# SWC30 - Sprint 14
# Aspects of Automated Testing


Azeem Ahmad
azeem.ahmad@liu.se

Eduard Enoiu
eduard.paul.enoiu@mdh.se

Francisco Gomes
*gomesf@chalmers.se*

Kristian Sandahl
kristian.sandahl@liu.se

Ola Leifler
ola.leifler@liu.se

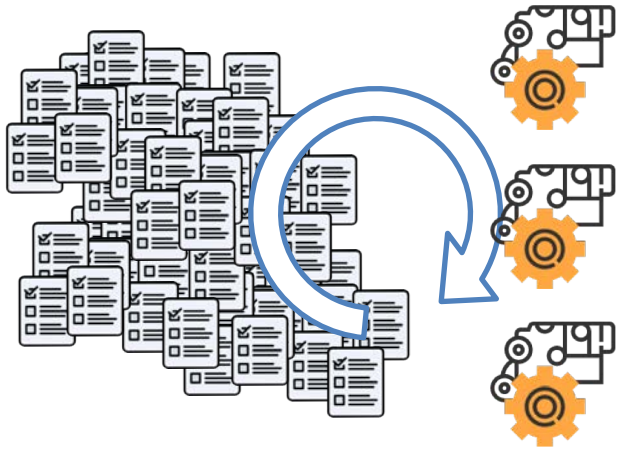
# The big picture

How do we **select them**?

Do we **need** that many?

How can we **reduce feedback cycles**?

