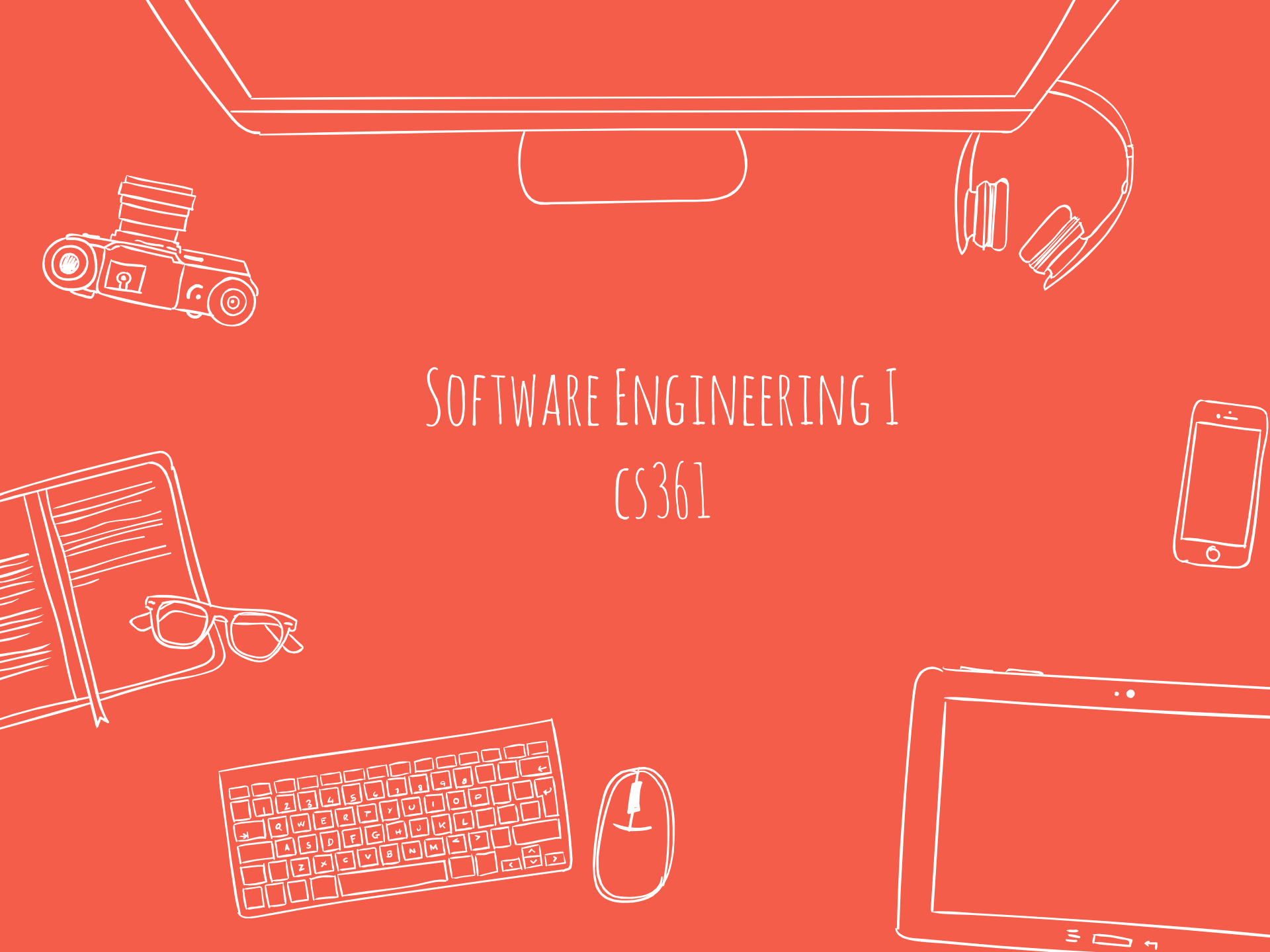
Software Engineering I

CS361



Announcements

- Office hours canceled on Tuesday Jan 19th
- Shane's Tue Office Hours moved to Thurs 11:30-12:30
- How to show you are using TDD
- Sources for Writing Assignment
- Question for Ciera Japan Due on Piazza by by 7pm, Monday Jan 18th.

