

Otto-Friedrich-Universität Bamberg



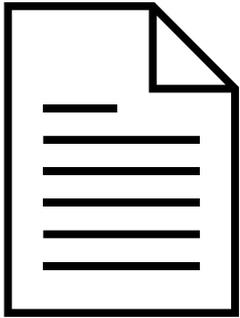
# An Architecture for the Automated Assessment of Web Programming Tasks



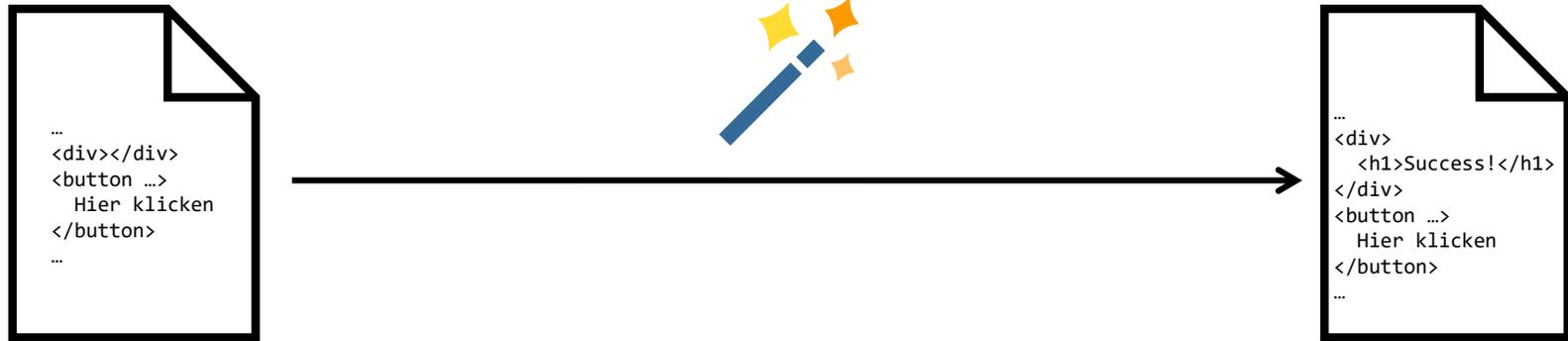
Lara Aubele, Leon Martin, Tobias Hirmer, Andreas Henrich  
ABP 2021



- Motivation
- Related Work & Foundations
- Design Decisions
- Behind the Scenes
- Show Case
- Conclusion



*„Implement a function that adds an h1-element reading ‚Success!‘ to the web page, if a button is clicked.“*



- even simple web programming tasks can have various solutions
- testing the effects of a script on an interface exceeds single units



How can an architecture for the automatic assessment of web programming tasks be designed?

