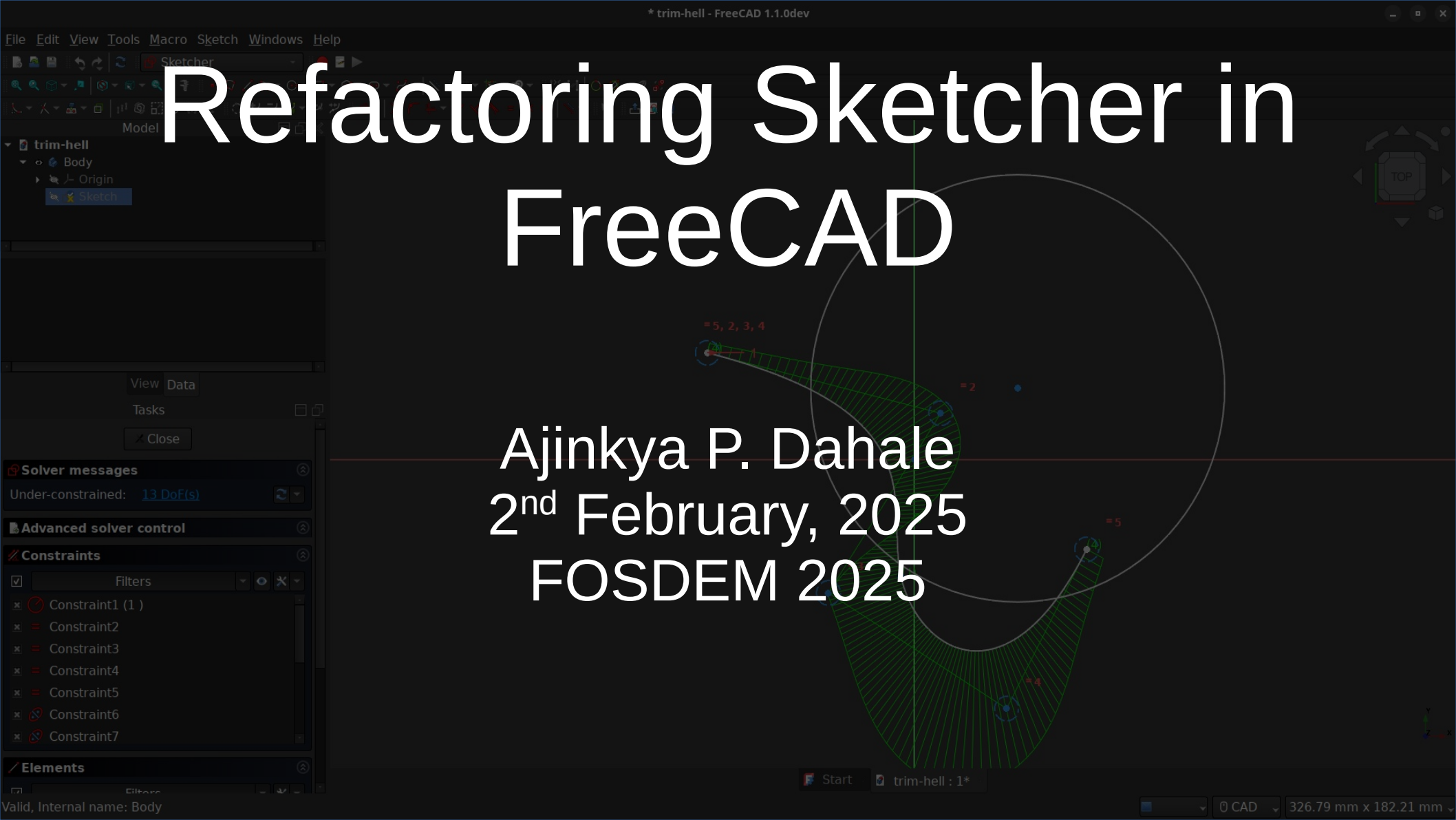


Refactoring Sketcher in FreeCAD

Ajinkya P. Dahale

2nd February, 2025

FOSDEM 2025

About FreeCAD

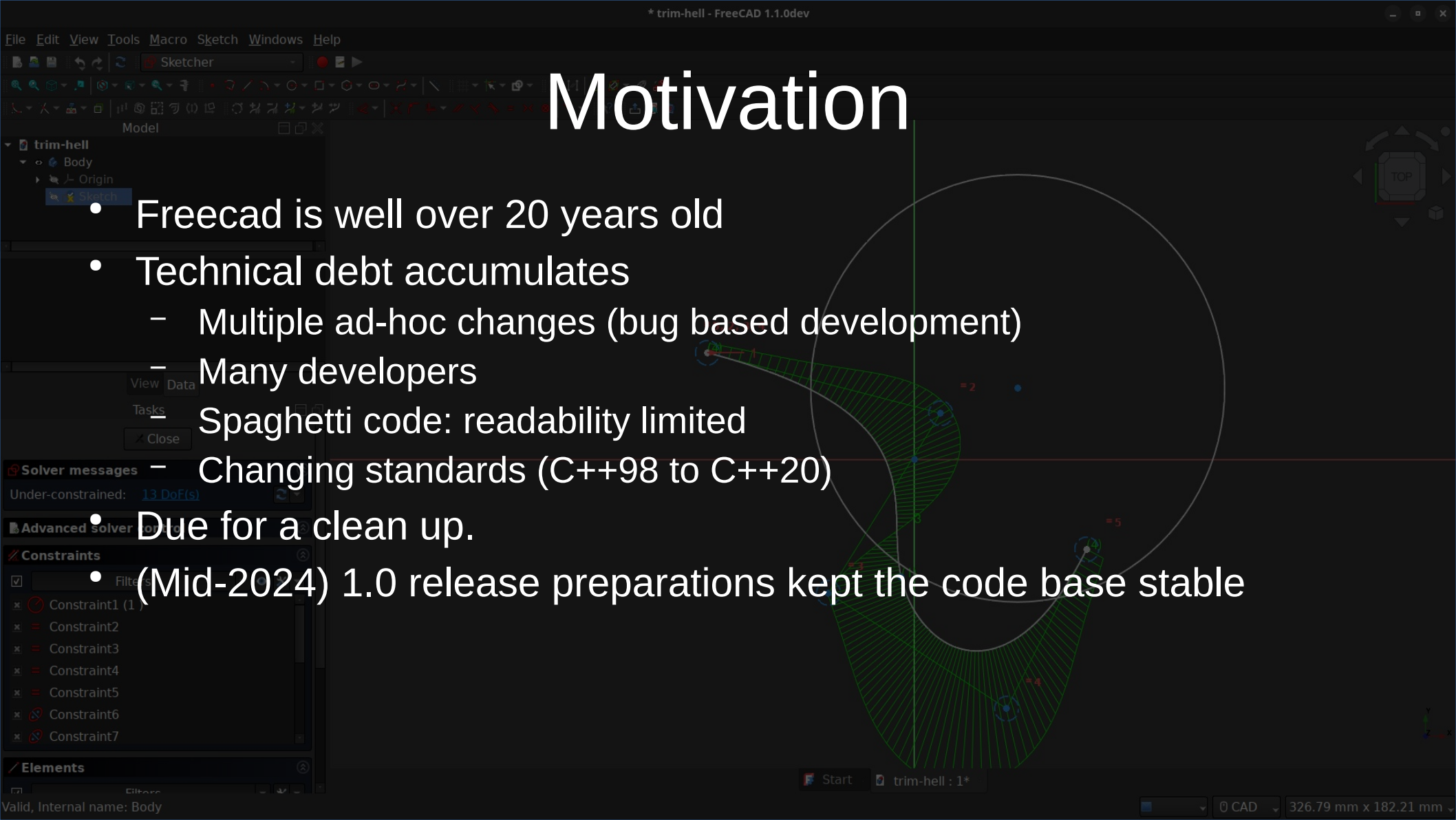
- Complete parametric 3D modeling tool, in development since 2001
- Ability to model physical objects of any size
- For more detail, check out our stand in the AW building (or https://freecad.org if watching remotely)

About Me

- FreeCAD contributor since 2016.
- Mainly focused on Sketcher, Part Design and related topics.
- Presently also working as project staff at FOSSEE, IIT Bombay (https://fossee.in)
- Occasionally available on most “professional” social media: just search for “Ajinkya Dahale”

Motivation

- Freecad is well over 20 years old
- Technical debt accumulates
  - Multiple ad-hoc changes (bug based development)
  - Many developers
  - Spaghetti code: readability limited
  - Changing standards (C++98 to C++20)
- Due for a clean up.
- (Mid-2024) 1.0 release preparations kept the code base stable

