

Understanding Software

Paul Klint



Centrum Wiskunde & Informatica



#UnderstandingSoftware

*Where we are
coming from
(circa 1800)*

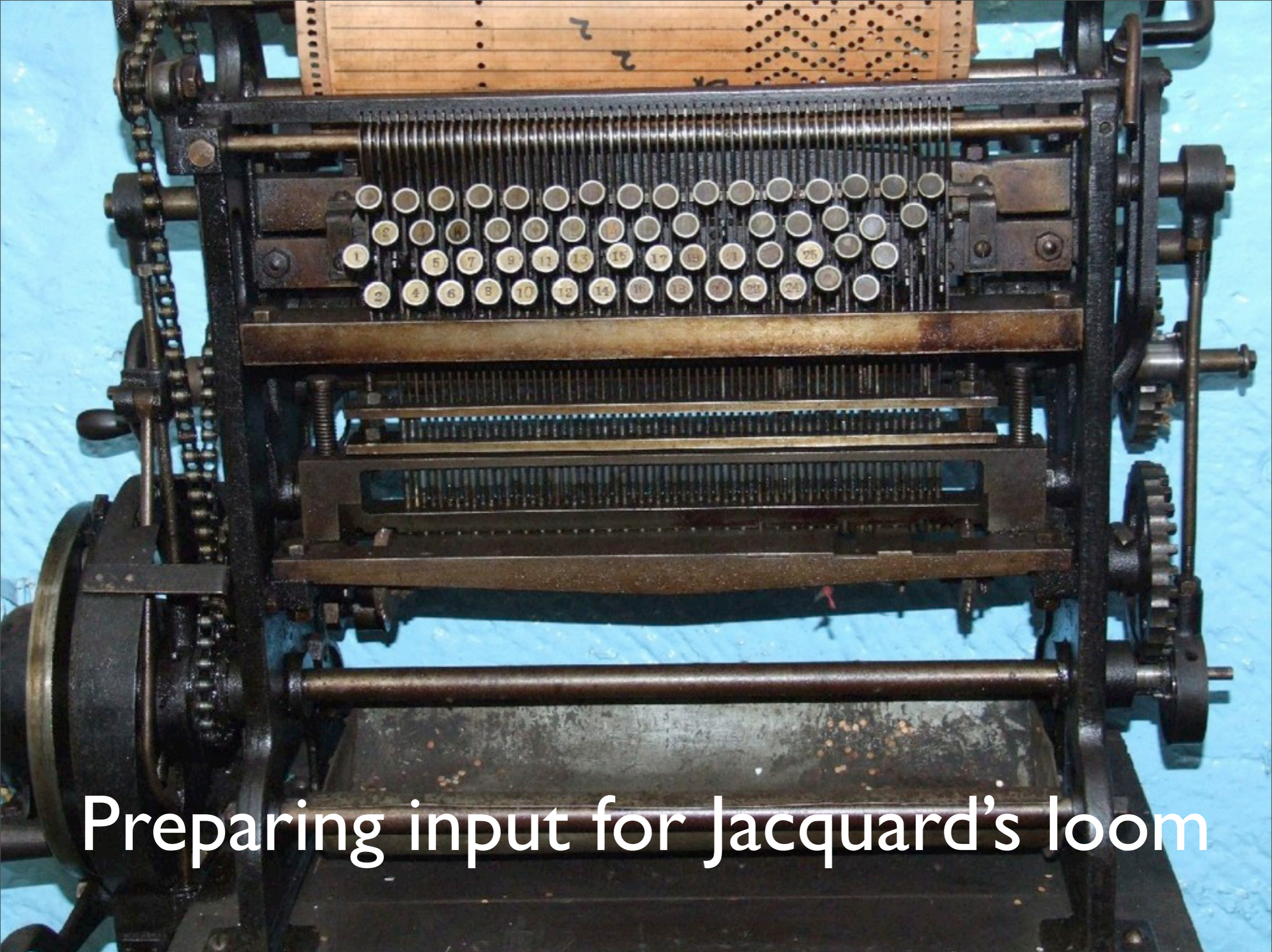
(circa 1800)

Jacquard Looms at
Family Heirloom
Weavers in Red
Lion, PA.

<http://youtu.be/NSjmFD6Q7hw>

*Using this
development
environment*

ENVIRONMENT



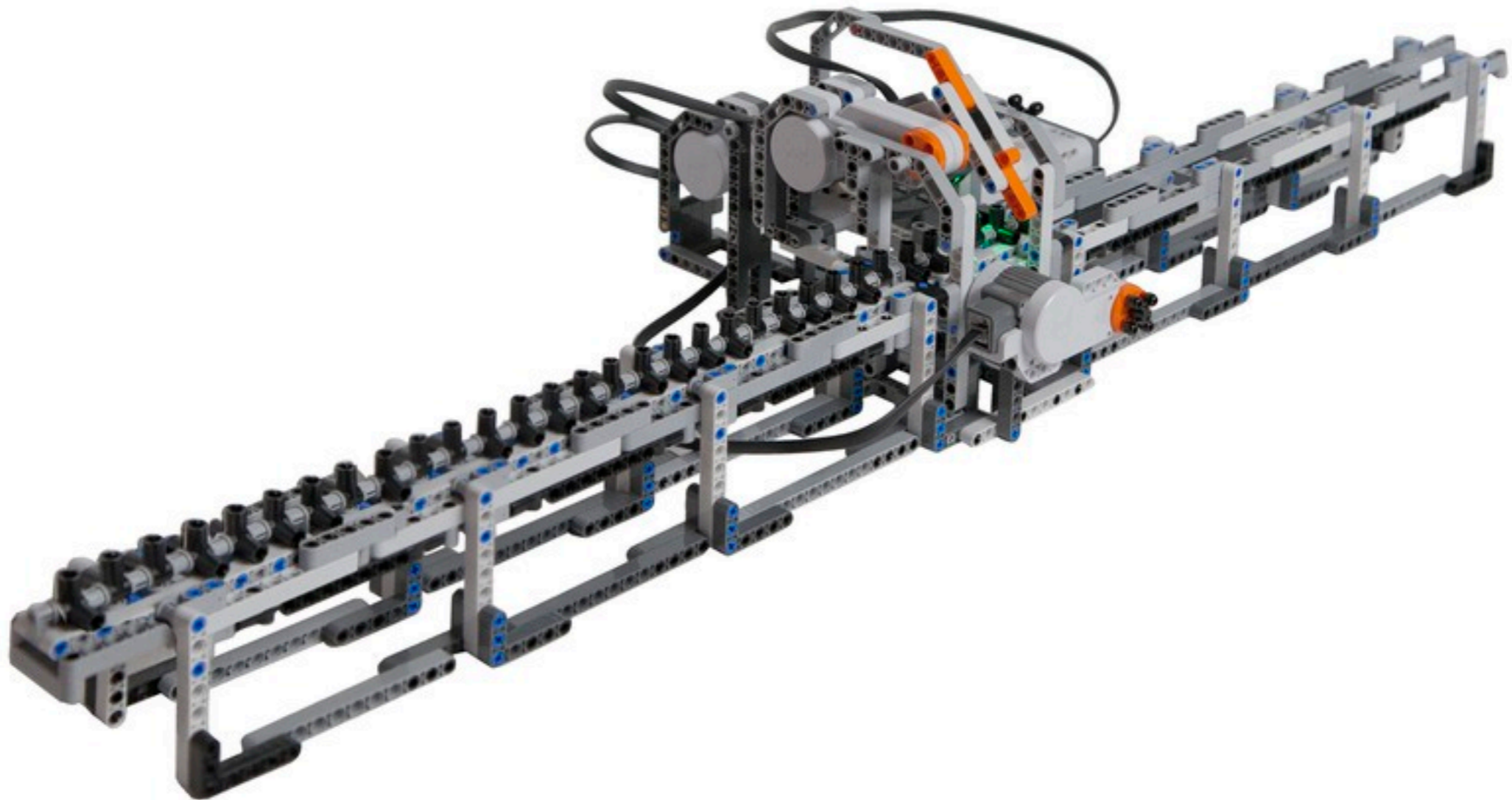
Preparing input for Jacquard's loom

For the rest of this story ...

For the rest of this story ...

... *visit the*
Turing's Legacy
exhibition during the
breaks or reception

Where you can also see the Lego Turing Machine



<http://www.legoturingmachine.org/>

Credits:

Construction: Jeroen van den Bos, Davy Landman

Film: Andre Theelen

Fast forward to today ...

Fast forward to today ...

