

# Sustainable Software Product Lines via Generation

**Spencer Smith** and Jacques Carette

Computing and Software Department  
Faculty of Engineering  
McMaster University

Huawei 2023 Visit: Feb?, 2023

# Generate All Things with Drasil

Smith & Carette,  
Slide 2 of 31

Introduction

GlassBR  
Example

Idealized  
Process

Inputs

Concluding  
Remarks

References

- **Goal** — Improve software sustainability and productivity
- **Ideas**
  - Adapt a software product line approach
  - Build on success of MDSE
  - Value all documents, not just code
  - Start with well-understood domains
- **Solution**
  - Capture (codify) knowledge **once**
  - Generate all **documentation** and code
  - Idealized dev process for **well-understood** software
- **Implement Partial Solution — Drasil**

# Product Lines



# Product Lines in User Manual

## PRODUCT SPECIFICATIONS

The appearance and specifications listed in this manual may vary due to constant product improvements.

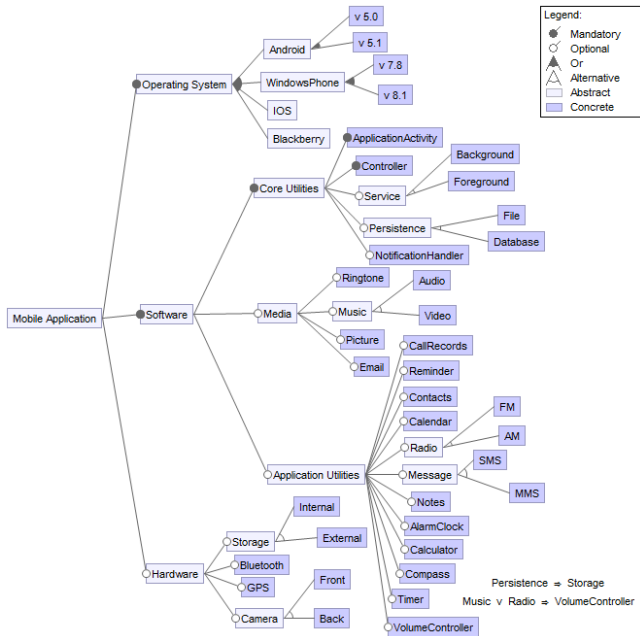
**Electrical requirements:** 115 V, 60 Hz

**Min. / Max. water pressure:** 20 - 120 psi (138 - 827 kPa)

<b>Model</b>	LFCC22426*
<b>Description</b>	Counter-depth, French door refrigerator, bottom freezer
<b>Net weight</b>	243 lb (110 kg)

<b>Model</b>	LFCS27596*
<b>Description</b>	Standard-depth, Door-in-Door French door refrigerator, bottom freezer
<b>Net weight</b>	284 lb (129 kg)

<b>Model</b>	LFCC23596*
<b>Description</b>	Counter-depth, Door-in-Door French door refrigerator, bottom freezer
<b>Net weight</b>	269 lb (122 kg)



Usman et al. (2017)