User Stories

User Stories

- ✖ The Card
- ✖ The Conversation
- ✖ The Confirmation

The Card

"As a [role], I want [function], so that [value]"

Often written on 3x5 card

Examples:

The Conversation

An open dialog between everyone working on the project and the client

Split up Epic Stories if Needed

The Confirmation

A test that will show when task is completed

Could be automated, or a script

Write user Stories

INVEST

I - Independent
N - Negotiable
V - Valuable
E - Estimable
S - Small
T - Testable

Source: Bill Wake http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/

Independent

- ✖ Schedule in any order.
- ✖ Not overlapping in concept
- ✖ Not always possible

Negotiable

✖ Details to be negotiated during development

✖ Good Story captures the Essence, not the details

Valuable

✖ This story needs to have value to someone (hopefully the customer)

✖ Especially relevant to splitting up issues

Estimable

- ✖ Helps keep the size small
- ✖ Ensure we negotiated correctly
- ✖ “Plans are nothing, Planning is everything” –Dwight D. Eisenhower

Small

- ✖ Fit on 3x5 card
- ✖ at most two person-weeks of work
- ✖ Too big == unable to estimate

Testable

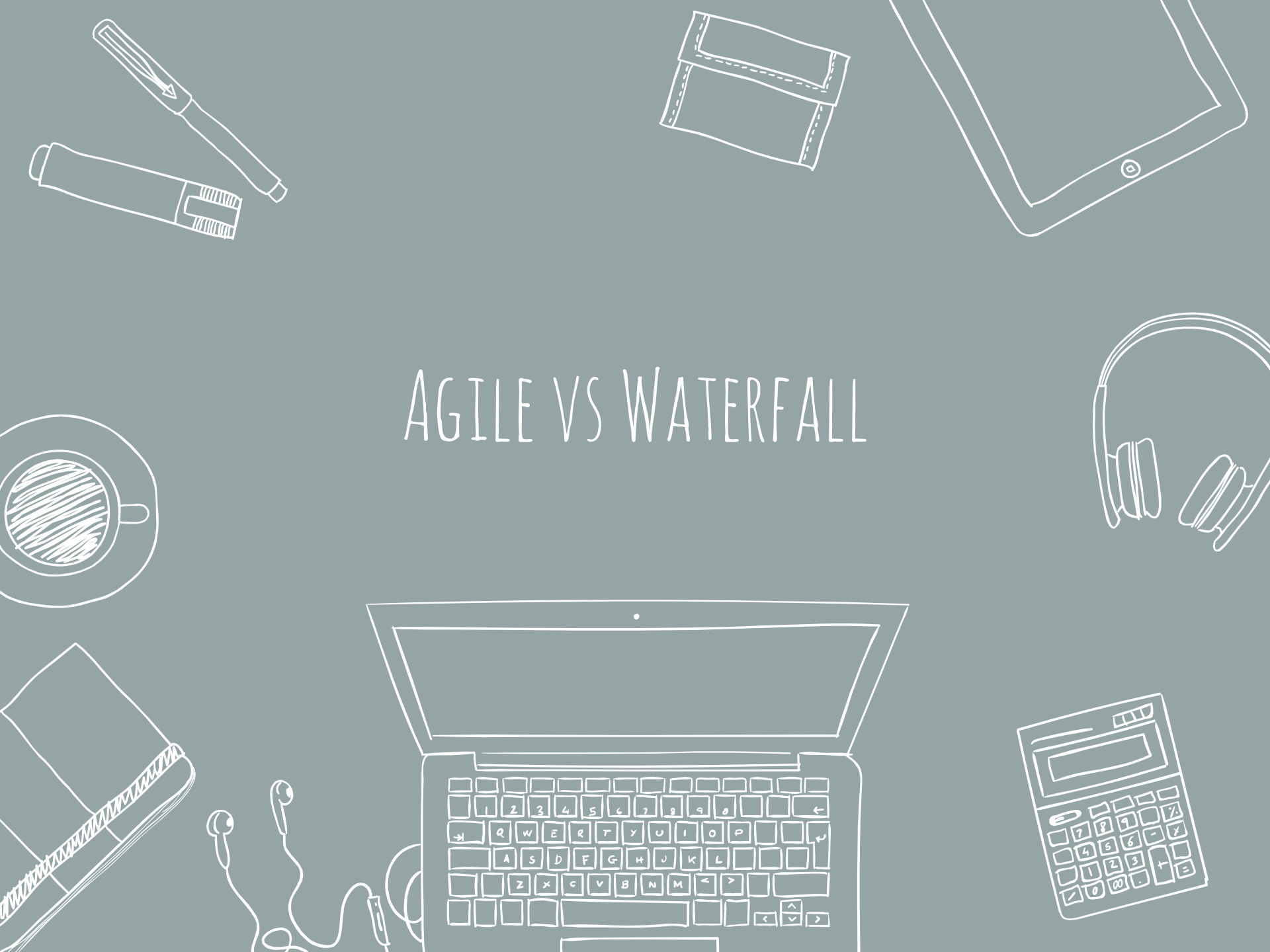
✖ Ensures Understanding of Task

✖ We know when we can mark task “Done”

