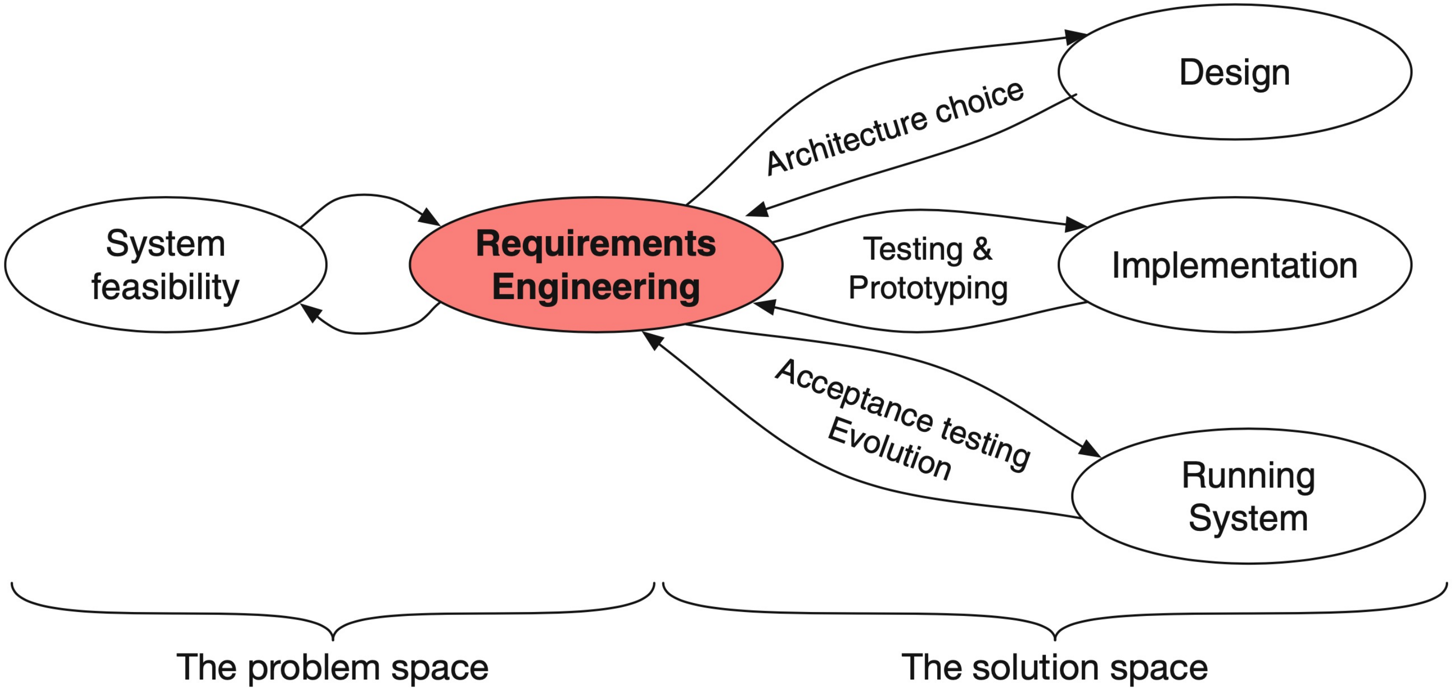
Requirements, Use Cases, and Scenarios



Amel Bennaceur, Thein Than Tun, Yijun Yu, Bashar Nuseibeh:
Requirements Engineering. Handbook of Software Engineering 2019: 51-92

Some Examples for Discussion

- The Web application shall be available online by September 1, 2024.
- The Web application shall support a minimum of 2,500 concurrent users.
- Java EE shall be used as development platform.
- All customer data shall be securely submitted.
- The user interface shall support layouts for different customer groups.
- An arbitrary user shall be able to find a desired product in less than three minutes.
- A user shall be able to select an icon to display articles included in the shopping cart at any given time.

Types of Requirements

- **Functionality.** What is the software supposed to do?
- **External interfaces.** How does the software interact with people, the system's hardware, other hardware, and other software?
- **Performance.** What is the speed, availability, response time, recovery time, … of various software functions?
- **Attributes.** What are the portability, correctness, maintainability, security, … considerations?
- **Design constraints imposed on an implementation.** Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.?

Source: IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications

Where do Requirements Come From?

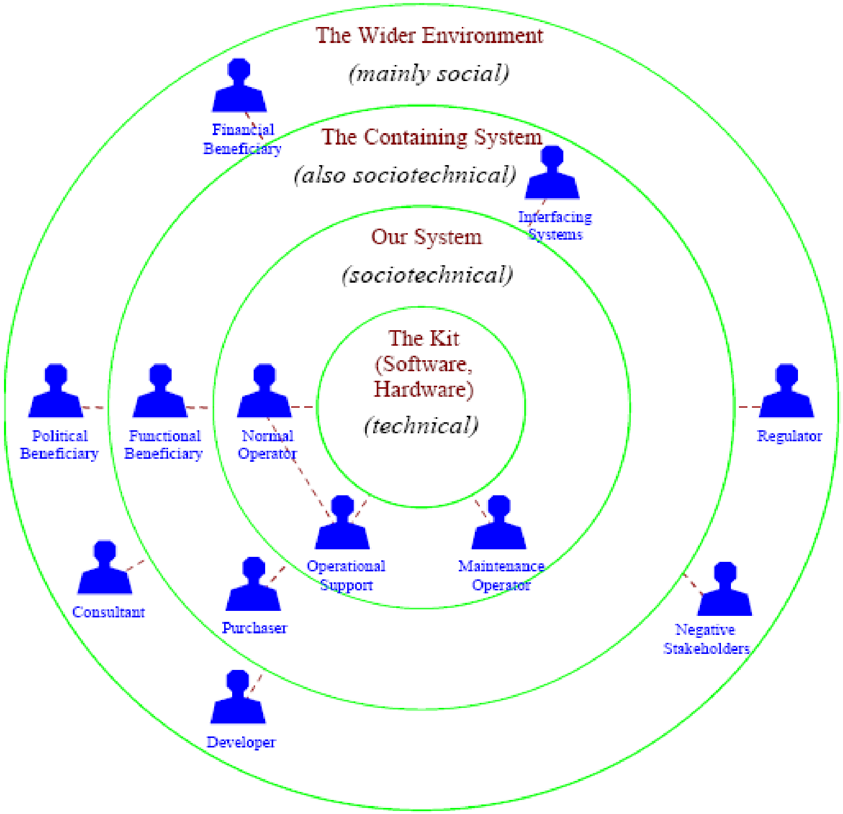
“A **stakeholder** is someone who has a justifiable claim to be allowed to influence the requirements”

- **Users** of the system und **development;**
- **People** whose lives are **affected by the system**, such as users, clients, and suppliers;
- **Managers** who are **concerned for the system to succeed**, although they do not use it as such;
- **Regulators** such as local and state governments and standards bodies, which are **concerned about the effects** the system may have **in its environment**.

