

**BA** - BOOTCAMP



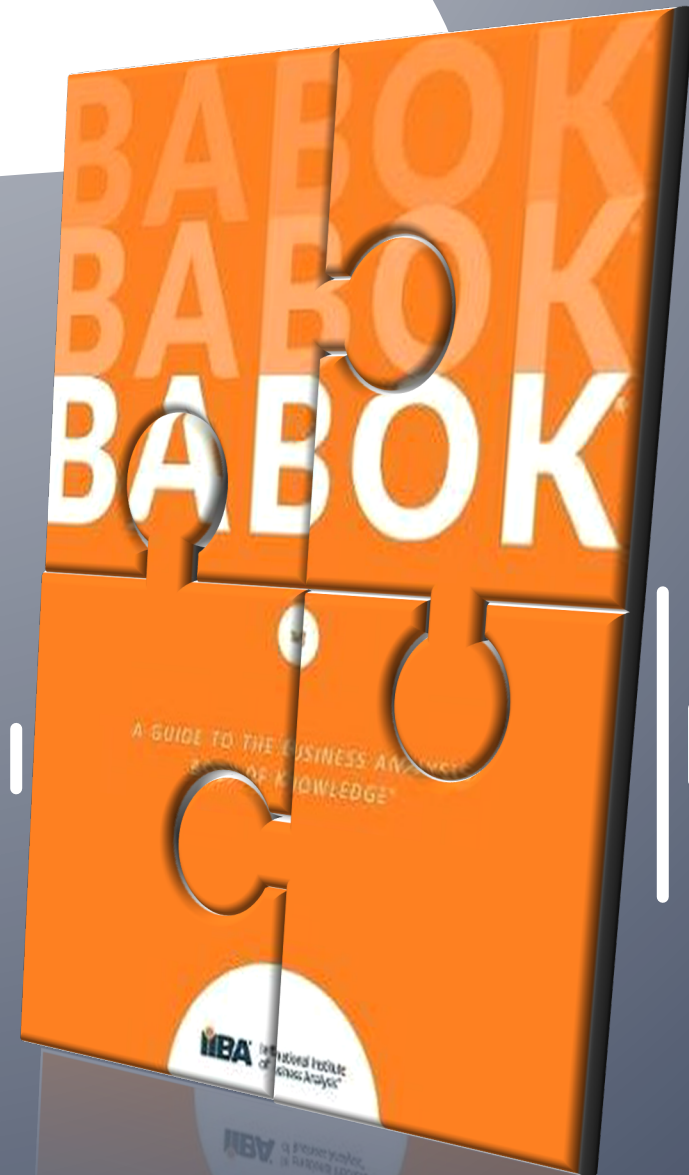
# THE BABOK® UNTANGLED SERIES

EPISODE 7

**REQUIREMENTS LIFE CYCLE  
MANAGEMENT  
(Chapter 5)**












Subscribe



**BA** BOOTCAMP



# THE BABOK® UNTANGLED SERIES

-  **EPISODE 01** Introduction to BABOK® and the Key Concepts
-  **EPISODE 02** Business Analysis Perspectives
-  **EPISODE 03** Strategy Analysis (incl Techniques)
-  **EPISODE 04** Business Analysis Planning and Monitoring (incl Techniques)
-  **EPISODE 05** Elicitation & Collaboration (incl Techniques)
-  **EPISODE 06** Requirements Analysis & Design Definition (incl Techniques)
-  **EPISODE 07** Requirements Life Cycle Management (incl Techniques)
-  **EPISODE 08** Solution Evaluation (incl Techniques)
-  **EPISODE 09** Business Analysis Competencies

# CONTEXT OF TODAY

## BABOK®

### Key Concepts



Foundation of BABOK and the conceptual framework for business analysis BACCM.

### Knowledge Areas

Knowledge areas represent areas of specific business analysis expertise that encompass several tasks.

### Underlying Competencies

Knowledge, skills, behaviours, characteristics, and personal qualities that help perform the role of the business analyst.

### Techniques

Techniques provide additional information on ways that a task may be performed.

### Perspectives



Perspectives provide focus to tasks and techniques specific to the context of the initiative

1

### Business Analysis Planning & Monitoring



Task...

Task...

A task is a discrete piece of work that may be performed as part of business analysis.

