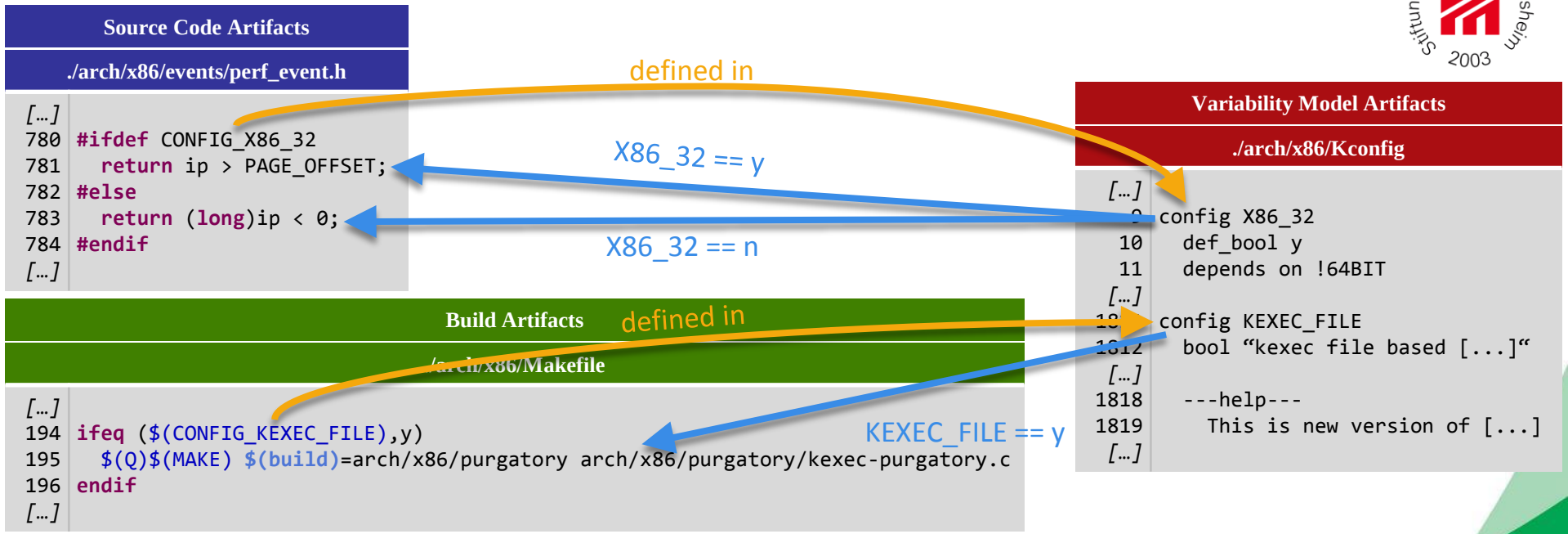
REVaMP²
PSS-CE

Problem-Solution-Space Co-Evolution

University of Hildesheim (SUH)

Purpose
PSS-CE

- Desired relations between problem and solution space artifacts:



Purpose
PSS-CE

- But divergences between artifacts may exist, like:

Source Code Artifacts

./arch/x86/events/perf_event.h

```
[…]
780 #ifdef CONFIG_X86_32
781   return ip > PAGE_OFFSET;
782 #else
783   return (long)ip < 0;
784 #endif
[…]
```

Find and correct unintended divergences

Variability Model Artifacts

./arch/x86/Kconfig

```
[…]
  9
 10
 11
[…]
1811 config KEXEC_FILE
1812   bool “kexec file based [...]“
[…]
1818   ---help---
1819     This is new version of [...]
[…]
```