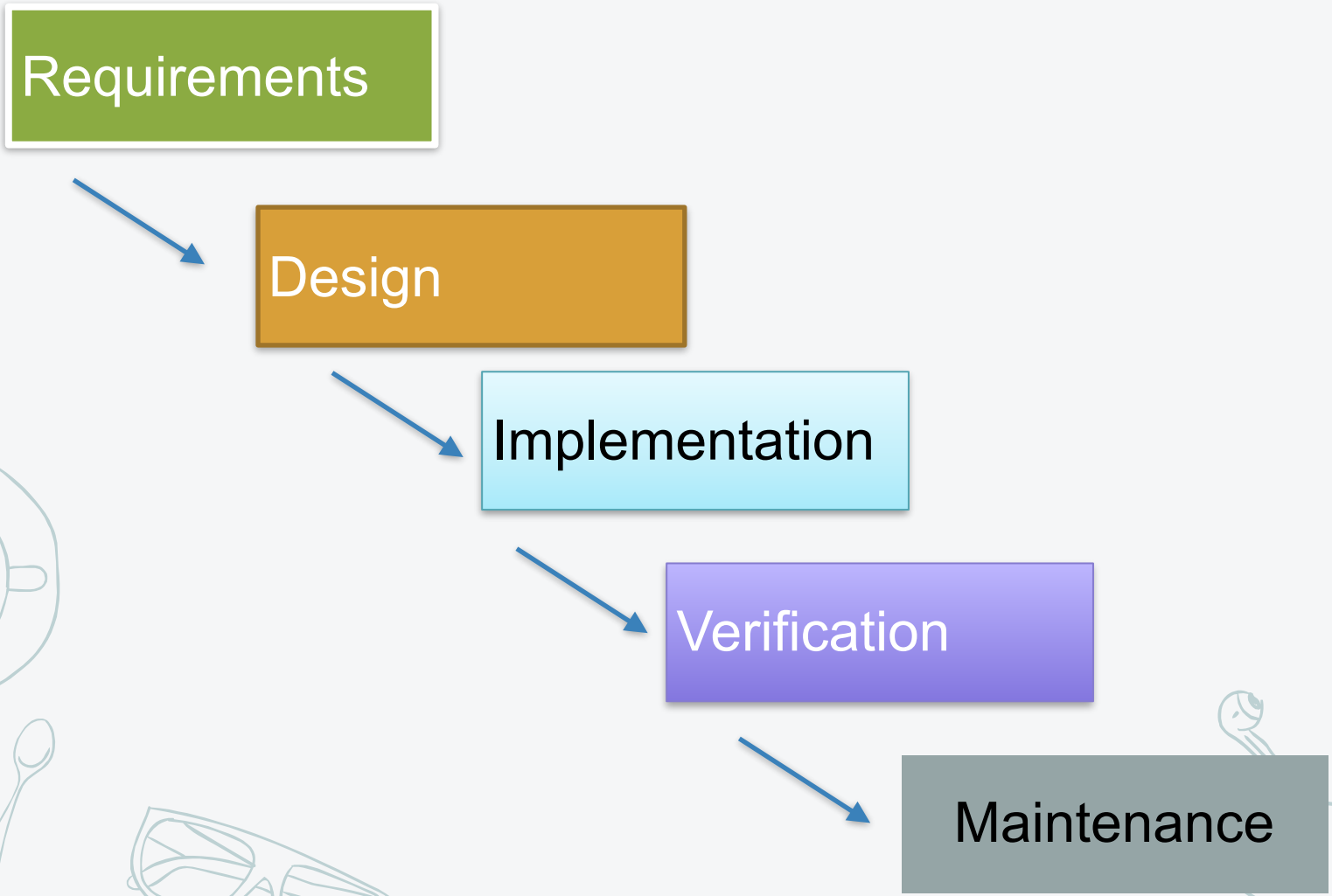
✖ Unable to test == do not understand

INVEST USER STORIES

Agile vs Waterfall



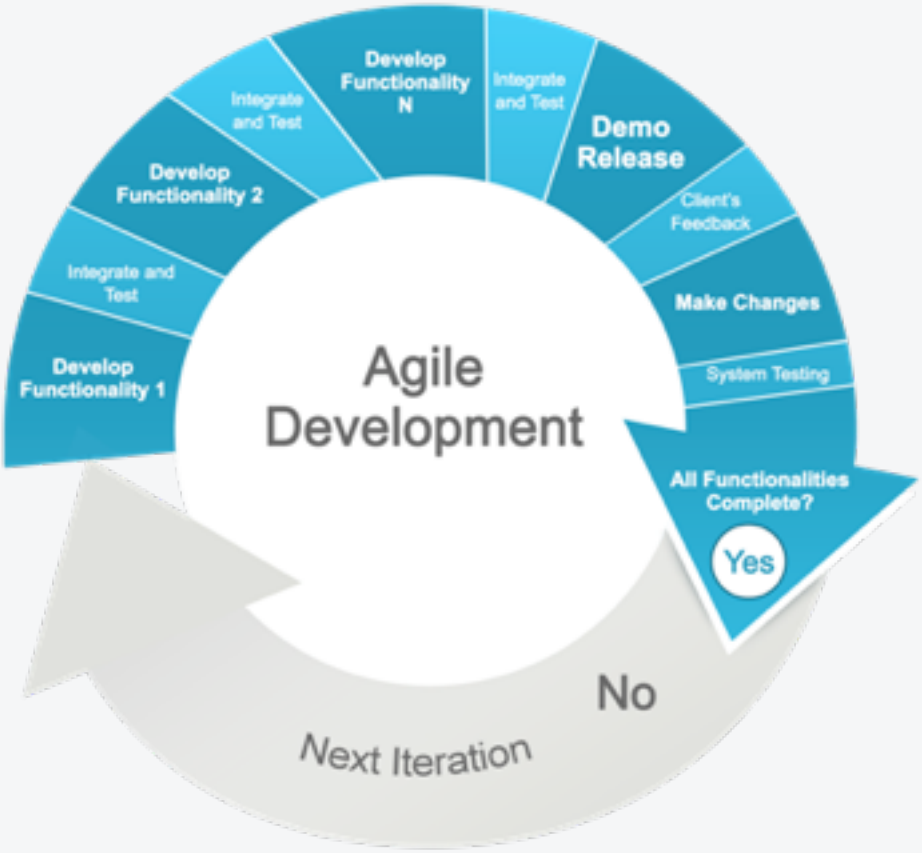