# Related Work & Foundations

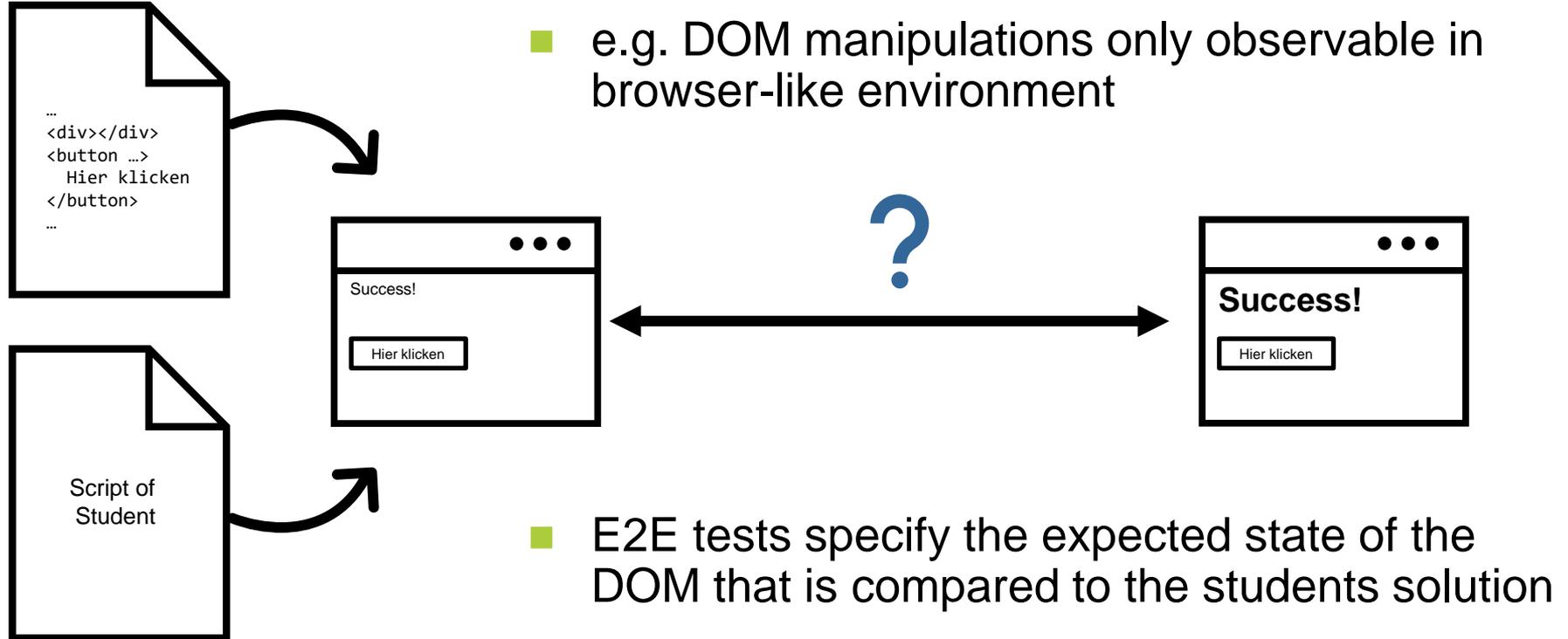


- many providers address the learning of web programming, e.g.



- none of them meet all our requirements due to ...
  - ... no support in adding own tasks
  - ... focussing on single-file level rather than realistic projects
  - ... missing availability on multiple devices
  - ... focus on unit testing [Ru06]

