Utilizing Gherkin For AI Testing

Ken Pugh

March 2021

1

Why Gherkin for AI?

- Streamlines prompting
- Decreases rework
- Readable
- Acts as an executable specification

2

2

Overall Rule

There are exceptions to every statement, except this one

3

Another Overall Rule

Context is Everything

4

Still Another Overall Rule

Explicitness beats implicitness

5

5

Specific Rule

No code goes in till the spec/test goes on

6

Equivalents for Purposes of this Session

- Specification Driven Development
- Behavior Driven Development
- Acceptance Test Driven Development

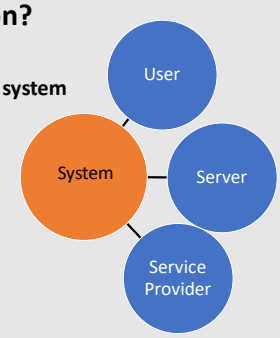
- Specifying Behavior and Testing Behavior exists

7

7

What Is in a Specification?

- Behavior from external view of system
  - Inputs and outputs
  - State changes
  - External interfaces



8

Definitions

- Acceptance criteria (general)
  - Outlines correct behavior
- Acceptance tests (specific scenarios)
  - Specific tests of behavior that either pass or fail
  - Implementation independent

9

Example of Defining Behavior

10

10

Fast Car Example

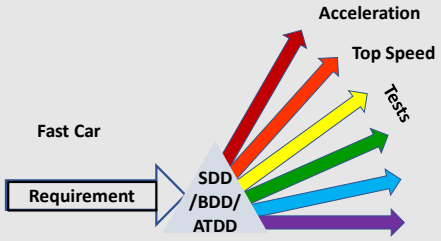
- Who wants a fast car?
- Criteria
  Must accelerate to desired speed within some time
- Test
  Accelerate 0 to 60 mph / 100 kph in X seconds

  Top speed?
  Time at top speed?

