

# Literate Scientific Software & The Drasil Framework

Dan Szymczak

Computing and Software Department  
Faculty of Engineering  
McMaster University

Ernie Mileta Visit, Jan. 24, 2017

# Overview

FTS

Drasil

Results

Next Steps

1 Fully Traceable Software

2 Drasil Today

3 Results

4 Next Steps

# Fully Traceable Software

Slide 3 of 25

FTS

Drasil

Results

Next Steps

- Motivation
  - Improve verifiability, maintainability and reusability.
  - Save money and time
- One “source,” multiple views
  - Requirements
  - Design
  - Test Cases
  - Build instructions
  - ...

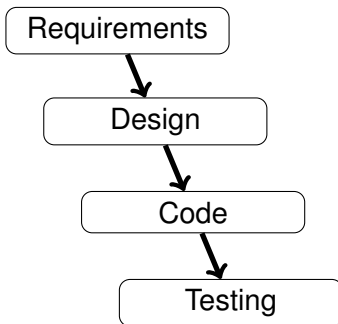
# Motivation

FTS

Drasil

Results

Next Steps



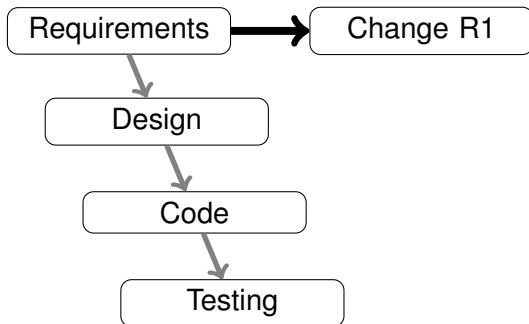
# Motivation

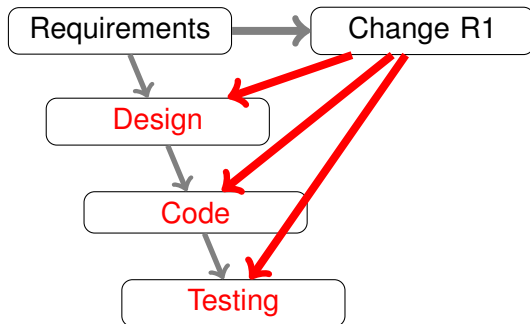
FTS

Brasil

Results

Next Steps





# Motivation

## An “Anonymous” Quote

FTS

Drasil

Results

Next Steps

*Last year one of a couple vendors presented results after they tried to update an analysis from 40 years ago and because the analysis models are not well documented they resulted in different final conclusions which had the potential to significantly increase operating costs. I've had to spend the last 3 months going over tracking down the impact of each new assumption in turn and testing them against available information to see how the original results were achieved. I've found several design features that were credited in the original work but not well explained in one case and assumptions about how the software worked that were misleading in another. Had this information been better documented a lot of wasted effort could have been avoided.*

- *Knowledge capture, chunks, and recipes*
- Common Knowledge Database
- Document Language
- Steve
- Summer Students Phase 1