? Undefined

Build Artifacts

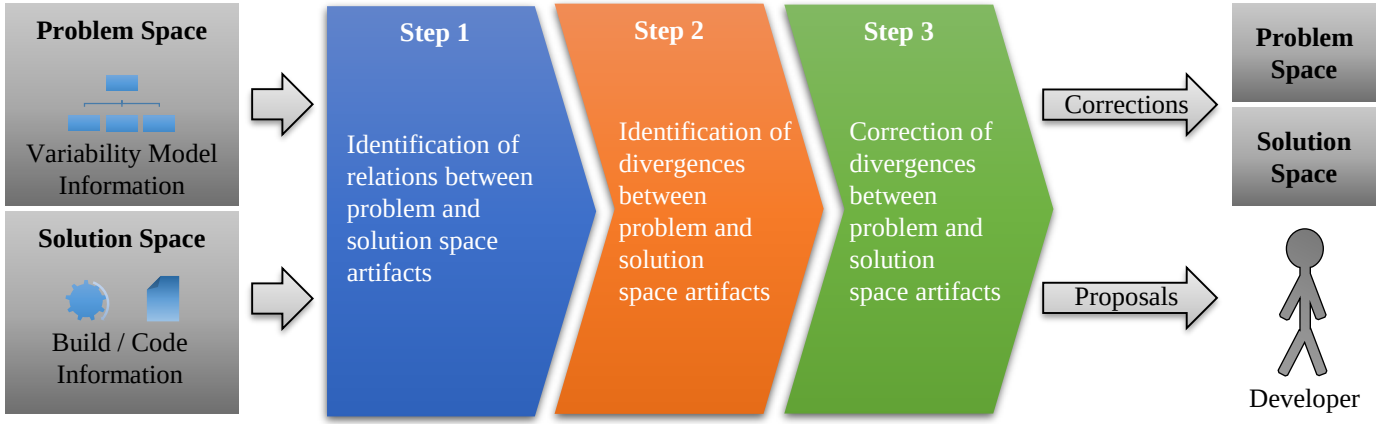
./arch/x86/Makefile

```
[…]
194
195
196
[…]
```

Unused ?

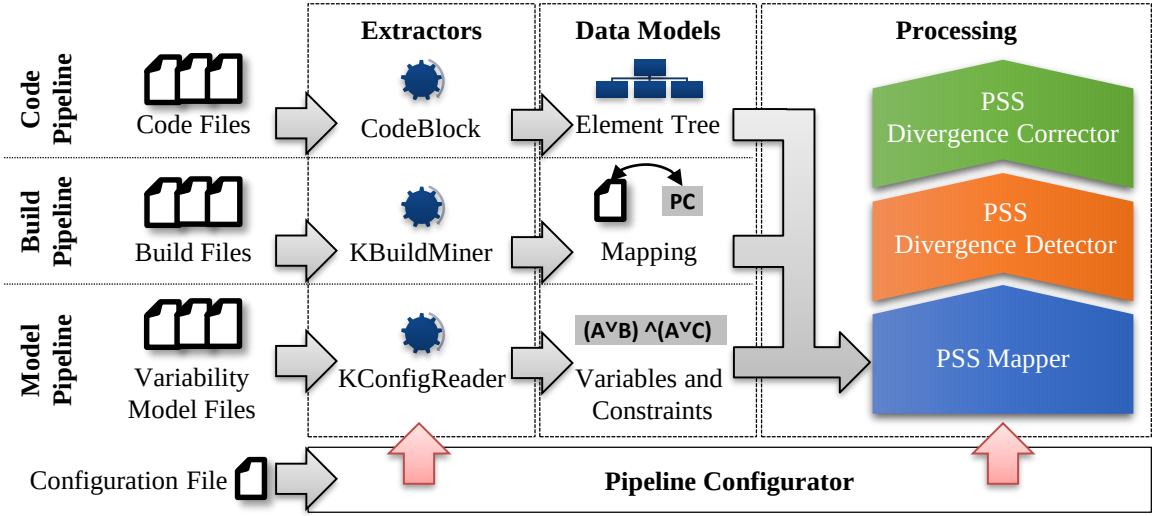
REVaMP²

Main Features
PSS-CE



Main Features
PSS-CE

- Technical realization as plug-ins for KernelHaven:



