

DevOps + Test Driven Development (TDD)

Charlie Garrod

Michael Hilton

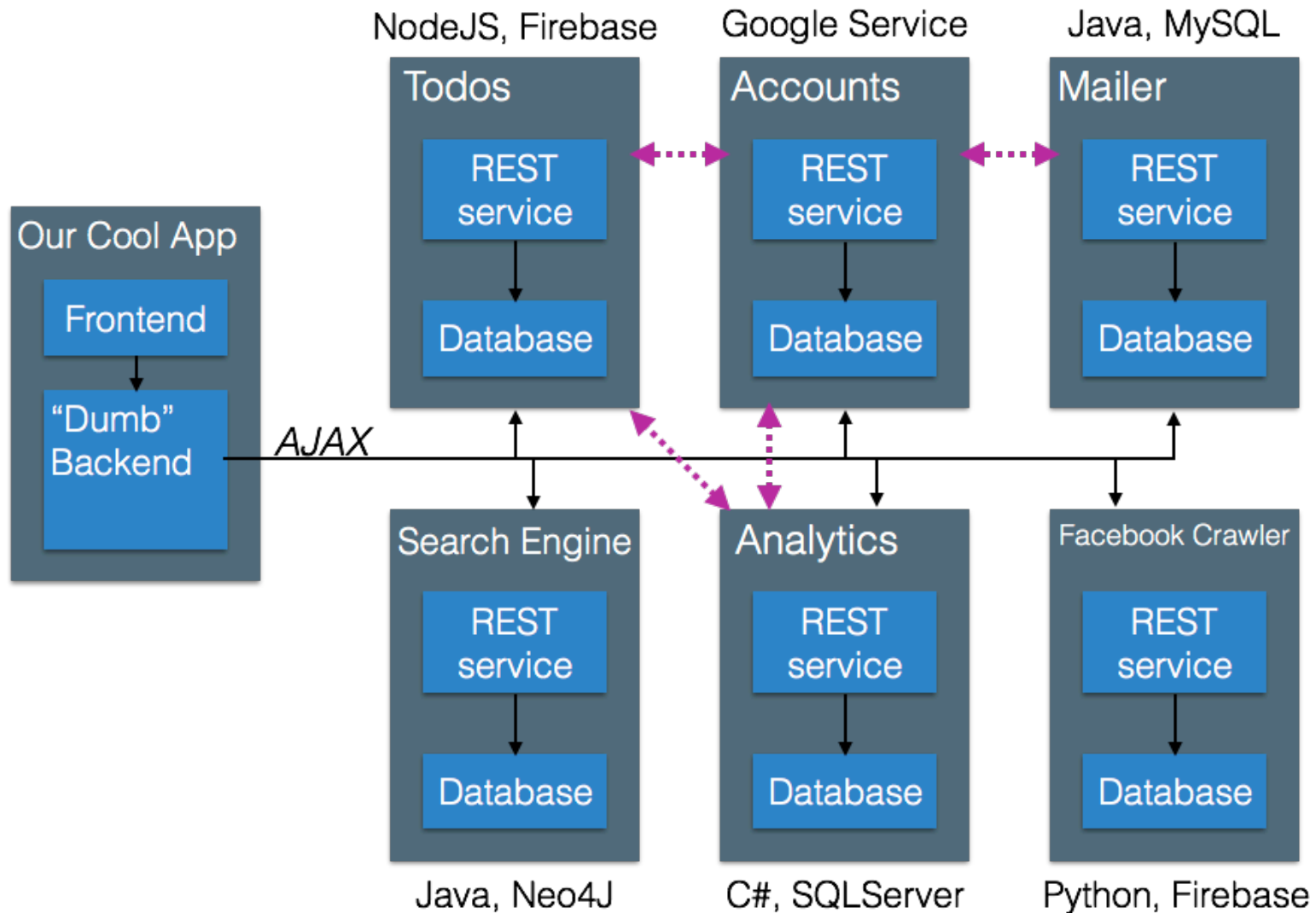
Administrivia

- Homework 6 checkpoint – Monday Dec 4th
- Final Exam Review: Dec 13th, 2-4pm Wean 5409
- Final Exam: Dec 15th, 5:30-8:30pm Wean 7500