Integrated Development Environments for All



IDE lovers



Emacs haters

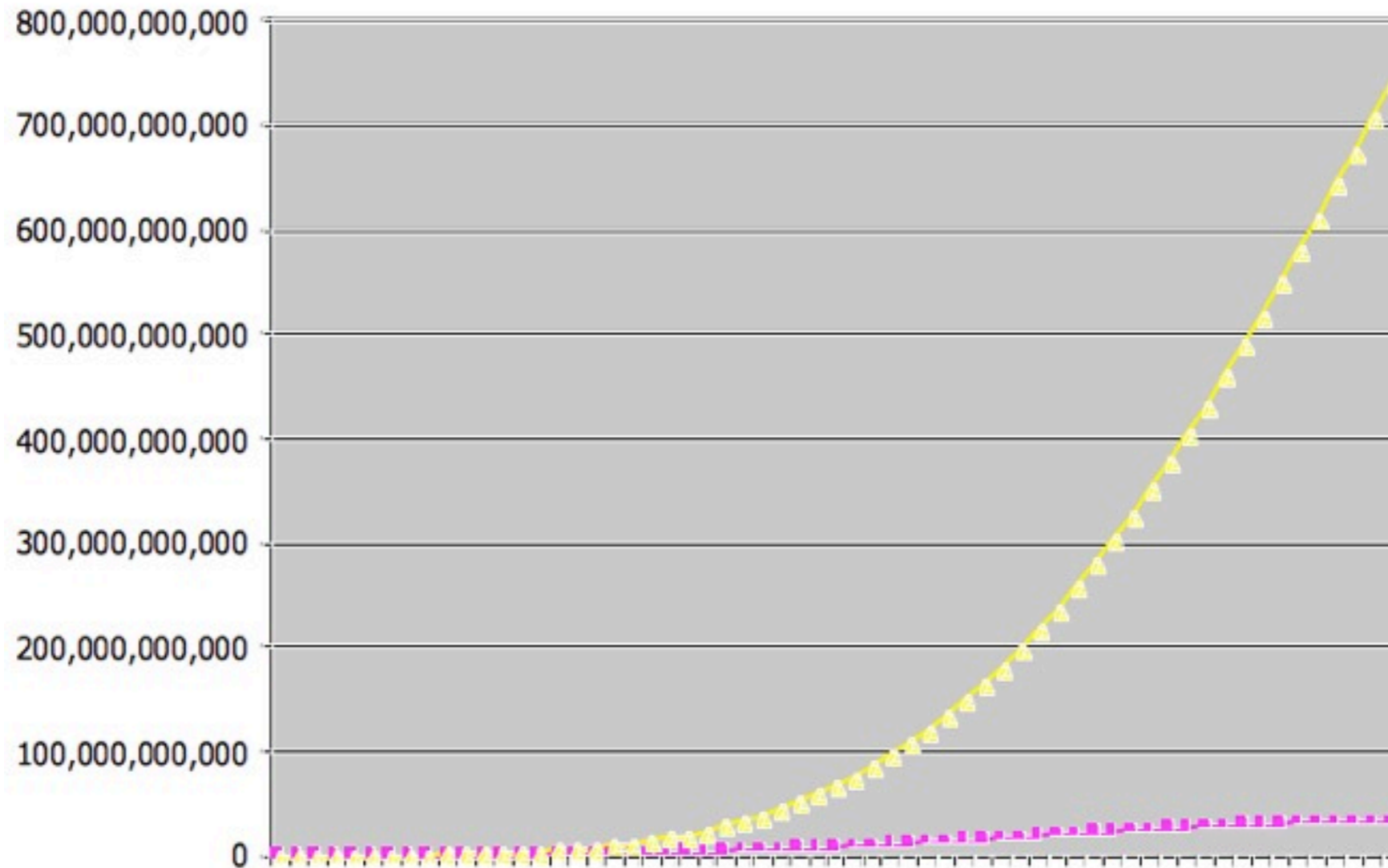
Vim haters

IDE haters

*We have created a lot
of software*

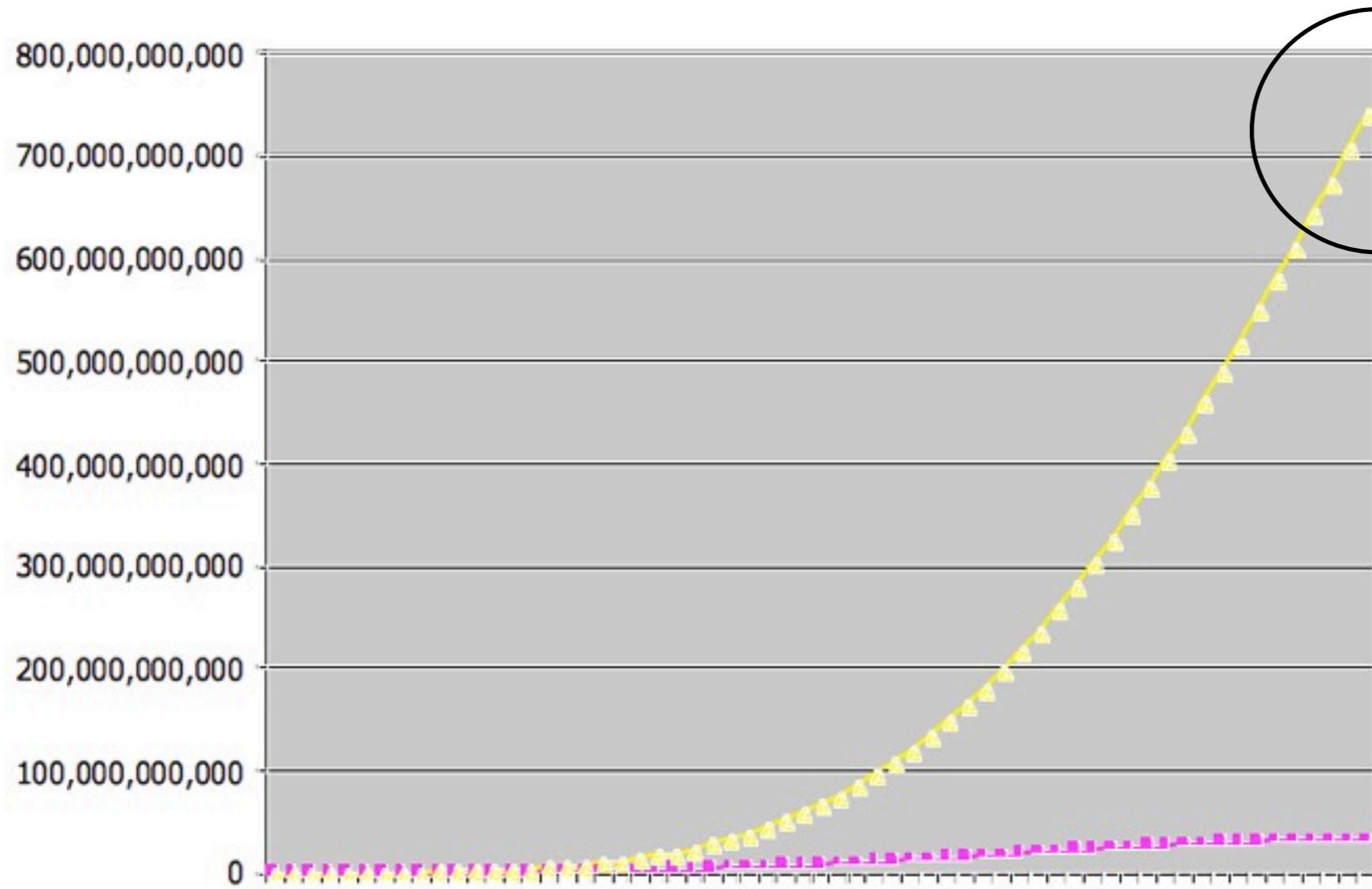
of software

~900.000.000.000 LOC



Source:
Booch, 2005
Also: Capers Jones

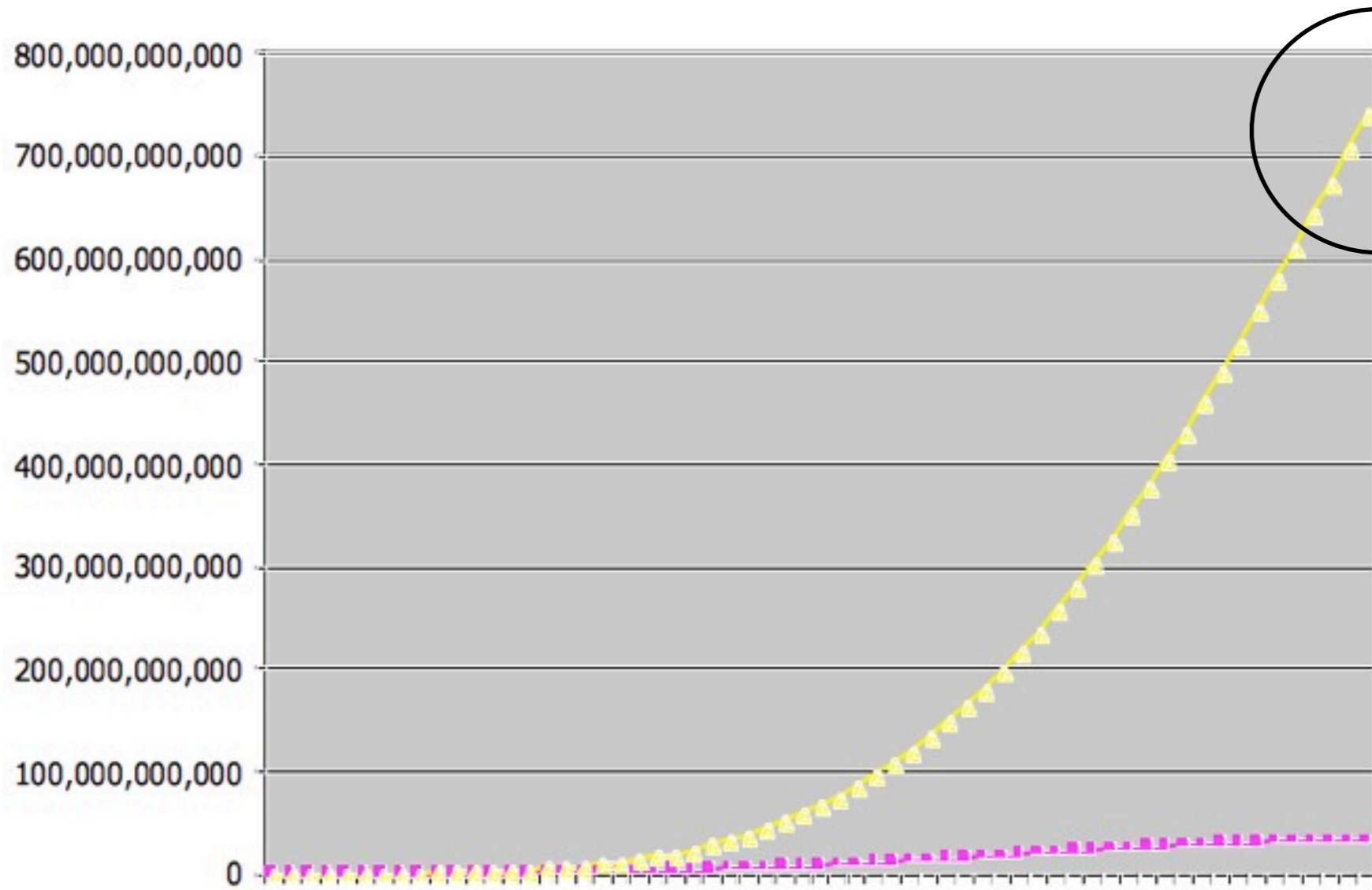
~900.000.000.000 LOC



Cumulative lines of code

Source:
Booch, 2005
Also: Capers Jones

~900.000.000.000 LOC



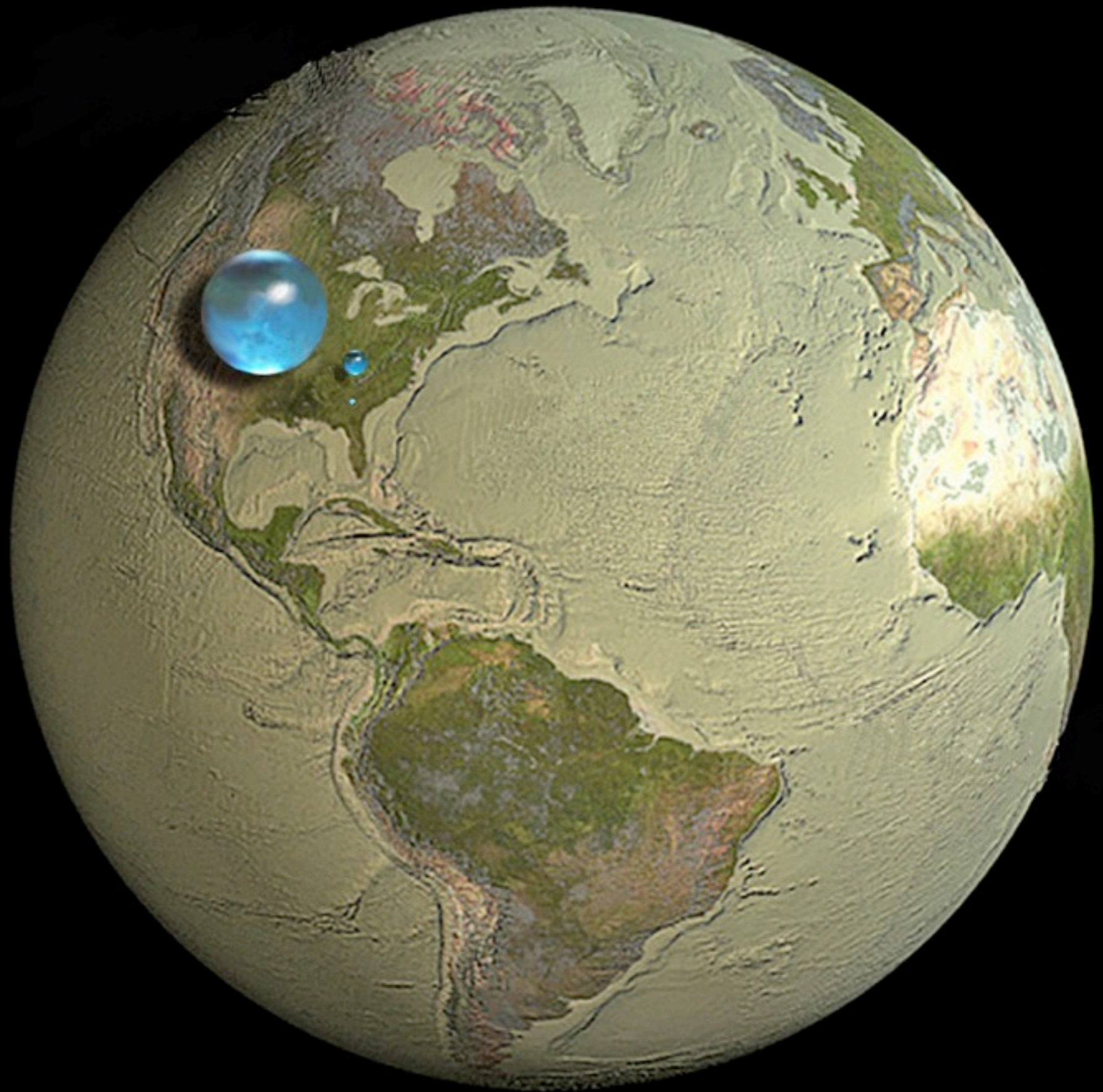
Cumulative lines of code

Source:
Booch, 2005
Also: Capers Jones

Nearly One (European) Billion Lines!

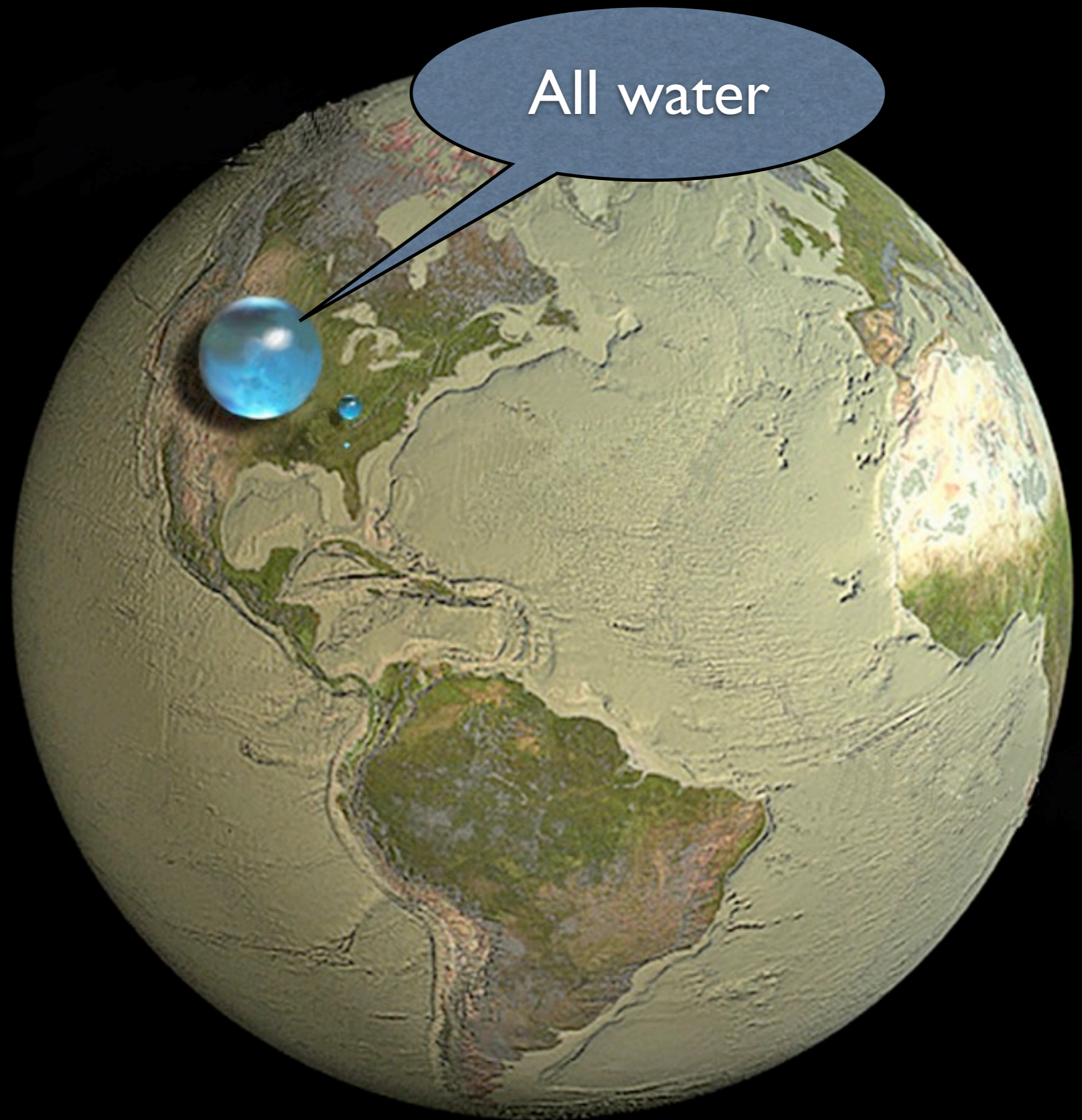
How
much
is
that?

**How
much
is
that?**



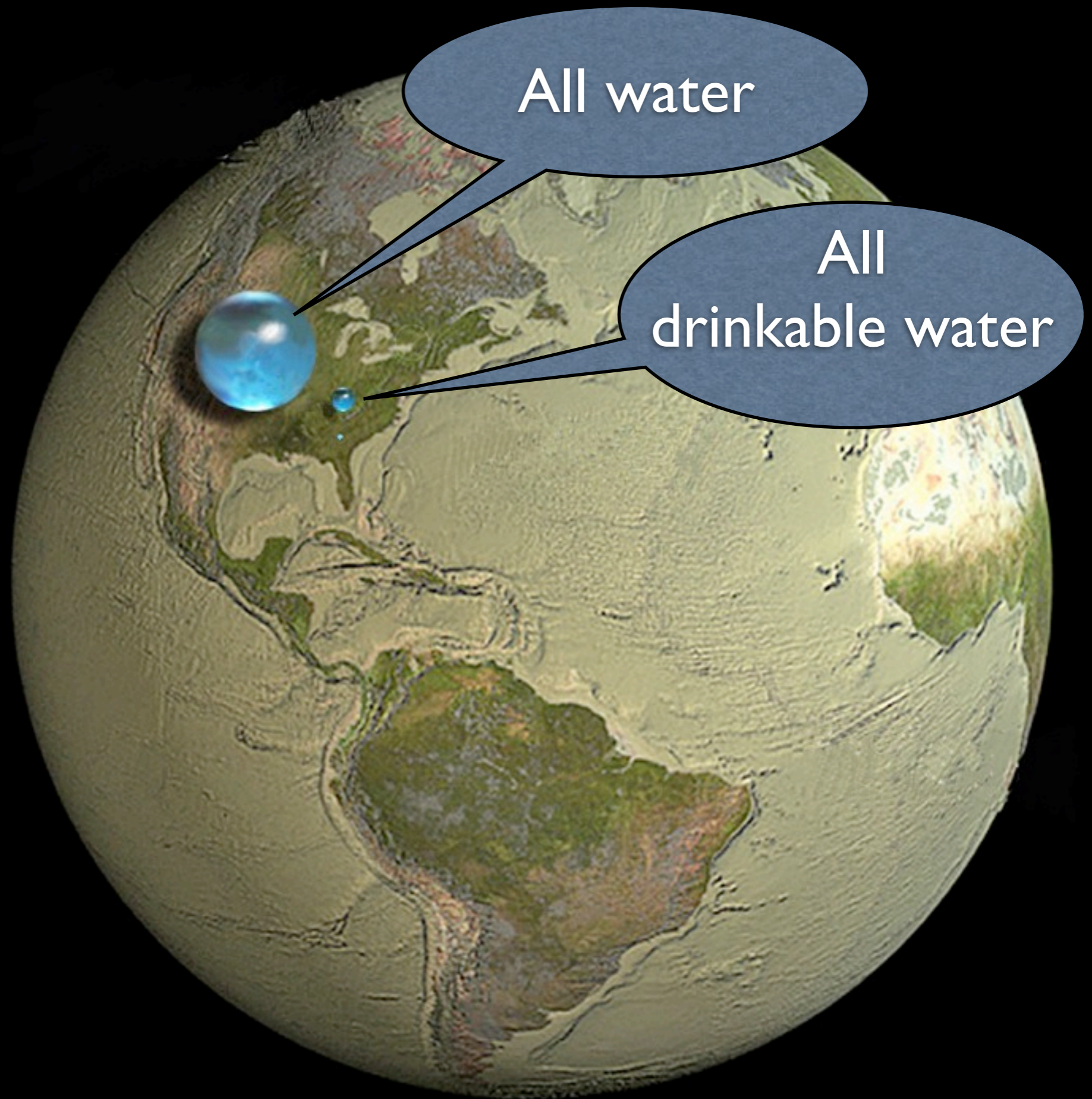
Source:
US Geological Survey

**How
much
is
that?**



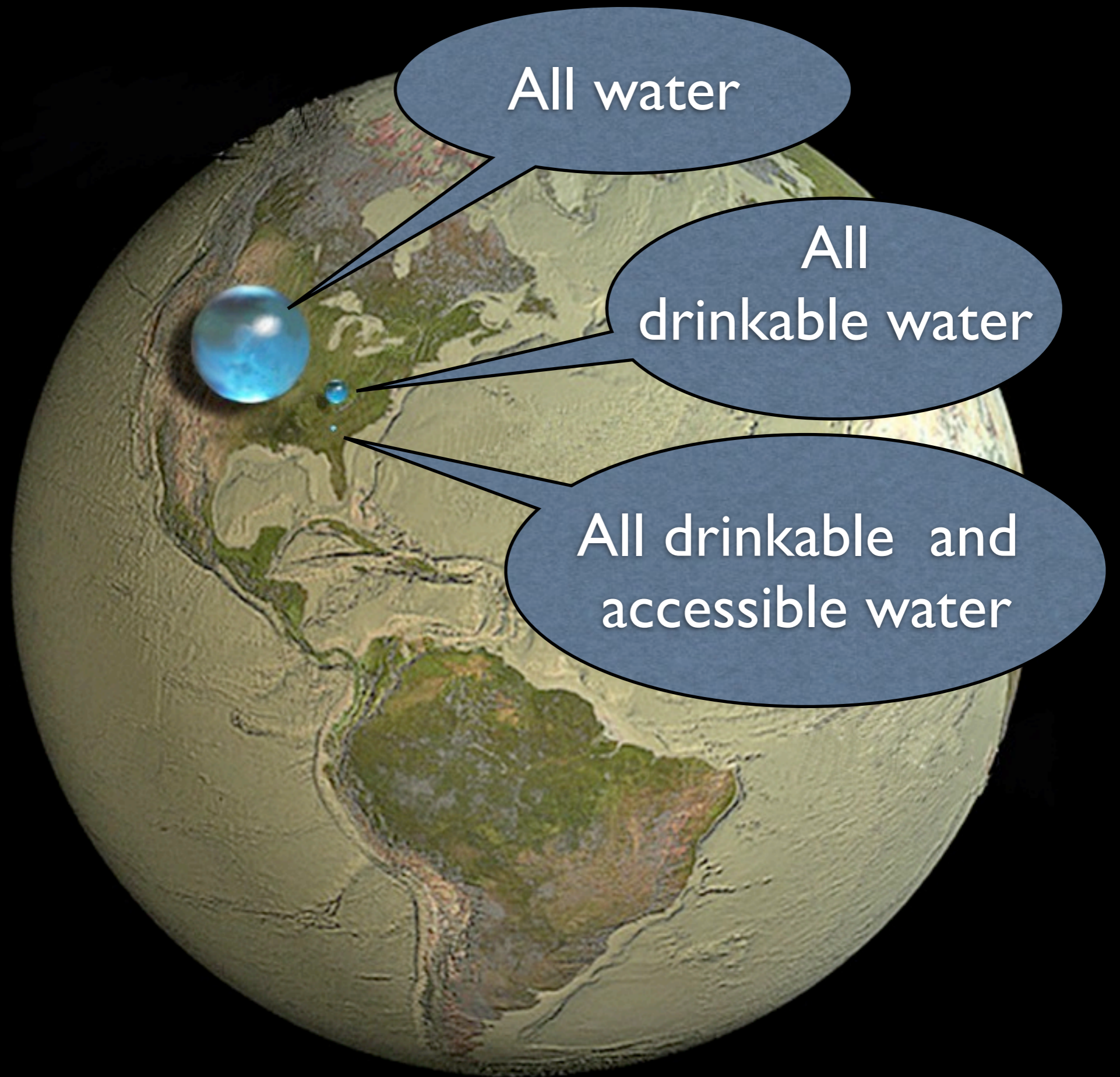
Source:
US Geological Survey

How much is that?



