

47006- ANÁLISE E MODELAÇÃO DE SISTEMAS

Metodologias ágeis e *user stories*

Ilídio Oliveira

V2022/05/17

Learning objectives for this lecture

Characterize the principles of backlog management in agile projects

Define and write stories for a given product.

Distinguish use story estimation and prioritization.

Write the acceptance criteria part of a user story.

Compare user stories and use cases with respect to commonalities and differences.

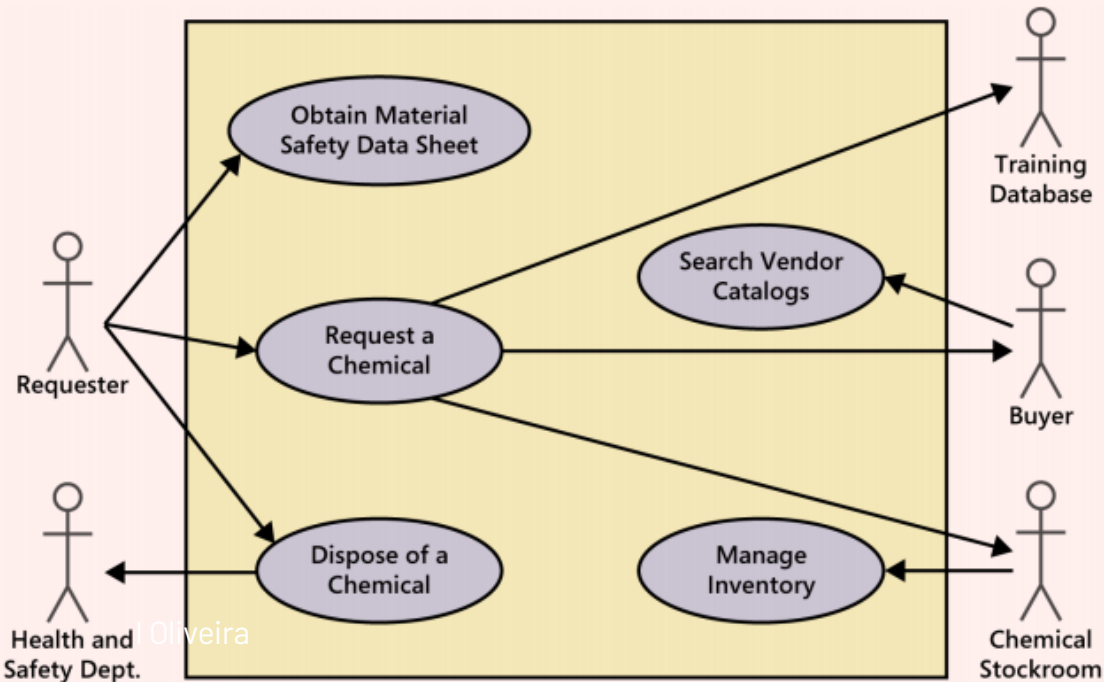
Describe the PivotalTracker story-based development workflow.

