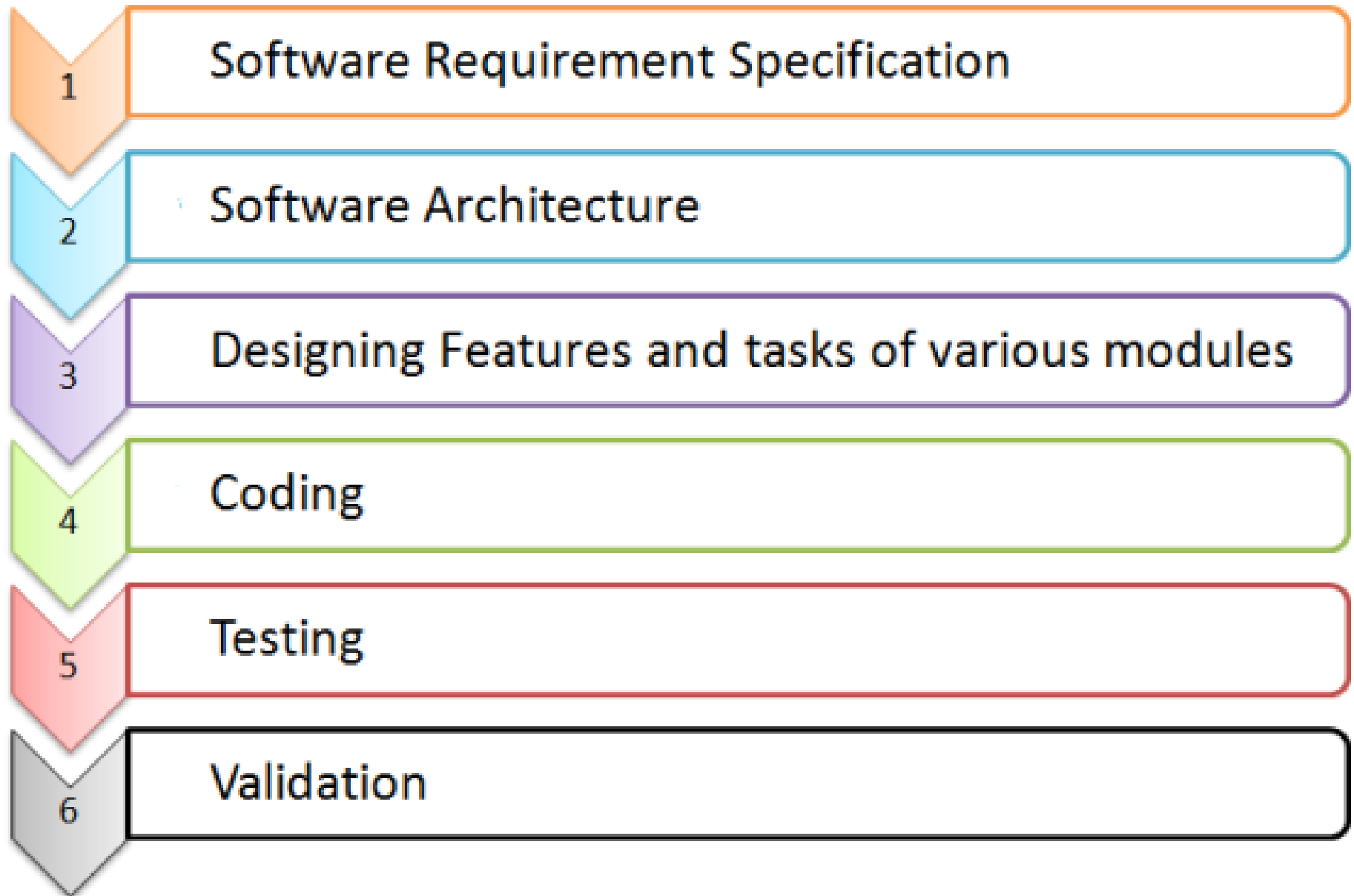


Task 2 : Project SRS Elicitation



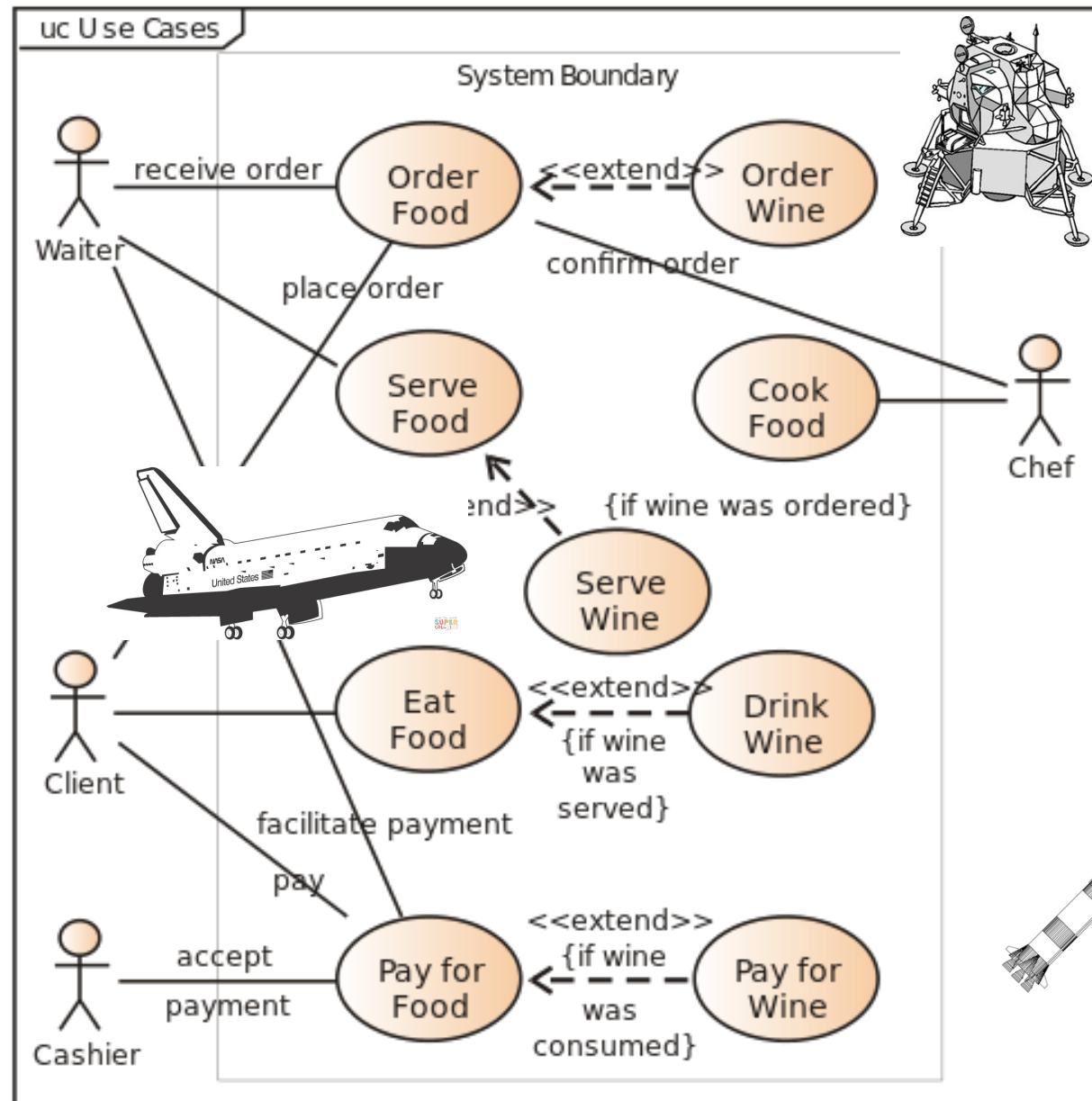
Task 2 : Project SRS Elicitation

- SE uses declarative approach(es) to define
 - what we want
 - what the constraints are
- Software Requirements Specification (SRS)
 - use cases / user stories
 - what tasks does the user do?
 - how does the user perform them?
 - why does the user perform them?
 - requirements
 - what we want / what is the form of a solution
 - what it should do (and not)
 - specifications
 - constraints, simplifications, etc.
 - what the constraints are / how it should do it (and not),
but very light on implementation-level details (unless important)