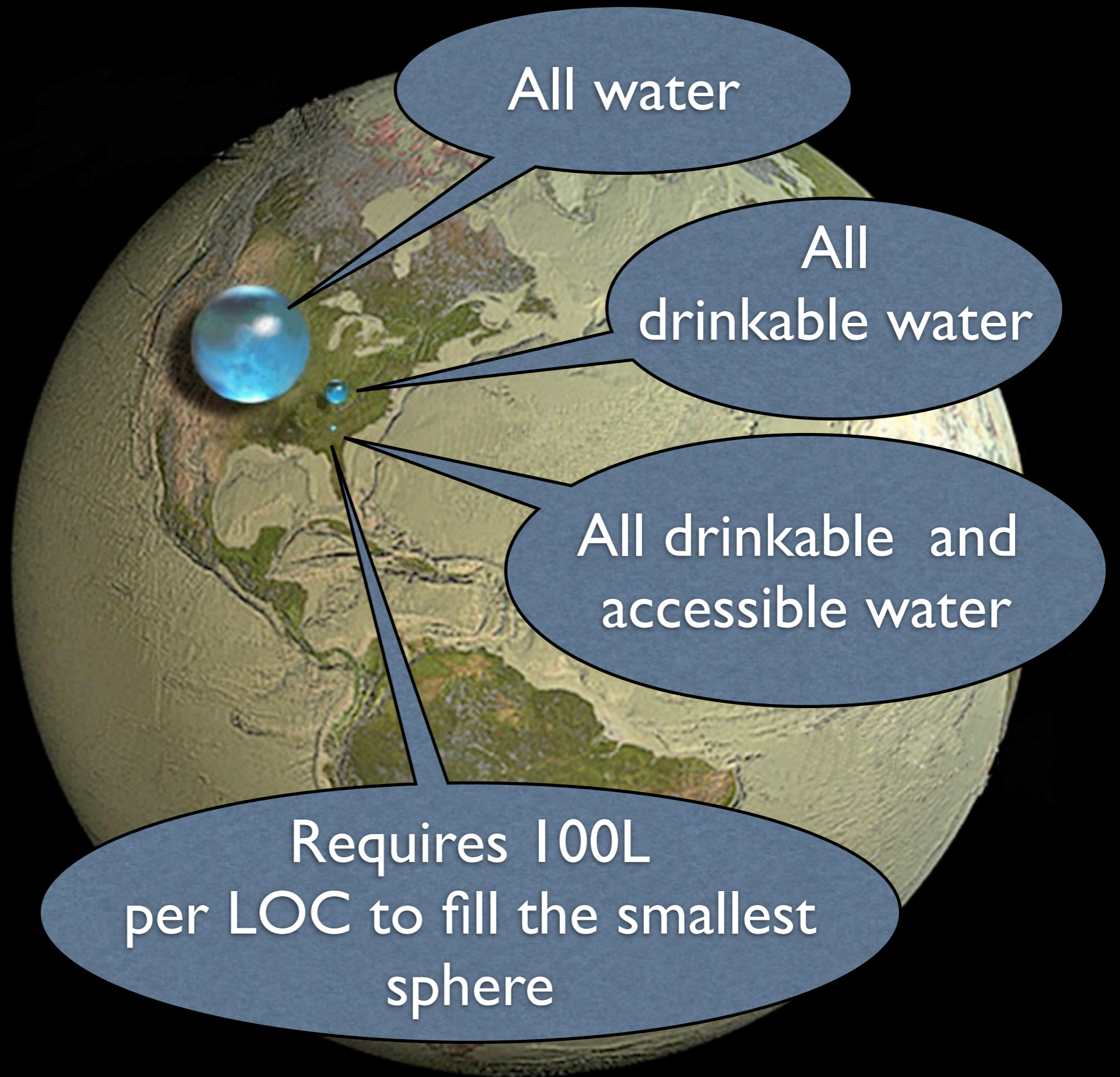
Source:
US Geological Survey

How much is that?



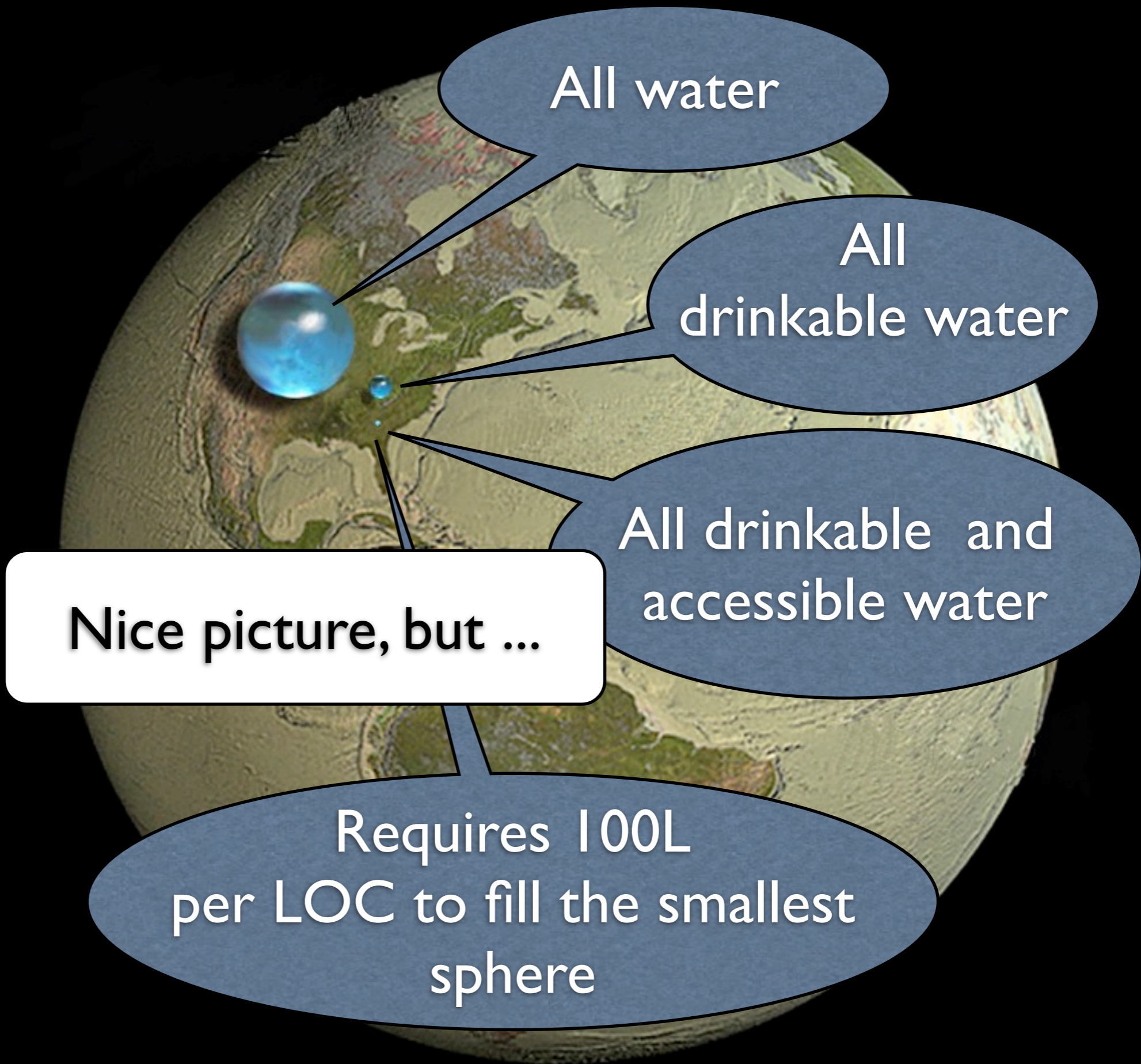
Source:
US Geological Survey

How much is that?



Source:
US Geological Survey

How much is that?



Source:
US Geological Survey

How much is that?

How much is that?

Print 1 billion lines in point size 10

How much is that?

Print 1 billion lines in point size 10



How much is that?

Print 1 billion lines in point size 10

0.8 times distance to the moon



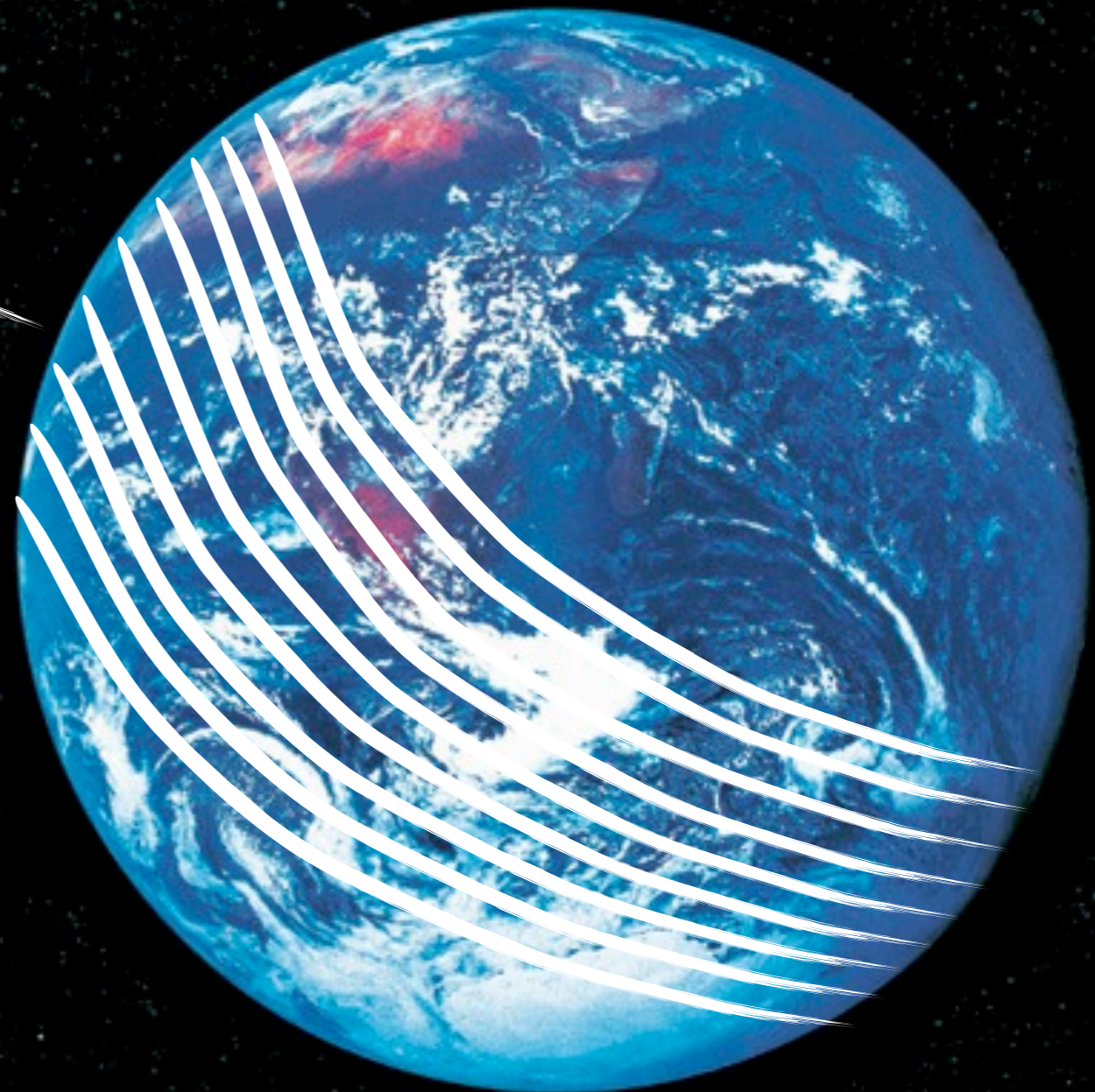
How much is that?

Print 1 billion lines in point size 10

0.8 times distance to the moon



Wrap Planet Earth 8 times



How can we
understand and
manage all that
software?



software;

Software Engineering

How can we understand and manage all that software?



software;

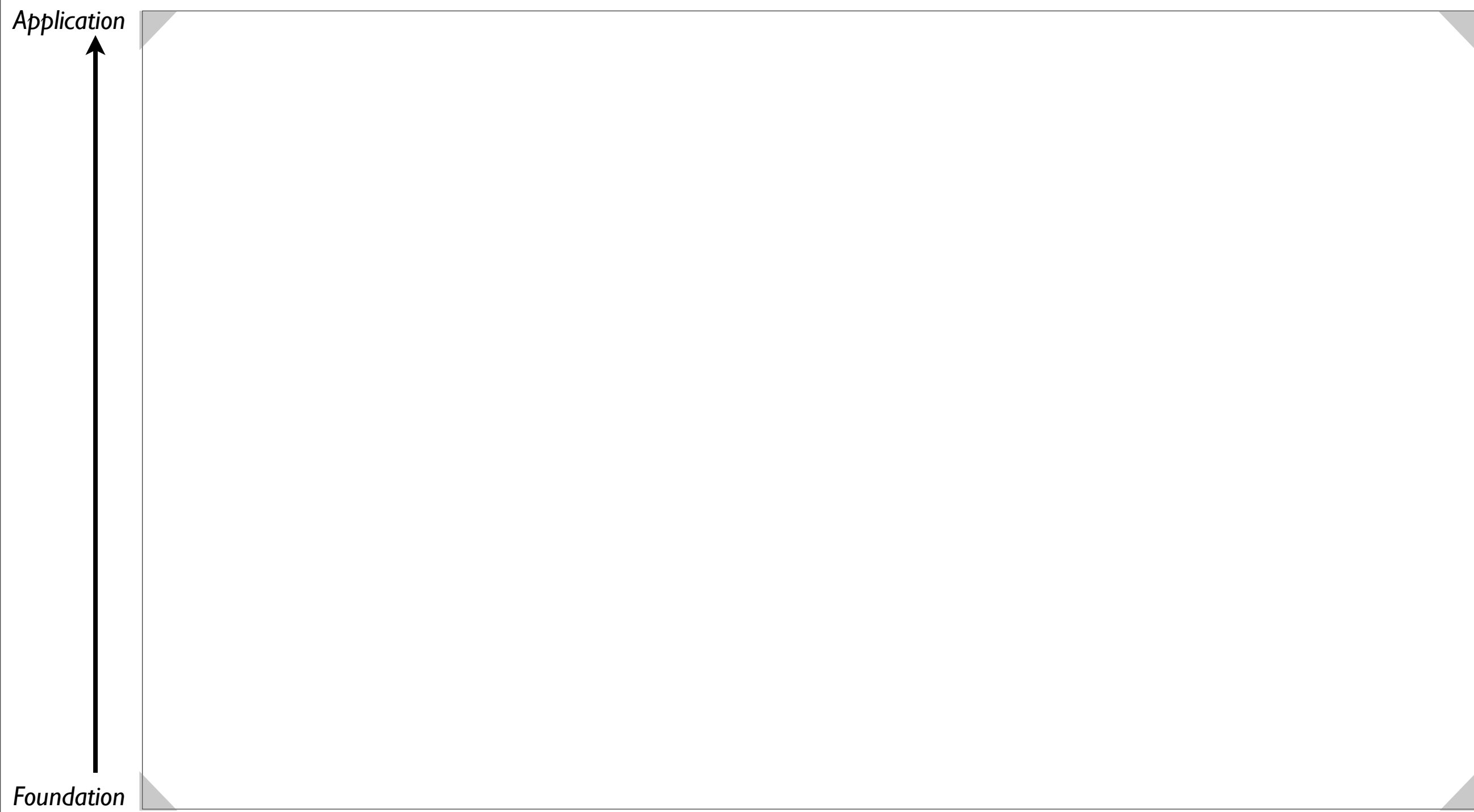


Word cloud of all papers at International Conference on Software Engineering (ICSE) 2012