Requirements elicitation by exploring user-centered scenarios

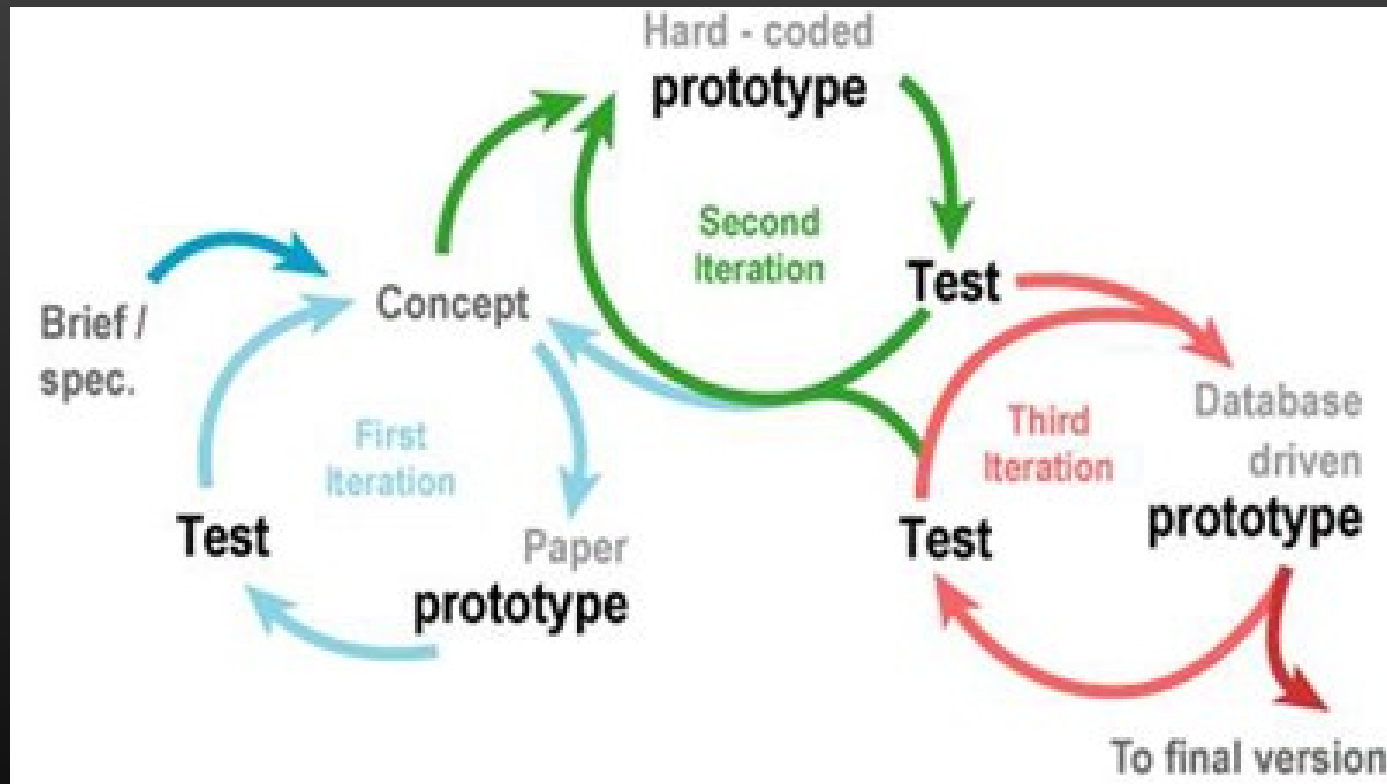
- A) Use cases
- B) User stories
- C) User-centered design (UCD)
- D) Customer Journey Map (Experience maps)

Use cases way

ID and Name:	UC-4 Request a Chemical		
Created By:	Lori	Date Created:	8/22/13
Primary Actor:	Requester	Secondary Actors:	Buyer, Chemical Stockroom, Training Database
Description:	The Requester specifies the desired chemical to request by entering its name or chemical ID number or by importing its structure from a chemical drawing tool. The system either offers the Requester a container of the chemical from the chemical stockroom or lets the Requester order one from a vendor.		
Trigger:	Requester indicates that he wants to request a chemical.		
Preconditions:	PRE-1. User's identity has been authenticated. PRE-2. User is authorized to request chemicals. PRE-3. Chemical inventory database is online.		
Postconditions:	POST-1. Request is stored in the CTS. POST-2. Request was sent to the Chemical Stockroom or to a Buyer.		
Normal Flow:	4.0 Request a Chemical from the Chemical Stockroom 1. Requester specifies the desired chemical. 2. System lists containers of the desired chemical that are in the chemical stockroom, if any. 3. System gives Requester the option to View Container History for any container. 4. Requester selects a specific container or asks to place a vendor order (see 4.1). 5. Requester enters other information to complete the request. 6. System stores the request and notifies the Chemical Stockroom.		
Alternative Flows:	4.1 Request a Chemical from a Vendor 1. Requester searches vendor catalogs for the chemical (see 4.1.E1). 2. System displays a list of vendors for the chemical with available container sizes, grades, and prices. 3. Requester selects a vendor, container size, grade, and number of containers. 4. Requester enters other information to complete the request. 5. System stores the request and notifies the Buyer.		
	4.2 Request a Chemical from a Vendor that is Not Commercially Available 1. System displays message: No vendors for that chemical. 2. Requester asks if he wants to request another chemical (3a) or to exit (4a). 3. If Requester asks to request another chemical, system returns normal flow over. 4. If Requester asks to exit, system terminates use case.		
	5 times per week by each chemist, 200 times per week by chemical if		



