

Methodology for Turning Concepts into Use Cases

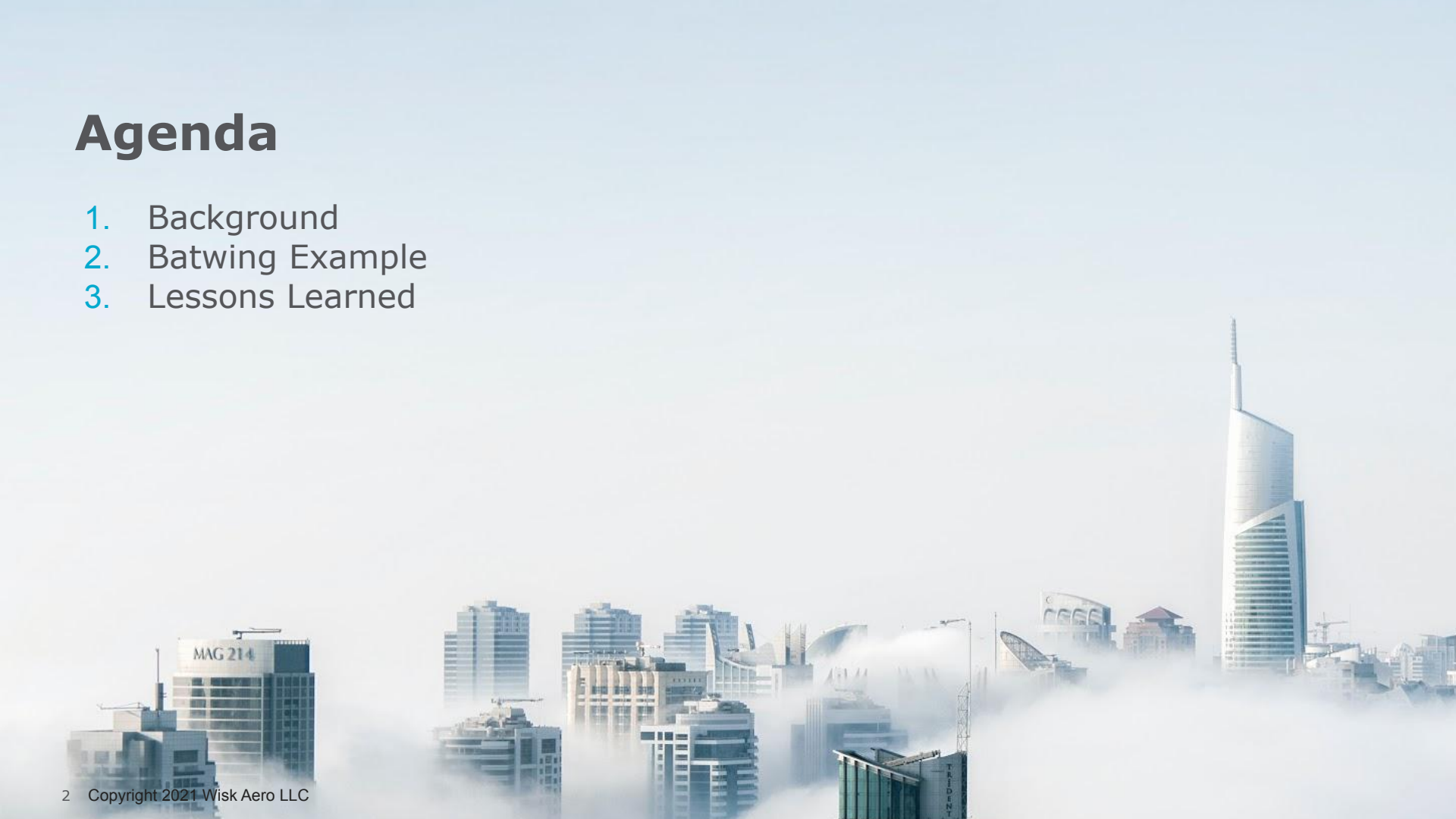
Jean Carlos Asencio - Systems Engineer at Wisk

August 31st, 2021

The Wisk logo is rendered in a bold, yellow, cursive script. The word "wisk" is written in lowercase letters, with a thick, sweeping underline that extends from the end of the word and angles upwards towards the top right corner of the slide.

