ITP2200
Introduction to Software Testing

Bogdan Marculescu

Høyskolen Kristiania

Lecture 9
Testing for Different Technologies

So, more programming this week?

Høyskolen Kristiania

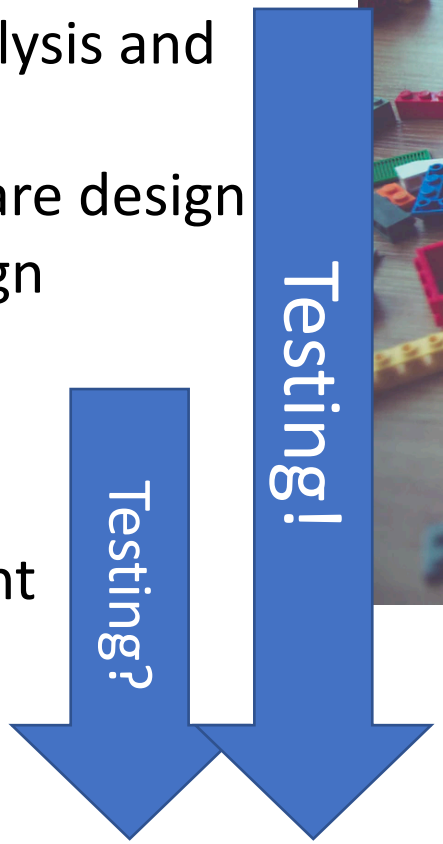
What do we remember from last time(s)?

Høyskolen Kristiania

Software Development Process

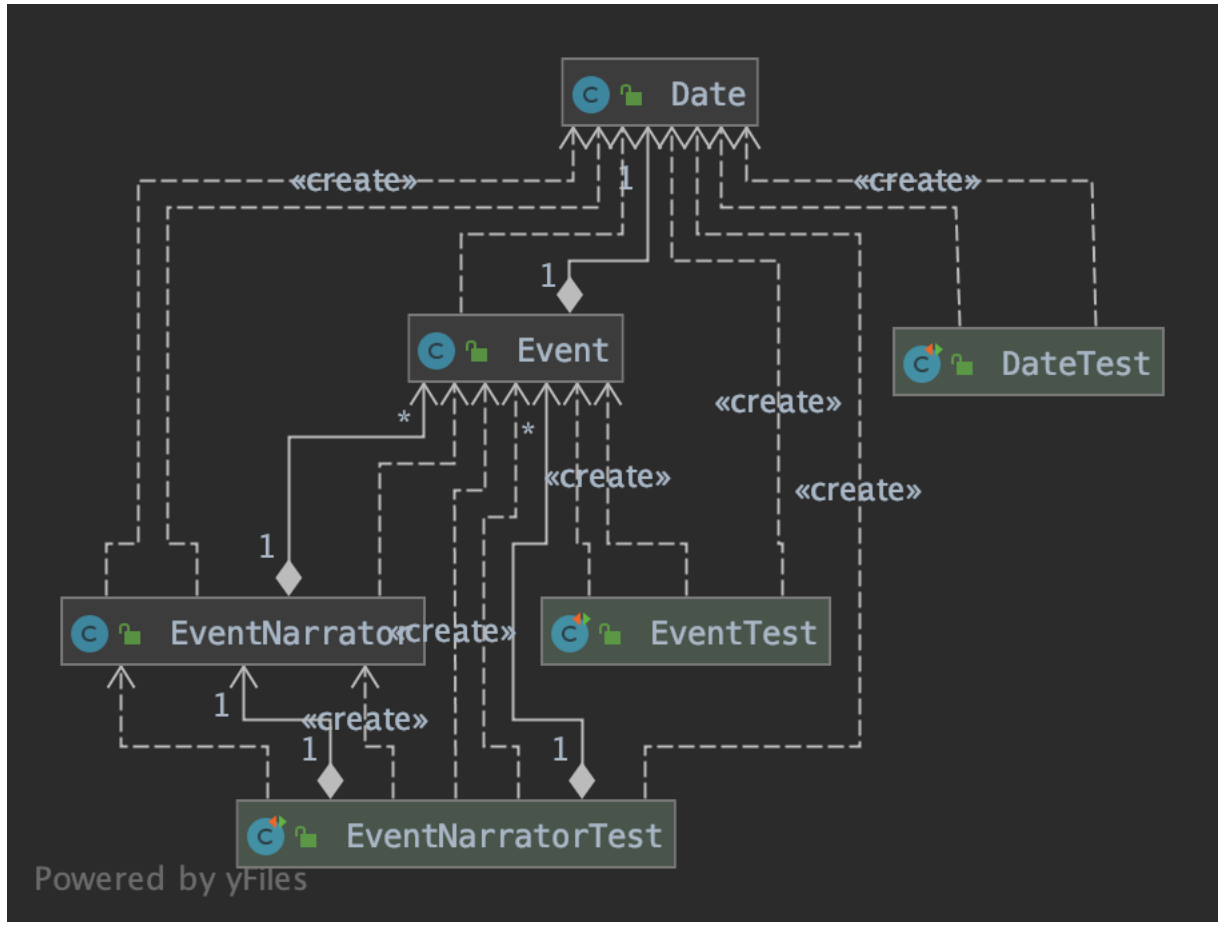
Quick primer on how software development works:

1. Requirements analysis and specification
2. System and software design
3. Intermediate design
4. Detailed design
5. Implementation
6. Integration
7. System deployment
8. Operation and maintenance



Høyskolen Kristiania

Object Oriented



This is familiar!

Unit Tests – at the method and Class level

Integration Tests – building modules

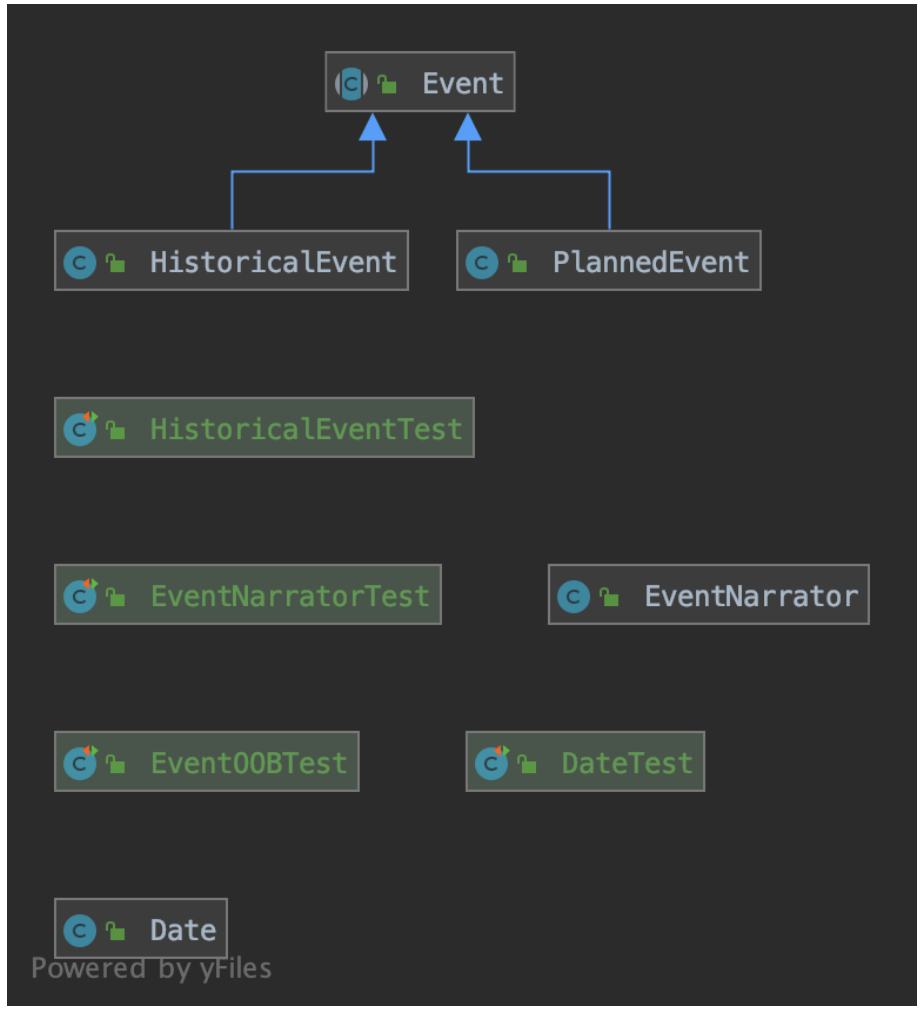
Regression Tests – Re-running old tests to verify new changes

All clear, right?

Who's wondering what the catch is?

Høyskolen Kristiania

Object Oriented



The catch(es):

- **Polymorphism** – how do we test when several classes have the same methods?