UCD: prototyping & acceptance



<https://www.museumsandtheweb.com/mw2007/papers/brown/brown.html>

Rail Europe Experience Map

Guiding Principles

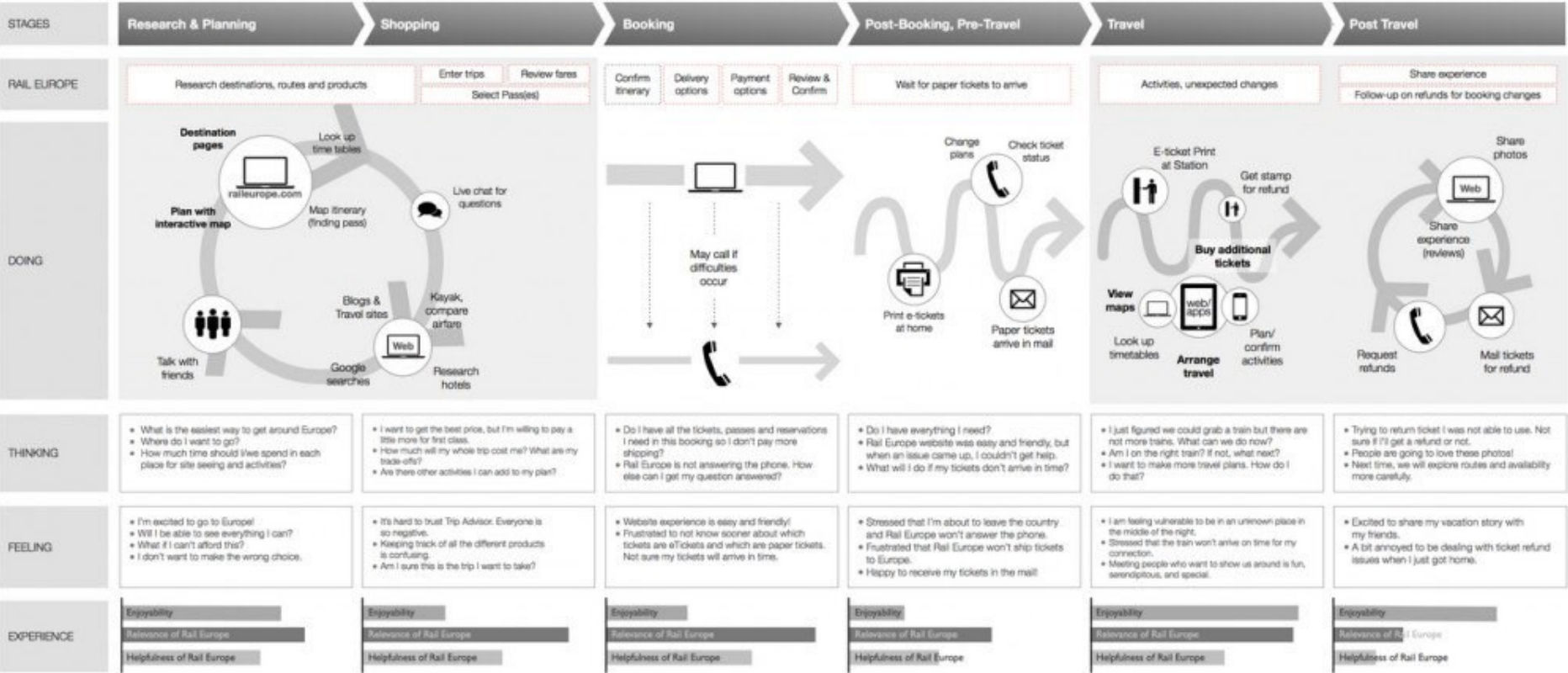
People choose rail travel because it is convenient, easy, and flexible.

Rail booking is only one part of people's larger travel process.

