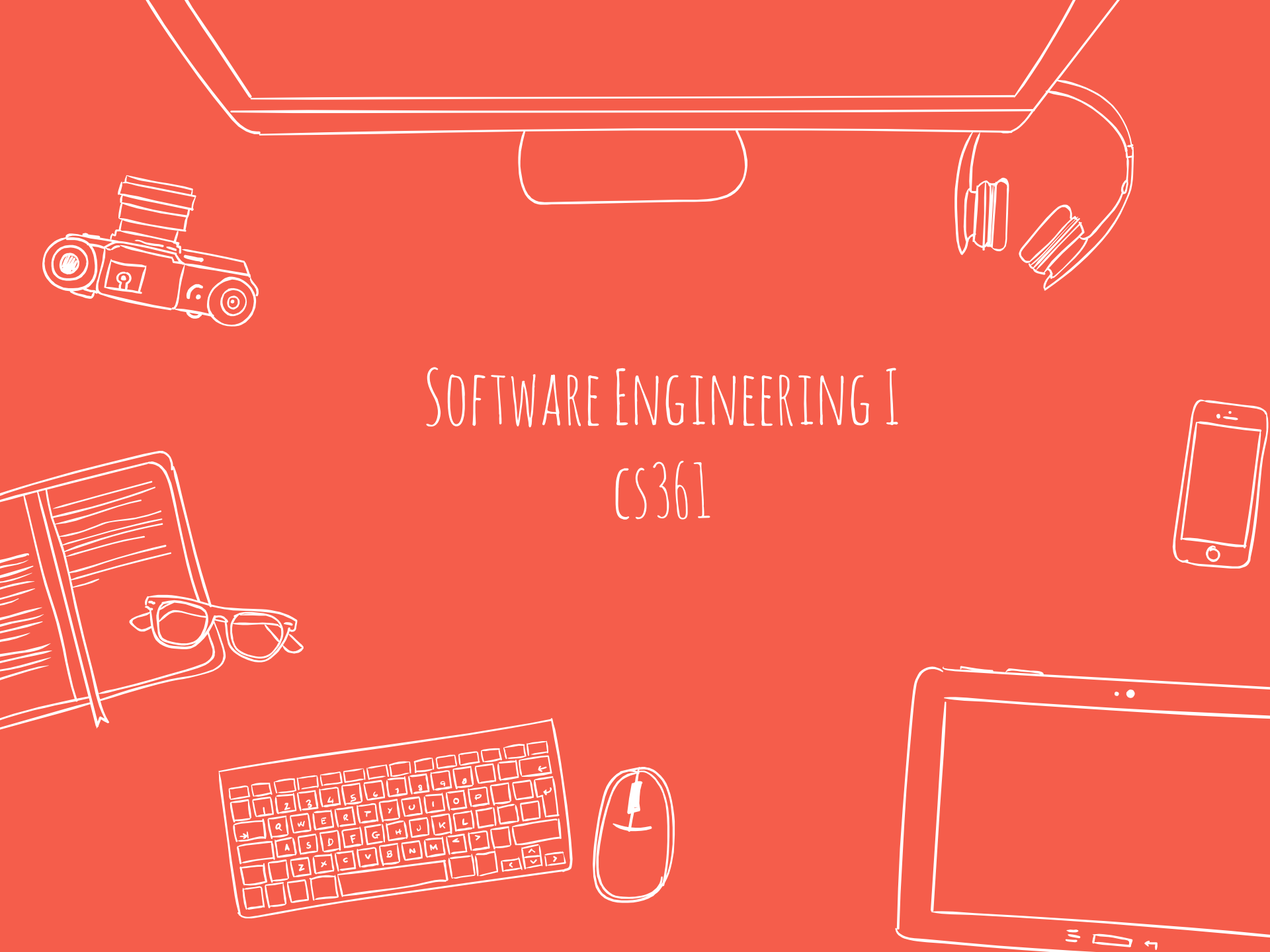


# SOFTWARE ENGINEERING I

## CS361

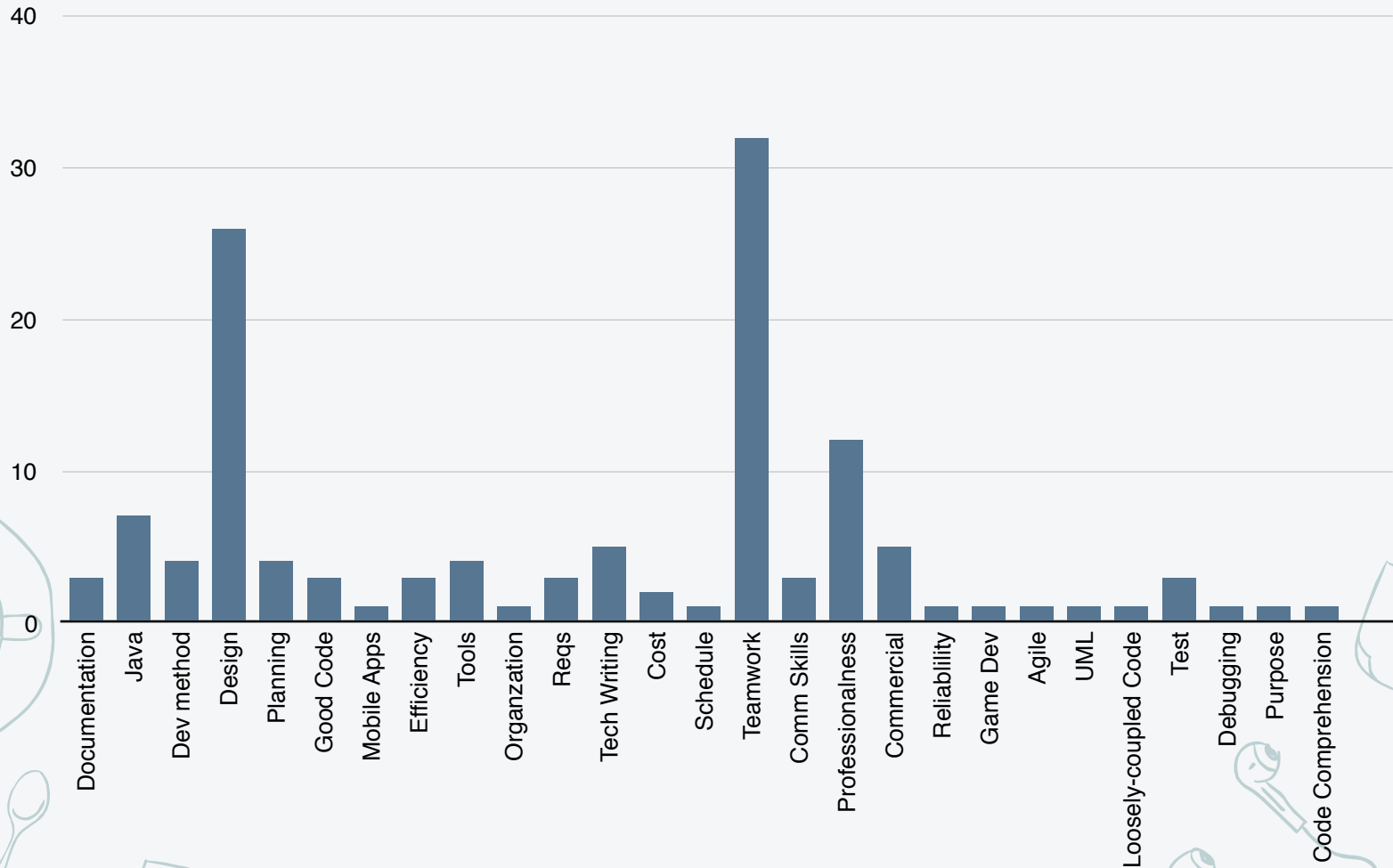




# ASSIGNMENT 0

- ✘ <http://web.engr.oregonstate.edu/~hiltonm/classes/cs361/assignment0.html>
- ✘ Individual, Creating the pull request gets credit