Walkthrough
PSS-CE

As progress beyond D5.4, we focus on the incremental PSS-CE support here

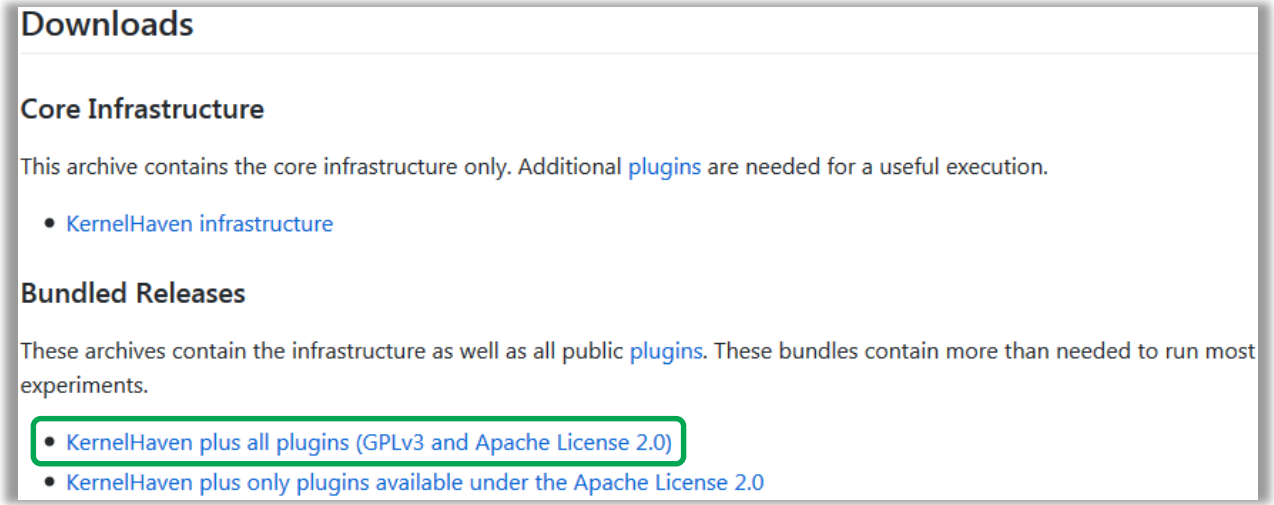
1.  Download and Installation
2.  Preparation of Evolution History
3.  Baseline Creation for Incremental PSS-CE
4.  Incremental PSS-CE Analysis

1. Download and Installation
PSS-CE

- General requirements:
  - Use machine with at least 16GB internal memory
  - Use Ubuntu as operating system
  - Install:
    - Java 8 or higher
    - Git
    - build-essential
    - libelf-dev
- Example: *sudo apt install openjdk-8-jdk git build-essential libelf-dev*