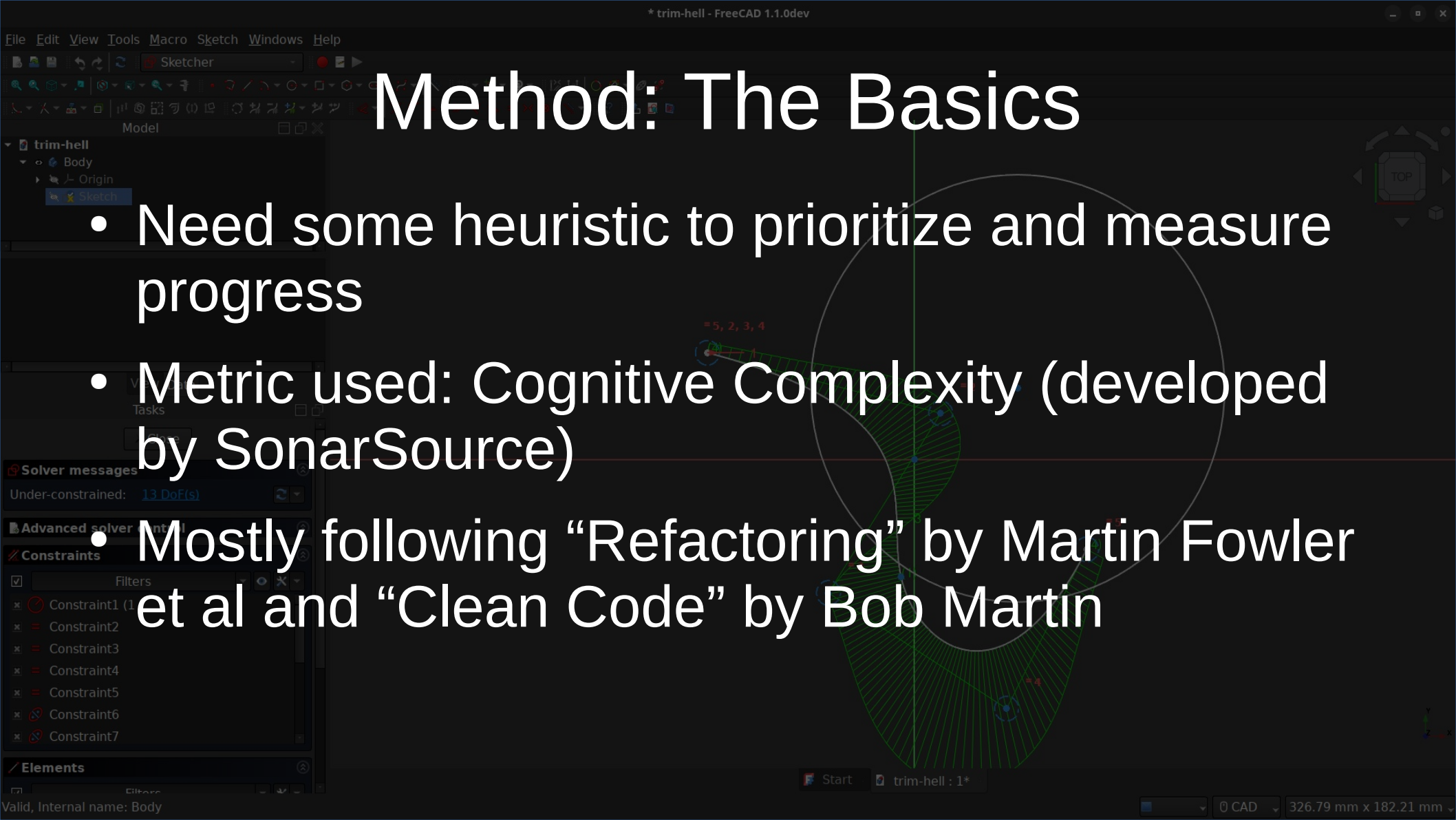


Why Sketcher?