data
control
behavior

Task 2 : Project SRS Elicitation

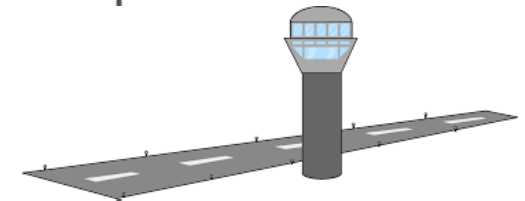
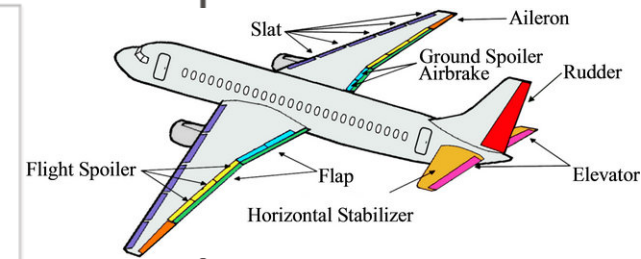
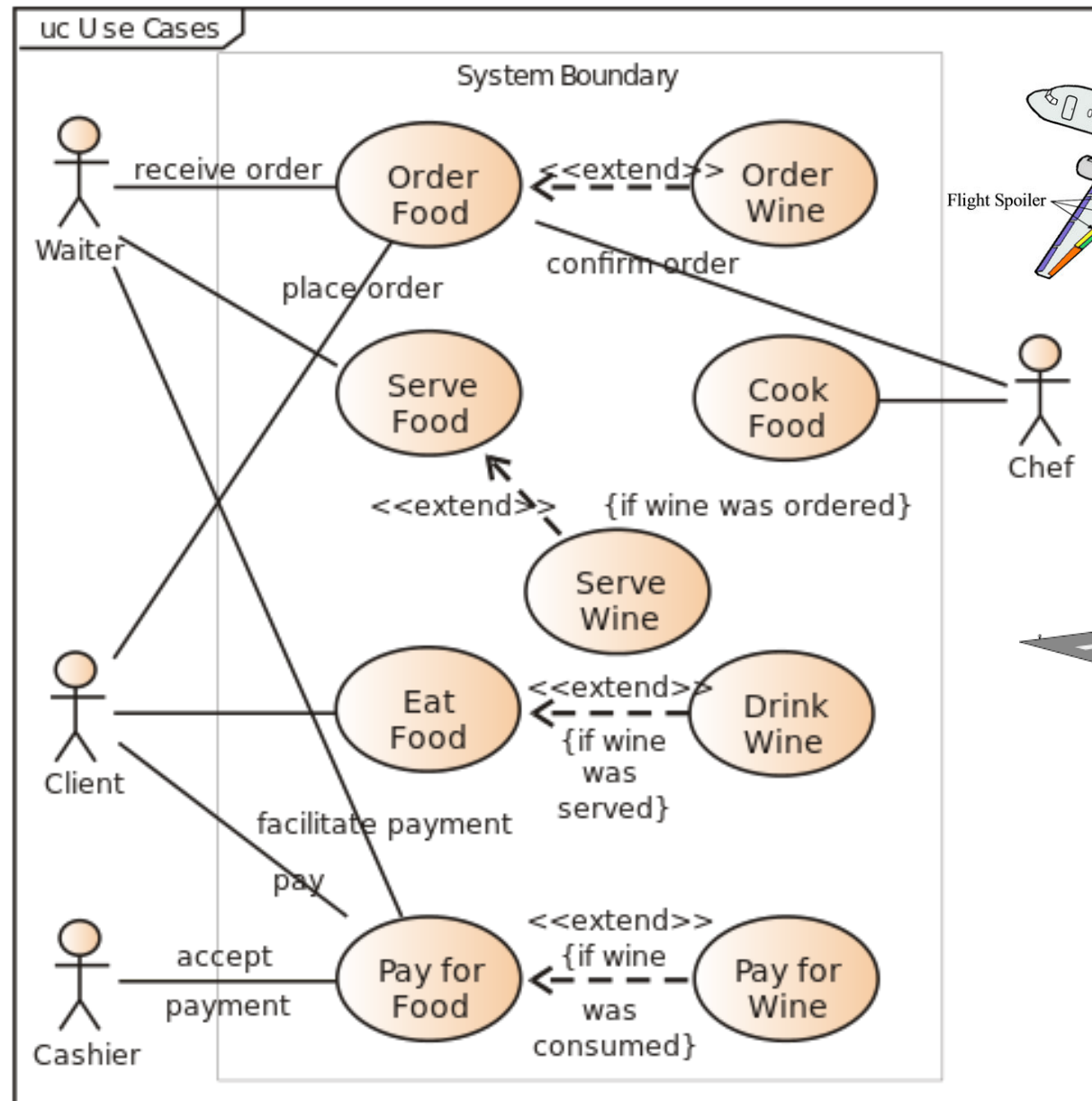
User story: maps to W⁵H to requirements to specifications to perform user tasks