# CLASS EXPECTATIONS



# COLLABORATIVE CODE DEVELOPMENT





# INDIVIDUAL GITHUB

## 1. Fork Project



# INDIVIDUAL GITHUB

1. Fork Project
2. Clone Repo



# INDIVIDUAL GITHUB

1. Fork Project
2. Clone Repo
3. Make Changes

A light blue line-art illustration serves as a background, featuring various items: a computer keyboard in the top left, a USB drive, a pen and a notepad with a drawing in the top right, a smartphone in the middle right, a cup of coffee on a saucer with a spoon in the bottom left, a pair of glasses, and a pair of earbuds with a charging cable in the bottom right.

## INDIVIDUAL GITHUB

1. Fork Project
2. Clone Repo
3. Make Changes
4. Commit Locally





## INDIVIDUAL GITHUB

1. Fork Project
2. Clone Repo
3. Make Changes
4. Commit Locally
5. Push to Repo



## INDIVIDUAL GITHUB

1. Fork Project
2. Clone Repo
3. Make Changes
4. Commit Locally
5. Push to Repo
6. Pull Request to Fork  
(optional)



## PROBLEMS AT SCALE

1. Where is the “truth”?
  - What do I grade?
  - What gets deployed?
  - Who do I pull from?
2. Who has to merge?
3. What do I grade?



# GITHUB FLOW

## 1. Team Repository Contains the Truth



## GITHUB FLOW

1. Team Repository Contains the Truth
2. Each Team member forks from the Team Repo



## GITHUB FLOW

1. Team Repository Contains the Truth
2. Each Team member forks from the Team Repo
3. Each Team Member does commit to main, instead used feature branch(es)



## GITHUB FLOW

4. Feature is developed locally in feature branch
5. Once feature is developed, pull from Team master, merge any changes
6. Submit Pull Request to Team Master

The background features a collection of light blue line-art illustrations. In the top left is a portion of a computer keyboard. To its right is a USB drive. Further right is a pen resting on a notepad with some scribbles. In the bottom left, there is a cup of coffee on a saucer, a spoon, and a pair of glasses. In the bottom right, there is a smartphone, a small square object, and a pair of earbuds with their cables.

## GITHUB FLOW

7. Pull Request reviewed by other team members.





## GITHUB FLOW

7. Pull Request reviewed by other team members.
8. Feature branch can be checked out to evaluate changes if needed



## GITHUB FLOW

7. Pull Request reviewed by other team members.
8. Feature branch can be checked out to evaluate changes if needed
9. Accepted by third party



## GITHUB FLOW

7. Pull Request reviewed by other team members.
8. Feature branch can be checked out to evaluate changes if needed
9. Accepted by third party
10. Team master contains “truth”
  1. Run CI
  2. Deploy
  3. Tag Milestone



## ISSUE TRACKING

# ✕ Create Issues on Github



## ISSUE TRACKING

- ✖ Create Issues on Github
- ✖ Clearly describe issue



## ISSUE TRACKING

- ✖ Create Issues on Github
- ✖ Clearly describe issue
- ✖ Tag issue with “Label” – [bug, duplicate, enhancement, help wanted, invalid, question, wontfix]

The background features a light blue line-art illustration of various items: a keyboard in the top left, a USB drive, a pen and a notepad with a drawing in the top right, a smartphone on the right side, a coffee cup on the bottom left, a spoon, a pair of glasses, and a pair of earbuds with a cable at the bottom right.