Credit: Adrian Kuhn

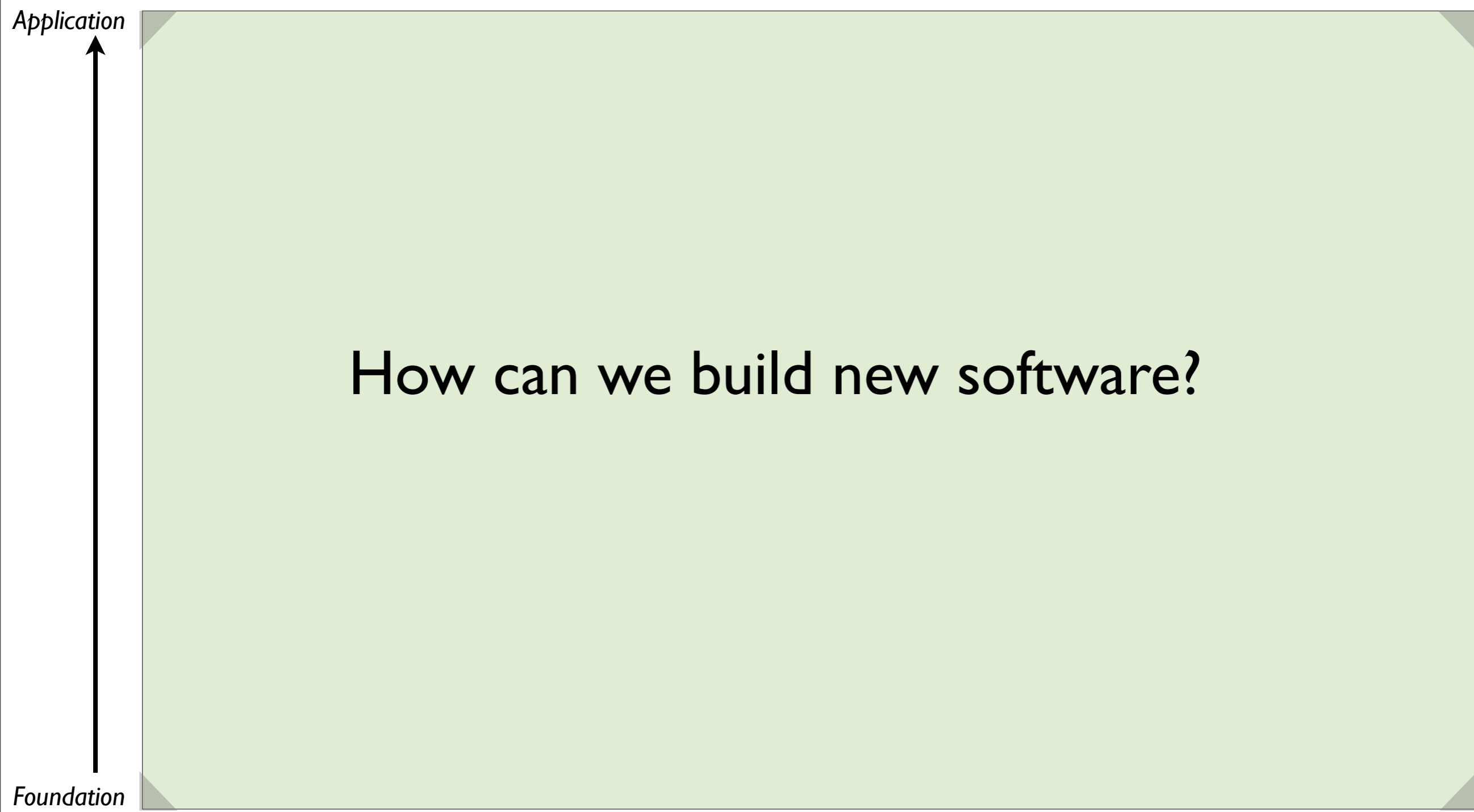
Software Engineering

Different Perspectives



Software Engineering

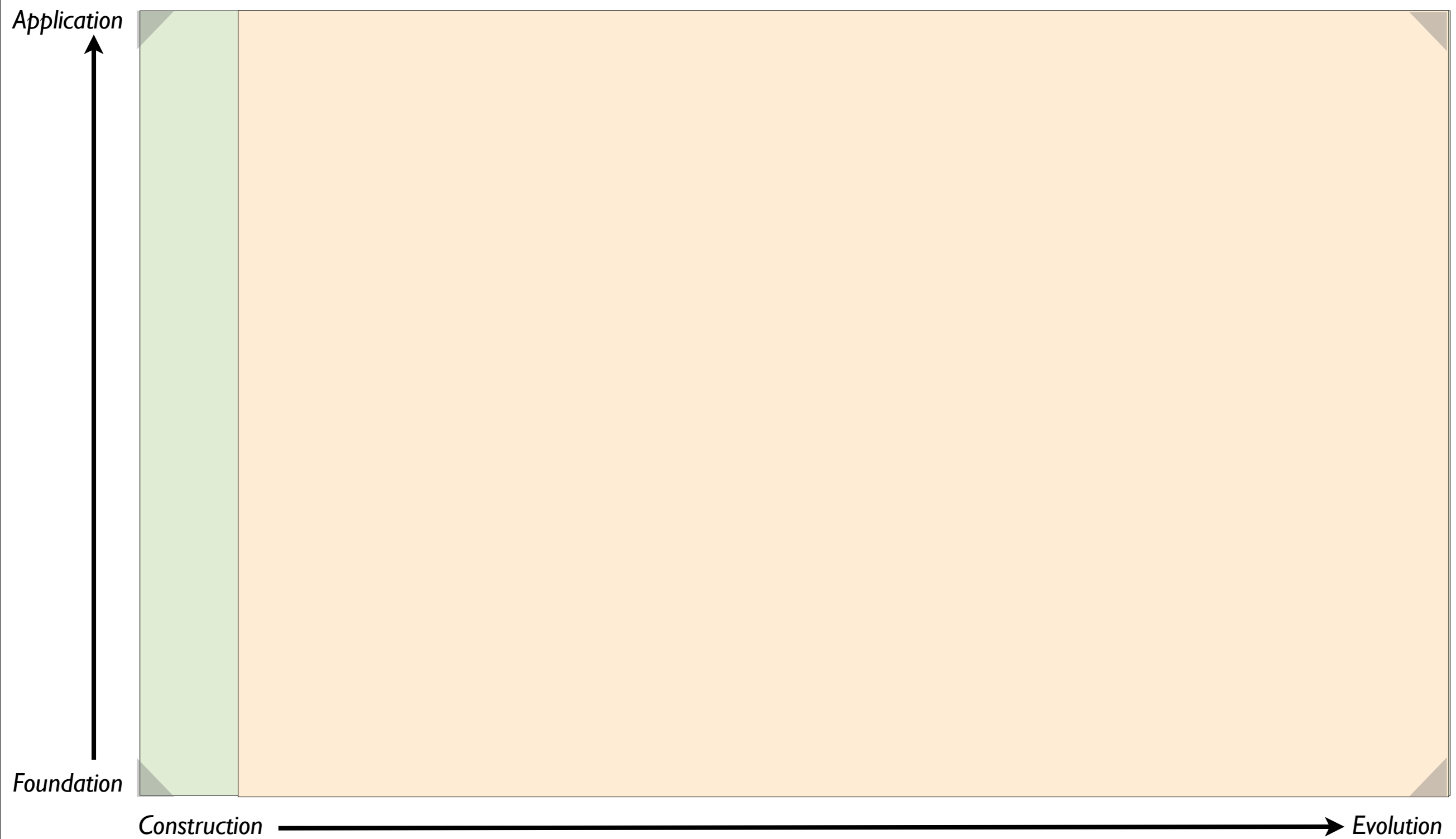
Traditional Academic View



How can we build new software?