**Test platform**

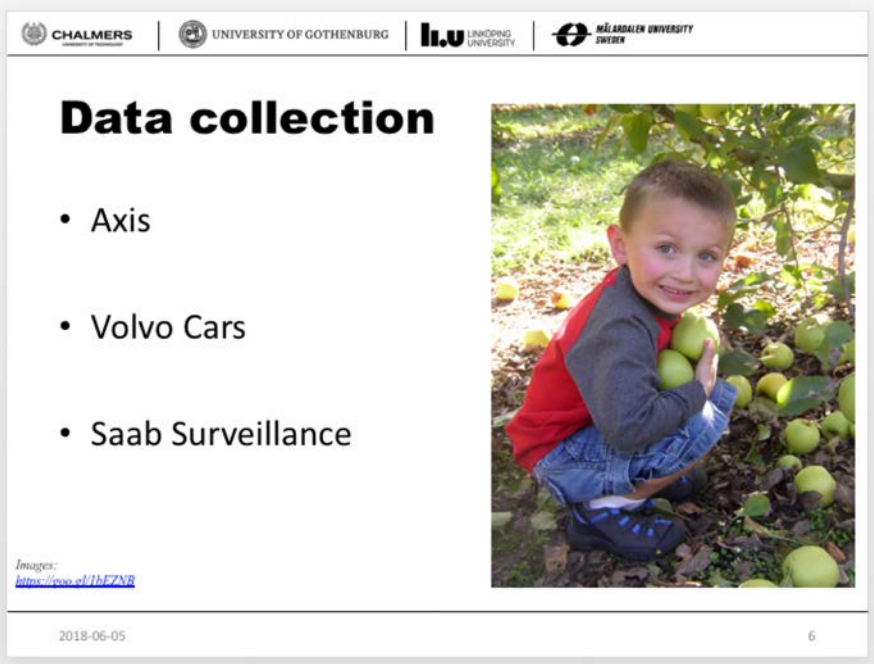


Automated system/integration tests

1000+ test cases

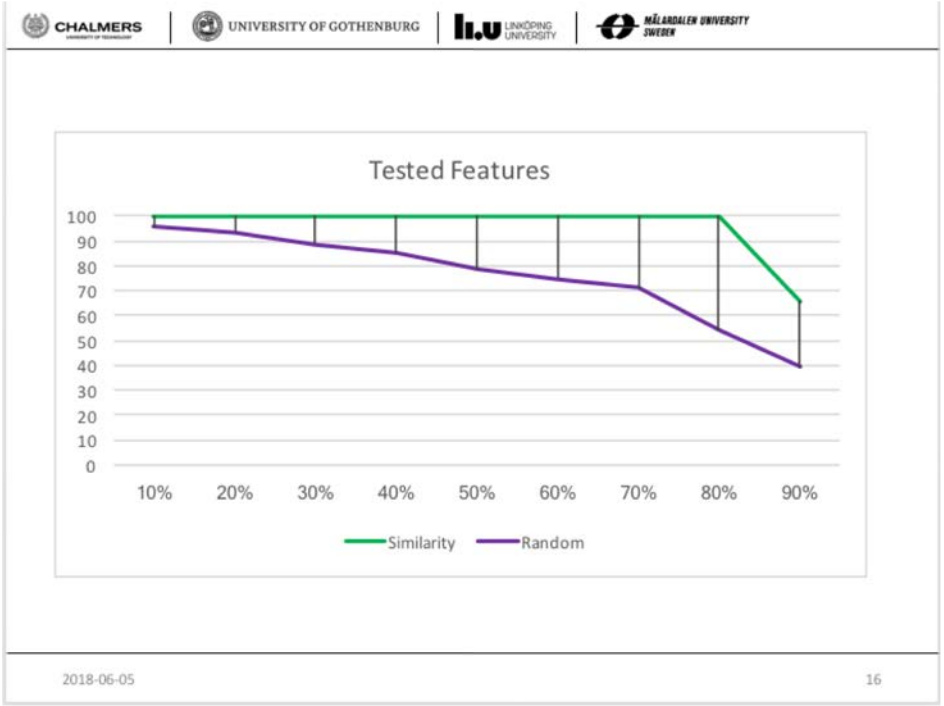
3+ hours

# Previously on SWC30 AAT3…




Preliminary studies with **test selection**…

… tools that yield a **diverse subset of tests.**



# Previously on SWC30 AAT3…

## AST 2018

13th IEEE/ACM International Workshop on Automation of Software Test

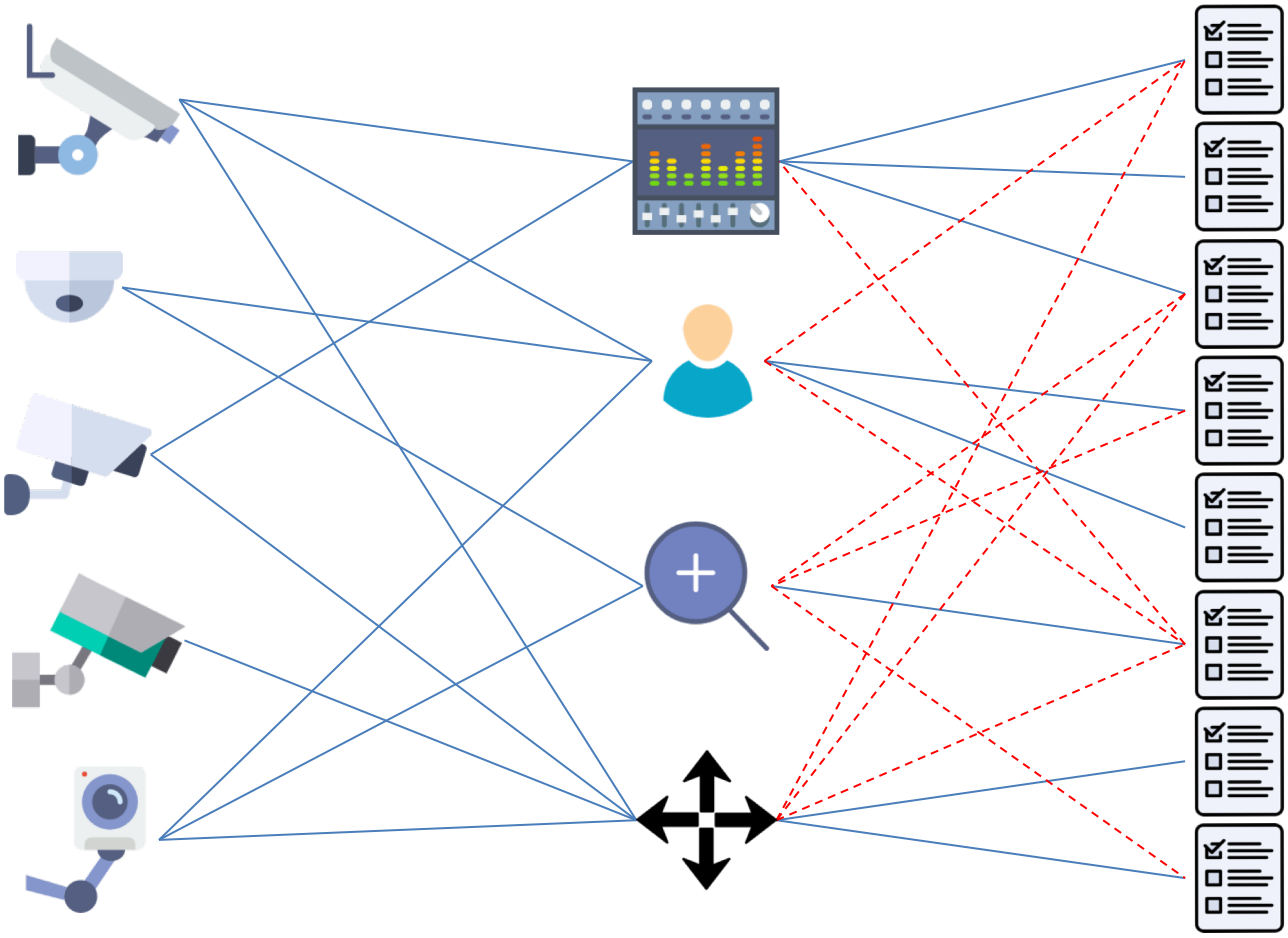
In Conjunction with ICSE 2018 (https://www.icse2018.org/)

Gothenburg, Sweden. May 28-29, 2018

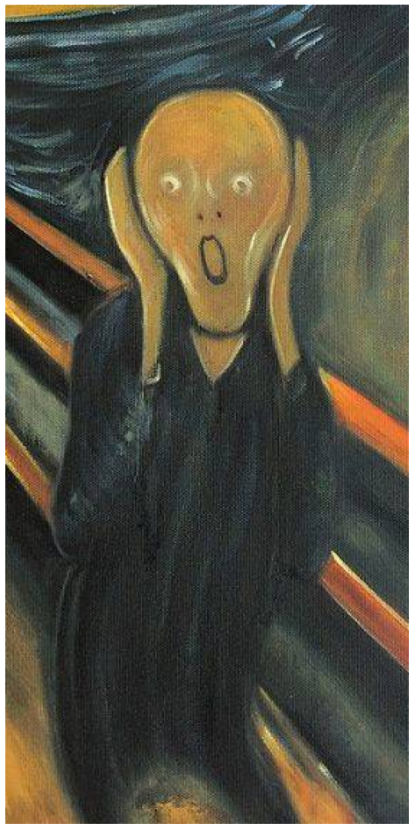
# What will I talk about?

- Results from including **another optimisation** strategy
- Test suite **minimization**
  - Automatically remove unnecessary redundancy.
- Goal: Optimize towards **resource allocation**
  - **E.g.: Hardware products**