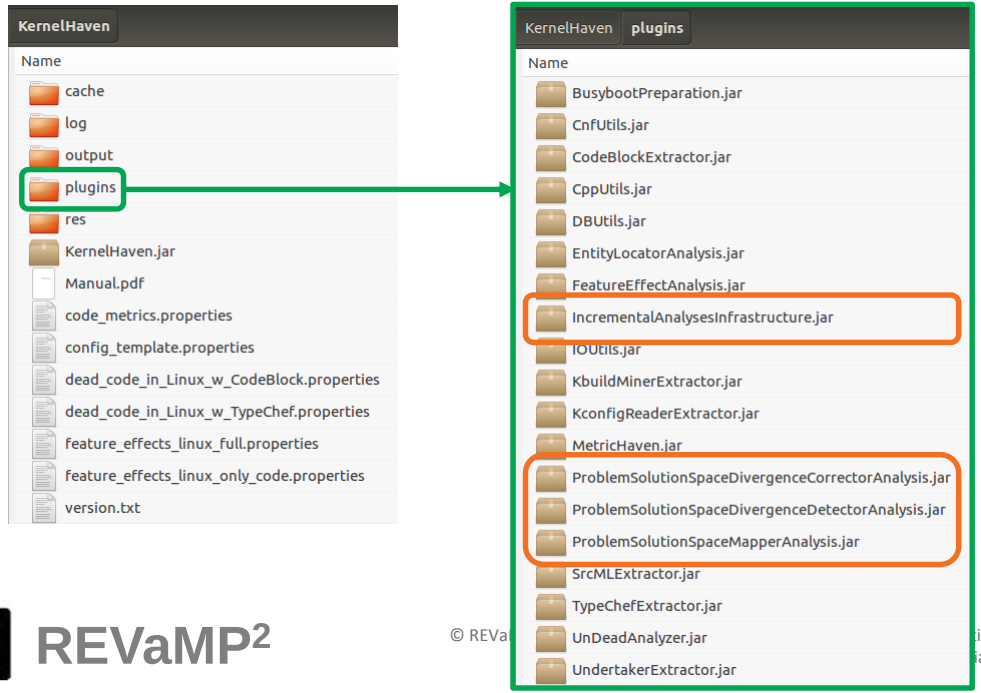
1. Download and Installation
PSS-CE

- Download latest KernelHaven release:
  - URL: https://github.com/KernelHaven/KernelHaven
  - **KernelHaven plus all plugins (GPLv3 and Apache License 2.0)**



1. Download and Installation
PSS-CE

- Extract the KernelHaven archive to your favorite location
- Here we use *KernelHaven/* as location

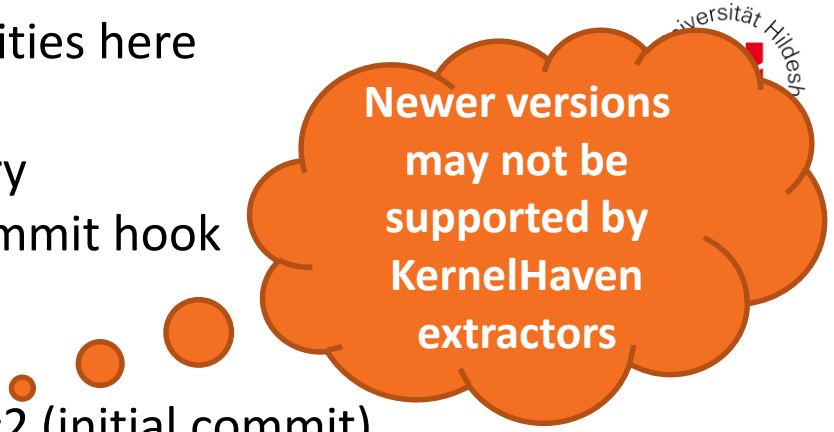


Plug-in for incremental PSS-CE support
(Extension to D5.4)

PSS-CE plug-ins

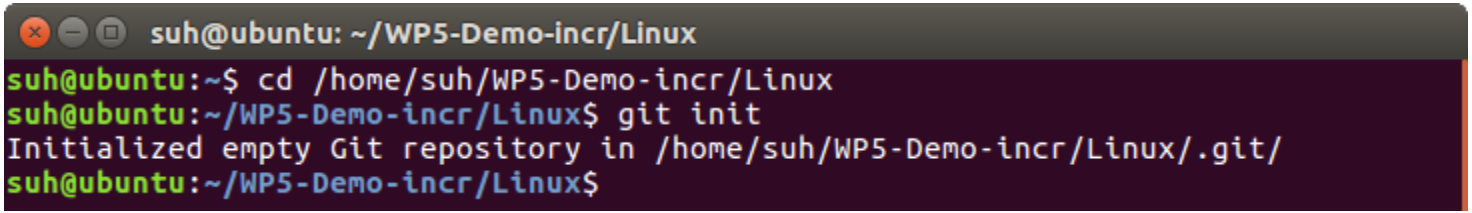
2. Preparation of Evolution History
PSS-CE

- Only necessary for illustration of capabilities here
- In practice:
  - Use of general commits to repository
  - Example: calling PSS-CE by a pre-commit hook
- Preparation consists of:
  - Create new repository
  - Commit Linux kernel version 4.10-rc2 (initial commit)
  - Define a new, but unused configuration option
  - Commit the change to the repository (change commit)
  - Extract both commits from the repository
- Extracted commits are input to incremental PSS-CE support