Source: Ian Alexander, Writing Better Requirements

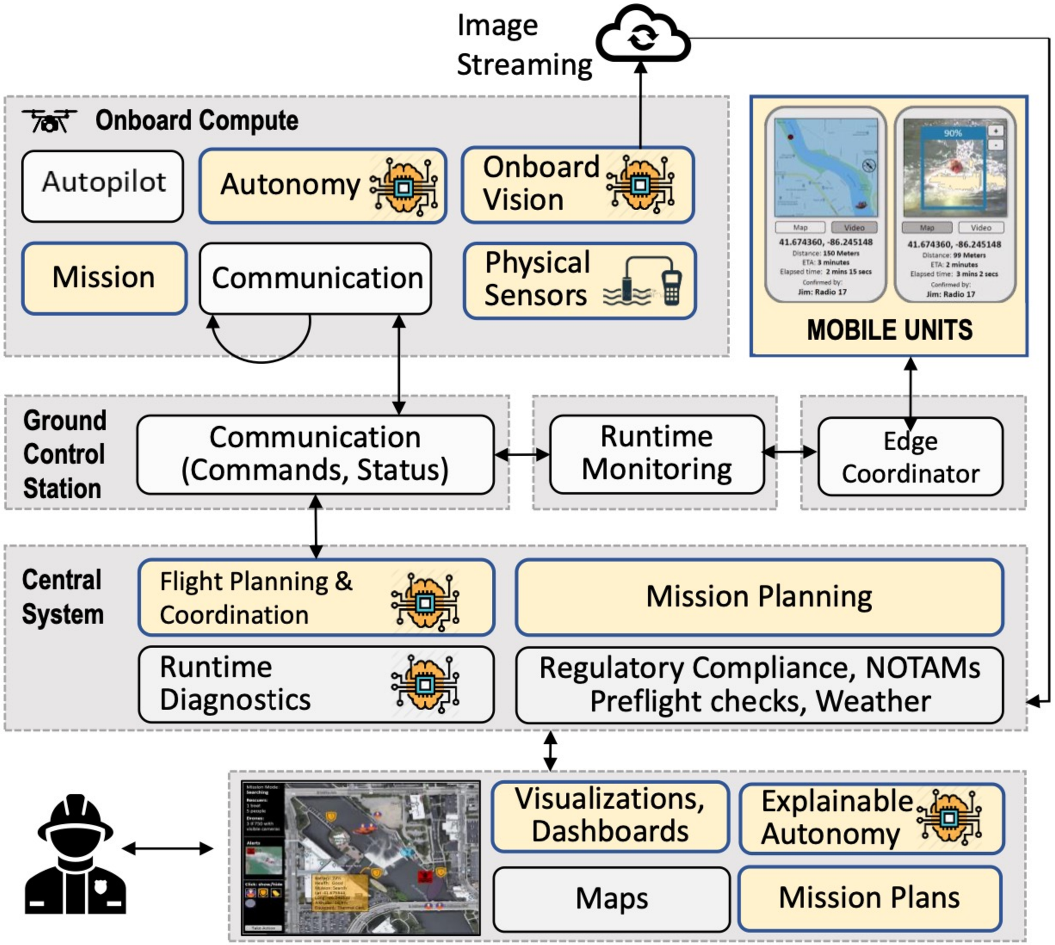
The Onion Model

1. **The Kit (or The Product):** the hardware and software under development
2. **Our System:** The Kit plus its human Operators and the rules governing its operation
3. **The Containing System:** Our System plus any human Beneficiaries of Our System (whether they are involved in operations or not)
4. **The Wider Environment:** The Containing System plus any other Stakeholders



Source: Ian Alexander and Suzanne Robertson, Understanding Project Sociology by Modeling Stakeholders

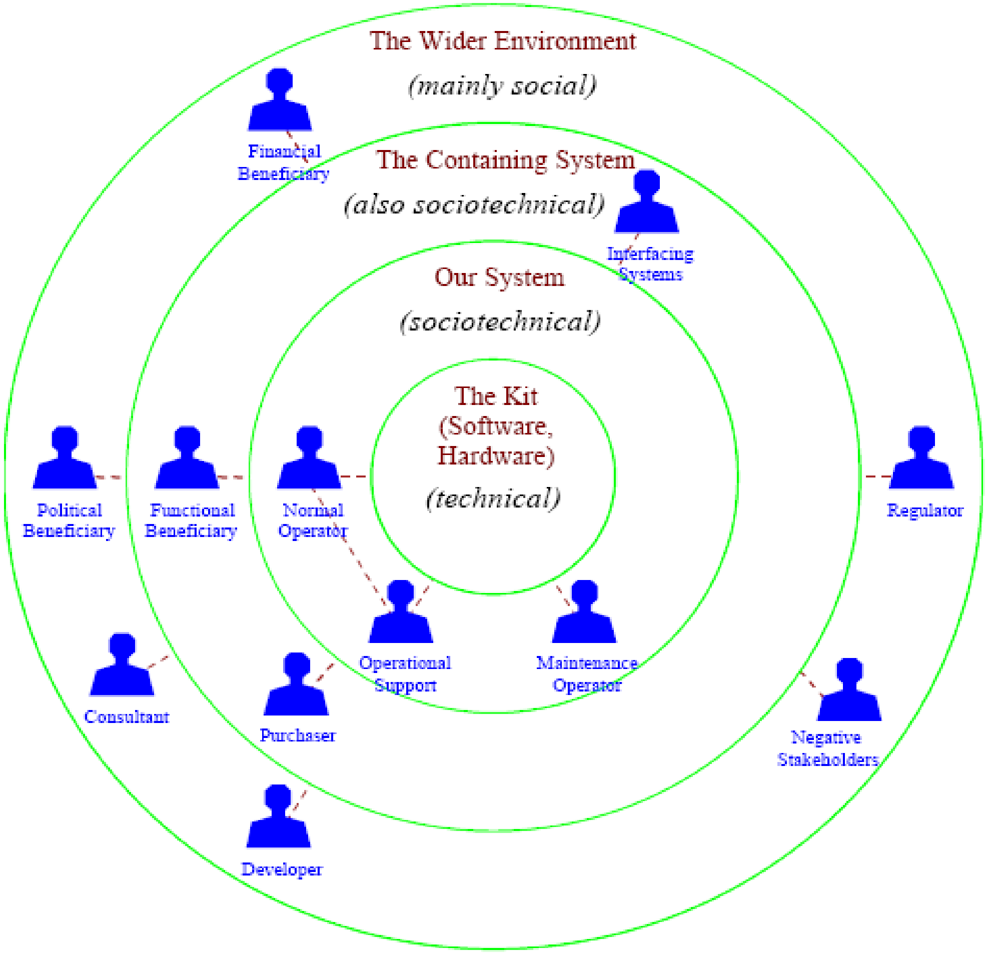
Example: Dronology



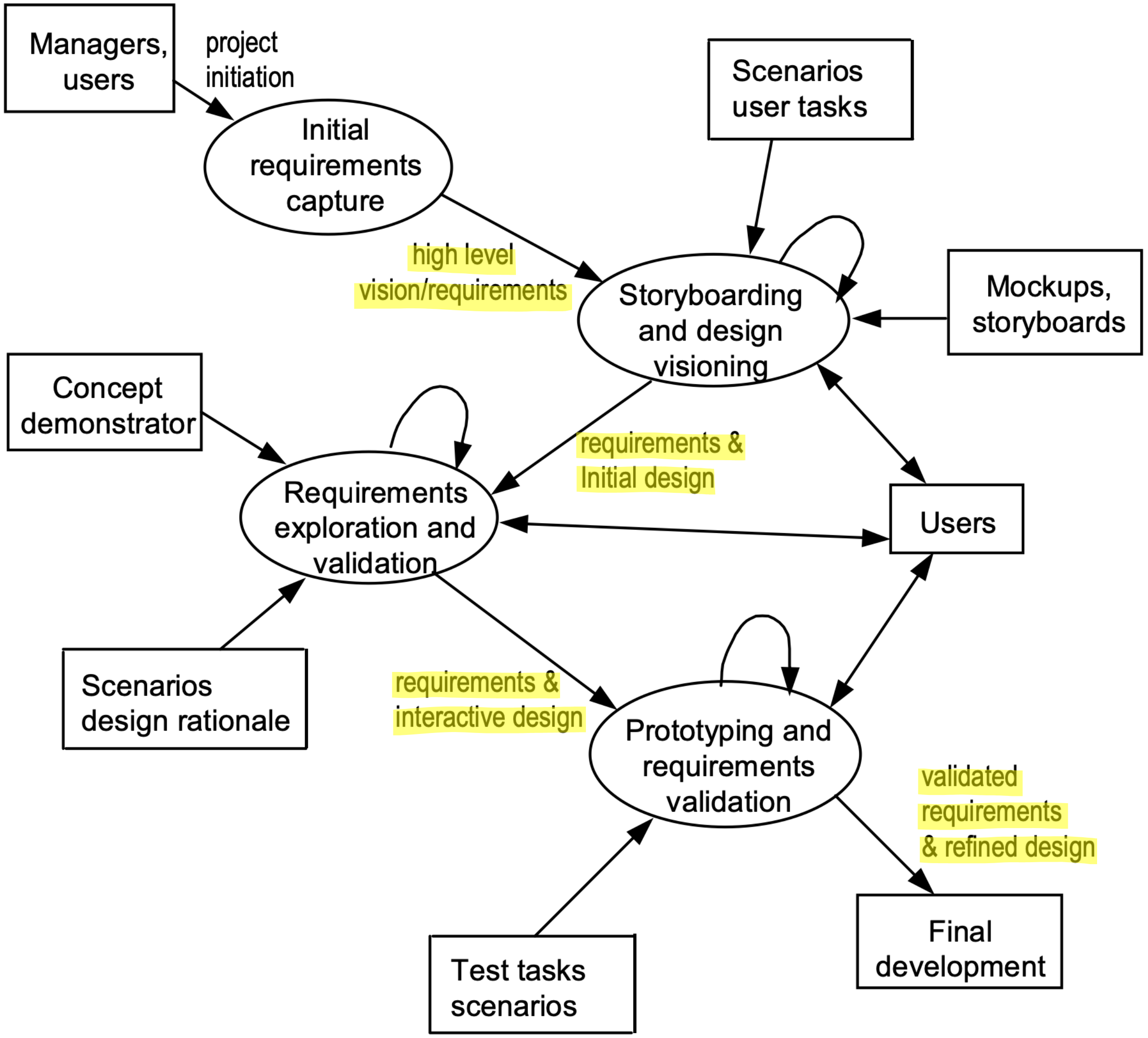
J. Cleland-Huang, A. Agrawal, M. N. Al Islam, E. Tsai, M. van Speybroeck, M. Vierhauser: Requirements-driven configuration of emergency response missions with small aerial vehicles. SPLC (A) 2020: 26:1-26:12

