

From Hindsight to Foresight: Enhancing Design Artifacts for Business Logic Flaw Discovery

Carmen Cheh¹, Nicholas Tay², and Binbin Chen²

¹*Advanced Digital Sciences Center (ADSC), Singapore*

²*Singapore University of Technology and Design (SUTD), Singapore*



Motivation

Financial Services: Web Application Attacks Grow by 38% In First Half of 2021

 by Terry Ray on August 19, 2021

(Imperva, 2021)

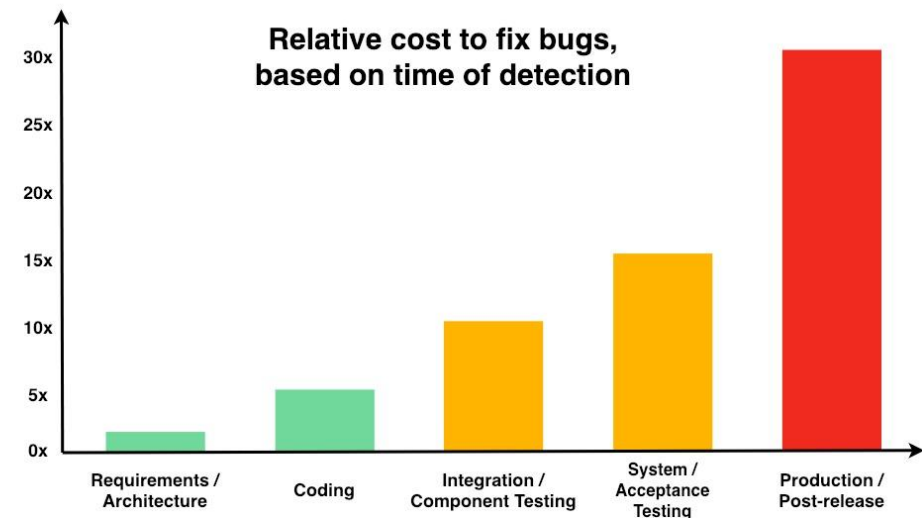
Empir Software Eng (2008) 13:289–302
DOI 10.1007/s10664-008-9062-z

Realizing quality improvement through test driven development: results and experiences of four industrial teams

**Nachiappan Nagappan • E. Michael Maximilien •
Thirumalesh Bhat • Laurie Williams**

(Microsoft)

High cost for fixing security issues in later stages
(source: NIST)



Background: Use Case Scenario Testing

Use case scenario

Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as "Saul Goodman"
And I specify the password as "Heisenberg"
And I confirm the password as "Heisenberg"
And I register this account
Then I should be notified that my account is created

Test results

Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as "Saul Goodman"
And I specify the password as "Heisenberg"
And I confirm the password as "Heisenberg"
And I register this account
Then I should be notified that my account is created

1 scenario (1 passed)
7 steps (7 passed)

Test code

```
/* @When I want to register a new account */
public function iWantToRegisterNewAccount(): void
{
    $this->fillContent();
}

/* @When I specify the password as :password */
public function iSpecifyThePasswordAs(string $password = ''):
void
{
    $this->content['password'] = $password;
}

/* @When I register this account */
public function iRegisterThisAccount(): void
{
    $this->client->request(,
        'POST', '/api/v2/shop/customers', [], [],
        ['HTTP_ACCEPT' => 'application/ld+json', 'CONTENT_TYPE'
=>
        'application/ld+json'], json_encode($this->content,
        \JSON_THROW_ON_ERROR));
    $this->content = [];
}
```

Our Vision: Create High-quality Misuse Case Scenarios

Use case scenario

Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as "Saul Goodman"
And I specify the password as "Heisenberg"
And I confirm the password as "Heisenberg"
And I register this account
Then I should be notified that my account is created



Misuse case scenarios



Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as "%&#^@&"
And I specify the password as "Heisenberg"
And I confirm the password as "Heisenberg"
And I register this account
Then I should be notified that my account is created