0. Major portion of my FreeCAD contributions
1. First workbench a user encounters: the smallest of details WILL be caught
2. Large percentage of issues
3. Yet larger percentage of complaints on FreeCAD Days
4. Has multiple parts including an “in-house” solver (planegcs)

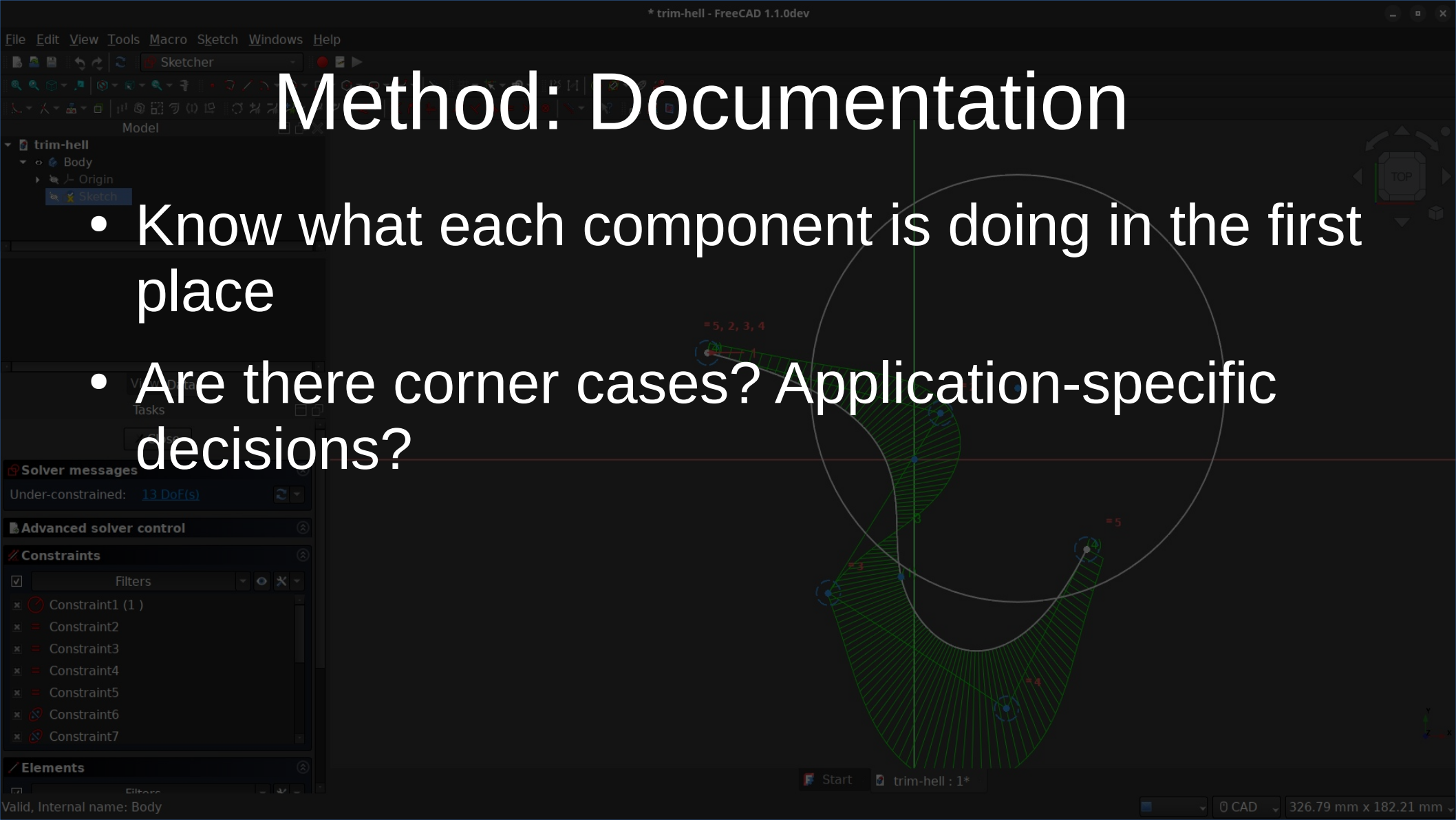
Method: The Basics

- Need some heuristic to prioritize and measure progress
- Metric used: Cognitive Complexity (developed by SonarSource)
- Mostly following “Refactoring” by Martin Fowler et al and “Clean Code” by Bob Martin



