

WE ARE HERE

BY: DEONATE BROWN
KARIMA DEWDNEY

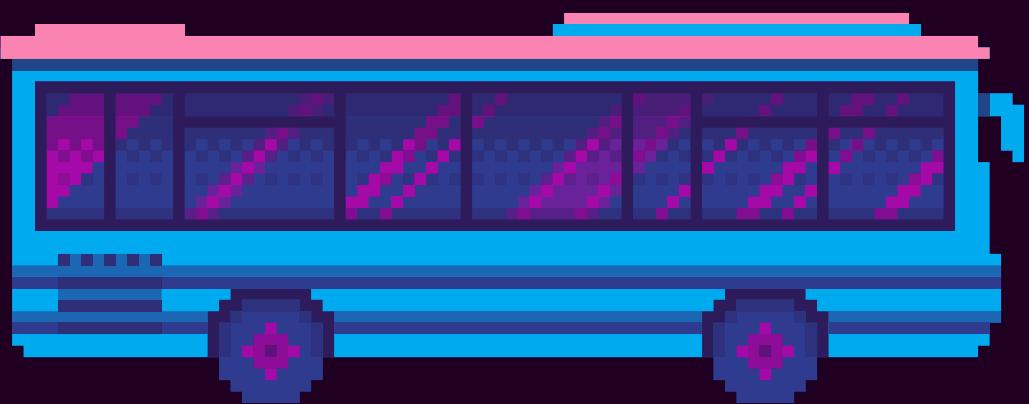
AGENDA



- 01 What is a Use Case?
- 02 Importance in Systems Analysis & Design
- 03 Written Use Cases and Their Structure
- 04 Creating Effective Use Cases
- 05 Rules, Best Practices, and Common Pitfalls
- 06 Examples & Case Studies
- 07 Summary & Next Steps

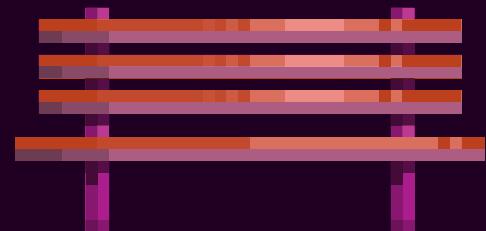
WHAT IS A USE CASE?

01 A use case describes how users (actors) interact with a system to achieve a goal.



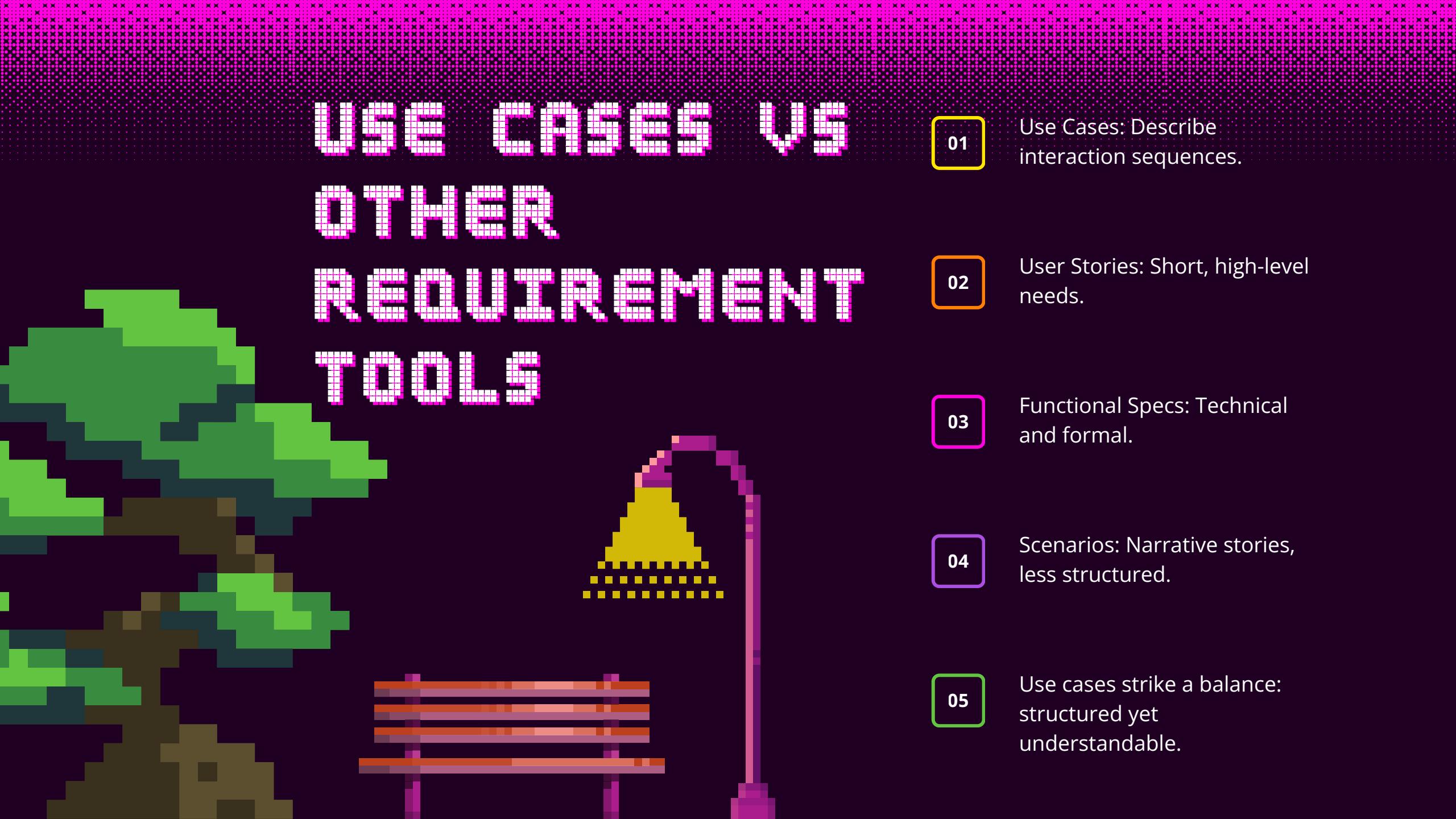
02 Captures functional requirements in a clear, structured format.