# Minimizing test suites

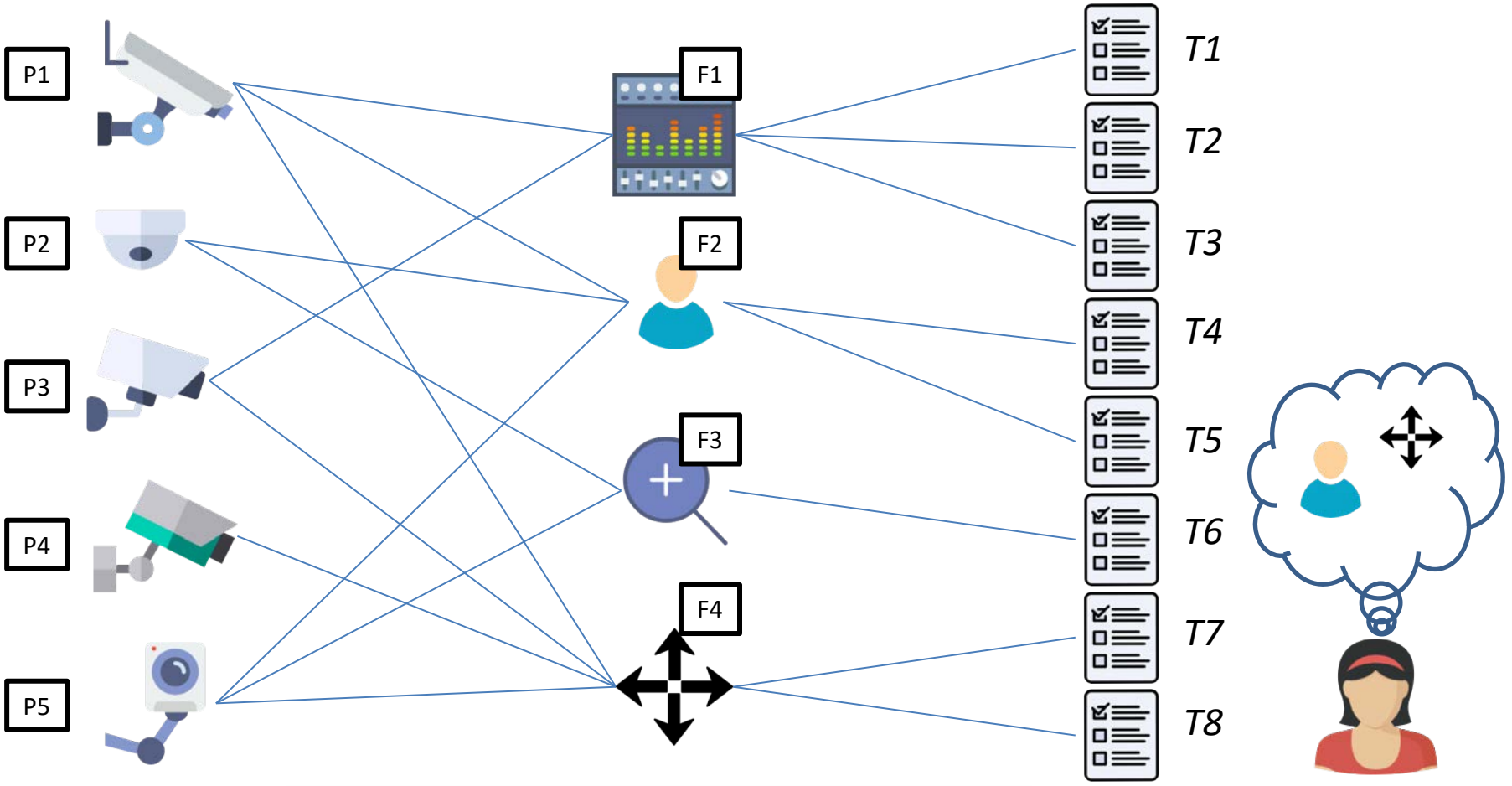


Products Features Test Cases



# Minimizing test suite

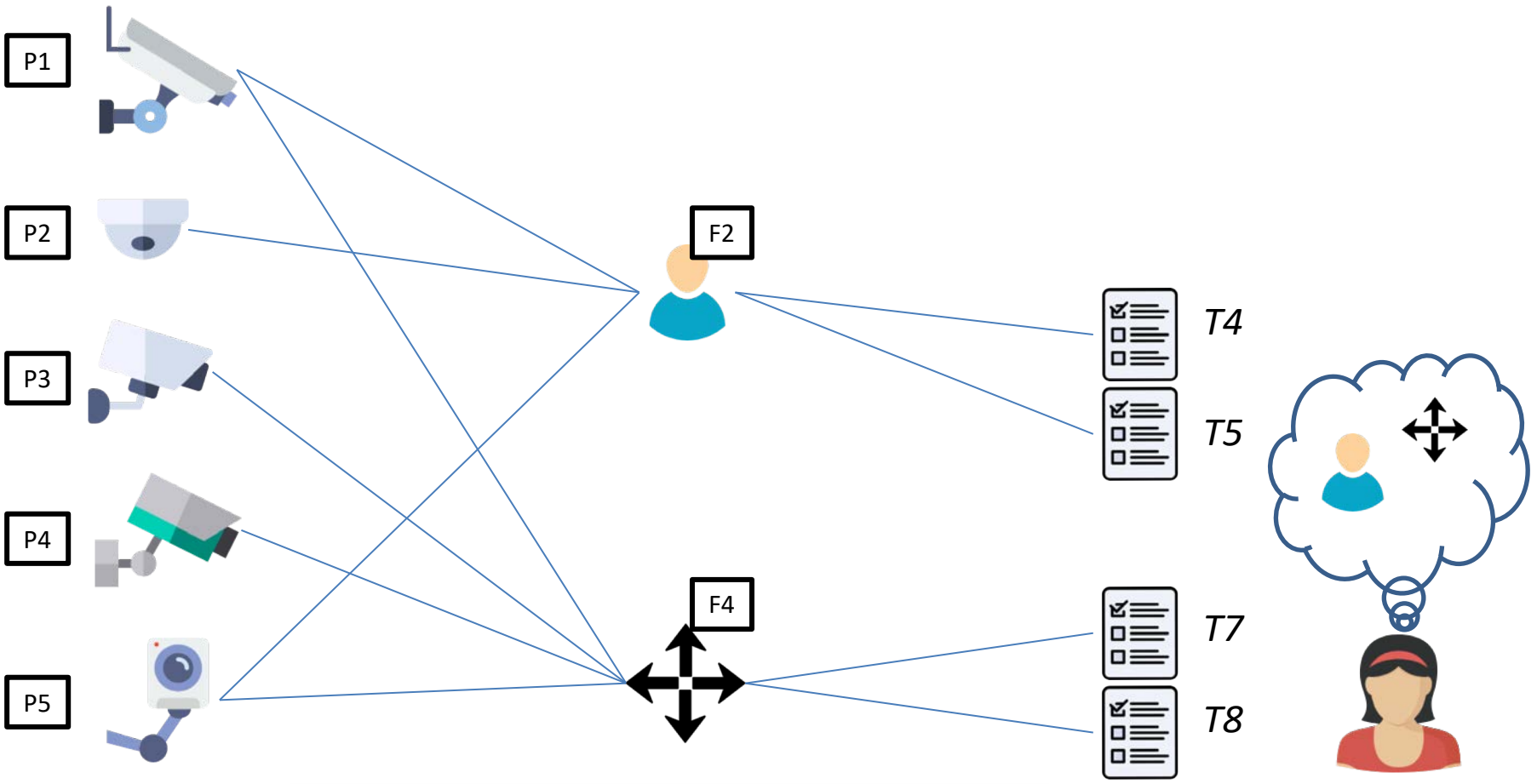
What's the **minimal #products** needed to test **Natalie's changes**?




Scenario 1

# Minimizing test suite

What's the **minimal #products** needed to test **Natalie's changes**?

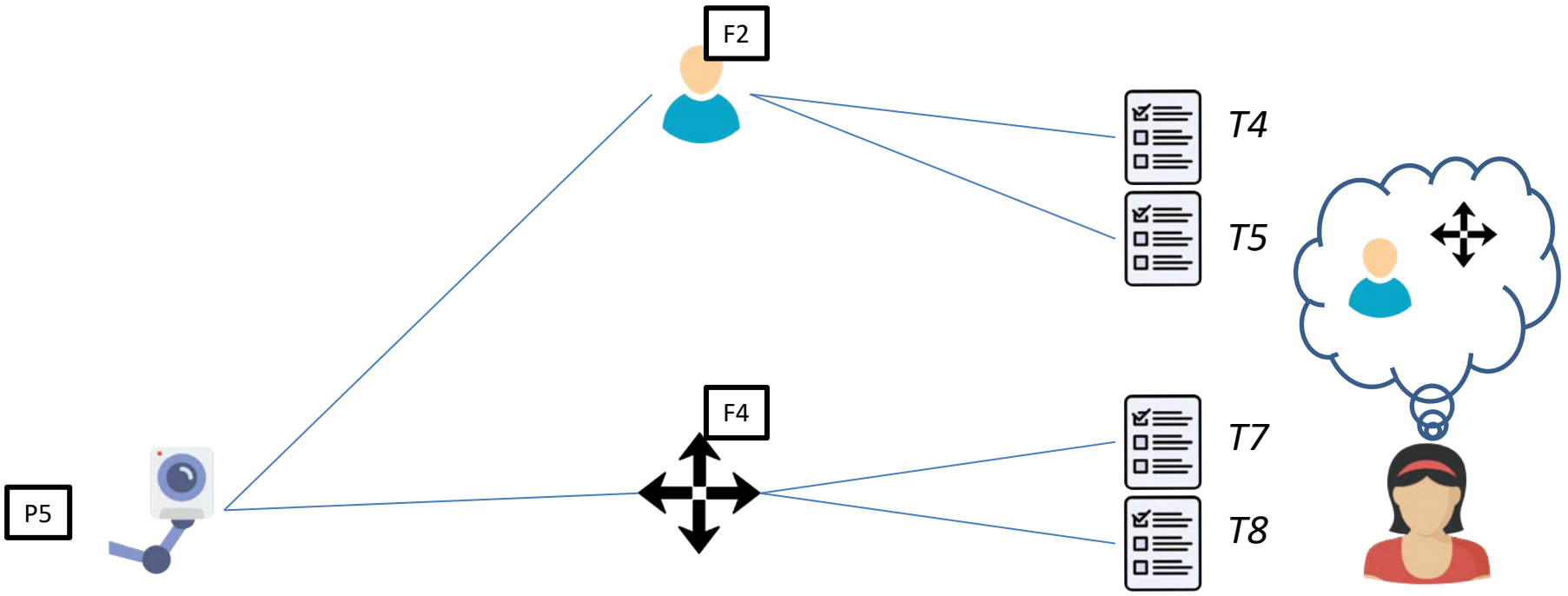



Scenario 2

# Minimizing test suite

What's the **minimal #products** needed to test **Natalie's changes**?

1 product: **P5**.




Scenario 3

# Minimizing test suites

- Can we have **fewer products**?
  - Resource allocation problem.

- Yes.
- Up to 90% less products

- Even with fewer product, **can we save time**?

- Yes.
- Best: 90% time reduction
- Worst: ~ 45% time reduction

- Is there an overall **redundancy on test** coverage?