Waterfall Software Development Model



Agile Manifesto

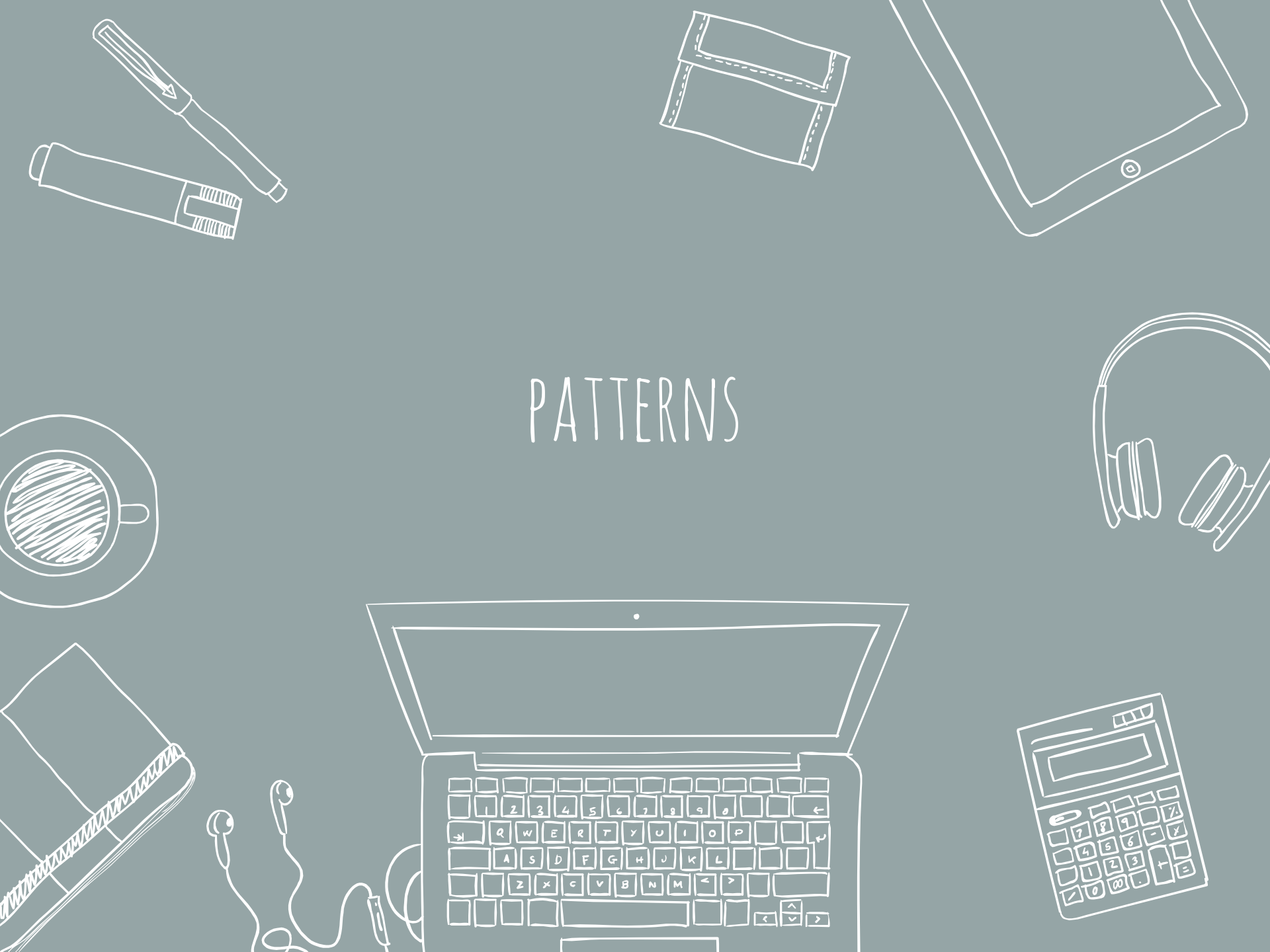
Img Src: http://www.strategybeach.com/our-agile-development-methodology/

Agile Software Development Model



Img Src: http://www.strategybeach.com/our-agile-development-methodology/

PATTERNS



Patterns

- A general, reusable solution to a commonly occurring problem in a given context