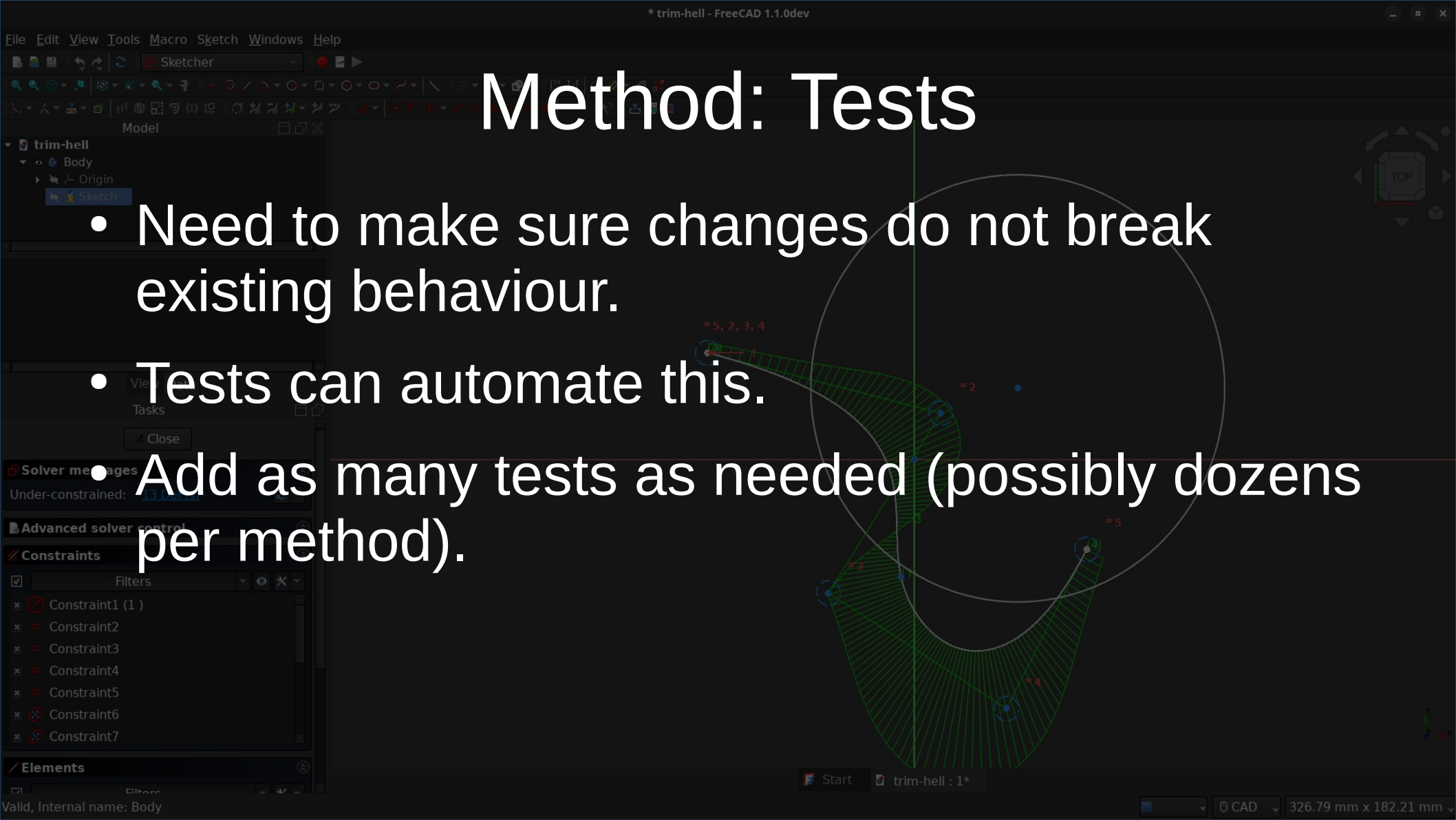
Method: Documentation

- Know what each component is doing in the first place
- Are there corner cases? Application-specific decisions?



Method: Tests

- Need to make sure changes do not break existing behaviour.
- Tests can automate this.
- Add as many tests as needed (possibly dozens per method).

