

SELF-CONTAINED AUTOMATED TESTS AND HOW TO REMOTE DEBUG THEM

TÕNU JAARMA

 **playtech**
SOURCE OF SUCCESS



London
Stock Exchange

LISTED

PREMIUM

HISTORY

- More than 1000 tests
- Some of them are over 10 years old
- People who designed these tests are gone
- ~100 people who might be somehow involved with those tests
- Some of them are legacy features
- We do not know how important those tests are
- Are those flows even used in production anymore?

THE IDEA OF SELF-CONTAINED AUTOMATED TESTS

Independent

- All the flows need to be converted to API calls
- API calls chained together
- All API request parameters are predefined or taken from previous responses
- No DB
- Every test must have all the needed data for its own flows

THE IDEA OF SELF-CONTAINED AUTOMATED TESTS

Independent

Parallely runnable

- TestNG will do the magic

THE IDEA OF SELF-CONTAINED AUTOMATED TESTS

Independent

Parallely runnable

N tests can be run to N environments

- Tests can be runned at any time with any order
- UI for better management

THE IDEA OF SELF-CONTAINED AUTOMATED TESTS

Independent

Parallely runnable

N tests can be run to N environments

Everyone can use this platform simultaneously

THE IDEA OF SELF-CONTAINED AUTOMATED TESTS

Independent

Parallely runnable

N tests can be run to N environments

Everyone can use this platform simultaneously



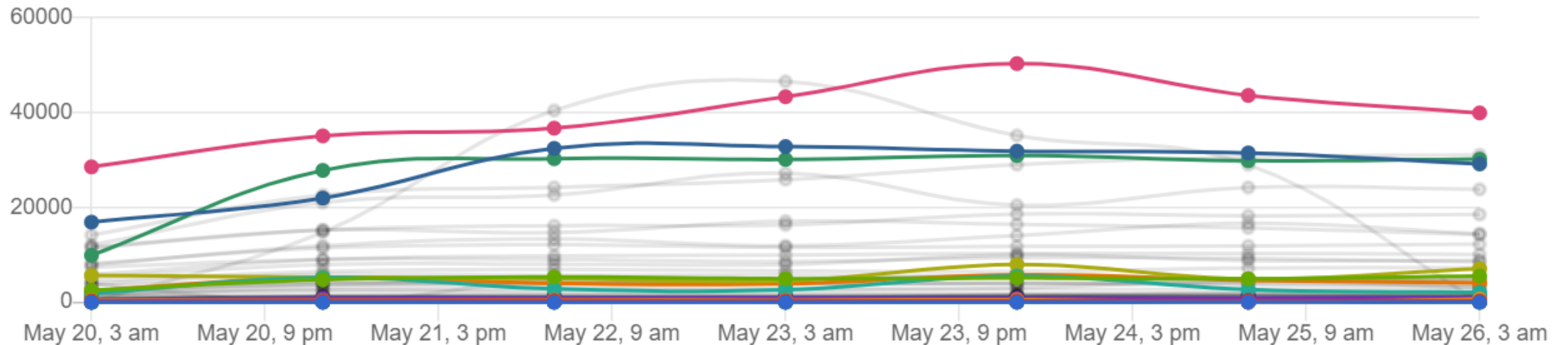
Good regular Autotesting Framework

WHAT ELSE DO WE NEED TO KNOW

- What flows are actually popular in production?
- How much our features are used?
- Is our feature usage trend increasing?
- In case of failures can we deploy our product to production?