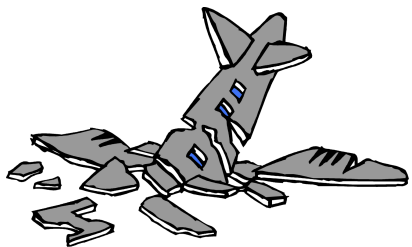
behavior

Task 2 : Project SRS Elicitation

User story: maps to W⁵H to requirements to specifications to perform user tasks



behavior



Task 2 : Project SRS Elicitation

User Story Template

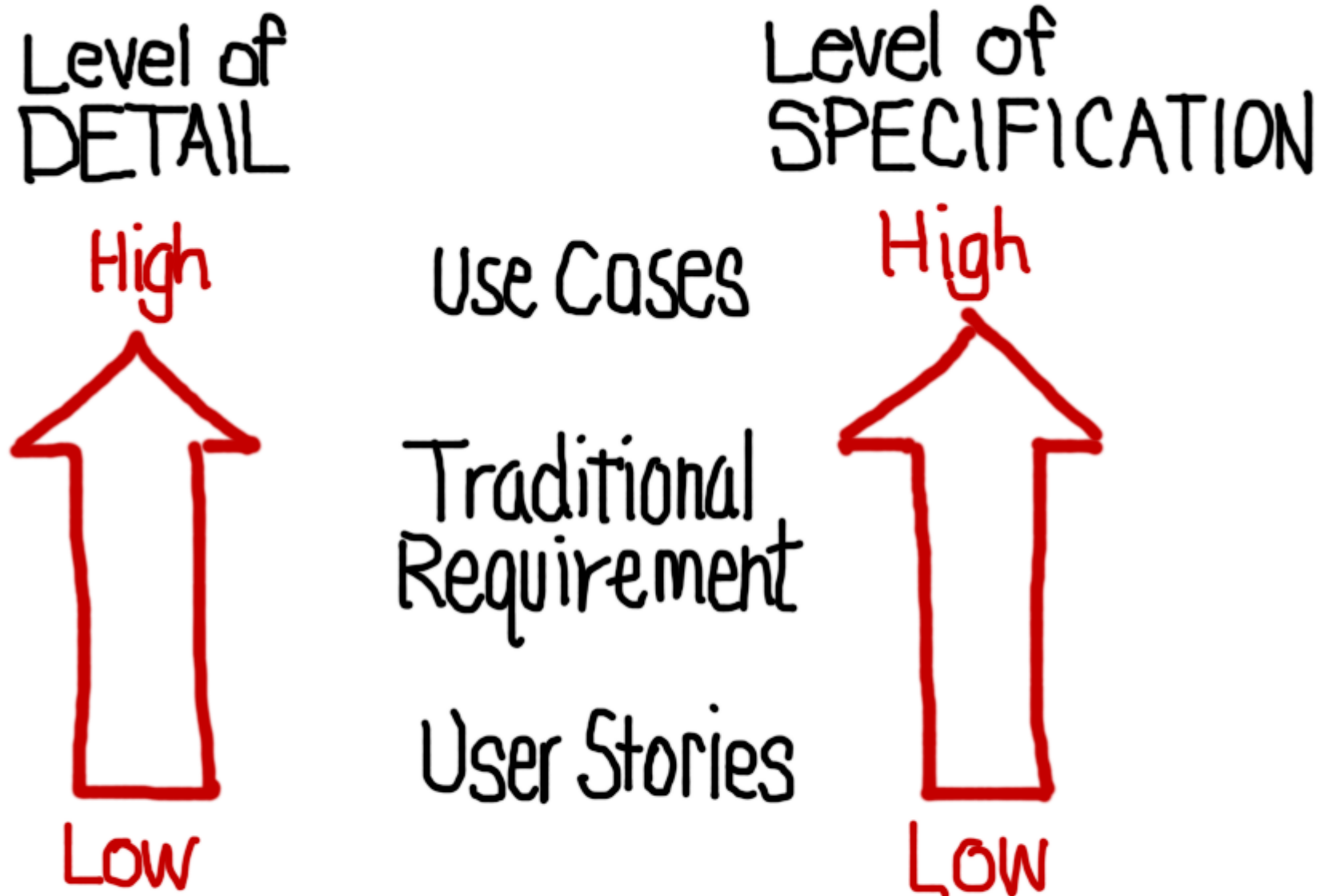
As a *[user role]*

I want to *[desired feature]*

So that *[value/benefit]*

behavior

Task 2 : Project SRS Elicitation



behavior

Task 2 : Project SRS Elicitation

The 3 C's

1. Card

Written on a card

