

BDD, ATDD, TDD, DDD, DP, OOAD, ... If You're D'd out, Try A Holistic Approach

Ken Pugh
ken@kenpugh.com

Who are you?

- Developers, testers, others?
- Why are you here?

Objectives

- Explain the different contexts of an application
- Identify what are external and internal behaviors and how to specify them
- Create a shared understanding of the behavior of business rules and domain terms
- Minimize the amount of redundancy in describing behavior

Path

- External Behavior
- Internal Behavior
- Domain Term and Business Rule Behavior
- Automation of Behavior Tests
- Quality Attributes for Behavior

Introduction

Ken Pugh



- ATDD/BDD, TDD, BVDD, Lean, Scrum, SAFe, Design Patterns
- Over 2/5 century of software development experience
- Co-author SAFe® Agile Software Engineering
- Author of seven books

ken@kenpugh.com

<https://www.linkedin.com/in/kenpugh/>

<http://acceptancetestdrivendevelopment.com>



Lean Agile Acceptance Test-Driven Development: Better Software Through Collaboration

Prefactoring: Extreme Abstraction, Extreme Separation, Extreme Readability

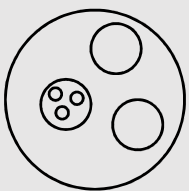
Interface Oriented Design

Overall Rule

There are exceptions to every statement, except this one

2nd Overall Rule

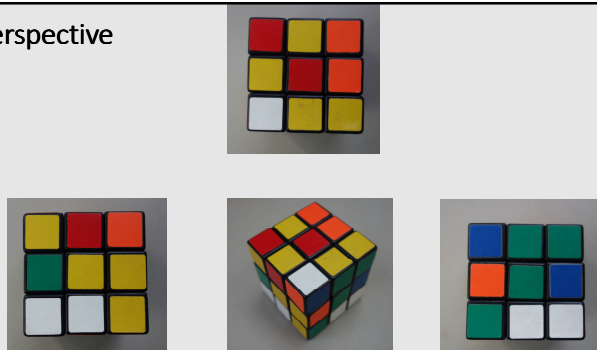
Context is everything
 Everything exists in a context
 Everything is always true in some context



Always Tradeoffs

- Commonality, e.g. libraries versus
- Dependence on those libraries

Perspective

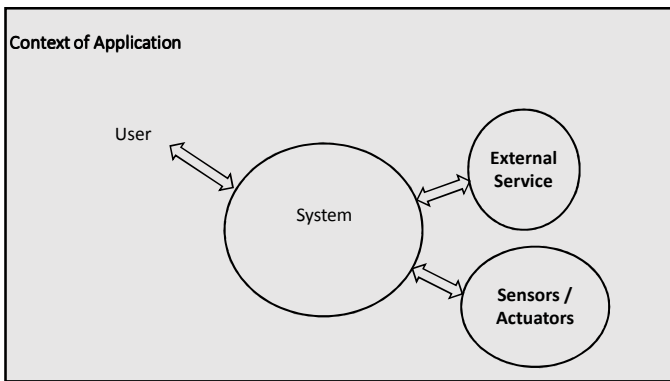
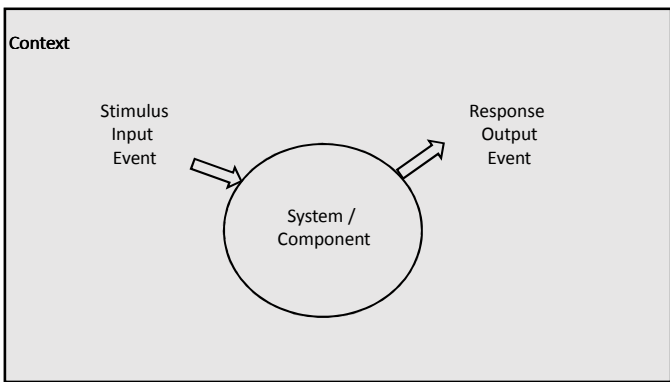
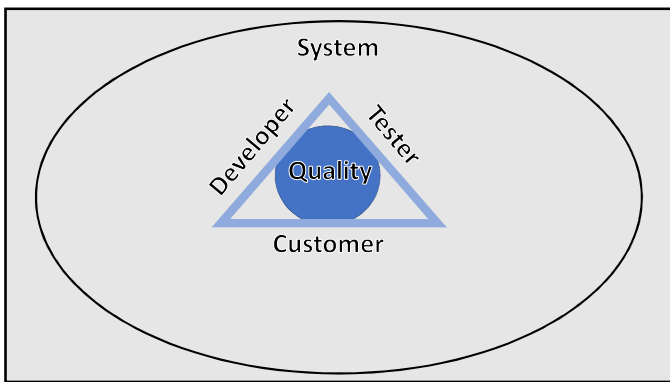


