Spring Boot Microservices

Beginner to Guru

Retrospective

Agile Software Development

- Agile Software Development is an iterative development process
  - Software is developed in small increments
  - Widely adopted by companies
    - Often poorly!
- Agile is a VERY large topic!
  - Only touching on the surface in this lecture
  - MUCH more complex than this discussion!

SCRUM

- SCRUM is a process framework
  - Tasks are planned into a backlog
  - Tasks are planned into a “Sprint”
- A Sprint is typically a 2 - 4 week period of work for 5 to 9 people
  - Length and team size are highly debated
- At the end of a Sprint, a Retrospective is held
  - Purpose is to reflect and improve for next iteration

A Typical Sprint Retrospective Model

What worked well?

What could be improved?

What will we commit to doing in the next Sprint?

Scrum Team members make actionable commitments

What Worked Well?

- Planning Issues in GitHub
  - Allows better visibility of what source code was changed
- Having CI Builds to catch problems
  - 3 failures detected

What Could be Improved?

- Ideally should not have had 3 build failures
  - Testing is VERY light
- Section of course was too long
  - Several videos too long
- Need better examples of Compensating Transactions

What Will We Commit to Doing in Next Sprint?

- Improve Test Coverage
  - Testing has not been a focus of the course - *(Whole 18+ hr course on testing!)*
  - Saga’s clearly have a lot of moving parts!
    - Verify the Saga with integration tests
  - **ADD** Section to course for Integration Testing
- Improve examples of compensating transactions
  - **ADD** section to course for compensating transactions

Integration Tests

- Unit Tests typically target a single class
- Integration tests will test the interactions between components
- In our service, we want to test:
  - Receiving of a new order
  - Sending / Receiving JMS messages
  - Persistance
  - State Changes

Compensating Transactions

- Problem - Current 'non-happy' path events end in terminal state
- Need example of compensating transactions
  - Non-happy path - call service to 'un-do' action

SPRING FRAMEWORK

GURU