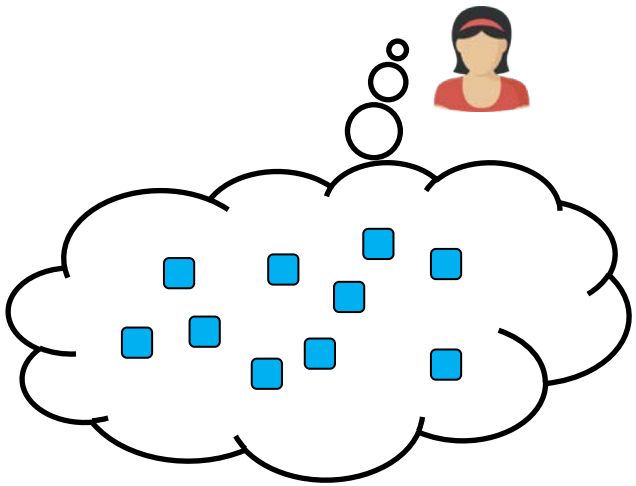
I'll explain later.

# What did we do?

1. Sample **N features** (randomly)
2. Get **all products AND tests** for those features
3. Calculate execution time

4. Find the **minimal** set of products
5. Find the minimal set of tests
6. Calculate execution time

7. Repeat
8. Get statistics



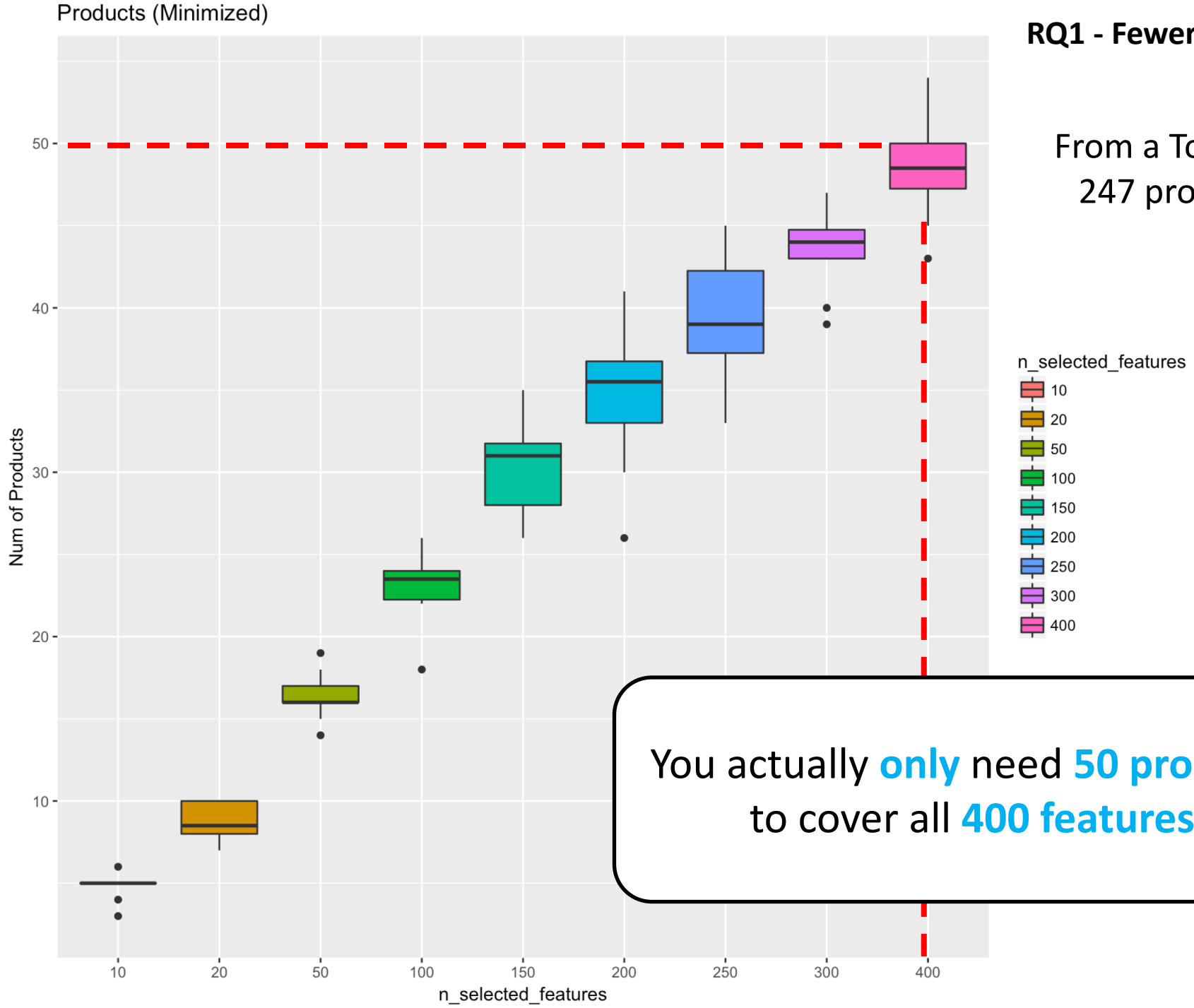
| Property | Total | 400 features (current) | 400 features (minimized) | 10 features (minimized) |
|---|---|---|---|---|
| Products | 249 | 247 | 50 | 5 |
| Num. of Tests | 44000+ | 3800+ | 700 | 19 |
| Time | 3 hours | 2.5 hours | 1.5 hours | 3.5 minutes |

# Before we begin…

- Couldn’t analyse on **failure** information.
- Redundancies were kept in the test suite.
  - **Data, as it is.**
- We used a **greedy-essential (GE) heuristic**.
- Max = 400 features

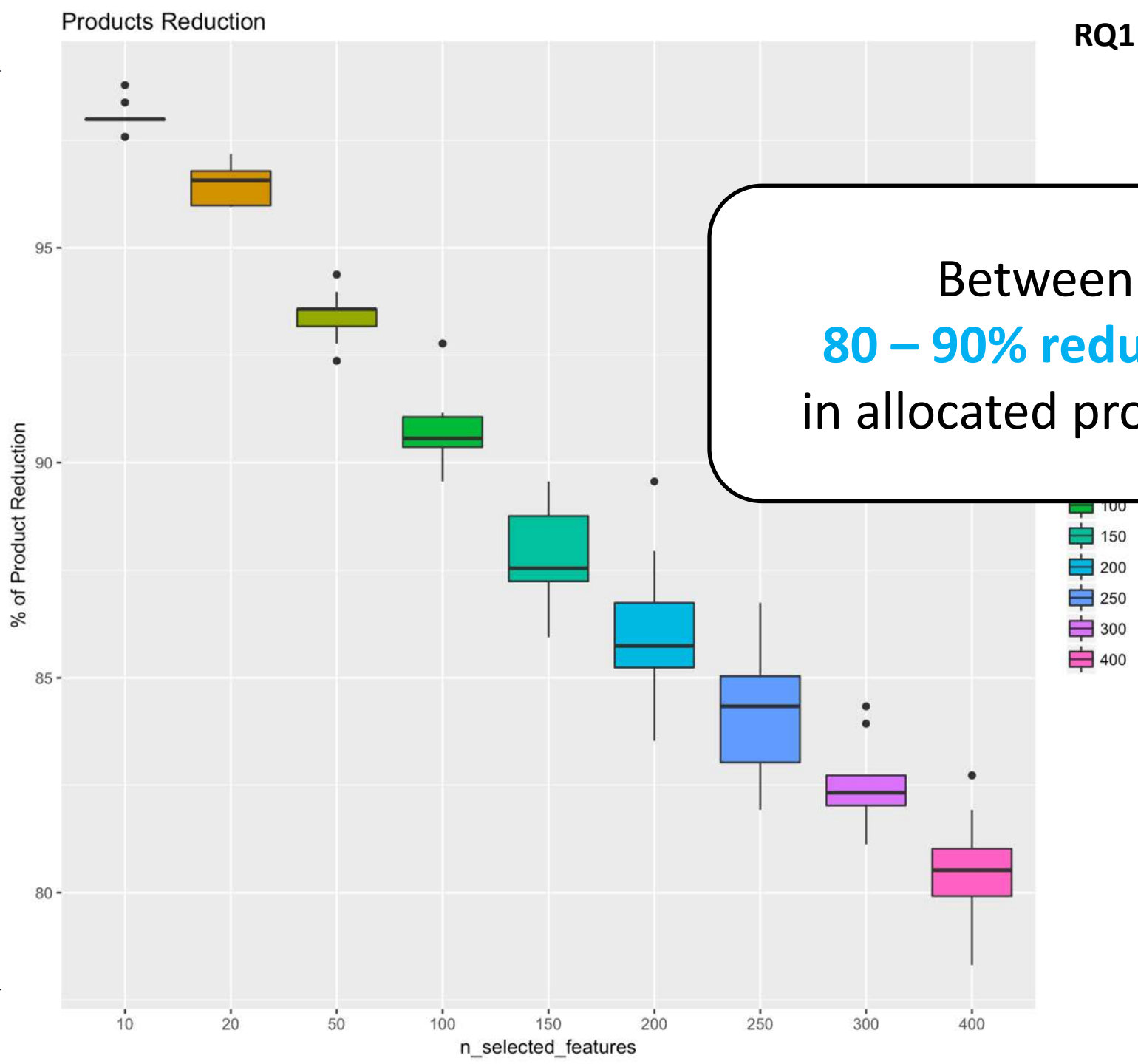
# RQ1 – Fewer Products?