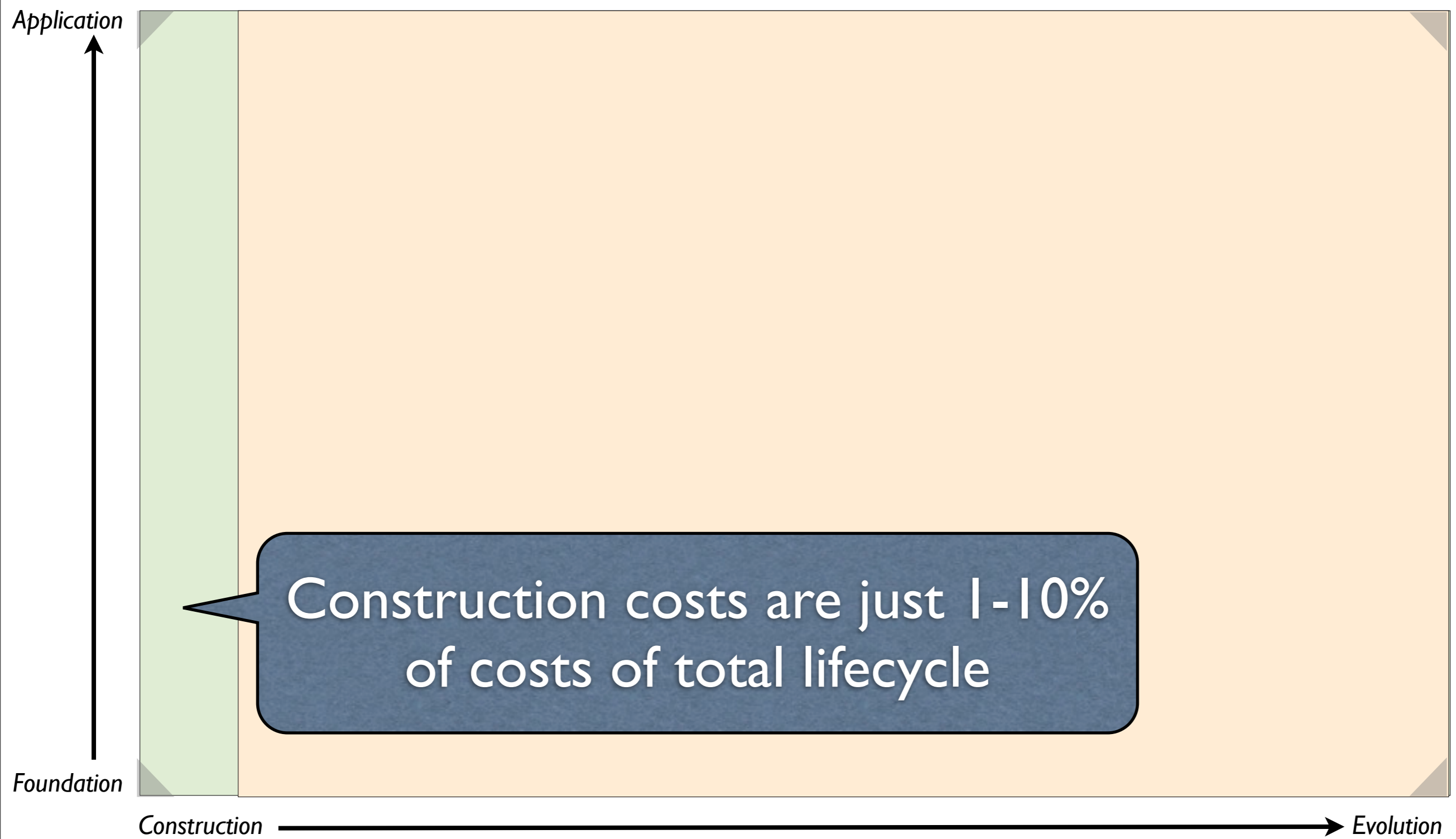
Software Engineering

Realistic View



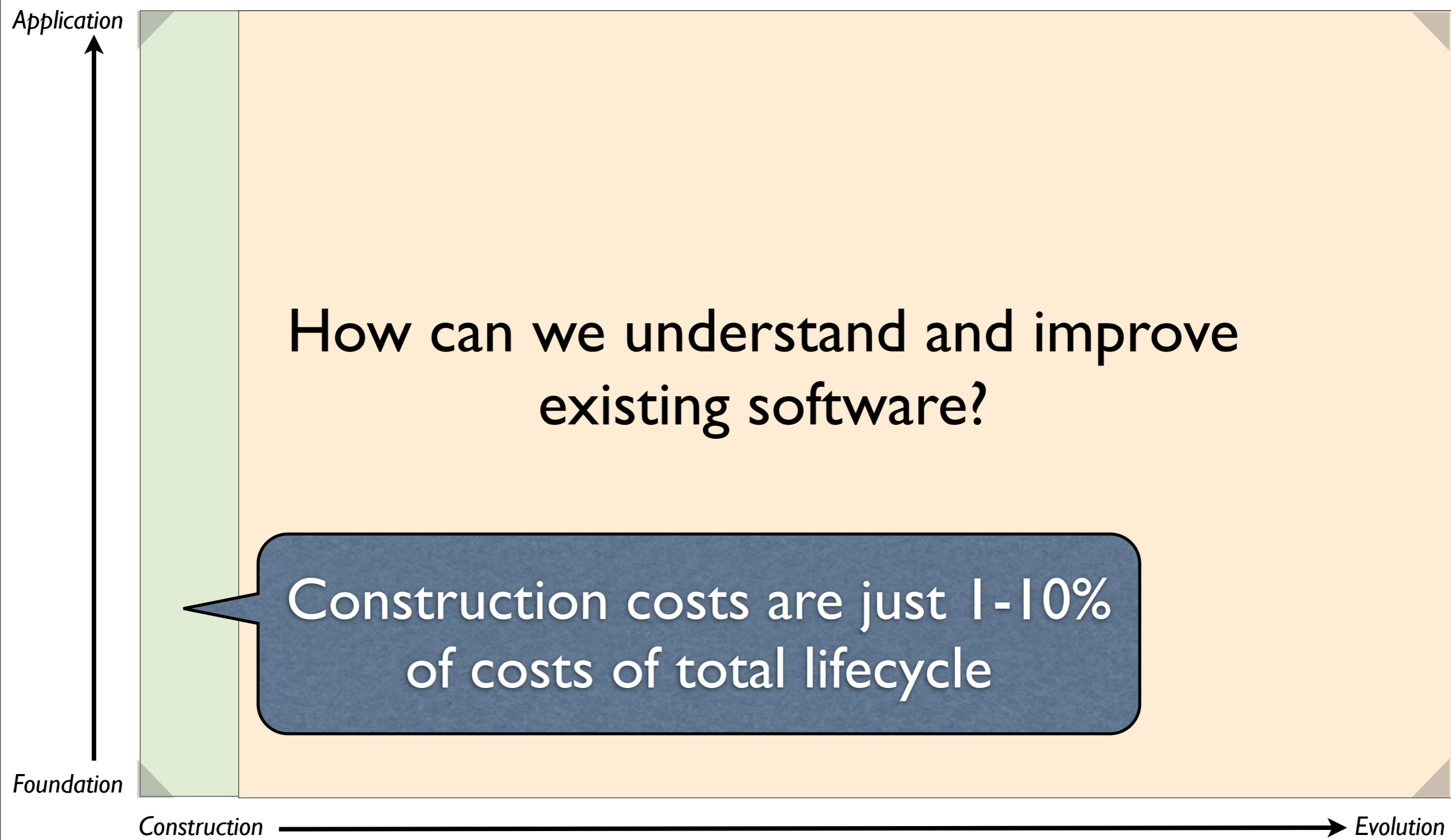
Software Engineering

Realistic View



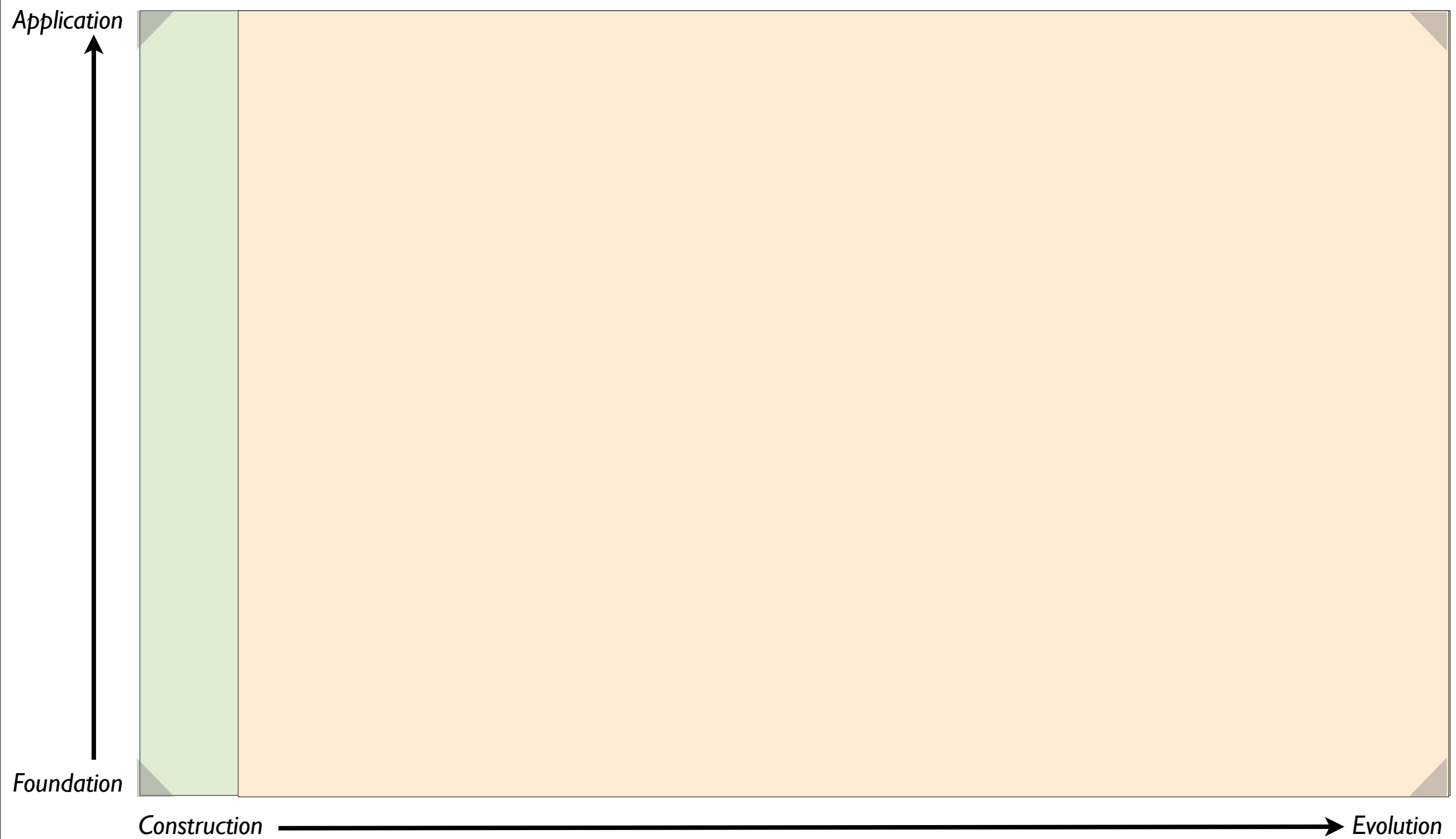
Software Engineering

Realistic View



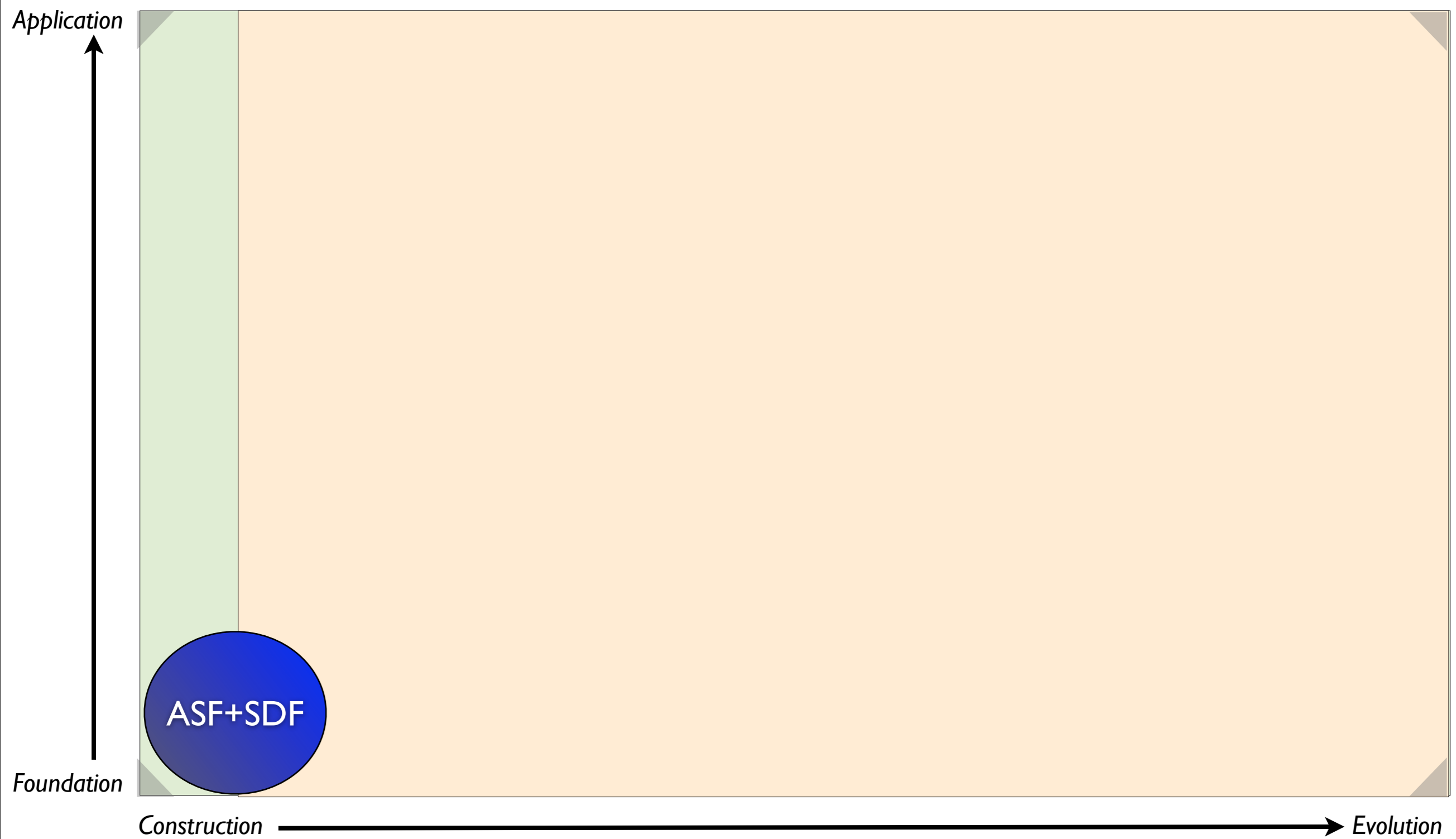
Software Engineering

Evolution of Research at CWI/SWAT



Software Engineering

Evolution of Research at CWI/SWAT



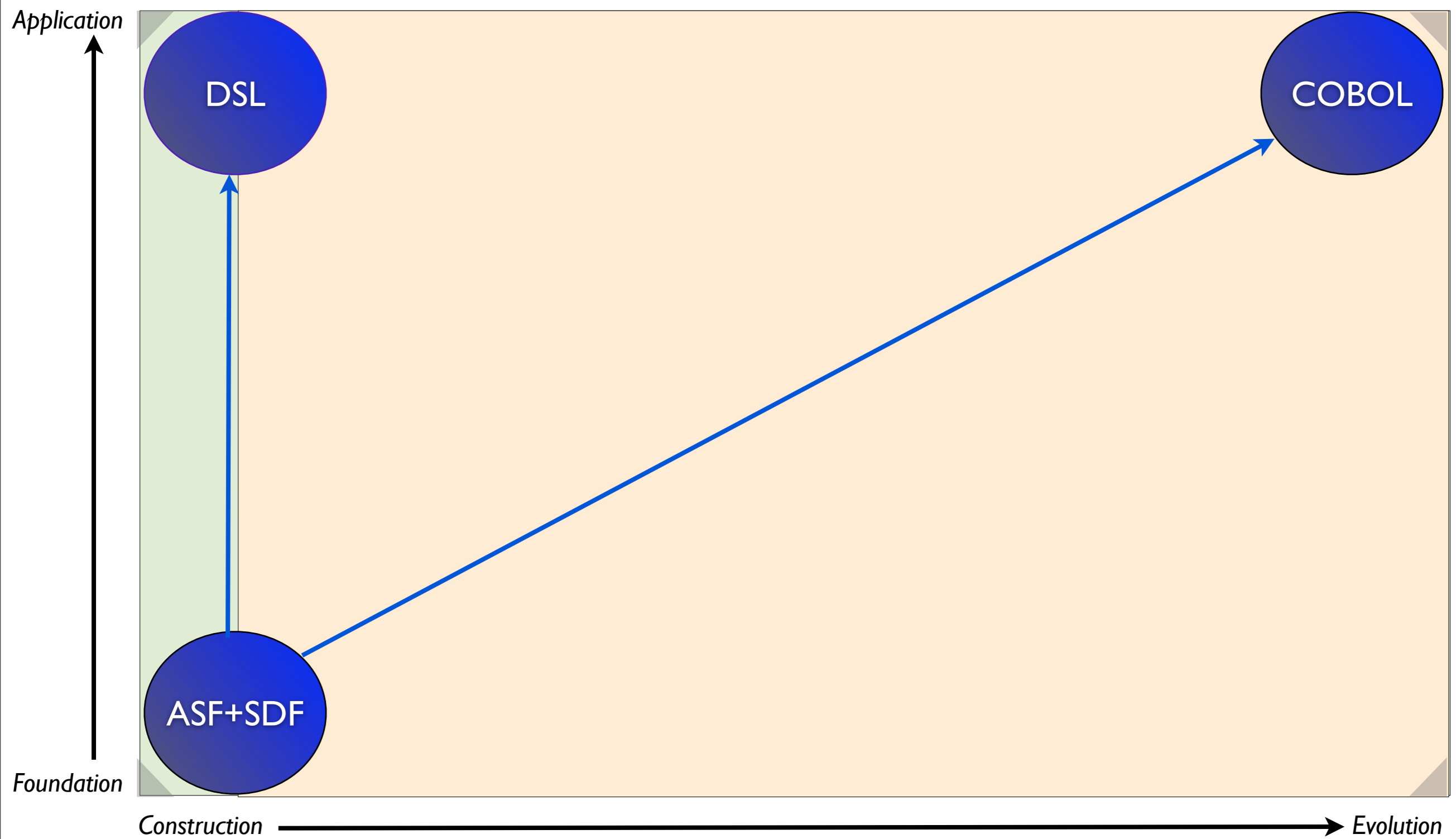
Software Engineering

Evolution of Research at CWI/SWAT



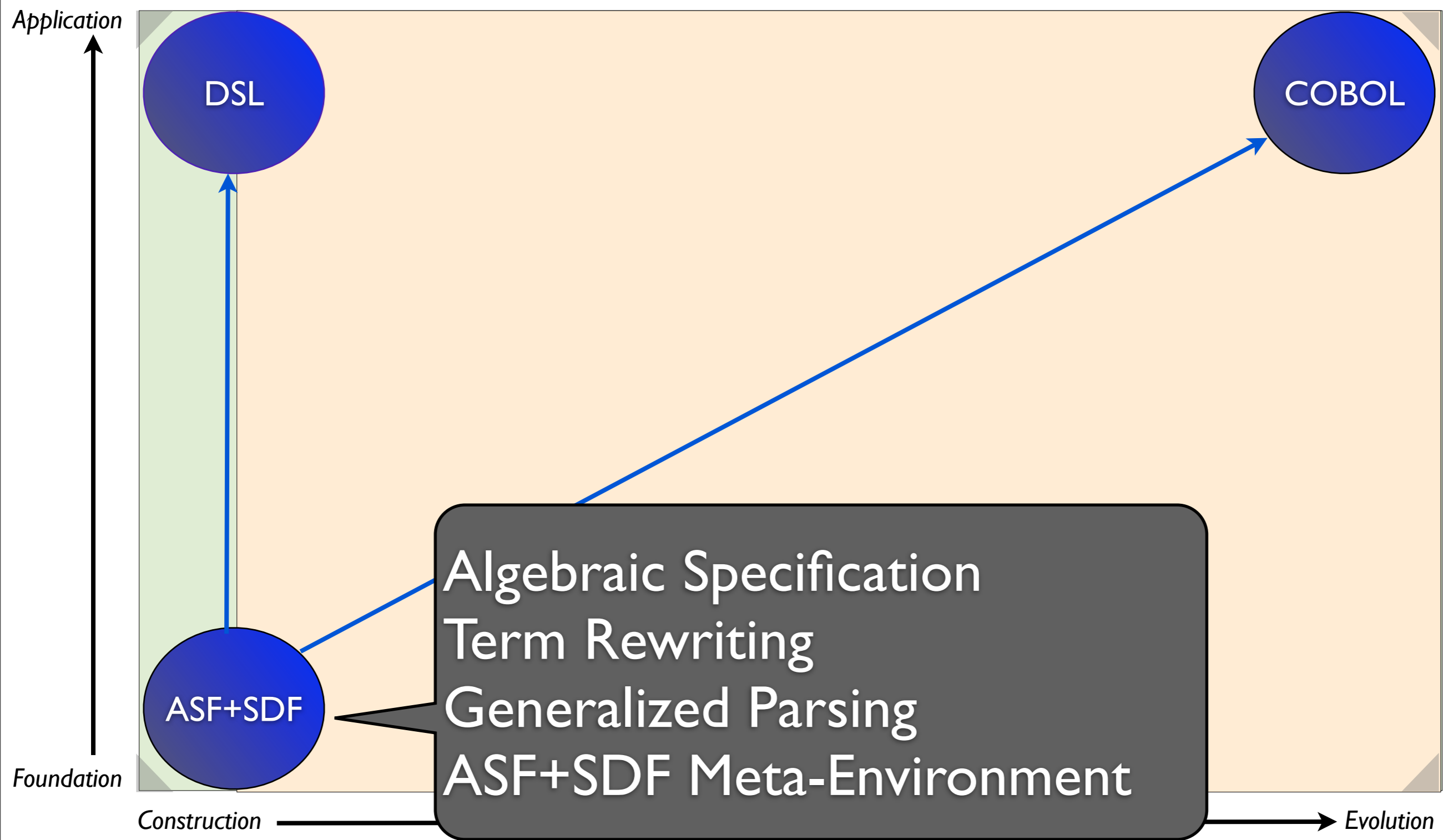
Software Engineering

Evolution of Research at CWI/SWAT



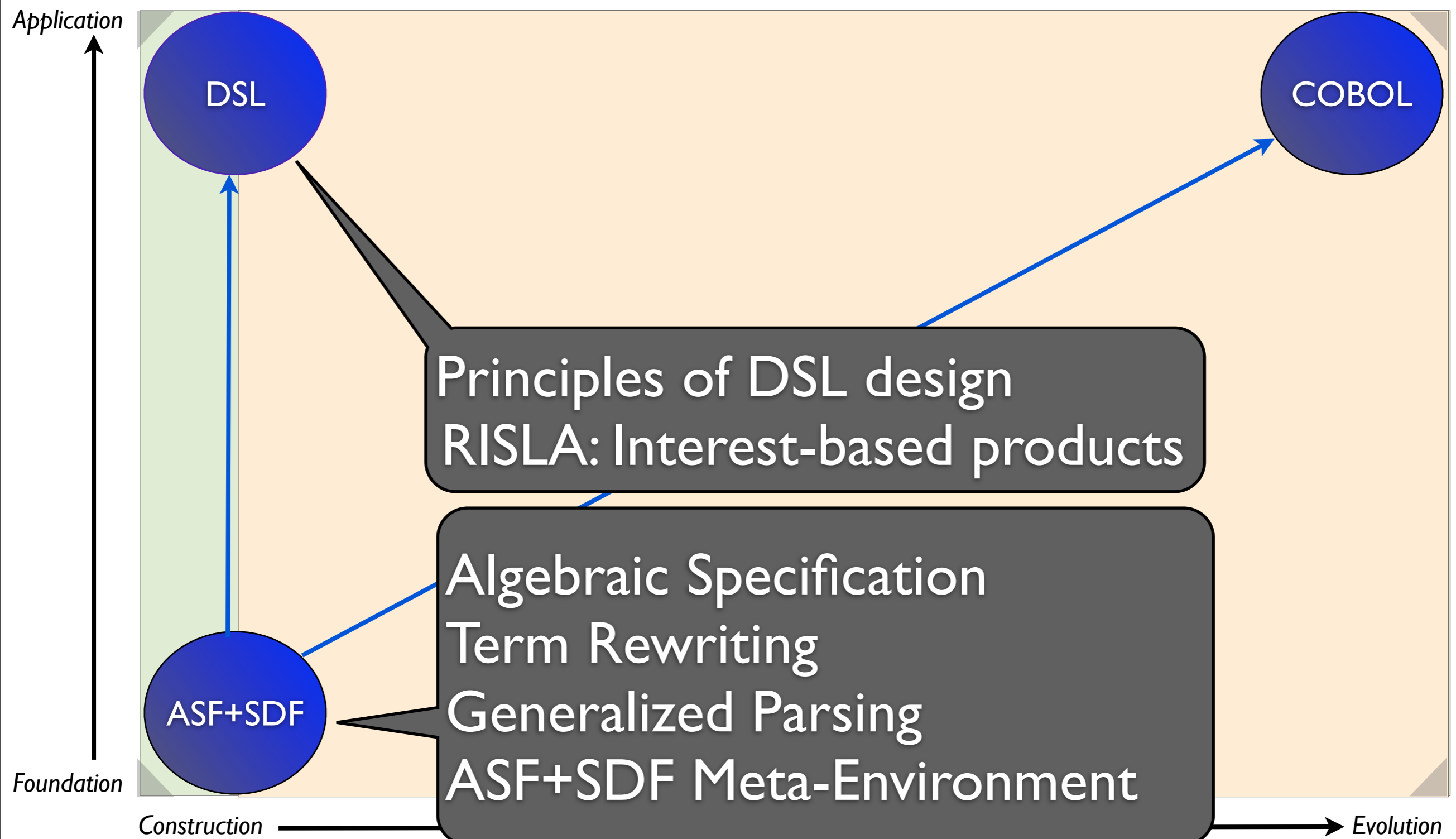
Software Engineering

Evolution of Research at CWI/SWAT



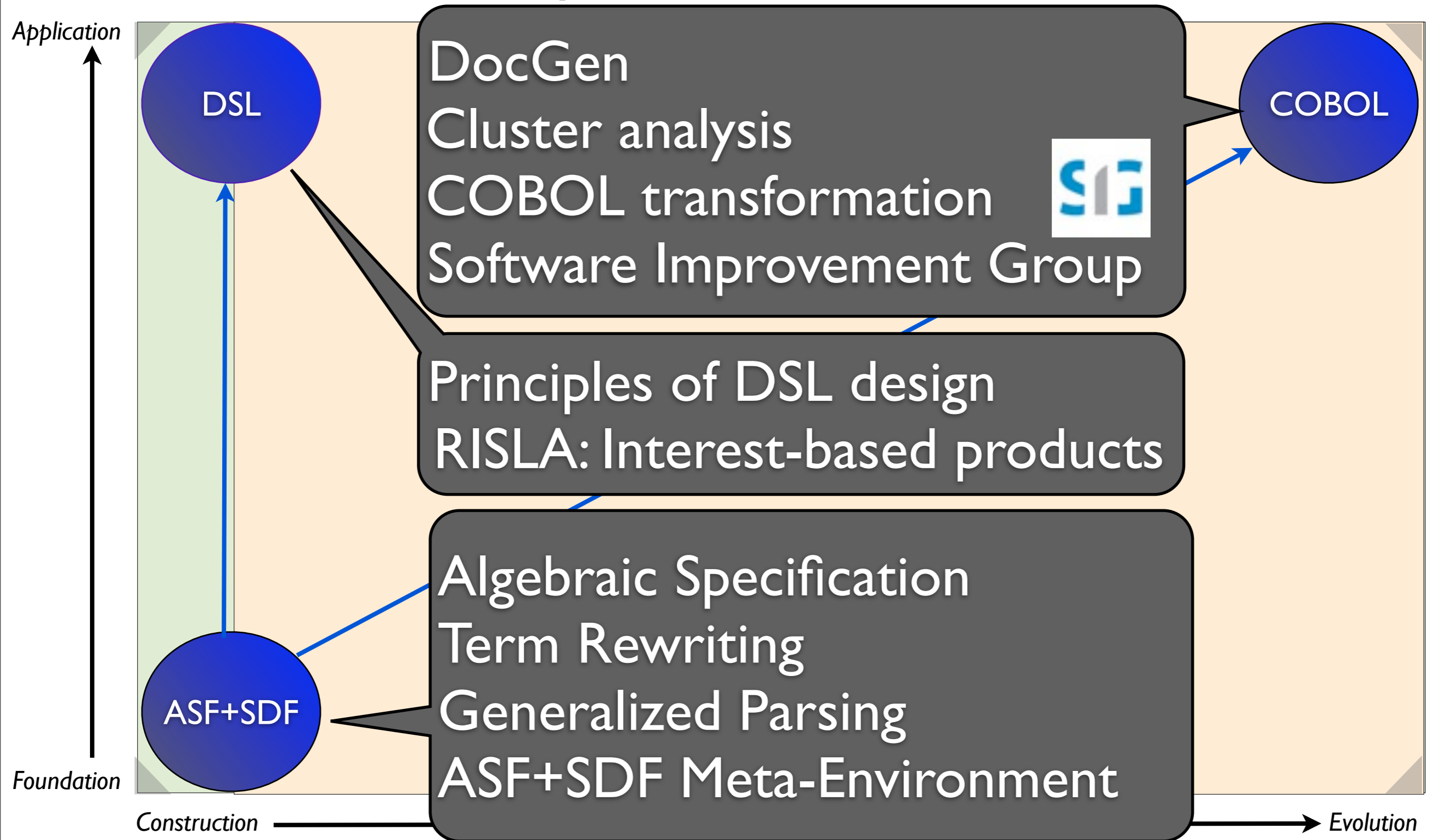
