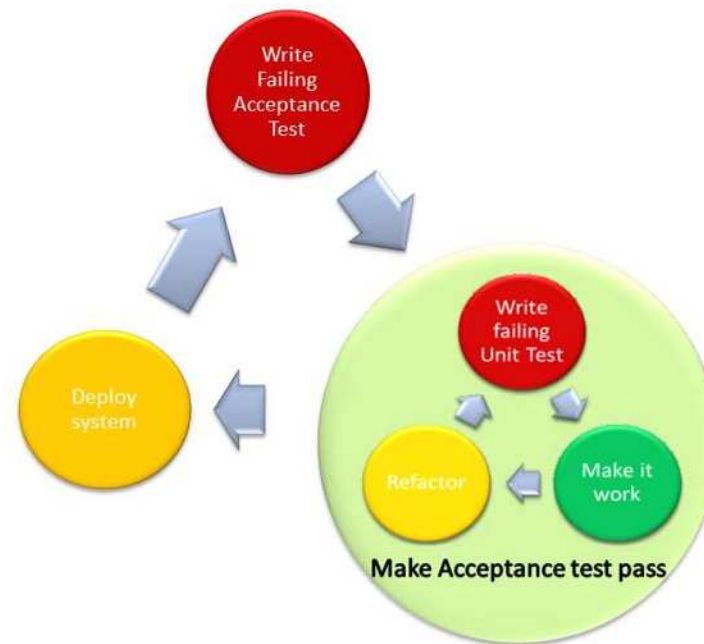


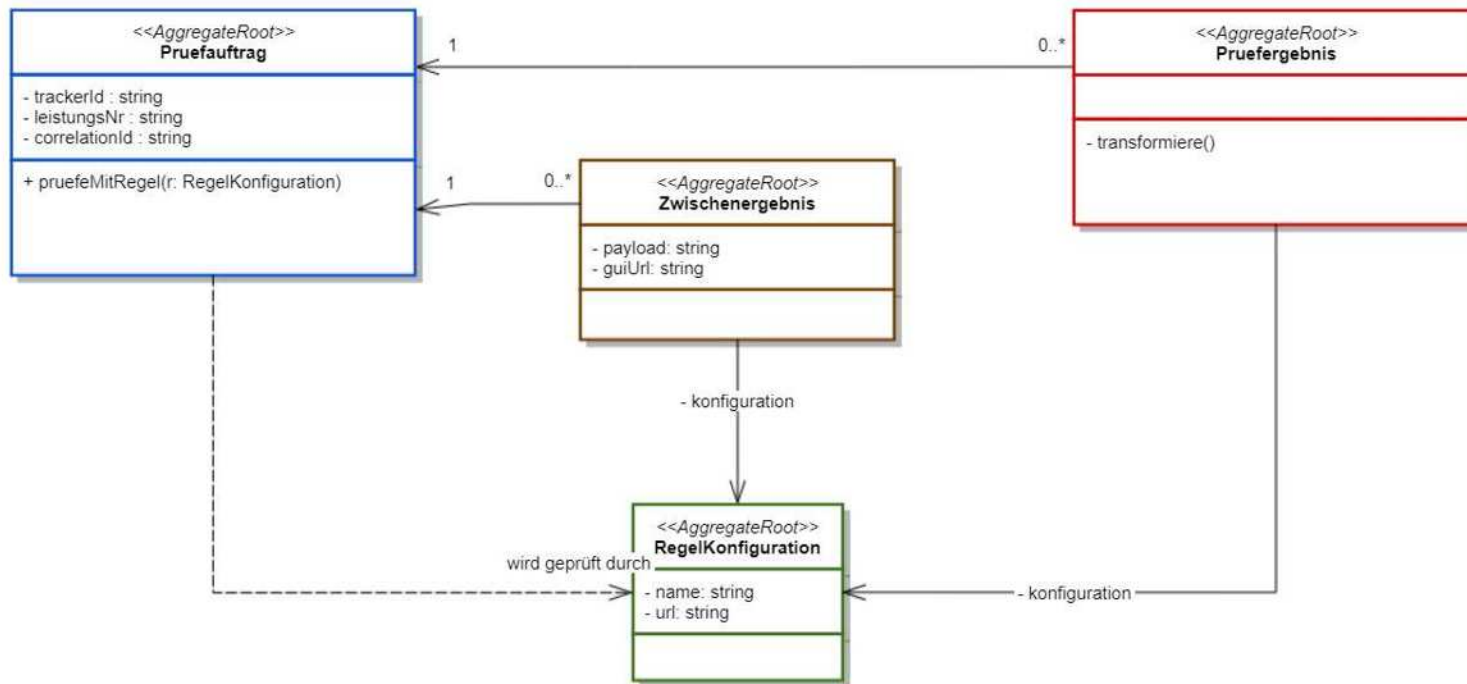


Axolotl is doing ATDD

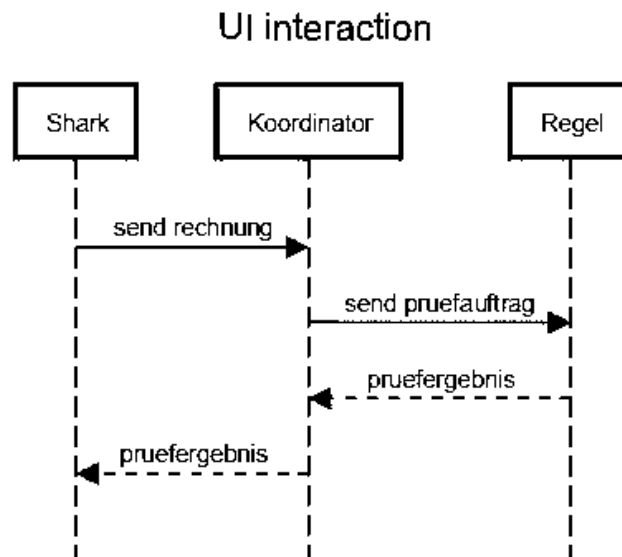
Acceptance test driven development ATDD



Domain Model



Sequencediagram happy case



Before

```
@Test
void testEndToEndMitAsynchronerRegelOhneGui() throws ESAntwortMappingFehlerhaftException {
    when(regelAccessor.sendePruefauftrag(any(), any(), any())).thenReturn(RegelPruefResultat.asynchron());

    final PruefauftragId pruefauftragId = erstellePruefauftrag();

    final ProcessInstance processInstance = verifyProcessInstanceVorhanden(pruefauftragId);

    // Verifizieren, dass noch keine Antwort vorliegt
    final ESReadPruefresultatIn readIn = createReadPruefresultatIn(pruefauftragId.getId());
    verifyNochKeinResultatVorhanden(esDelegate.readPruefresultat(readIn));

    prozessWeiterfuehren(processInstance, 3);

    // Verifizieren, dass Regel beauftragt wurde
    verify(regelAccessor).sendePruefauftrag(any(Pruefauftrag.class), any(), argThat(new RegelKonfigurationMatcher(REGEL_KONFIGURATION_ID_AKTIV)));

    // Query-API verifizieren
    final Response queryOut = pruefauftragResource.getPruefauftrag(pruefauftragId.getId(), request);
    verifyQueryApiResponse(queryOut);

    // Pruefergebnis von Regelkomponente faken
    final PruefergebnisDto pruefergebnisDto = createPruefergebnisDtoBuilder(pruefauftragId, REGEL_KONFIGURATION_ID_AKTIV).build();

    final Response response = inTransaction(() -> pruefauftragResource.speicherePruefergebnis(pruefergebnisDto, pruefauftragId.getId(), request));

    verifySpeicherePruefergebnisResponse(pruefauftragId, response);

    prozessWeiterfuehren(processInstance, 1);

    verifyPruefresultatVorhanden(esDelegate.readPruefresultat(readIn));
}
```