2

### Elicitation & Collaboration



3

### Requirements Life Cycle Management

4

### Strategy Analysis



5

### Requirements Analysis & Design Definition



6

### Solution Evaluation

Purpose

.. Why to perform

Description

.. What to perform

Inputs

.. Prerequisites

Elements

.. How to perform

Guidelines / Tools

.. Undertake action

Techniques

.. Using to perform

Stakeholders

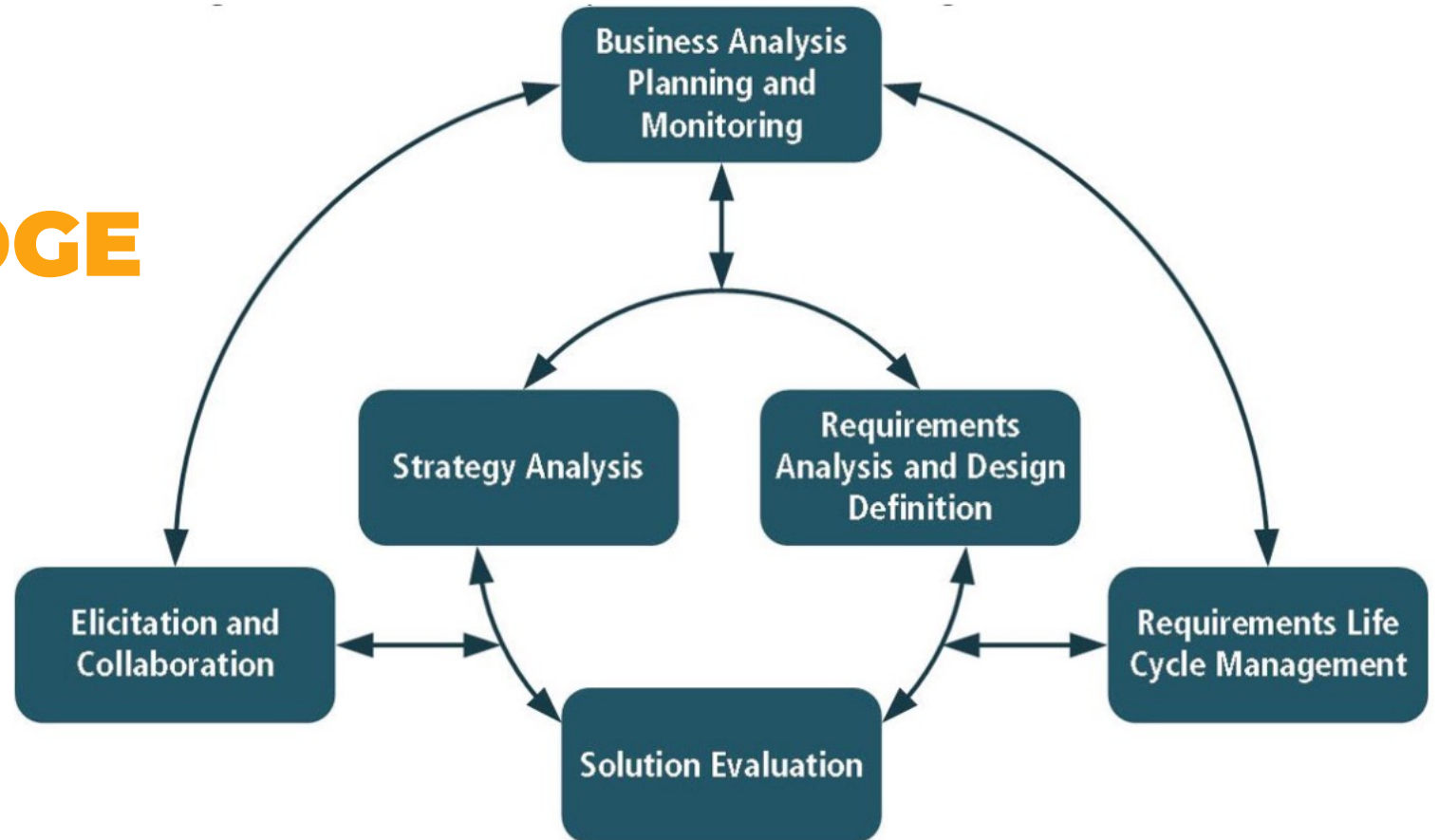
.. People involved

Outputs

.. The result

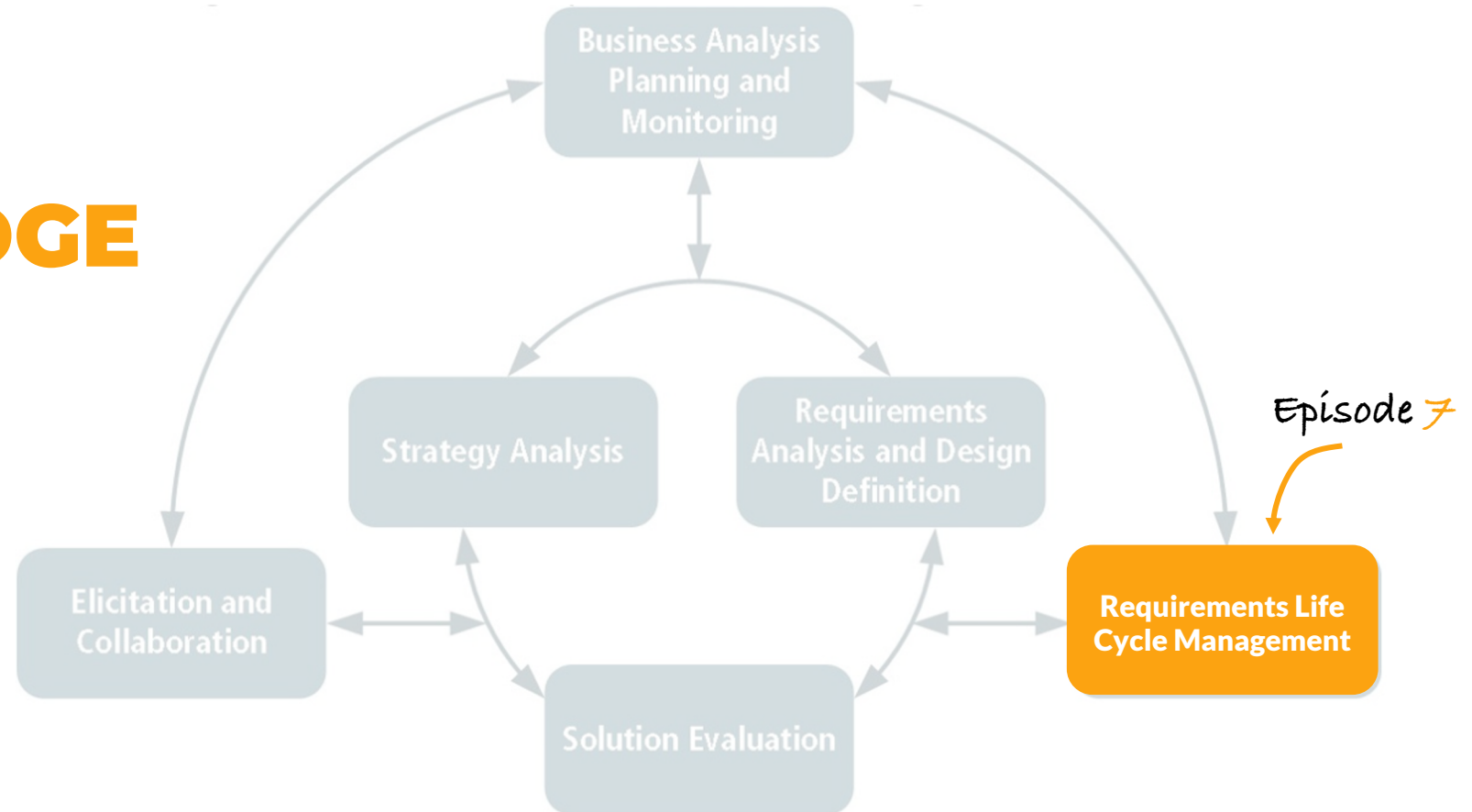


# BABOK KNOWLEDGE AREAS





# BABOK KNOWLEDGE AREAS



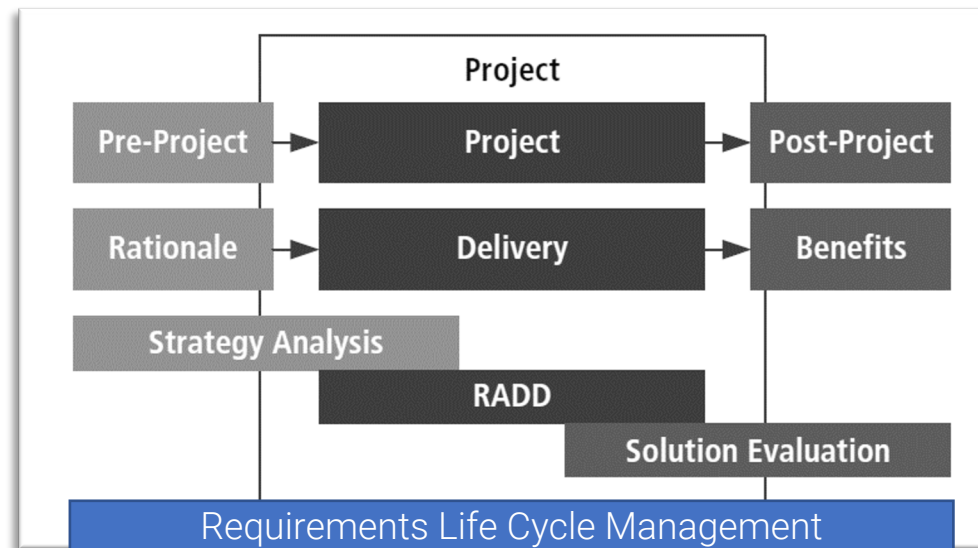


# REQUIREMENTS LIFE CYCLE MANAGEMENT

- 5.1 Trace requirements
- 5.2 Maintain requirements
- 5.3 Prioritize requirements
- 5.4 Assess Requirements Changes
- 5.5 Approve Requirements