Software Engineering

Evolution of Research at CWI/SWAT



Software Engineering

Evolution of Research at CWI/SWAT



Triggers for Change

Technology	Usability	Application domains
New ideas did not fit in old approach	Students without formal background	Analysis & transformation of real languages
Enable many experiments and easy extension	Usable for real programmers	Repository mining
Connect to external libraries	IDE features for DSLs	New DSLs
Efficient built-in datatypes	Error reporting & recovery	Software Visualization

Current Goals

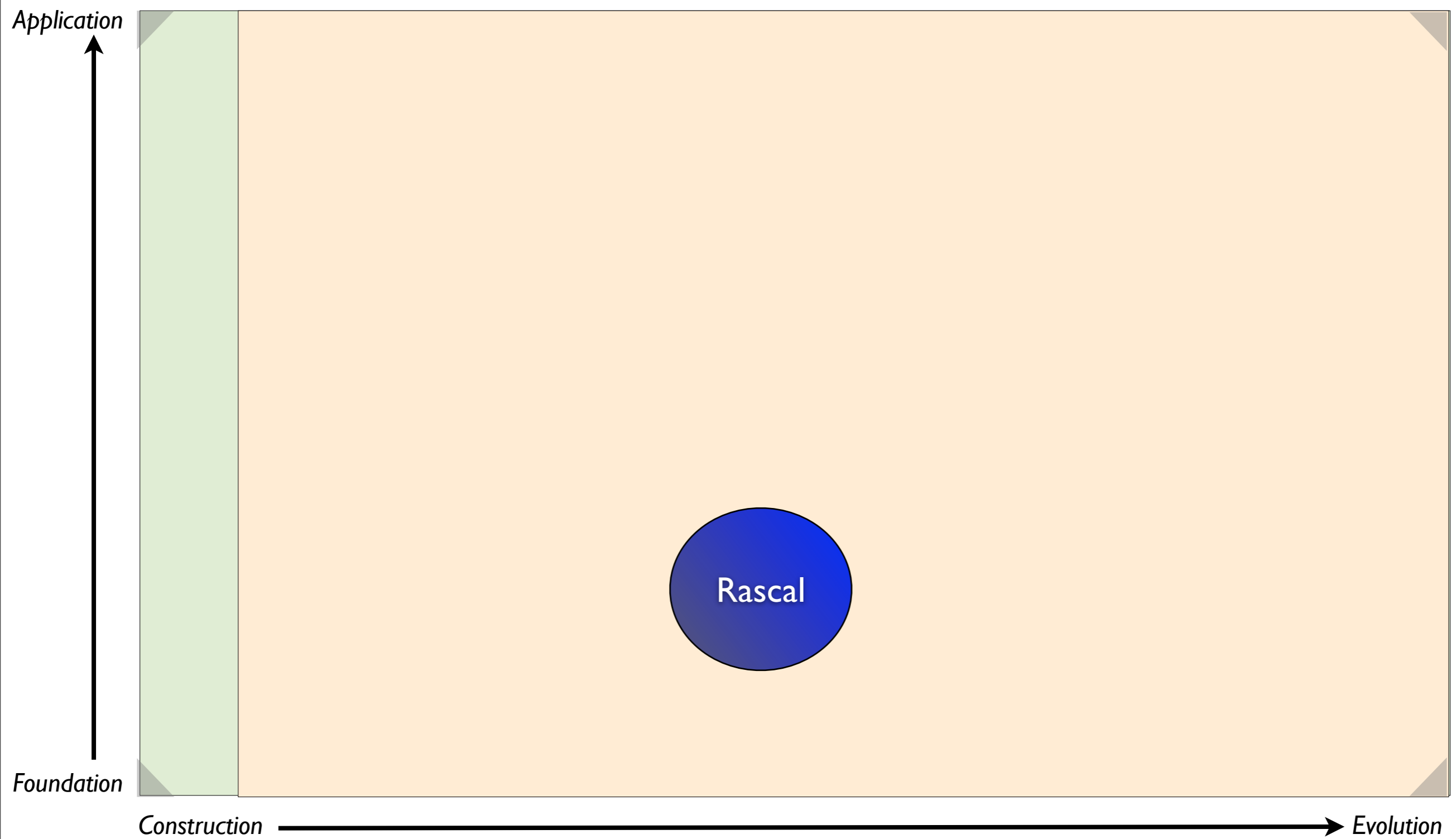
- “One-stop-shop” for
 - Meta-programming
 - Meta-data analysis
 - Visualization
- Lab infrastructure
- Transfer medium