Test results

Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as "%&#^@&"
And I specify the password as "Heisenberg"
And I confirm the password as "Heisenberg"
And I register this account
Then I should be notified that my account is created

1 scenario (1 failed)
7 steps (the third step failed)

Design Constraints Can Help

Use case scenario

Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as "Saul Goodman"
And I specify the password as "Heisenberg"
And I confirm the password as "Heisenberg"
And I register this account
Then I should be notified that my account is created



Design constraints

1. Only new users can register an account
2. Name must not have any special character
3. Password must follow company's password policy
4. Confirmed password must be the same as the specified password
5. All fields must be filled in
6. Name and password can be specified in any order



Misuse case scenarios



Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as "%&#^@&"
And I specify the password as "Heisenberg"
And I confirm the password as "Heisenberg"
And I register this account
Then I should be notified that my account is created



Test results

Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as "%&#^@&"
And I specify the password as "Heisenberg"
And I confirm the password as "Heisenberg"
And I register this account
Then I should be notified that my account is created

1 scenario (1 failed)
7 steps (the third step failed)

What Kind of Design Constraints?

Use case scenario

Scenario: Registering an account

- (1) **Given** I am a (a): new user
- (2) **When** I want to register a new account
- (3) **And** I specify the name as (b): "Saul Goodman"
- (4) **And** I specify the password as (c): "Heisenberg"
- (5) **And** I confirm the password as (d): "Heisenberg"
- (6) **And** I register this account
- (7) **Then** I should be notified that my account is created

Data Constraints

Access

Only new users can register an account

Range

Name must not have any special character

Correlation

Confirmed password must be the same as the specified password

Action Constraints

Prerequisite

All fields must be filled in

Ordering

Name and password can be specified in any order

Repetition

A user cannot register the same account more than once

From Design Constraints to Misuse Case Scenario

Constraints		From Constraints to MUCS	Example MUCS Snippets
Data Constraints	Access	Replace input parameter with invalid value.	Given I am an existing user
	Range		And I specify the name as “!#\$@#%”
	Correlation		And I specify the password as “Heisenberg” And I confirm the password as “ hackingthis ”
Action Constraints	Prerequisite	Remove each prerequisite step	And I specify the name as “Saul Goodman” And I specify the password as “Heisenberg” And I confirm the password as “Heisenberg” And I register this account
	Ordering	Swap ordering of steps in a way that does not satisfy the constraint	And I confirm the password as “Heisenberg” And I specify the password as “Heisenberg”
	Repetition	Duplicate the group of steps	When I want to register a new account And I specify the name as “Saul Goodman” And I specify the password as “Heisenberg” And I confirm the password as “Heisenberg” And I register this account And I want to register a new account And I specify the name as “Saul Goodman” And I specify the password as “Heisenberg” And I confirm the password as “Heisenberg” And I register this account