People build their travel plans over time.

People value service that is respectful, effective and personable.

















Customer Journey



Opportunities

GLOBAL			PLANNING, SHOPPING, BOOKING			POST-BOOK, TRAVEL, POST-TRAVEL	
Communicate a clear value proposition.	Help people get the help they need.	Support people in creating their own solutions.	Enable people to plan over time.	Visualize the trip for planning and booking.	Arm customers with information for making decisions.	Improve the paper ticket experience.	Accommodate planning and booking in Europe too.
STAGES: Initial visit	STAGES: Global	STAGES: Global	STAGES: Planning, Shopping	STAGES: Planning, Shopping	STAGES: Shopping, Booking	STAGES: Post-Booking, Travel, Post-Travel	STAGE: Traveling
Make your customers into better, more savvy travelers.	Engage in social media with explicit purposes.		Connect planning, shopping and booking on the web.	Aggregate shipping with a reasonable timeline.		Proactively help people deal with change.	Communicate status clearly at all times.
STAGES: Global	STAGES: Global		STAGES: Planning, Shopping, Booking	STAGE: Booking		STAGES: Post-Booking, Traveling	STAGES: Post-Booking, Post Travel

User stories



CIS board

Story Map by Easy Agile

+ Create Epic

Quick filters ▾

Sprint swimlanes ▾

...

?

Backlog

Navigation

CIS-1

Car Statistics

CIS-4

Phone Integration

CIS-3

Play Media

CIS-2

Fatigue Management

CIS-5

Sprint 1

21 2 0

The 'Young Professional' Driver / Install maps so that I can navigate to places easier

CIS-8

The 'Young Professional' Driver / Touch Screen to navigate easily

CIS-38

The 'Young Professional' Driver / Apple CarPlay Integration so that I can safely send and receive calls, texts and emails from my iOS device while driving

CIS-41

The 'Young Adult' Passenger / Allow Wifi Hotspot to support up to 5 devices

CIS-39

The 'Sunday' Driver / Enable 'Tourist Mode Assist' when travelling outside of standard travel radius

CIS-12

The 'Young Professional' Driver / Integrate local traffic data to better estimate travel times

CIS-10

The 'Sunday' Driver / Show miles/km to empty so that I don't run out of fuel

CIS-23

Sprint 2

32 0 0

The 'Sunday' Driver / Showcase local landmarks if travelling outside of standard travel radius

CIS-11

The 'Young Professional' Driver / Wear and Tear Report so that I can take preventative action to preserve the life of the car if needed

CIS-26

The 'Family' Driver / Microphone so that I can make phone calls safely while I'm driving

CIS-19

The 'Family' Driver / Graphical User Interface for easier use of media while driving

CIS-18

The 'Young Professional' Driver / Android Auto Integration so that I can safely send and receive calls, texts and emails while driving

CIS-42

The 'Family' Driver / Music Streaming service so that I can listen to music on trips