Agenda

1. Background
2. Batwing Example
3. Lessons Learned



Background



About Us

- Wisk is an urban air mobility company dedicated to delivering safe, everyday flight for everyone.
- We are the first company in the U.S. to design, develop and fly an entirely autonomous air taxi for passenger use.

- Employees: ~340 with most in engineering, manufacturing, and flight test
- Locations: Bay Area CA (HQ), New Zealand and Atlanta
- Patents issued: over 140
- Test flights: over 1500 test flights with full-scale aircraft
- Backed by leaders in aviation: The Boeing Company and Kitty Hawk (investors and strategic partners)

Background

- Designing novel flight ops require we look at all of the behaviors and use cases that our system needs to perform
- MBSE is at the intersection of Methodology, Tools, and Language
 - All three are required for a successful MBSE approach
- Leveraging CONOPs as an integral component of Use Case Analysis streamlines and behavioral/functional development

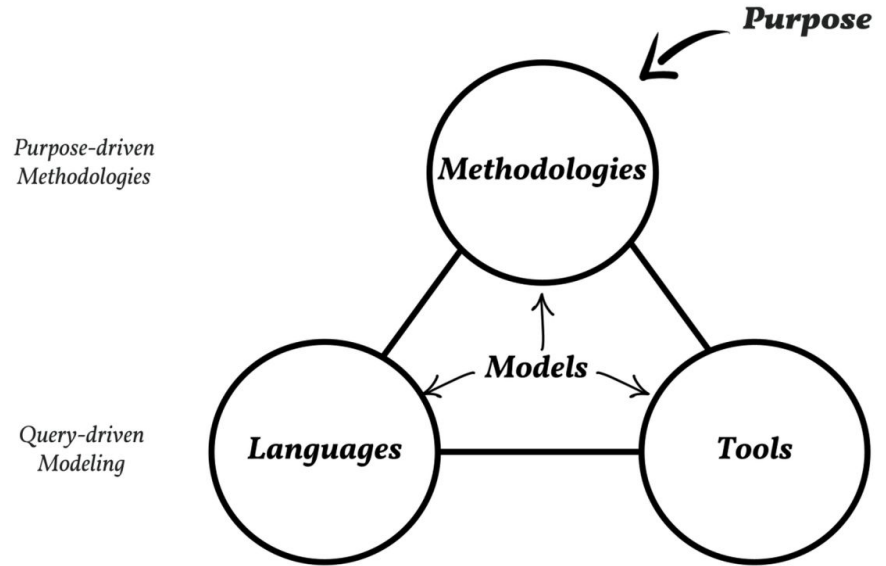
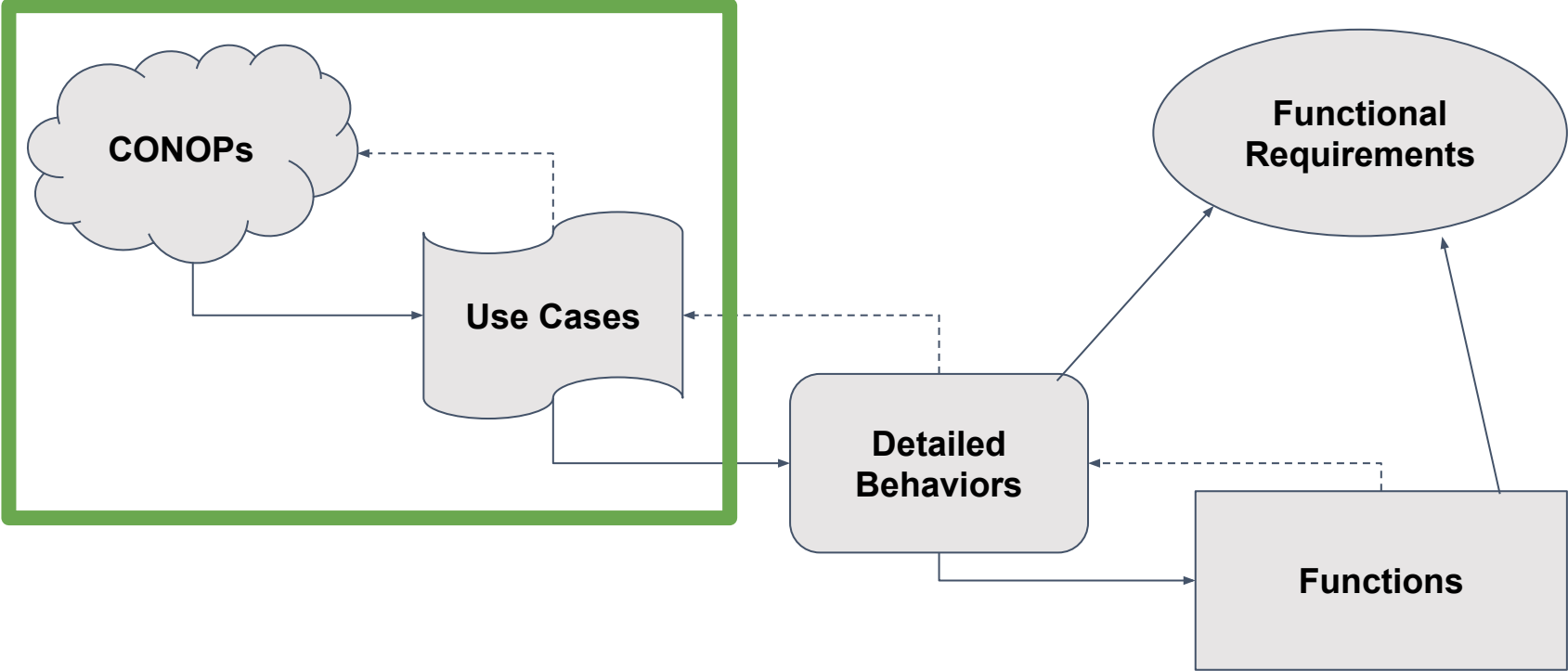