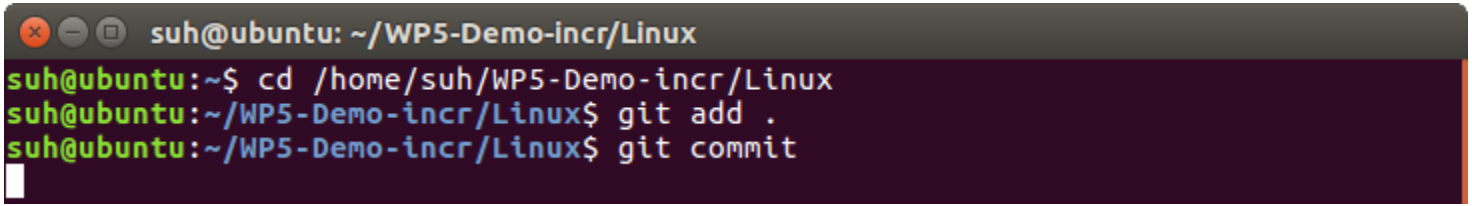
1. Create new repository

- Create empty directory at your favorite location
- Absolute path here: */home/suh/WP5-Demo-incr/Linux*
- Initialize Git repository:
  - Open Terminal
  - Navigate to */home/suh/WP5-Demo-incr/Linux*
  - Execute command *git init*



2. Commit Linux kernel version 4.10-rc2 (initial commit)

- Copy entire Linux kernel into */home/suh/WP5-Demo-incr/Linux*
- Commit the files to the Git repository:
  - Open Terminal
  - Navigate to */home/suh/WP5-Demo-incr/Linux*
  - Execute command *git add .*
  - Execute command *git commit*



3. Define a new, but unused configuration option

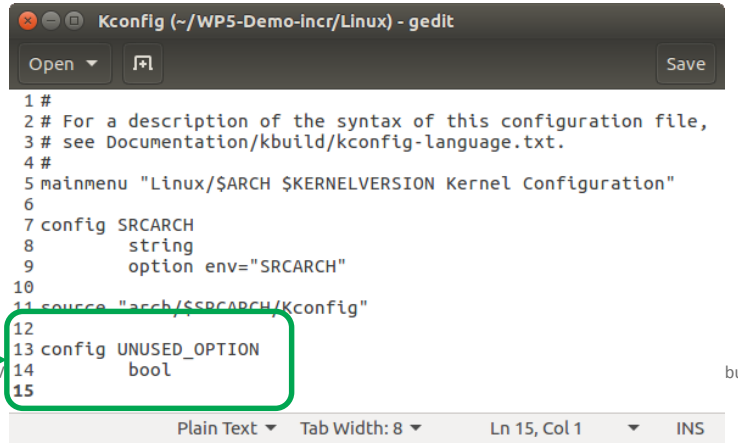
- Open file *Kconfig* in */home/suh/WP5-Demo-incr/Linux* in text editor
- Add the following lines at the end of the file:

  config UNUSED_OPTION
  → *bool*

- Save the changes and close text editor

No indentation

Single tab indentation



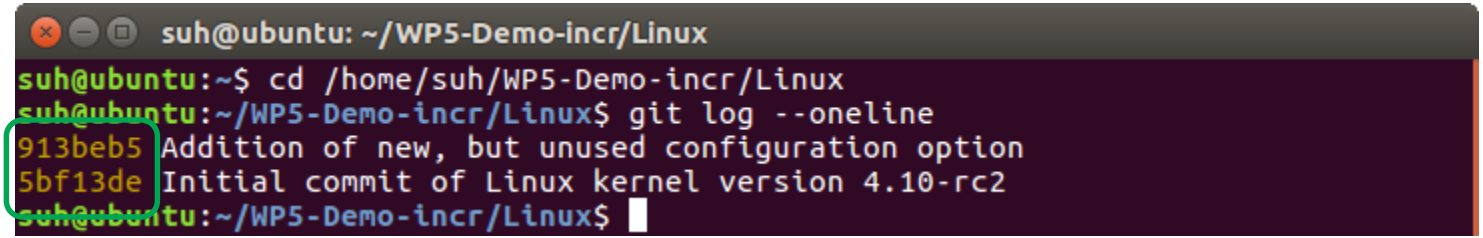
4. Commit the change to the repository (change commit)