# RQ1 - Fewer Products?




From a Total of:
247 products

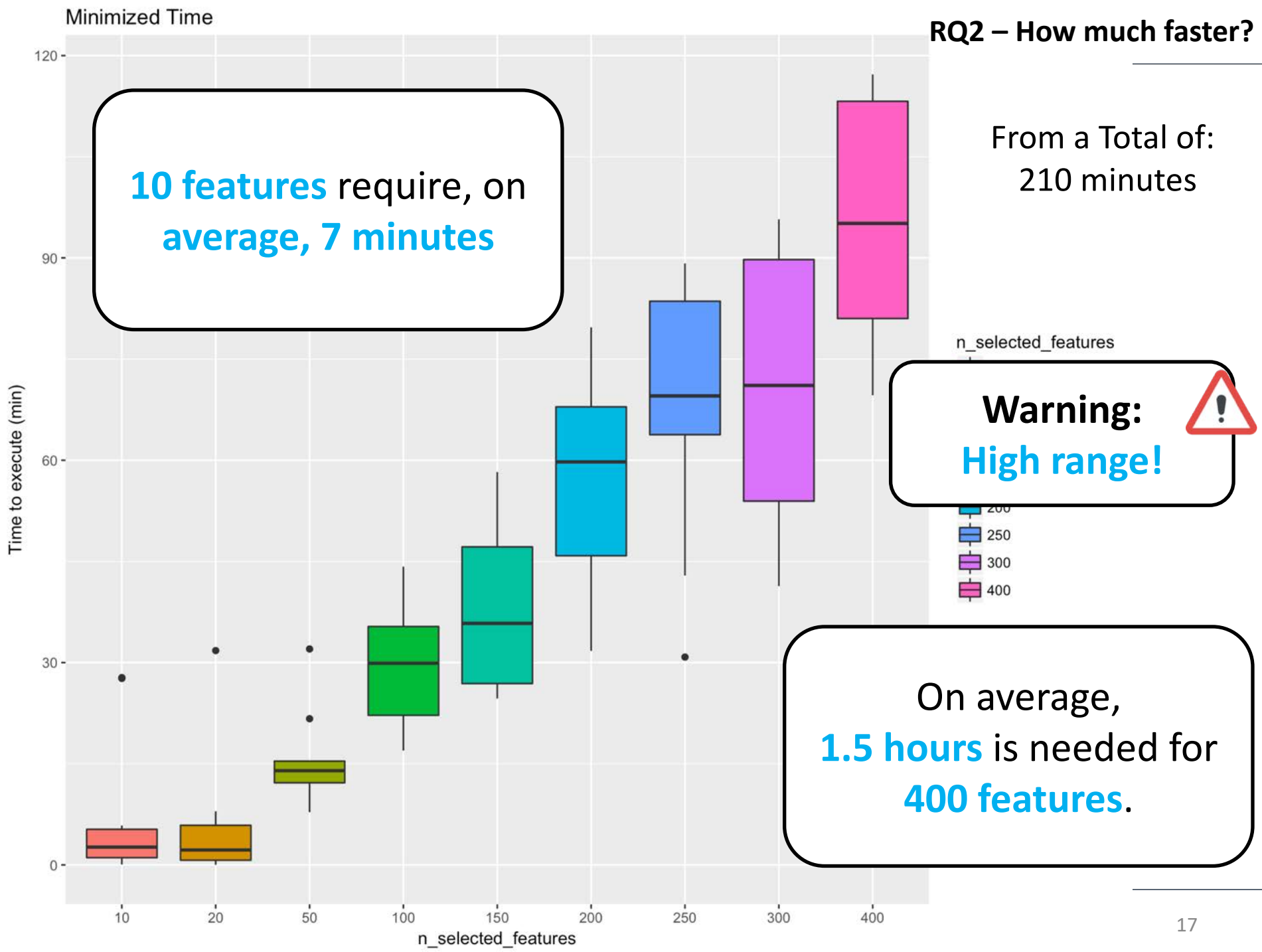
You actually **only** need **50 products** to cover all **400 features**