# Design Decisions – Why E2E-Testing?



# Design Decisions – Why Docker?

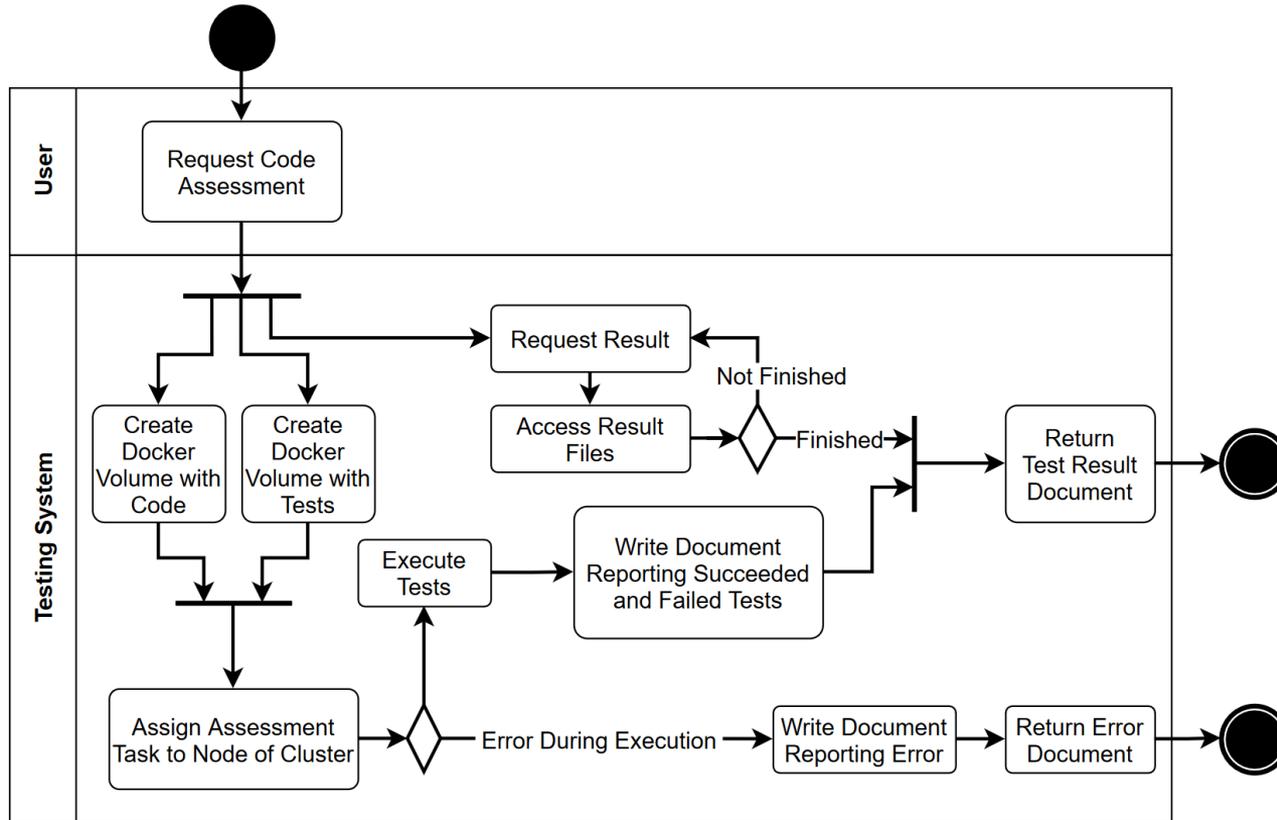


- Virtualizes the operating system and not the hardware
- Portability [Be14] and networking
- Extra layer of security (containerization) [Bu15]
- Executing code produced by users with low risk
- Tools like Docker Swarm allow for scalability



How can an architecture for the automatic assessment of web programming tasks **that combines E2E-Testing and Docker** be designed?

# Behind the Scenes



# Show Case – Teacher View



Initial steps:

- Create web page and tests locally
- Write task in application
- Upload the created files

```
1 context('Der Text', () => {
2   beforeEach(() => {
3     cy.visit('localhost:8081')
4   })
5
6   it('wird in einem H1 Element angezeigt.', () => {
7     cy.get('button').click()
8     cy.contains('Success!').get('h1');
9     cy.contains('Success!').children().should('have.length', 0);
10
11   })
12 })
13
```



# Show Case – Teacher View



LernApp

Forum

Admin



Name

Beschreibung

Aufgabentyp javascript

Maximale Punktzahl

Code auswählen

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

## Aufgaben

Name der Aufgabe

# Show Case – Teacher View



LernApp

Forum

Admin



Name

Kleine Programmieraufgabe

Beschreibung

Beschreibung

Aufgabentyp javascript



Maximale Punktzahl

10



Code auswählen 

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

## Aufgaben

Name der Aufgabe

Name

# Show Case – Teacher View



LernApp

Forum

Admin



Name

Kleine Programmieraufgabe

Beschreibung

Beschreibung

Aufgabentyp javascript

Maximale Punktzahl

10

Code auswählen

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

## Aufgaben

Name der Aufgabe

Name

Name	Size
functions	2 items
node_modules	30 items
index.html	232 bytes
index.js	39 bytes
package.json	299 bytes
package-lock.json	8,6 kB

# Show Case – Teacher View



## Aufgaben

Name der Aufgabe

Success

Aufgabenbeschreibung

Aufgabenbeschreibung

Nummer

1

Cypress E2E-Tests für die Aufgabe auswählen ⓘ

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

Cypress Unit-Tests für die Aufgabe auswählen

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

+ Programmieraufgabe zufügen

Programmieraufgaben speichern

Diese Anwendung wurde im Rahmen der Masterarbeit von Lara Aubele im Jahr 2021 am Lehrstuhl für Medieninformatik entwickelt.