- Open Terminal
- Navigate to */home/suh/WP5-Demo-incr/Linux*
- Execute command *git add .*
- Execute command *git commit*

```
suh@ubuntu: ~/WP5-Demo-incr/Linux
suh@ubuntu:~$ cd /home/suh/WP5-Demo-incr/Linux
suh@ubuntu:~/WP5-Demo-incr/Linux$ git add .
suh@ubuntu:~/WP5-Demo-incr/Linux$ git commit
[master 913beb5] Addition of new, but unused configuration option
 1 file changed, 4 insertions(+)
suh@ubuntu:~/WP5-Demo-incr/Linux$
```

5. Extract both commits from the repository (1/2)

- Identify the commit hashes:
  - Open Terminal
  - Navigate to */home/suh/WP5-Demo-incr/Linux*
  - Execute command *git log --oneline*
- Copy the commit hashes for actual extraction



5. Extract both commits from the repository (2/2)

- Extract the commits into individual files:
  - Execute command *git show -U100000 [COMMIT] > [FILE]*
  - *[COMMIT]* being one of the identified commit hashes
  - *[FILE]* being the file to print the commit content to; here:
    - Initial commit: */home/suh/WP5-Demo-incr/Linux-Diffs/Initial-git.diff*
    - Change commit: */home/suh/WP5-Demo-incr/Linux-Diffs/Change-git.diff*

```
suh@ubuntu: ~/WP5-Demo-incr/Linux
suh@ubuntu:~$ cd /home/suh/WP5-Demo-incr/Linux
suh@ubuntu:~/WP5-Demo-incr/Linux$ git log --oneline
913beb5 Addition of new, but unused configuration option
suh@ubuntu:~/WP5-Demo-incr/Linux$ git show -U100000 5bf13de > ../Linux-Diffs/Initial-git.diff
suh@ubuntu:~/WP5-Demo-incr/Linux$ git show -U100000 913beb5 > ../Linux-Diffs/Change-git.diff
suh@ubuntu:~/WP5-Demo-incr/Linux$
```

3. Baseline Creation for Incremental PSS-CE
PSS-CE

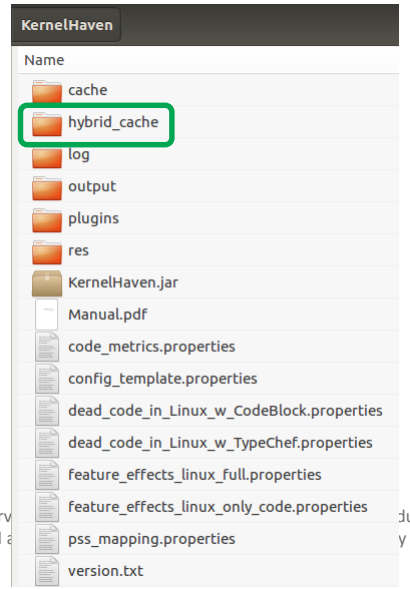
- Incremental variant requires initial models as baseline
- Baseline is mandatory input for:
  - Detecting changes (in detail)
  - Determining necessary (but potentially partial) re-analyses
- Baseline creation consists of:
  - Creation of required directory
  - Definition of PSS-CE setup for initial commit
  - Execution of PSS-CE using initial commit

3. Baseline Creation for Incremental PSS-CE
PSS-CE

1. Creation of required directory

- Create a new and empty directory in *KernelHaven/*
- Required by the incremental variant
- Storage of (partial) models for reuse

The directory name does not matter, but we need the path to and name of it on the next slide



2. Definition of PSS-CE setup for initial commit

- Create file *pss_ce.properties* in *KernelHaven/*
- Open that file with a text editor:
  - Specify required configuration options, in particular:
    - “source_tree” must point to an empty directory
    - “incremental.hybrid_cache.dir” must be the directory from previous slide
    - “incremental.input.source_tree_diff” must be */home/suh/WP5-Demo-incr/Linux-Diffs/Initial-git.diff*
  - Full examples: https://github.com/KernelHaven/ProblemSolutionSpaceCorrectorAnalysis
- Save the changes and close text editor