As a user, I can login and gain access to the intranet, so that I can collaborate with all the organization.

2. Conversation

Details captured in conversations

What about expired accounts?

Can it remember my login?

3. Confirmation

Acceptance criteria **confirm** that the story is Done.

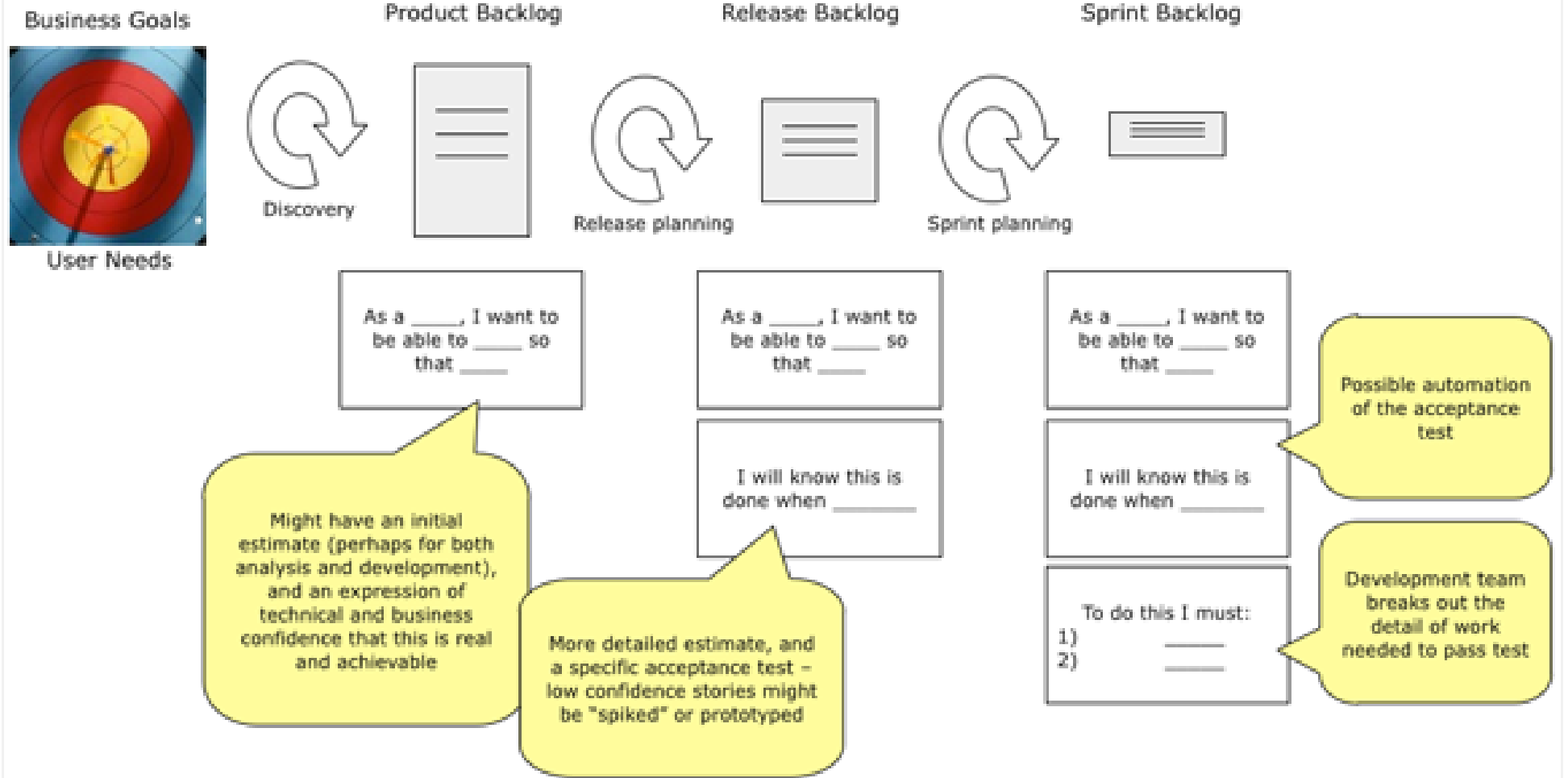
- Expired accounts fail
- Remember the login, not the password
- After 3 attempts the account is locked out for 24h (SOX compliance)