Definitions

- **Component**
 Service, Module, Class
- **Responsibility**
 Fulfilled by cohesive set of behaviors

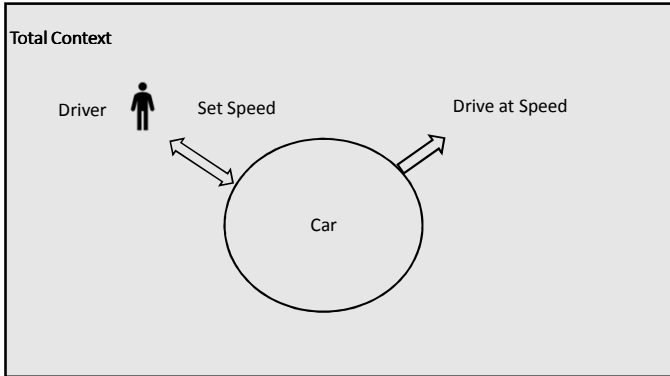
Context

- Context of Behavior
- What is inside and outside the system
 - Triad (customer, developer, tester) defines external behavior
 - Decouple behavior specification and implementation

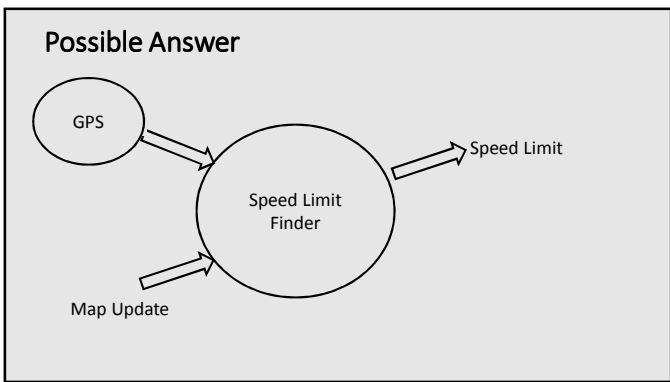
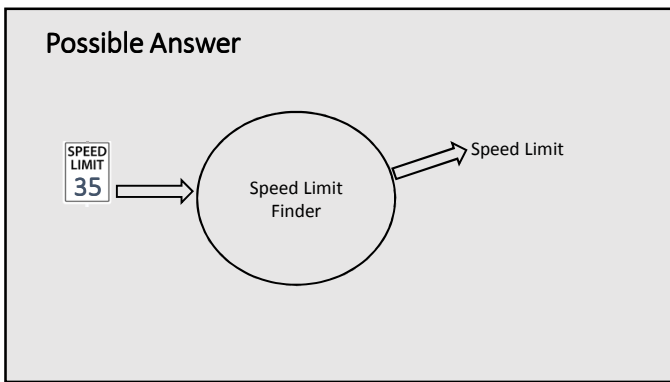


A Story – Speed Control

- As a driver, I want to set the speed for my car, and have it travel at that speed

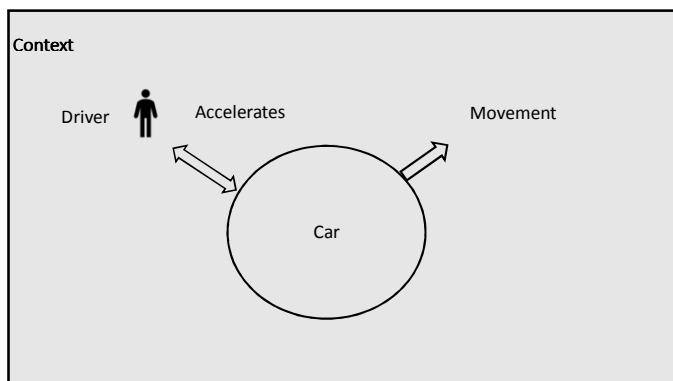


- Exercise**
- Create a simple context diagram for a story in your backlog
 - OR
 - You have been chosen to develop a speed limit finder which uses speed signs to determine the speed limit
 - Form a triad and create a simple context diagram
 - OR
 - You have been chosen to develop a speed limit finder which uses GPS location and map data to determine the speed limit