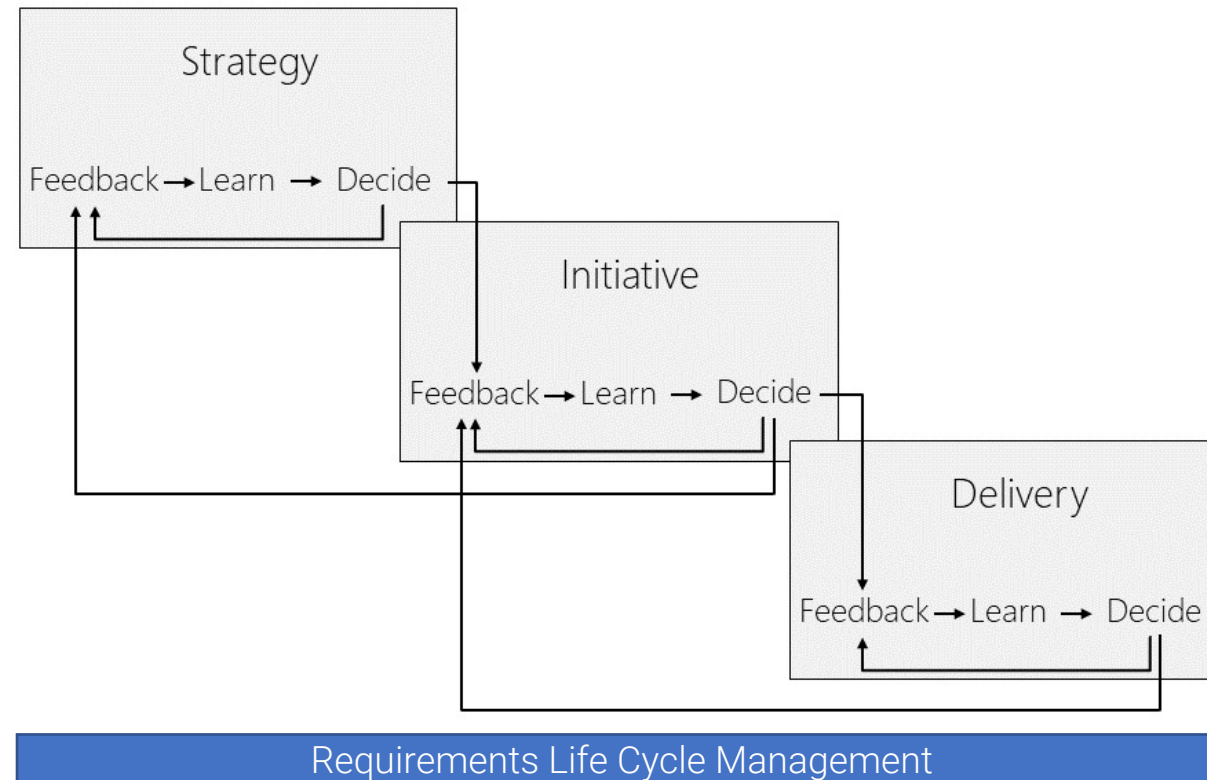


# RLCM in Traditional projects





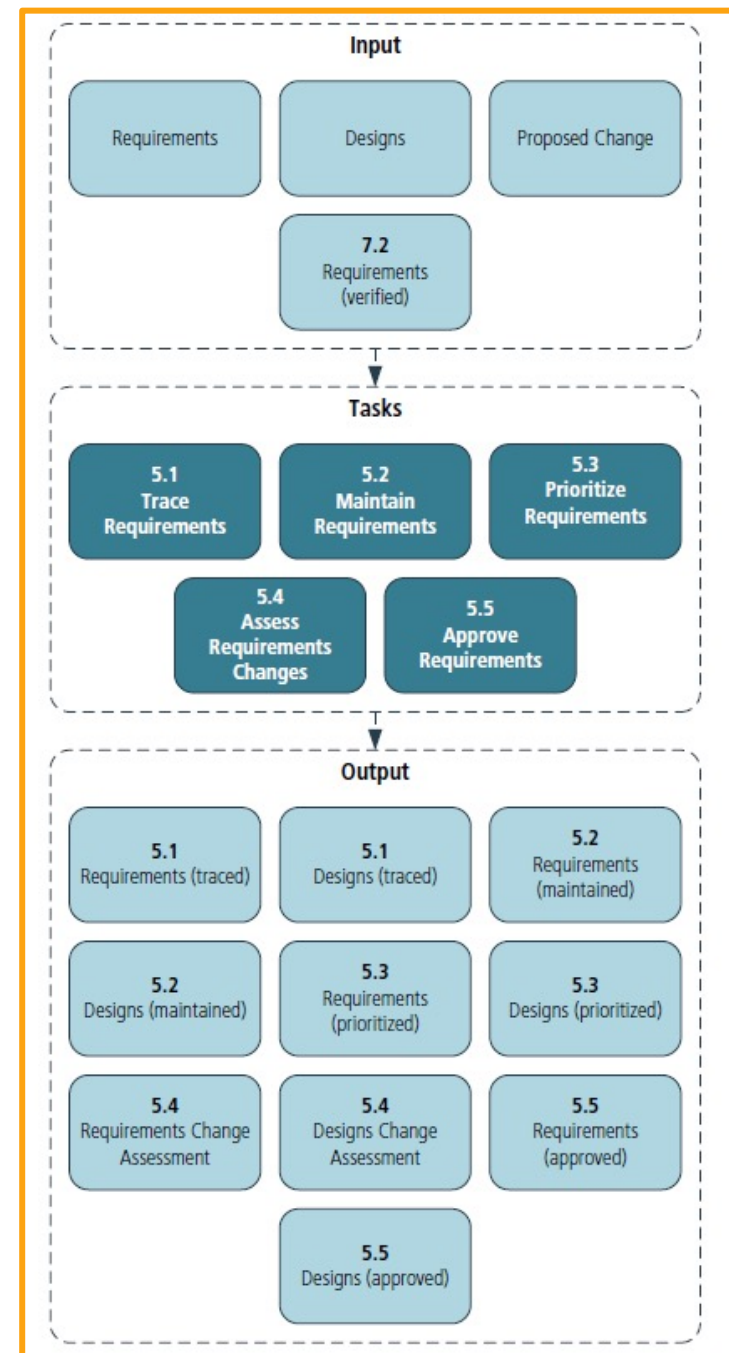
# RLCM in Agile







# There's no fixed order to RLCM!



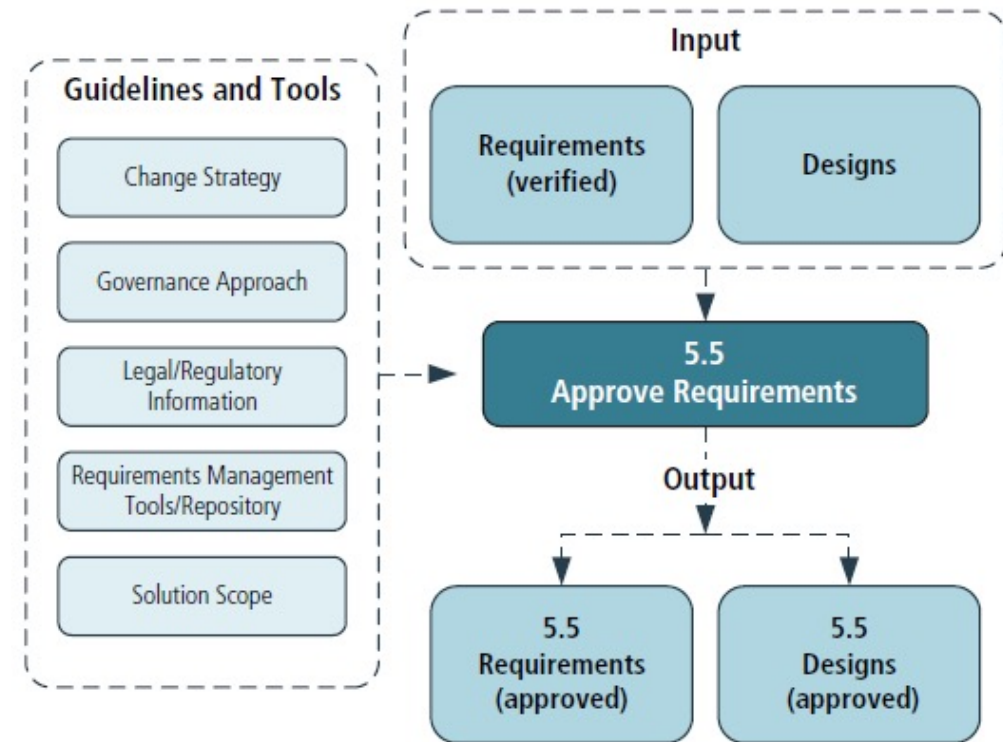
## 5.5

**APPROVE  
REQUIREMENTS**

To obtain agreement on and approval of requirements and designs for business analysis work to continue and/or solution construction to proceed.

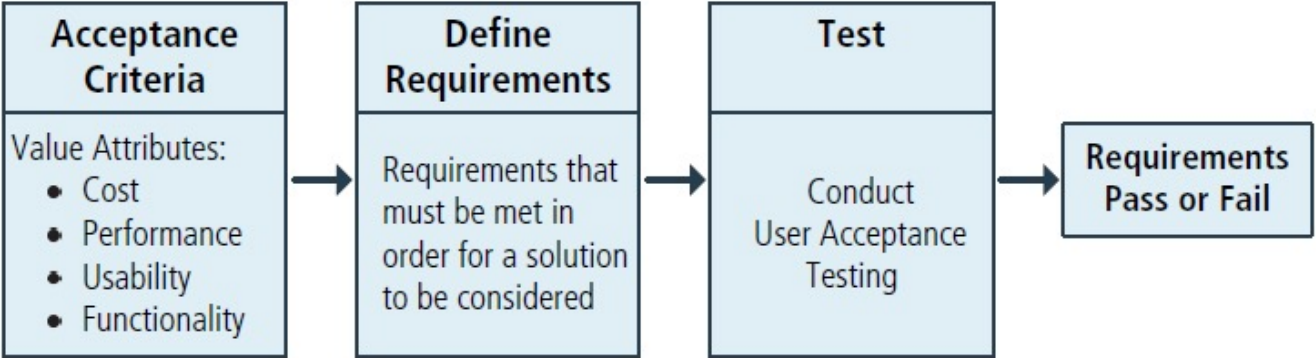
**Elements:**

- Clarity with regard to stakeholder participation
- Reaching consensus
- Conflict and issue management
- Establishing and communicating approvals



# Acceptance Criteria

One Solution



As a customer  
I need to be able to supercharge my  
vehicle using a credit card  
So that I can continue my journey

Acceptance criteria associated with this user story are:

- testing with a Visa card;
- testing with a MasterCard;
- testing with an expired Visa card.

# Consensus Workshop

Aligning all relevant stakeholders plus resolving requirements conflicts.

