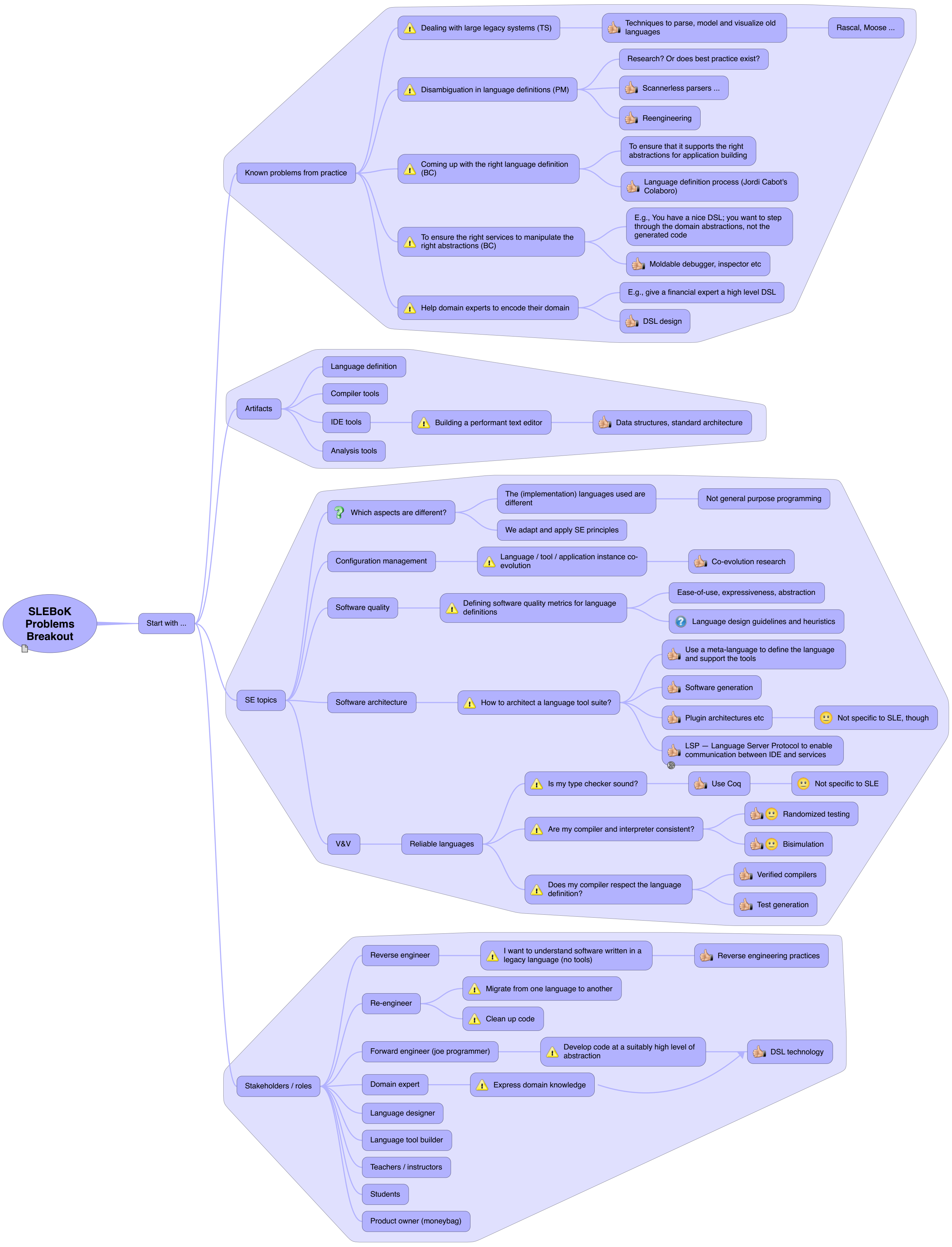
SLEBoK Problems Breakout
Start with ...
Known problems from practice
Dealing with large legacy systems (TS)
Techniques to parse, model and visualize old languages
Rascal, Moose ...
Disambiguation in language definitions (PM)
Research? Or does best practice exist?
Scannerless parsers ...
Reengineering
Coming up with the right language definition (BC)
To ensure that it supports the right abstractions for application building
Language definition process (Jordi Cabot's Colaboro)
To ensure the right services to manipulate the right abstractions (BC)
E.g., You have a nice DSL; you want to step through the domain abstractions, not the generated code
Moldable debugger, inspector etc
Help domain experts to encode their domain
E.g., give a financial expert a high level DSL
DSL design
Artifacts
Language definition
Compiler tools
IDE tools
Building a performant text editor
Data structures, standard architecture
Analysis tools
SE topics
Which aspects are different?
The (implementation) languages used are different
Not general purpose programming
We adapt and apply SE principles
Configuration management
Language / tool / application instance co-evolution
Co-evolution research
Software quality
Defining software quality metrics for language definitions
Ease-of-use, expressiveness, abstraction
Language design guidelines and heuristics
Software architecture
How to architect a language tool suite?
Use a meta-language to define the language and support the tools
Software generation
Plugin architectures etc
Not specific to SLE, though
LSP — Language Server Protocol to enable communication between IDE and services
V&V
Reliable languages
Is my type checker sound?
Use Coq
Not specific to SLE
Are my compiler and interpreter consistent?
Randomized testing
Bisimulation
Does my compiler respect the language definition?
Verified compilers
Test generation
Stakeholders / roles
Reverse engineer
I want to understand software written in a legacy language (no tools)
Reverse engineering practices
Re-engineer
Migrate from one language to another
Clean up code
Forward engineer (joe programmer)
Develop code at a suitably high level of abstraction
DSL technology
Domain expert
Express domain knowledge
Language designer
Language tool builder
Teachers / instructors
Students
Product owner (moneybag)

Oscar Nierstrasz (Collector), Tijs van der Storm, Peter Mosses, Andrei Chis, Benoit Combemale, Thomas Deguele
2017-08-22

/ SLEBoK Problems Brea… / Start with ... / SE topics / Software architectur… / How to architect a l… / LSP — Language Serve…

http://langserver.org/