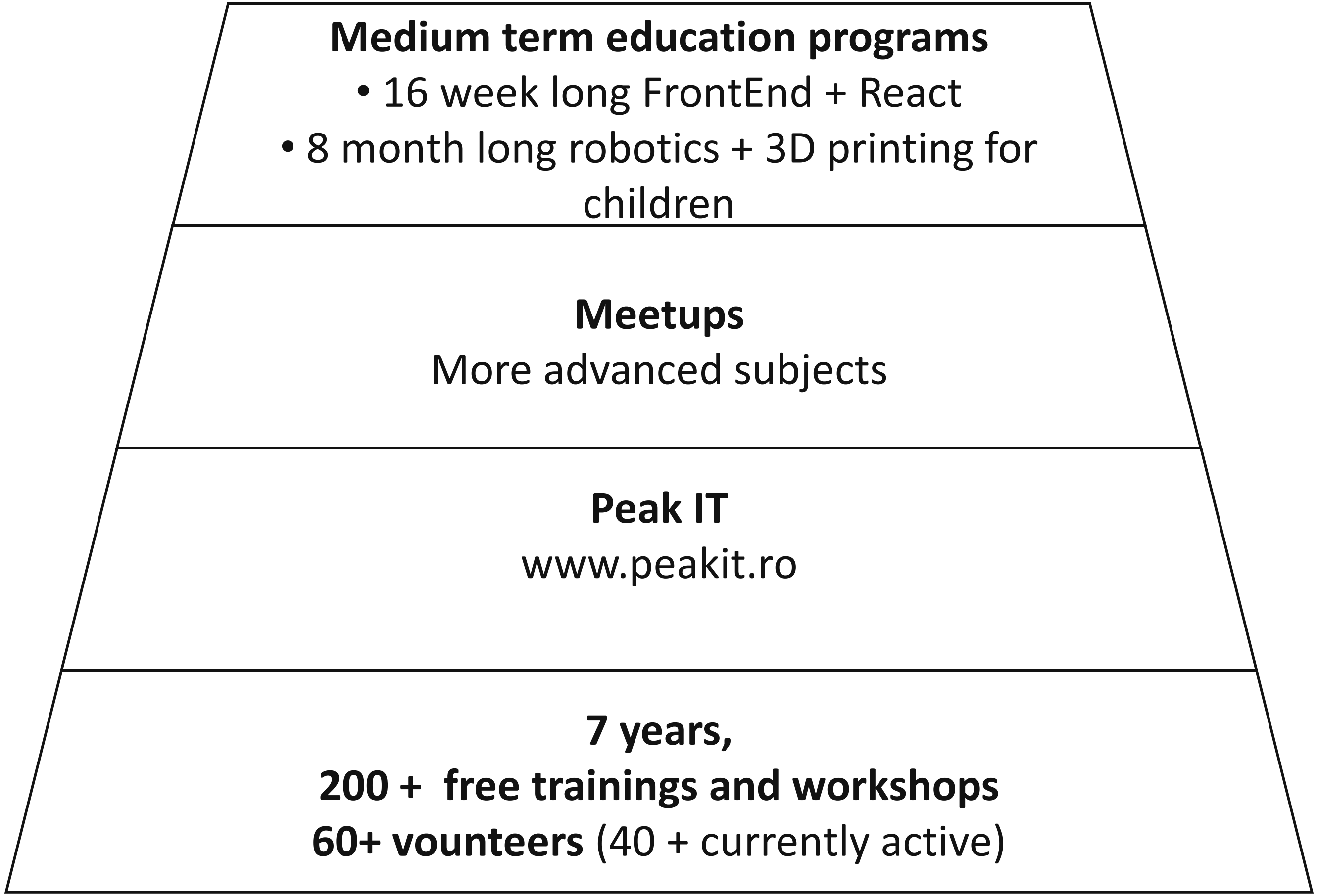
AgileHub



Medium term education programs

- 16 week long FrontEnd + React
- 8 month long robotics + 3D printing for children

Meetups

More advanced subjects

Peak IT

www.peakit.ro

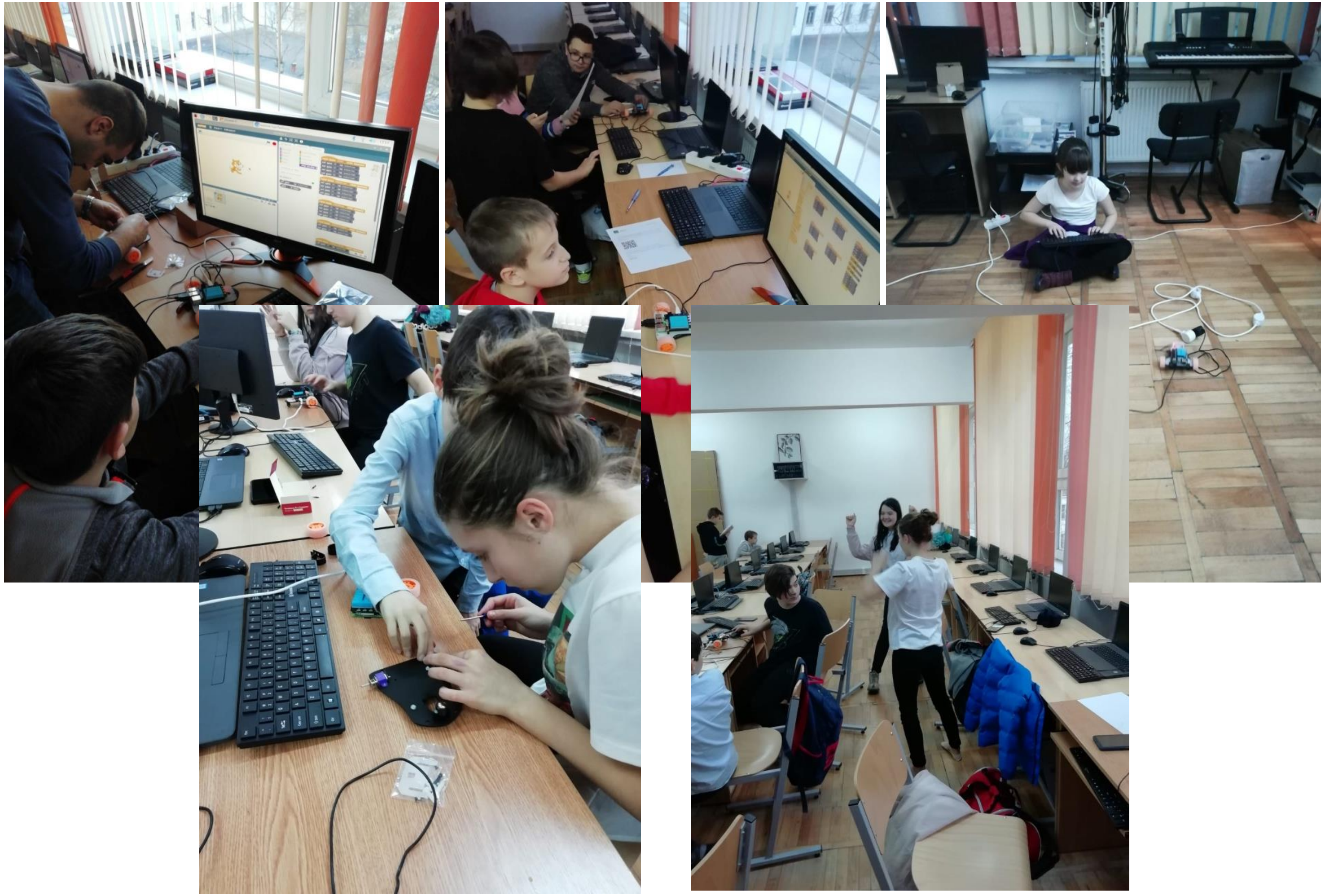
7 years,

200 + free trainings and workshops

60+ vounteers (40 + currently active)

AgileHub invests in free education

Robotics training (8 Month long) – for two groups of children from Școala 11, Brașov



Introduction

About myself and the presentation

Present:

- Working with 3SS since nearly 8 years
- Involved in projects around TV, VOD on various platforms

History:

- Started as a freelance web- and backend-developer
- Co-founder of a small development and design agency

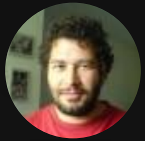
The DevOps Team @ 3SS

The foundation and backbone of this presentation

Driving DevOps at 3SS:

- Radu Curteanu

- Petre Toma

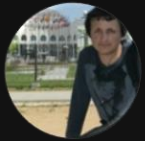


- Calin Paltinean

- Crina Mocian

- Florin Spartaru

- Nick Yunchyk



Agenda

DevOps Meetup

- Who is 3SS and what are we doing
- Where our journey started
- A short case study
- Where we are right now – our setup
- Challenges, Learnings, Do’s & Don’ts

Who is 3SS and what are we doing

Quick profile

- Specialized on frontend solutions (Multiscreen and Set-Top-Boxes)
- ~200 employees, 80% of them are part of delivery
- Organized in technology departments
- Working with Scaled Agile Framework (SAFe)

- Coming from classic “body-leasing” and development outsourcing
- Ongoing transformation into product & solution provider

Our Customers

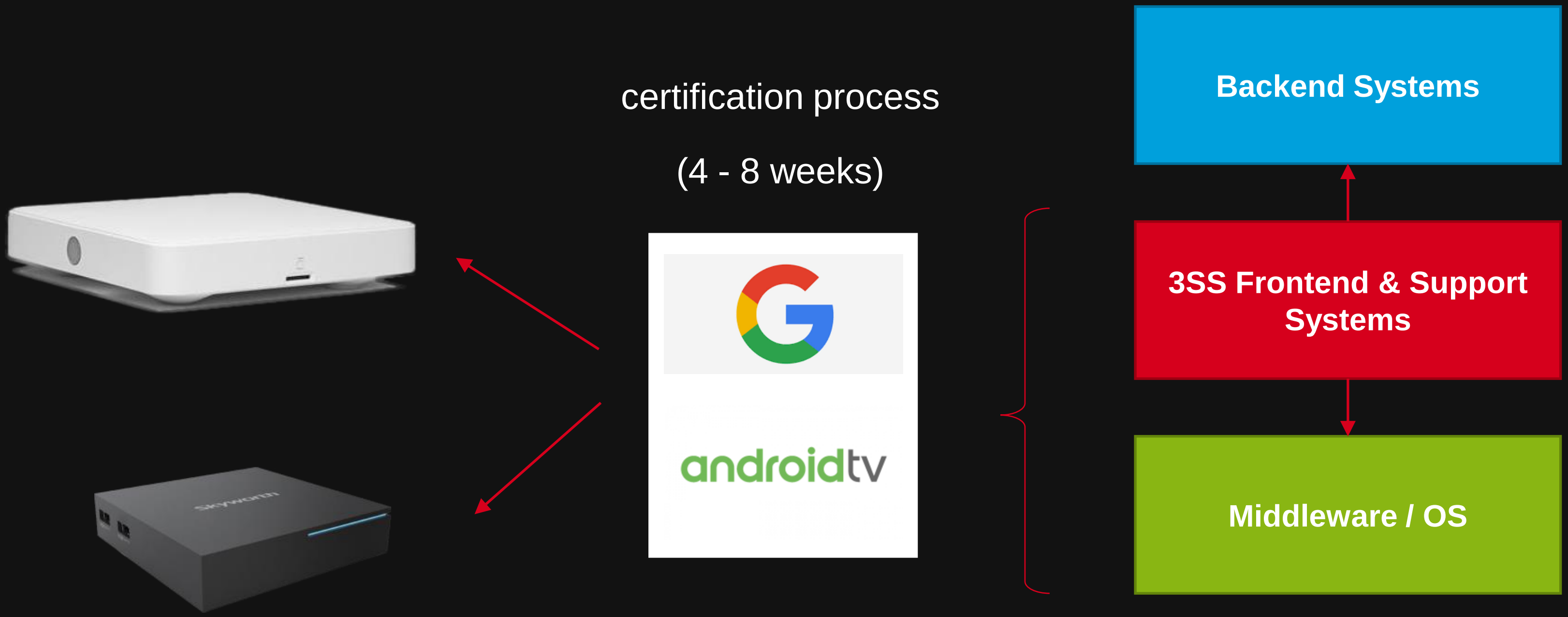
Average profile

- Operators & Broadcasters providing TV and VOD Services via DVB, IPTV and OTT
- Set-Top-Boxes as main platform