<http://www.rascal-mpl.org>

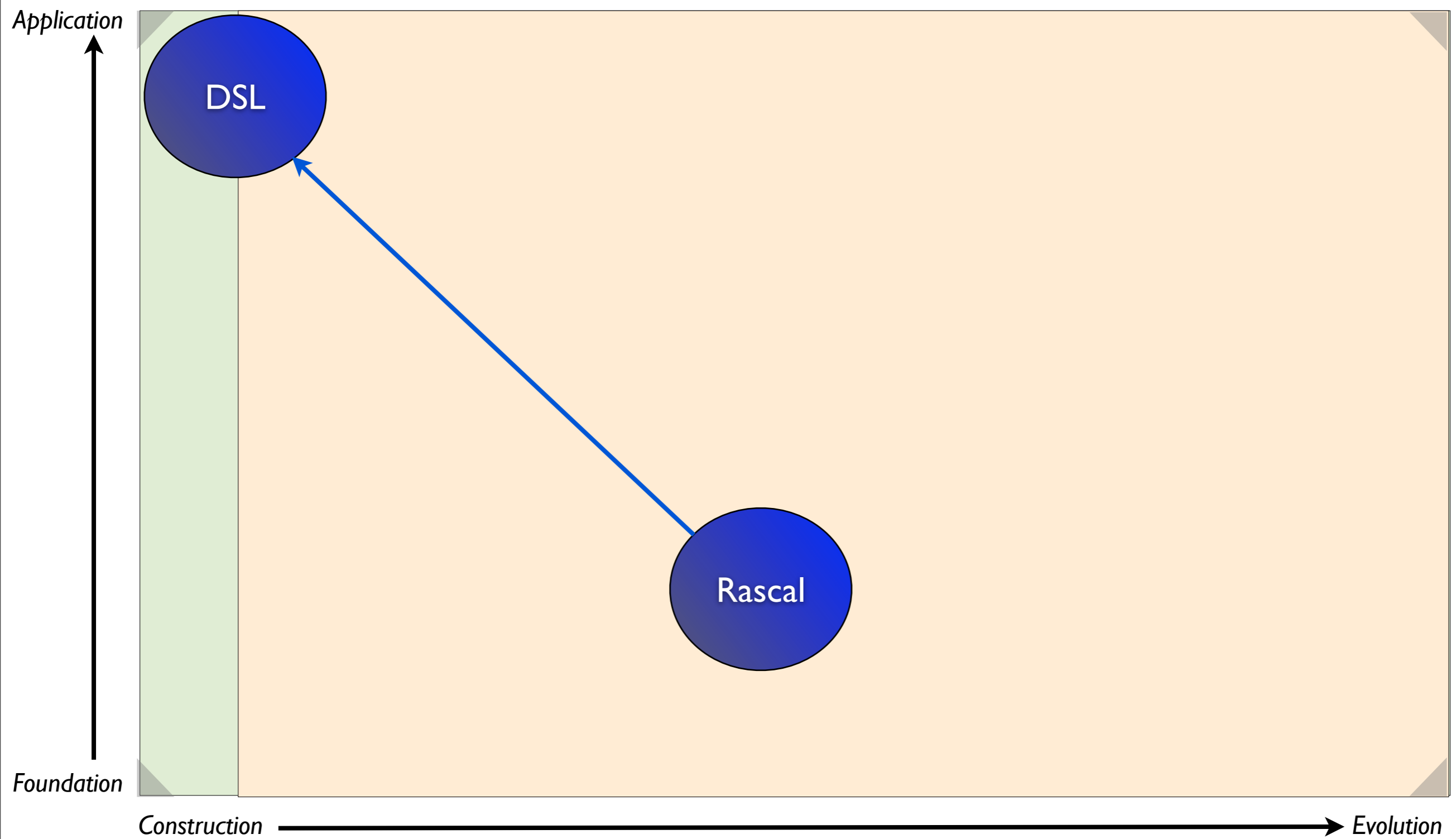
Software Engineering

Evolution of Research at CWI/SWAT



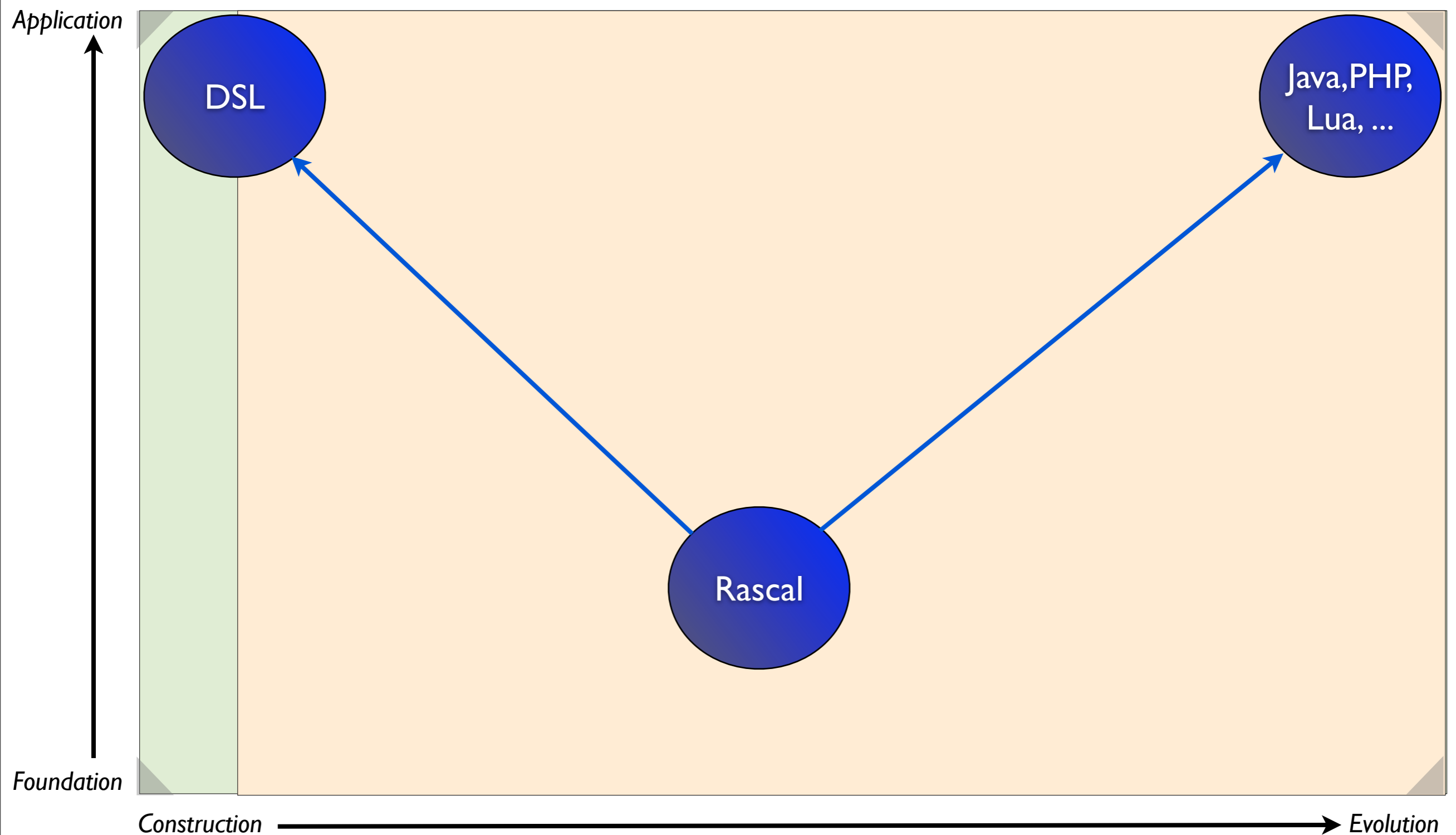
Software Engineering

Evolution of Research at CWI/SWAT



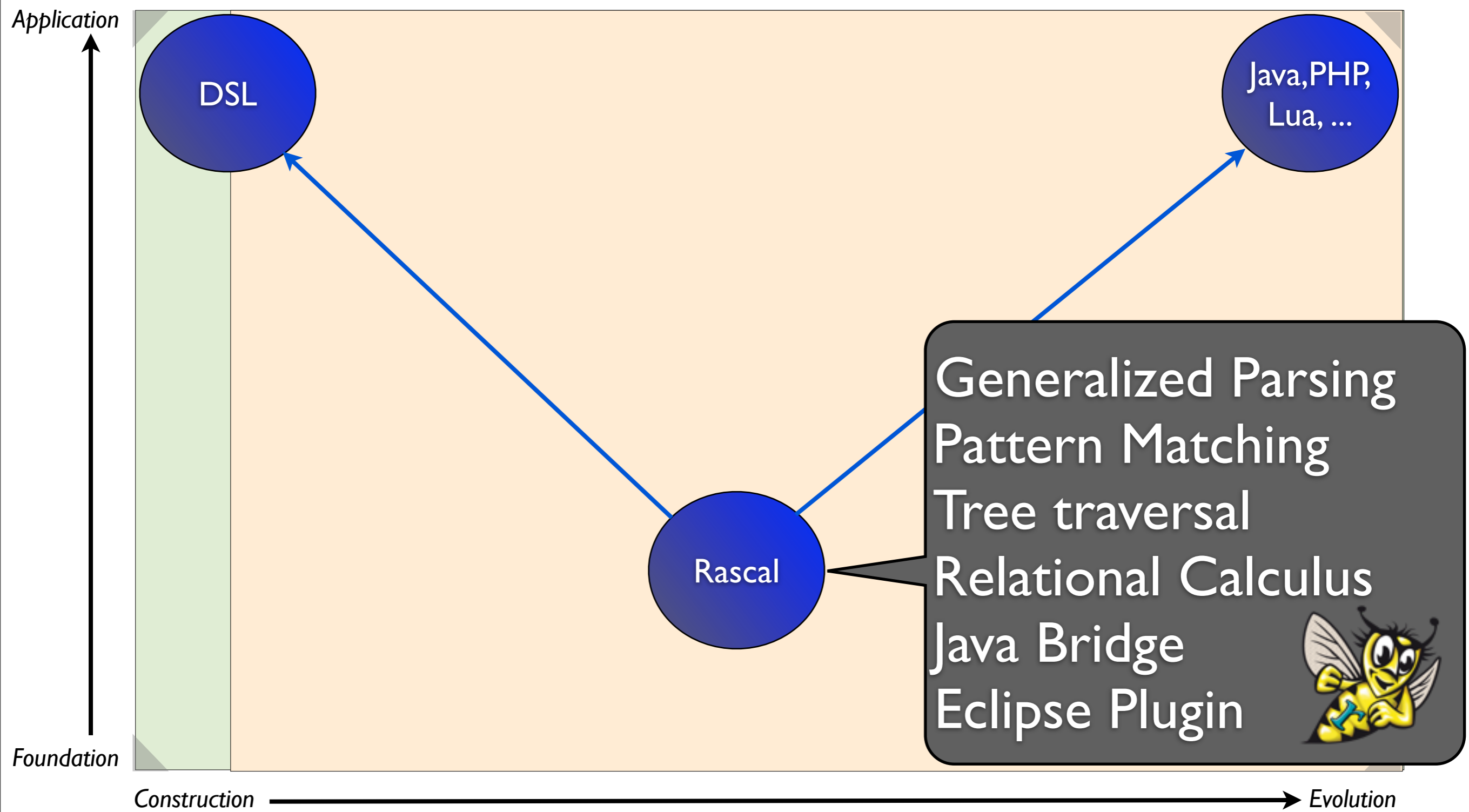
Software Engineering

Evolution of Research at CWI/SWAT



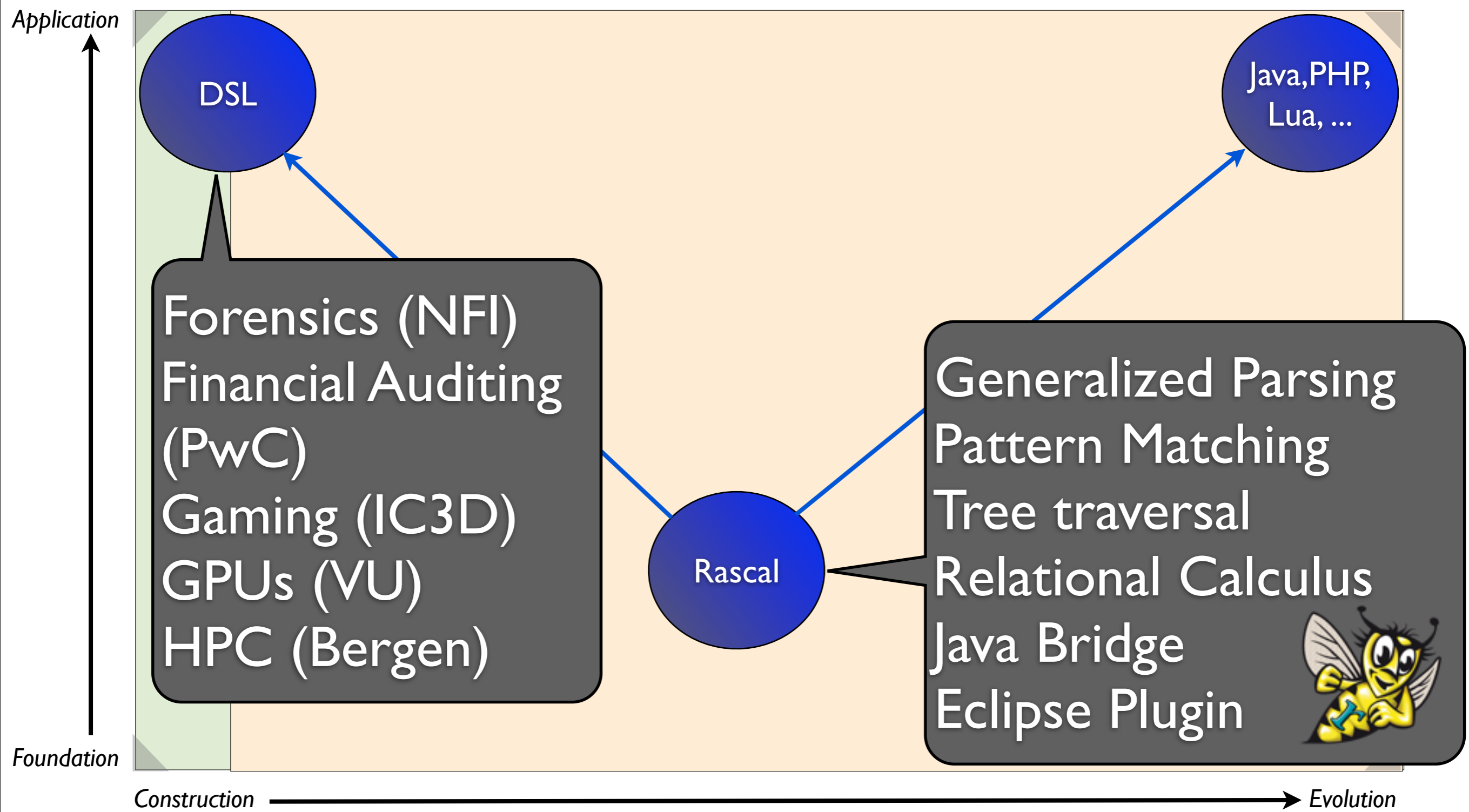
Software Engineering

Evolution of Research at CWI/SWAT



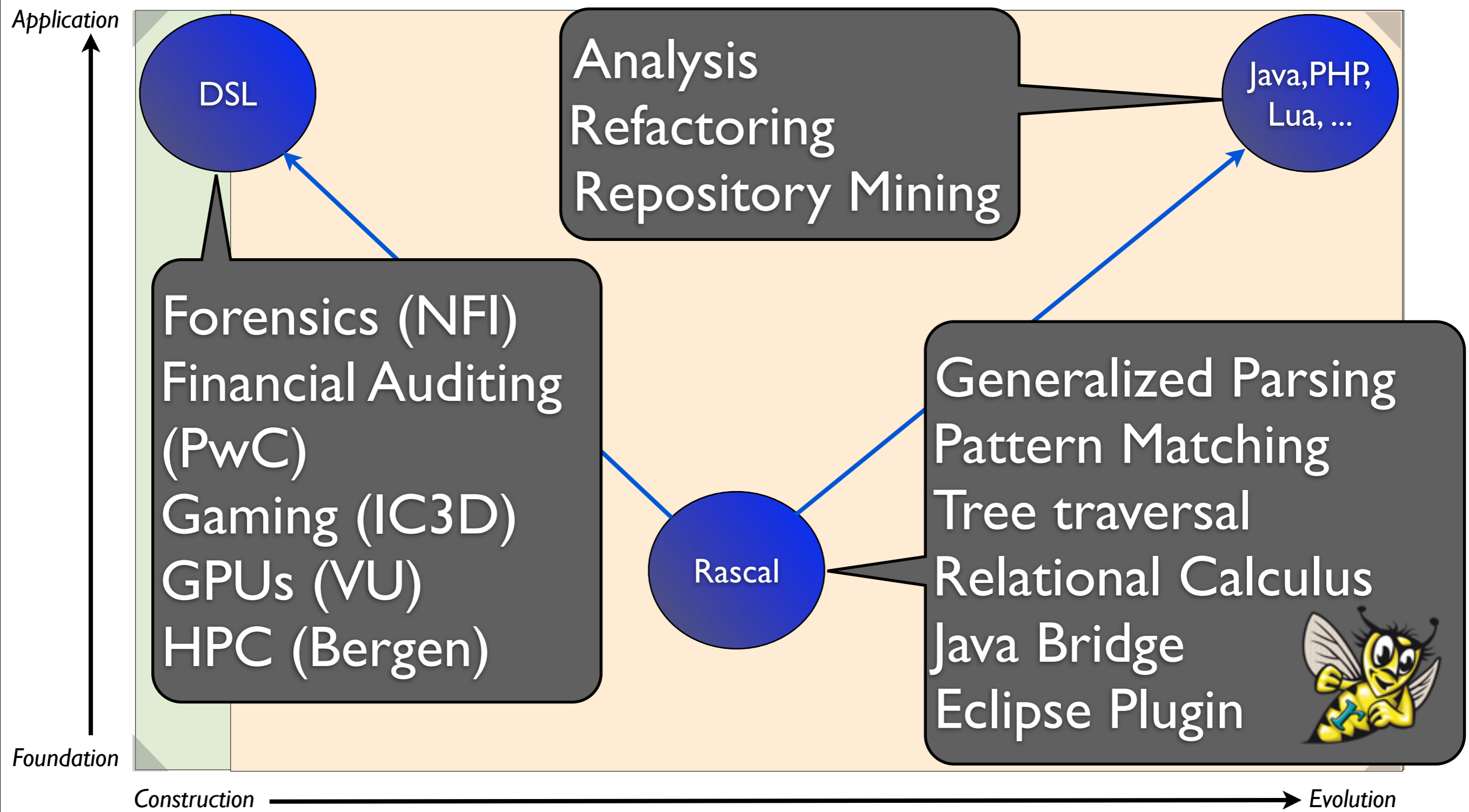
Software Engineering

Evolution of Research at CWI/SWAT



Software Engineering

Evolution of Research at CWI/SWAT



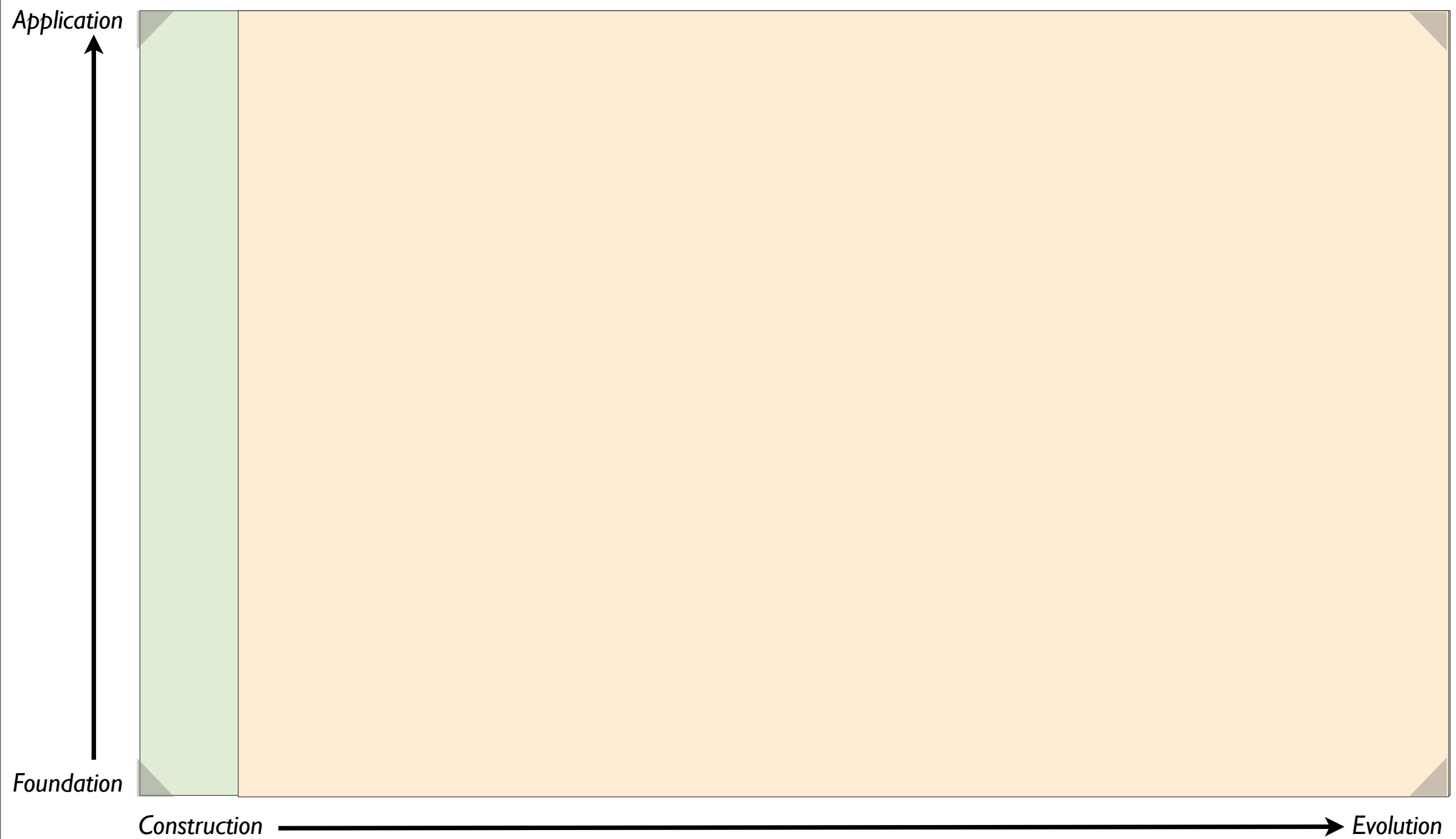
Rascal Results so far

Technology	Usability	Application domains
Types, locations, resources, visit, pattern-directed invocation, ...	Layered design: learn on demand	Java, PHP, Lua analysis & refactoring
New GLL parsing	Eclipse IDE features	Repository mining
Native support for visualization	REPL	Metrics
Libraries Math, Statistics, JDBC, SVN, ...	Interactive Tutor	Various DSLs

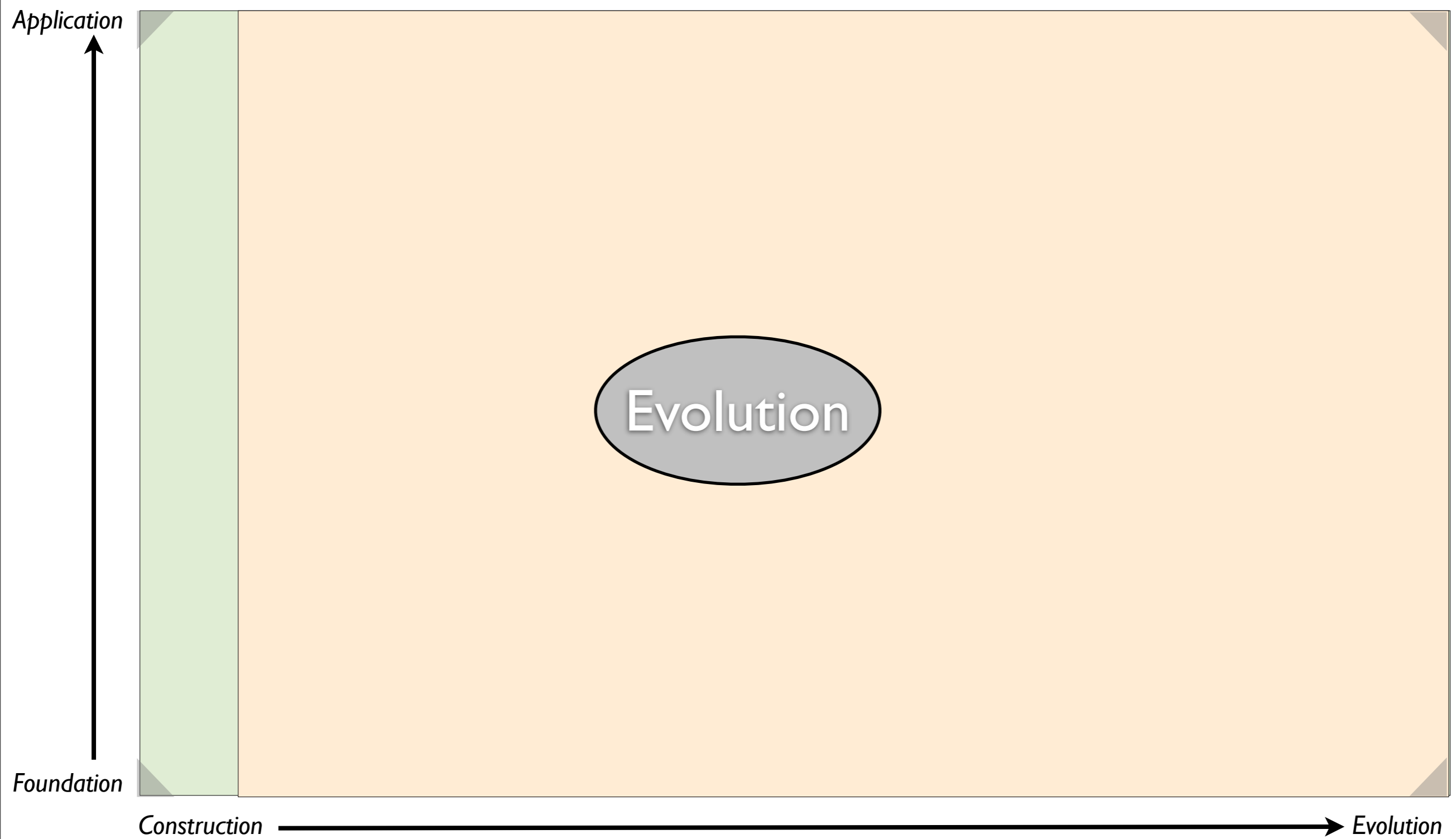
*These considerations
motivate today's topic:
Understanding Software*