## Build on Success of MDSE

Smith & Carette,  
Slide 6 of 31

Introduction

GlassBR  
Example

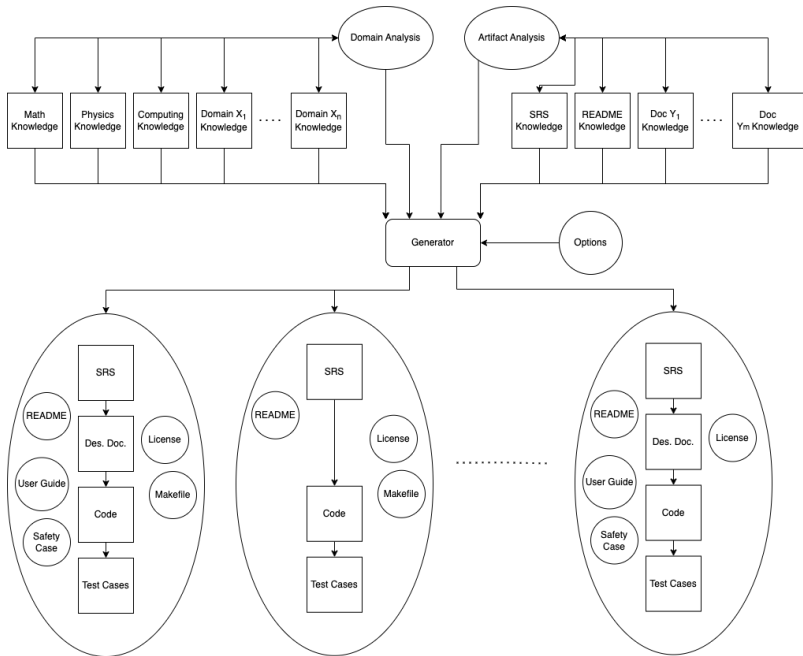
Idealized  
Process

Inputs

Concluding  
Remarks

References

- Codify (capture) code and non-code info together
  - Natural language (text)
  - Definitions
  - Assumptions
  - Derivations
  - Rationale
  - Abstract theory
  - User characteristics
  - Nonfunctional Requirements
  - Etc.
- Generate all artifacts from one framework
  - Requirements
  - User manuals
  - README
  - Build scripts, dev environment (CI etc)
  - Assurance case
  - Code (in different languages)
  - Test cases
  - etc.



# Why Generate All Things?

Smith & Carette,  
Slide 8 of 31

Introduction

GlassBR  
Example

Idealized  
Process

Inputs

Concluding  
Remarks

References

- Different views of the same knowledge
- Generate the view needed by the audience
- Change to different natural languages
- Stay abstract longer
- Certification
- Documentation improve productivity, sustainability
- Reusability
- Modifiability
- One source helps synchronization



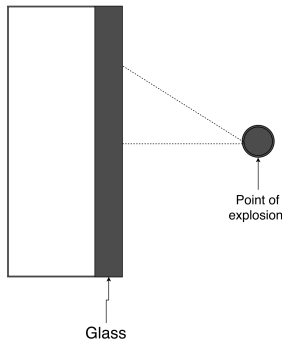


# GlassBR

## Given

- dimensions of glass plane
- glass type
- explosion characteristics
- tolerable breakage probability

Predict whether the glass will withstand the explosion



## Drasil Inputs:

- Program Name: GlassBR
- Authors: Nikitha K and Spencer S
- Symbols: tolerable load ( $\hat{q}_{tol}$ ), Risk of failure ( $B$ ), ...
- Assumptions: Load duration factor constant,
- Data definitions: relation for  $B$ , ...
- Design decisions:
  - Modularity (input module),
  - Implementation Type (Program),
  - Logging (Yes),
  - Input Structure (Bundled),
  - Constant Structure (Inlined),
  - Constant Rep (Constants),
  - Real Number Rep (Double),
  - ...

## Drasil Inputs:

- Program Name: GlassBR
- Authors: Nikitha K and Spencer S
- Symbols: tolerable load ( $\hat{q}_{tol}$ ), Risk of failure ( $B$ ), ...
- Assumptions: Load duration factor constant,
- Data definitions: relation for  $B$ , ...
- Design decisions:
  - Modularity (input module),
  - Implementation Type (Program),
  - Logging (Yes),
  - Input Structure (Bundled),
  - Constant Structure (Inlined),
  - Constant Rep (Constants),
  - Real Number Rep (Double),
  - ...

```
/glassbr
/Website/GlassBR_SRS.html
/Website/GlassBR_SRS.css
/SRS/bibfile.bib
/SRS/Makefile
/SRS/GlassBR_SRS.tex
/SRS/GlassBR_SRS.pdf
/src/python
/src/python/README.md
/src/python/InputParameters.py
/src/python/Calculations.py
/src/python/Makefile
/src/python/doxConfig
...
```

```
...
/src/java/GlassBR/Calculations.java
/src/java/Makefile
/src/java/README.md
...
/src/cpp/GlassBR
/src/cpp/ReadTable.cpp
/src/cpp/InputFormat.hpp
/src/cpp/Calculations.cpp
...
/src/swift/Calculations.swift
...
/src/csharp/Control.cs
...
```

