

# DevOps, Software Evolution and Software Maintenance

**IT University of Copenhagen**

Thomas Tyge Andersen  
thta@itu.dk

Ask Harup Sejsbo  
asse@itu.dk

Joakim Hinnerkov  
jhhi@itu.dk

Petya Buchkova  
pebu@itu.dk

Kasper Olsen  
kols@itu.dk

May 4, 2021

# Contents

<b>1</b>	<b>System's Perspective</b>	<b>3</b>
1.1	Design of your ITU-MiniTwit systems . . . . .	3
1.2	Architecture of your ITU-MiniTwit systems . . . . .	3
1.3	Important interactions of subsystems . . . . .	3
<b>2</b>	<b>Process' perspective</b>	<b>4</b>
2.1	How do you interact as developers? . . . . .	4
2.1.1	How is the team organized? . . . . .	4
2.2	Stages and tools . . . . .	4
2.3	Organization of repositories . . . . .	4
2.4	Applied branching strategy - Kasper . . . . .	4
<b>3</b>	<b>Lessons learned perspective</b>	<b>5</b>
3.1	Evolution and refactoring . . . . .	5
3.2	Operation . . . . .	5
3.3	Maintenance . . . . .	5

# **1 System's Perspective**

Automatic CI build test

## **1.1 Design of your ITU-MiniTwit systems**

## **1.2 Architecture of your ITU-MiniTwit systems**

## **1.3 Important interactions of subsystems**

## **2 Process' perspective**

### **2.1 How do you interact as developers?**

#### **2.1.1 How is the team organized?**

### **2.2 Stages and tools**

### **2.3 Organization of repositories**

### **2.4 Applied branching strategy - Kasper**

### **3 Lessons learned perspective**

#### **3.1 Evolution and refactoring**

#### **3.2 Operation**

#### **3.3 Maintenance**