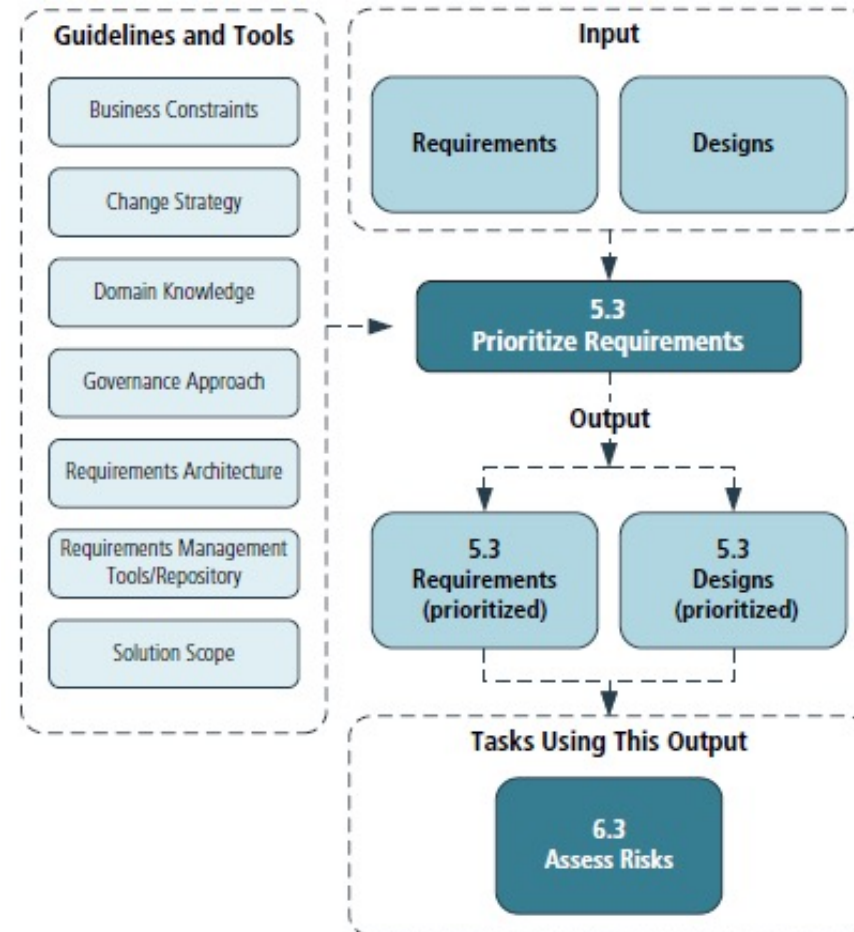
## Techniques:

1. Reach agreement → all involved agree
2. Compromise → give and take
3. Voting → Most votes count
4. Defining variants → implementing both (conflicting) requirements. Check if both are feasible by means of parameterization
5. Overruling → boss decides (only do it if all others fail). In Scrum the Product Owner has final responsibility

5.3

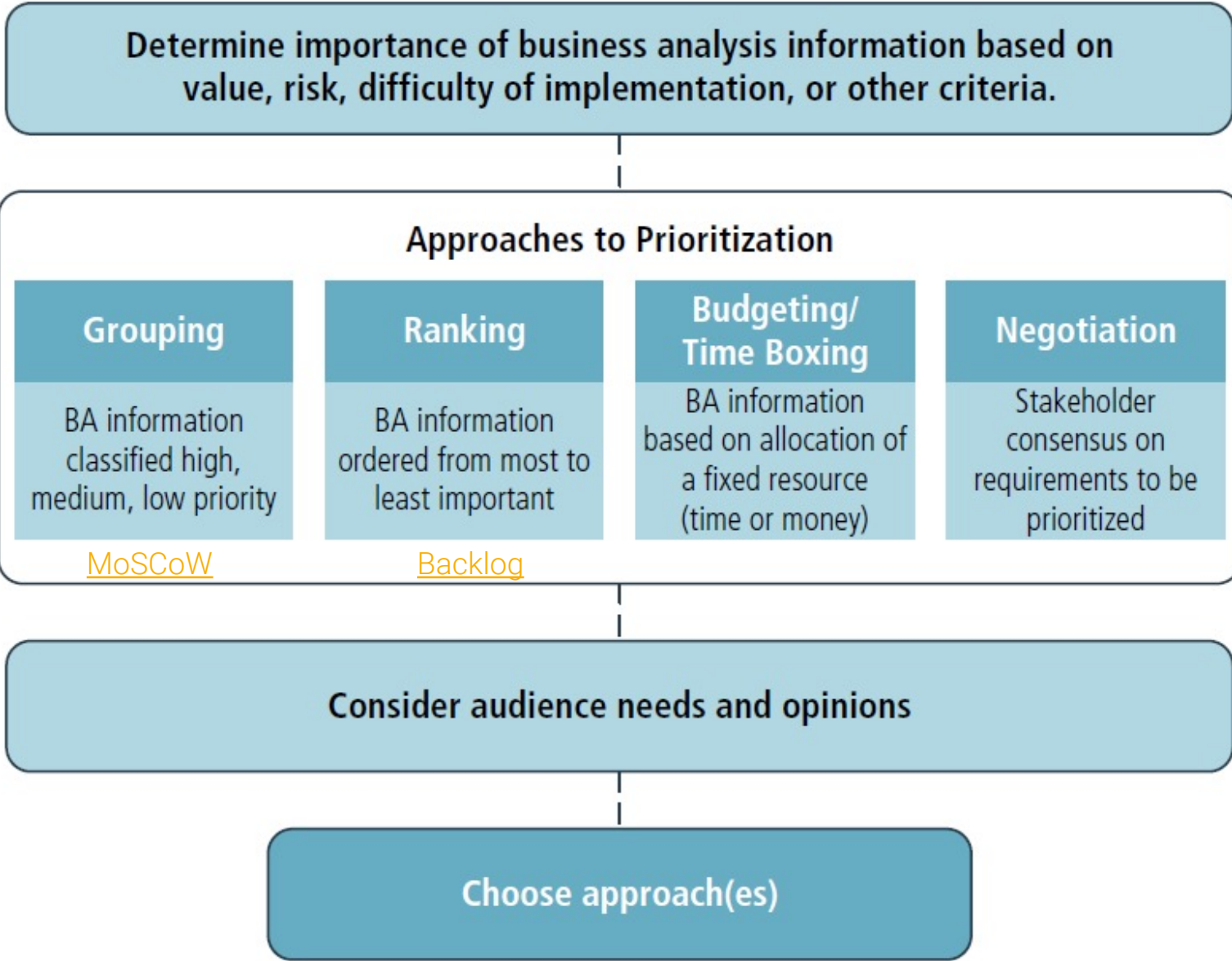
# PRIORITIZE REQUIREMENTS

To rank requirements in the order of relative importance.



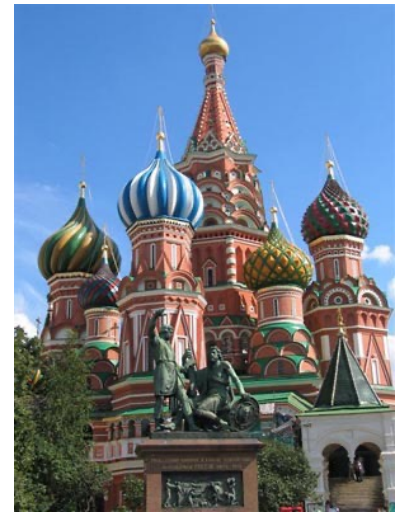
TECHNIQUE

Prioritization



# MoSCoW (Grouping)

- Must-have: Highest priority. If the product doesn't have this requirement, the initiative will fail.
- Should-have: High priority. This is an important requirement for the product.
- Could-have: Middle priority. Nice to have.
- Won't-have: Low priority. Will not be implemented (this time).





# Backlog Management (Ranking)

FCS board

Backlog

Q

QUICK FILTERS:

Only My Issues

Recently Updated

VERSIONS

All issues

> Oplevering 6.2.0.1

...

> 7.6

...

> 7.8

...

> 7.10

...

> 7.12

...

Issues without versions

EPICS

ZZP/EP/MPT

> VIII: Retourbericht FZ812

...

> i: Koppeling met FiZZa (Rijkszaak)

...

> iii: Versturen verantwoordingsgegevens ijm Matchen

...

> A: Ontvangen ziekenhuisdeclaraties

...

> B: Controleren tarieven ziekenhuisdeclaraties

...

> C: Aanbieden gecontroleerde ziekenhuisdeclaratie

...

Sprint 30

11 issues

ACTIVE

055

F

+1

...

14/Dec/20 2:44 PM • 04/Jan/21 2:44 PM

View linked pages

NPS-821 Als SIC wil ik, een automatische controle op de "verantwoordingsprocedure" die controleert "...

7.10

16: Automatische co...

NPS-710 Implementeren FZ301 t/m FZ304 als berichtstroom op de DSB

7.10

Functionele Release

NPS-701 Exporteren naar Qlikview

7.10

14: Exporteren naar ...

NPS-699 Overzichtelijk presenteren contractregels

7.10

10: Presenteren con...

NPS-546 Implementeren van tabel CL-Productgroep inclusief het vullen daarvan

7.10

1: Controles in FCS i...

5

NPS-842 Analyseren van IC1002 voor 571-berichten

7.10

IV: Controleren ZZP/...

NPS-871 Verduidelijkingstekst SC1 komt niet overeen met regel

Backlog

62 issues

Create Sprint

...

NPS-864 Aanpassing FCS-database mbt Ziekenhuisberichten

7.10

A: Ontvangen zieke...

5

NPS-865 Inlezen ZH308 bericht

7.10

A: Ontvangen zieke...