Figure from SYSMOD

Process Background



Batwing Example



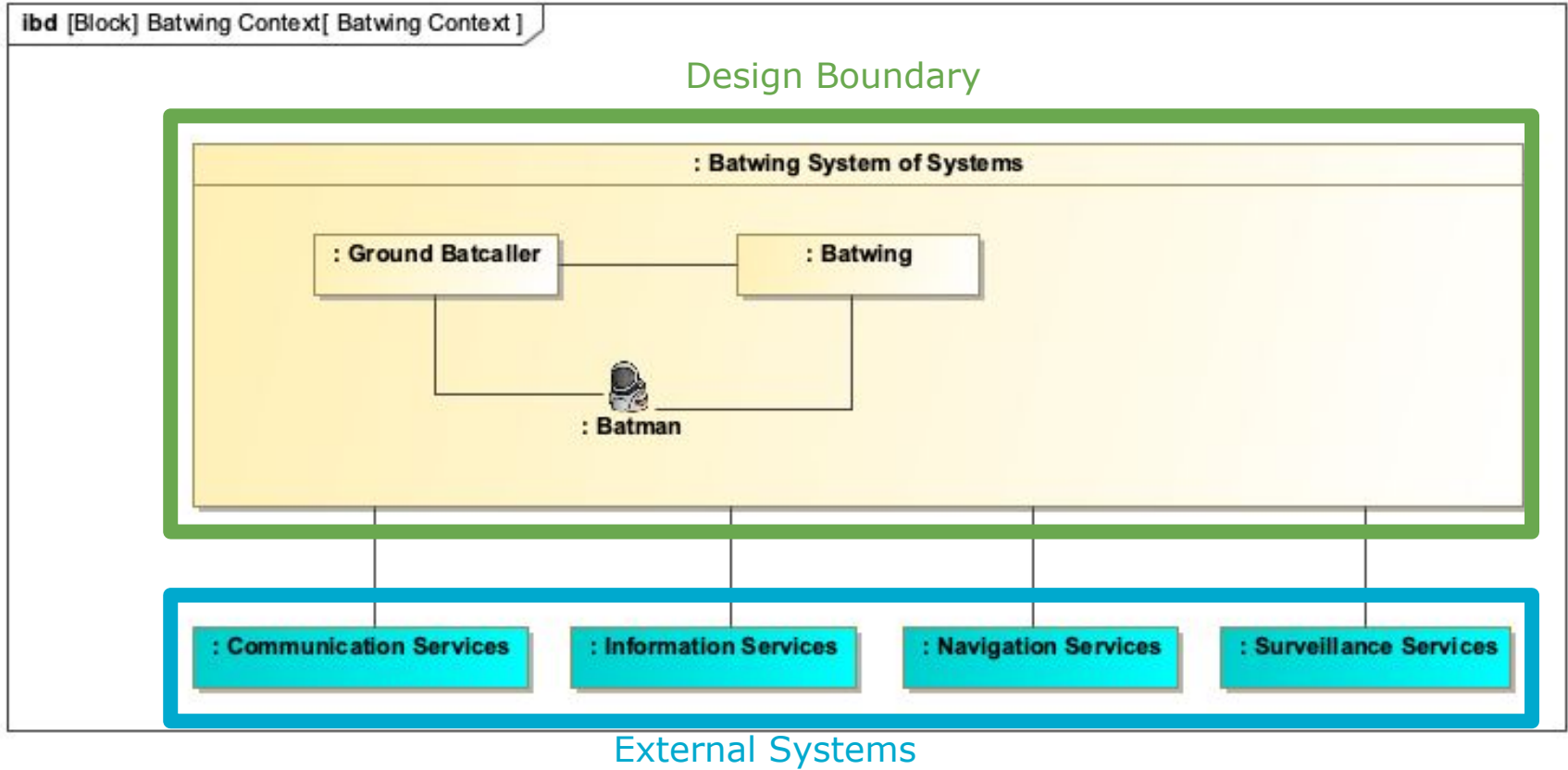
Batwing CONOPs Section (Notional)

Batman decided that his old Batwing needs to be updated, so he gets an engineering team together to design a new Batwing.

The CONOPs for the new batwing has the following paragraph:

"When Batman is fighting a villain and the villain takes flight, Batman will call the batwing using a remote watch. The Batwing will be able to plan the flight and fly towards batman without a human actor. The Batwing will be able to modify the mission in the case where Batman changes locations significantly."

Creating the Initial Context (Notional)



Identifying the Use Cases (Notional)

From the Batwing CONOPs paragraph, we get a diverse set of people in the room and decide, what are the use cases that come from this:

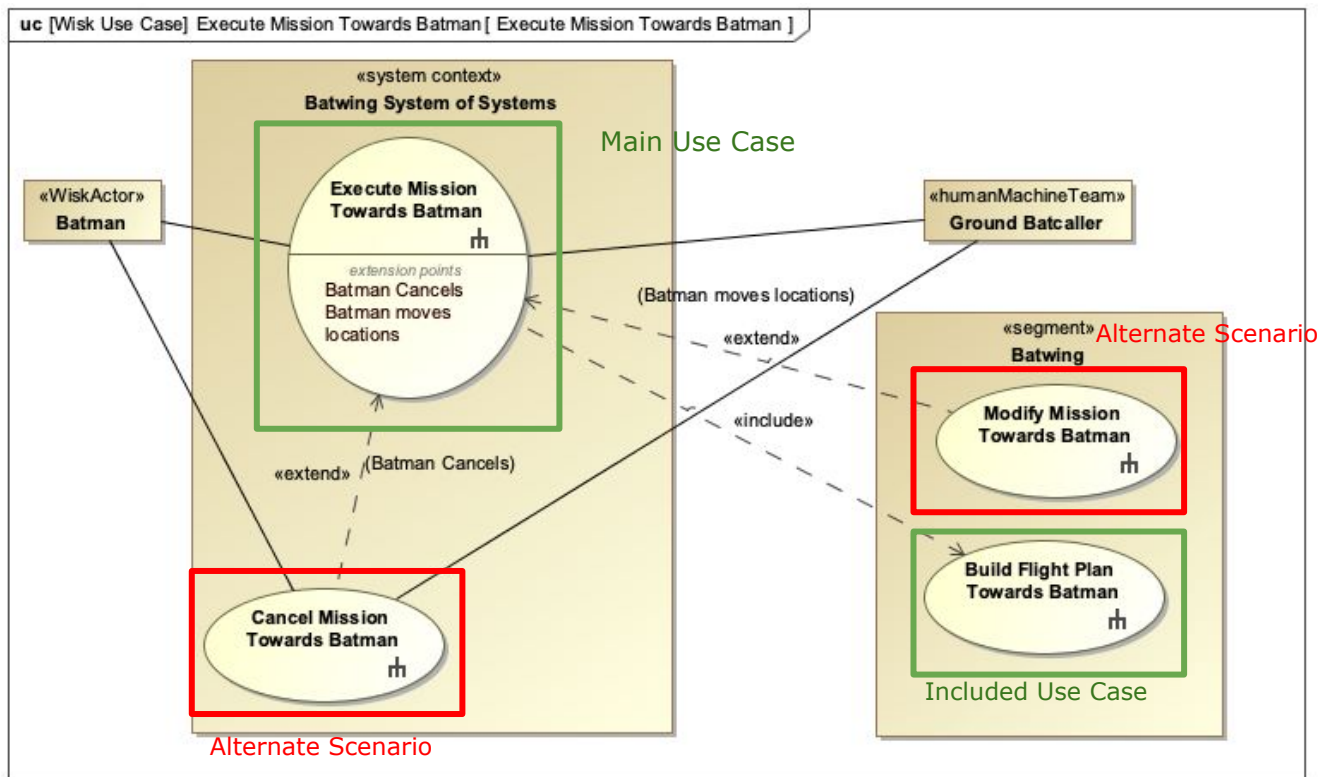
We can determine a nominal use case of "***Execute Mission towards Batman***"

We can determine a nominal use case included of "***Build Flight Plan Towards Batman***"