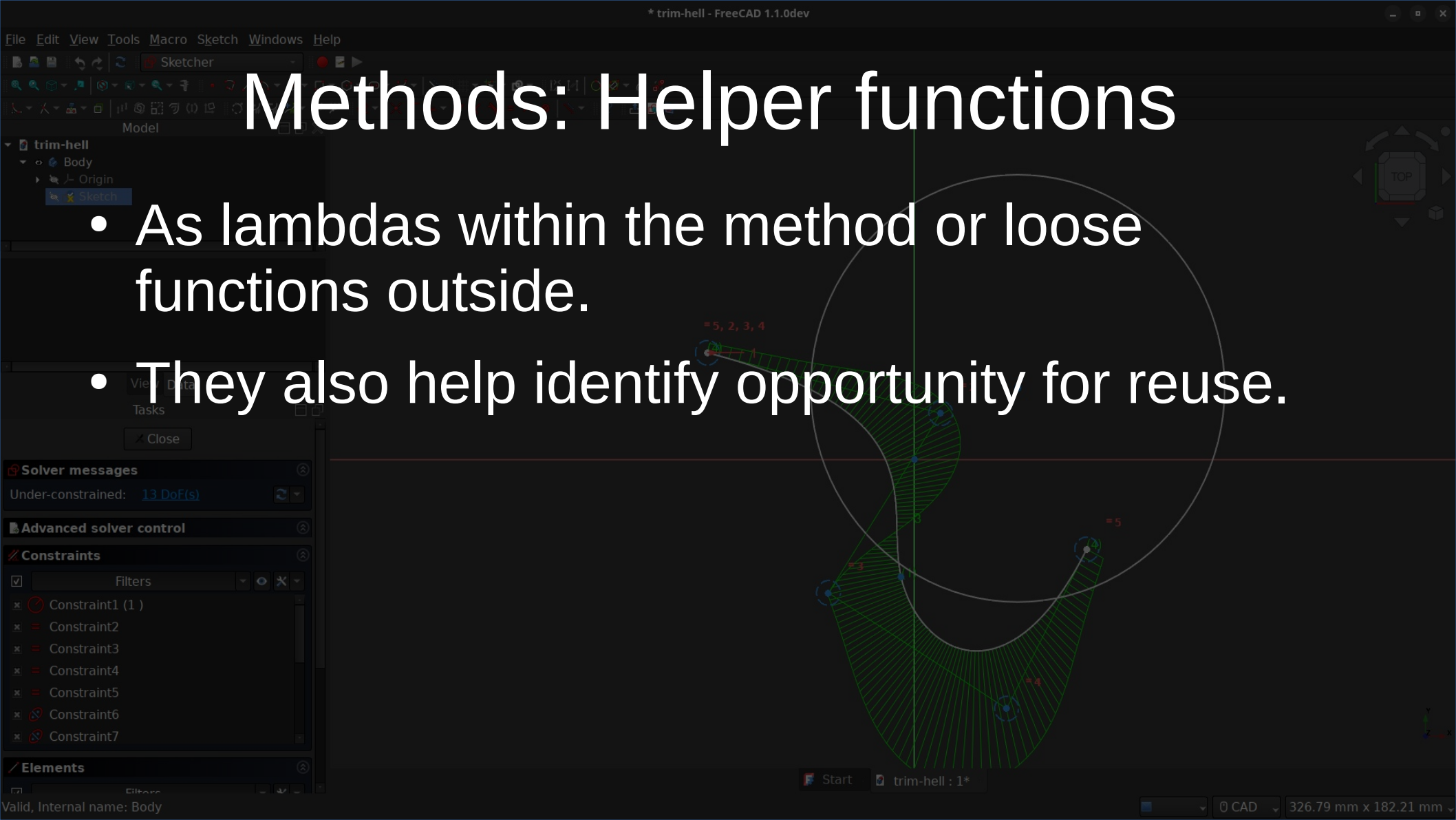


Method: Loops and Conditions

- Use modern C++ tools like ranged for and initialization within if statements as far as possible
- As far as possible, avoid nesting of if statements and for/while loops
- This includes the ternary operator “x?y:z”
- If needed, do not hesitate to repeat small bits of code (or create a helper function and repeat that)
  - This often comes into picture if one part of the condition is significantly larger than the other, and, after exiting, the method just wraps up and returns.

Methods: Helper functions

- As lambdas within the method or loose functions outside.
- They also help identify opportunity for reuse.