Between
**80 – 90% reduction**
in allocated products

# RQ2 – What about feedback time?

## RQ2 – How much faster?




**10 features** require, on **average, 7 minutes**

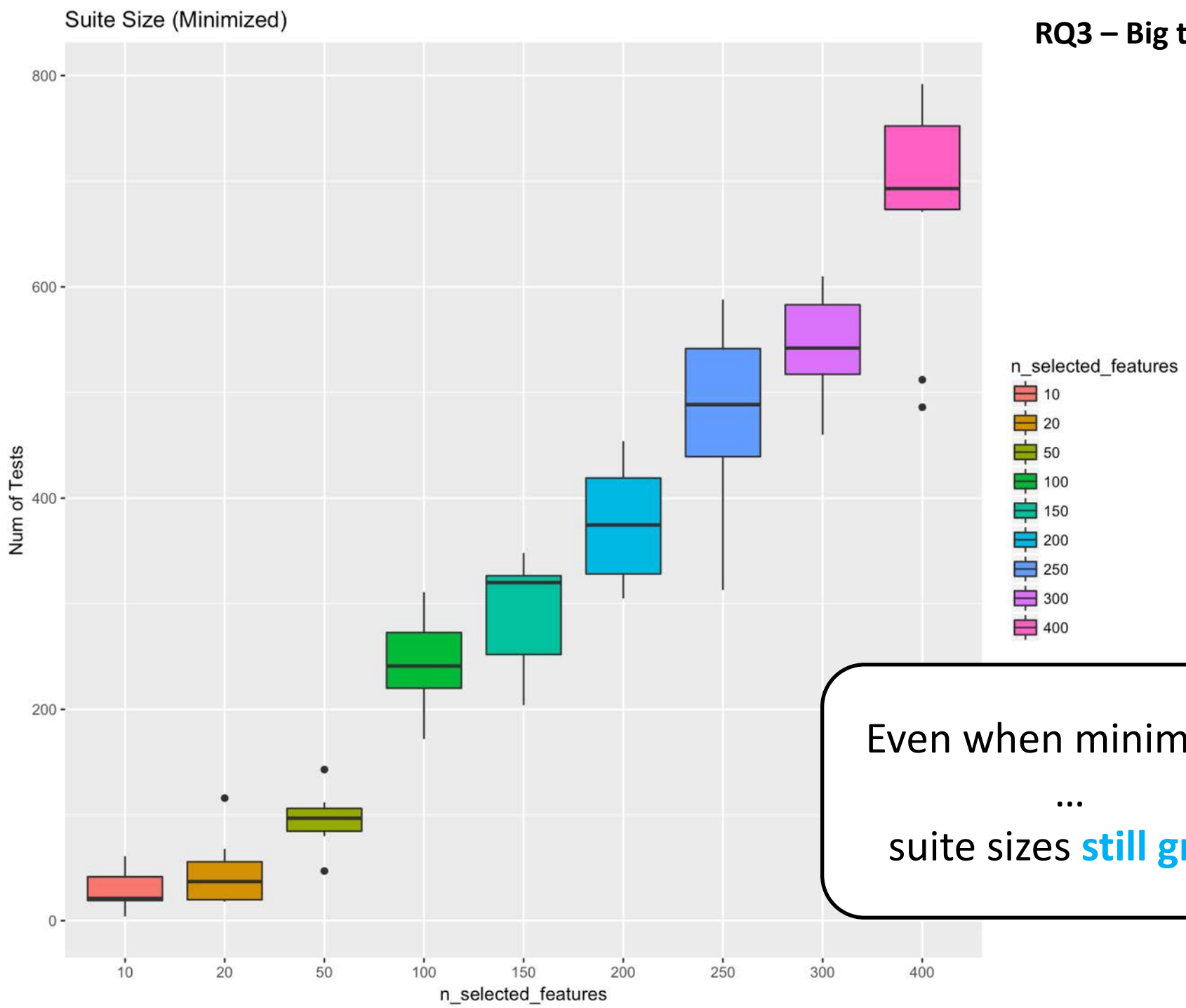
From a Total of:
210 minutes

**Warning:**
**High range!**

On average,
**1.5 hours** is needed for
**400 features**.