- Often have best practices associated with them

Patterns

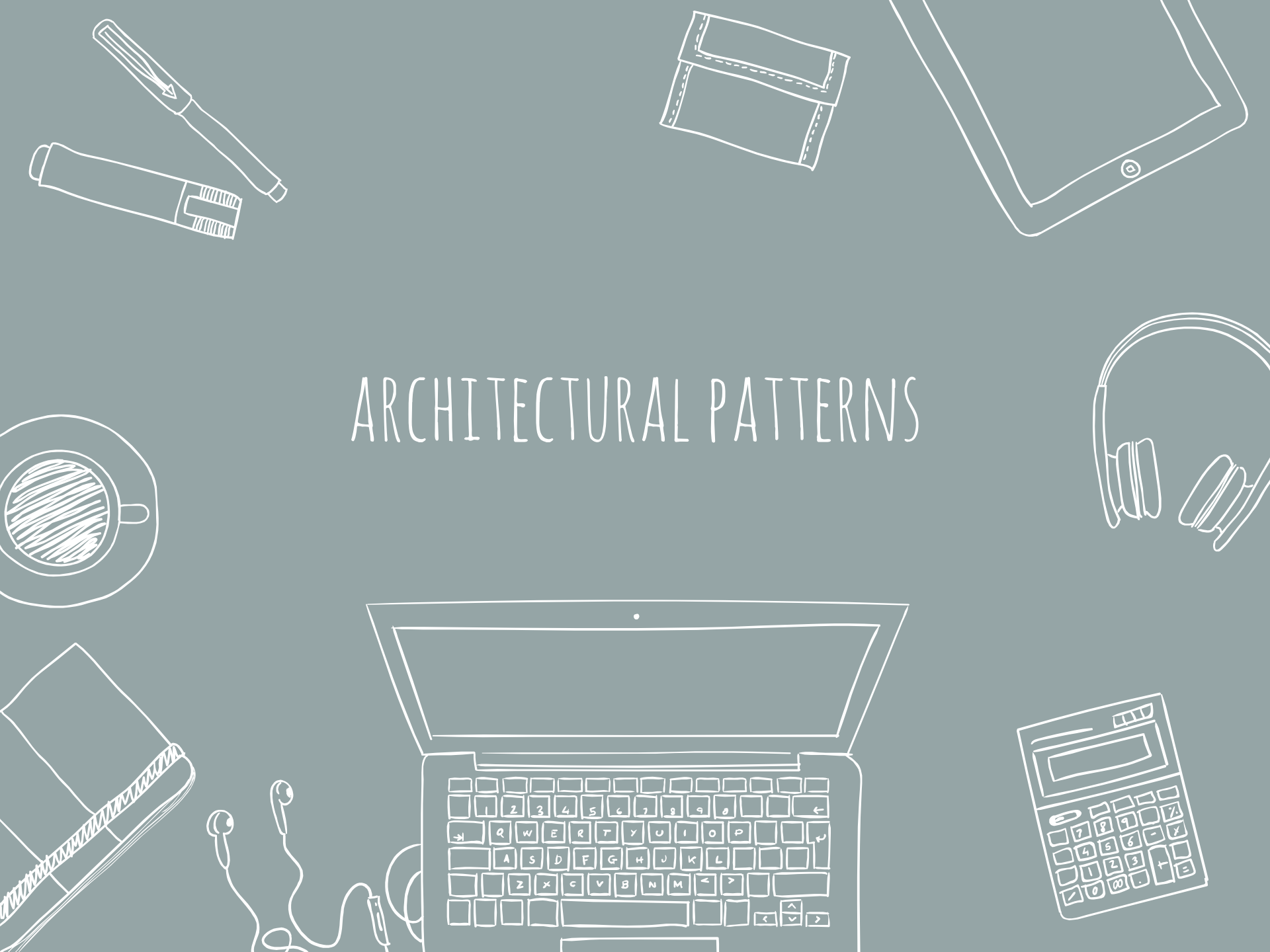
Architectural Patterns

Fundamental structural organization for software systems.