# Recap - Knowledge Capture

Slide 9 of 25

FTS

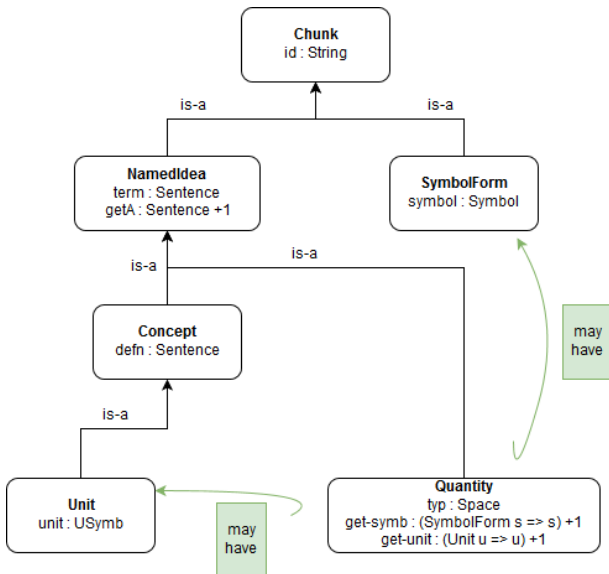
Drasil

Results

Next Steps



## Recap - Chunks



## Recap - Generation

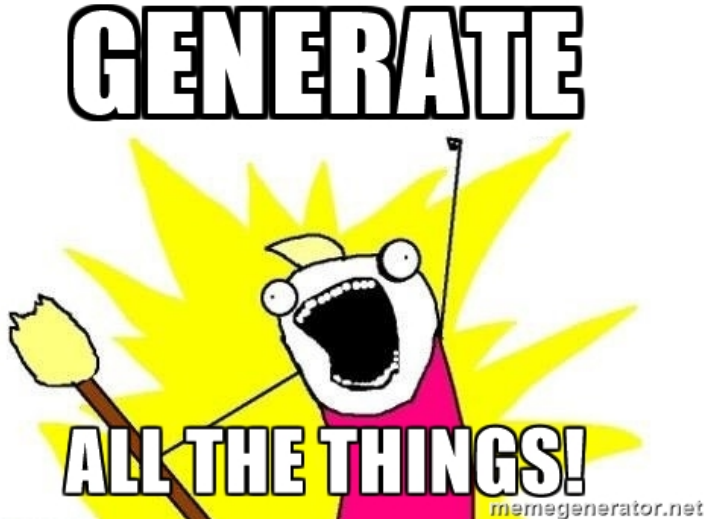
Slide 11 of 25

FTS

Drasil

Results

Next Steps



# Recap - Drasil

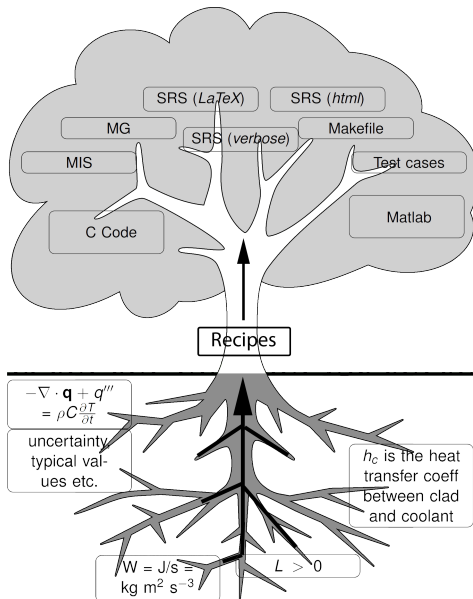
Slide 12 of 25

FTS

Drasil

Results

Next Steps



# What's New

## Summer Students Phase 2

FTS

Brasil

Results

Next Steps

- Example clean-up
- Knowledge extraction (common + specific)
- Pattern finding & combinator creation

# What's New

Data.Drasil

FTS

Drasil

Results

Next Steps

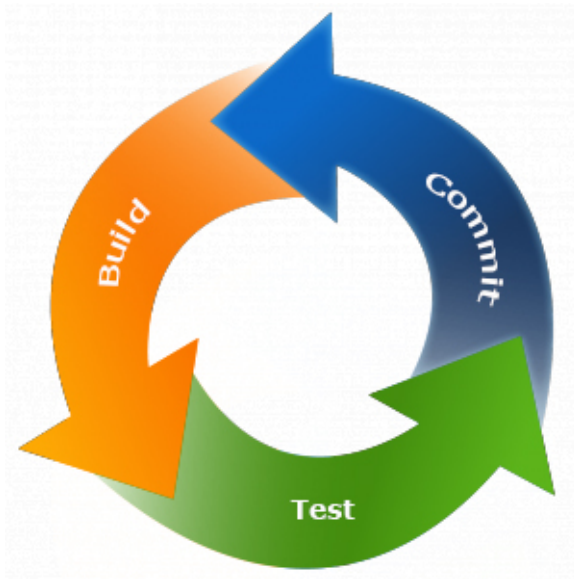
Common Knowledge base expanded! Now includes:

- 1 Documentation
- 2 Thermodynamics
- 3 Computation
- 4 Physics
- 5 Math
- 6 Solid Mechanics

and more!