External Behavior

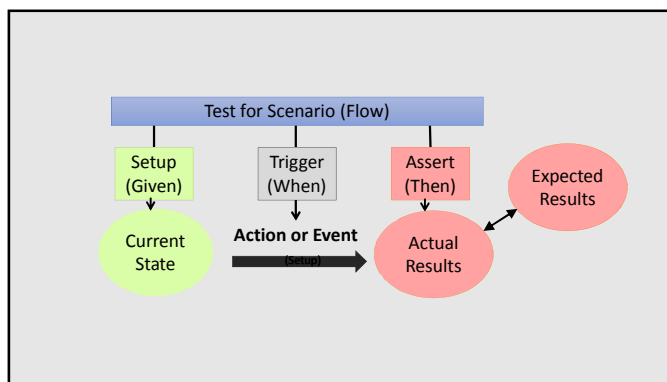
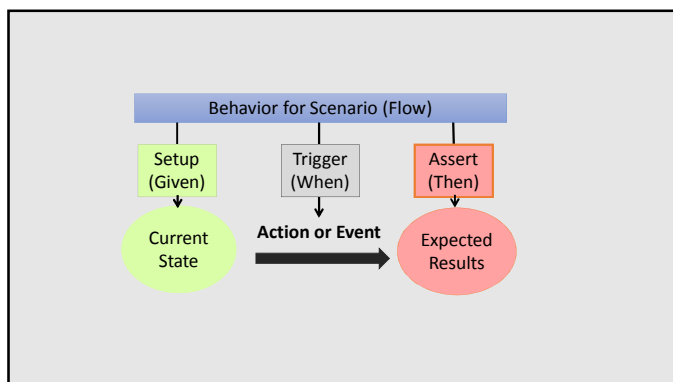
- Who wants a fast car?
- What behaviors do you desire for this car?



Behavior
 How a system/component reacts to a stimulus (input, event, etc.)

Requirement → Required behavior of a system/component

External behavior can be described by
 Use cases, requirements, acceptance tests, user acceptance test, scenarios etc.



Scenario Template

Scenario describes one behavior

Given some state
When event or action occurs
Then new state or output

Behavior / Tests for Speed Control

Setting speed causes speed to be close to set speed
 Given car is active
 When speed is set to 30 mph
 Then car travels between 29 mph and 30 mph

BDD focuses on desired behavior which then leads to tests

TEST setting speed causes speed to be close to set speed
 Given car is active
 When speed is set to 30 mph
 Then CHECK car travels between 29 mph and 30 mph

ATDD focuses on tests of the desired behavior

Behavior / Tests

Setting speed causes change to speed to occur rapidly
 Given car is traveling at 20 mph
 When speed is set to 30 mph
 Then car reaches 29 mph within 3 seconds

TEST Setting speed causes change to speed to occur rapidly
 Given car is traveling at 20 mph
 When speed is set to 30 mph
 Then CHECK car reaches 29 mph within 3 seconds

Alternatives for Describing Behavior

Initial State	Given	Setup	Arrange	Assemble	Pre-conditions
Action	When	Trigger	Act	Activate	Main course / exceptions / alternatives
Final State and/or Output	Then	Verify	Assert	Assert	Post-conditions

Requirements and Tests

Failing test is a behavior not yet met
 Passing test is specification of the system behavior

Requirements and tests are inter-related

- You can't have one without the other

Think Test First

No code goes in until the test goes on

If a requirement changes, a behavior changes, so a test should change

Think Test-First

Don't test code, code to the test


Exercise

- Create a scenario for the story from your backlog
- OR
- Create a scenario for the speed limit from signs
- OR
- Create a scenario for the speed limit from GPS

Possible Answers

Given car is active
 When speed limit sign is viewed
 Then speed limit is displayed

Given speed limit sign



When car views it
 Then 35 mph is displayed

Possible Answers

Given speed limit for N 45.1, W 47.2 is 35 mph
 When car is at N 45.1, W 47.2
 Then 35 mph is displayed


Allocating Behavior

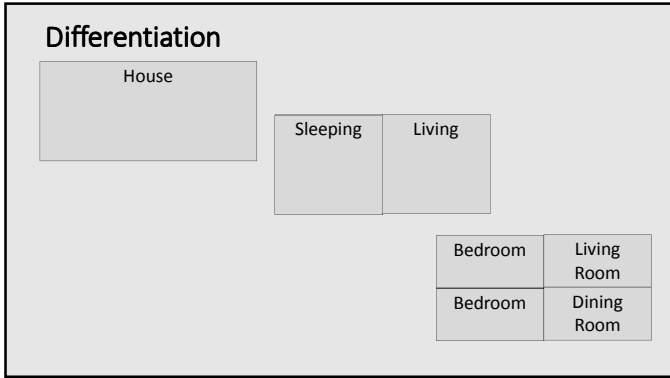
Synthesis versus Differentiation

“Within this process, every individual act of building is a process in which space gets differentiated. It is not a process of addition, in which preformed parts are combined to create a whole, but a process of unfolding, like the evolution of an embryo, in which the whole precedes the parts, and actually gives birth to them, by splitting”