Mini Exercise

Find stakeholders for the Dronology system using Alexander’s Onion Model.



Scenario-based Requirements Engineering Process



Managers, users
project initiation
Initial requirements capture
high level vision/requirements
Scenarios user tasks
Storyboarding and design visioning
Mockups, storyboards
Concept demonstrator
requirements & Initial design
Requirements exploration and validation
Users
Scenarios design rationale
requirements & interactive design
Prototyping and requirements validation
validated requirements & refined design
Test tasks scenarios
Final development

Alistair G. Sutcliffe: Scenario-Based Requirements Engineering. RE 2003: 320-329

Use Cases and Scenarios

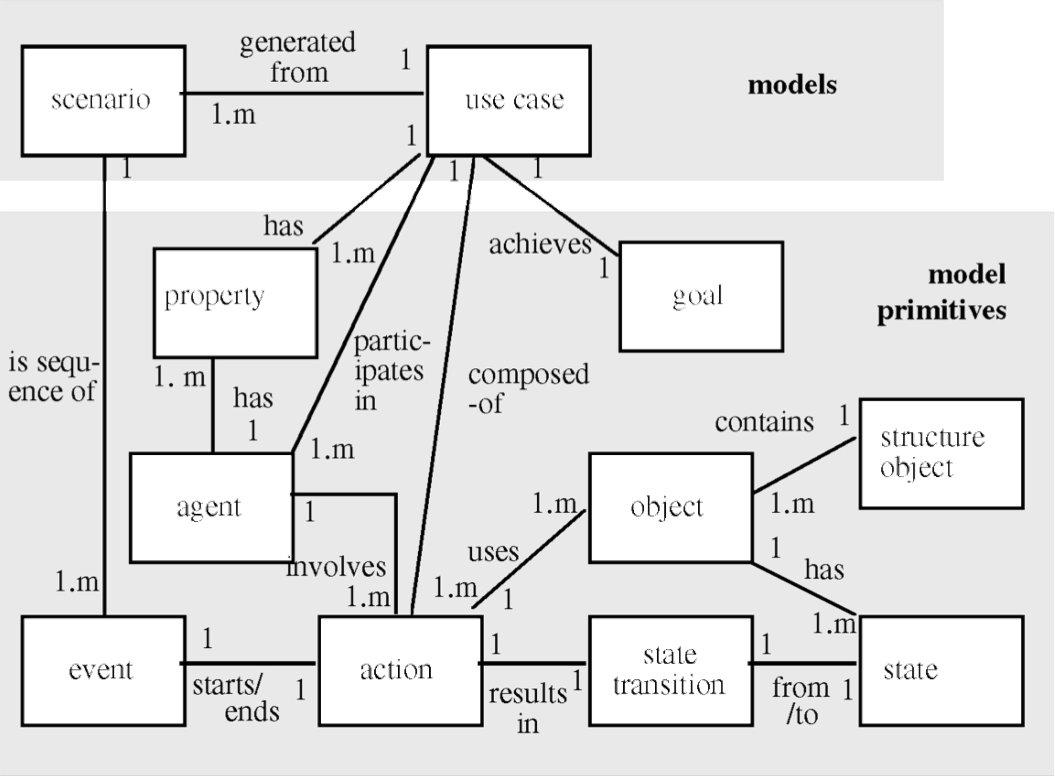
- **Use Cases.** Model the functionality and behavior offered by a system.
  - Withdraw Cash (Bank Customer)
  - Make Payment (Bank Customer)

- **Scenarios.** A sequence of actions that is one possible pathway through a use case. Each scenario may describe both normal and abnormal behavior.
  - Withdraw Money (*Normal Course* Scenario)
  - Withdraw Money (*Alternative Course* Scenario if only 100€ bills are left)

→ Multiple scenarios may be specified for one use case and each scenario represents an instance or example of actions that could happen.

Use Cases and Scenarios (Example)

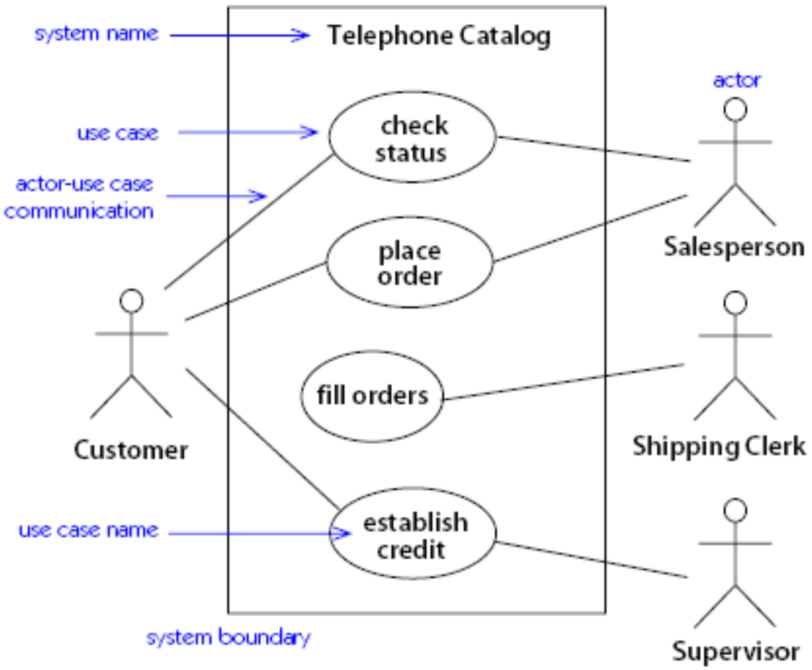
- *Use Case.* Withdraw Cash
- *Goal* of Actor. Get Cash from ATM
- *Scenario.* Normal Course
- *Event.*
  - (Start): Insert Card into ATM
  - (End): Take Money
- *Agent.* Customer
- *Action.*
  - Customer enters pin, customer
  - Customer takes Card (*Object*)
  - ...
- *State Transition.* Withdraw Fund
- *State.* Account balance reduced by the withdrawn amount



Alistair G. Sutcliffe, Neil A. M. Maiden, Shailey Minocha, Darrel Manuel: Supporting Scenario-Based Requirements Engineering. IEEE Trans. Software Eng. 24(12): 1072-1088 (1998)

What is a Use Case?