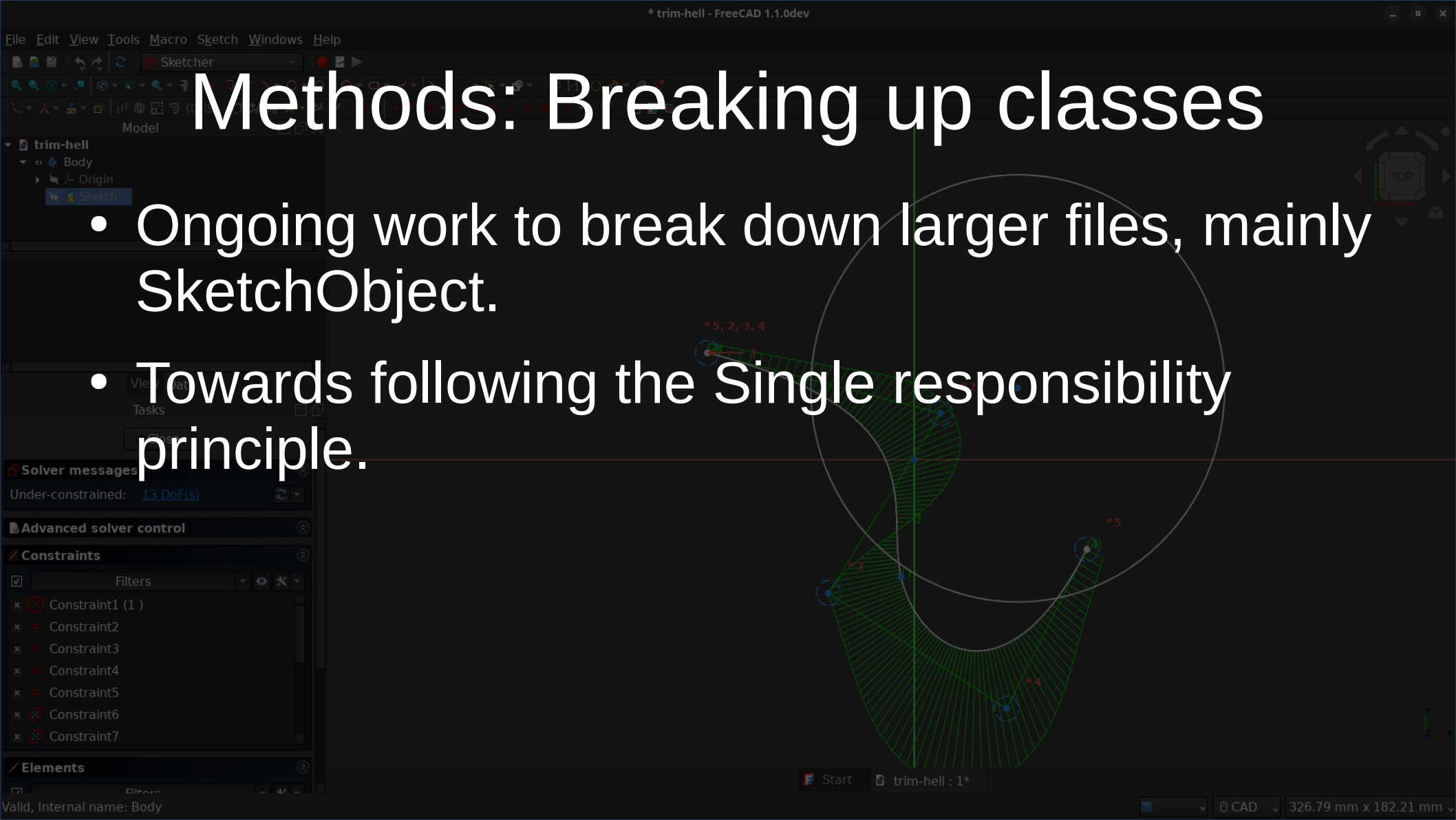


Methods: Use algorithm library

- As far as possible, use the algorithm library in C++ to replace combinations of loops and conditionals.
- Some examples: copy(_if), (all/any/none)_of, move, transform, (min/max)(_element), sort, partition
- Even a for_each can be useful
  - These methods also directly convey the intent
  - Possible opportunity to parallelize