03 Key parts: Actor , System , Goal and Flow



WHY USE CASES MATTERS

- Bridge the gap between stakeholders and developers.
- Define system scope and boundaries.
- Help identify requirements early in analysis.
- Provide foundation for test cases and acceptance criteria.



USE CASES VS OTHER REQUIREMENT TOOLS

01

Use Cases: Describe interaction sequences.

02

User Stories: Short, high-level needs.

03

Functional Specs: Technical and formal.

04

Scenarios: Narrative stories, less structured.

05

Use cases strike a balance: structured yet understandable.

WRITTEN USE CASE DEFINITION

01

A written description
of actor-system
interaction.

02

Brief / Casual (short
summary).

03

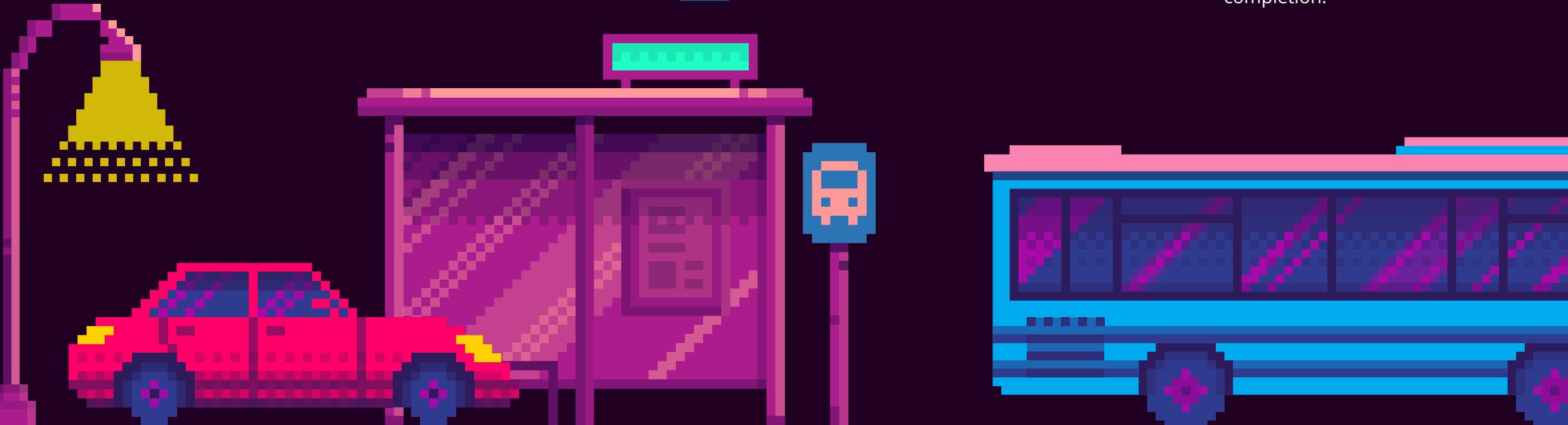
Detailed/ Fully Dressed
(structured with
sections)