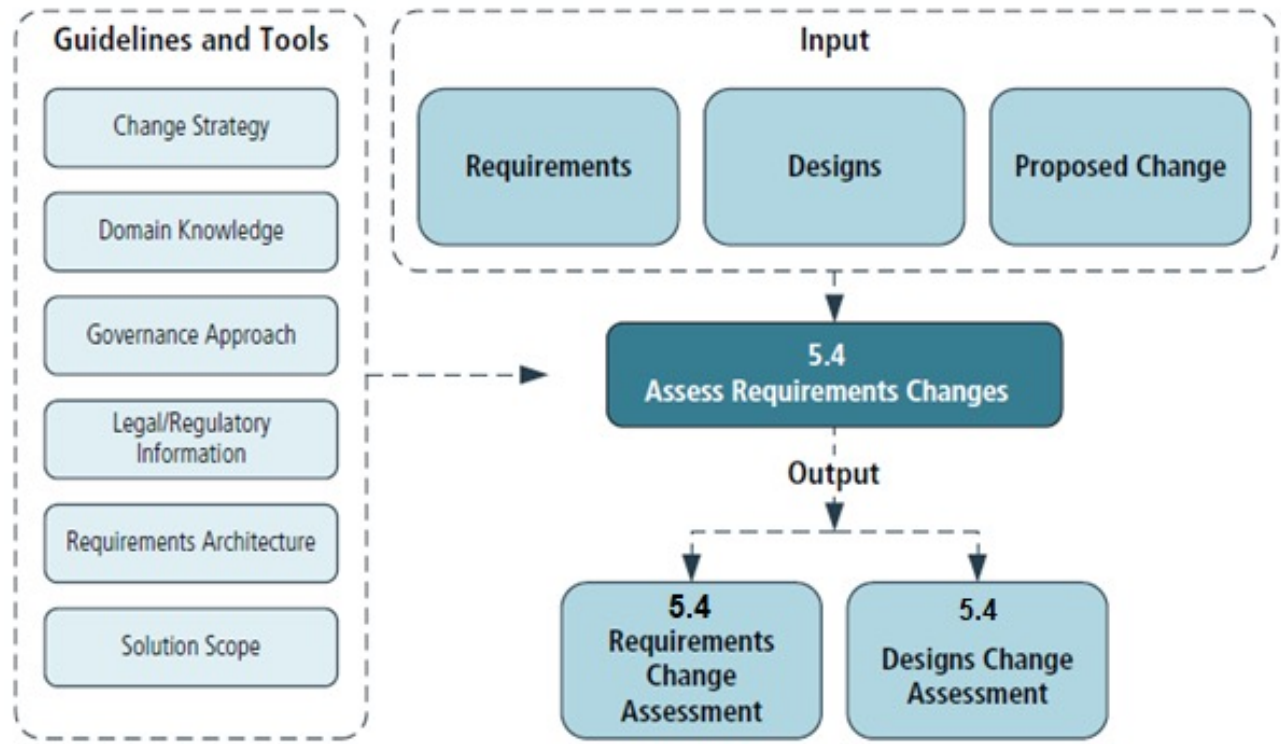
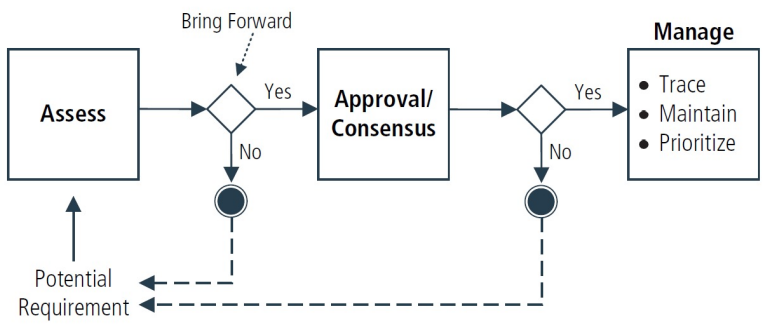
5



5.4

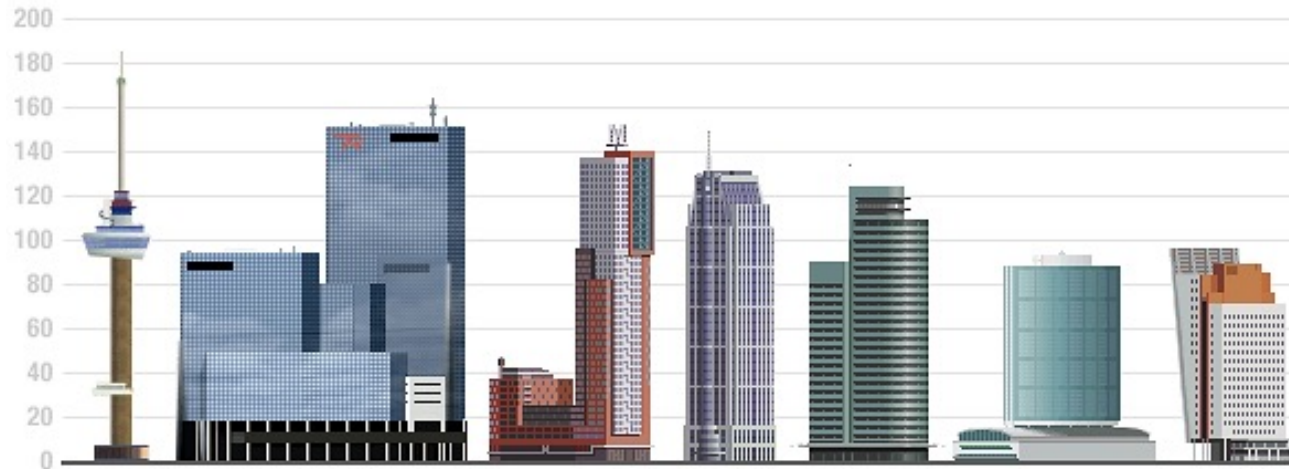
ASSESS  
REQUIREMENTS  
CHANGES

To evaluate the implications of proposed changes to requirements and designs.



# Relative Estimation

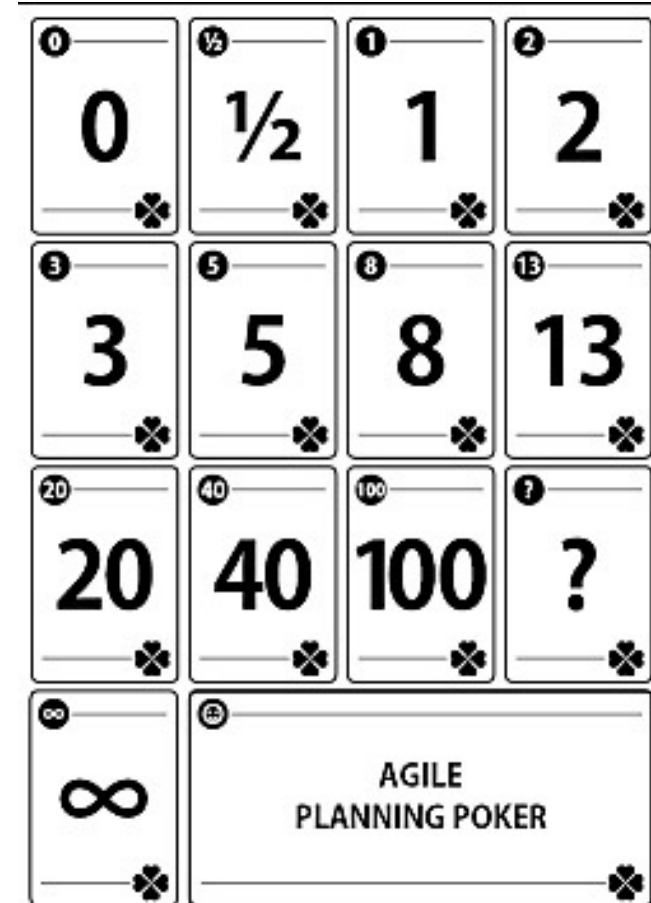
- Relative Estimation



- If we have 2 user stories (A and B), I can determine that A is smaller than B. Therefore, A will have a lower score than B.
- Techniques:
  - T-shirts (S-M-L-XL)
  - Planning Poker (Story Points)
  - WSJF