Source: XP Magazine 8/30/01, Ron Jeffries

behavior

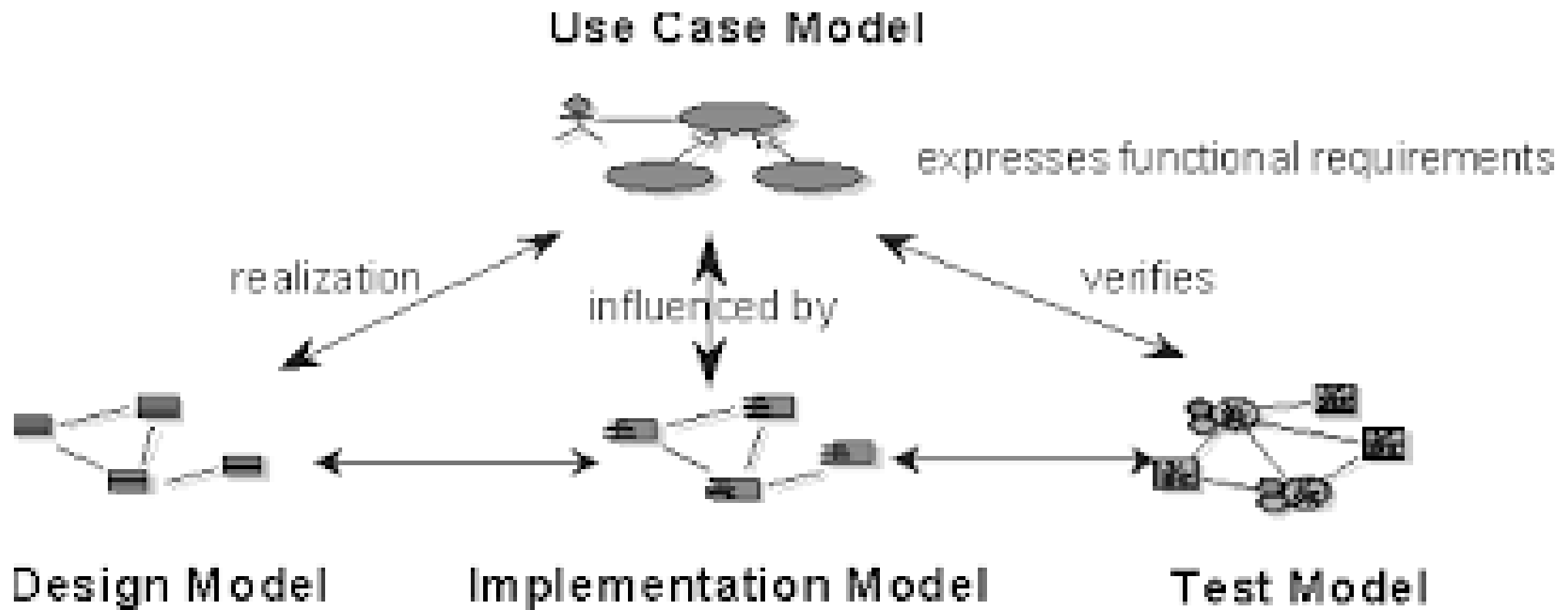
Task 2 : Project SRS Elicitation

User Story Lifecycle



behavior

Task 2 : Project SRS Elicitation



Task 2 : Project SRS Elicitation

- Given our scope from Task 1 and in lecture, **consider** the following elements (**E**):
 - I track building
 - II track usage (e.g., switching)
 - III train engineering: realistic motion
 - IV train engineering: multiple engines
 - V track/train engineering: control/safety
 - VI train building: coupling/uncoupling
 - VII train usage: staging
 - VIII train usage; e.g., delivering passengers

Task 2 : Project SRS Elicitation

- Part I: Generate representative user stories (U)
 - for each element, **narrate** in English how the user might create, configure, connect, and/or use it such that it generally performs one task in its (brief) entirety. For example:
 - **XVII** salad shooter
 - As a chef, I want to configure the salad shooter for different slice widths in order to vary the presentation of the salad.
 - **XVIII** salad
 - As a chef, I want to select ingredients and their quantities in order to vary the composition of the salad.
 - **XIX** ingredient
 - As a chef, I want to load the available ingredients into the salad shooter in advance in order to have them available before starting The Salad Experience instead of acquiring them piecemeal later, which is disruptive to a chef's train of thought.