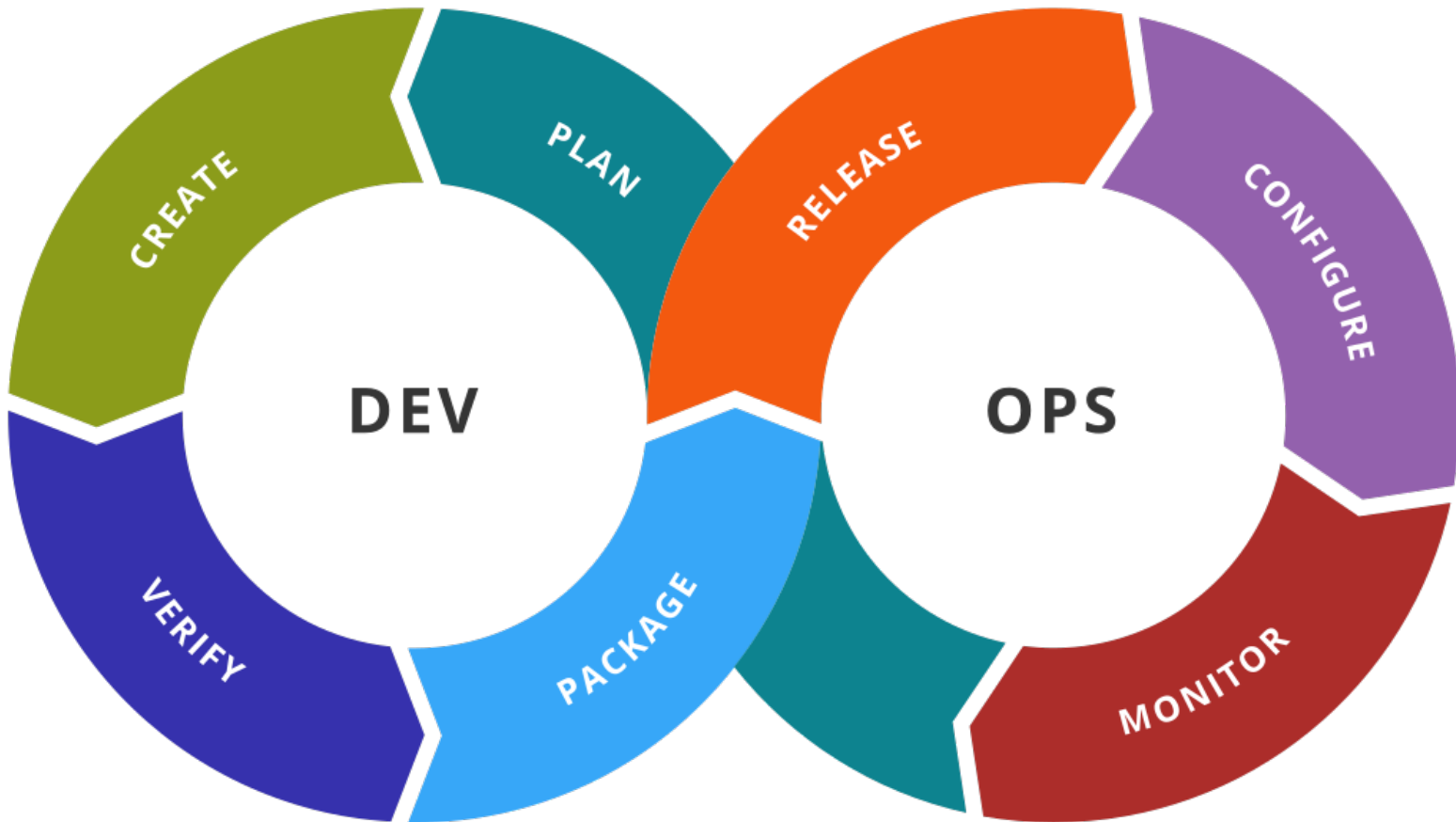
Last Time:

- Microservices and Devops

Microservices rely on DevOps

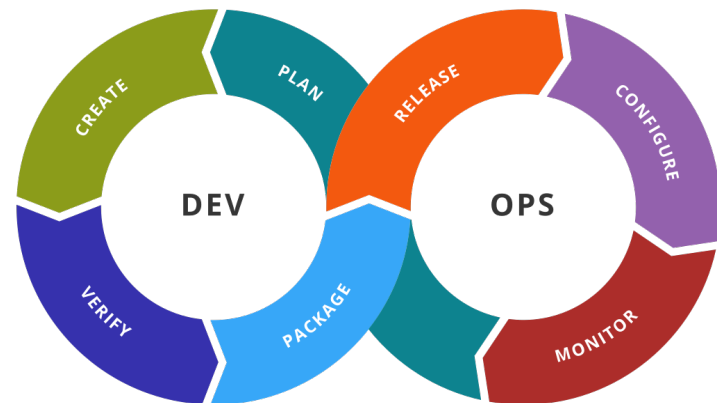


DevOps Toolchain



DevOps Toolchain

- Code — code development and review, source code management tools, code merging
- Build — continuous integration tools, build status
- Test — continuous testing tools that provide feedback on business risks
- Package — artifact repository, application pre-deployment staging



- Continuous testing tools that provide feedback on business risks

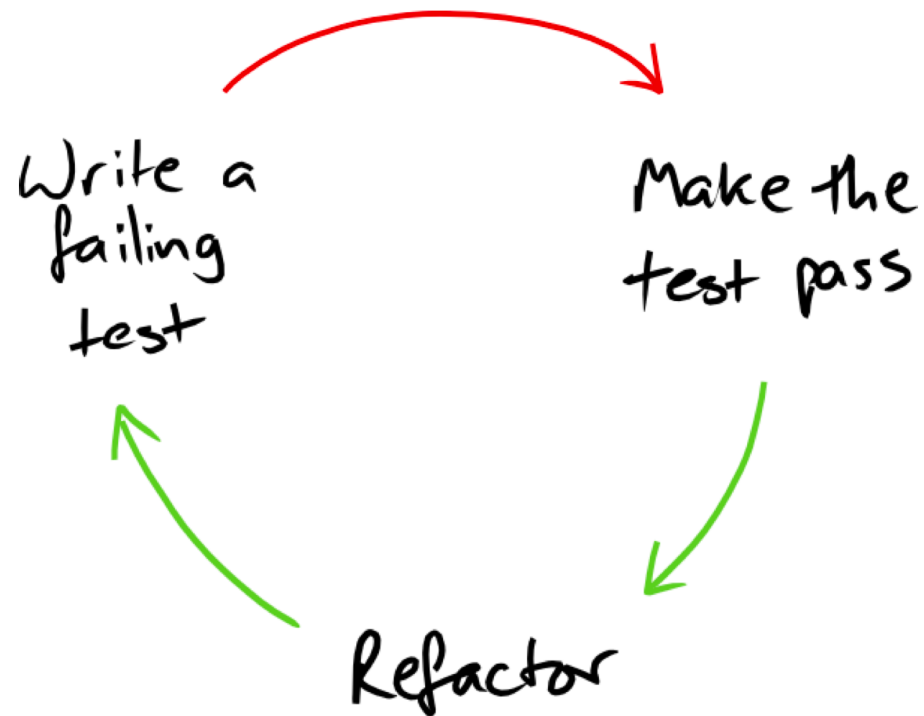
Testing tools must
have **tests** to be
valuable!!

Test Driven Development (TDD)

Three Simple Rules

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 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.

Test Driven Development (TDD)



From Growing Object-Oriented Software by Nat Pryce and Steve Freeman

<http://www.growing-object-oriented-software.com/figures.html>

@sebrose

<http://cucumber.io>

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”.

Advantages of TDD

- Clear place to start
- Much less code thrown away, less wasted effort
- Less Fear
- Side Effect: Robust test suite

TDD DEMO

Scoring Bowling.

1	4	4	5	6	▲	5	▲	■	0	1	7	▲	6	▲	■	2	▲	6
5	14	29	49	60	61	77	97	117	133									

- The game consists of 10 frames. In each frame the player has two rolls. 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.
- A strike is when the player knocks down all 10 pins on their 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.

The Requirements.

- Write a class named “Game” that has two methods
 - roll(pins : int) is called each time the player rolls a ball. The argument is the number of pins knocked down.
 - score() : int is called only at the very end of the game. It returns the total score for that game.