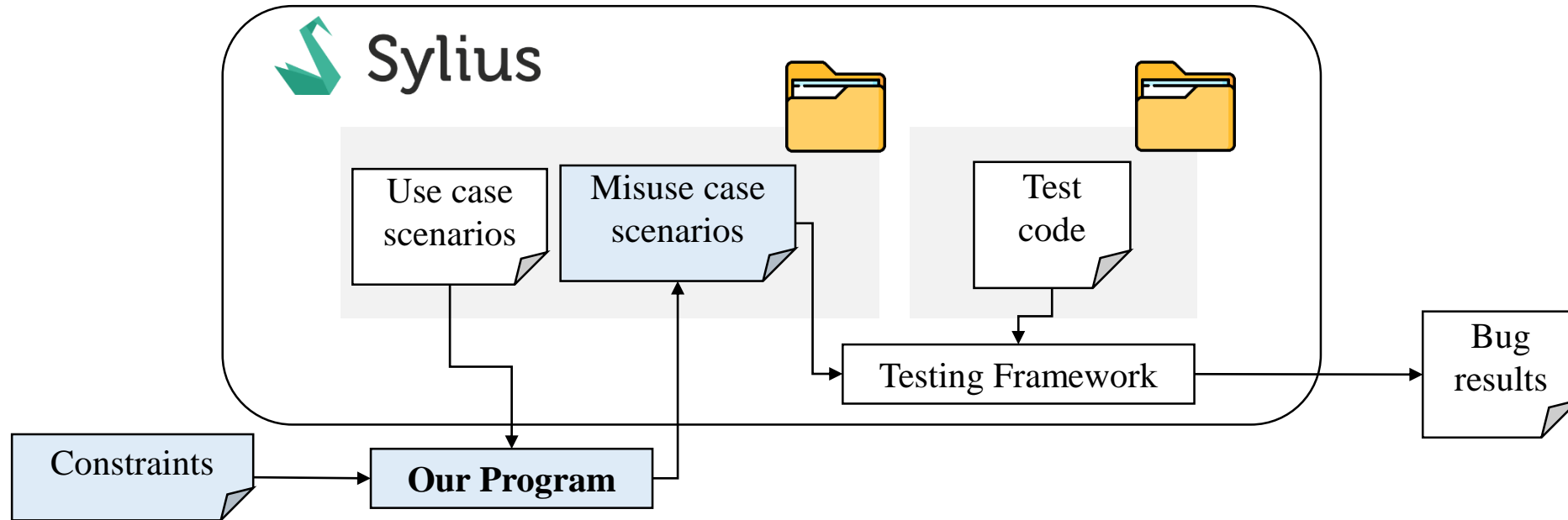
Our User Study: Objective & Approach

- Main research question: Can developers easily specify useful design constraints?
- Baseline: Developers will manually enumerate misuse case scenarios that they would like to test.
- Project used: Ecommerce open source project, Sylius, and file-sharing open source project, ownCloud

User Study Results: Constraints Specified

Constraints	Examples specified by participants
Access	Only administrators can change the tax rate, add a product, and create a new promotion
Range	<p>Shipping amount, product price, tax rate, and coupons' per customer usage limit should not be negative</p> <p>Passwords should follow the format set by the application's password policy</p> <p>HTTP method for accessing remote files must be POST or GET</p>
Correlation	<p>New password should be the same as re-confirmed password during resetting password.</p> <p>Only pre-defined variants can be used when adding product to cart</p>
Pre-requisite	<p>User must login first before doing any action</p> <p>User must add a product to cart to continue with checkout</p> <p>User must select shipping method to continue with checkout</p>
Ordering	<p>Checkout steps must be performed in strict order from addressing to shipping, and then to payment</p> <p>Administrator must log in first before performing administrative duties</p>
Repetition	<p>Following same reset password link twice should not allow user to change password</p> <p>Using a promotion multiple times should result in usage being decreased by equal amount</p>

Evaluation



- Ran misuse case scenarios on:
 - Old version of Sylius (2016)
 - Latest version of Sylius (as of the writing of the paper)
- Misuse case scenarios and results of running them have been released as artifacts¹

Turning Hindsight into Foresight

- Context: A security issue was reported by someone. After the Sylius developers fix the flaws, they add a misuse case scenario which is used to validate that the flaws are mitigated
- Findings: The generated misuse case scenarios are similar to those created by Sylius developers

Turning Hindsight into Foresight: Example

Use case scenario

@ui

Scenario: Adding a new shipping method

Given I am logged in as an administrator

When I want to create a new shipping method

And I specify its code as "Fedex"

And I specify its amount as 50

And I add it

Then I should be notified that it has been successfully created

Range Constraint:
Amount > 0

Sylus misuse case scenario

@ui

Scenario: Adding a new shipping method

Given I am logged in as an administrator

When I want to create a new shipping method

And I specify its code as "Fedex"

And I specify its amount as -50

And I try to add it

Then I should be notified that shipping cannot be below 0

And shipping method code "Fedex" should not be added

Expected Result:
FAIL

Our generated misuse case scenario

@ui

Scenario: Adding a new shipping method

Given I am logged in as an administrator

When I want to create a new shipping method

And I specify its code as "Fedex"