After

```
@Test
void send_rechung_and_receive_pruefergebnis(){
    SHARK_SEND_RECHNUNG();
    final Pruefauftrag pruefauftrag = KOORDINATOR_CREATE_RRUEFAUFTRAG();
    KOORDINATOR_SEND_RRUEFAUFTRAG(pruefauftrag);

    final Response pruefergebnisDto = RESPONSE_PRUEFERGEBNIS();

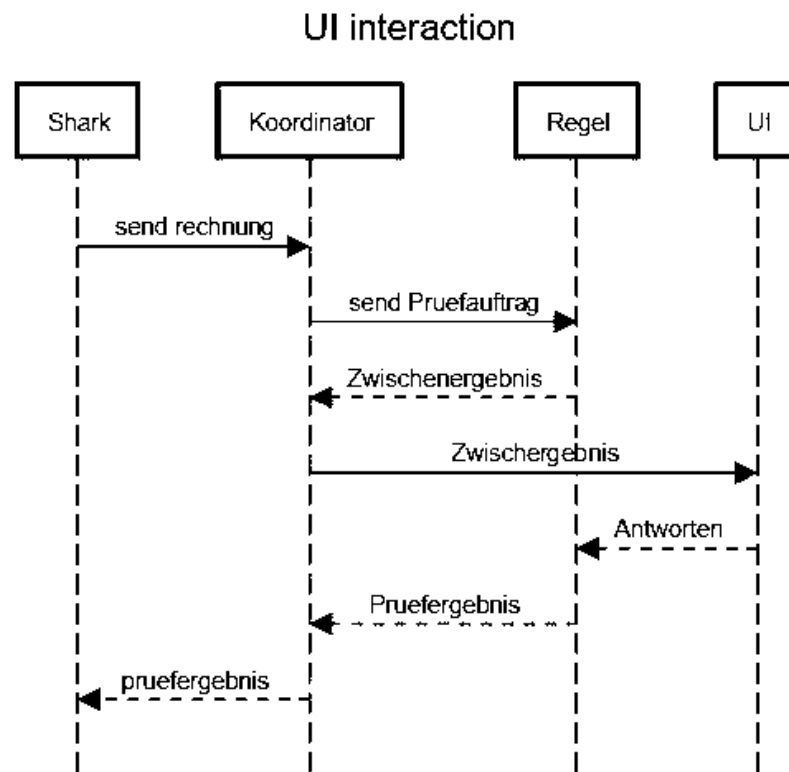
    THE_PRUEFERGENIS_SHOULD_BE(pruefergebnisDto);
}
```

```
@Test
void send_rechung_and_receive_pruefergebnis() {
    SHARK_SEND_RECHNUNG();

    final Response pruefergebnisDto = RESPONSE_PRUEFERGEBNIS();

    THE_PRUEFERGENIS_SHOULD_BE(pruefergebnisDto);
}
```

Sequencediagram UI interaction



New test case

```
@Test
void send_rechung_user_interaction_receive_pruefergebnis() {
    SHARK_SEND_RECHNUNG();
    final Pruefauftrag pruefauftrag = KOORDINATOR_CREATE_RRUEFAUFTRAG();
    KOORDINATOR_SEND_RRUEFAUFTRAG(pruefauftrag);

    RESPONSE_ZWISCHENERGEBNIS();
    USER_INTERACTS();
    final Response pruefergebnisDto = RESPONSE_PRUEFERGEBNIS();

    THE_PRUEFERGENIS_SHOULD_BE(pruefergebnisDto);
}
```

```
@Test
void send_rechung_user_interaction_receive_pruefergebnis() {
    SHARK_SEND_RECHNUNG();

    final Response pruefergebnisDto = RESPONSE_PRUEFERGEBNIS();

    THE_PRUEFERGENIS_SHOULD_BE(pruefergebnisDto);
}
```


Concluision

- Long discussions
- Difficult to determine abstraction layer
- Better readability
- We should have done it at the begining
- What's the difference to BDD?

References

- <https://www.browserstack.com/guide/tdd-vs-bdd-vs-atdd>
- <https://gaboequivel.com/blog/2014/differences-between-tdd-atdd-and-bdd/>