- Slow release-cycles, long product life-cycles
- Large organizations, often with traditional waterfall approach
- Complexity through middleware and DVB

Our Customers

Average profile



certification process

(4 - 8 weeks)

Backend Systems

3SS Frontend & Support Systems

Middleware / OS

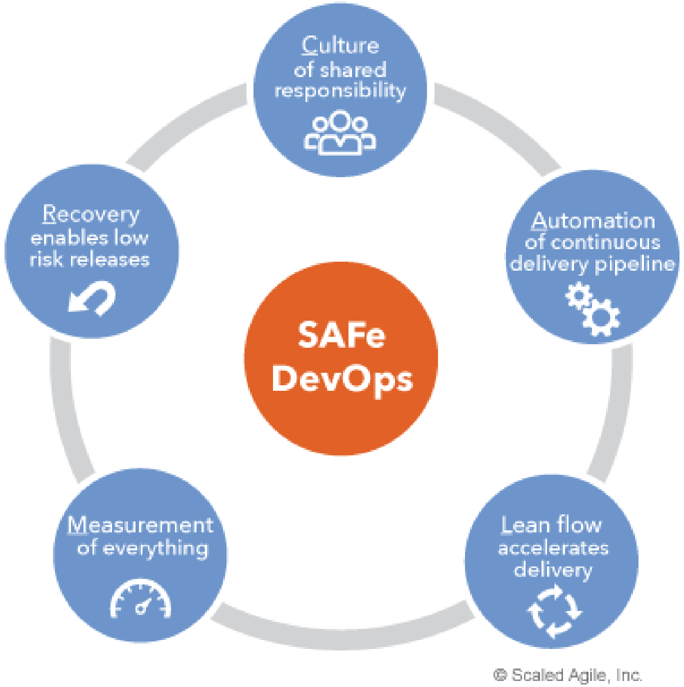
DevOps in our Context

What it means for 3SS

Dev	Ops
Development	Operations
3SS	Customer

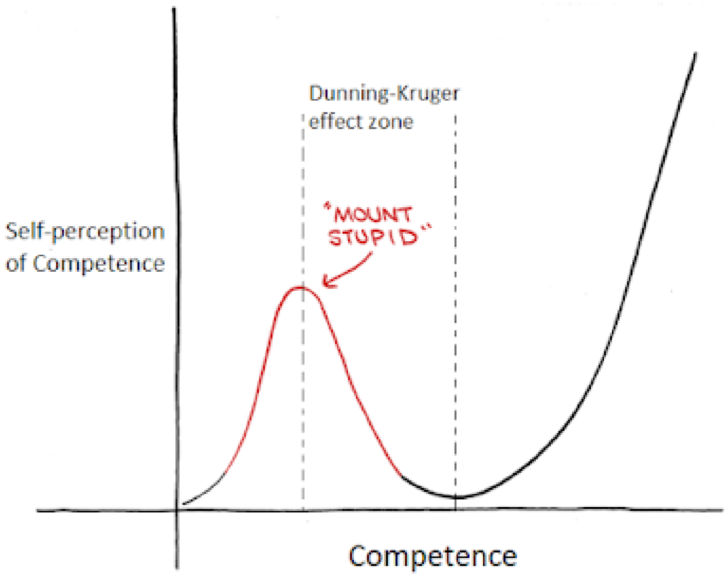
DevOps in our Context

What it means for 3SS


Culture of shared responsibility
Automation of continuous delivery pipeline
Lean flow accelerates delivery
Measurement of everything
Recovery enables low risk releases
SAFe DevOps
© Scaled Agile, Inc.

Where our journey started

A small remark before we get started…



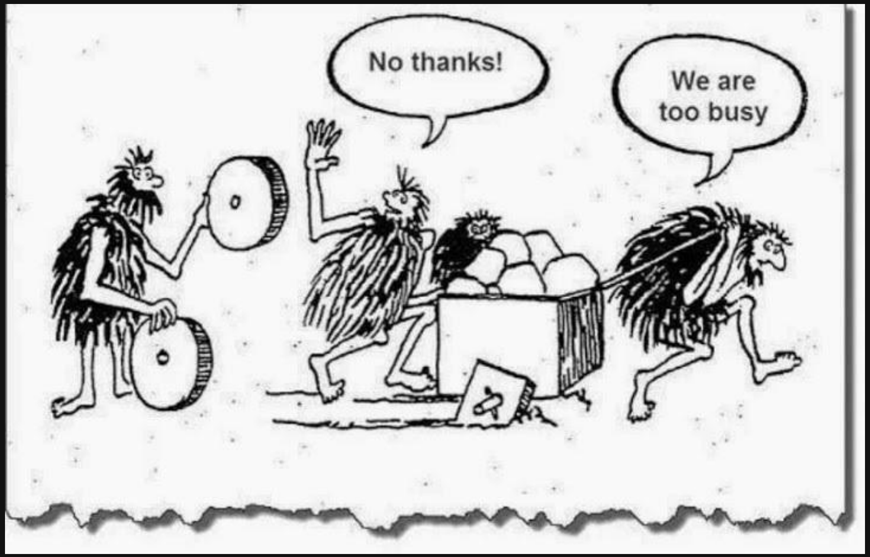
Where our journey started

Challenges we faced (and in parts are still facing)

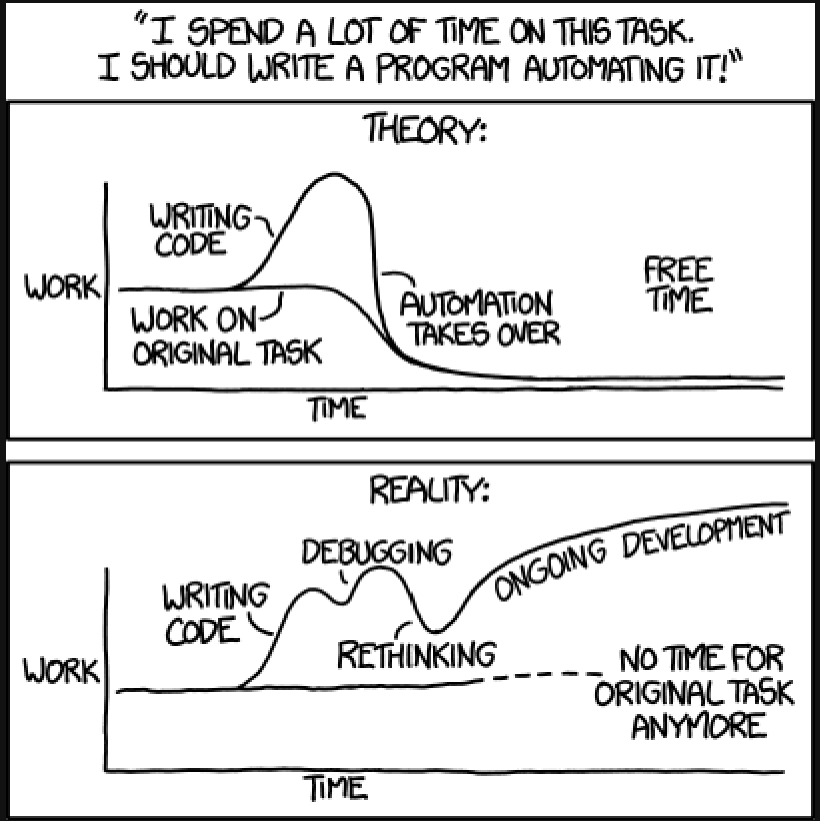
- Large user stories (multiple sprints)
- Long-living feature-branches - hard to merge
- Merges at the end of sprint
- daily/nightly builds
- All QA done at the end of sprint
- Only finished user stories were delivered to the customer
- “Release-Day” hectic

Where our journey started

It’s a matter of perspective



VS



Where our journey started

The start:

- All builds were done manual
- A lot of different environments & stages
- Manual creation of release-notes

Iteration #1:

- Trying to define principles
- DevOps team with senior devs to support teams
- Supplying the project teams

Iteration #2:

- Dedicated Team
- Supporting the project teams and become a prt of it

What we are doing

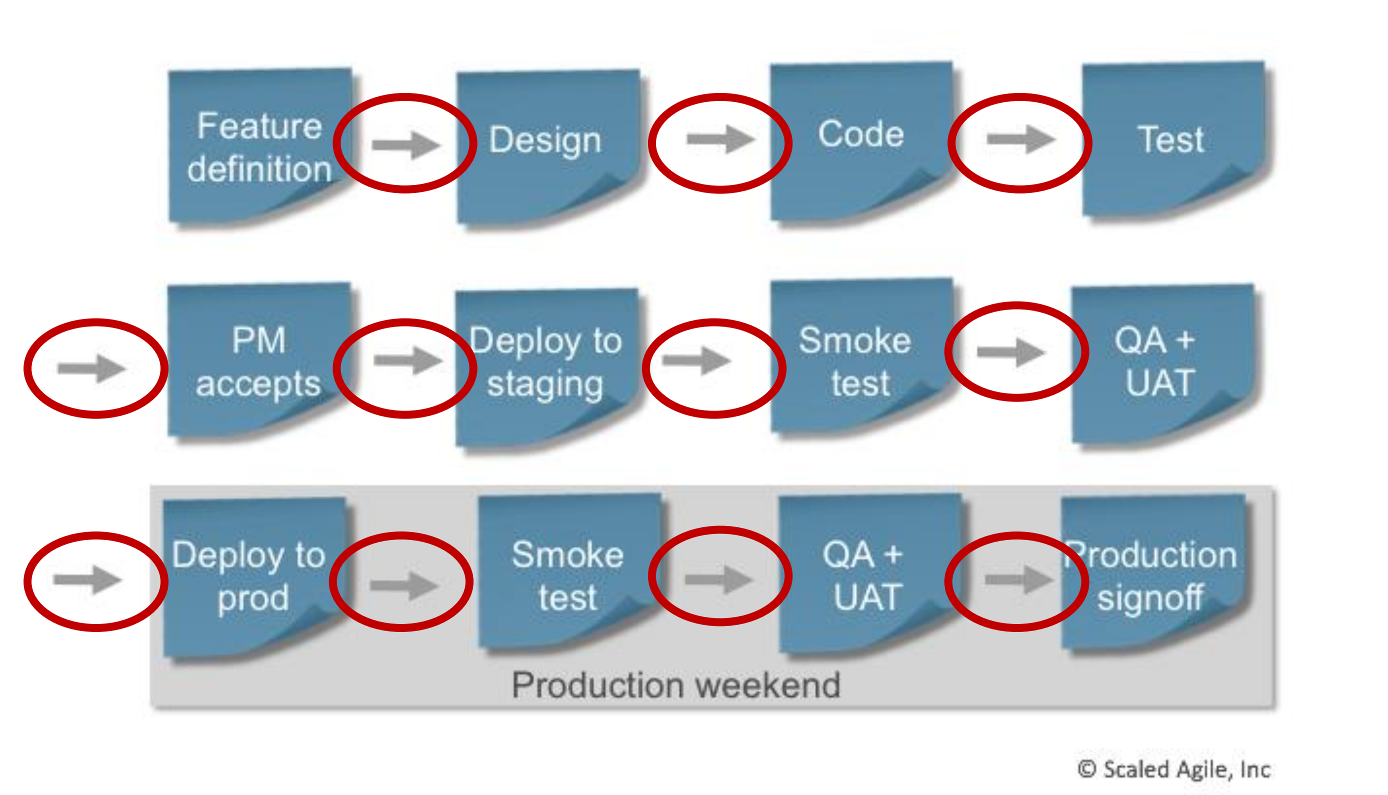
“Small batches”

- Result of the question: “What’s the **one** most relevant item for us?”
- General rule: “No User Story bigger than 3 Story Points”
- Feature-Flags to toggle functionality & features
- Decouple release from deploy
- Delivery fast and often

Where we are right now

- Focus on our “small batches” approach
- Focus on explaining the why and ongoingly support the team
- Establish “one-team” work-mode with customer and team

Why are small batches important?