04

Complements visual
use case diagrams.



STRUCTURE OF A WRITTEN USE CASE

- 
- 01 Title – identifies the goal.
 - 02 Brief / Casual (short summary).
 - 03 Trigger – event starting the use case.
 - 04 Preconditions – what must be true before starting.
 - 05 Normal Course – the main flow (happy path)
 - 06 Alternative Courses – variations of success
 - 07 Exceptions – failure conditions.
 - 08 Postconditions – state of system after completion.

LEVELS OF USE CASES RULES

01

Business use cases –
broad organizational
processes.

02

User-goal use cases –
typical user tasks (most
common level).

03

Subfunction use cases –
smaller support tasks.

04

Focus on user-goal level
for clarity and
usefulness.



USE CASE IN DEVELOPMENT LIFECYCLE

01 During requirements analysis clarifies needs.

02 During design : supports UML sequence diagrams.

03 During testing : basis for test case creation.

04 During maintenance : traceability for changes.

DIAGRAM VS WRITTEN NARRATIVE

01

Diagrams give an overview of actors and goals.

02

Narratives explain step-by-step flows.

03

Together, they provide clarity and completeness

RULES OF GOOD USE CASES

- 
- 01** Must involve a clear actor and goal.
 - 02** Should be goal-driven
 - 03** Written in simple, active voice
 - 04** Avoid design details like UI layout or internal algorithms
 - 05** Always describe what happens, not how it happens

BEST PRACTICES



01

One goal per use case for focus and clarity.

02

Number each step in the flow.

03

Separate normal flows, alternate flows, and exceptions.

04

Validate with both technical staff and end-users.

ALTERNATE FLOWS EXCEPTIONS

01

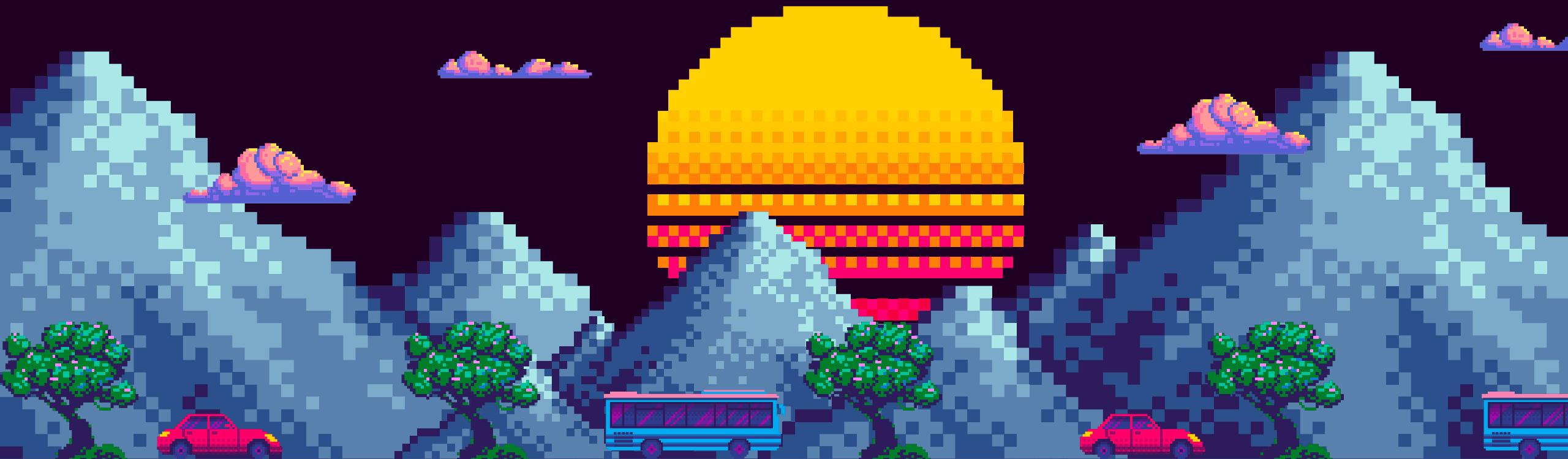
Normal Course: The happy path

02

Alternate Courses:
Variations that still achieve success.