WHY IT IS GOOD TO QUERY THE PRODUCTION DATABASE EVERY NIGHT

Product Operations Management System - POMS



User Account Management > Player account service > Account lifecycle > Number of accounts that have login or deposit (are active in this period)

Number of accounts that have login or deposit (are active in this period)

WHY IT IS GOOD TO QUERY THE PRODUCTION DATABASE EVERY NIGHT

- We know what flows are actually popular in production
- We know how much our features are used
- Feature usage trend is automatically visible
- In case of failures we know where can we safely deploy our product

Constantly proving its relevance

THE IDEA OF SELF-CONTAINED AUTOMATED TESTS

Independent

Parallely runnable

N tests can be run to N environments

Everyone can use this platform simultaneously

Constantly proving its relevance

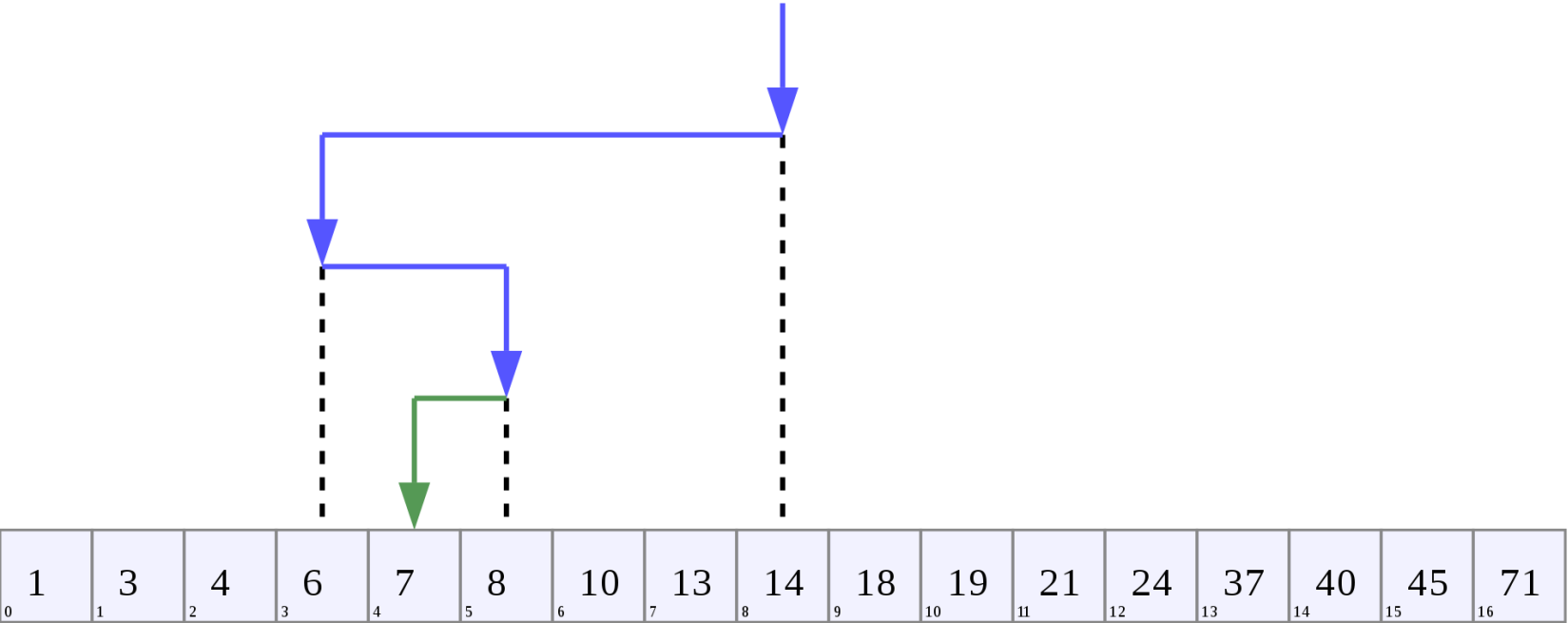


Up-to-date Autotesting Framework

WHAT IF OUR AUTOMATED TESTS ARE FAILING

- We need to know
 - If tests are broken?
 - If our product has defect?
 - When did this happen?
 - Who is going to fix that?
- What if they are failing during night?