© Scaled Agile, Inc

- Lead time vs. Process Time
- Process Time is **Wasted** Time

A case study

- Same team
- Same customer
- Different work-mode
- *Different Outcome?*

The situation

- Customer team, 3SS team. Prioritization, grooming etc. was done in sync but “separate”
- Huge delay, quality issues
- Frustrated team
- Unhappy customer, no trust → escalations
- Requirements and implementation did not meet customer nor product expectation
- Slow and cumbersome release cycles

The change

- Involvement of the customer in all steps incl. definition and testing
- 100% transparency
- Working together as one team with different roles and joined participation in all scrum procedures etc.
- Constant, frequent delivery of builds, even if unfinished features

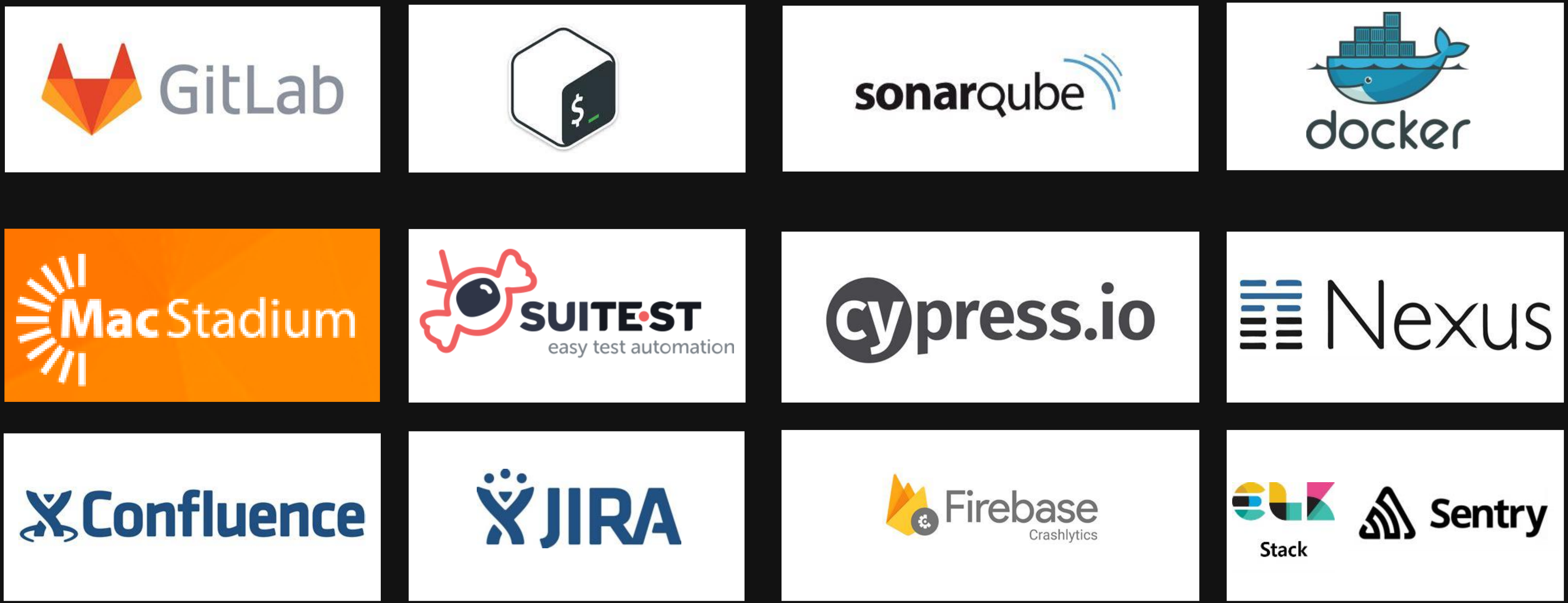
Results

- Better alignment of definition and implementation
- Common understanding about status, challenges and progress
- 12 hours from report to fix release

→Better outcome (Quality, Scope, Time To Market)

→180° in customer satisfaction

Our technology stack



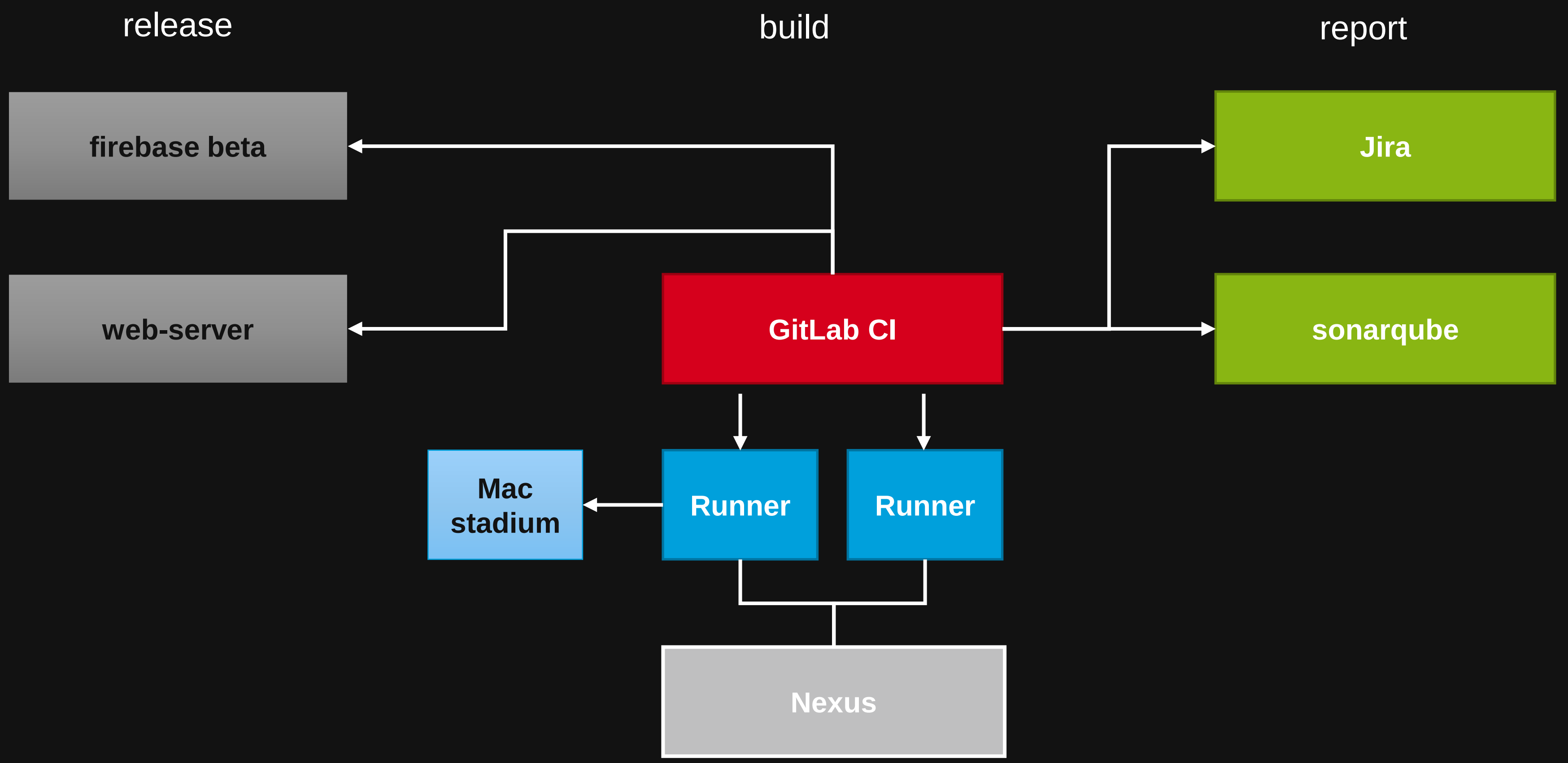
Our technology stack

Full Pipeline

- All pipelines (except iOS) as pre-configured **docker** containers
- Easy maintenance, usage and clean-installation
- Enables migration of pipelines across projects
- **bash** scripts to keep gitlab-ci.yml clean and simple
- **SonarQube** as monitoring/information tool, not quality gate to stop the pipeline

Our technology stack

Full Pipeline



Our technology stack

Jira & Confluence integrated with GitLab CI

stages:

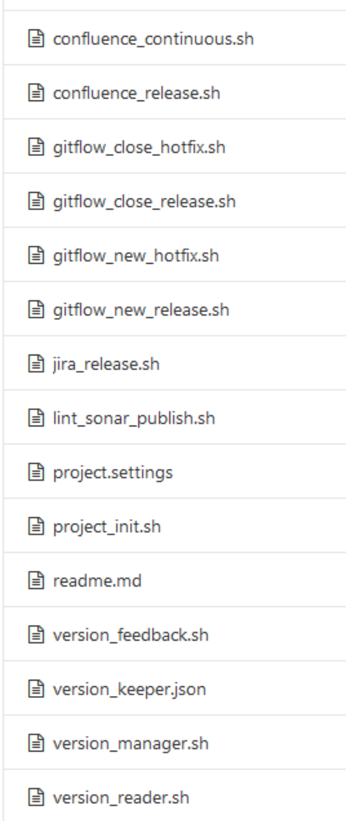
- lint
- build
- deploy
- release
- fixes
- finish

– each step invokes separate bash-scripts

– same stages for all projects and platforms

– different scripts per platform

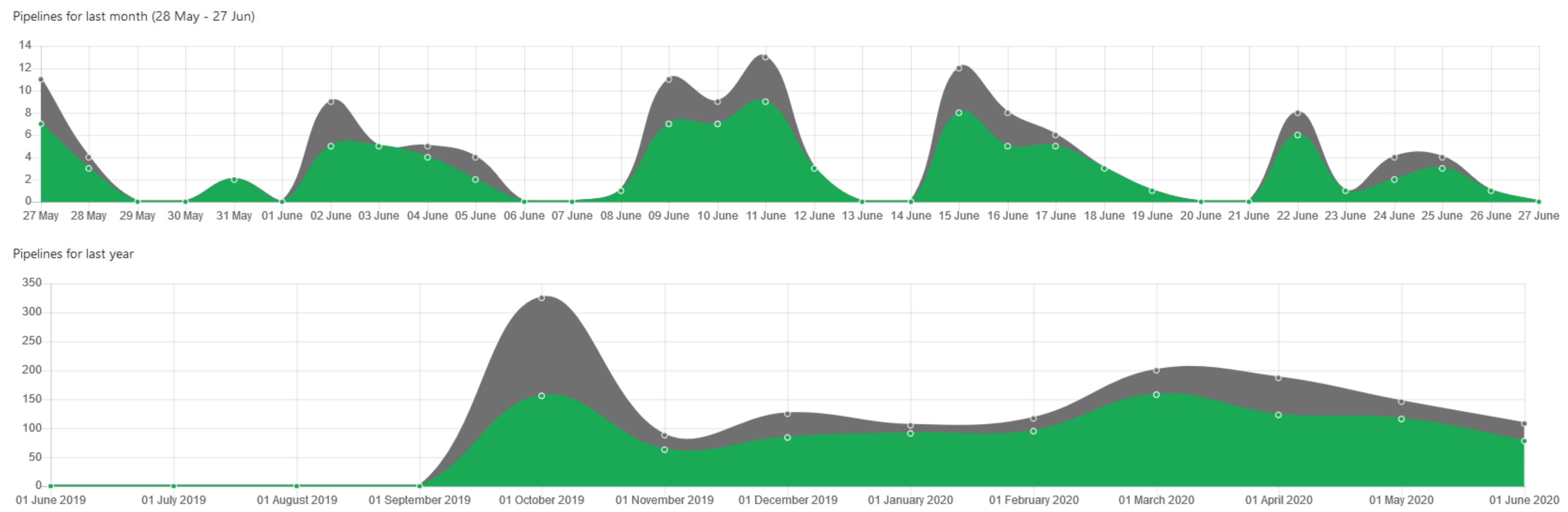
– update of Jira and Confluence via APIs



confluence_continuous.sh
confluence_release.sh
gitflow_close_hotfix.sh
gitflow_close_release.sh
gitflow_new_hotfix.sh
gitflow_new_release.sh
jira_release.sh
lint_sonar_publish.sh
project.settings
project_init.sh
readme.md
version_feedback.sh
version_keeper.json
version_manager.sh
version_reader.sh