Game
+ roll(pins : int) + score() : int

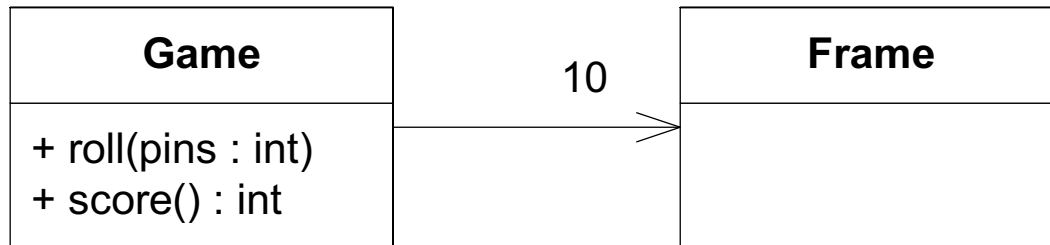
A quick design session

- Clearly we need the Game class.

Game
+ roll(pins : int) + score() : int

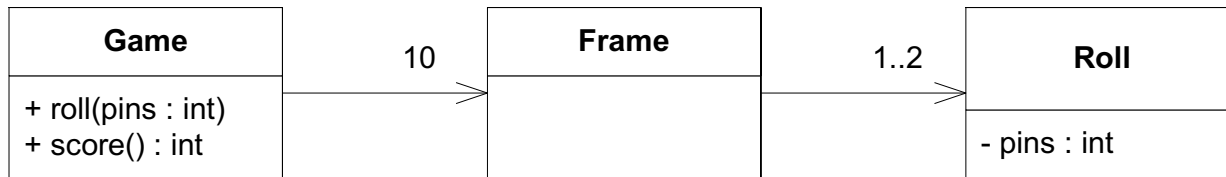
A quick design session

- A game has 10 frames



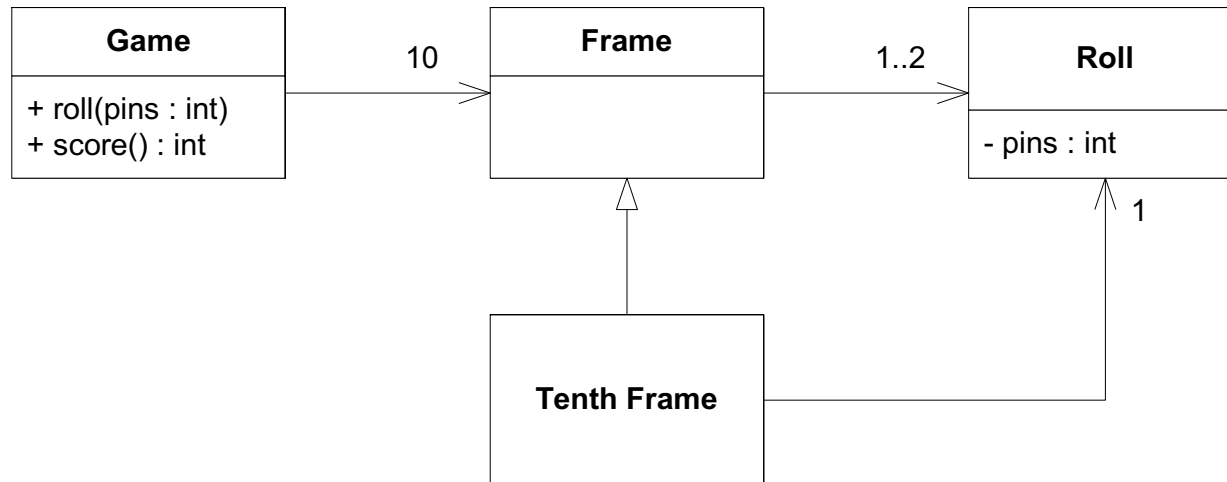
A quick design session

- A frame has 1 or two rolls.