/glassbr  
/Website/GlassBR\_SRS.html  
/Website/GlassBR\_SRS.css  
/SRS/bibfile.bib  
/SRS/Makefile  
/SRS/GlassBR\_SRS.tex  
/SRS/GlassBR\_SRS.pdf  
/src/python  
/src/python/README.md  
/src/python/InputParameters.py  
/src/python/Calculations.py  
/src/python/Makefile  
/src/python/doxConfig  
...

...  
/src/java/GlassBR/Calculations.java  
/src/java/Makefile  
/src/java/README.md  
...  
/src/cpp/GlassBR  
/src/cpp/ReadTable.cpp  
/src/cpp/InputFormat.hpp  
/src/cpp/Calculations.cpp  
...  
/src/swift/Calculations.swift  
...  
/src/csharp/Control.cs  
...

# Software Requirements Specification for GlassBR

Nikitha K and Spencer S

html

## Table of Symbols

$\hat{q}_{\text{tol}}$

$B$

...

## Introduction

... The software, herein called GlassBR, ...

## Assumptions

ldfConstant: LDF is constant, depends on assumed value of  $t_d$  and  $m$ , ...

## Data Definitions

$$B = \frac{k}{(ab)^{m-1}} (Eh^2)^m \text{LDF} e^J$$

...

$$B = \frac{k}{(ab)^{m-1}} (Eh^2)^m \text{LDF} e^J$$

sBR

## **GlassBR**

Authors Nikitha K and Spencer S

**How to Run the Program:** In your terminal command line, enter the same directory as this README file. Then enter the following line

```
make run RUNARGS=input.txt
```

**Configuration Files:** SDF.txt, TSD.txt must be in the same directory as the executable to run successfully

Versioning: Python Version 3.5.1

```
...
```

```
build:
```

```
run: build
```

```
python Control.py
```

```
...
```

```
build: GlassBR/Control.class
```

```
...
```

```
GlassBR/Control.class:
```

```
GlassBR/Control.java ...
```

```
javac GlassBR/Control.java
```

```
run: build
```

```
java GlassBR.Control $(RUNARGS)
```

```
...
```



Calculations.py

Calculations.java

```
## \file Calculations.py
# \author Nikitha Krithnan and W. Spencer Smith
# \brief package GlassBR
...
## \file Calculations.java
## \author Nikitha Krithnan and W. Spencer Smith
# \para \brief Provides functions for calculating the outputs
# \para
# \retu */
def func... public static double func_B(InputParameters inParams, double J) throws IOException {
    out PrintWriter outfile;
    pri outfile = new PrintWriter(new FileWriter(new File("log.txt"), true));
    ... outfile.println("function func_B called with inputs: {}");
    out
    ...
    ret outfile.close();

    return 2.86e-53 /Math.pow(inParams.a * inParams.b, 7.0 - 1.0) *
        Math.pow(7.17e10 * Math.pow(inParams.h, 2.0), 7.0) * inParams.LDF
        * Math.exp(J);
}
```



# $J_{tol}$ in SRS.pdf

Smith & Carette,  
Slide 18 of 31

Introduction

GlassBR  
Example

Idealized  
Process

Inputs

Concluding  
Remarks

References

Refname	DD:sdfTol
Label	Stress distribution factor (Function) based on Pbtol
Symbol	$J_{tol}$
Units	Unitless
Equation	$J_{tol} = \ln \left( \ln \left( \frac{1}{1 - P_{btol}} \right) \frac{\left( \frac{a}{1000} \frac{b}{1000} \right)^{m-1}}{k \left( E \cdot 1000 \left( \frac{h}{1000} \right)^2 \right)^m LDF} \right)$
Description	<p><math>J_{tol}</math> is the stress distribution factor (Function) based on Pbtol (Unitless)</p> <p><math>P_{btol}</math> is the tolerable probability of breakage (Unitless)</p> <p><math>a</math> is the plate length (long dimension) (m)</p> <p><math>b</math> is the plate width (short dimension) (m)</p> <p><math>m</math> is the surface flaw parameter (<math>\frac{m^{12}}{N^7}</math>)</p> <p><math>k</math> is the surface flaw parameter (<math>\frac{m^{12}}{N^7}</math>)</p> <p><math>E</math> is the modulus of elasticity of glass (Pa)</p> <p><math>h</math> is the minimum thickness (m)</p> <p><math>LDF</math> is the load duration factor (Unitless)</p>