## ISSUE TRACKING

- ✖ Create Issues on Github
- ✖ Clearly describe issue
- ✖ Tag issue with “Label” – [bug, duplicate, enhancement, help wanted, invalid, question, wontfix]
- ✖ Assign Issue to team member



## ISSUE TRACKING

- ✖ Create Issues on Github
- ✖ Clearly describe issue
- ✖ Tag issue with “Label” – [bug, duplicate, enhancement, help wanted, invalid, question, wontfix]
- ✖ Assign Issue to team member
- ✖ Close Issue by referencing the #number in Pull Request.





PROPER COLLABORATION WILL BE PART  
OF ALL ASSIGNMENTS GOING FORWARD



# TEST DRIVEN DEVELOPMENT



The background features a collection of light blue line-art illustrations. In the top left is a portion of a computer keyboard. To its right is a USB drive. Further right is a pen resting on a notepad with some scribbles. In the bottom left, there is a cup of coffee on a saucer, a spoon, and a pair of glasses. In the bottom right, there is a smartphone, a small square object, and a pair of earbuds with their cables.

## WHAT IS TEST DRIVEN DEVELOPMENT (TDD)

**Test-driven development (TDD)** is a **software development process** based on three simple rules



## TDD RULES (FROM UNCLE BOB MARTIN)

1. You are not allowed to write any production code unless it is to make a failing unit test pass



## TDD RULES (FROM UNCLE BOB MARTIN)

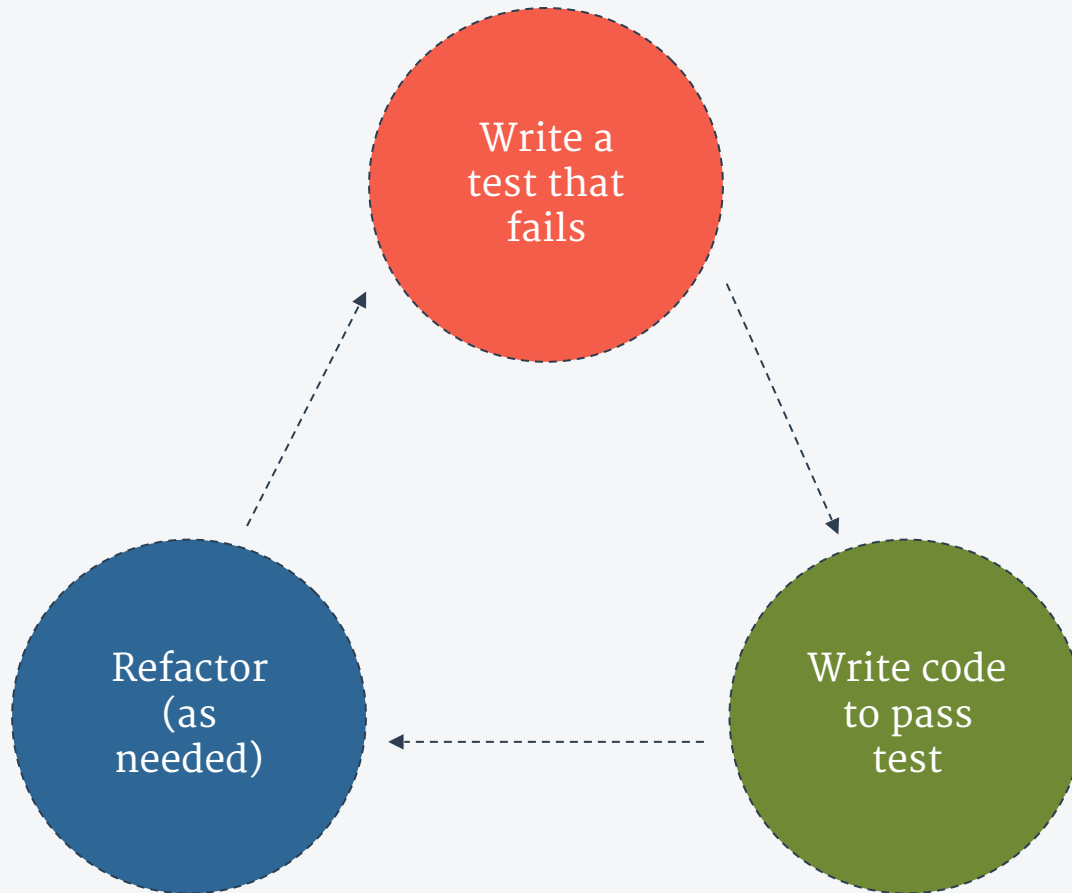
1. You are not allowed to write any production code unless it is to make a failing unit test pass
2. You are not allowed to write any more of a unit test than is sufficient to fail; and any compilation failures are failures



