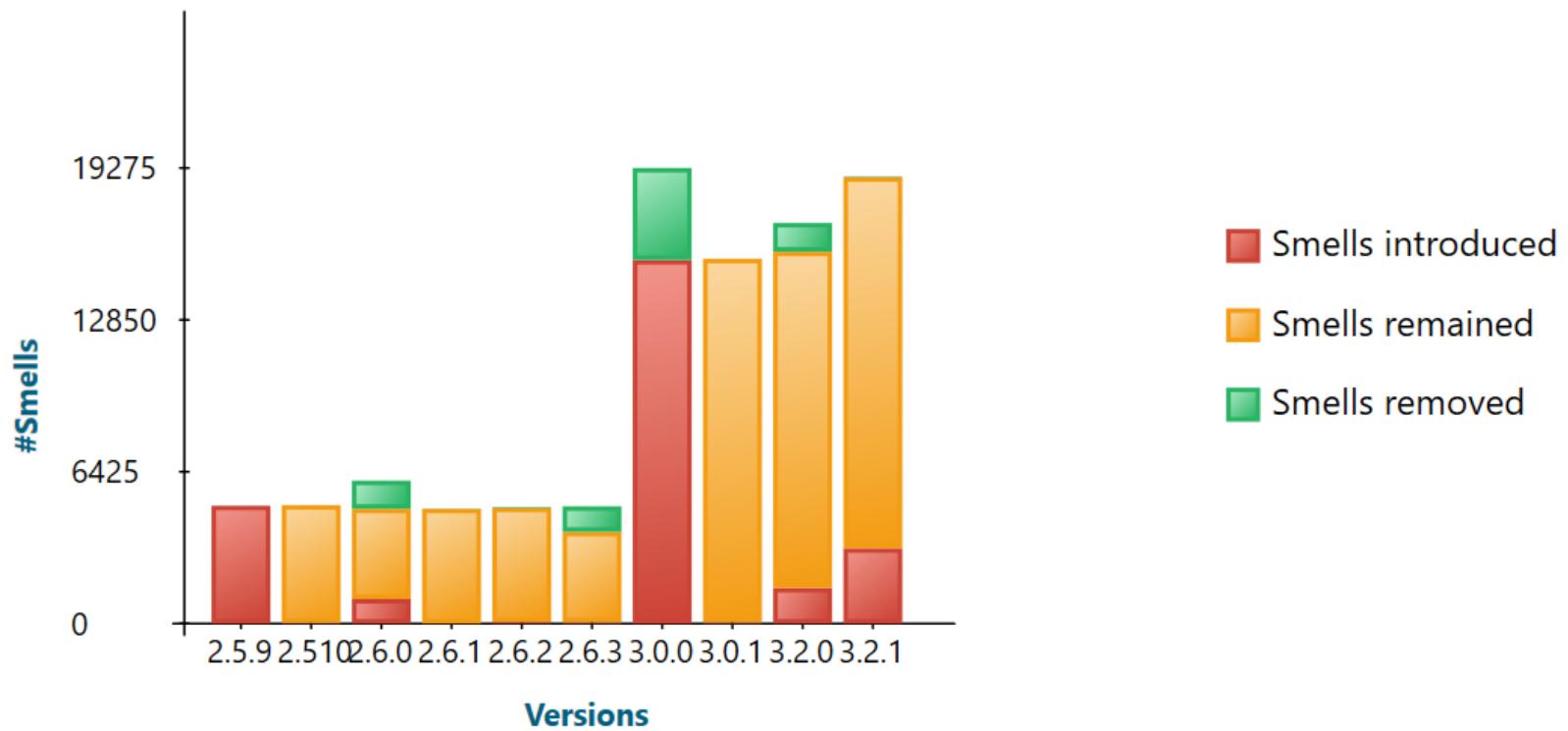
Select metric

- DIT
- Fan-in
- Fan-out
- LCOM
- LOC
- NC
- NOF
- NOM
- NOPF
- NOPM
- NOP
- WMC



Paying attention to user requirements

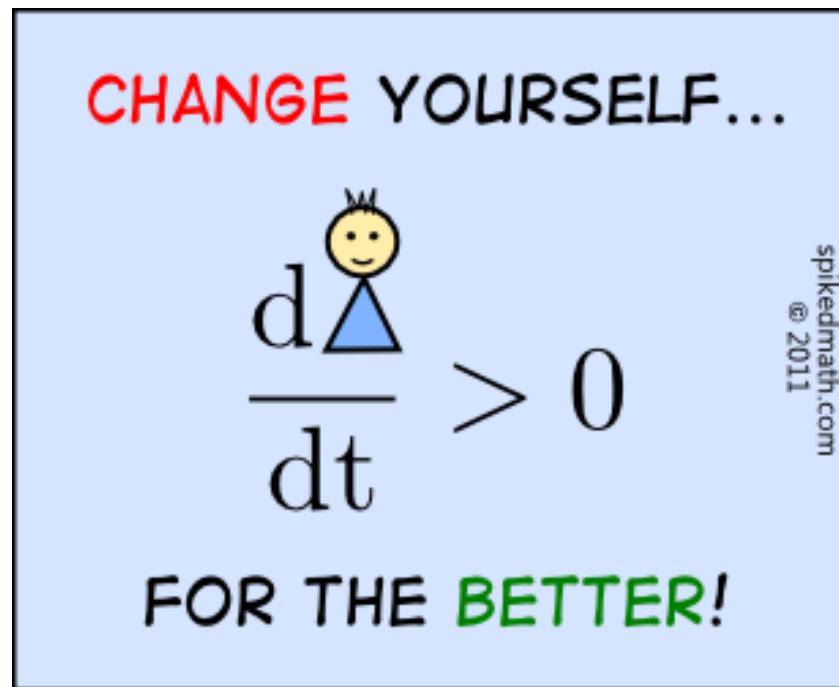
Trend Analysis



References

- [1] Tushar Sharma, Pratibha Mishra, and Rohit Tiwari. 2016. Designite: a software design quality assessment tool. In *Proceedings of the 1st International Workshop on Bringing Architectural Design Thinking into Developers' Daily Activities* (BRIDGE '16). ACM, New York, NY, USA, 1-4. DOI: <http://dx.doi.org/10.1145/2896935.2896938>
- 2] Tushar Sharma, Girish Suryanarayana. Augur: Incorporating Hidden Dependencies and Variable Granularity in Change Impact Analysis. Submitted at SCAM 2016, waiting for the decision.

Thank you!!



Courtesy: spikedmath.com

Tushar Sharma
tusharsharma@ieee.org
@Sharma__Tushar