WHY USE A BINARY SEARCH ALGORITHM TO FIND BUGS



Finding the broken package by itself

THE IDEA OF SELF-CONTAINED AUTOMATED TESTS

Independent

Parallely runnable

N tests can be run to N environments

Everyone can use this platform simultaneously

Constantly proving its relevance

Finding the broken package by itself



SELF-CONTAINED AUTOMATED TESTS

THE IDEA OF SELF-CONTAINED AUTOMATED TESTS

Independent

Parallely runnable

N tests can be run to N environments

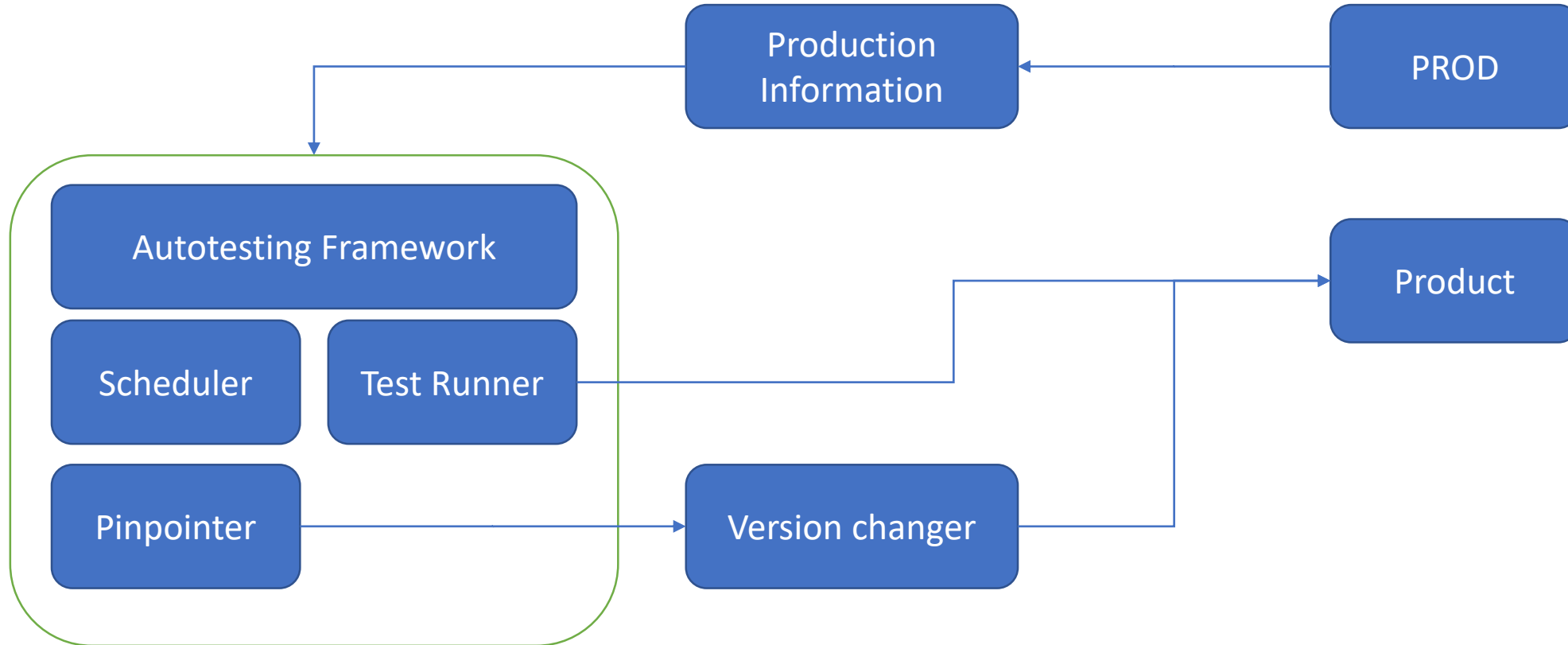
Everyone can use this platform simultaneously