# Planning Poker

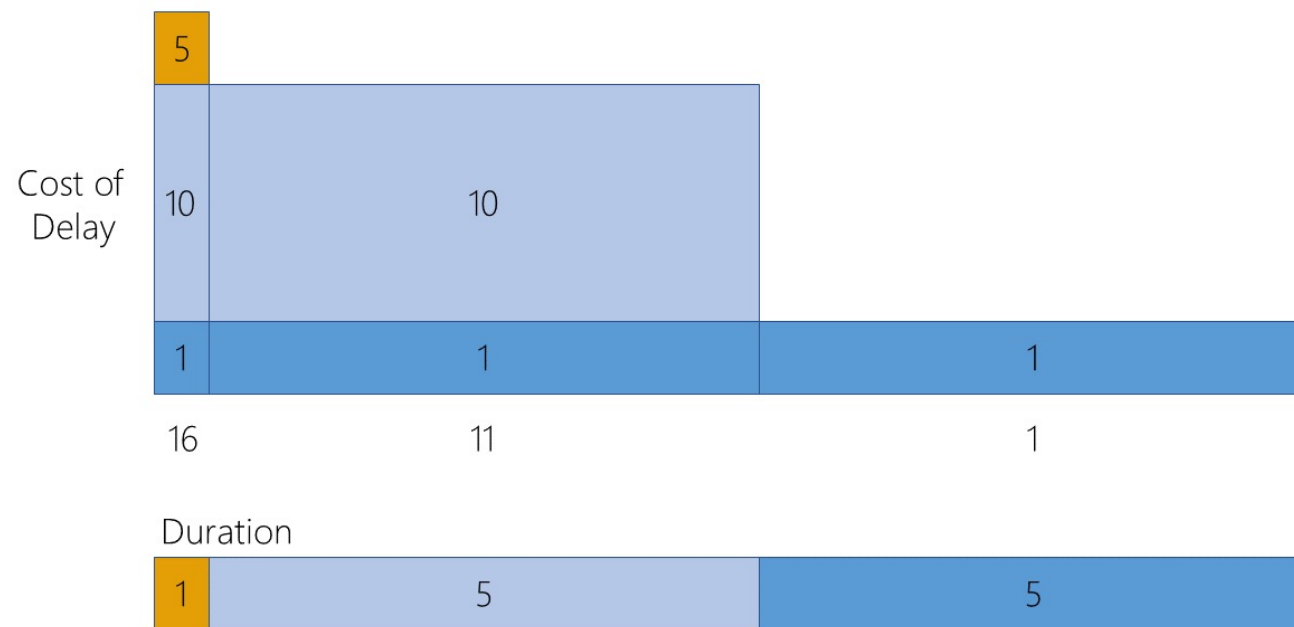
- Planning Poker based on row of Fibonacci.
- Several rounds
- In case of large differences:
  - Explanation lowest estimate
  - Explanation highest estimate
- Use reference user story
- Advantages:
  - Simpler than absolute
  - Team building
  - Discussion backlog items



# Weighted Shortest Job First (WSJF)

$$\text{WSJF} = \frac{\text{Cost of Delay}}{\text{Job Duration (Job Size)}}$$

Job	CoD	Duration	WSJF=CoD/Duration
A	10	5	2
B	5	1	5
C	1	5	0.2



Total Cost of Delay:  $1 \cdot 16 + 5 \cdot 11 + 5 \cdot 1 = 76$

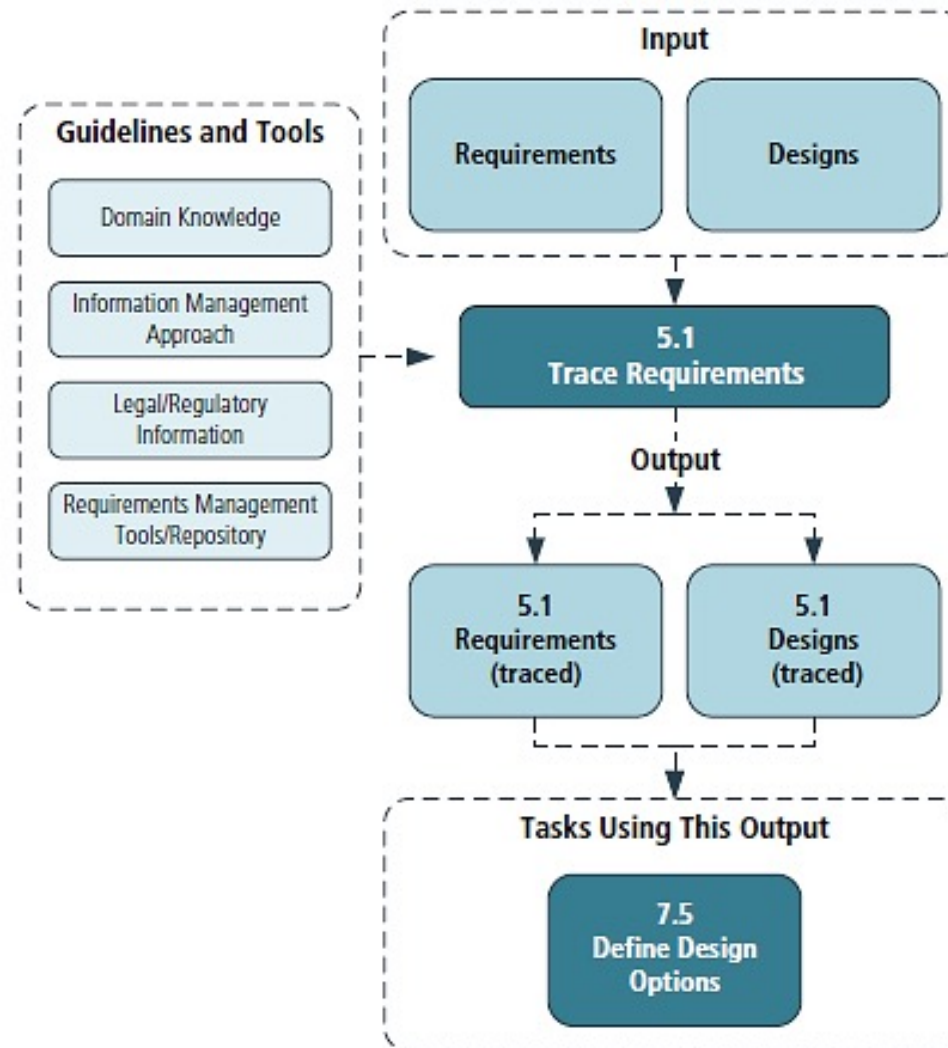
## 5.1

**TRACE  
REQUIREMENTS**

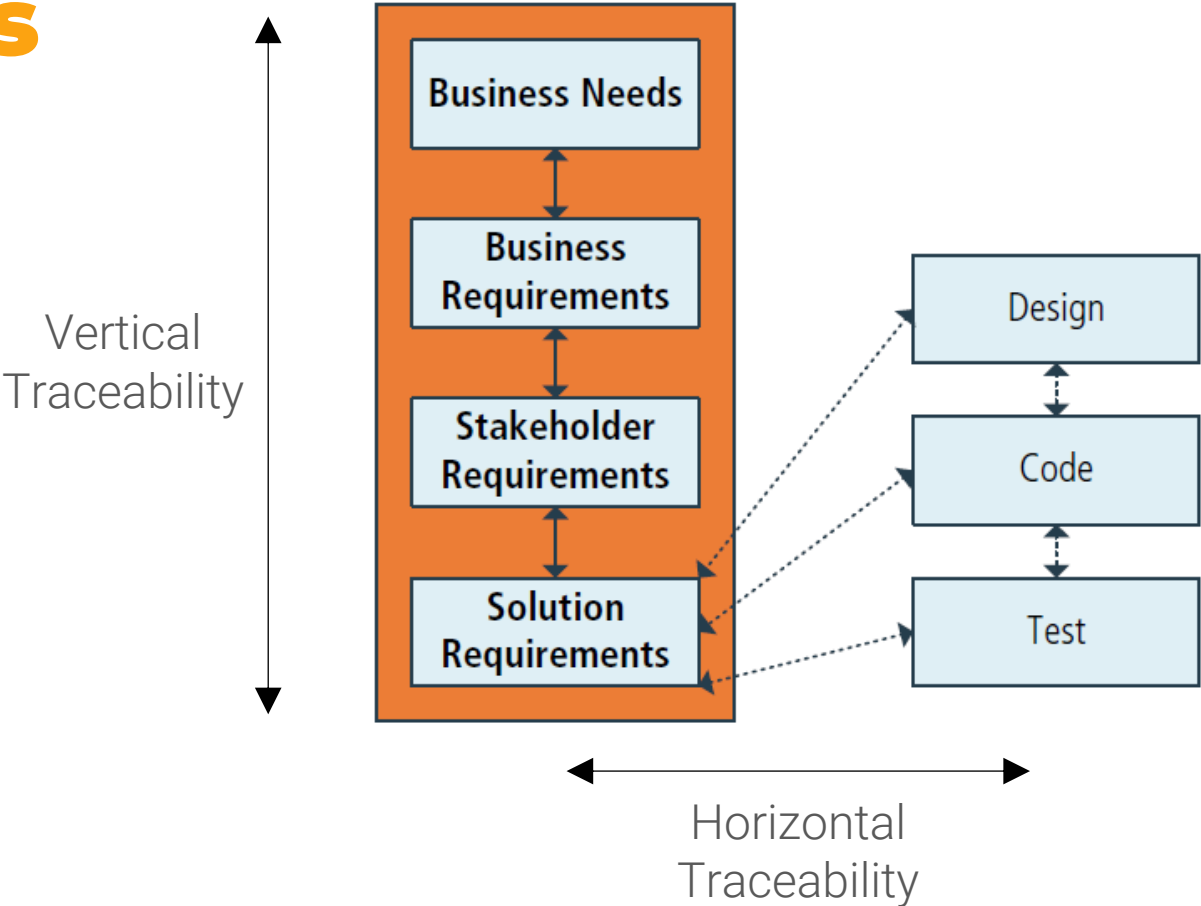
To ensure that requirements and designs at different levels are aligned to one another, and to manage the effects of change to one level on related requirements.