Understanding Software

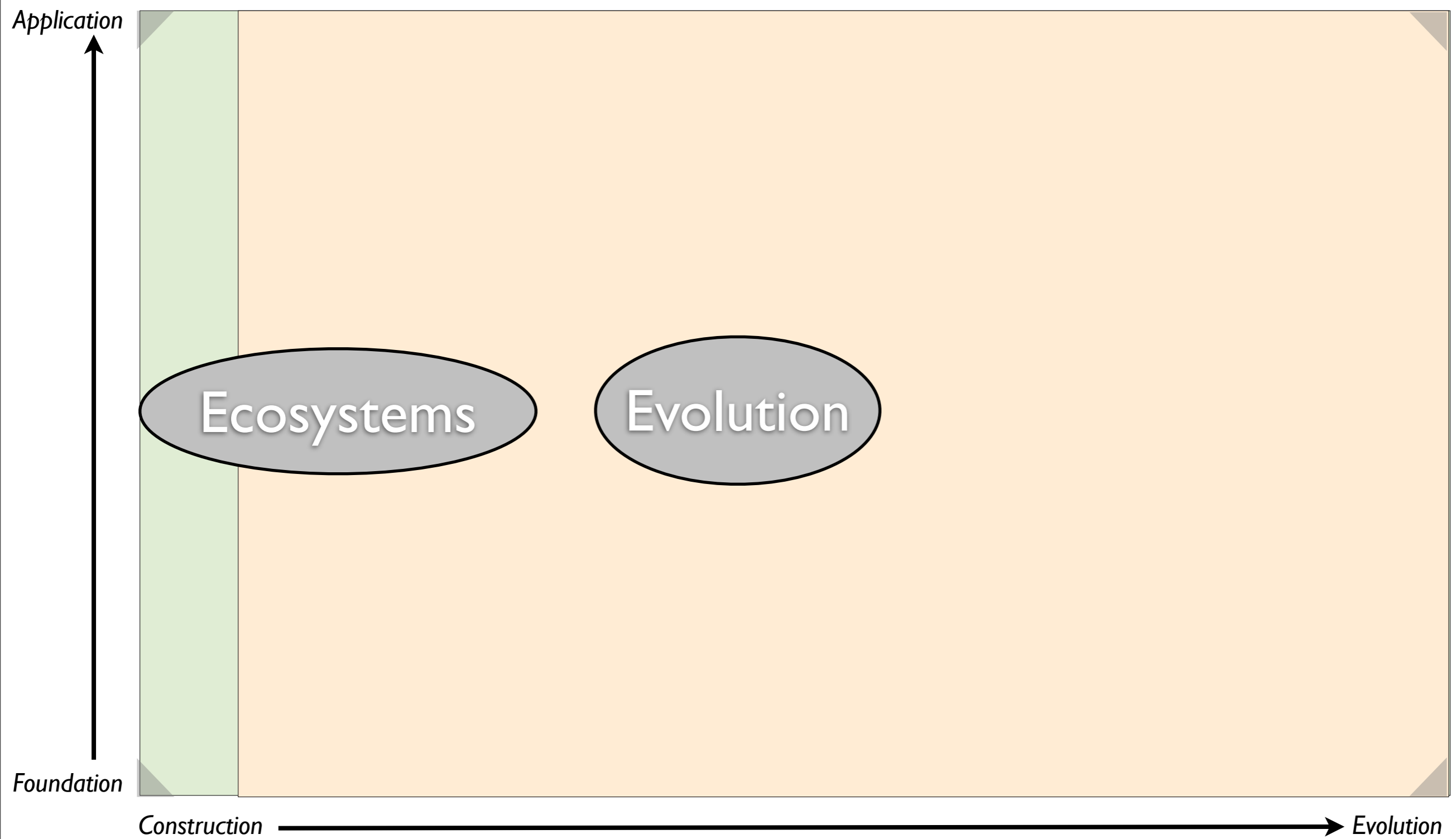
Understanding Software



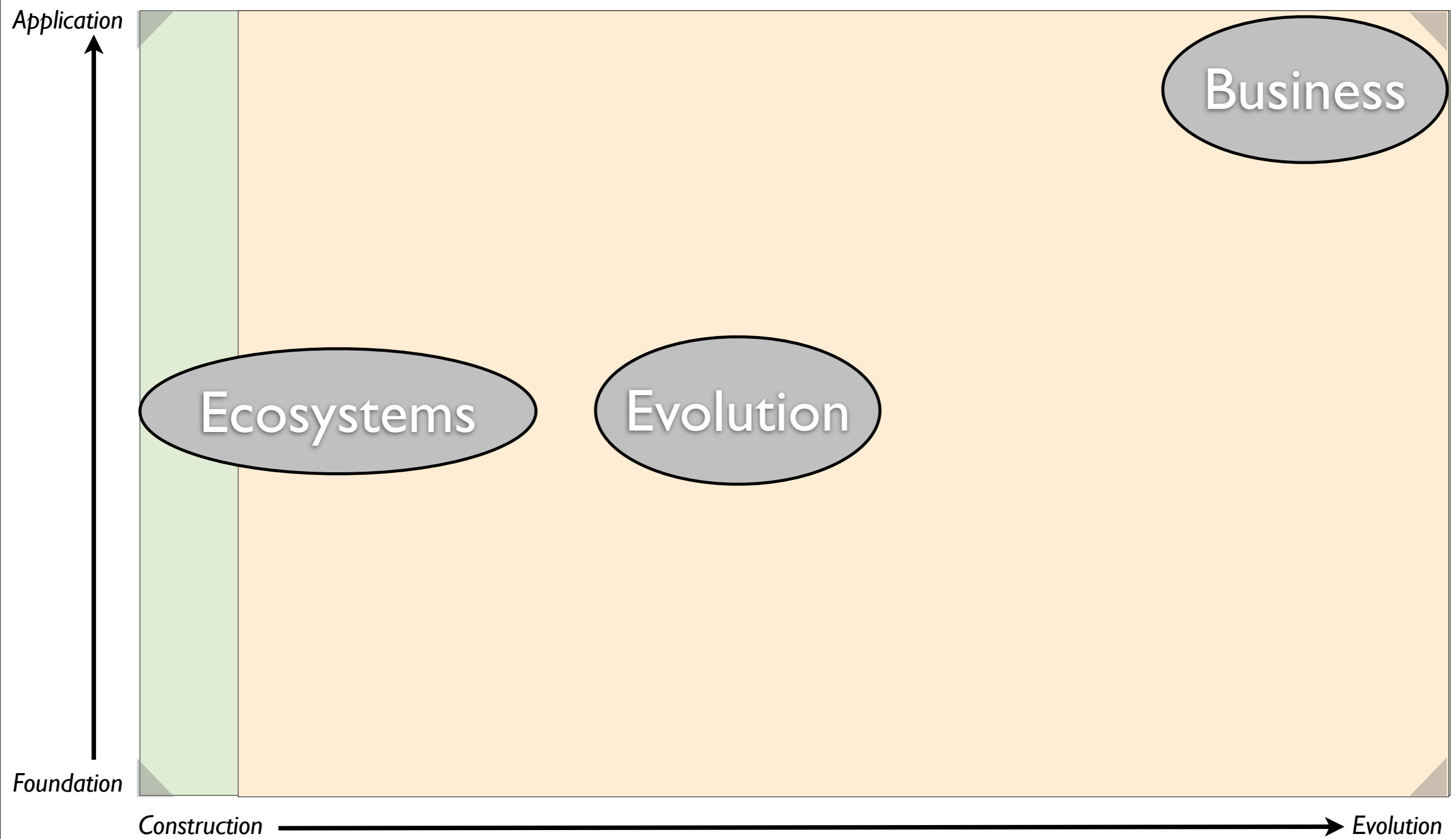
Understanding Software



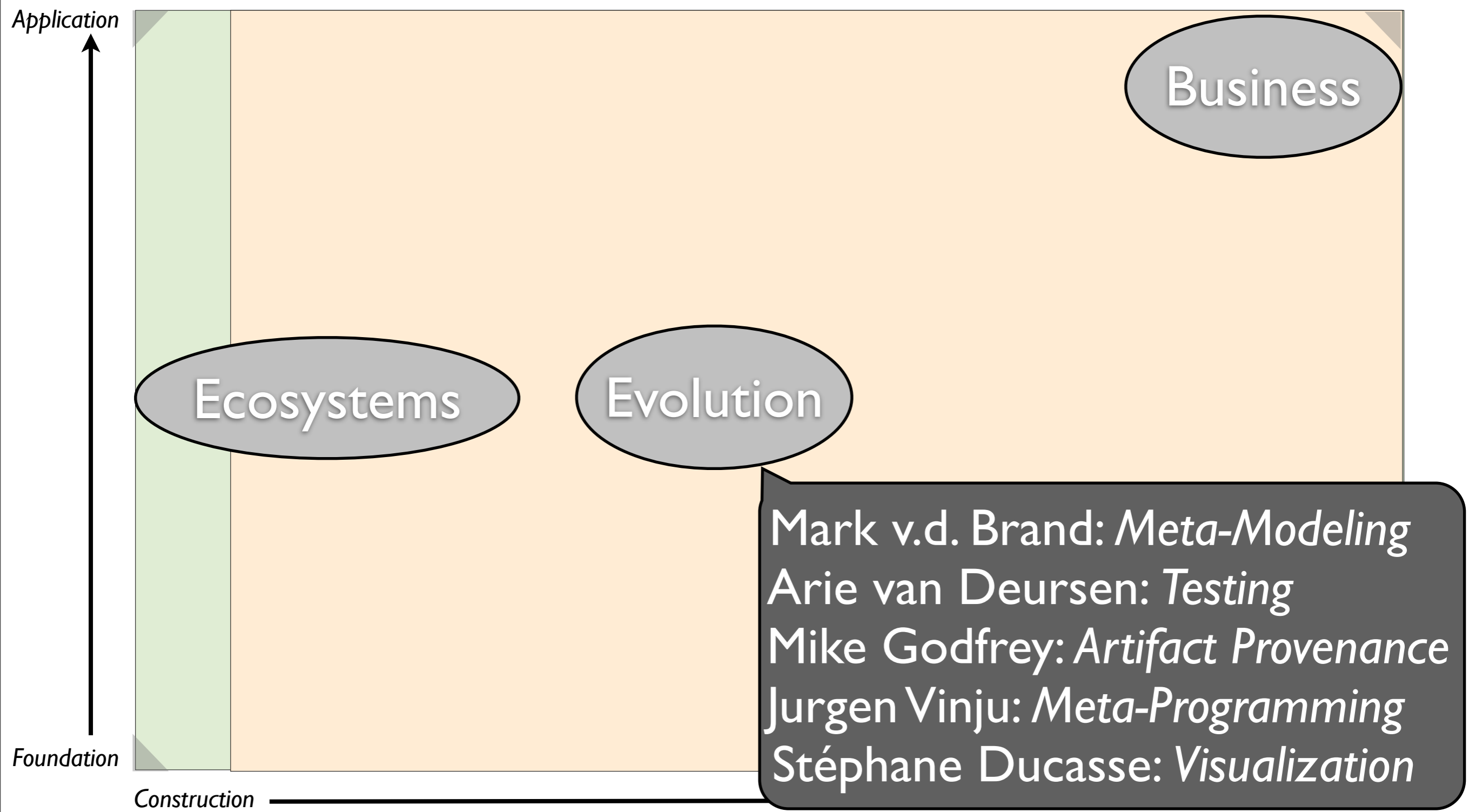
Understanding Software



Understanding Software



Understanding Software



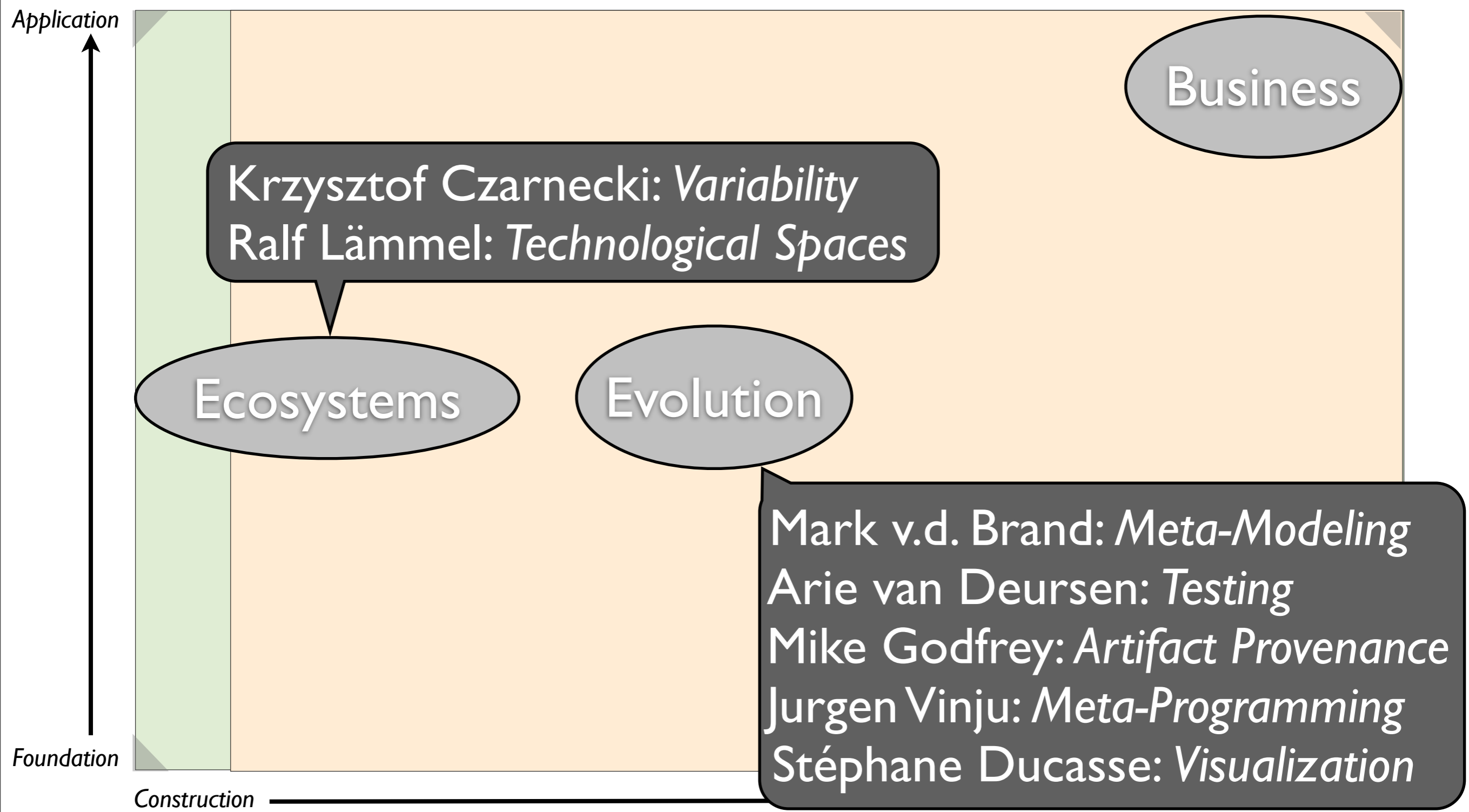
Business

Ecosystems

Evolution

Mark v.d. Brand: *Meta-Modeling*
Arie van Deursen: *Testing*
Mike Godfrey: *Artifact Provenance*
Jurgen Vinju: *Meta-Programming*
Stéphane Ducasse: *Visualization*

Understanding Software



Understanding Software

Application
↑
Foundation

Oege de Moor: *Business Intelligence*
Joost Visser: *Benchmarking Quality*

Business

Krzysztof Czarnecki: *Variability*
Ralf Lämmel: *Technological Spaces*

Ecosystems

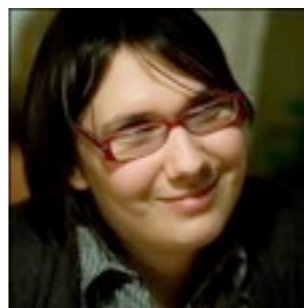
Evolution

Mark v.d. Brand: *Meta-Modeling*
Arie van Deursen: *Testing*
Mike Godfrey: *Artifact Provenance*
Jurgen Vinju: *Meta-Programming*
Stéphane Ducasse: *Visualization*

Construction

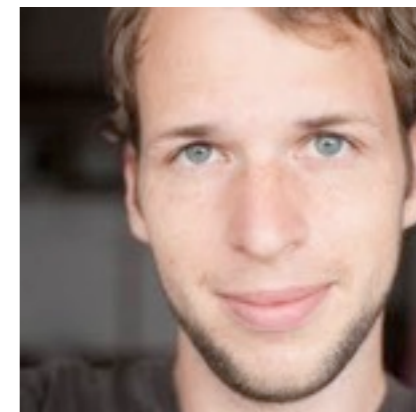


Incomplete list of former SWAT and UvA colleagues





The **SWAT** Team



Enjoy
Understanding Software