Christopher W. Alexander

Synthesis

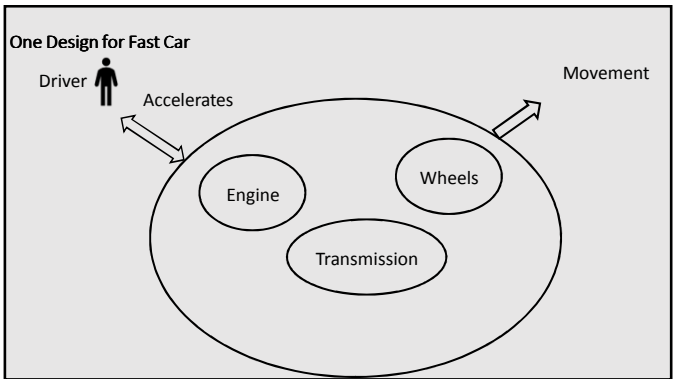
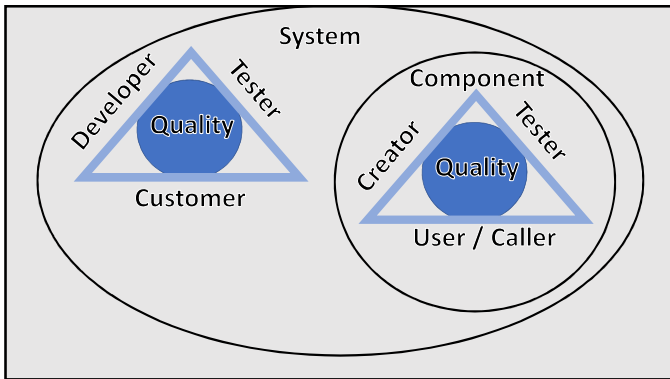


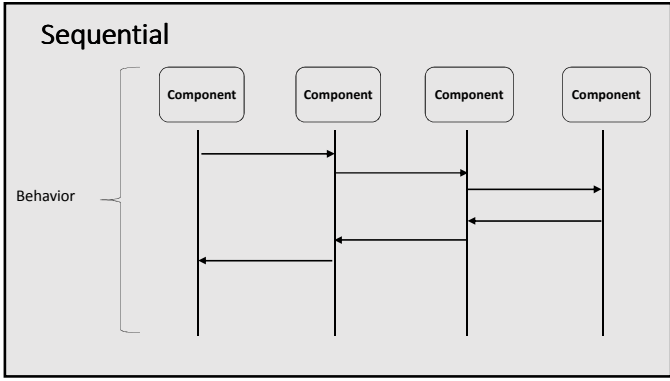
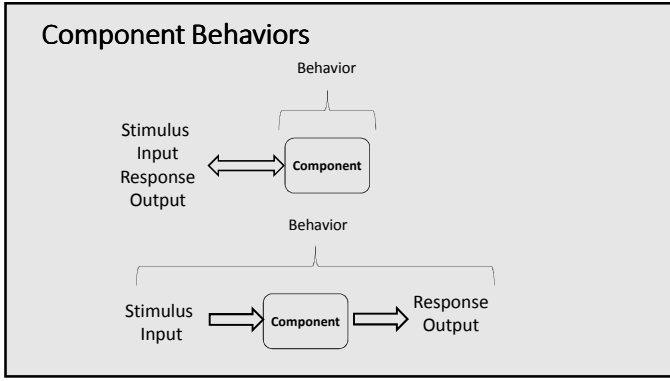
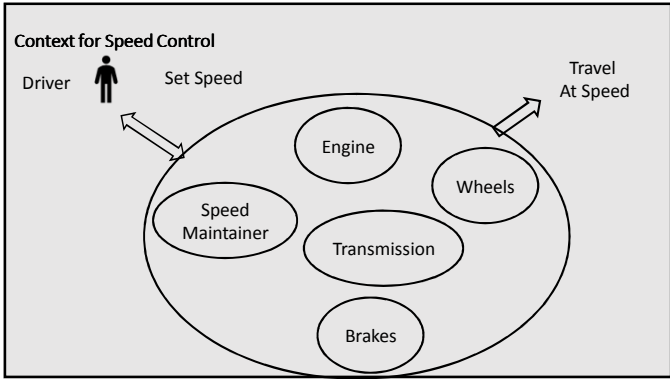
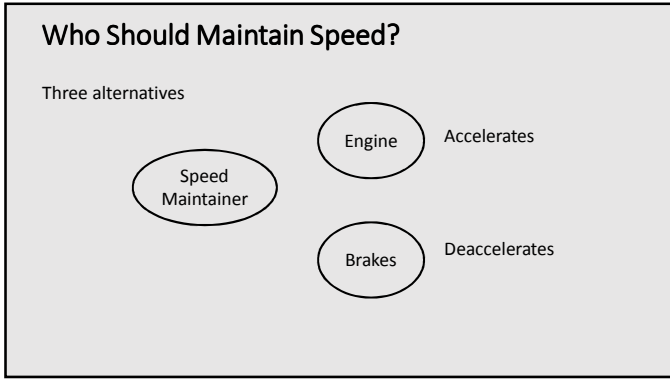
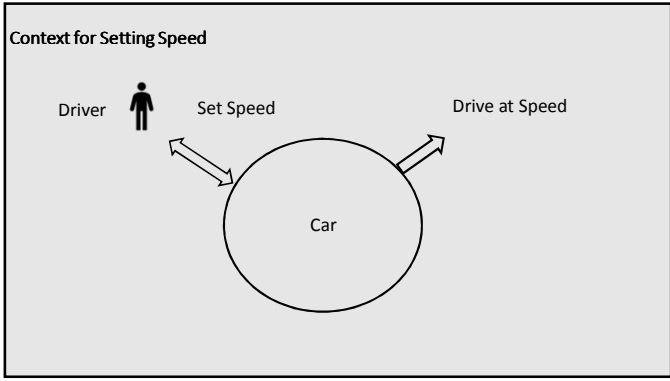
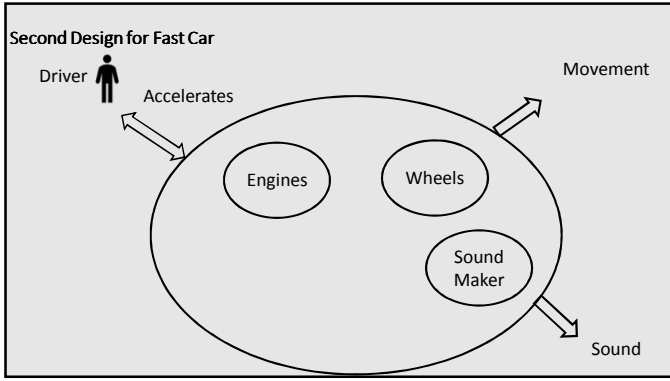