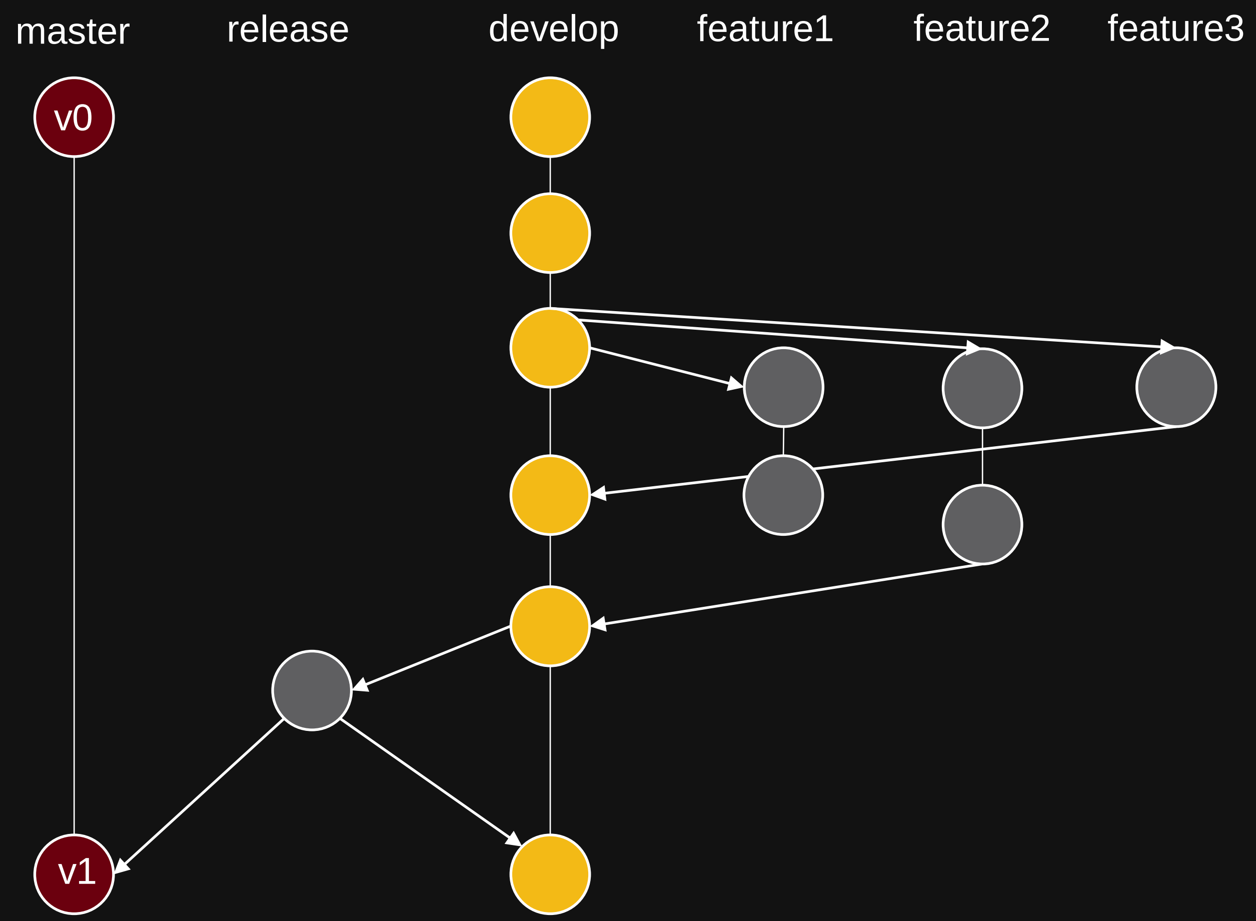
Our technology stack

#flattenthecurve



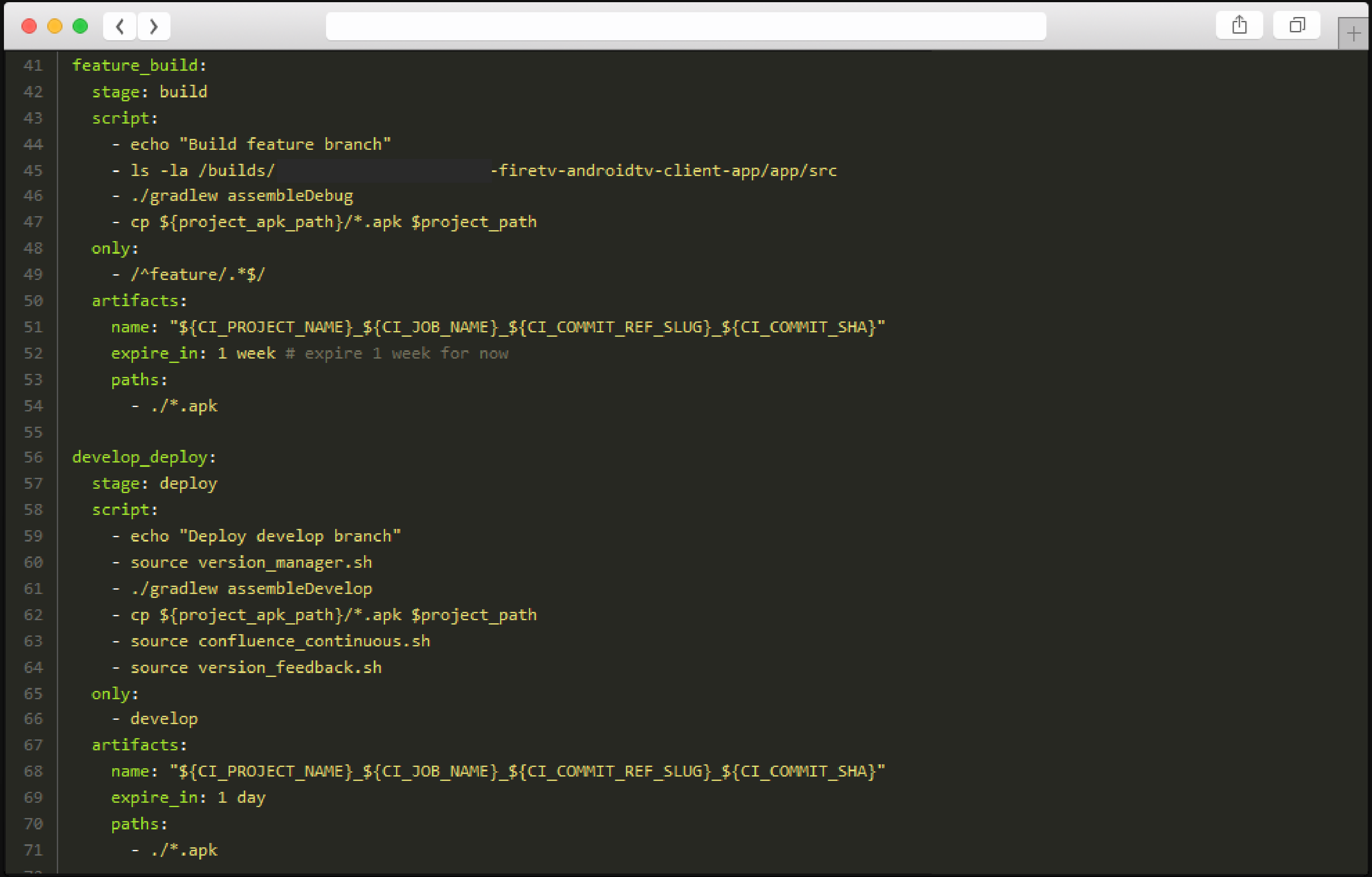
Our technology stack

How we use git and git-flow



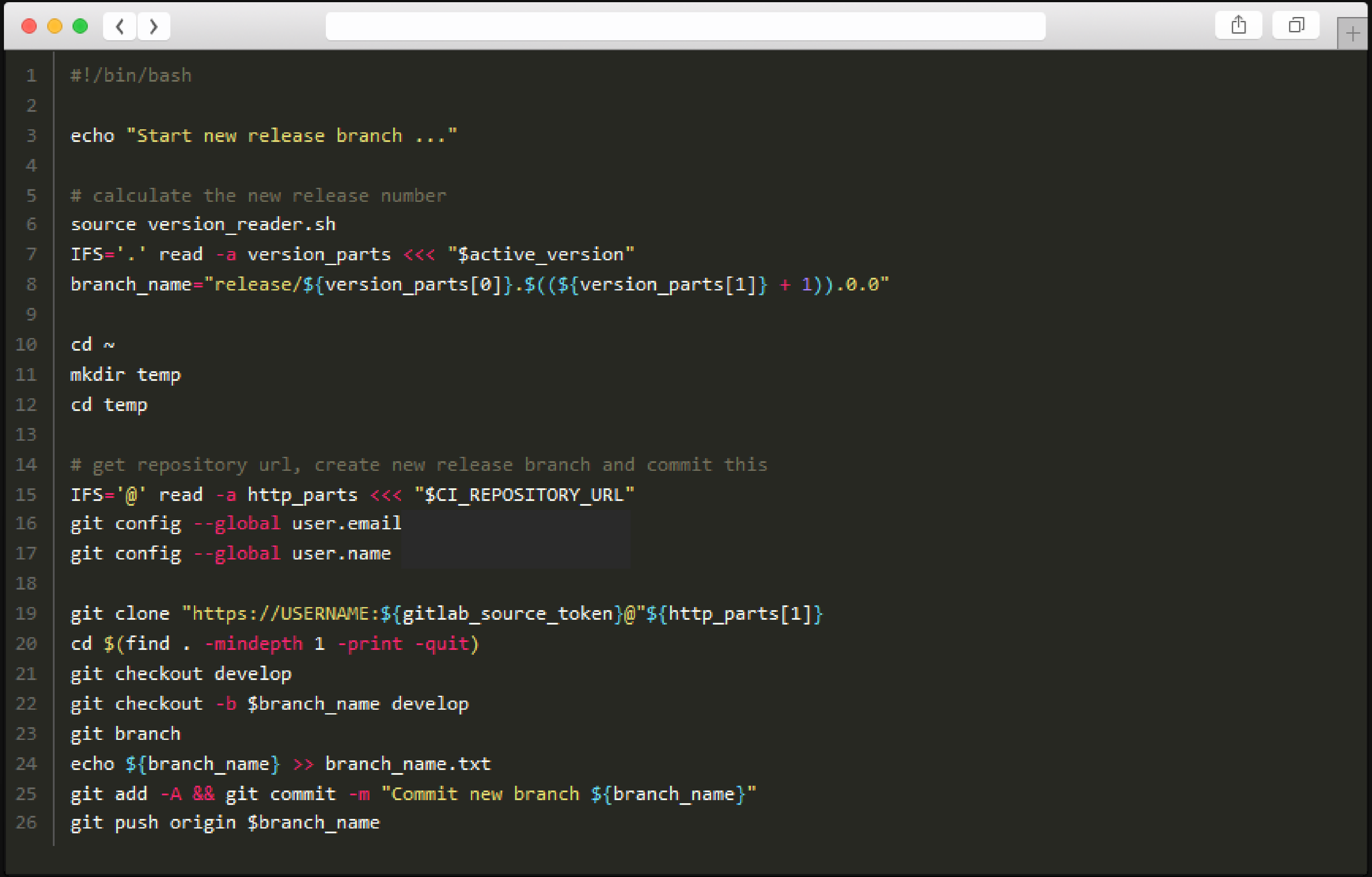
- git-flow with trunk-based-development strategy
- building from develop, short-lived feature-branches
- everything on develop is considered releasable
- no release candidates and hot fixes
- fixes are treated like any other branch
- merge and tag on master
- automatic branch management through bash scripts

gitlab-ci.yml extract – build stages



```yaml
feature_build:
  stage: build
  script:
    - echo "Build feature branch"
    - ls -la /builds/[redacted]-firetv-androidtv-client-app/app/src
    - ./gradlew assembleDebug
    - cp ${project_apk_path}/*.apk $project_path
  only:
    - /^feature/.*$/
  artifacts:
    name: "${CI_PROJECT_NAME}_${CI_JOB_NAME}_${CI_COMMIT_REF_SLUG}_${CI_COMMIT_SHA}"
    expire_in: 1 week # expire 1 week for now
    paths:
      - ./*.apk

develop_deploy:
  stage: deploy
  script:
    - echo "Deploy develop branch"
    - source version_manager.sh
    - ./gradlew assembleDevelop
    - cp ${project_apk_path}/*.apk $project_path
    - source confluence_continuous.sh
    - source version_feedback.sh
  only:
    - develop
  artifacts:
    name: "${CI_PROJECT_NAME}_${CI_JOB_NAME}_${CI_COMMIT_REF_SLUG}_${CI_COMMIT_SHA}"
    expire_in: 1 day
    paths:
      - ./*.apk
```