03

Exceptions: Unexpected events or errors that prevent success.



PRECONDITIONS POSTCONDITIONS

01 Preconditions:
Conditions required
before execution

02 Postconditions:
Outcomes
guaranteed after
execution

03 Ensure use cases are
consistent and
testable

04 They help define the
scope of
responsibility of the
system



USE CASE RELATIONSHIPS

01

Include: Mandatory
shared behavior

02

Extend: Optional or
conditional steps

03

Generalization:
Specialized versions of
broader use cases.

04

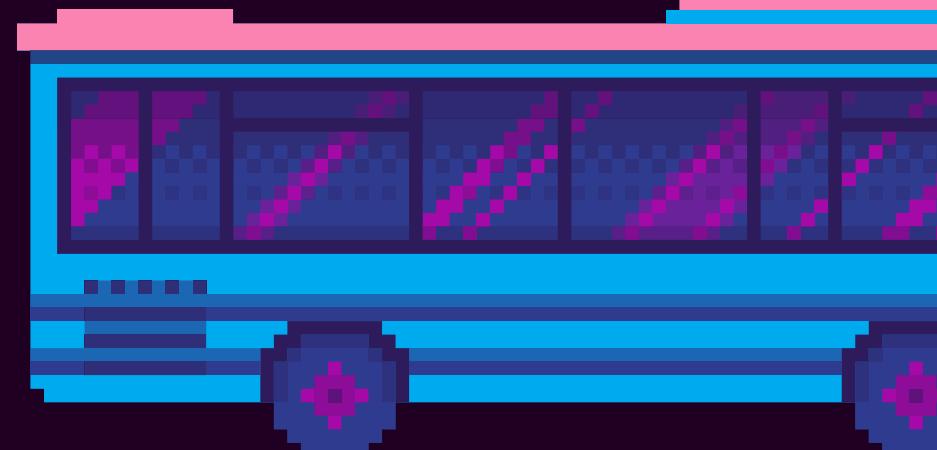
Relationships reduce
duplication and
improve clarity