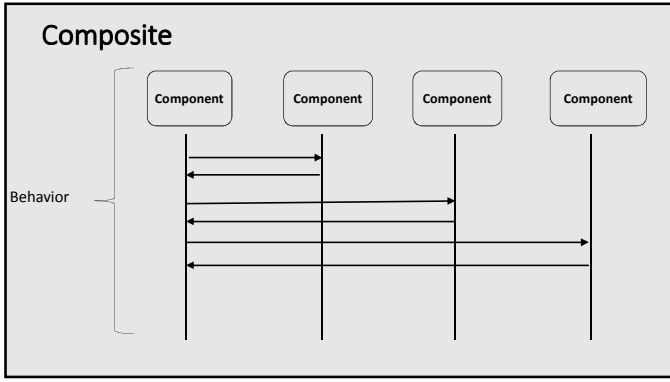
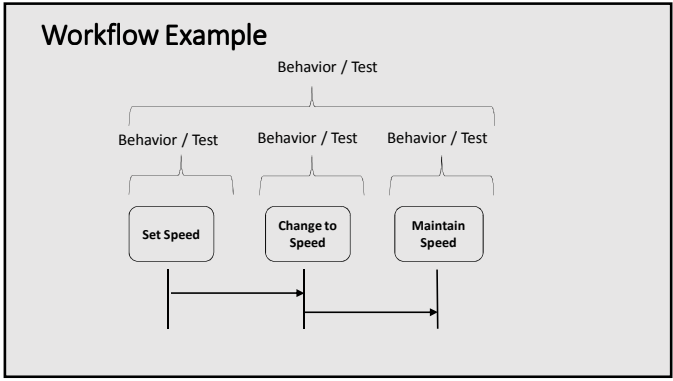
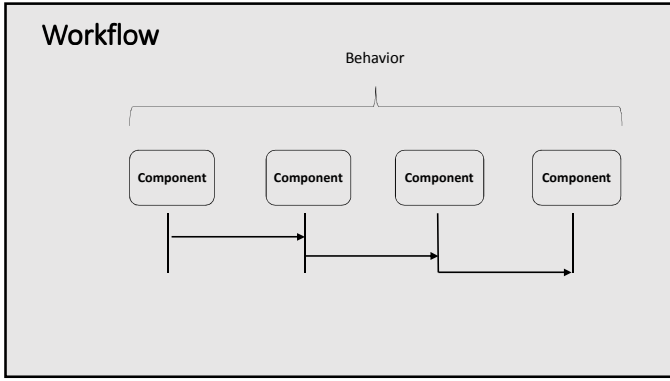
- Process is both
- Differentiate for cohesive sets of behaviors
- Synthesize from existing components with desired behaviors

- ### Power of Three
- Three ways to do something
So that tradeoffs become clearer
 - Three people creating behavior specifications (the triad)
 - Three people looking at code (pair plus one)