# RQ3 – But how many tests?

## RQ3 – Big test suites?



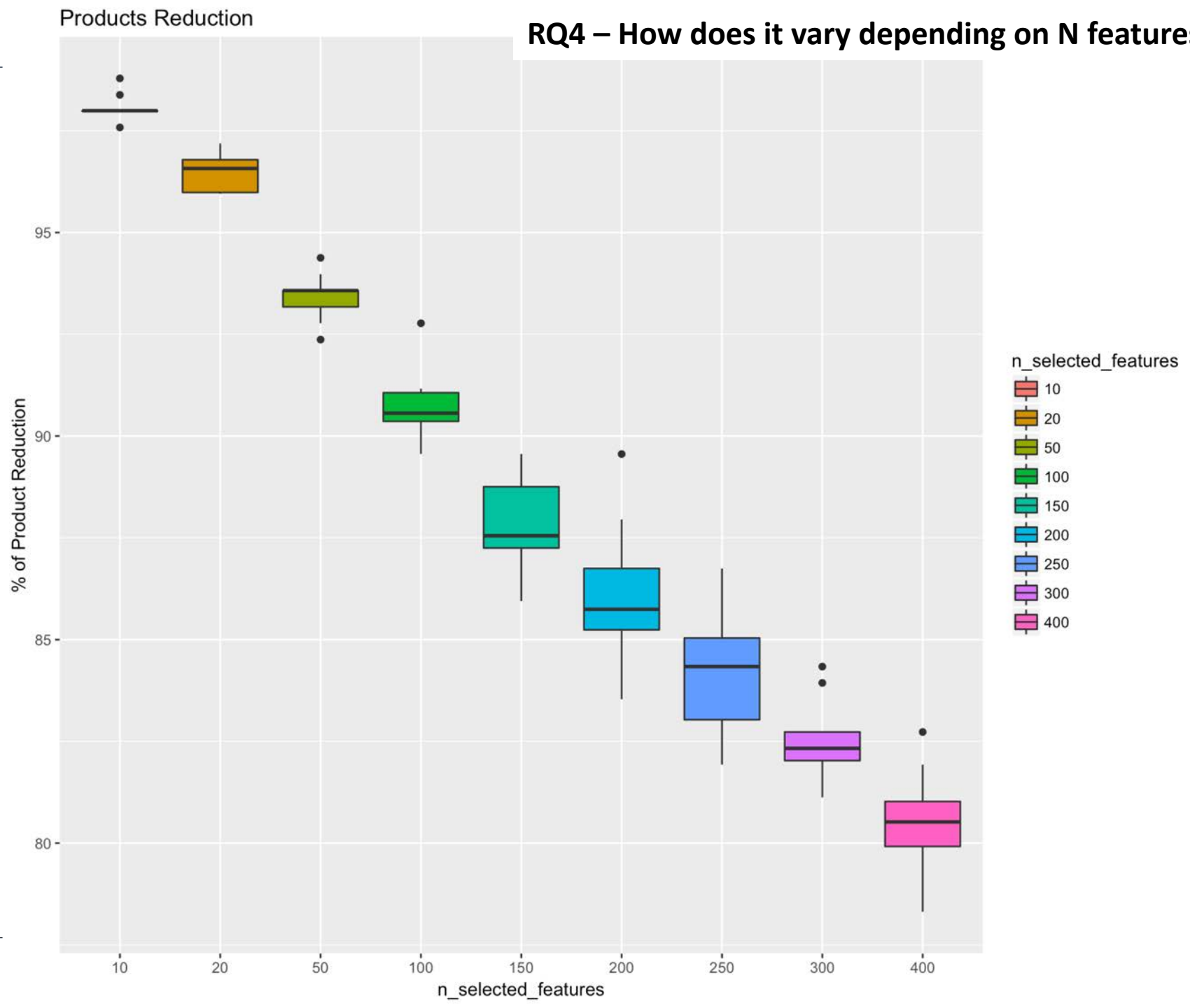

Even when minimized,

…

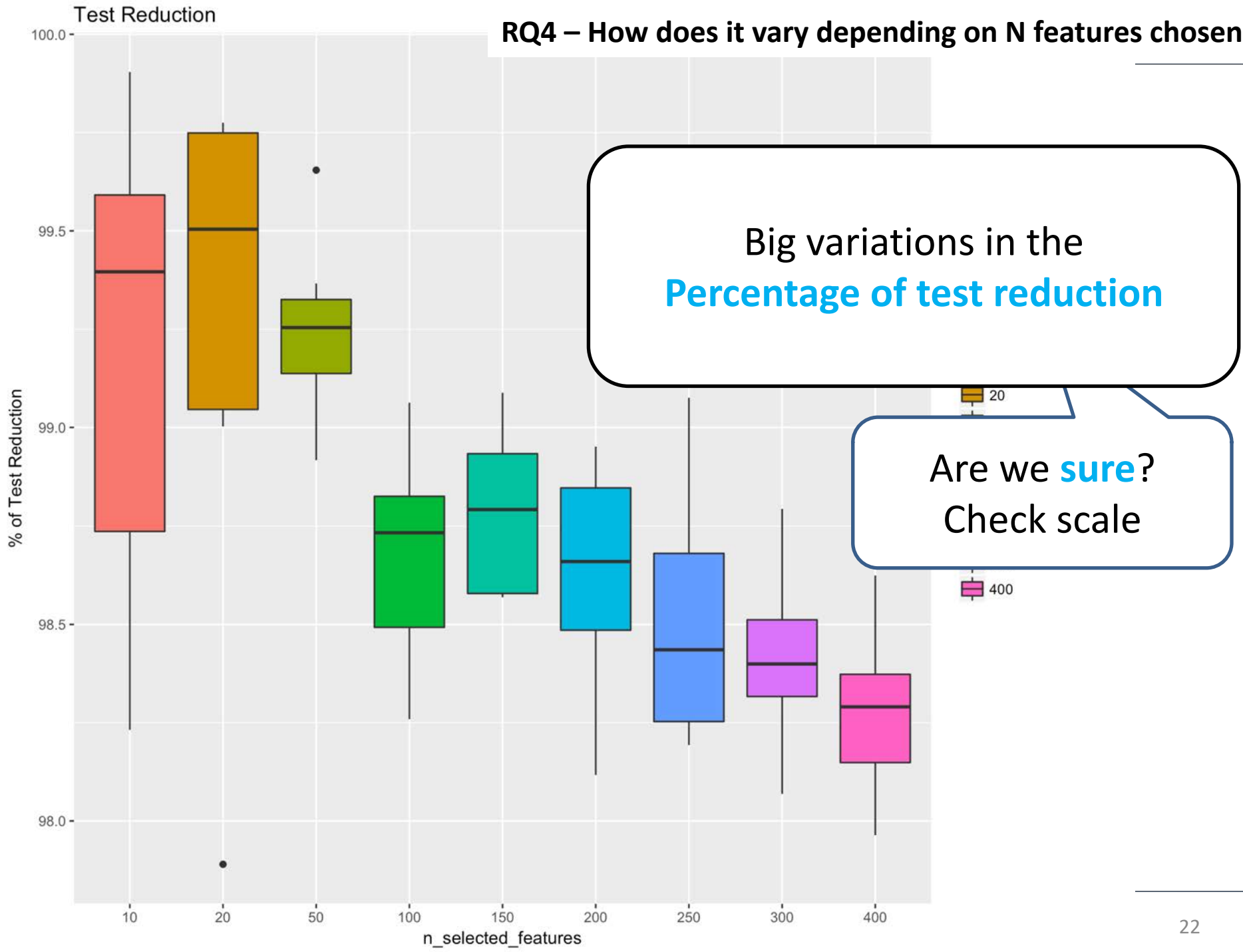
suite sizes **still grow**

# RQ4 – How does it vary depending on N features chosen?

# RQ4 – How does it vary depending on N features chosen?



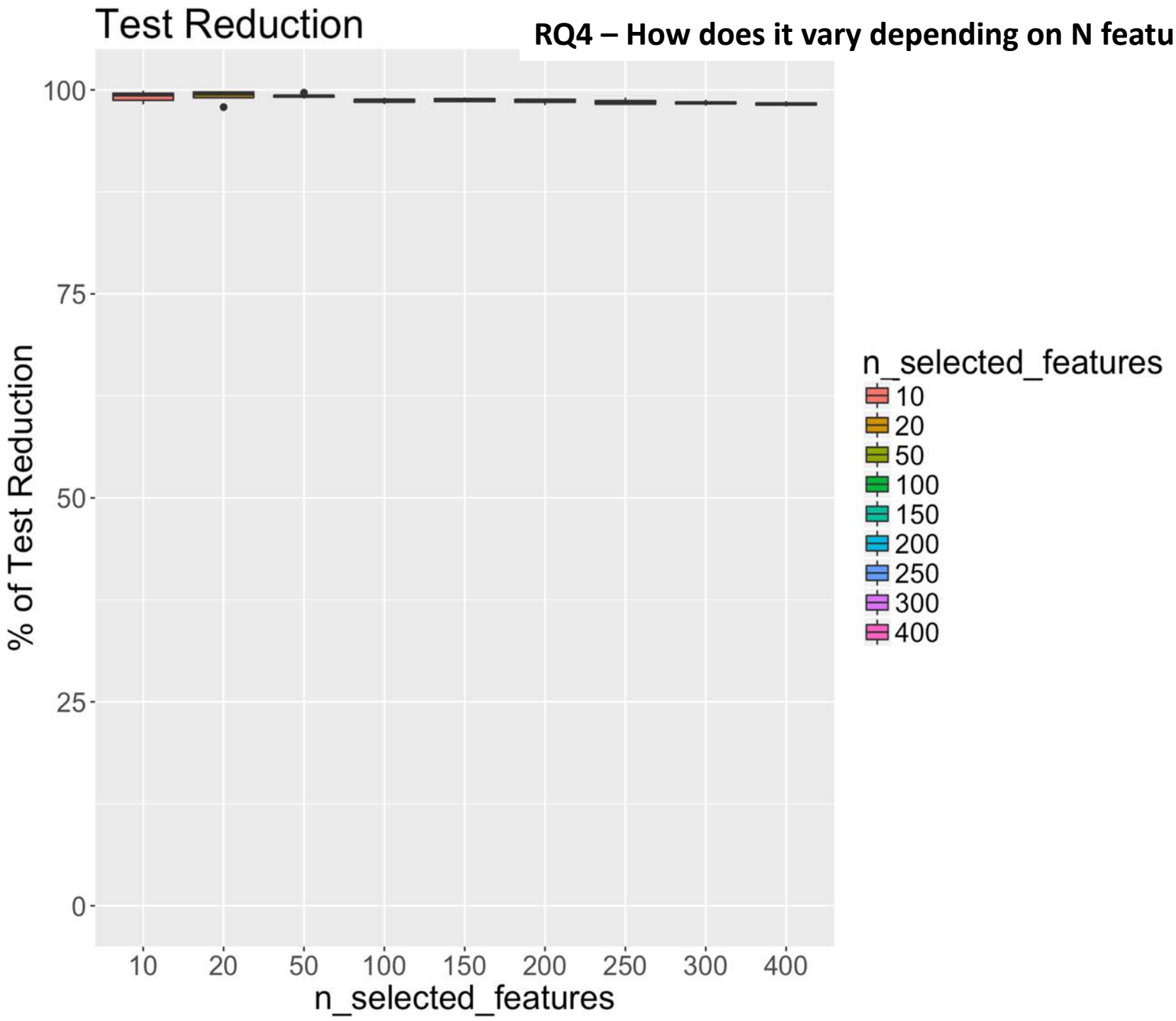
## RQ4 – How does it vary depending on N features chosen?