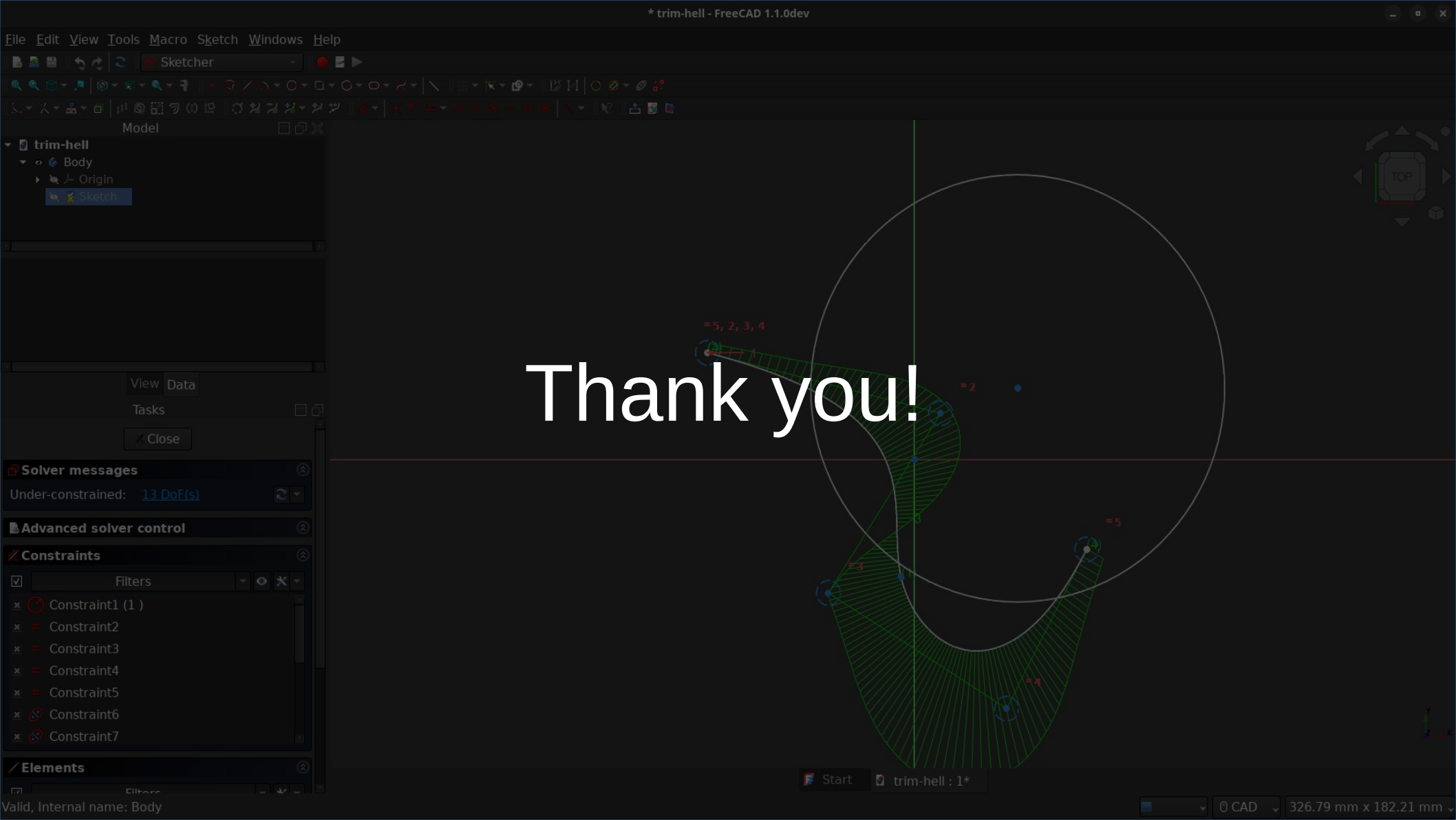
Methods: Breaking up classes

- Ongoing work to break down larger files, mainly SketchObject.
- Towards following the Single responsibility principle.



Challenges

- Limited Documentation
- Release done: floodgates of new PR, possibly causing conflicts
- All of this makes refactoring an iterative process
- Also best done by all developers
- Know when to stop
  - Is this code going to be changed again soon?


* trim-hell - FreeCAD 1.1.0dev
File Edit View Tools Macro Sketch Windows Help
Sketcher
Model
trim-hell
Body
Origin
Sketch
View Data
Tasks
Close
Solver messages
Under-constrained: 13 DoF(s)
Advanced solver control
Constraints
Filters
Constraint1 (1)
Constraint2
Constraint3
Constraint4
Constraint5
Constraint6
Constraint7
Elements
Thank you!
=5, 2, 3, 4
=2
=5
=3
=4
Start
trim-hell : 1*
Valid, Internal name: Body
CAD
326.79 mm x 182.21 mm