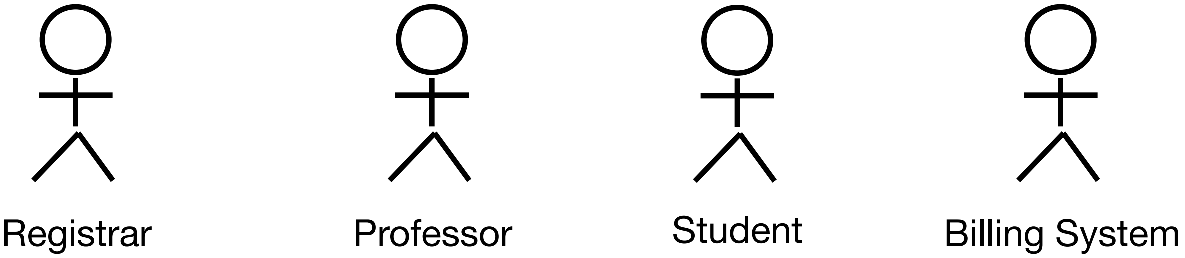
"[…] a series of interactions between an outside entity and the system, which ends by providing business value."



Source: Use Cases: Requirements in Context, Daryl Kulak, Eamonn Guiney, July 25, 2003, ISBN 0-321-15498-3

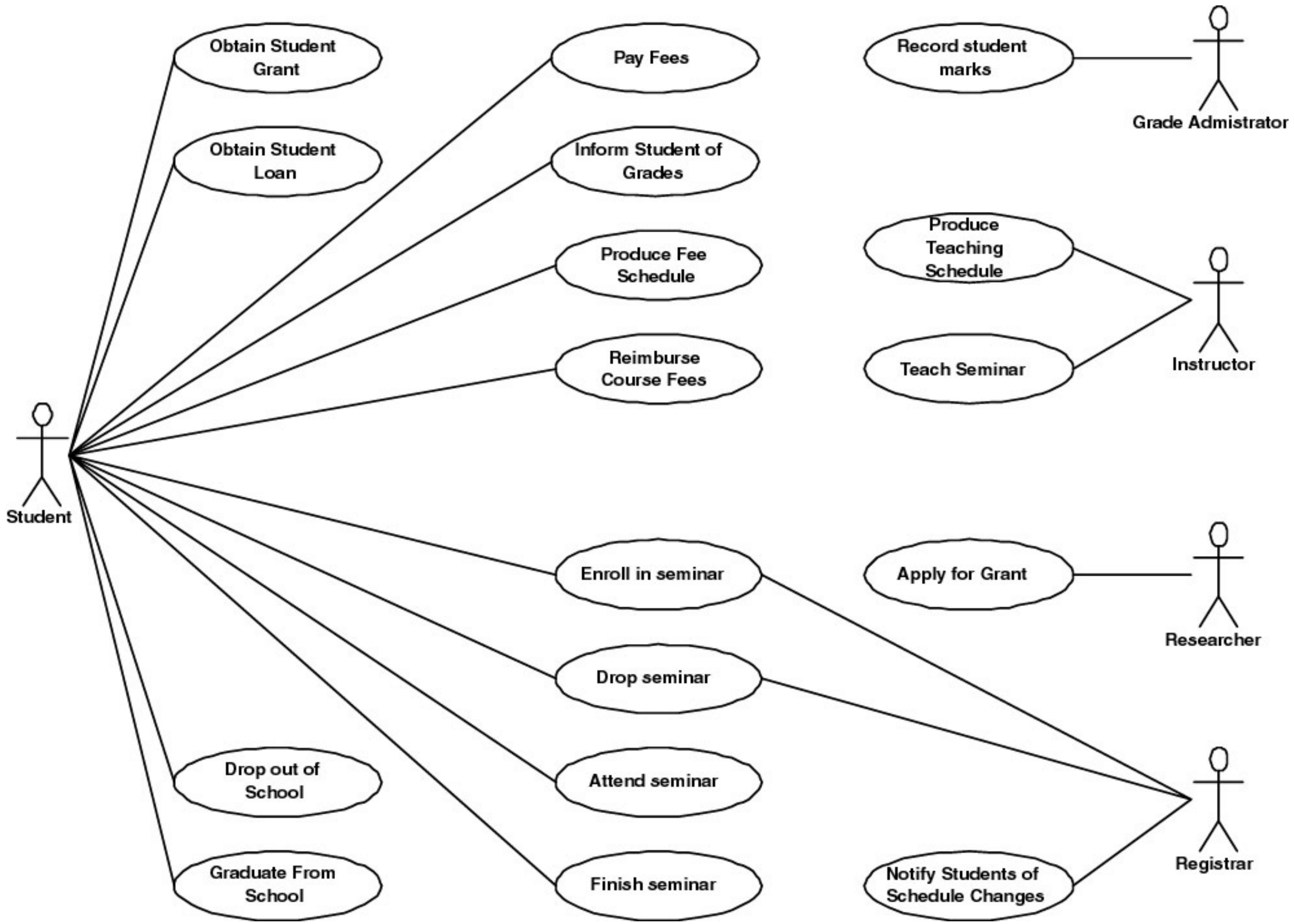
What is an Actor?

- Someone or some thing that interacts with the system under development
- External to the system, can be human or machine
- Primary actors use a system in daily activities
- Secondary actors enable the primary actors to the use system



Use Case Diagram

Source: The Object Primer 3rd Edition, Scott Ambler, http://www.agilemodeling.com/artifacts/useCaseDiagram.htm



Registrar

What is a Scenario?

- „A type of work item, **describing a specific usage of the envisaged software system** by a particular persona. Scenarios should be goal directed. As a persona attempts to reach a goal, the scenario records the specific steps taken in attempting to reach that goal.“

  Agile Management

- “**Each execution of a use case** can be viewed as an instance of that use case and can be specified using scenarios.”

  Neil Maiden

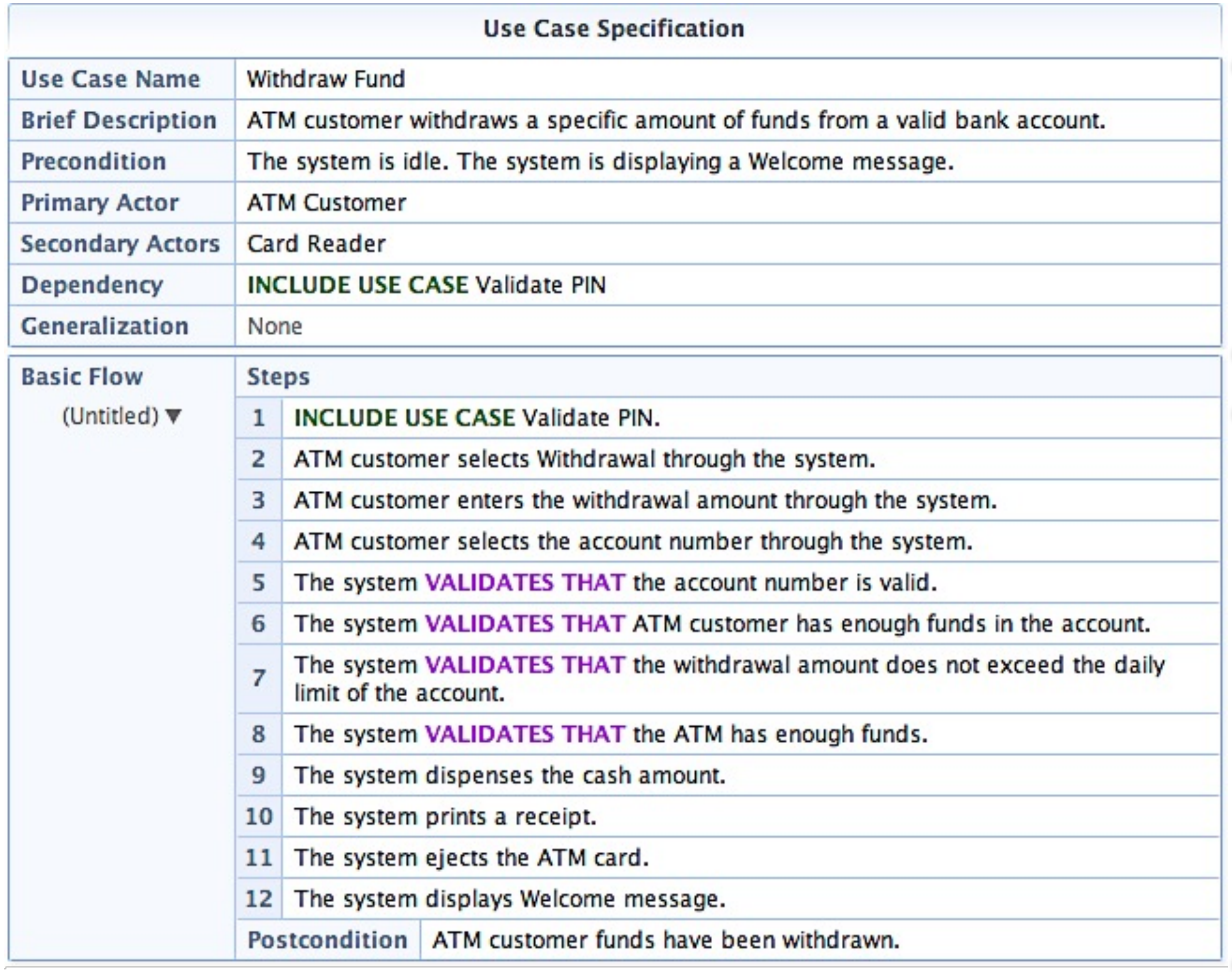
- “…they focus on describing particular instances of use, and on **a user’s view of what happens, and why…**”