**Elements:**

- Level of Formality
- Relationships
- Traceability Repository



# Software Requirements Traceability



# Scope Modelling (Traceability Matrix)

Purpose: Used to define the nature of one or more limits or boundaries and place elements inside or outside those boundaries.

Target artifacts: requirement elaborations/documents

Initial artifacts:  
requirements

<div>Use Cases</div> <div>User Stories</div>	UC01	UC02	UC03	UC04	UC05	UC06
US01	X					
US02	X					
US03	X					
US04		X				
US05			X			
US06					X	
US07				X		
US08						X

TECHNIQUE

Scope Modelling (Hyperlinks)

Jira



Confluence



**Aju Sprint 3** 28 issues **ACTIVE** 60 31 23

25/Jun/19 9:48 AM • 16/Jul/19 9:48 AM View linked p:

- AJU-116 Opvragen actuele verblijfsgegevens uit de database AJu Core 3
- AJU-128 Vastleggen aanwezigheid elders (free format locatie) in DB AJu Core 5
- AJU-111 Vastleggen verblijf (instantie) in DB AJu Core 8**
- AJU-121 Stub t.b.v. KR P&I Stubs 3
- AJU-137 Wijzigen actuele verblijfsgegevens AJu Core 5
- AJU-141 JMS ondersteuning Stubs 20
- AJU-155 Wijziging verblijf bericht op de queue zetten AJu Core 8
- AJU-156 Aanmaken verblijf bericht op de queue zetten AJu Core 5
- AJU-158 Beëindigen verblijf in DB AJu Core 3
- AJU-157 Uitschrijven (verblijf) bericht op de queue zetten AJu Core 3
- AJU-198 aanmaken tabellen voor facades
- AJU-169 Aanmaken verblijf krijg could not extract ResultSet
- AJU-170 Hernoemen plaatsing naar verblijf AJu Core 5
- AJU-74 Informeren KIC AJu Core 8
- AJU-174 Optimaliseren tests AJu Core
- AJU-175 Validatieframework toevoegen AJu Core 5
- AJU-176 Aanmaken verblijf resulteert in een SQL fout

**Aju / AJU-111**

Vastleggen verblijf (instantie) in DB

Estimate: 8

**Issue Links**

is blocked by

- AJU-176 Aanma... TEST
- AJU-169 Aanma... READY FOR DE

is cloned by

- AJU-128 Vastle... IN PROGRESS
- AJU-158 Beëind... TO DO

relates to

- AJU-206 Aanma... READY FOR DE

Show 4 more links (1 links to, 3 mentione

**Description**

Als gebruiker wil ik mijn verblijf vastgelegd de AJU database.

Hieronder vallen:

- het registreren van het Verblijf plus Administratieve instantie o.b.v een inschrijfbericht: zie UC01
- het registreren van de Aanwezigheid elders o.b.v. Toevoegberichten: zie UC08

Backward traceability

Forward traceability

**3.8. Opvragen Justitiabele**

Zie UC12 Opvragen Justitiabele P&I → bepaal DJI-nr obv SKN, V\_NUMMER of BSN.

**3.9. Registreren verblijf & AI (Use case UC01a)**

Bron: **AJU-111 - Vastleggen verblijf (instantie) in DB IN PROGRESS**

Event (bericht): operatie: Maken (AJU-Core)

Stappen:

- Lees JUS\_ID uit Justitiabele-tabel obv DJI\_NUMMER,
- Maak objecten aan obv berichtvelden → zie onderstaande tabel.
- Schrijf gegevens weg

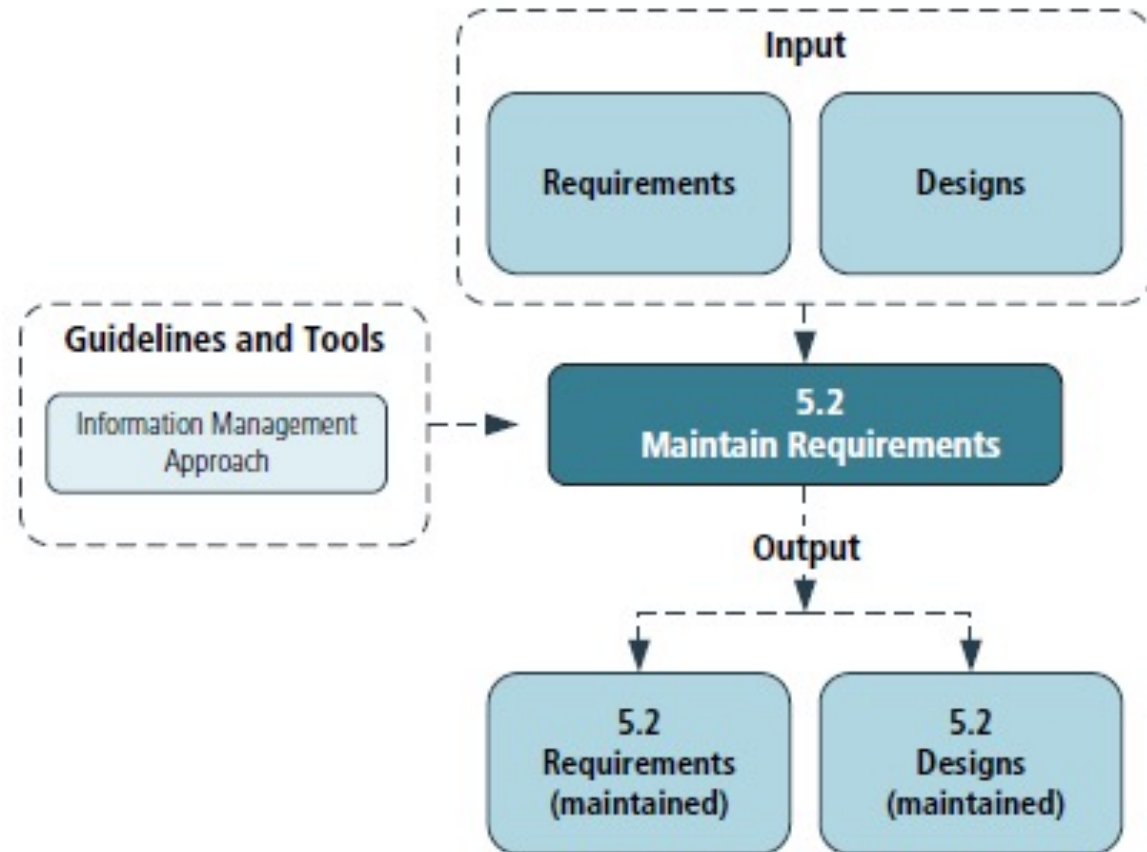
AJu-Tabel.veld	Waarde/Berichtveld
Administratieve_Instanties.Jus_id	JUS-IS
Administratieve_Instanties.Instantie_code	globalePlaatsing.verantwoordelijkeEenheid.identificatie

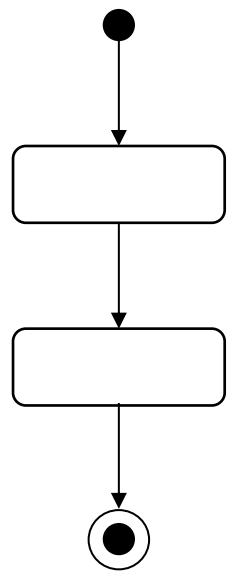


## 5.2

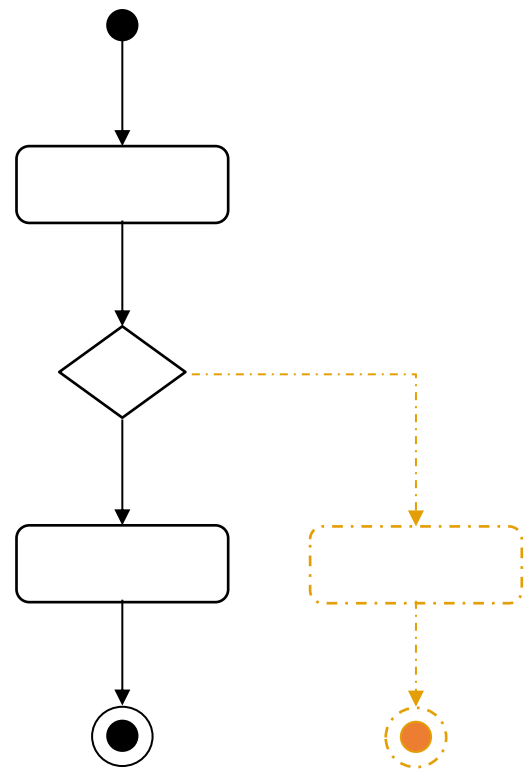
**MAINTAIN  
REQUIREMENTS**

To retain requirement accuracy and consistency throughout and beyond the change during the entire requirements life cycle, and to support reuse of requirements in other solutions.