## $J_{tol}$ in SRS.tex

...

Label & Stress distribution factor (Function) based on  
Pbtol

\\ \midrule \\  
Symbol &  $\{J_{\text{tol}}\}$

\\ \midrule \\  
Units & Unitless

\\ \midrule \\  
Equation & 
$$\{J_{\text{tol}}\} = \ln \left( \ln \left( \frac{1}{1 - \{P_{\text{b}}\}_{\text{tol}}}\right) \frac{\frac{1}{1000} - \frac{\{b\}}{1000}}{\frac{\{a\}}{1000} - \frac{\{b\}}{1000}} \right)^{\{m-1\}} \{k \left( E \cdot 1000 \left( \frac{\{h\}}{1000} \right)^2 \right)^{\{m\}} \text{LDF} \right)$$
  
\\ \end{displaymath} \\  
\\ \midrule \\  
Description & ...

## $J_{tol}$ in SRS.html

```
...
<th>Equation</th>
<td>
\[{J_{\text{tol}}}=\ln\left(\ln\left(\frac{1}{1-{P_{\text{b}}}\text{tol}}\right)\right)\frac{\left(\frac{a}{1000}\frac{b}{1000}\right)^{m-1}}{k\left(E\cdot 1000\frac{h}{1000}\right)^2\left(LDF\right)^m}\right]
</td>
...
```

## $J_{tol}$ in Python

```
## \brief Calculates stress distribution factor (Function
    ) based on Pbtol
# \param inParams structure holding the input values
# \return stress distribution factor (Function) based on
    Pbtol
def func_J_tol(inParams):
    outfile = open("log.txt", "a")
    print("function func_J_tol called with inputs: {" ,
        file=outfile)
    print("  inParams = " , end="", file=outfile)
    print("Instance of InputParameters object", file=
        outfile)
    print("  }", file=outfile)
    outfile.close()

    return math.log(math.log(1.0 / (1.0 - inParams.P_btoll
        )) * ((inParams.a / 1000.0 * (inParams.b /
        1000.0)) ** (7.0 - 1.0) / (2.86e-53 * (7.17e10 *
        1000.0 * (inParams.h / 1000.0) ** 2.0) ** 7.0 *
        inParams.LDF)))
```

## $J_{tol}$ in Java

```
/** \brief Calculates stress distribution factor (
    Function) based on Pbtol
    \param inParams structure holding the input
        values
    \return stress distribution factor (Function)
        based on Pbtol
*/
public static double func_J_tol(InputParameters
    inParams) throws IOException {
    PrintWriter outfile;
    outfile = new PrintWriter(new FileWriter(new File
        ("log.txt"), true));
    ...
    return Math.log(Math.log(1.0 / (1.0 - inParams.
        P_btoll)) * (Math.pow(inParams.a / 1000.0 * (
        inParams.b / 1000.0), 7.0 - 1.0) / (2.86e-53
        * Math.pow(7.17e10 * 1000.0 * Math.pow(
        inParams.h / 1000.0, 2.0), 7.0) * inParams.
        LDF))));
}
```

## $J_{tol}$ in Drasil (Haskell)

```
tolStrDisFacEq :: Expr
tolStrDisFacEq = ln (ln (recip_ (exactDbl 1 $- sy pbTol))
  `mulRe` (((sy plateLen $/ exactDbl 1000) `mulRe` (sy
    plateWidth $/ exactDbl 1000)) $^ (sy sflawParamM $-
    exactDbl 1) $/
    (sy sflawParamK `mulRe` ((sy modElas `mulRe` exactDbl
      1000 `mulRe`
    square (sy minThick $/ exactDbl 1000)) $^ sy
    sflawParamM) `mulRe` sy lDurFac)))
```