Design Patterns

Solves reoccurring problems in software construction

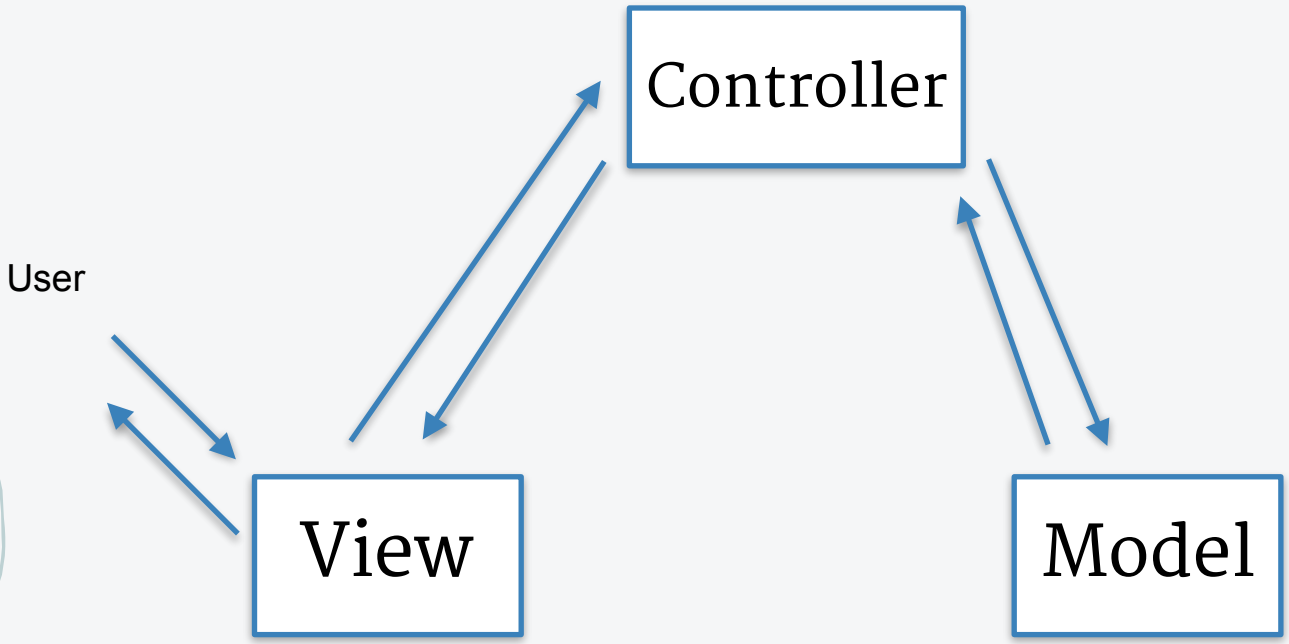
ARCHITECTURAL PATTERNS



Architectural Patterns

- ✖ MVC
- ✖ Event-Driven Architecture
- ✖ Layers
- ✖ Microservices
- ✖ Peer-to-Peer
- ✖ Pipe and Filter
- ✖ Service-Oriented architecture
- ✖ Blackboard

MVC



Common MVC Frameworks

- ✖ Ruby on Rails
- ✖ Spring Framework for Java
- ✖ Django for Python
- ✖ Elm (Functional Reactive Programming Lang)
- ✖ Google Web Toolkit for Java
- ✖ AngularJS for Javascript
- ✖ CodeIgniter for php
- ✖ CakePHP
- ✖ Zend Framework for PHP
- ✖ Play for Scala
- ✖ Ninja Framework for Java

Model

- Contains the Business logic. (application logic and structure)

- Maintains the application specific data

View

- ✖ Renders the model
- ✖ Allows interaction with the user
- ✖ Passes input to the controller

Controller

- ✖ Receives input
- ✖ Makes appropriate calls to the model
- ✖ Updates the view

MVC Demo

Credits

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

- Presentation template by SlidesCarnival
- Photographs by Unsplash