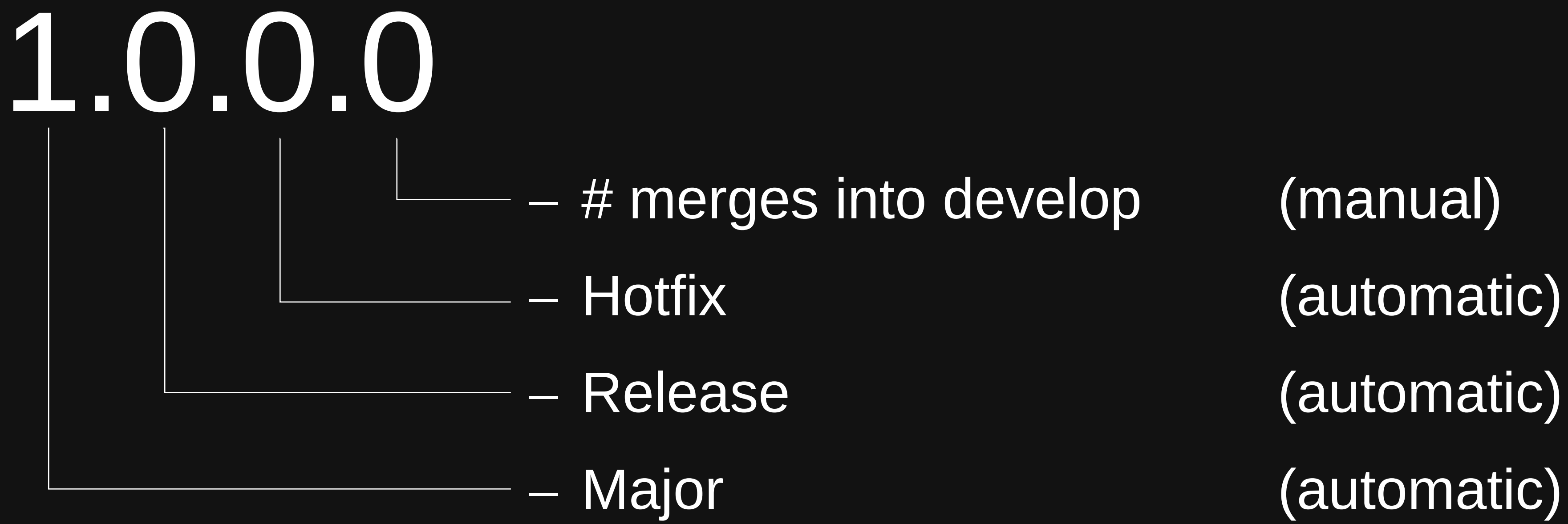
release branch creation in bash



```bash
#!/bin/bash

echo "Start new release branch ..."

# calculate the new release number
source version_reader.sh
IFS='.' read -a version_parts <<< "$active_version"
branch_name="release/${version_parts[0]}.$((${version_parts[1]} + 1)).0.0"

cd ~
mkdir temp
cd temp

# get repository url, create new release branch and commit this
IFS='@' read -a http_parts <<< "$CI_REPOSITORY_URL"
git config --global user.email [redacted]
git config --global user.name

git clone "https://USERNAME:${gitlab_source_token}@"${http_parts[1]}
cd $(find . -mindepth 1 -print -quit)
git checkout develop
git checkout -b $branch_name develop
git branch
echo ${branch_name} >> branch_name.txt
git add -A && git commit -m "Commit new branch ${branch_name}"
git push origin $branch_name
```

Our technology stack

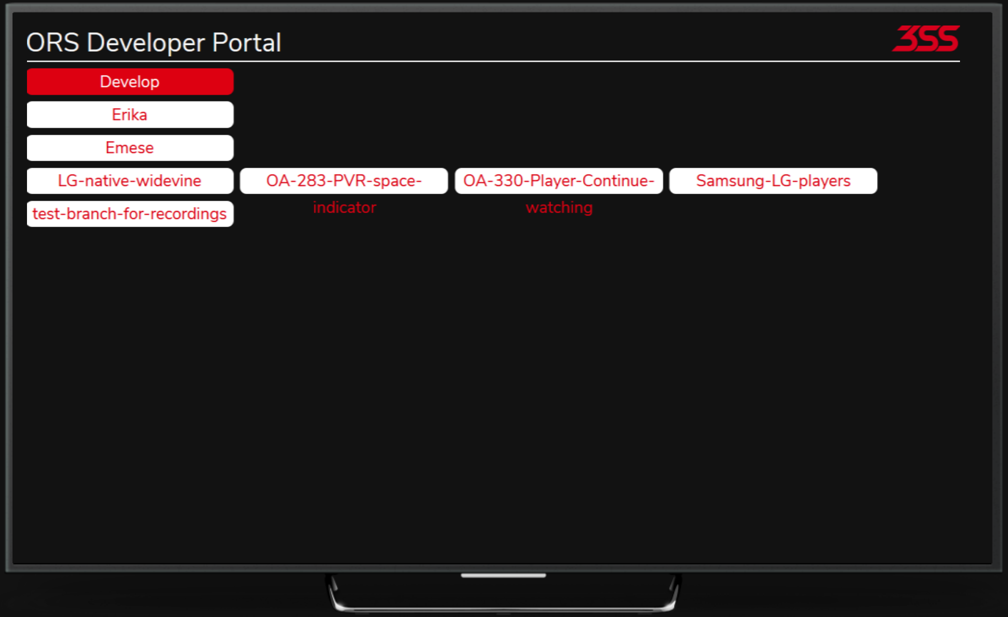
git, git-flow, versioning and releases



1.0.0.0

– # merges into develop	(manual)
– Hotfix	(automatic)
– Release	(automatic)
– Major	(automatic)

Deployment for TV


ORS Developer Portal
3SS
Develop
Erika
Emese
LG-native-widevine
OA-283-PVR-space-indicator
OA-330-Player-Continue-watching
Samsung-LG-players
test-branch-for-recordings

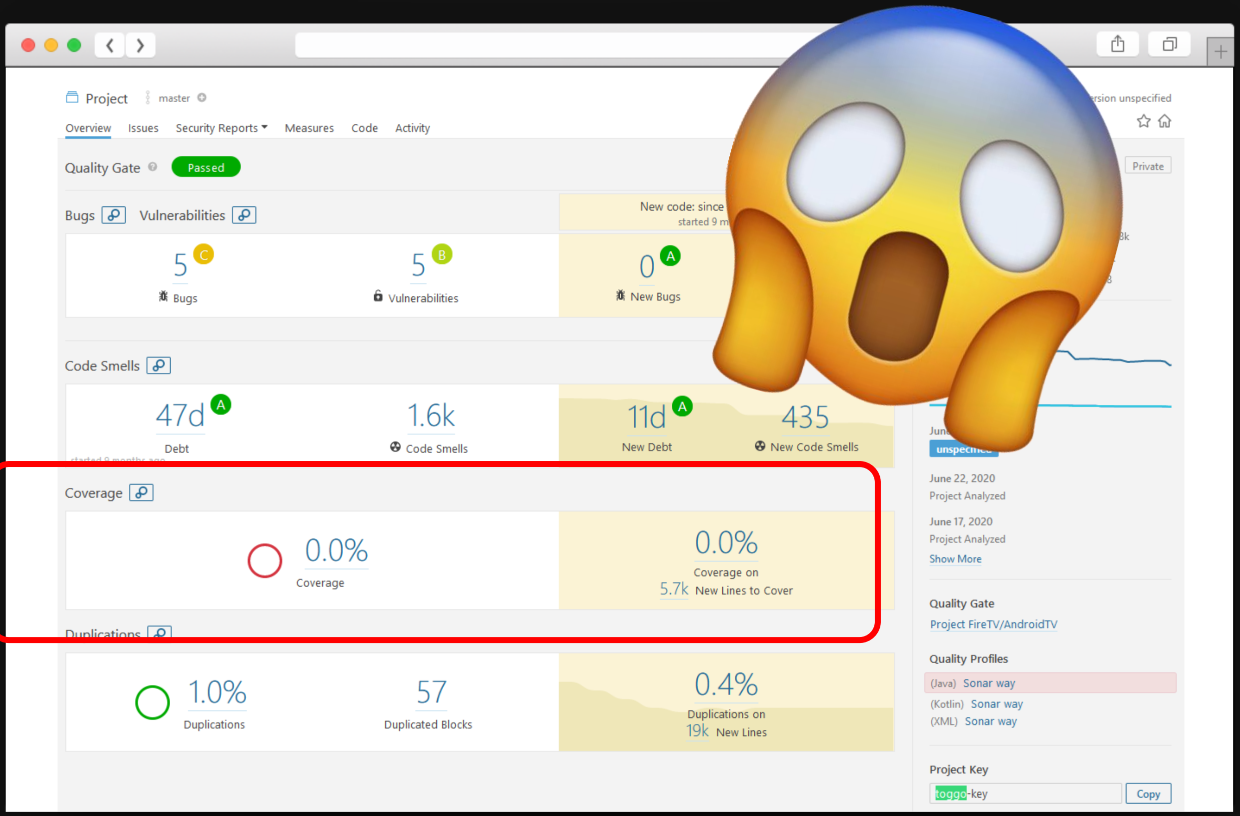
Web & Smart-TV:

- folder per branch
- Automatically generated “portal” page to navigate through environments and branches

Our technology stack

SonarQube

- Code-Smells & linting
- Vulnerabilities
- Potential (and actual) bugs)
- Taken as input by the team and planned as improvements if needed
- SonarLint for Dev IDEs



Project master

Quality Gate Passed

Bugs: 5 (C) Bugs

Vulnerabilities: 5 (B) Vulnerabilities

New code: since ... started 9 m...