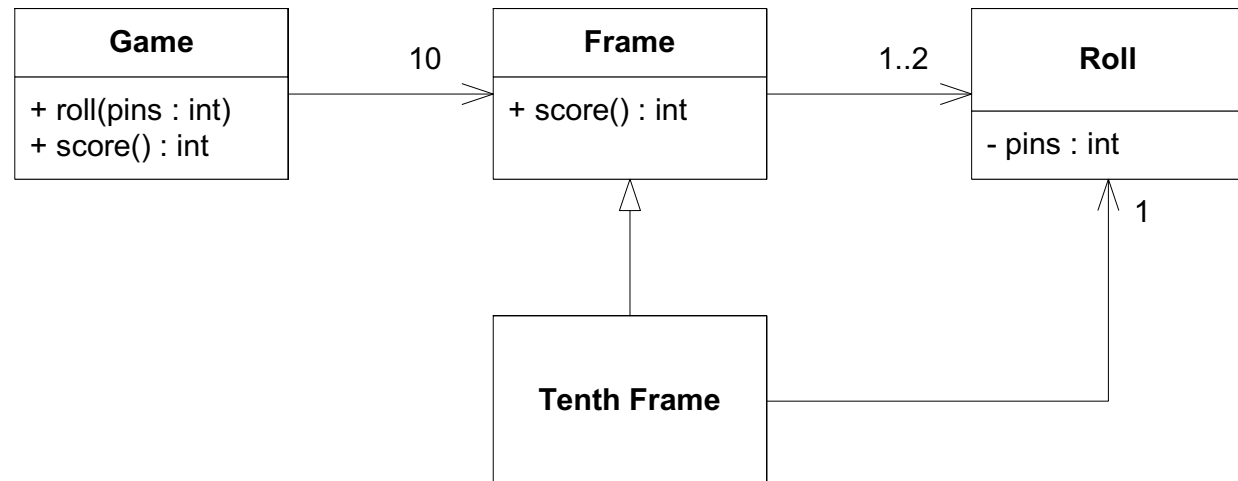
A quick design session

- The tenth frame has two or three rolls.
- It is different from all the other frames.



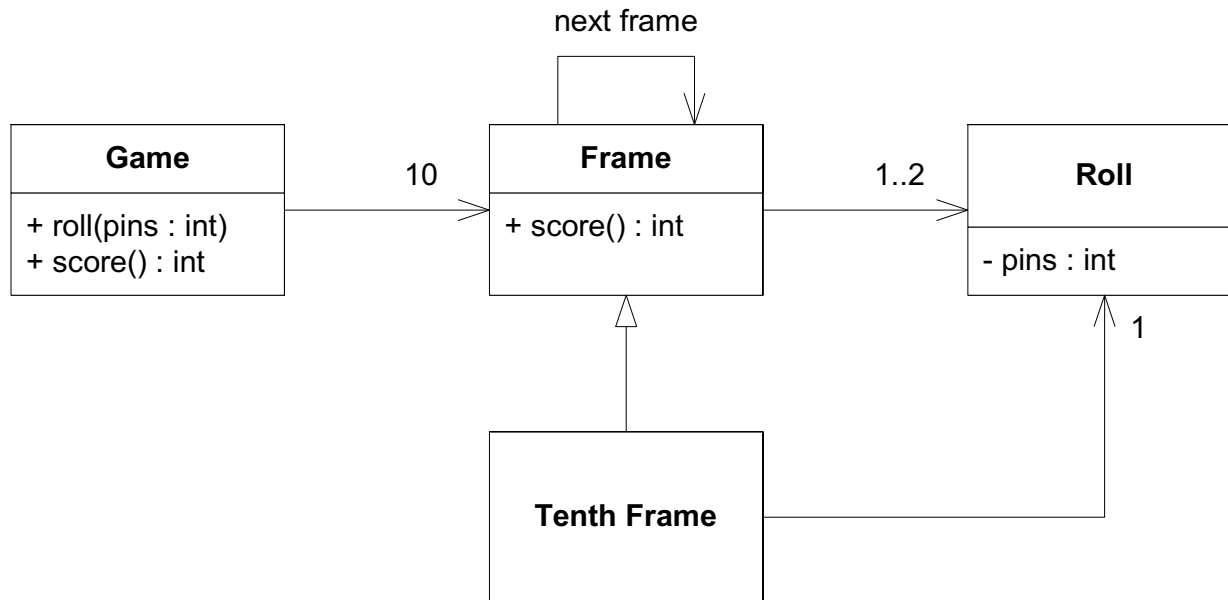
A quick design session

- The score function must
- iterate through all the
- frames, and calculate
- all their scores.



A quick design session

- The score for a spare or a strike depends on the frame's successor



TDD DEMO