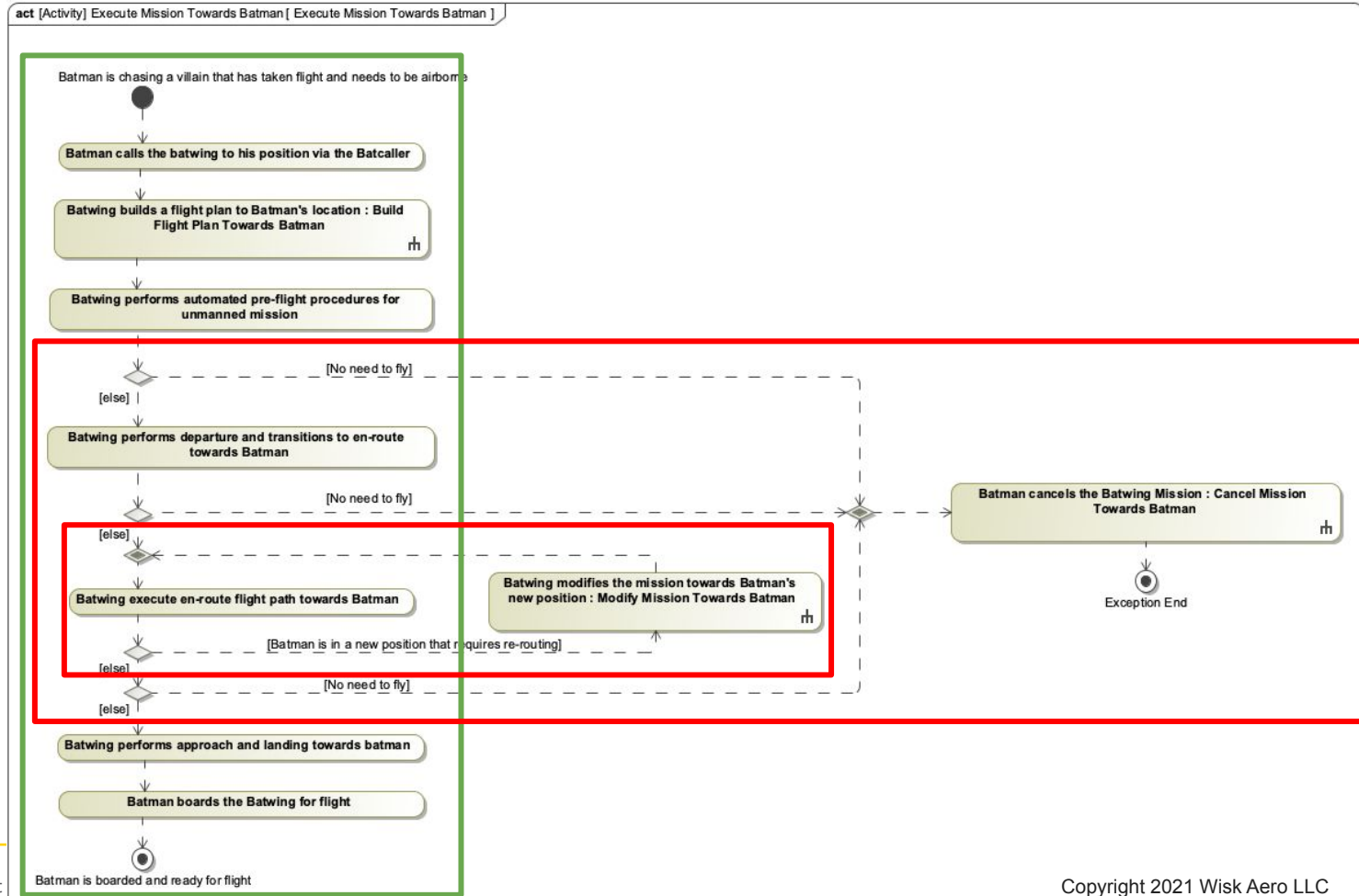
We can determine an off-nominal use case of "***Modify Mission towards Batman***" for when Batman may be on the move

We can determine an off-nominal use case of "***Cancel Mission towards Batman***" in the case where Batman doesn't need the Batwing anymore

Creating a Use Case Diagram (Notional)



Creating an Activity Diagram (Notional)



Checking for Completeness (Notional)

Specification of Use Case Description
The Use Case Description contains a list of specific Use Case Description properties.

Use Case Description

- Execute Mission Towards Batman
 - Documentation/Comments
 - Navigation/Hyperlinks
 - Usage in Diagrams
 - Use Case Scenario Sketch
 - Extension Points
 - Behaviors
 - Inner Elements
 - Relations
 - Tags
 - Constraints
 - Use Case Description**
 - Actors

Use Case ID	100
Actors Associated	Ground Batcaller [Design:Behavioral:Batman::Context:Ba Batman [Design:Behavioral:Batman::Context]
Subject	Batwing System of Systems [Design:Behavioral:Batman::G
Pre Condition	Batwing is in the batcave Batwing is fully operational
Trigger	● Batman is chasing a villain that has taken flight and needs ○ Batman calls the batwing to his position via the Batcaller [C ○ Batwing builds a flight plan to Batman's location:Build Flight ○ Batwing performs automated pre-flight procedures for uni ○ Batwing performs departure and transitions to en-route to ○ Batwing execute en-route flight path towards Batman [Des ○ Batwing performs approach and landing towards batman ○ Batman boards the Batwing for flight [Design:Behavioral:Ba
Main Success Flow	No need to fly No need to fly Batman is in a new position that requires re-routing No need to fly
Extension Conditions	* Exception End [Design:Behavioral:Batman::Use Case::Exe * Batman is boarded and ready for flight [Design:Behaviora
Post Condition	○ Cancel Mission Towards Batman [Design:Behavioral:Batm ○ Modify Mission Towards Batman [Design:Behavioral:Batm ○ Build Flight Plan Towards Batman [Design:Behavioral:Bat
Use Case Dependencies	
To Do	
Documentation	