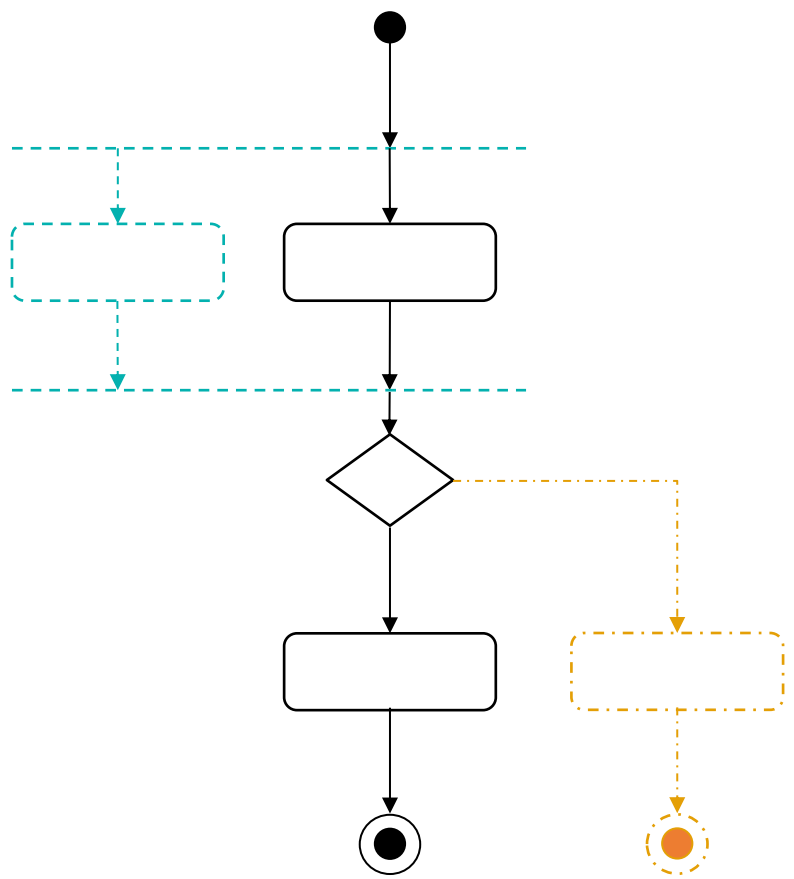




User story 1: Happy flow (Sprint 1)



User story 2:  
Alternative flow 1  
(Sprint 2)



User story 3  
New insight in Happy flow (Sprint 3)

Requirements and Designs change with time

## TECHNIQUE

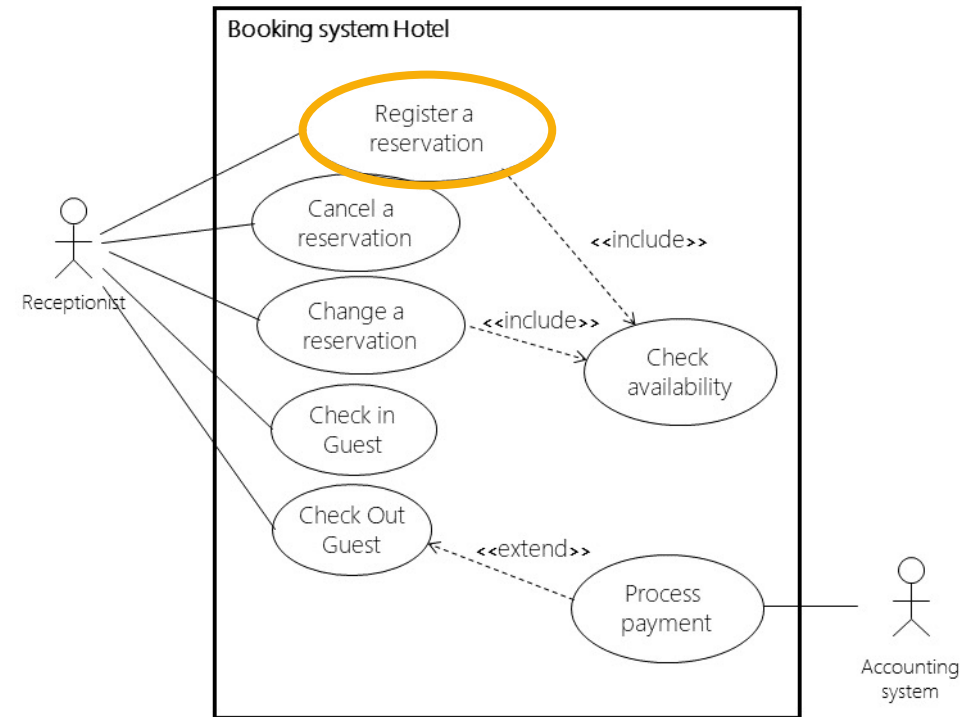
# Use Cases & Scenarios

### Basic Flow:

1. Receptionist activates the menu 'Register reservation'
2. Booking system shows the guest screen
3. Receptionist enters guest name, address and birth date.
4. Booking system registers the entered data and shows the screen with reservations.
5. Receptionist enters room type, date, amount of persons, .....
6. Booking system checks the room availability on the selected date (included use case Check availability)
7. Booking system registers the reservation (there is a room available).

### Alternatief scenario 1:

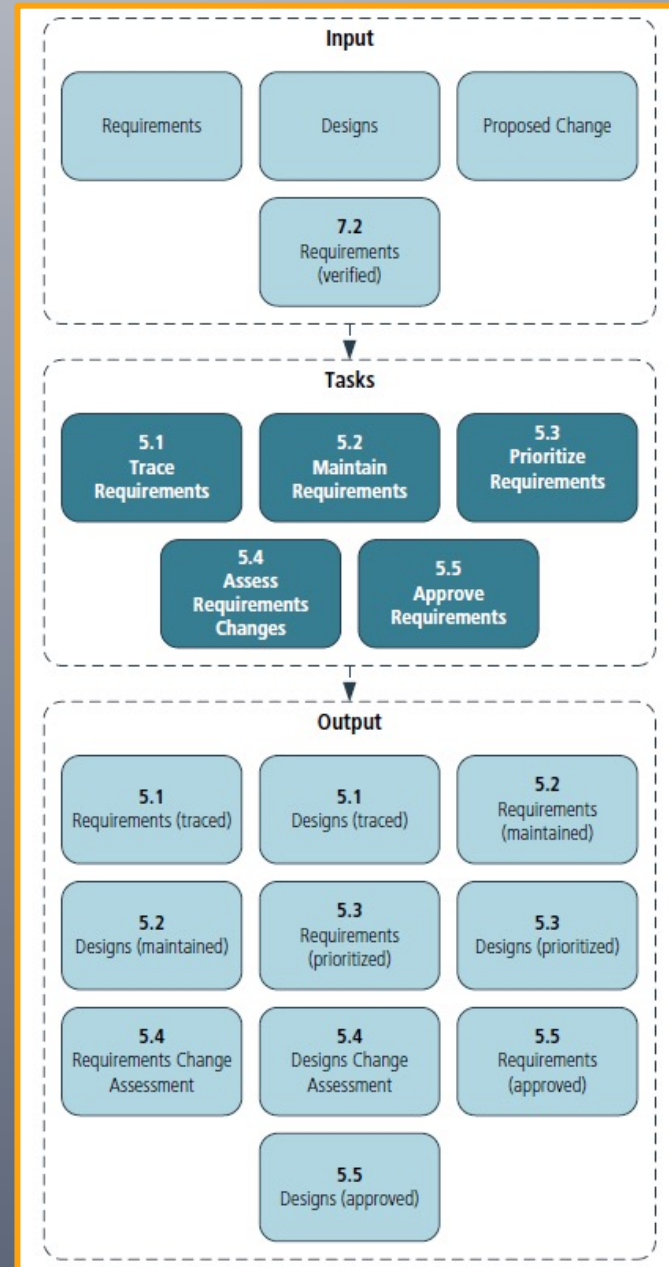
- 6a. There is no room available.
  - 6a1. Booking system gives an error message 'No room available on date'
  - 6a2. Booking system asks Receptionist for new date.
  - 6a3. Continue with nr. 5.



# SUMMARY

# SESSION

## 7



**BA** BOOTCAMP



**THANK YOU FOR  
YOUR ENGAGEMENT**