- ### Internal Behavior
- External behavior – defined by the Triad
 - Every internal component (e.g. function, class, microservice, etc.) helps produce external behavior
 - Design → Determine what internal components and their behavior required to meet external behavior
 - Look at three alternatives
 - Decide on the tradeoffs
 - Triad (user/caller, creator, tester) defines behavior of components
 - These are roles, not individuals

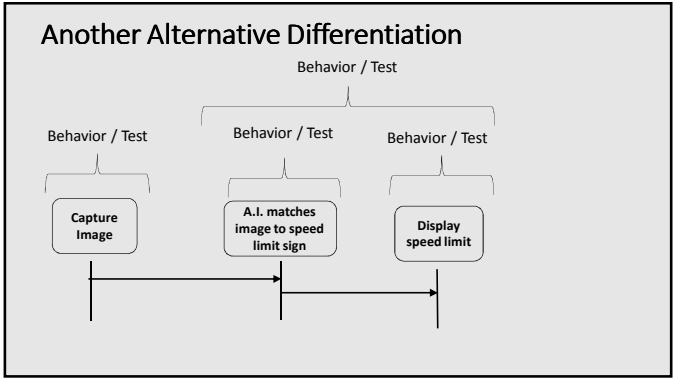
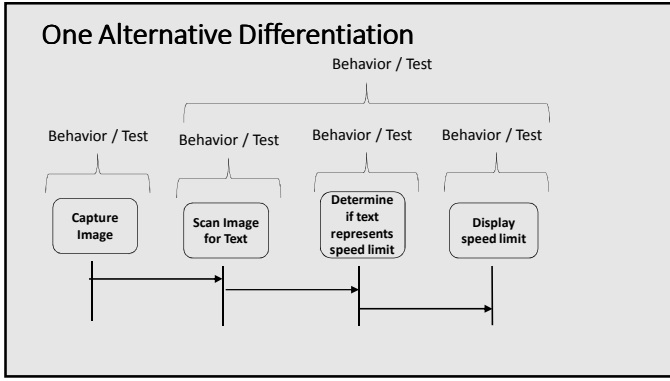


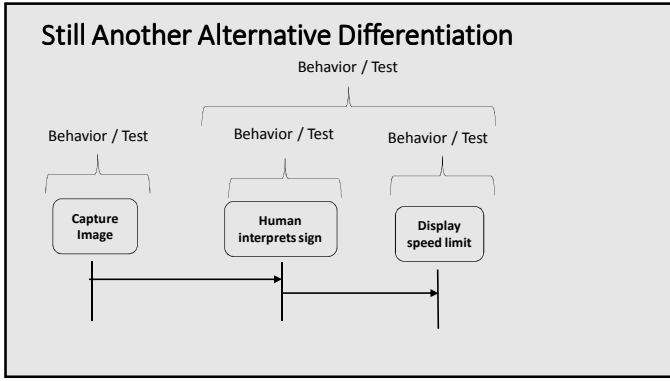




Exercise

- Behavior is:
 - Input Sign Image
 - Output Speed Limit
- What are some possible implementations?

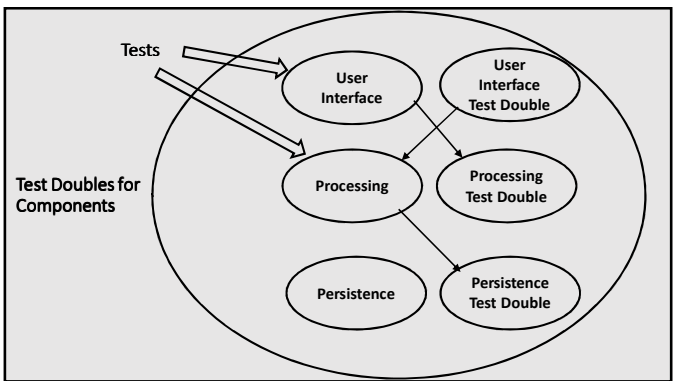
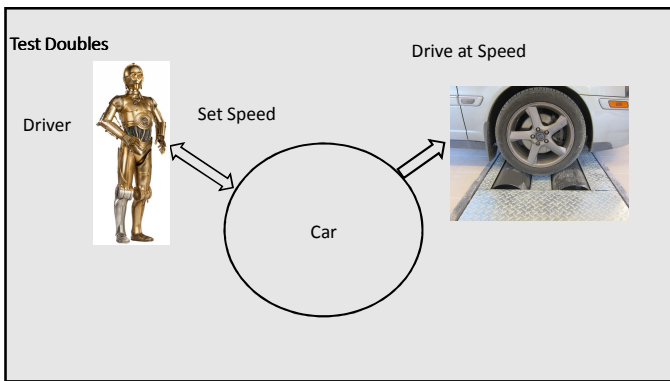