  J.M. Carroll

Specifying Use Case Scenarios

- A flow of actions, written from an actor's point of view
- Defines what the system must provide to the actor when executing the use cases
- Defines how the use case starts and ends (events)
- Different kinds of scenarios
  - Normal flow of events
  - Alternative flow of events
  - Exceptional flow of events

Example: Normal Flow of Events



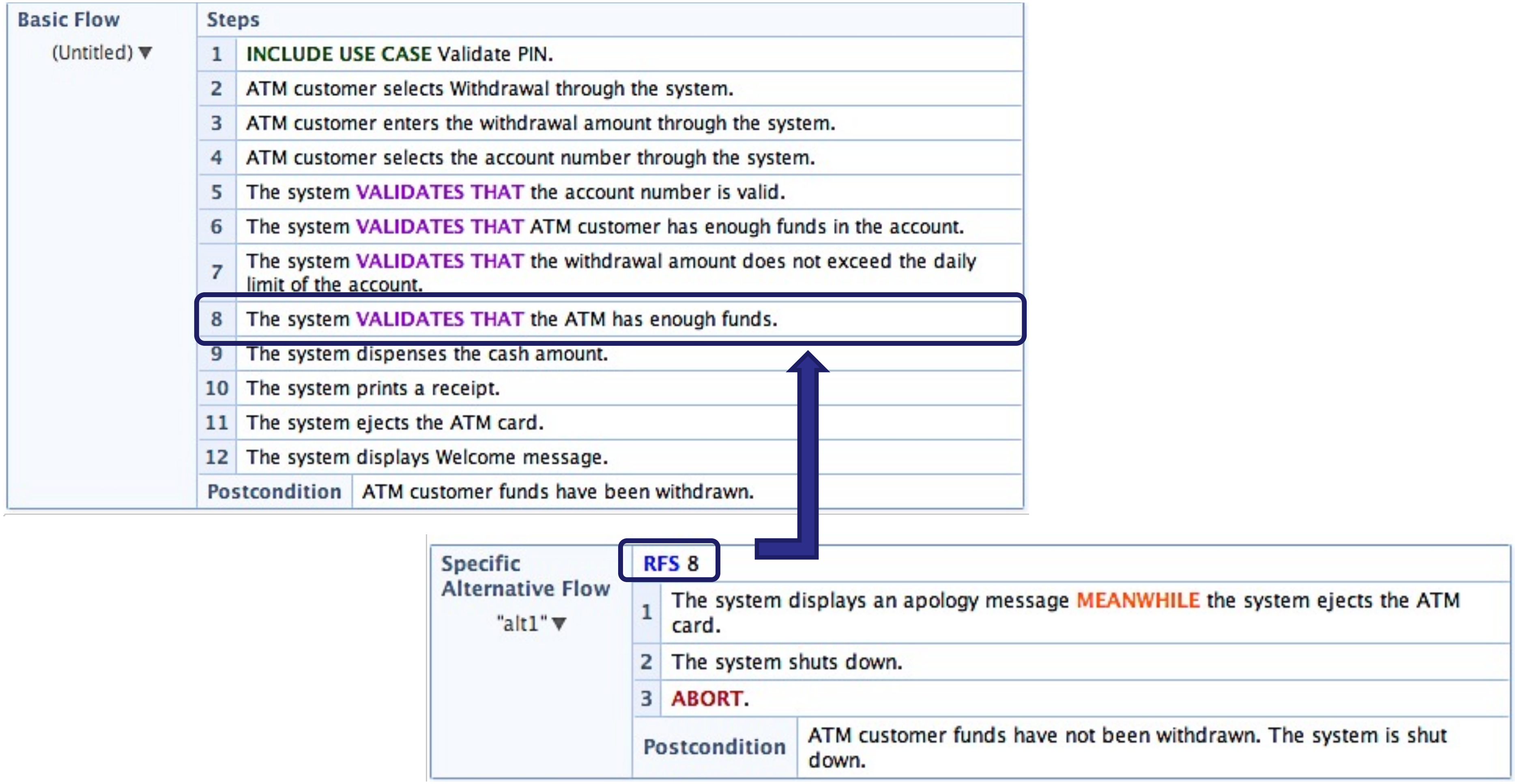
Use Case Specification		
Use Case Name	Withdraw Fund	
Brief Description	ATM customer withdraws a specific amount of funds from a valid bank account.	
Precondition	The system is idle. The system is displaying a Welcome message.	
Primary Actor	ATM Customer	
Secondary Actors	Card Reader	
Dependency	INCLUDE USE CASE Validate PIN	
Generalization	None	
Basic Flow	Steps	
(Untitled) ▼	1	INCLUDE USE CASE Validate PIN.
	2	ATM customer selects Withdrawal through the system.
	3	ATM customer enters the withdrawal amount through the system.
	4	ATM customer selects the account number through the system.
	5	The system VALIDATES THAT the account number is valid.
	6	The system VALIDATES THAT ATM customer has enough funds in the account.
	7	The system VALIDATES THAT the withdrawal amount does not exceed the daily limit of the account.
	8	The system VALIDATES THAT the ATM has enough funds.
	9	The system dispenses the cash amount.
	10	The system prints a receipt.
	11	The system ejects the ATM card.
	12	The system displays Welcome message.
	Postcondition	ATM customer funds have been withdrawn.

Typical Fields

Basic Flow

Tao Yue, Lionel C. Briand, Yvan Labiche: Facilitating the transition from use case models to analysis models: Approach and experiments. ACM Trans. Softw. Eng. Methodol. 22(1): 5 (2013)