3. Execution of PSS-CE using initial commit

- Execute KernelHaven with the *pss_ce.properties*:
  - Open a terminal
  - Navigate to *KernelHaven/*
  - Execute command *java -Xmx12G -jar KernelHaven.jar pss_ce.properties*

```
suh@ubuntu: ~/WP5-Demo-incr/KernelHaven
suh@ubuntu:~/WP5-Demo-incr/KernelHaven$ java -Xmx12G -jar KernelHaven.jar pss-ce.properties
```

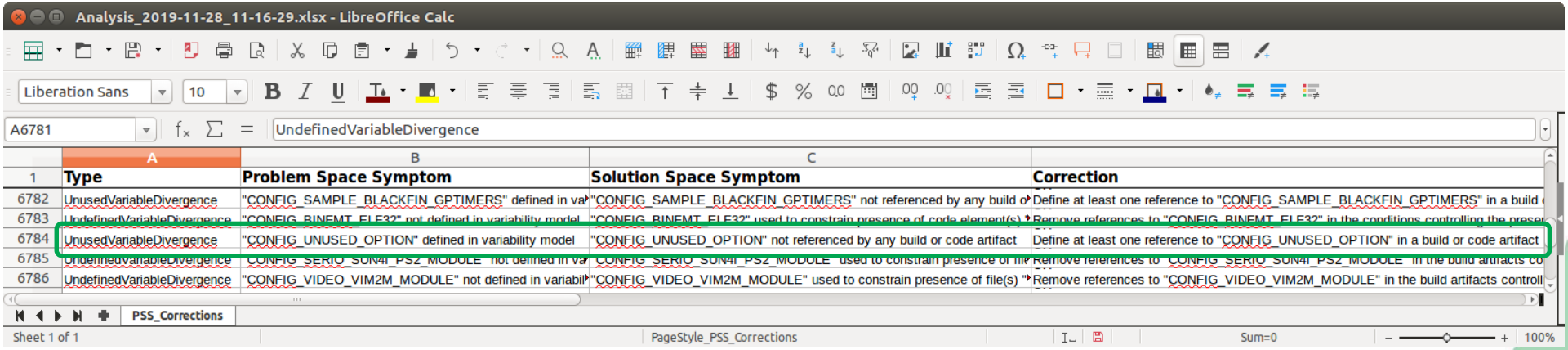
4. Incremental PSS-CE Analysis
PSS-CE

- Purpose:
  - Identify the changes introduced by change commit
  - Re-analyze only those changes
- This requires only:
  - Change property “incremental.input.source_tree_diff” in *pss_ce.properties* to */home/suh/WP5-Demo-incr/Linux-Diffs/Change-git.diff*
  - Execute KernelHaven with changed *pss_ce.properties* again

```
suh@ubuntu: ~/WP5-Demo-incr/KernelHaven
suh@ubuntu:~/WP5-Demo-incr/KernelHaven$ java -Xmx12G -jar KernelHaven.jar pss-ce.properties
```

4. Incremental PSS-CE Analysis
PSS-CE

- The result is located in */KernelHaven/output*
- Excel sheet now contains new, but unused configuration option as introduced by the change commit



Partners and Contact Details
PSS-CE

- **Partners involved**
  - University of Hildesheim (SUH), Germany

- **Contact Information**
  - Christian Kröher (kroeher@sse.uni-hildesheim.de)

- **Download**
  - KernelHaven: https://github.com/KernelHaven
  - PSS-Mapper: https://github.com/KernelHaven/ProblemSolutionSpaceMapperAnalysis
  - PSS-DivergenceDetector: https://github.com/KernelHaven/ProblemSolutionSpaceDivergenceDetectorAnalysis
  - PSS-DivergenceCorrector: https://github.com/KernelHaven/ProblemSolutionSpaceDivergenceCorrectorAnalysis