And I specify its amount as **-50**

And I add it

Then I should be notified that it has been successfully created

@ui

Scenario: Adding a new shipping method

Given I am logged in as an administrator

When I want to create a new shipping method

And I specify its code as "Fedex"

And I specify its amount as **-50**

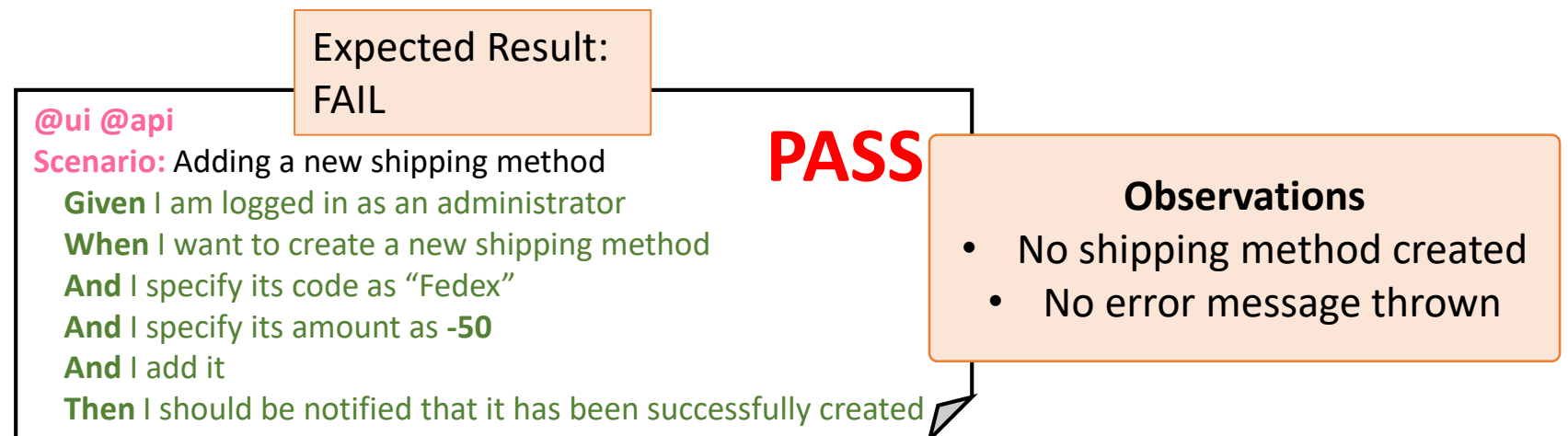
And I add it

Then I should be notified that it has been successfully created

PASS

Discovery of New Flaws

- Findings: 5 new flaws are discovered and brought to Sylius developers' attention. Flaws are currently (being) fixed.
- Our results show that we generate a larger coverage of potential misuse case scenarios for a given constraint



Discovery of New Flaws: Examples

Expected Result:
FAIL

@ui

PASS

Scenario: Modify tax rate

Given the store has "US Sales tax" tax rate of 20%
And I am logged in as an administrator
And I want to modify a tax rate "US Sales tax"
When I want to specify its amount as "-16%"
And I save my changes
Then this tax rate amount should be changed

Range Constraint:
Amount > 0%

Expected Result:
FAIL

@ui

FAIL

Scenario: Purchase product

Given the store has a "US Sales tax" tax rate of -10%
And the store has a product "T-Shirt" at \$100
And I am logged in as "John"
When I add product "T-Shirt" to the cart
Then my cart total should be \$110

Range Constraint:
Tax rate > 0%

The screenshot shows a web browser window with the URL 127.0.0.1:8000/en_US/cart/. The page displays a success message: "Success: Item has been added to cart". Below this is the "Your shopping cart" section, which includes a table of items and a summary table. The summary table is highlighted with a red border, and the "Taxes total" row is also highlighted with a red border.