0 (A) New Bugs

Code Smells: 47d (A) Debt; 1.6k Code Smells; 11d (A) New Debt; 435 New Code Smells

Coverage: 0.0% Coverage; 0.0% Coverage on 5.7k New Lines to Cover

Duplications: 1.0% Duplications; 57 Duplicated Blocks; 0.4% Duplications on 19k New Lines

...rsion unspecified

Private

June 22, 2020 Project Analyzed

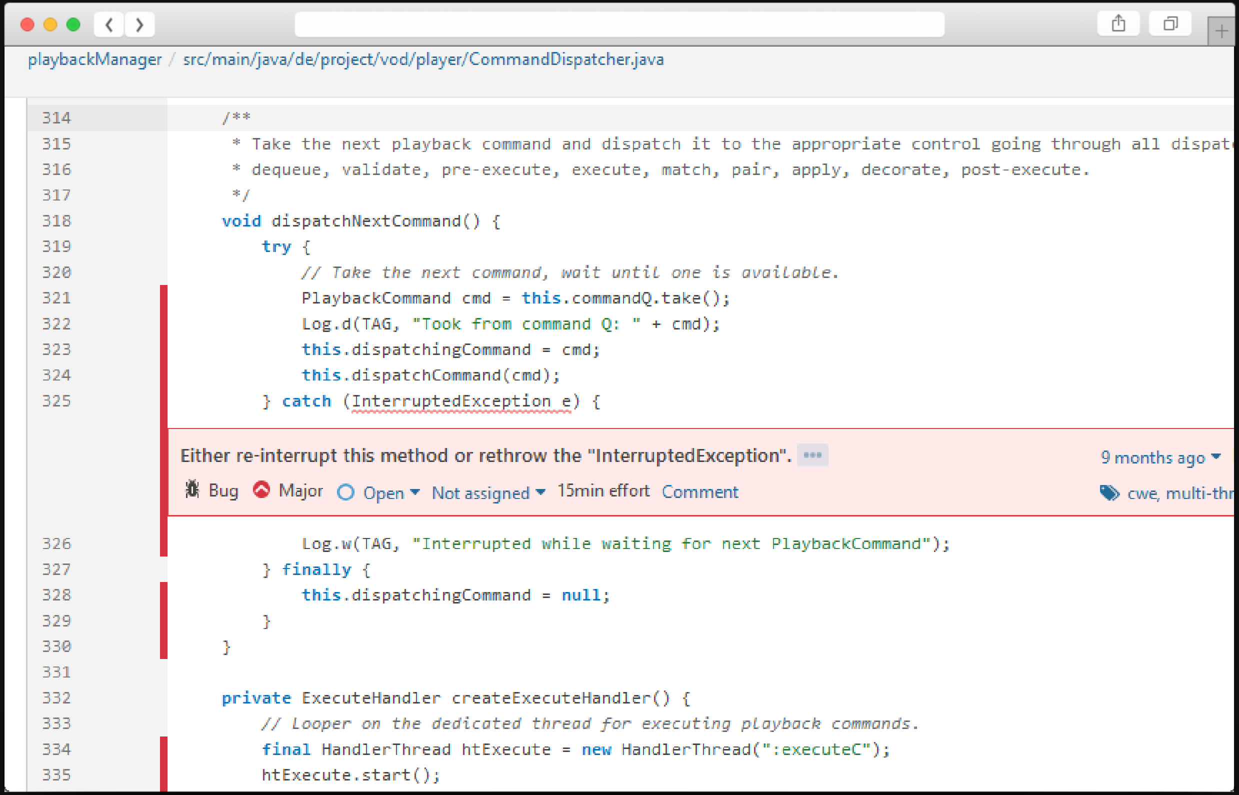
June 17, 2020 Project Analyzed

Show More

Quality Gate: Project FireTV/AndroidTV

Quality Profiles:
(Java) Sonar way
(Kotlin) Sonar way
(XML) Sonar way

Project Key: toggo-key



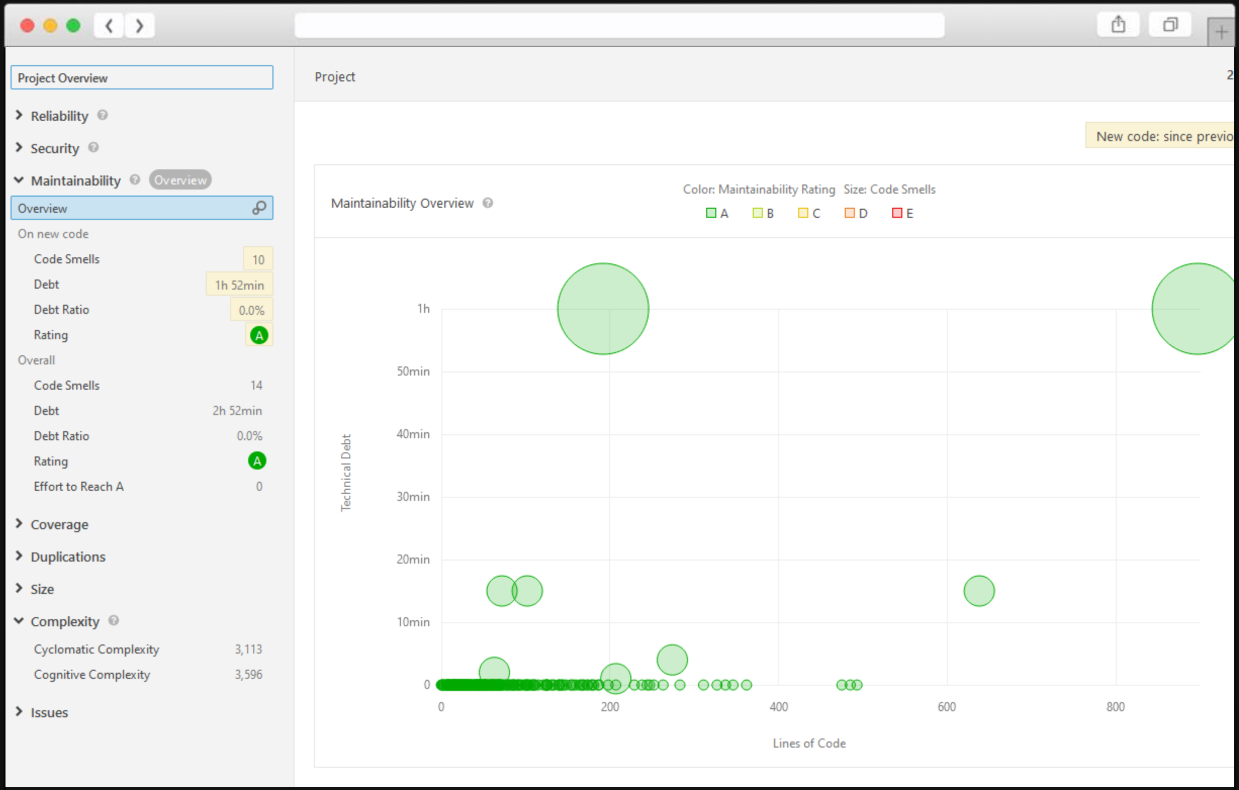
playbackManager / src/main/java/de/project/vod/player/CommandDispatcher.java

```java
    /**
     * Take the next playback command and dispatch it to the appropriate control going through all dispat
     * dequeue, validate, pre-execute, execute, match, pair, apply, decorate, post-execute.
     */
    void dispatchNextCommand() {
        try {
            // Take the next command, wait until one is available.
            PlaybackCommand cmd = this.commandQ.take();
            Log.d(TAG, "Took from command Q: " + cmd);
            this.dispatchingCommand = cmd;
            this.dispatchCommand(cmd);
        } catch (InterruptedException e) {
```

Either re-interrupt this method or rethrow the "InterruptedException". ••• 9 months ago ▾

Bug Major ○ Open ▾ Not assigned ▾ 15min effort Comment cwe, multi-thr

```java
            Log.w(TAG, "Interrupted while waiting for next PlaybackCommand");
        } finally {
            this.dispatchingCommand = null;
        }
    }

    private ExecuteHandler createExecuteHandler() {
        // Looper on the dedicated thread for executing playback commands.
        final HandlerThread htExecute = new HandlerThread(":executeC");
        htExecute.start();
```



Our technology stack

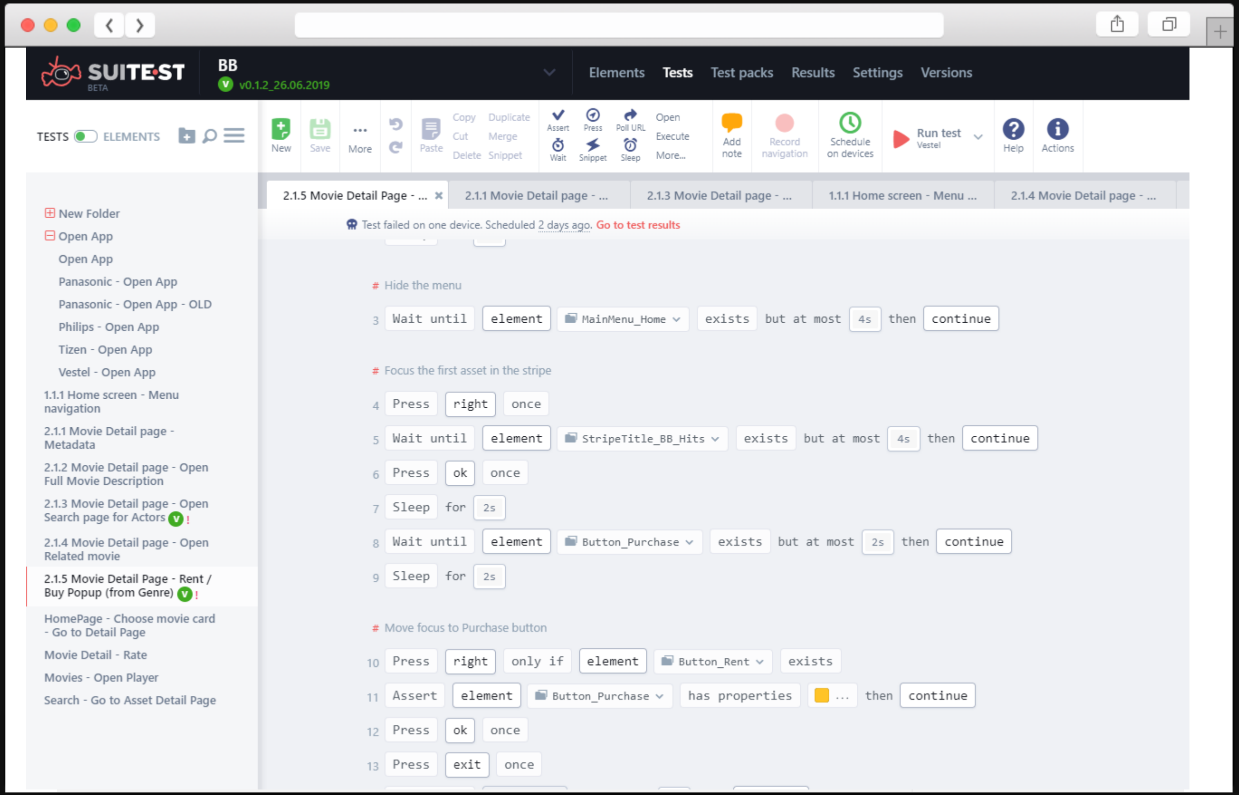
A word on testing

- Environments and backends in constant development → Tests break a lot
- No Test Driven Development (TDD)
- Limited client-side logic
- Unit-Tests only for specific cases
- Integration-Tests come at a later stage once the UI is stable enough and not evolving that fast

Our technology stack

Suite.st

- Cross-Device testing with actual IR input
- Easy test-editing also for non-developers
- Huge decrease of time needed for regression



New Folder
Open App
Open App
Panasonic - Open App
Panasonic - Open App - OLD
Philips - Open App
Tizen - Open App
Vestel - Open App
1.1.1 Home screen - Menu navigation
2.1.1 Movie Detail page - Metadata
2.1.2 Movie Detail page - Open Full Movie Description
2.1.3 Movie Detail page - Open Search page for Actors V !
2.1.4 Movie Detail page - Open Related movie
2.1.5 Movie Detail Page - Rent / Buy Popup (from Genre) V !
HomePage - Choose movie card - Go to Detail Page
Movie Detail - Rate
Movies - Open Player
Search - Go to Asset Detail Page

2.1.5 Movie Detail Page - ... | 2.1.1 Movie Detail page - ... | 2.1.3 Movie Detail page - ... | 1.1.1 Home screen - Menu ... | 2.1.4 Movie Detail page - ...