PRIORITIZATION DEPENDENCY

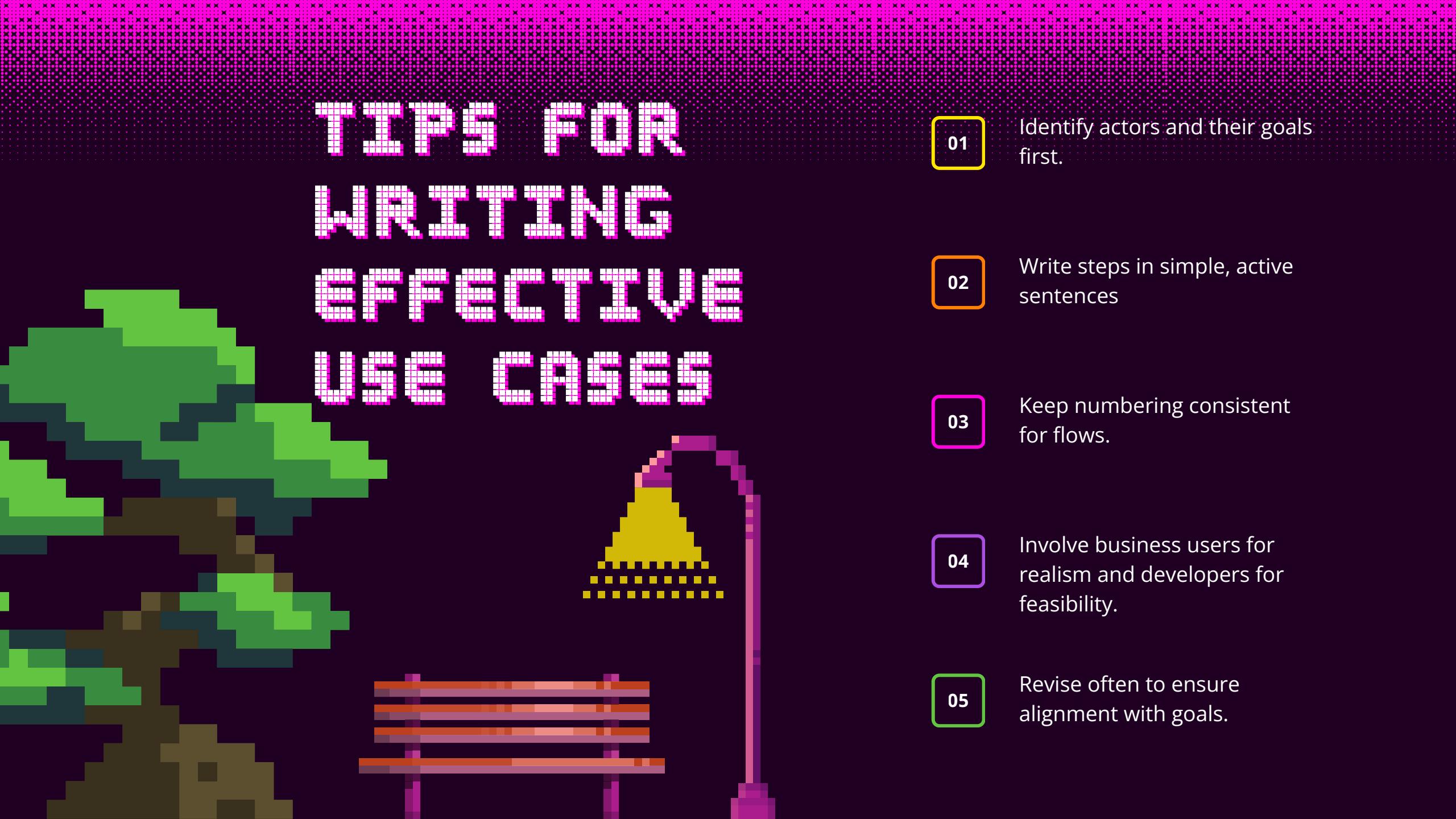


- 01 Some use cases are more critical than others.
- 02 Prioritize by business value, frequency, and risk.
- 03 Recognize dependencies (ex Login)
- 04 Prioritization guides development schedules and testing focus.



VALIDATING USE CASES

- Validate with stakeholders through walkthroughs
- Use checklists to check clarity, completeness, and consistency
- Ensure alternate flows and exceptions are realistic.
- Link use cases to requirements for traceability



TOPS FOR WRITING EFFECTIVE USE CASES

01

Identify actors and their goals first.

02

Write steps in simple, active sentences

03

Keep numbering consistent for flows.

04

Involve business users for realism and developers for feasibility.

05

Revise often to ensure alignment with goals.

EXAMPLE: HAPPY PATH

Use Case Name: Grocery Shopping

ID: UC-1

Priority: High

Actor: Shopper (Customer)

Description: This use case describes how a customer selects and purchases groceries from a store or online platform.

Trigger: The customer needs to restock household groceries.

Type: External

Preconditions:

1. The store or online grocery system is operational.
2. The customer has an account or shopping list.
3. Inventory data is up to date.

Normal Course (Happy Flow)

1. Customer enters the store or logs into the online grocery app.
2. Customer browses and selects grocery items.
3. System displays selected items in the cart with quantities and prices.
4. Customer reviews the cart and proceeds to checkout.
5. System prompts for payment and delivery/pickup preferences.
6. Customer provides payment details and confirms the order.
7. System processes the payment and generates a receipt.
8. Order confirmation or printed receipt is provided to the customer.

Postconditions:

- Payment is processed successfully.
- The customer receives confirmation, and items are scheduled for pickup or delivery.

Exceptions:

E1. Payment fails — system notifies the customer and prompts for an alternative payment method.

E2. Selected items are out of stock — system offers substitutes or removes them from the cart.

EXAMPLE: ALTERNATE FLOW

Use Case Name: Order Clothes Online
ID: UC-3
Priority: Medium
Actor: Online Shopper
Description: This use case describes how a customer orders clothing from an e-commerce site, with alternate steps when products are unavailable.
Trigger: Customer wants to buy clothes online.
Type: External

Preconditions:

1. Customer is logged into the shopping site.
2. Internet connection and website are functioning properly.

Normal Course

1. Customer browses the catalog and selects desired items.
2. System adds items to the cart and displays prices and sizes.
3. Customer proceeds to checkout.
4. Customer provides payment and shipping details.
5. System processes order and sends confirmation.