Extension Conditions
(This property is read-only because it is used for the review purposes only)
Collects the ordered set of the Guards on the Decision Nodes in the Classifier Behavior of a use case that are part of the main success flow. This capability can also handle simple parallel actions in the main success flow

Type here to filter properties

Close Back Forward Help

Name: Execute Mission Towards Batman

ID: UC 100

Scope: Batwing System of Systems

Traceability: All from Batwing CONOP

- Section X.X

Precondition:

- Batwing is in the batcave
- Batwing is fully operational

Trigger: Batman is chasing a villain that has taken flight and needs to be airborne

Postcondition: Batman is boarded and ready for flight

Main Success Flow:

1. Batman calls the batwing to his position via the Batcaller
2. Batwing builds a flight plan to Batman's location [UC 101]
3. Batwing performs automated pre-flight procedures for unmanned mission
4. Batwing performs departure and transitions to en-route towards Batman
5. Batwing executes en-route flight path towards Batman
6. Batwing performs approach and landing towards Batman
7. Batman boards the batwing for flight

Extensions:

3a. Batman no longer needs to fly

3a1. Batman cancels the Batwing mission [UC 102]

4a. Batman no longer needs to fly

4a1. Batman cancels the Batwing mission [UC 102]

5a. Batman is in a new position that requires the Batwing to re-route

5a1. Batwing modifies the mission towards Batman's new position [UC 103]

Proceed to Step 5

5b. Batman no longer needs to fly

5b1. Batman cancels the Batwing mission [UC 102]

Lessons Learned

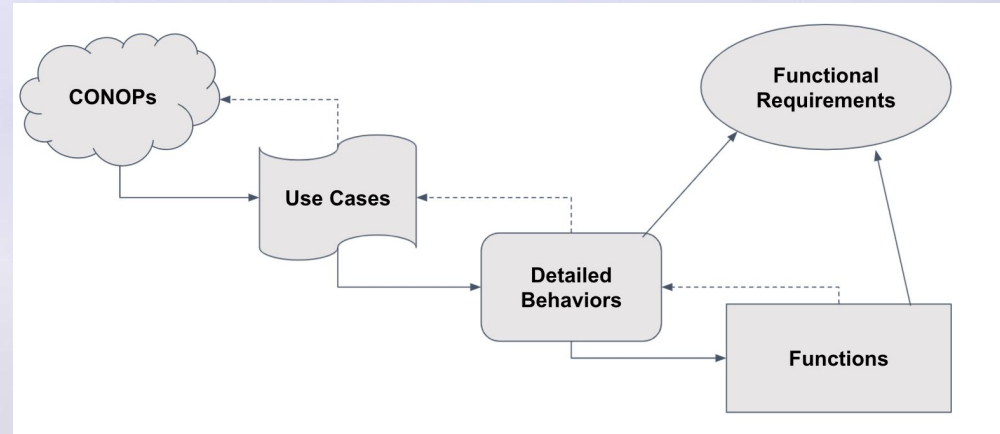


Lessons Learned From Use Cases

- Include a diverse set of Subject Matter Experts in your use case development
- Use Cases are mainly for the top two levels of the design, for us L0 and L1
- Good use case development will lead to good behavior development, which then leads to great functional requirements and functional development
- All audiences are quickly able to assimilate the visual use cases from MBSE tools compared to the textual formats (which can become dense)
 - Textual Use Case Descriptions can be a good starting point for users who are new to MBSE
- The cost paid by representing the use case steps in the activity diagrams without swimlanes was worth the gain of improved initial communication

In Conclusion

- Use case development is pivotal in getting from concepts to design
- Always include a diverse set of experts in your development
- MBSE is a great facilitator for use case development



wisk