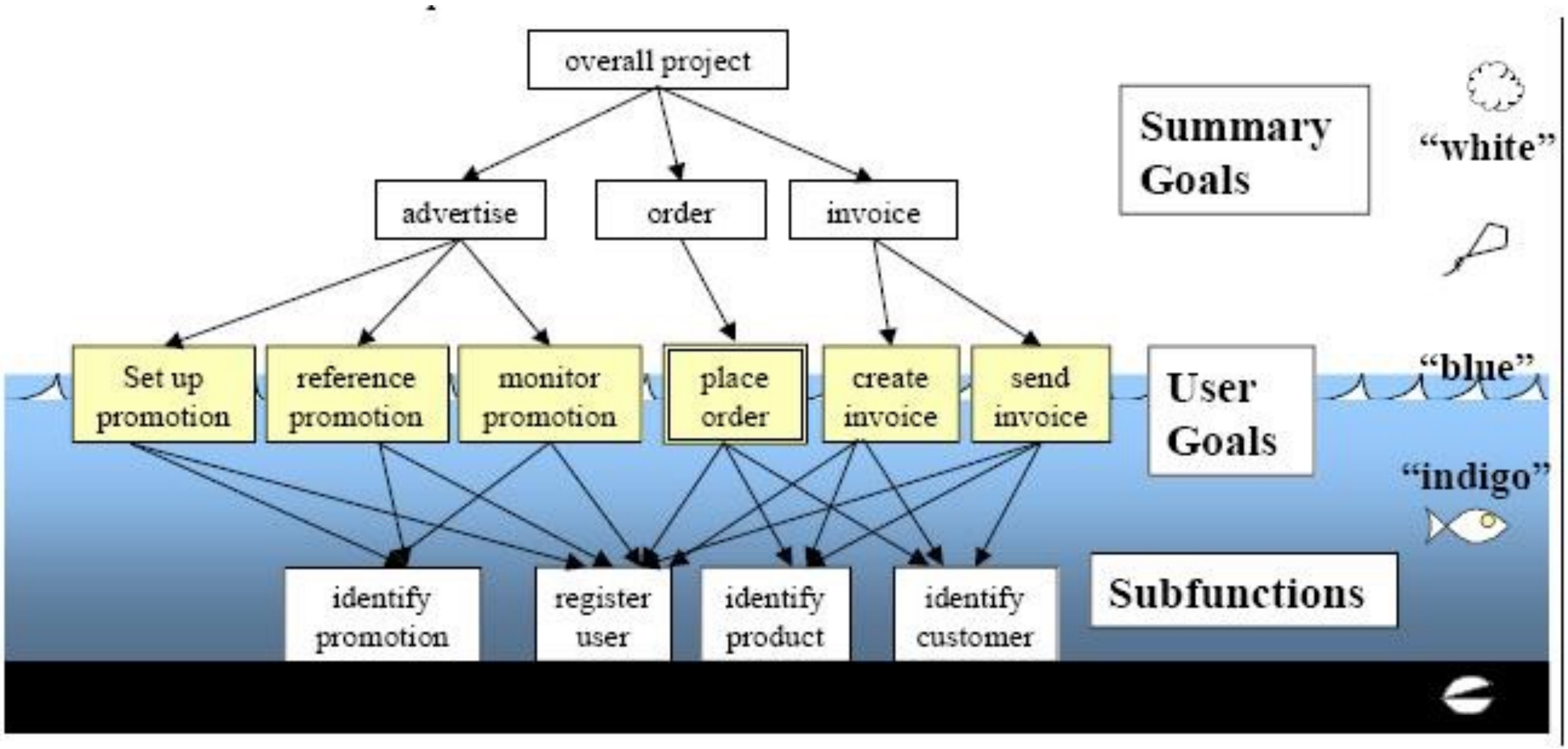
Example: Alternative Flow Events



Basic Flow	Steps	
(Untitled) ▼	1	INCLUDE USE CASE Validate PIN.
	2	ATM customer selects Withdrawal through the system.
	3	ATM customer enters the withdrawal amount through the system.
	4	ATM customer selects the account number through the system.
	5	The system VALIDATES THAT the account number is valid.
	6	The system VALIDATES THAT ATM customer has enough funds in the account.
	7	The system VALIDATES THAT the withdrawal amount does not exceed the daily limit of the account.
	8	The system VALIDATES THAT the ATM has enough funds.
	9	The system dispenses the cash amount.
	10	The system prints a receipt.
	11	The system ejects the ATM card.
	12	The system displays Welcome message.
	Postcondition	ATM customer funds have been withdrawn.

Specific Alternative Flow	RFS 8	
"alt1" ▼	1	The system displays an apology message MEANWHILE the system ejects the ATM card.
	2	The system shuts down.
	3	ABORT.
	Postcondition	ATM customer funds have not been withdrawn. The system is shut down.

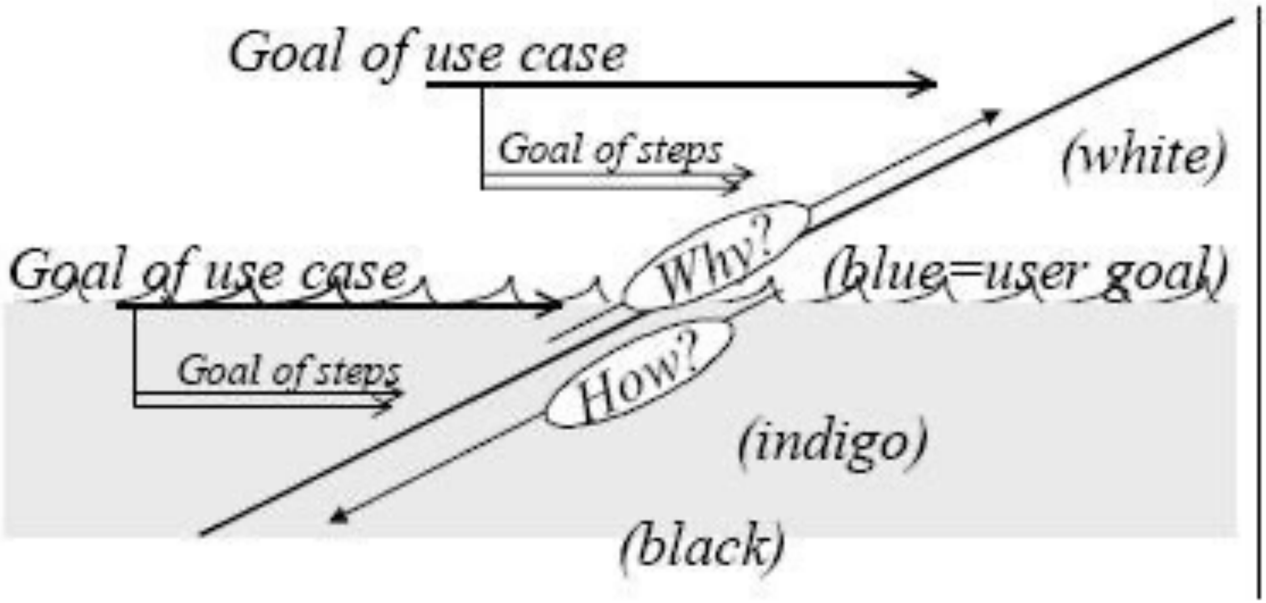
Granularity of Use Cases



Source: Writing Effective Use Cases Use Cases, Alistair Cockburn, 2000, Addison-Wesley, ISBN: 0201702258