- **Complexity** – each method should be tested individually and in combination
    - each class should be tested individually and in combination

Høyskolen Kristiania

Object Oriented

Extra catch:

- **Dynamic binding** – exactly what class, what implementation, what methods and variables are available can change at runtime

- **Inheritance** - overriding methods with different semantics
  - different accessibility of variables

```
private Event[] sillyInit(){
    return new Event[]{new HistoricalEvent( name: "Midway", n
            new HistoricalEvent( name: "Death of Caligula", n
            new HistoricalEvent( name: "Reign of Aurelian", n
            new HistoricalEvent( name: "New Year's Party", ne
            new HistoricalEvent( name: "Online lecture due to
            new PlannedEvent( name: "Summer holidays", new Da
    };
}
```

Høyskolen Kristiania

Object Oriented

Managing complexity:

- **Data flow**
- **Coupling**

Concepts are used in analysing and testing such code.

Clear understanding of design helps.
Clear design also helps.

Høyskolen Kristiania

Høyskolen Kristiania

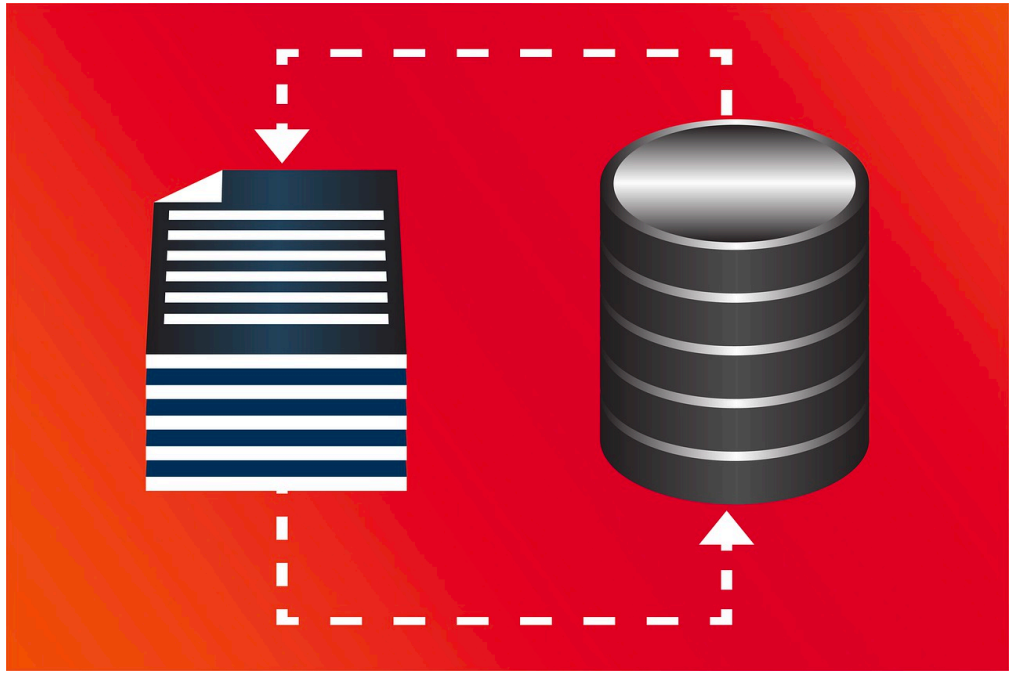
Questions so far?

Web Apps and Web Services

Online apps:

- Websites – html
- Dynamic web applications –javascript
- Web services – Java, Kotlin, etc.

Often, following a RESTful approach.



Høyskolen Kristiania

Web Apps and Web Services

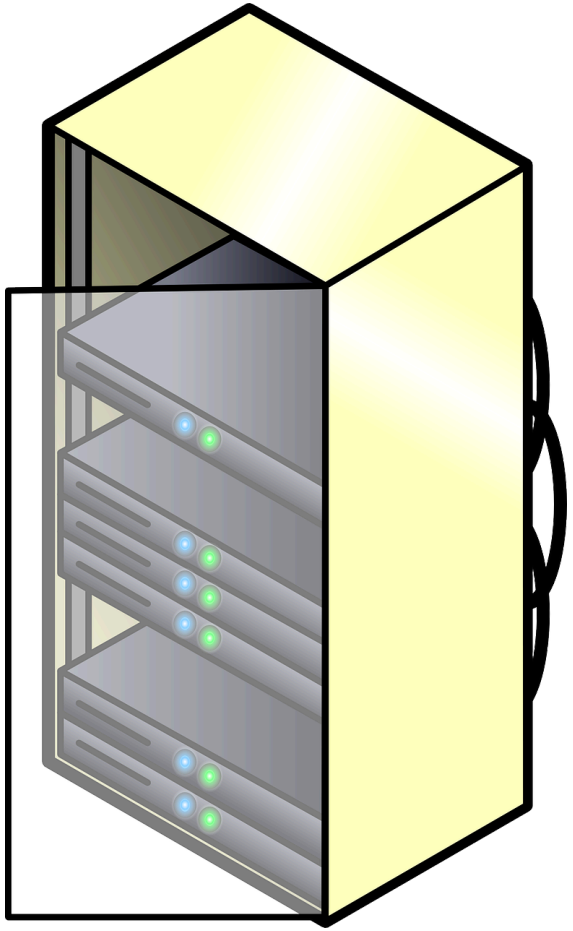
Websites and
client side of dynamic web applications