# What's New

## Continuous Integration



FTS

Drasil

Results

Next Steps

# What's New

Yuzhi and Devi

FTS

Brasil

Results

Next Steps

New graduate students as of September 2017.

- Reviewed and updated manual versions of current examples
- Implementing Document Language through examples.
- ... and more to come!



# Design Changes

## New Knowledge Capture Mechanisms

FTS

Drasil

Results

Next Steps

We are able to capture much more information in a ‘useful’ form

- ① Theories
- ② Assumptions
- ③ Requirements
- ④ Instance Models

...

# Design Changes

## Document Language – Old (Recipes)

FTS

Drasil

Results

Next Steps

```
RefSec (RefProg intro [
  TUnits,
  tsymb" s1_2_intro (TermExcept [norm_vect]),
  TAandA]) :
map Verbatim [s2, s3, s4, s5, s6, s7]

s2 = Section ... ..
s3 = Section ... ..
s4 = Section ... ..
s5 = Section ... ..
s6 = Section ... ..
s7 = Section ... ..
```

# Design Changes

## Document Language – New (Recipes)

```

RefSec (RefProg intro
  [TUnits, tsymb [TSPurpose, SymbOrder], TAandA]) :
IntroSec ( IntroProg (startIntro ...) (short gLassBR)
  [IPurpose (s2_1_intro_p1 document gLassBR glaSlab),
    IScope incScoR endScoR,
  ...
GSDSec (GSDProg2 [UsrChars ...], SystCons [] []]) :
ScpOfProjSec (ScpOfProjProg ...) :
SSDSec (SSDProg
  [SSDProblem (PDProg ... [s6_1_1, s6_1_2, s6_1_3])
    , SSDSolChSpec
      (SCSProg
        [ TMs ([Label] ++ stdFields) [t1IsSafe]
          , GDs [] [] HideDerivation — No Gen Defs for GlassBR
          , DDs ... , IMs ... ]))) :
ReqmntSec (ReqsProg [
  FReqsSub s7_1_list,
  NonFReqsSub [performance] (gBRpriorityNReqs) :
  ...
AuxConstntSec (AuxConsProg gLassBR auxiliaryConstants) :
Bibliography gbCitations :
AppndxSec (AppndxProg [s12_intro, fig_5, fig_6]) : []

```

FTS

Drasil

**Results**

Next Steps

# Results

## Document Language – New (Recipes)

GlassBR\_SRS.html

# Results

## Sanity Checking

### SSP Example (Issue #348)

$$S_i = \frac{P_i}{FS} \quad (1) \qquad FS = \frac{S_i}{\tau_i} \quad (2)$$

Where did  $\tau_i$  come from?

Were  $S_i$  and  $P_i$  swapped?

- $\tau_i$  was not defined anywhere in the documents
- Found with Drasil – undefined symbols throw errors
- Equation based on concepts – symbols automatically retrieved

# Results

## Sanity Checking Cont'd

- Stricter (more formalized) approach reveals hidden errors.
- Fixing these kinds of errors is easy thanks to tool support from Drasil

# Results

## Conceptual Inconsistencies

Manually created artifacts are human-readable.  
Problems arise when attempting to explain things to a machine.

- What do our artifacts *mean*?
- What is each section contributing?
- Why do we organize things a given way?
- Are models/definitions different? How?

Need to be more rigorous!

## What next?

- Recipe language implementation
- More examples (in new domains)
- More artifacts
- Design Language



FTS

Drasil

Results

Next Steps

# Thank You!