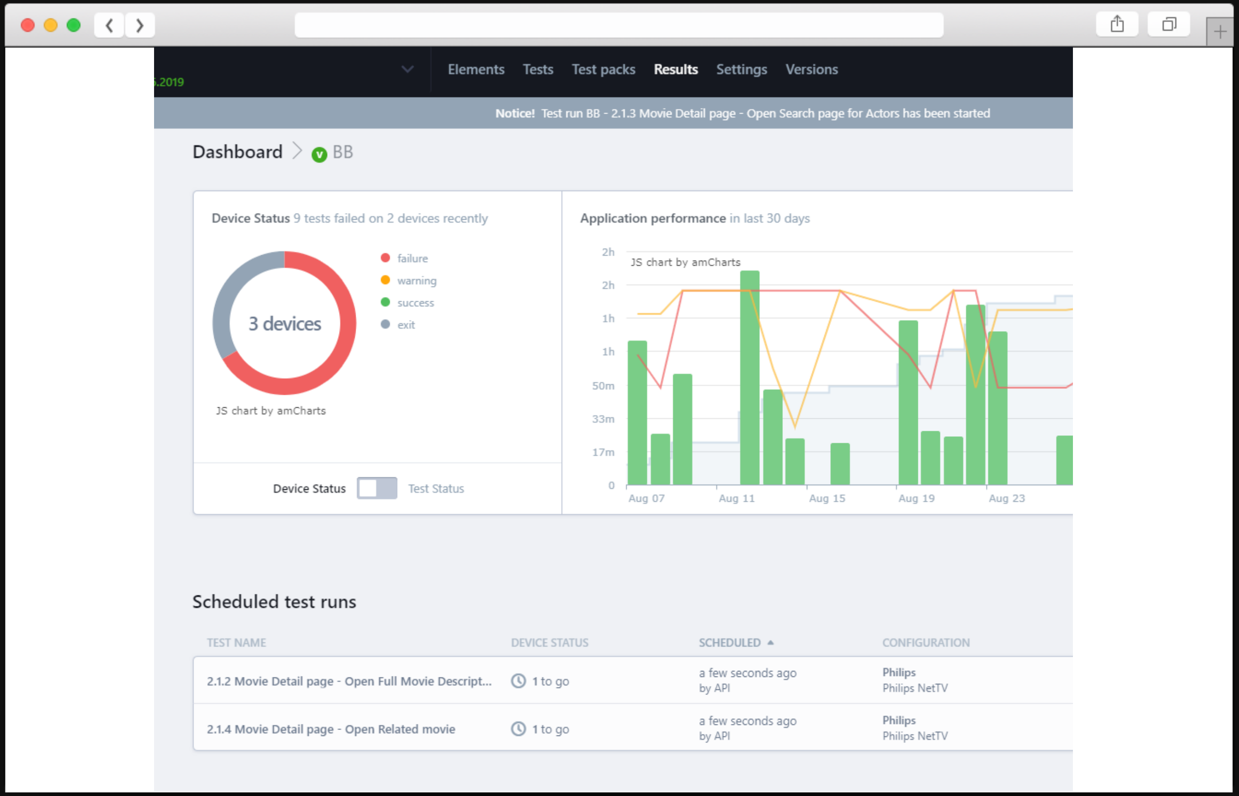
Test failed on one device. Scheduled 2 days ago. Go to test results

```
# Hide the menu
3  Wait until  element  MainMenu_Home  exists  but at most  4s  then  continue

# Focus the first asset in the stripe
4  Press  right  once
5  Wait until  element  StripeTitle_BB_Hits  exists  but at most  4s  then  continue
6  Press  ok  once
7  Sleep  for  2s
8  Wait until  element  Button_Purchase  exists  but at most  2s  then  continue
9  Sleep  for  2s

# Move focus to Purchase button
10  Press  right  only if  element  Button_Rent  exists
11  Assert  element  Button_Purchase  has properties  🟨 ...  then  continue
12  Press  ok  once
13  Press  exit  once
```

Our technology stack

Suite.st

- Cross-Device testing with actual IR input
- Easy test-editing also for non-developers
- Huge decrease of time needed for regression



Notice! Test run BB - 2.1.3 Movie Detail page - Open Search page for Actors has been started

Dashboard > BB

Device Status 9 tests failed on 2 devices recently

Application performance in last 30 days

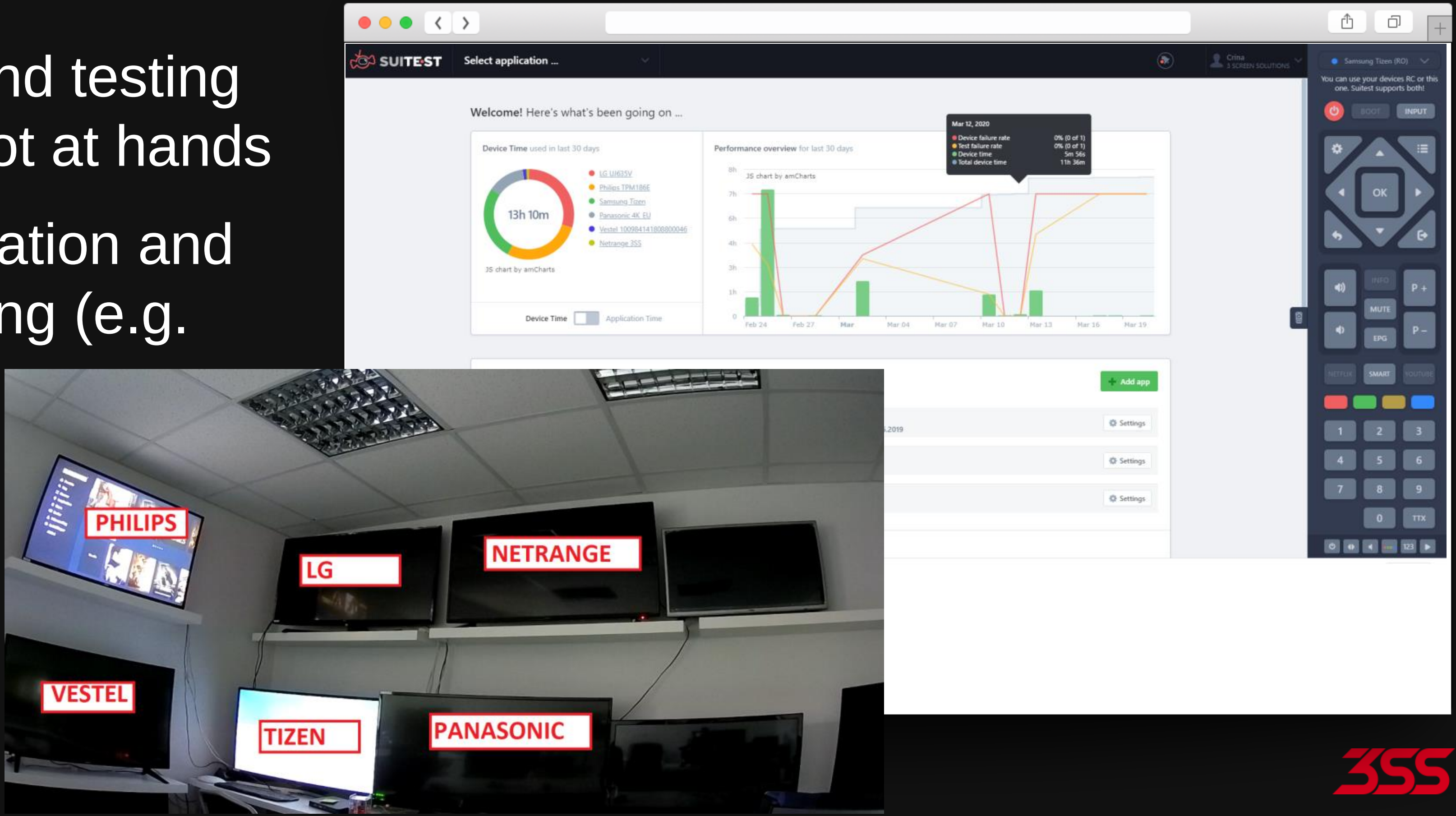
Scheduled test runs

TEST NAME	DEVICE STATUS	SCHEDULED ▲	CONFIGURATION
2.1.2 Movie Detail page - Open Full Movie Descript...	1 to go	a few seconds ago by API	Philips Philips NetTV
2.1.4 Movie Detail page - Open Related movie	1 to go	a few seconds ago by API	Philips Philips NetTV

Our technology stack

Remote Testlab

- Debugging and testing on devices not at hands
- Feature validation and troubleshooting (e.g. playback)
- COVID19 safe ☺



Welcome! Here's what's been going on ...

Device Time used in last 30 days

Performance overview for last 30 days

Mar 12, 2020

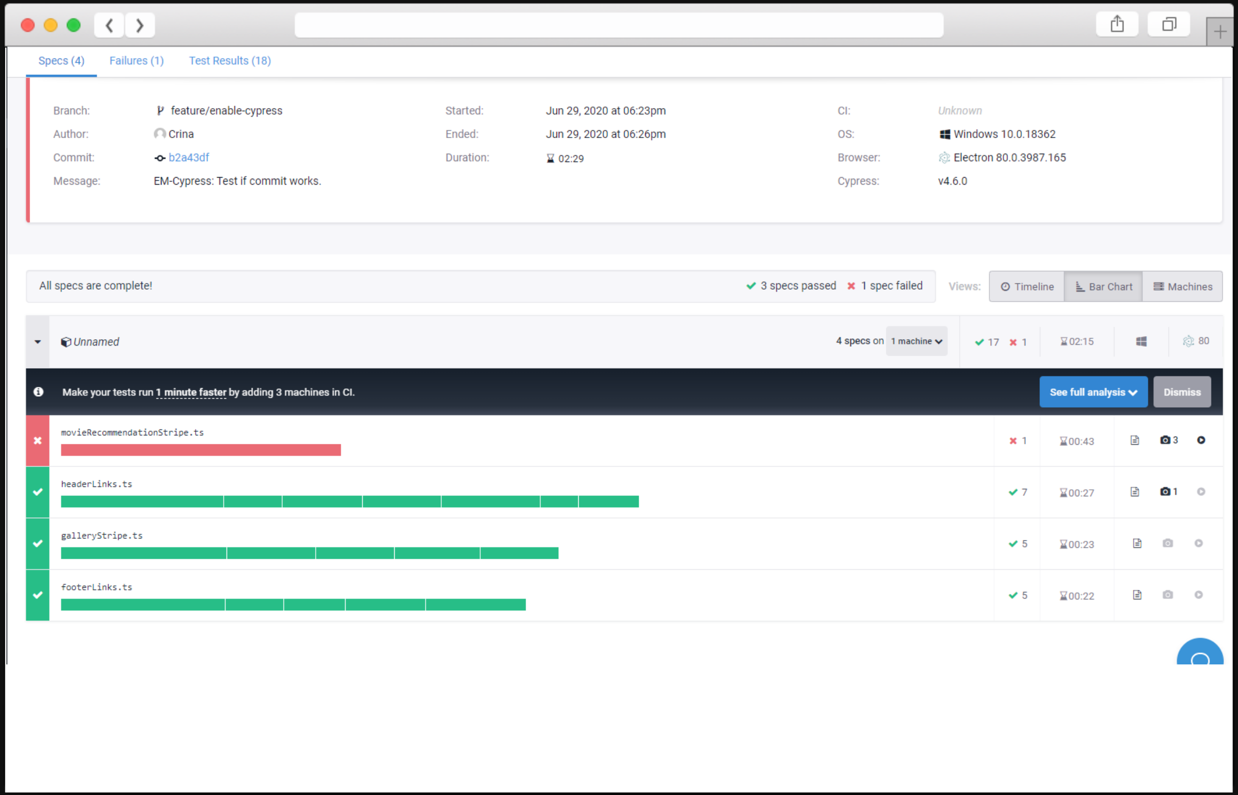
🟥 Device failure rate	0% (0 of 1)
🟧 Test failure rate	0% (0 of 1)
🟩 Device time	5m 56s
🟦 Total device time	11h 36m

You can use your devices RC or this one. Suitest supports both!