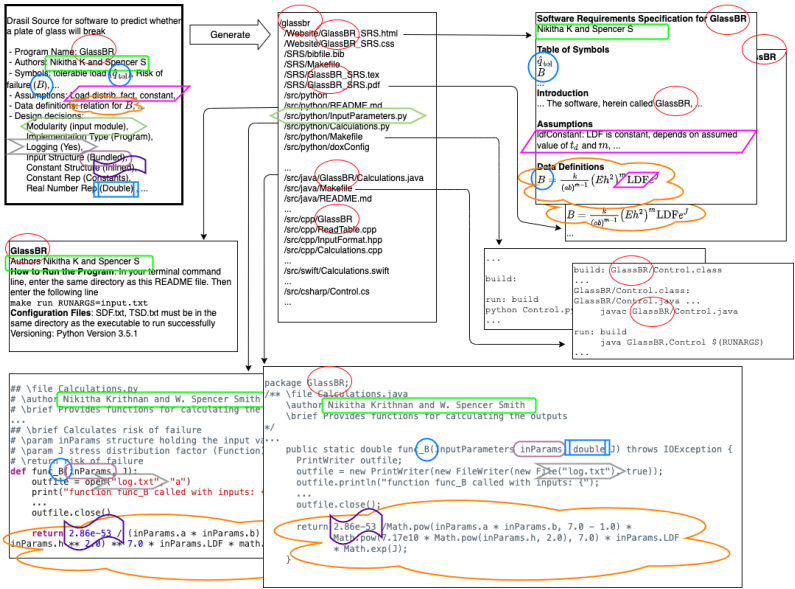
## $J_{tol}$ without Unit Conversion

```
tolStrDisFacEq :: Expr
tolStrDisFacEq = ln (ln (recip_ (exactDbl 1 $- sy pbTol))
  `mulRe` ((sy plateLen `mulRe` sy plateWidth) $^ (sy
    sflawParamM $- exactDbl 1) $/
    (sy sflawParamK `mulRe` ((sy modElas `mulRe`
      square (sy minThick)) $^ sy sflawParamM) `mulRe` sy
      lDurFac)))
```



## Drasil Inputs:

- Program Name: GlassBR
- Authors: Nikitha K and Spencer S
- Symbols: tolerable load ( $\hat{q}_{tol}$ ), Risk of failure ( $B$ ), ...
- Assumptions: Load duration factor constant,
- Data definitions: relation for  $B$ , ...
- Design decisions:
  - Modularity (input module),
  - Implementation Type (Program),
  - Logging (Yes),
  - Input Structure (Bundled),
  - Constant Structure (Inlined),
  - Constant Rep (Constants),
  - Real Number Rep (Double),
  - ...



# Idealized Process

- From wu paper

# Capturing Knowledge: Inputs

Smith & Carette,  
Slide 28 of 31

Introduction

GlassBR  
Example

Idealized  
Process

Inputs

Concluding  
Remarks

References

- Probably won't have time to discuss

# Concluding Remarks

Smith & Carette,  
Slide 29 of 31

Introduction

GlassBR  
Example

Idealized  
Process

Inputs

Concluding  
Remarks

References

- **Take Home Message** — Sustainability and Productivity can potentially be improved via a Generate All Things Approach

Muhammad Usman, Muhammad Zohaib Iqbal, and Muhammad Uzair Khan. A product-line model-driven engineering approach for generating feature-based mobile applications. *Journal of Systems and Software*, 123:1–32, 01 2017. doi: 10.1016/j.jss.2016.09.049.

# Image Credits

Smith & Carette,  
Slide 31 of 31

Introduction

GlassBR  
Example

Idealized  
Process

Inputs

Concluding  
Remarks

References

- Apple Product Line Image
- Dodge Lineup