Task 2 : Project SRS Elicitation

- Part II: Elicit questions (Q) from user stories
 - for each element ($\in \{I..V\}$) **enumerate** set of three most representative W⁵H questions, starting at 1; e.g., salad shooter (*note 8 > 3 here*):
 - XVII salad shooter
 - who
 1. who decides whether to make the salad by hand or with the appliance?
 - what:
 2. what is the mechanism to aim the salad at the receiving bowl?
 3. what is the process to activate the appliance?
 4. what is the maximum effective range for shooting cucumber slices?
 - when:
 5. when does the chef decide to activate the appliance?
 - where:
 6. where is the raw salad stored before processing?
 - why:
 7. why is the appliance effective for making a Caesar salad?
 - how:
 8. how does the appliance prevent the user from losing a finger?

Task 2 : Project SRS Elicitation

- for each of your questions ($\in \{\text{who, what, when, where, why, how}\}$)
 - **provide** representative requirements (**R**) as you think are necessary to address them
 - one per question
 - need only be reasonable and consistent, not necessarily 100% correct or optimal
 - formal procedures, terminology, and phraseology not required, but try
 - requirement states form of solution; it does not answer the question
 - **cross-reference** each requirement with its question number, starting with [a]
 - [a] The qualified chef must be in charge of the appliance at all times for safety. (1)
 - [b] The appliance needs a telescopic sight to see what the neighbors are cooking. (2)
 - [c] The appliance needs night-vision capability to see the neighbors at night. (2)
 - [d] The appliance needs a removable hopper for salad storage after dinner. (6)
 - [e] The appliance needs an exciting brand name to sell well. (7)
 - [f] The appliance must appeal to Stu Steiner so he will buy it. (8)

Task 2 : Project SRS Elicitation

- for each requirement
 - **provide** a form of the solution as a specification (**S**), starting with [**A**]
 - one per requirement
 - [**A**] The chef shall be certified according to ISO 7143 SSS v2.0. [a]
 - [**B**] The sight shall have an adjustable range of magnification from 8 to 10x. [b]
 - [**C**] The hopper dimensions shall be minimum 10(h)×10(w)×20(d) cm. [c]
 - [**D**] Acceptable materials are: 6061 aluminum or titanium. [d]
 - [**E**] The product shall be called UltraSaladShooter 9000. [e]
 - [**F**] The product shall be endorsed by Justin Bieber. [f]
 - [**G**] The product shall be endorsed by Miley Cyrus. [f]
- use your best judgment, but be consistent within the project so far and your own solution
 - somehow must address how to build, connect, and use components
 - must be reasonably consistent forward and backward: **E↔U↔Q↔R↔S**
 - be useful: factually correct does not imply useful
- each level as separate section of document in order; do not nest

USER STORIES (8)

I element 1 name
use case

II element 2 name
use case

...

QUESTIONS (18)

I element 1 name
1. question 1
2. question 2
3. question 3

II. element 2 name
1. question 1
2. question 2
3. question 3

...

REQUIREMENTS (18)

I element 1 name
[a] requirement 1 (1)
[b] requirement 2 (2)
[c] requirement 3 (3)

II element 2 name
[a] requirement 1 (1)
[b] requirement 2 (2)
[c] requirement 3 (2)

...

SPECIFICATIONS (18)

I element 1 name
[A] specification 1 [a]
[B] specification 1 [b]
[C] specification 1 [c]

II element 2 name
[A] specification 1 [a]
[B] specification 1 [b]
[C] specification 1 [b]

...

$$8E + 8U + 18Q + 18R + 18S = 70 \text{ entries}$$

Note that every element is self-contained and independent. We have no way to crossreference between elements, which in real life is actually common. If you end up with the same entry in two places, repeat it.