- Allocate behavior between components
 - All the way down (differentiate)
 - Until lowest level or existing component (synthesize)

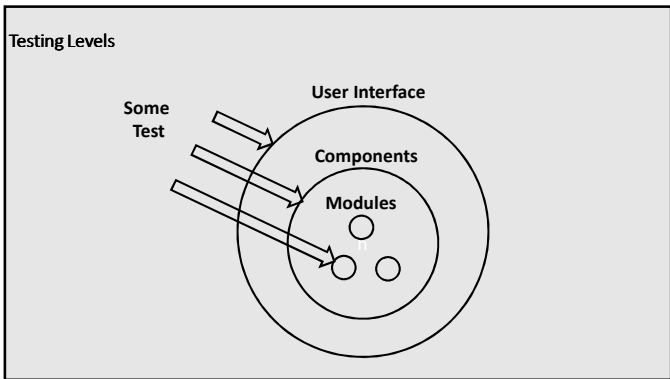
Automation of Behavior Tests

- ### Testing and behavior
- Testing →
 - Checking that desired behavior is provided by an implementation
 - Regression testing →
 - Double checking that behavior has not changed when other behaviors introduced
 - Behaviors with dependencies on slow/expensive/random components need test doubles / mocks



Automation

- Test automation is not something bolted on, but built in
- Use tests of behavior to design for automation
 - Identify appropriate test doubles
- Tests of behavior should be decoupled from implementation
 - Use "glue code" to connect test to production code
 - One test may be reused to connect to different levels e.g. UI, mid-tier, components



Exercise

- What test doubles might you need for your own story?
- Or
- What test doubles might be needed for the speed limit finder story?

Possible Answer

- External Behavior
 - Video of driving
- Internal Behavior
 - Set of sign images
 - Display

Understanding Domain Term and Business Rule Behavior

Domain terms

- Domain terms – used in defining the domain
 - Domain terms often become classes

Domain Term Behavior

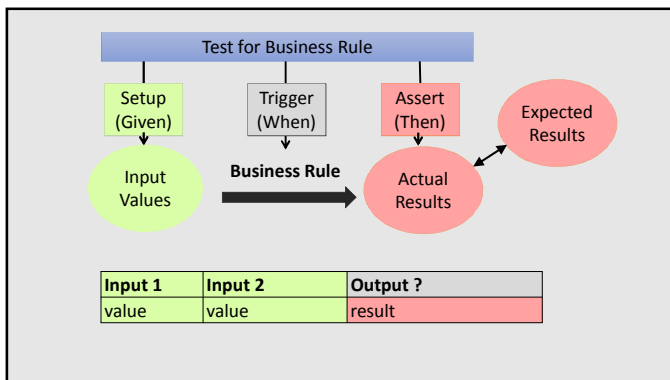
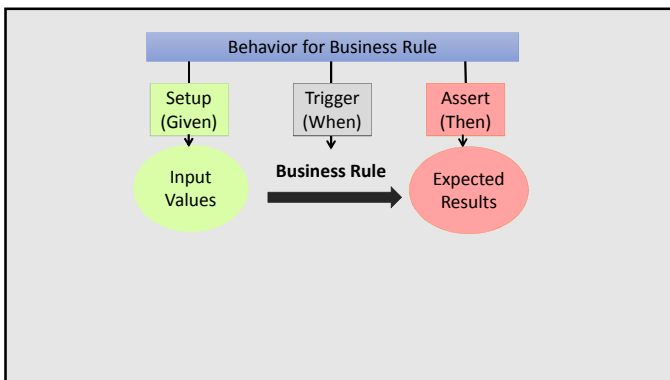
- Terms have behavior – e.g. what values they can have

Speed	Valid
-1	No
0	Yes
149	Yes
150	No

Direction	Valid	Notes
-1	No	
0	Yes	
359	Yes	
360	No	Wraps to 0

Business Rules

- Business rules
 - Usually specified by customer
 - Independent of implementation
- Business rules have behavior
 - The result they produce
 - Possible change in flow created by that result