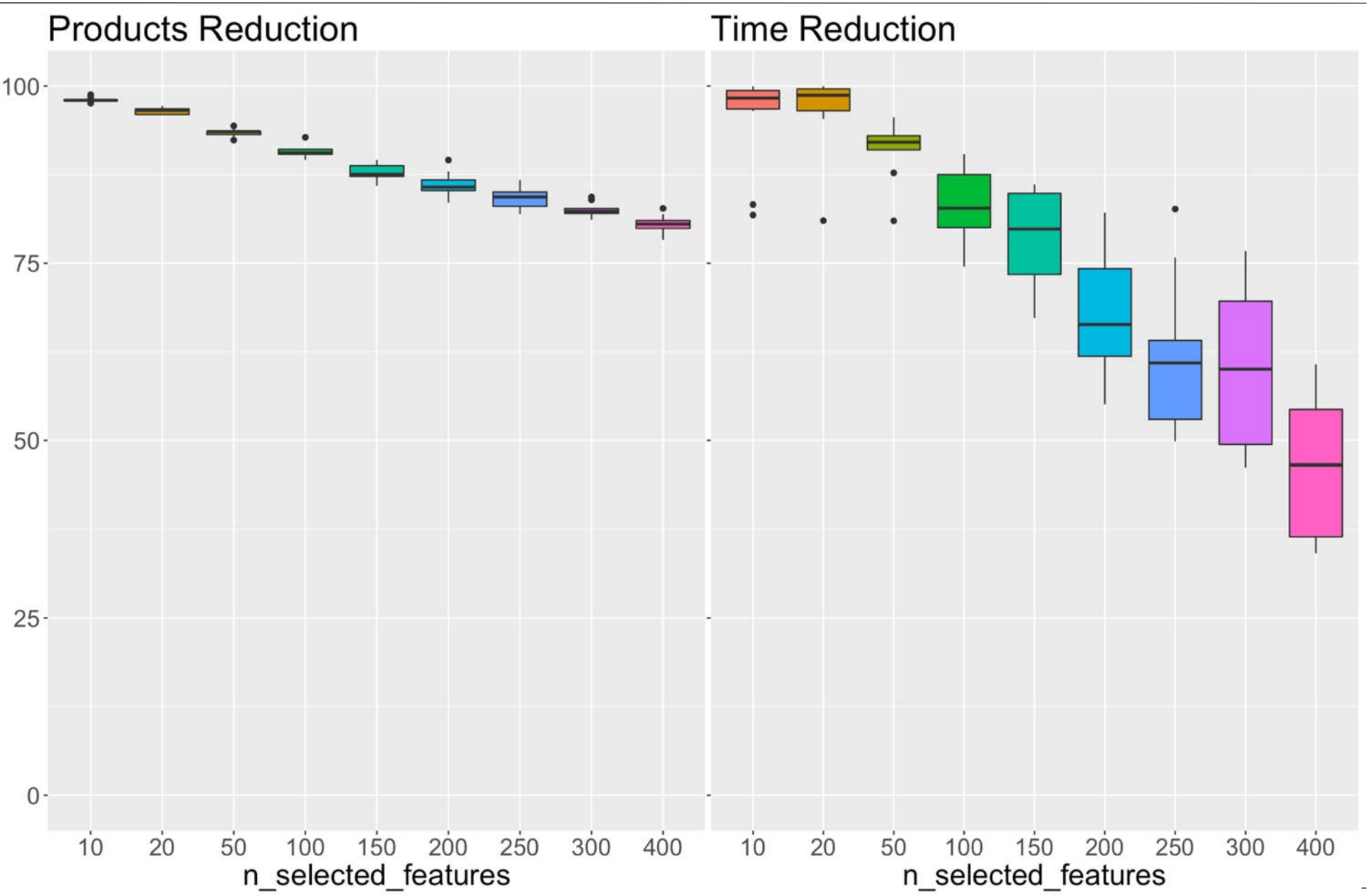


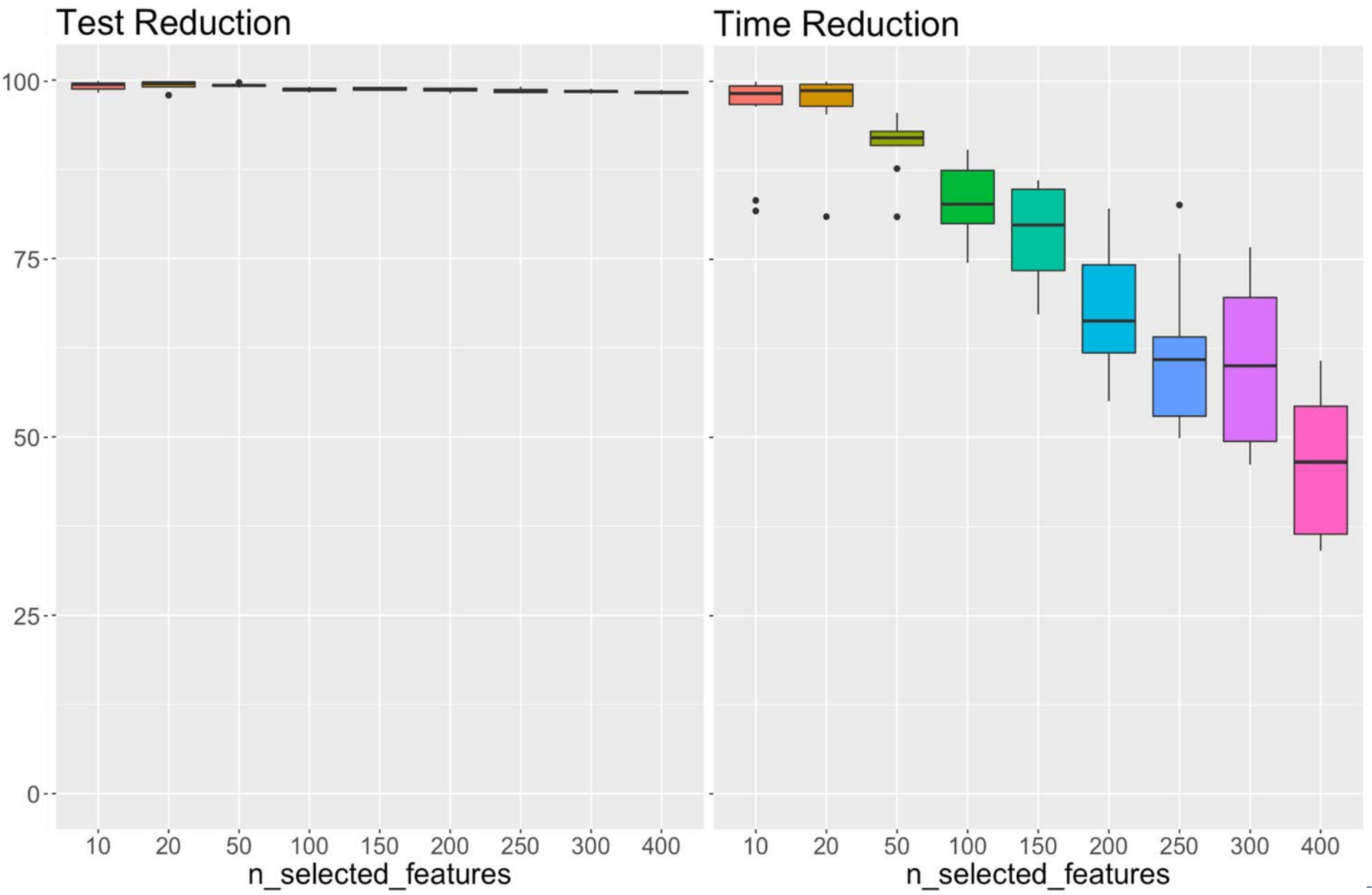

Big variations in the
**Percentage of test reduction**

Are we **sure**?
Check scale

**RQ4 – How does it vary depending on N features chosen?**




Products Reduction
Time Reduction
100
75
50
25
0
10
20
50
100
150
200
250
300
400
n_selected_features


Test Reduction
Time Reduction
100
75
50
25
0
10
20
50
100
150
200
250
300
400
n_selected_features
n_selected_features

# Take away

- Minimization yields **cost-effective testing**:
  - Time to execute the technique can be prohibitive
- Provides **controllability** to testers
  - Selective of features for prohibitive combinations
  - Human-in-the-loop is important
- **Holistic view** of your test dependencies

# Technology transfer

- **Traceability** is required.
- **Some** governance on how tests are designed
- Collect and store **test data**:
  - Time, failures, coverage.
- Preferably, **automated** integration/system level.

# SWC30 - Sprint 14
# Aspects of Automated Testing


Azeem Ahmad
azeem.ahmad@liu.se

Eduard Enoiu
eduard.paul.enoiu@mdh.se

Francisco Gomes
*gomesf@chalmers.se*

Kristian Sandahl
kristian.sandahl@liu.se

Ola Leifler
ola.leifler@liu.se