Alternate Flow

- A1. Some selected items are out of stock:
1. System displays a notification of unavailable items.
 2. System suggests similar or related items.
 3. Customer chooses to:
 - a. Replace items with alternatives, or
 - b. Remove unavailable items and continue checkout.
 4. System updates order summary and proceeds to payment.

Postconditions:

- Customer order is confirmed.
- System updates order and inventory records.

Exceptions:

- Payment gateway timeout: system saves order in “pending” status and alerts customer.
- Invalid shipping address: system prompts user to correct it.

EXAMPLE EXCEPTION FLOW

Use Case Name: Make a Chocolate Soufflé

ID: UC-4

Priority: Low

Actor: Home Cook

Description: This use case describes how a user follows a recipe to make a chocolate soufflé, including how to handle exceptions during preparation.

Trigger: User starts the "Chocolate Soufflé" recipe in a cooking app.

Type: External

Preconditions:

1. User has ingredients and tools ready.
2. Cooking app and timer are functional.

Normal Course

1. User opens recipe and reviews ingredient list.
2. System guides user through preparation steps.
3. User mixes batter and preheats oven.
4. System sets a timer for baking.
5. User removes soufflé when timer ends.

Exception Flow

E1. Oven malfunction or power outage occurs:

1. System detects interruption and alerts the user.
2. Timer pauses automatically.
3. Recipe progress is saved for later continuation.
4. System offers troubleshooting tips or option to restart recipe.

E2. Soufflé collapses during baking:

1. System provides troubleshooting guidance for next attempt.
2. User logs failed attempt for feedback tracking.

Postconditions:

- Soufflé is completed or cooking session saved.
- User can review progress history and retry.

COMMON MISTAKES

01

Writing use cases
that are too
technical or
system-focused.

03

Combining
multiple goals in
one case.

02

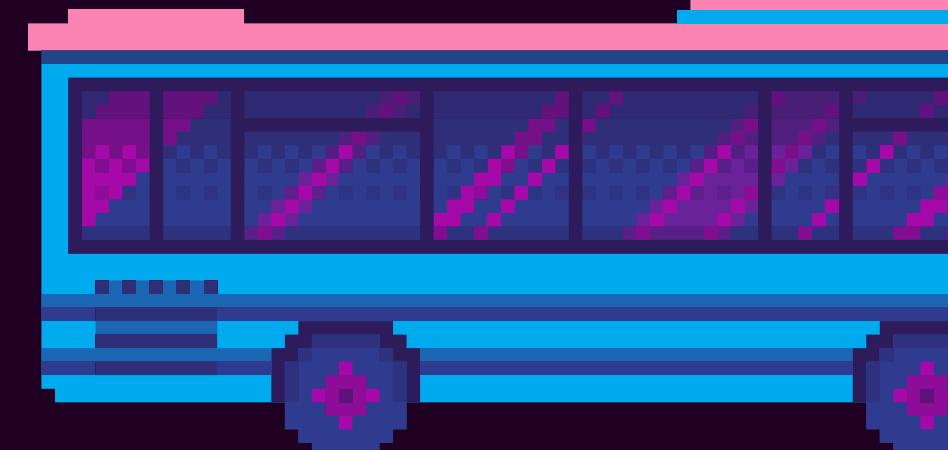
Skipping error
handling or
alternate flows.

04

Using
inconsistent
formats across
cases.

05

Overcomplicating
with too much
detail.



REAL-WORLD LESSONS

WORKSHEET/ REFLECTION
&
KAHoot

REFERENCES

📍 BA-EXPERTS - Learn Business Analysis by Example. (2014, October 5). What Is A Use Case. YouTube.

📞 Tilley, S. (2024) Requirements Engineering. In Systems Analysis and Design (12th ed.), Cengage Learning.

🌐 Dennis, A., Wixom, B., Roth, R. (2018) Use Cases. In Systems Analysis and Design (7th ed.), Wiley

Brandenburg, L. (2018, December 10). How to Write a Use Case. YouTube.

Brandenburg, L. (2018, December 10). How to Write a Use Case. YouTube.

THANK YOU

