Agile Software Assessment

Oscar Nierstrasz
Software Composition Group
scg.unibe.ch
Agility in Moose

Motivation

Challenges
The need for Agile Software Assessment
Legacy code is hard to understand
The architecture

... is not in the code
Developers spend more time reading than writing code
Specialized analyses require custom tools
Agility in Moose

Softwareaut | BugsLife | Clustering | Metanool | ...
Chronia    | CodeCity | DynaMoose | Hapax    | SmallDude | Yellow Submarine
Repository | FAMIX    | Fame      | UI       | Mondrian  | EyeSee
Java       | iPlasma  | MSE       | Smalltalk |
Moose is a platform for software and data analysis.
Nierstrasz et al. The Story of Moose. ESEC/FSE 2005
System complexity

Lanza et al. Polymetric Views. TSE 2003
Demo: finding deprecated classes that are still in use...
Demo: a package browser for name cohesion

Bunge et al. Scripting Browsers with Glamour. ESUG 2009
Challenges for Agile Software Assessment

Agile Modeling

Customization

Continuous Assessment
Agile Modeling
Load the model in the morning, analyze it in the afternoon
Problems

Unstructured text

Unknown languages

Heterogeneous projects

Perin et al. Recovery and Analysis of Transaction Scope from Scattered Information in Java Enterprise Applications. ICSM 2010
Ideas

Incrementally refine island grammars

Exploit example mappings to generate fact extractors

Exploit similarities between languages (adapt and compose)

Exploit eg indentation as a proxy for structure

Nierstrasz et al. Example-Driven Reconstruction of Software Models. CSMR 2007
Backtracking is expensive!

Changing the islands changes the water!

Tokenization and memoization help :-)
Customization
Build a new assessment tool in ten minutes
Problems

What tools do developers really need?

What is a unifying meta-model for tool construction?

What are appropriate meta-tools?
Analyze developer needs (!)

“Malleable” IDE (not just plug-ins)

From meta-models to interactive DSLs
Progress

Need dynamic flows for Glamour

Need better ways to navigate in Moose, in the IDE ...
Continuous Assessment
Challenge

“What will my code change impact?”
Problems

Understanding architectural constraints

Implicit dependencies

Ripple effects


Aryani, Perin et al. Can We Predict Dependencies Using Domain information? WCRE 2011


Ideas

Architecture monitoring (beyond layers)

Ecosystem monitoring
Uncovering “Software Architecture in the Wild”

Various techniques to ensure conformance

Diverse views of SA

Analysis is problematic
Conclusion

Current IDEs offer developers poor support for software assessment

Developers need support for agile modeling, customization and continuous assessment