Item	Unit price	Qty	Total
Grey Jeans grey_jeans	\$10.00	1	\$10.00

Summary	
Items total:	\$10.00
Estimated shipping cost:	\$3.09
Taxes total:	-\$0.70
Order total:	\$12.39

Discovery of New Flaws: Examples

Our generated misuse case scenario

@ui **PASS**
Scenario: Completing shipping step for checkout
Given the store has "Post" shipping method with "\$10" fee
And I have product "TShirt" in the cart
And I specify the shipping address as "908 LA"
When I select "~~Post~~" shipping
And I complete the shipping step
Then I should be on the payment step

Prerequisite Constraint:
User must select shipping method

Expected Result:
FAIL

Sylius misuse case scenario

@ui **FAIL**
Scenario: Completing shipping step for checkout
Given the store has "Post" shipping method with "\$10" fee
And I have product "TShirt" in the cart
And I specify the shipping address as "908 LA"
When I do not select a shipping method
And I try to complete the shipping step
Then I should still be on the shipping step

Before the flaw was fixed

After the flaw was fixed

@ui **PASS**
Scenario: Completing shipping step for checkout
Given the store has "Post" shipping method with "\$10" fee
And I have product "TShirt" in the cart
And I specify the shipping address as "908 LA"
When I select "~~Post~~" shipping
And I complete the shipping step
Then I should be on the payment step

@ui **PASS**
Scenario: Completing shipping step for checkout
Given the store has "Post" shipping method with "\$10" fee
And I have product "TShirt" in the cart
And I specify the shipping address as "908 LA"
When I do not select a shipping method
And I try to complete the shipping step
Then I should still be on the shipping step

Test code has not been
completed yet

Discovery of New Flaws: Examples

Use case scenario

@ui
Scenario: Counting promotion usage
Given there is a promotion "Limit" limited to 2 usage
And there is a customer "A" that bought a "Tshirt"
And the customer used "Limit" promotion
And the customer cancelled this order
When the administrator browses promotions
Then the "Limit" promotion should be used 0 times

Repetition Constraint:
Buying a product using
the promotion twice
should reduce the
promotion usage to 1

Expected Result:
FAIL

Our generated misuse case scenario

@ui
Scenario: Counting promotion usage **PASS**
Given there is a promotion "Limit" limited to 2 usage
And there is a customer "A" that bought a "Tshirt"
And the customer used "Limit" promotion
And there is a customer "A" that bought a "Tshirt"
And the customer used "Limit" promotion
And the customer cancelled this order
When the administrator browses promotions
Then the "Limit" promotion should be used 0 times

Promotion is not even applied to the order!
Missing promotion definition step:
And the promotion gives \$10 discount to every order.

Challenges

- Making it easier for developers to specify constraints
 - Intuitive syntax vs. expressiveness
- Generating misuse case scenarios for data constraints
 - Fuzzing parameters in a “smart” way
 - Maintaining certain syntax to match regex expression in code annotation
- Specifying detailed expected results when constraints are violated
 - Failure of misuse case scenario can be due to improper formulation of scenario/test code (which is not a design flaw)
 - Failure of misuse case scenario can happen at different steps in scenario

Related Work

State-of-the-Practice Tools



GAUNTLT
BE MEAN TO YOUR CODE AND LIKE IT

State-of-the-Art Research

- Model-based security testing and Design-by-Contract are two applicable concepts that can be introduced early on in design phase
- Types of design documents or models used for creating tests
 - Formal language models [1], [2]
 - Graphical models [3], [4]
 - Manual documentation [5]
 - Misuse case specification [6]

[1] M. R. Buchler, "Semi-automatic security testing of web applications with fault models and properties", PhD dissertation, Technical University of Munich, 2015.

[2] P. A. Pari Salas, P. Krishnan, and K. J. Ross, "Model-based security vulnerability testing," in *Australian Software Engineering Conference (ASWEC)*, 2007, pp. 284-296.