# Show Case – Teacher View



## Aufgaben

Name der Aufgabe

Success

Aufgabenbeschreibung

Aufgabenbeschreibung

Nummer

1

Cypress E2E-Tests für die Aufgabe auswählen ⓘ

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

Cypress Unit-Tests für die Aufgabe auswählen

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

+ Programmieraufgabe zufügen

Programmieraufgaben speichern

Diese Anwendung wurde im Rahmen der Masterarbeit von Lara Aubele im Jahr 2021 am Lehrstuhl für Medieninformatik entwickelt

# Show Case – Teacher View



## Aufgaben

Name der Aufgabe

Success

Aufgabenbeschreibung

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.  
Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Nummer

1

Cypress E2E-Tests für die Aufgabe auswählen ⓘ

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

Cypress Unit-Tests für die Aufgabe auswählen

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

+ Programmieraufgabe zufügen

Programmieraufgaben speichern

Diese Anwendung wurde im Rahmen der Masterarbeit von Lara Aubele im Jahr 2021 am Lehrstuhl für Medieninformatik entwickelt

# Show Case – Teacher View



## Aufgaben

Name der Aufgabe

Success

Aufgabenbeschreibung

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.  
Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Nummer

1

Cypress E2E-Tests für die Aufgabe auswählen ⓘ

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

Cypress Unit-Tests für die Aufgabe auswählen

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

+ Programmieraufgabe zufügen

Programmieraufgaben speichern

Name	Size
JS h1_spec.js	488 bytes

# Show Case – Teacher View



## Aufgaben

Name der Aufgabe

Success

Aufgabenbeschreibung

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.  
Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Nummer

1

Cypress E2E-Tests für die Aufgabe auswählen ⓘ

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

h1.spec.js ✖

Cypress Unit-Tests für die Aufgabe auswählen

Drag 'n' drop some files or directories here, or click to select files

Ausgewählt:

+ Programmieraufgabe zufügen

Programmieraufgaben speichern

# Show Case – Student View



JS LernApp

Forum

Admin



## Kleine Programmieraufgabe

▼ Aufgabe: Success

To Do

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.

Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Aufgabe Validieren

Code Speichern

Ausgabe anzeigen

functions/h1.js

index.html

index.js

package.json

```
1 export function h1(){
2
3   const h1Element = document.createElement("p");
4   h1Element.innerHTML="Success!";
5   return h1Element;
6
7 }
8
```

Code zurücksetzen

Diese Anwendung wurde im Rahmen der Masterarbeit von Lara Aubele im Jahr 2021 am Lehrstuhl für Medieninformatik entwickelt.

# Show Case – Student View



LernApp

Forum

Admin



## Kleine Programmieraufgabe

▼ Aufgabe: Success

To Do

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.

Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Aufgabe Validieren

Code Speichern

🔄 Ausgabe refreshen

Ausgabe anzeigen

functions/h1.js

index.html

index.js

package.json

```
1 export function h1(){
2
3   const h1Element = document.createElement("p");
4   h1Element.innerHTML="Success!"
5   return h1Element;
6
7 }
8
```

vxun2p-run.stackblitz.io

Click me

# Show Case – Student View



LernApp

Forum

Admin



Bitte warten Sie. Das Testen dauert ein paar Sekunden.

## Kleine Programmieraufgabe

▼ Aufgabe: Success

To Do

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.

Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Aufgabe Validieren

Code Speichern

🔄 Ausgabe refreshen

Ausgabe anzeigen

functions/h1.js

index.html

index.js

package.json

```
1 export function h1(){
2
3   const h1Element = document.createElement("p");
4   h1Element.innerHTML="Success!"
5   return h1Element;
6
7 }
8
```

vxun2p--run.stackblitz.io

Click me

# Show Case – Student View



LernApp



Forum

Admin



## Testergebnis

Aufgabe leider nicht geschafft.

Fehlgeschlagener Test: Der Text wird in einem H1 Element angezeigt.

Schließen

## Kleine Programmieraufgabe

▼ Aufgabe: Success

To Do

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.

Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

# Show Case – Student View



JS LernApp

Forum

Admin



## Kleine Programmieraufgabe

▼ Aufgabe: Success

To Do

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.

Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Aufgabe Validieren

Code Speichern

Ausgabe refreshen

Ausgabe anzeigen

functions/h1.js

index.html

index.js

package.json

```
1 export function h1(){
2
3   const h1Element = document.createElement("h1");
4   h1Element.innerHTML="Success!";
5   return h1Element;
6
7 }
8
```

← → 88qinm-run.stackblitz.io

Click me

# Show Case – Student View



JS LernApp

Forum

Admin



## Kleine Programmieraufgabe

▼ Aufgabe: Success

To Do

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.

Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Aufgabe Validieren

Code Speichern

↻ Ausgabe refreshen

Ausgabe anzeigen

functions/h1.js

index.html

index.js

package.json

```
1 export function h1(){
2
3   const h1Element = document.createElement("h1");
4   h1Element.innerHTML="Success!";
5   return h1Element;
6
7 }
8
```

← → ↻ 88qinm-run.stackblitz.io

Click me

# Show Case – Student View



JS LernApp

Forum

Admin



## Kleine Programmieraufgabe

▼ Aufgabe: Success

To Do

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.

Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Aufgabe Validieren

Code Speichern

↻ Ausgabe refreshen

Ausgabe anzeigen

functions/h1.js

index.html

index.js

package.json

```
1 export function h1(){
2
3   const h1Element = document.createElement("h1");
4   h1Element.innerHTML="Success!";
5   return h1Element;
6
7 }
8
```

vxun2p-run.stackblitz.io

Click me

**Success!**

# Show Case – Student View



LernApp

Forum

Admin



Bitte warten Sie. Das Testen dauert ein paar Sekunden.

## Kleine Programmieraufgabe

▼ Aufgabe: Success

To Do

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.

Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Aufgabe Validieren

Code Speichern

↻ Ausgabe refreshen

Ausgabe anzeigen

functions/h1.js

index.html

index.js

package.json

```
1 export function h1(){
2   const h1Element = document.createElement("h1");
3   h1Element.innerHTML="Success!";
4   return h1Element;
5
6 }
7
```

← → ↻ p2wmag--run.stackblitz.io

Click me

**Success!**

# Show Case – Student View



LernApp



Forum

Admin



## Testergebnis

Aufgabe bestanden! Gratuliere!

Schließen

## Kleine Programmieraufgabe

▼ Aufgabe: Success

Gelöst

Klickt der Nutzer auf den Button, so wird eine Überschrift als H1 Element in die Seite eingefügt und zeigt die Meldung "Success!" an. Solange keine Interaktion stattgefunden hat soll das Element nicht auf der Seite vorzufinden sein.

Ihre Aufgabe ist diese Funktionalität in den vorliegenden JavaScript Dateien umzusetzen.

Aufgabe Validieren

Code Speichern

Ausgabe refreshen

Ausgabe anzeigen

# Conclusion



- JavaScript learning application for E2E testing
- Support for multiple components
- Docker Swarm for testing architecture
- Cypress as testing framework
- Tool for teaching but success depends on test cases
- Prototypical implementation (focus on architecture, not interfaces)

# Limitations and Future Work



- Limited number of packages for tasks
- High cost of maintenance & computational facilities
- Only prototypical implementation
- No white-box testing
- Assessment of approach

# References



- [Be14] Bernstein, D.: Containers and Cloud: From LXC to Docker to Kubernetes. IEEE Cloud Comput. 1/3, pp. 81–84, 2014, url: <https://doi.org/10.1109/MCC.2014.51>
- [Bu15] Bui, T.: Analysis of Docker Security. CoRR abs/1501.02967/, 2015, arXiv: 1501.02967, url: <http://arxiv.org/abs/1501.02967>
- [Cyp] Cypress.io. <https://www.cypress.io> (Accessed: 28.10.2021)
- [Doc] Docker, <https://www.docker.com> (Accessed: 28.10.2021)
- [Ru06] Runeson, P.: A Survey of Unit Testing Practices. IEEE Softw. 23/4, pp. 22–29, 2006, url: <https://doi.org/10.1109/MS.2006.91>



# Thank you!