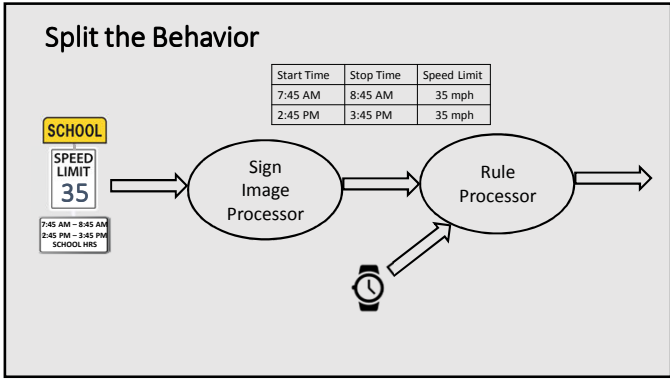
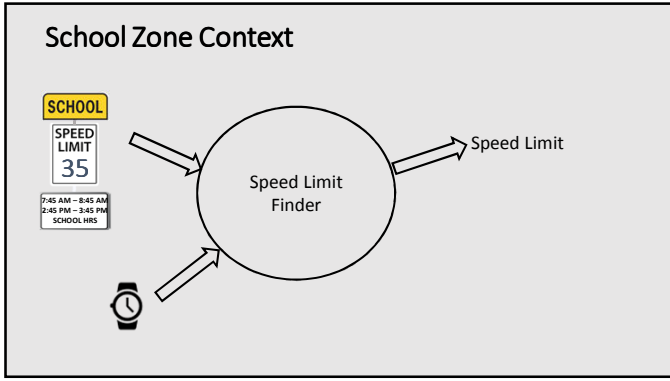
Change in Speed

- Multiple ways to change speed
 - Mileage sensitive
 - Time sensitive

Current speed	Desired speed	Distance	Goal	Deceleration rate?	Jerk Deceleration?	Start at distance
40 mph	30 mph	200 feet	mileage	5 feet/sec^2	2 feet/sec^3	0 feet
40 mph	30 mph	200 feet	time	10 feet/sec^2	5 feet/sec^3	100 feet

Exercise

- Write the business rule behavior test for a school zone speed limit sign
- Details follow



One Possible Answer

Time	Speed Limit	Notes
7:44 AM	45 mph	Speed limit for road
7:45 AM	35 mph	School zone applies
8:44 AM	35 mph	School zone applies
8:45 AM	45 mph	Speed limit for road

Define behavior for quality attributes

- ### Behavior is not just functional
- Behavior has quality attributes
 - AKA Cross-functional
 - AKA Non-functional
 - Quality attributes may include
 - Performance
 - Reliability
 - What else?

Change speed energy usage

Difference in speed	Gasoline Used?
10 mph	.01 gals
50 mph	.04 gals

Exercise

- What are the quality attributes for the behavior you have specified?

Signs identified	
Conditions	Percentage identified
Daylight clear	99.99%
Daylight rainy	99%
Dark clear	98%
Dark rainy	96%

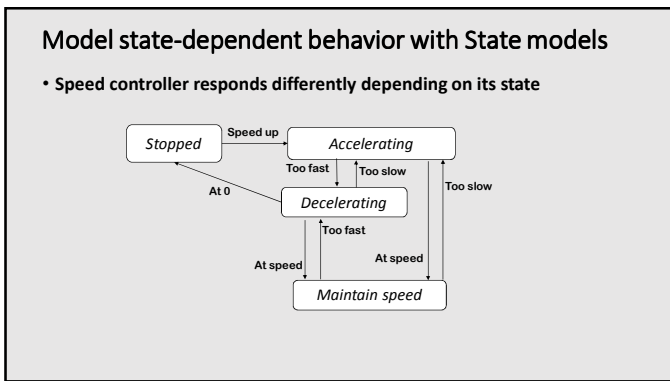
State Dependent Behavior

If there's time

State-dependent behavior

Components (e.g. Entities or objects) often:

- ▶ Have states and go through transitions between states
- ▶ May behave differently based on what state they are in
 - See what condition their condition is in
 - Then act/transition based on events



State transition table shows transitions between states

- Table can be used in place of the diagram
- Shows non-applicable state/transition pairs (N/A)

State/Event	Speed up	Too Slow	Too Fast	At Speed	At 0
Stopped	Accelerating	N/A	N/A	N/A	N/A
Accelerating	N/A	N/A	Decelerating	Maintain Speed	N/A
Decelerating	N/A	Accelerating	N/A	Maintain Speed	Stopped
Maintain Speed	N/A	Accelerating	Decelerating	N/A	N/A

Are we missing any transitions?

- Behavior for event may cause
 - Change to state
 - An operation or output

```
Given state is Maintain Speed
When event Too Fast occurs
Then state becomes Decelerating
    ask Brake to deaccelerate
```

▶ Every state/event transition should have a test for it

- For the correct state transition
- For the correct response

Exercise

- Is there a state-dependent behavior in your system?

It's Not the Ending, But a Beginning

Outcomes

- Explain the different contexts of an application
- Identify what are external and internal behaviors and how to specify them
- Create a shared understanding of the behavior of business rules and domain terms
- Minimize the amount of redundancy in describing behavior

Go Forth and Behave Yourself

Think Behavior All the Way Down

ken@kenpugh.com
<https://www.linkedin.com/in/kenpugh/>
<http://acceptancetestdrivendevelopment.com>