11

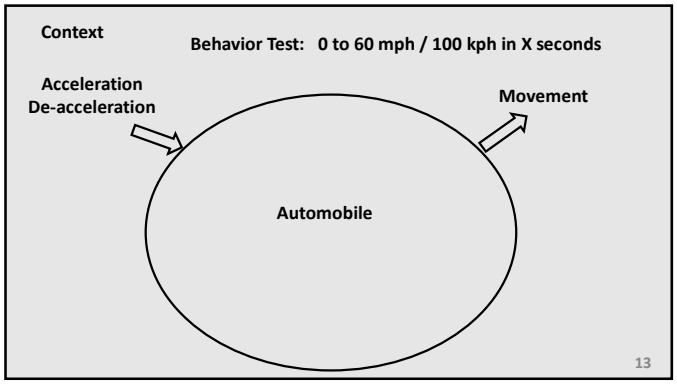
11

Behavior as a Prism

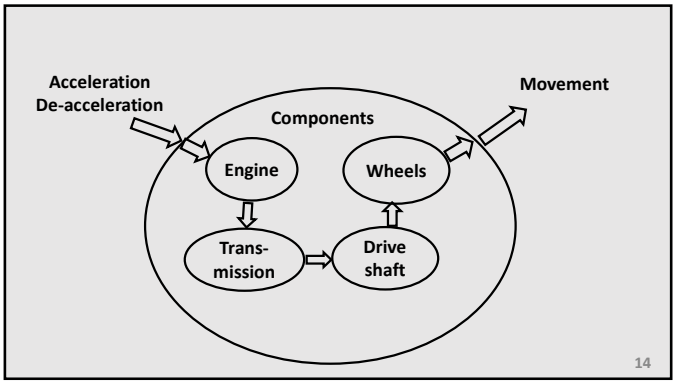


12

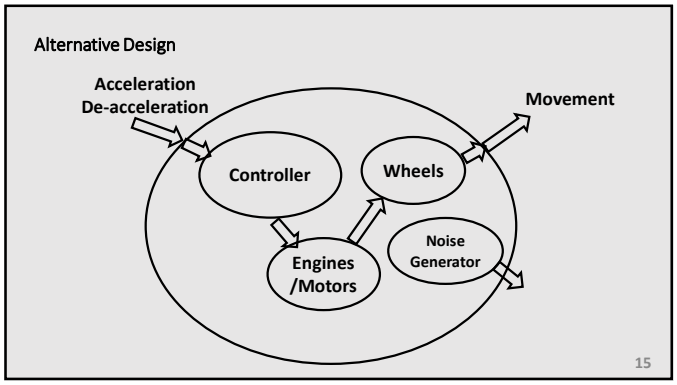
12



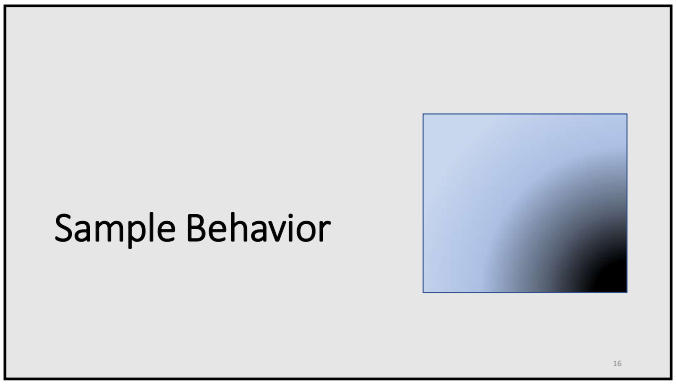
13



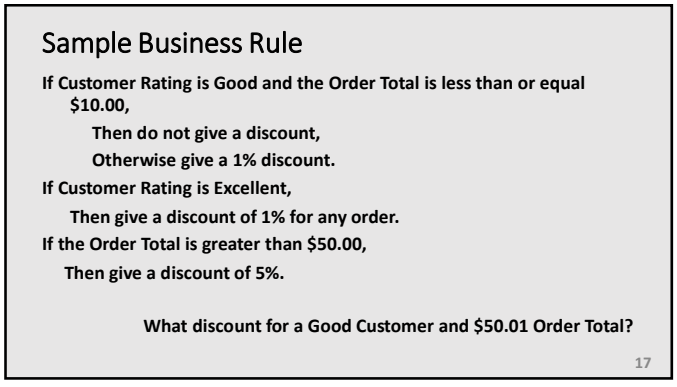
14



15



16



17

18

```
Given total is $50.01 and rating is Good
When discount computed
Then percent is 1%

Given total is $10.00 and rating is Good
When discount computed
Then percent is 0%

Given total is $10.01 and rating is Good
When discount computed
Then percent is 1%
```

19

19

Specifications and Tests

Specification

```
Given total is $10.01 and rating is Good
When discount computed
Then percent is 1%
```

Test

```
Given total is $10.01 and rating is Good
When discount computed
CHECK percent is 1%
```

20

20

Gherkin

```
Scenario: Compute discount
* Discount based on Order Total and Customer Rating
|Order Total|Customer Rating|Discount Percentage|
|50.01      |Good           |1                  |
|10.00      |Good           |0                  |
|10.01      |Good           |1                  |
|0.01       |Excellent      |1                  |
|50.00      |Excellent      |1                  |
|50.01      |Excellent      |5                  |
```

21

21

Flow and Context

22

22

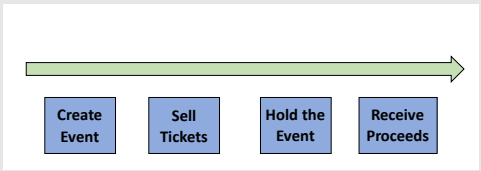
System Example

- Tickzon company presents on-line events for a charge:
  - As an event manager, I want to create events, sell tickets, and receive the proceeds.

23

23

Flow of Tickzon



24

24

Behavior - High Level

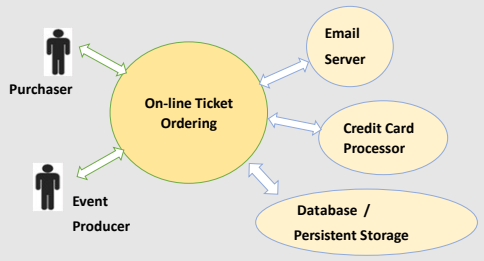
```
Scenario: Create Event
Given event does not exist
When producer creates event
Then event is available for ticket sales

Scenario: Sell Tickets
Given event is available for ticket sales
When a purchaser purchases a ticket
Then purchaser receives ticket in an email
And purchaser is charged for the ticket
```

25

25

Context Diagram



26

26

Behavior - Detailed

```
Scenario: Create event
Given event does not exist
When producer creates event
| Name            | Adele           |
| Date            | 6/1/2021        |
| Time            | 1:00 PM EDT     |
| Number Tickets  | 100             |
| Price           | $5.00           |
Then event is available for ticket sales
| Name  | Date      | Time        |Number Tickets | Price |
| Adele | 6/1/2021  | 1:00 PM EDT | 100           | $5.00 |
```

27

27

Another Behavior - Detailed

```
Scenario: Sell Ticket
Given event is available for ticket sales
| Name  | Date      | Time        |Number Tickets | Price |
| Adele | 6/1/2021  | 1:00 PM EDT | 100           | $5.00 |
When a purchaser buys a ticket
| Event               | Adele                 |
| Number Tickets      | 1                     |
| Email               | sam@thisisonlyatest.com |
| Credit Card Number  | 4005550000000019      |
```

28

28

Another Behavior - Detailed (continued)

```
Then purchaser receives ticket in an email containing
| Name   Date      | Time         | Number Tickets  |
| Adele| 6/1/2021 | 1:00 PM EDT  | 1               |
And purchaser is charged for the ticket
| Credit Card Number  | Item | Charge  |
| 4005550000000019    | Adele| $5.00   |
And number of tickets decreases
| Name | Number Tickets  |
| Adele| 99              |
```

29

29

More Details

30

30

Domain Terms

```
Scenario: Domain Term TicketCount
* TicketCount represents how many tickets are
available or being sold
| Value  | Valid  | Notes               |
| -1     | No     |                     |
| 0      | Yes    | Must be at least 0  |
| 1      | Yes    |                     |
| 10000  | Yes    | Maximum             |
| 10001  | No     | Over maximum        |
```