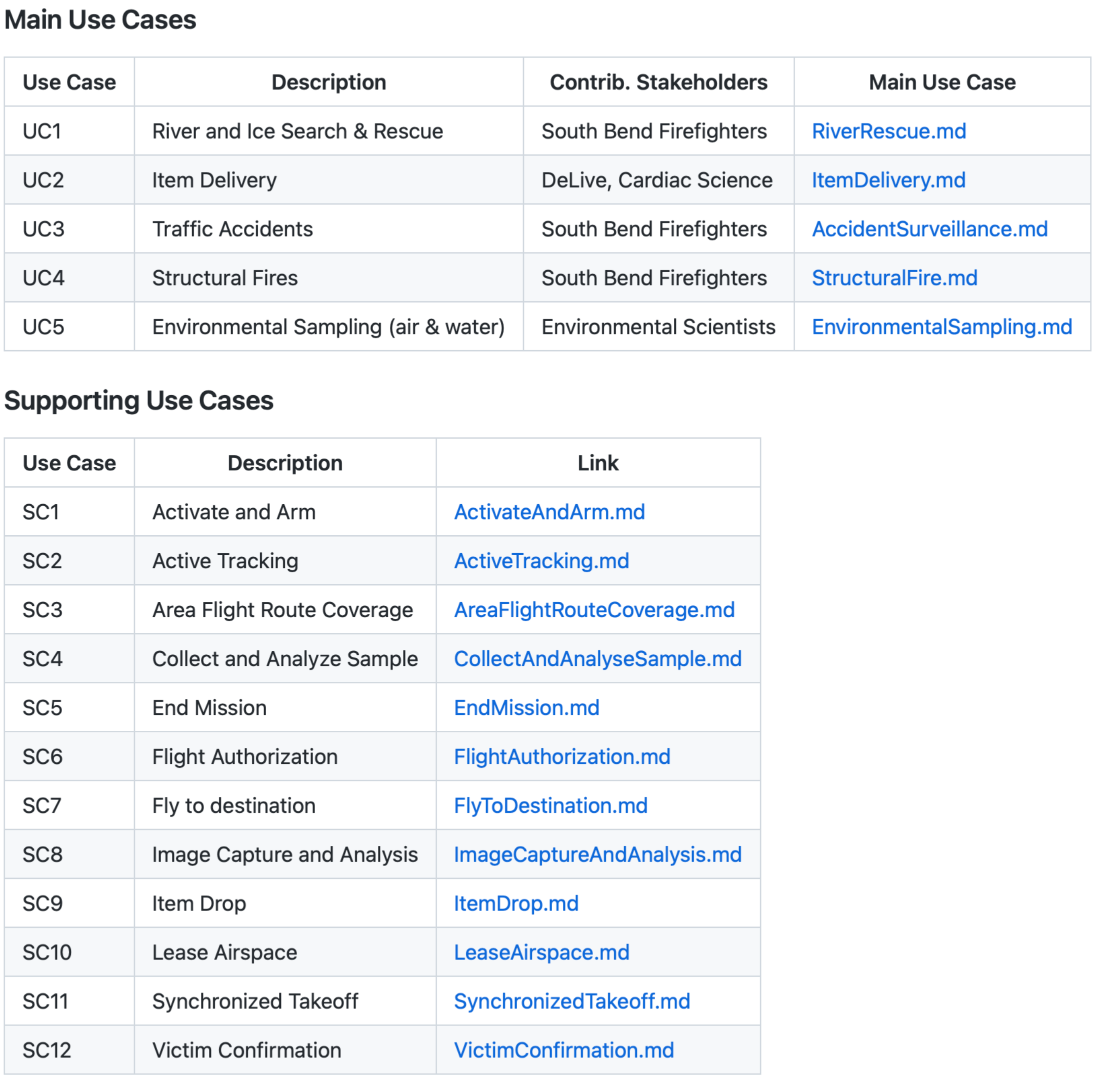
Finding the Right Level of Detail

- Asking “why” can help to “go up”
- Asking “how“ can help to “go down”



Source: Writing Effective Use Cases Use Cases, Alistair Cockburn, 2000, Addison-Wesley, ISBN: 0201702258

Dronology Use Cases



Main Use Cases

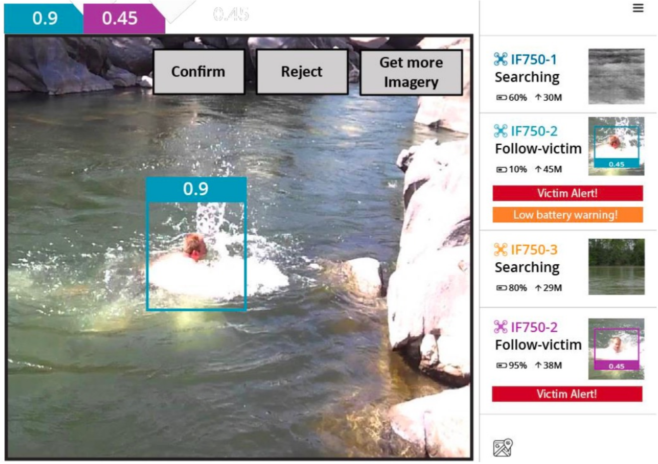
Use Case	Description	Contrib. Stakeholders	Main Use Case
UC1	River and Ice Search & Rescue	South Bend Firefighters	RiverRescue.md
UC2	Item Delivery	DeLive, Cardiac Science	ItemDelivery.md
UC3	Traffic Accidents	South Bend Firefighters	AccidentSurveillance.md
UC4	Structural Fires	South Bend Firefighters	StructuralFire.md
UC5	Environmental Sampling (air & water)	Environmental Scientists	EnvironmentalSampling.md

Supporting Use Cases

Use Case	Description	Link
SC1	Activate and Arm	ActivateAndArm.md
SC2	Active Tracking	ActiveTracking.md
SC3	Area Flight Route Coverage	AreaFlightRouteCoverage.md
SC4	Collect and Analyze Sample	CollectAndAnalyseSample.md
SC5	End Mission	EndMission.md
SC6	Flight Authorization	FlightAuthorization.md
SC7	Fly to destination	FlyToDestination.md
SC8	Image Capture and Analysis	ImageCaptureAndAnalysis.md
SC9	Item Drop	ItemDrop.md
SC10	Lease Airspace	LeaseAirspace.md
SC11	Synchronized Takeoff	SynchronizedTakeoff.md
SC12	Victim Confirmation	VictimConfirmation.md

https://github.com/SAREC-Lab/sUAS-UseCases/blob/SPLC-2020/README.md

Dronology Use Case:
Ice Search and Rescue (Success Scenario)



Description

UAV(s) dispatched with a flotation device for ice rescue

Primary Actor

Drone Commander

Trigger

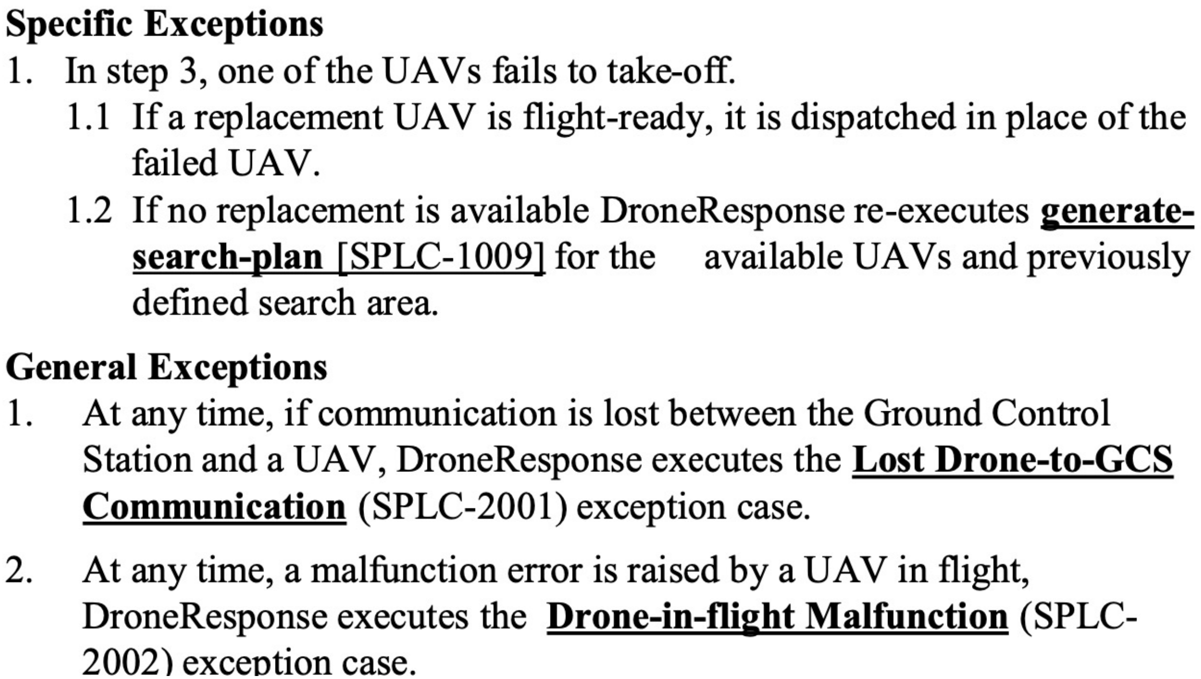
The Drone Commander activates the delivery.

Main Success Scenario

1. Emergency responders **plan_area_search** [SPLC-1001]
2. The DroneResponse commander issues a command to start the mission.
3. The UAV(s) **takeoff** [SPLC-1007]
4. The UAVs **perform_search** [SPLC-1002]
5. The UAV **requests_victim_confirmation** [SPLC-1005] from the human operator.
6. The UAV receives confirmation from the human operator that the victim sighting is valid.
7. DroneResponse automatically sends the GPS coordinates to the mobile_rescue system.
8. The UAV switches to **flotation_device_delivery** [SPLC-1006] mode.
9. Human responders reach the victim's location and execute a rescue.
10. The Drone Commander **ends_mission** [SPLC-1007].

J. Cleland-Huang, A. Agrawal, M. N. Al Islam, E. Tsai, M. van Speybroeck, M. Vierhauser: Requirements-driven configuration of emergency response missions with small aerial vehicles. SPLC (A) 2020: 26:1-26:12

Dronology Use Case:
Ice Search and Rescue (Exceptions)



Specific Exceptions

1. In step 3, one of the UAVs fails to take-off.
   1.1 If a replacement UAV is flight-ready, it is dispatched in place of the failed UAV.
   1.2 If no replacement is available DroneResponse re-executes **generate-search-plan** [SPLC-1009] for the available UAVs and previously defined search area.