Our technology stack

Cypress

- Integration tests in browser with screenshots and video
- Simple test-syntax
- Integrated into CI/CD



Branch: feature/enable-cypress
Author: Crina
Commit: b2a43df
Message: EM-Cypress: Test if commit works.

Started: Jun 29, 2020 at 06:23pm
Ended: Jun 29, 2020 at 06:26pm
Duration: 02:29

CI: Unknown
OS: Windows 10.0.18362
Browser: Electron 80.0.3987.165
Cypress: v4.6.0

All specs are complete! ✓ 3 specs passed ✗ 1 spec failed

Unnamed 4 specs on 1 machine ✓ 17 ✗ 1 02:15 80

Make your tests run 1 minute faster by adding 3 machines in CI.

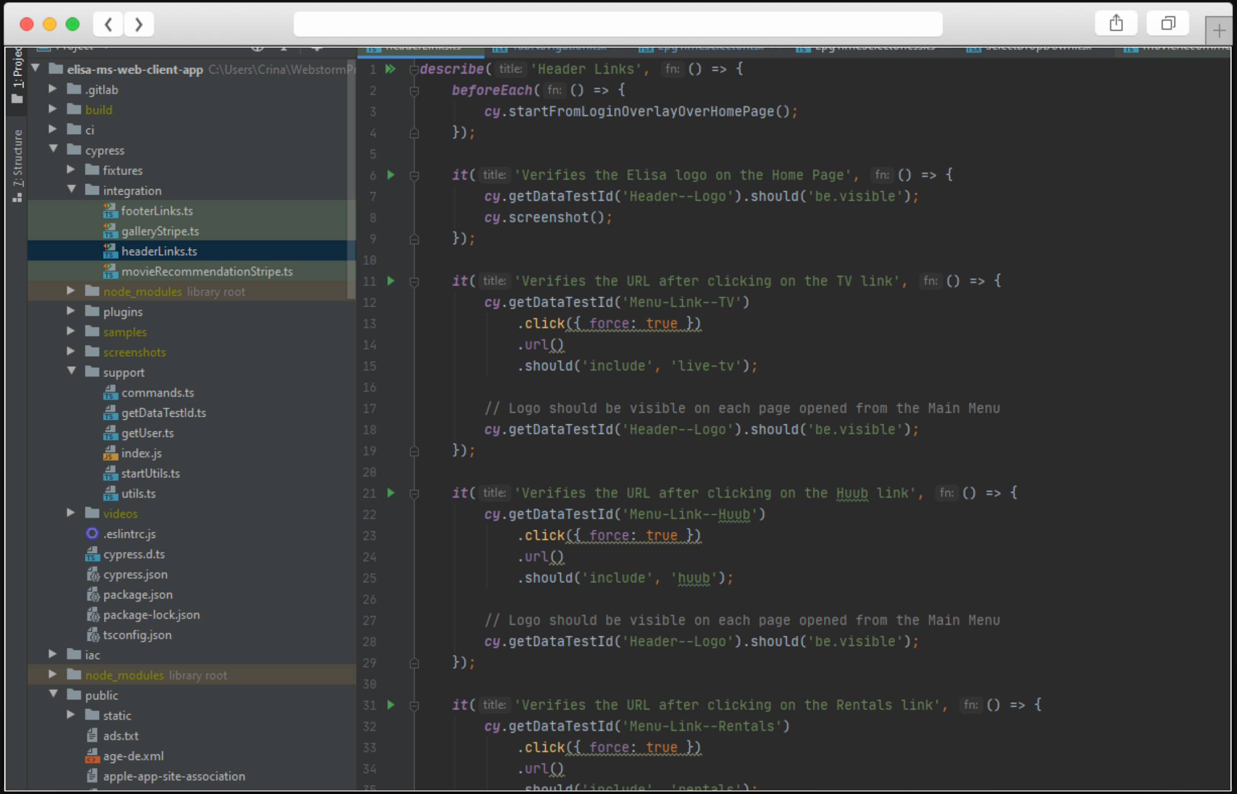
✗	movieRecommendationStripe.ts	✗ 1	00:43	3
✓	headerLinks.ts	✓ 7	00:27	1
✓	galleryStripe.ts	✓ 5	00:23	
✓	footerLinks.ts	✓ 5	00:22	

Our technology stack

Cypress

– Requires planning in project teams to support

– Development can be done by DevOps team or trained QA engineer

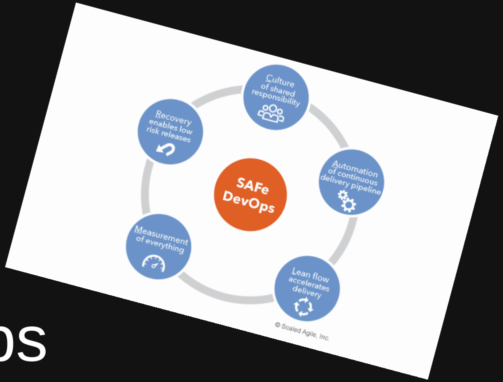


```
▼ elisa-ms-web-client-app  C:\Users\Crina\WebstormPr
  ▶ .gitlab
  ▶ build
  ▶ ci
  ▼ cypress
    ▶ fixtures
    ▼ integration
        footerLinks.ts
        galleryStripe.ts
        headerLinks.ts
        movieRecommendationStripe.ts
    ▶ node_modules  library root
    ▶ plugins
    ▶ samples
    ▶ screenshots
    ▼ support
        commands.ts
        getDataTestId.ts
        getUser.ts
        index.js
        startUtils.ts
        utils.ts
    ▶ videos
      .eslintrc.js
      cypress.d.ts
      cypress.json
      package.json
      package-lock.json
      tsconfig.json
  ▶ iac
  ▶ node_modules  library root
  ▼ public
    ▶ static
      ads.txt
      age-de.xml
      apple-app-site-association
```

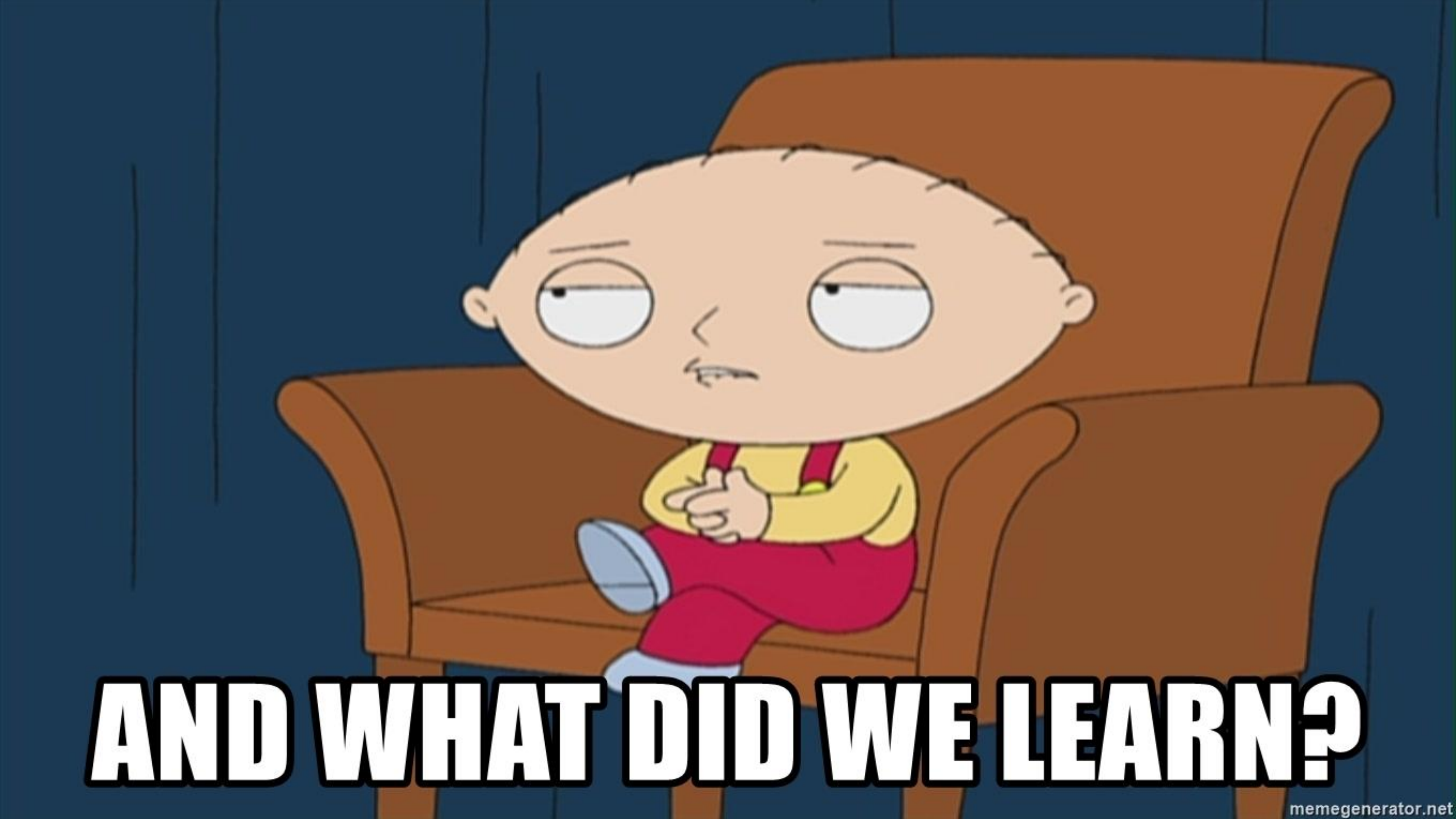
```
describe( 'Header Links', () => {
    beforeEach( () => {
        cy.startFromLoginOverlayOverHomePage();
    });

    it( 'Verifies the Elisa logo on the Home Page', () => {
        cy.getDataTestId('Header--Logo').should('be.visible');
        cy.screenshot();
    });

    it( 'Verifies the URL after clicking on the TV link', () => {
        cy.getDataTestId('Menu-Link--TV')
            .click({ force: true })
            .url()
            .should('include', 'live-tv');

        // Logo should be visible on each page opened from the Main Menu
        cy.getDataTestId('Header--Logo').should('be.visible');
    });

    it( 'Verifies the URL after clicking on the Huub link', () => {
        cy.getDataTestId('Menu-Link--Huub')
            .click({ force: true })
            .url()
            .should('include', 'huub');

        // Logo should be visible on each page opened from the Main Menu
        cy.getDataTestId('Header--Logo').should('be.visible');
    });

    it( 'Verifies the URL after clicking on the Rentals link', () => {
        cy.getDataTestId('Menu-Link--Rentals')
            .click({ force: true })
            .url()
            .should('include', 'rentals');
```

Summary?

- It took us at least 3 iterations to find an approach that (so far) seem to start working
- We still have a lot of untapped potential for implementing DevOps
- We needed to get to a culture that allowed the DevOps principles to be applied


SAFe DevOps
Culture of shared responsibility
Automation of continuous delivery pipeline
Lean flow accelerates delivery
Measurement of everything
Recovery enables low risk releases

© Scaled Agile, Inc.


AND WHAT DID WE LEARN?

Learnings

- Explain the why and provide context – to everyone involved
- Work in small batches
- Focus on delivering value
- Think holistic

Learnings

- The benefits of DevOps are more processes then technical
- The technology choices are secondary to the flows
- Have a team of supporters and enthusiasts, expand from there by providing value to the teams
- Involve, teach and train teams and customers early
- Support constantly, customer but especially also the team

Thank you for your attention!

ENGINEERING ENTERTAINMENT