31

31

Domain Terms

```
Scenario: Domain Term Percentage
* Percentages are between 0 and 100
| Value  | Valid  | Notes       |
| -1     | No     | Below zero  |
| 0      | Yes    |             |
| 100    | Yes    |             |
| 101    | No     | Over 100    |
```

32

32

Business Rules

```
Scenario: Business Rule Discount
* Discount is given for buying more than one ticket
| Number Tickets | Discount Percentage| Notes                  |
| 1              | 0%                 |                        |
| 2              | 5%                 | Discount for 2 to 5    |
| 5              | 5%                 |                        |
| 6              | 10%                | Discount for 6 or more |
| 100            | 10%                |                        |
```

33

33

Discover the Possibilities

```
Scenario: Create Event for Date in the Past
Scenario: Sell a Ticket after Event is over
Scenario: Purchaser's Purchase Information is
Invalid
Scenario: Purchaser Does Not Receive a Ticket
```

34

34

Executing the Tests

35

35

Alternatives

- AI
  - Gherkin files are used as the input.
  - It creates the glue code and calls to production code
  - Alternatively, it creates unit tests from the Ghekin
- AI is the specifier
  - Ask it to create additional Gherkin files

36

36

More Information

- Gherkin Executor
  - Adds a Data statement
  - Describes the datatypes for each table field
- Data statement gives more information to AI

37

37

Domain Term and Data Types

```
# From a scenario:
When a purchaser buys a ticket
| Event               | Adele                    |
| Number Tickets      | 1                        |
| Email               | sam@thisisonlyatest.com  |
| Credit Card Number  | 4005550000000019         |

Data
| Name                | Data Type          |
| Event               | String             |
| Number Tickets      | TicketCount        |
| Email               | EmailAddress       |
| Credit Card Number  | CreditCardNumber   |
```

38

38

Business Rules

```
# Rule Scenario: Business Rule Discount
* Discount is given for buying more than one ticket
| Number Tickets | Discount Percentage| Notes
| 1              | 0%                 |

Data
| Name                | Data Type    |
| Number Tickets      | TicketCount  |
| Discount Percentage | Percentage   |
```

39

39

Smaller Steps

40

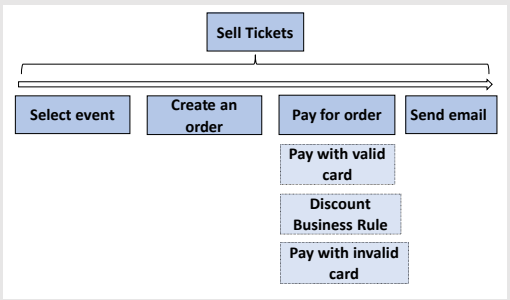
40

Act Locally, Think Globally

41

41

Decomposing Flows



42

42

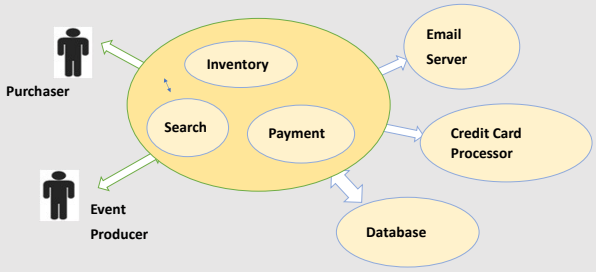
Design

- Decomposing behavior into smaller behaviors
  - May have multiple levels
- AI implements the smaller behaviors

43

43

Decomposing Context



44

44

In Conclusion

45

45

Goal of Writing Gherkin

Replace misunderstanding
with
Shared understanding

46

46

Examples

- https://github.com/atdd-bdd
  - https://github.com/atdd-bdd/TestRecorderFromClaude
    - Full stack (UI/logic/database) created by Claude
    - Create manual tests and record test run results
  - https://github.com/atdd-bdd/BowlingFromClaude
    - Bowling program
    - Other modules

47

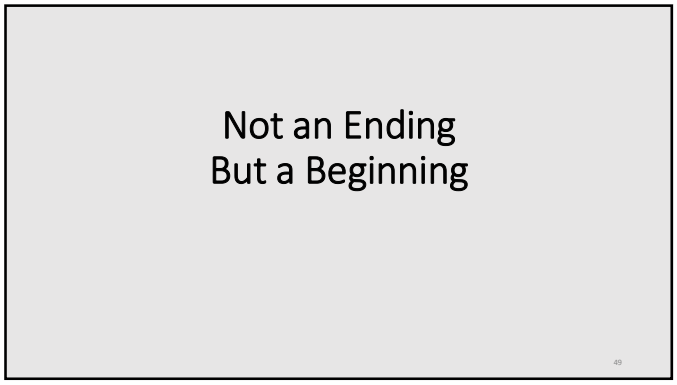
47

Immersive training in
Behavior Driven Development /
Acceptance Test-Driven Development /
Specification Driven Development

ken@kenpugh.com
https://www.linkedin.com/in/kenpugh/
https://kenpugh.com

48

48



49