General Exceptions

1. At any time, if communication is lost between the Ground Control Station and a UAV, DroneResponse executes the **Lost Drone-to-GCS Communication** (SPLC-2001) exception case.

2. At any time, a malfunction error is raised by a UAV in flight, DroneResponse executes the **Drone-in-flight Malfunction** (SPLC-2002) exception case.

J. Cleland-Huang, A. Agrawal, M. N. Al Islam, E. Tsai, M. van Speybroeck, M. Vierhauser: Requirements-driven configuration of emergency response missions with small aerial vehicles. SPLC (A) 2020: 26:1-26:12

Observation and Scenarios

- Observe the stakeholders in their work environment
- Try to understand what they do and why they do it to really understand their needs
- Ask questions in quieter moments
- Use audio or video recording as support

Observation and Structured Interviews

Capturing requirements directly in the work context

Maiden, N., Ncube, C., Kamali, S., Seyff, N., Grünbacher, P.: Exploring Scenario Forms and Ways of Use to Discover Requirements on Airports that Minimize Environmental Impact". 15th IEEE International Requirements Engineering Conference, New Delhi, October 15-19, 2007

Scenario Walkthrough at Belfast City Airport

What-if questions ease finding new requirements

1: The in-bound pilot safely parks the aircraft on-stand

3: The in-bound pilot switches off the engines

5: The BEST system records information about throughout the following activities

7: The ramp staff plug-in aircraft to ground power

9: The ramp staff insert ‘chocks’

11: The ramp staff insert steps for passengers

13: The ramp staff open aircraft hold doors and offload bags and cargo

15: The in-bound cabin crew prepare to deplane the aircraft

17: All passengers leave the aircraft

19: The customer service agent assist special needs passengers

21: The customer service agent directs arriving passengers to the arrival

23: The dispatcher co-ordinates aircraft servicing

25: The cleaners clean the aircraft

27: The caters cater for the aircraft

29: The fuelers re-fuel the aircraft

…

69: The outbound cabin crew closes aircraft doors

71: The check-in system relays passenger details to the destination airpo

73: The outbound pilot calls ATCO to confirm flight plan details, slot time and airfield weather

75: The outbound pilot calls ATCO to request engine start-up and push back

What if Pilot is unavailable or malfunctioning during this action?

What if this event does not occur in this scenario?

What if this event occurs earlier or later in time than expected?

What if this event occurs more or less frequently than expected?

What if a different event occurs instead of this event in the scenario?

What if this event repeats at an earlier or later time in this scenario?

What if Pilot has some unusual physical characteristics that affect his/her behaviour during this action?

What if Pilot is physically unable to undertake this action?

What if Pilot has some unusual physical characteristics that affect his/her behaviour during this action?

What if Pilot has some unusual physical characteristics that affect his/her behaviour during this action?

What if something unusual occurs in Pilot environment during this action?

Requirements Validation and Scenarios

- IKIWISI (I'll Know It When I See It)
- Prototypes demonstrate ideas but are not the final product
- Seeing a working model helps to understand what we want to build and how it could be improved
- Prototypes are frequently used for sketching user interfaces ("wireframes")

Source: Ian Alexander, Writing Better Requirements

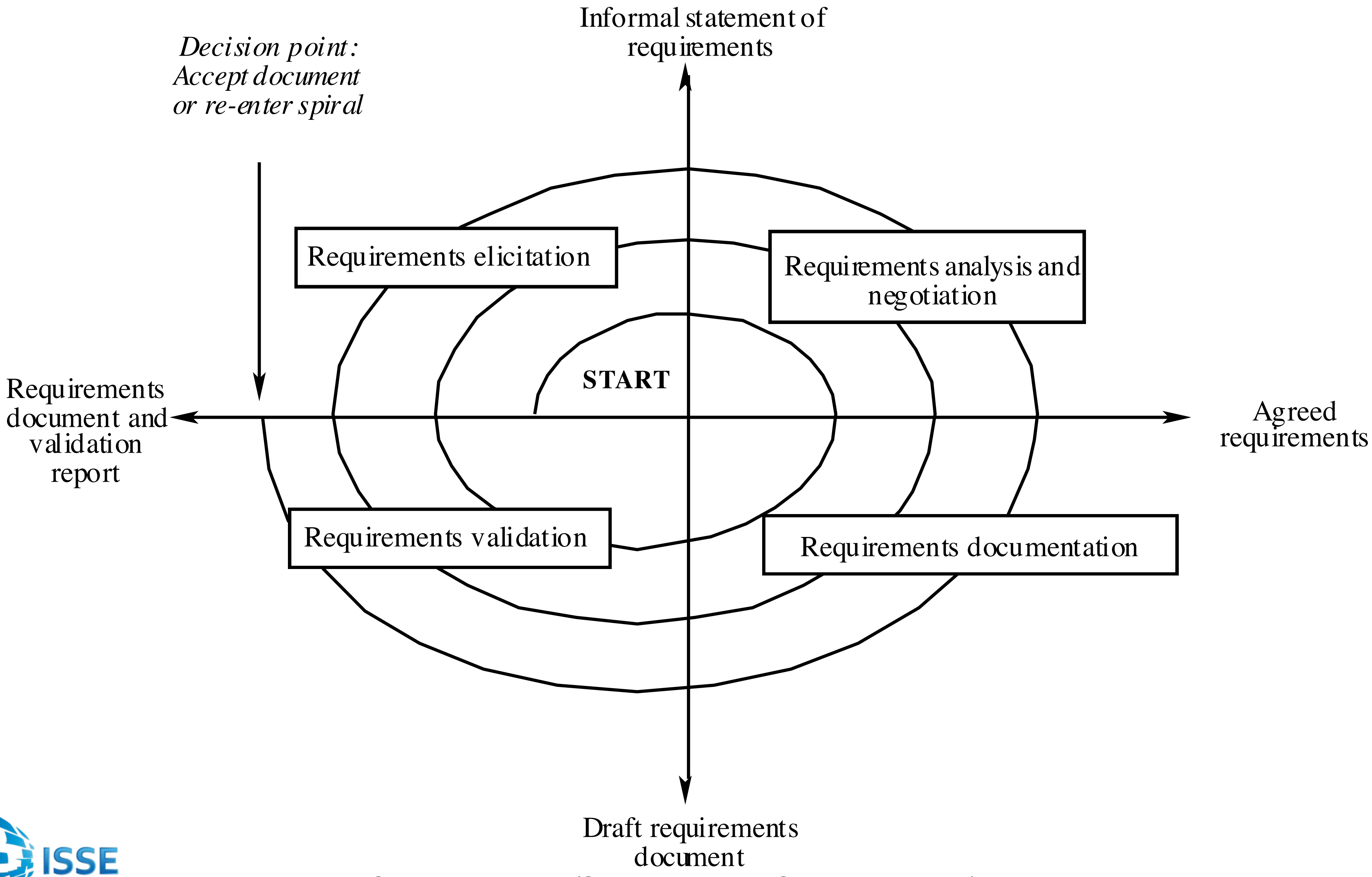
Documenting Software Requirements

- Unstructured prose
- User stories
- Use cases
- Structured prose, e.g., IEEE Std. 830
- UML models (e.g., state charts, activity diagrams)
- Feature models (in product lines)
- Goal modelling (KAOS, i*)
- Formal approaches (e.g., Z)
- (Test cases)
- (User manuals)
- …

Purpose of a Requirements Specification

- **Basis for agreement** between customers and suppliers on what the software product is to do
- Reduction of development **effort**
- Basis for **estimating** costs and schedules
- Baseline for **validation and verification**
- Easier **transfer of software product** to new users or machines
- Basis for **enhancement**

Requirements Engineering Process


Informal statement of requirements
Decision point: Accept document or re-enter spiral
Requirements elicitation
Requirements analysis and negotiation
START
Requirements document and validation report
Agreed requirements
Requirements validation
Requirements documentation
Draft requirements document

Spiral RE Process (G. Kotonya and I. Sommerville 1998)