- Validity of links
- Displayed pages and accessibility
- Security
- Performance

- Often involves subjective components

Høyskolen Kristiania

Web Apps and Web Services



Web services and
The server side of web applications

- Often through HTTP (stateless transfer)
- Often RESTful (at the time of writing)
- Focused on accessing the system via endpoints
- Include service finding, mocking, stubs

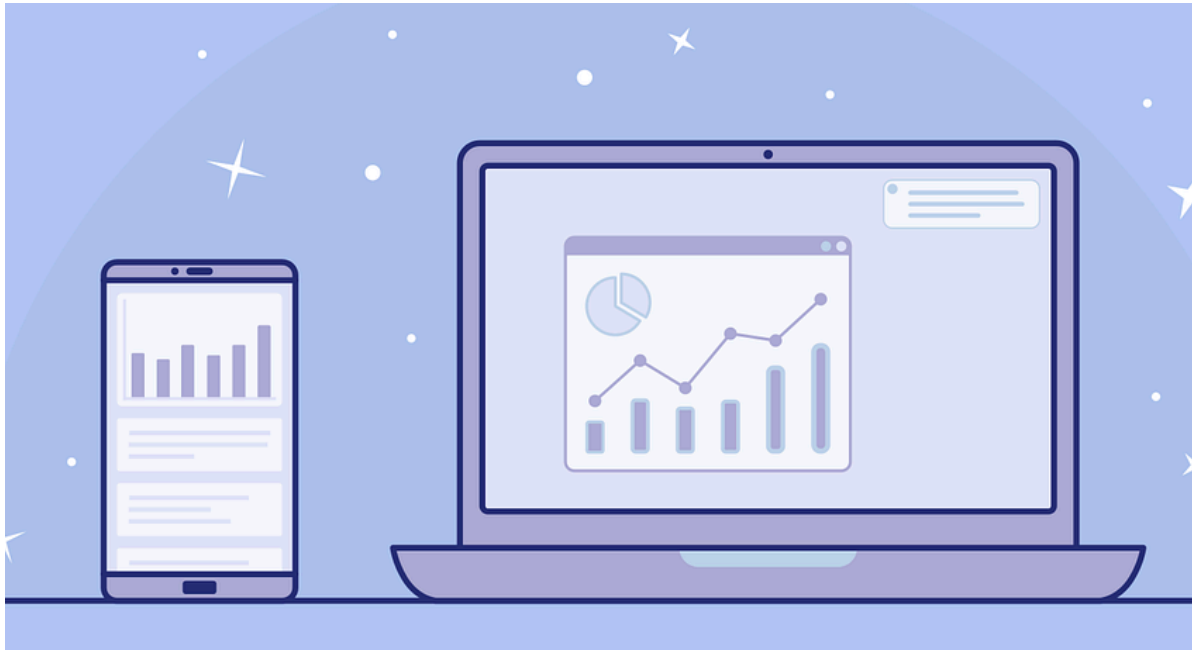
- Book mentions SOAP and WSDL
- Now it’s more JSON and OpenAPI

Høyskolen Kristiania

Høyskolen Kristiania

Questions so far?

Graphical User Interfaces



Focus on:

- User experience, user behaviour
- Capture-replay known behaviours

- Old friends: graphs and finite state machines

Høyskolen Kristiania

Real-time and Embedded Systems

Focus on:

- Observability and reproducibility
- Probe effect (the measurement itself has an impact)

- More focus on timeliness (and strict enforcement of timing constraints).

Høyskolen Kristiania

Høyskolen Kristiania

Questions so far?

Exercise for the Seminar

Online!

- Two submissions
- Good first step. Now, let’s go into detail.
- How can we ensure each requirement is met?
- Are there any changes you feel should be made to the requirements/design
- What bugs did you find and how would you report them?

Høyskolen Kristiania

Exercise for the Seminar

Online!

- Put together a **test plan** for our Narrator
- Focus on the motivation, link the requirements to a clear plan for testing
- Also have tests that are in line with your plan
- End with a section about what is missing, and what should be done next

- Send me the document for comments and feedback (not grades!)

Høyskolen Kristiania

Høyskolen Kristiania