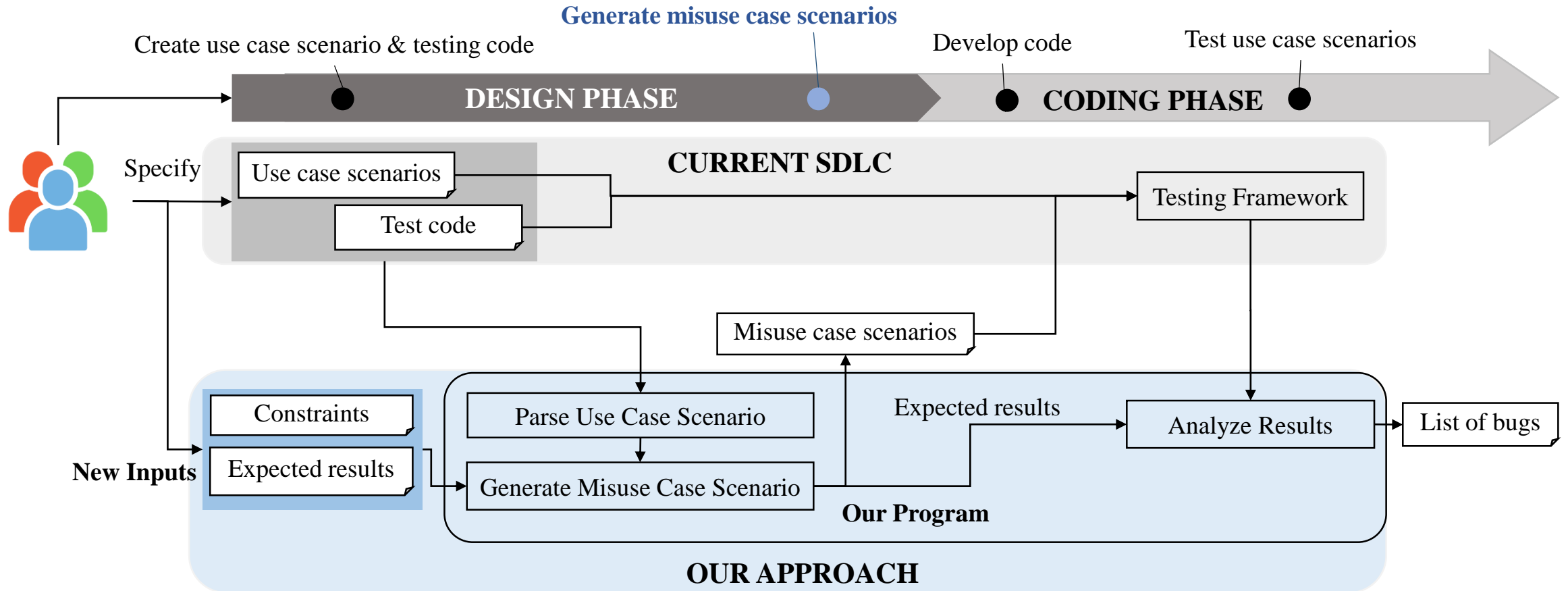
[3] A. Marback, H. Do, K. He, S. Kondamarri, and D. Xu, "Security test generation using threat trees," in *Proceedings of the 4th International Workshop on Automation of Software Test*, D. Dranidis, S. P. Masticola, and P. A. Strooper, Eds. IEEE Computer Society, 2009, pp. 62-69.

[4] K. He, Z. Feng, and X. Li, "An attack scenario based approach for software security testing at design stage," in *International Symposium on Computer Science and Computational Technology*, vol. 1, 2009, pp. 782-787.

[5] Y. Chen, L. Xing, Y. Qin, X. Liao, X. Wang, K. Chen, and W. Zou, "Devils in the guidance: Predicting logic vulnerabilities in payment syndication services through automated documentation analysis," in *28th USENIX Security Symposium*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 747-764.

[6] P. X. Mai, F. Pastore, A. Goknil, and L. C. Briand, "A natural language programming approach for requirements-based security testing," in *IEEE 29th International Symposium on Software Reliability Engineering (ISSRE)*, 2018, pp. 58-69.

Our Framework



Questions

*This research/project is supported in part by the National Research Foundation, Singapore and Infocomm Media Development Authority, and in part by the SUTD Start-up Research Grant
The authors would like to thank Ms. Alice Kham and Mr. En Low for their contributions to performing the experimental evaluation.*