Constantly proving its relevance

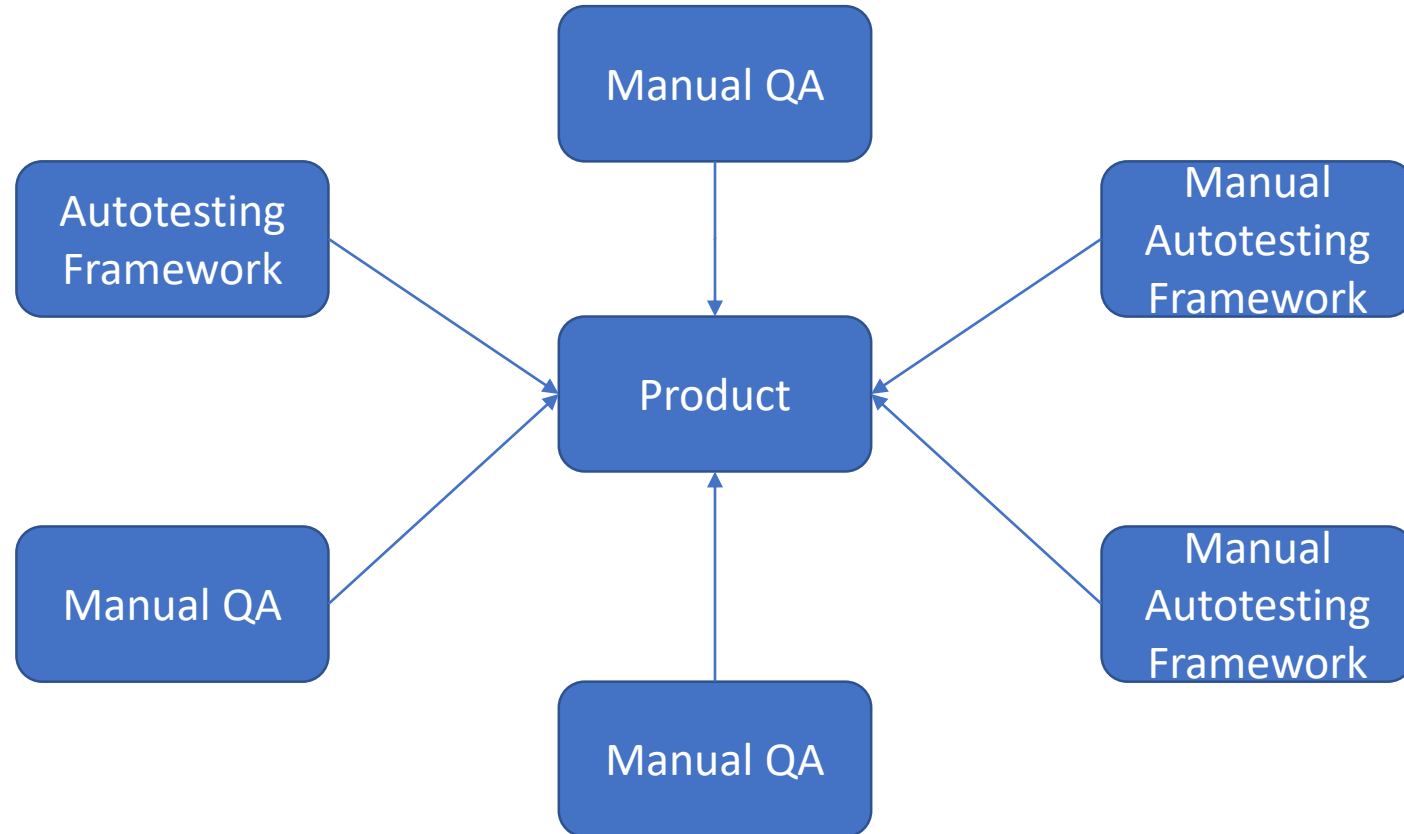
Finding the broken package by itself

Assignes problems to relevant people

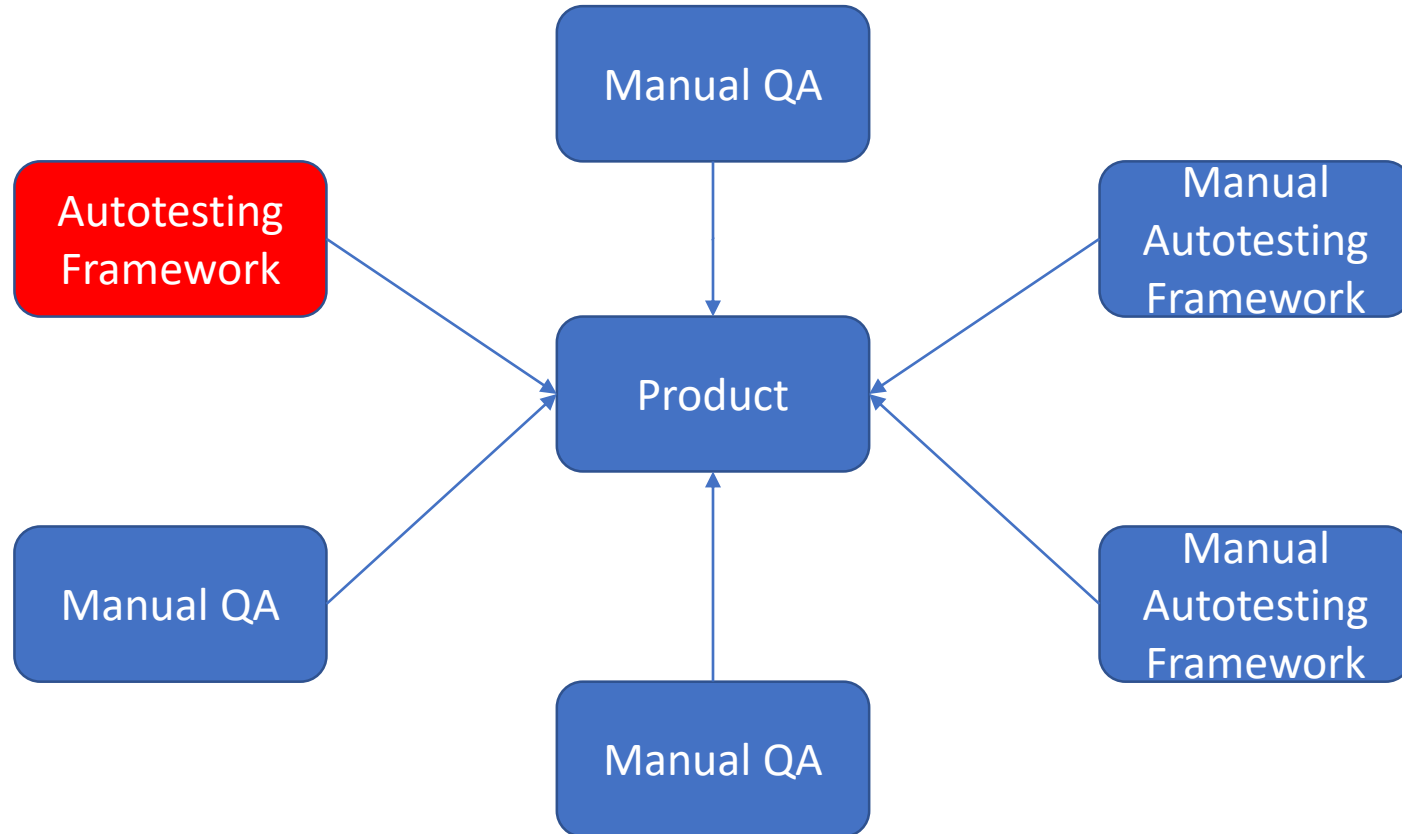
HOW IT ALL LOOKS TOGETHER



EVERYDAY LIFE



IT WORKS ON MY MACHINE



*HOW TO REMOTELY DEBUG PROBLEMS SO THAT OTHERS
CAN USE THE SAME ENVIRONMENT MEANWHILE*



QA.pptx



23:13
26.05.2019





THANK YOU