## TDD RULES (FROM UNCLE BOB MARTIN)

1. You are not allowed to write any production code unless it is to make a failing unit test pass
2. You are not allowed to write any more of a unit test than is sufficient to fail; and any compilation failures are failures
3. You are not allowed to write any more production code than is sufficient to pass the one failing unit test

## TDD CYCLE





## WHY TDD?

“The act of writing a unit test is more an act of design than of verification.





## WHY TDD?

“The act of writing a unit test is more an act of design than of verification.

It is also more an act of documentation than of verification.



## WHY TDD?

“The act of writing a unit test is more an act of design than of verification.

It is also more an act of documentation than of verification.

The act of writing a unit test closes a remarkable number of feedback loops, the least of which is the one pertaining to verification of function”

-Uncle Bob in  
Agile Software Development, Principles, Patterns, and Practices



## PERSONAL EXPERIENCE

✕ Clear place to start



## PERSONAL EXPERIENCE

- ✖ Clear place to start
- ✖ Much less code thrown away, less wasted effort





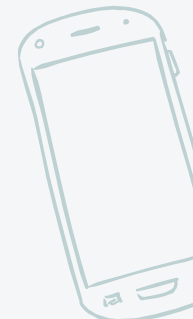

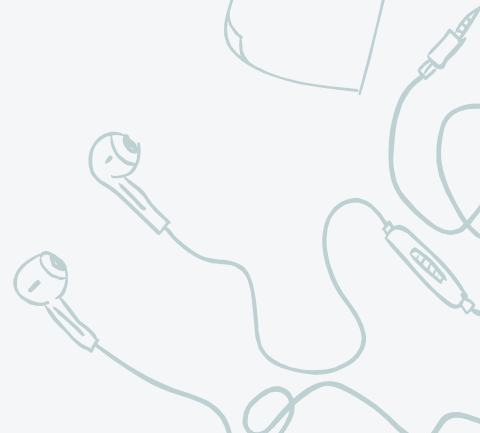


## PERSONAL EXPERIENCE

- ✖ Clear place to start
- ✖ Much less code thrown away, less wasted effort
- ✖ Less hassle with I/O



## PERSONAL EXPERIENCE

- 
- 
- 
- 
- ✖ Clear place to start
  - ✖ Much less code thrown away, less wasted effort
  - ✖ Less hassle with I/O
  - ✖ Less Fear
- 
- 
- 



## DOWNSIDERS TO TDD

- ✖ Reliant on quickly running automated tests
- ✖ No overarching design
- ✖ Effort needed to maintain tests

DEMO TIME







## SCORING BOWLING

The game consists of 10 frames as shown above. In each frame the player has two opportunities to knock down 10 pins. The score for the frame is the total number of pins knocked down, plus bonuses for strikes and spares.

A spare is when the player knocks down all 10 pins in two tries. The bonus for that frame is the number of pins knocked down by the next roll. So in frame 3 above, the score is 10 (the total number knocked down) plus a bonus of 5 (the number of pins knocked down on the next roll.)

A strike is when the player knocks down all 10 pins on his first try. The bonus for that frame is the value of the next two balls rolled.

In the tenth frame a player who rolls a spare or strike is allowed to roll the extra balls to complete the frame. However no more than three balls can be rolled in tenth frame.



## CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- ✖ Presentation template by [SlidesCarnival](#)
- ✖ Photographs by [Unsplash](#)