CIS-43

The 'Sunday' Driver / Safe Time Driving Display

CIS-44

Quick filters ▾

Sprint 1

The 'Family' Driver / 'Hot Cues' to make ... CIS-28

Sprint 2

Unscheduled

The 'Young Professional' Driver / Custom ... CIS-9

The 'Family' Driver / A 'Favourites' Cont... CIS-37

The 'Sunday' Driver / Engine Temperatu... CIS-24

The 'Young Professional' Driver / Amaz... CIS-40

The 'Sunday' Driver / Show designated '... CIS-31

The 'Family' Driver / Object Detection fo... CIS-33

The 'Family' Driver / Safe Volume Adjus... CIS-17

The 'Young Professional' Driver / Aux C... CIS-16

The 'Young Professional' Driver / Do No... CIS-21

The 'Family' Driver / Time/Distance to m... CIS-25

The 'Young Adult' Passenger / Spotify In... CIS-35

Use cases e os métodos ágeis → Use Cases 2.0

A granularidade dos casos de uso é algo “pesada” para a gestão do dia-a-dia

Proposta Use Cases 2.0

- “Fatias” de funcionalidade

Ponto de partida: use cases

- ...com a flexibilidade das *user stories/use case slices*



<https://youtu.be/p5gDbf0je8k>

Unidade mais conveniente: “fatia” de funcionalidade



a use case and its properties captured on a sticky note

7.1 select and buy
1 product

flows: BF
test: 1 product,
default payment,
valid details

5

7.2 select and buy
100 products

flows: BF
test: 100 products,
default payment,
valid details

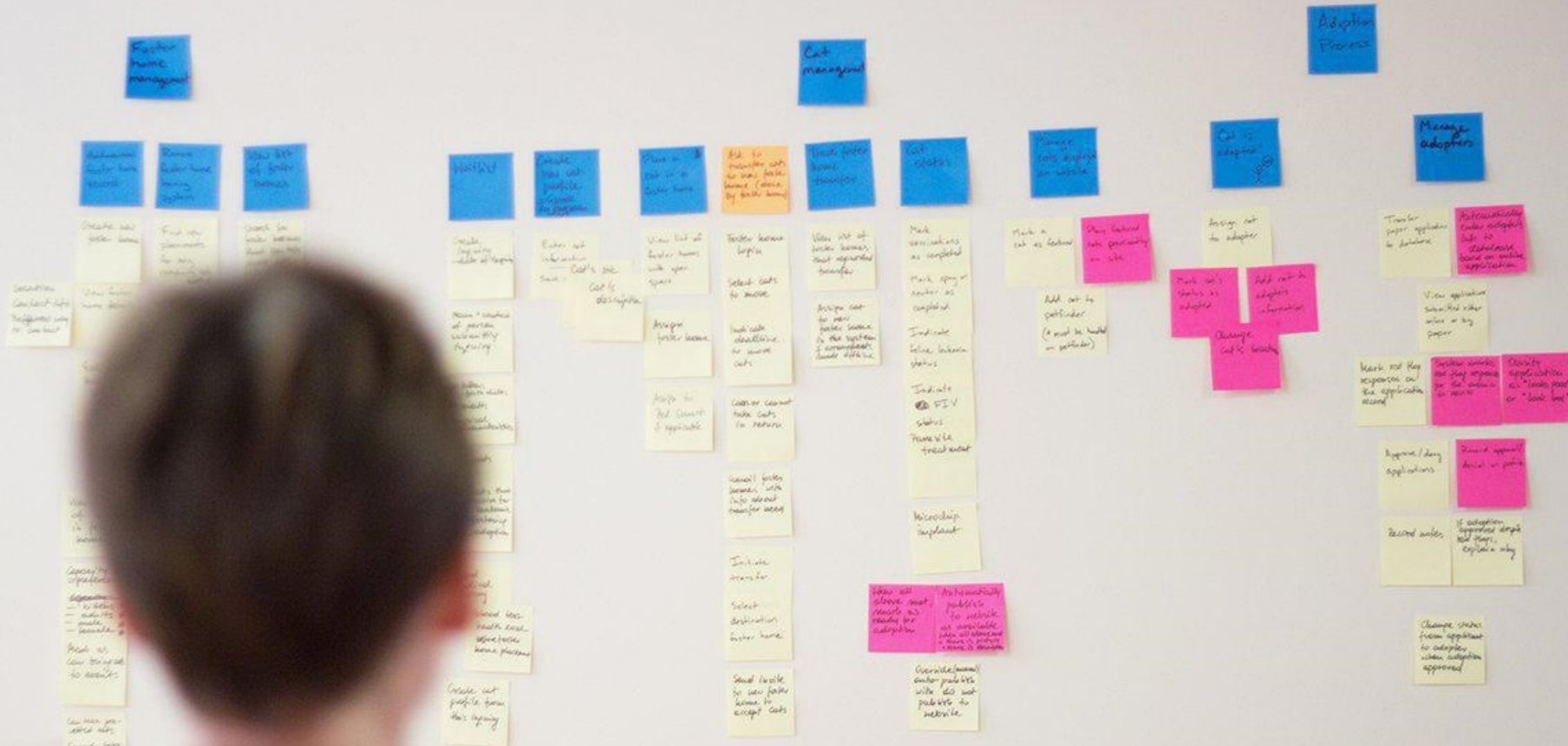
5

7.3 support systems
unavailable

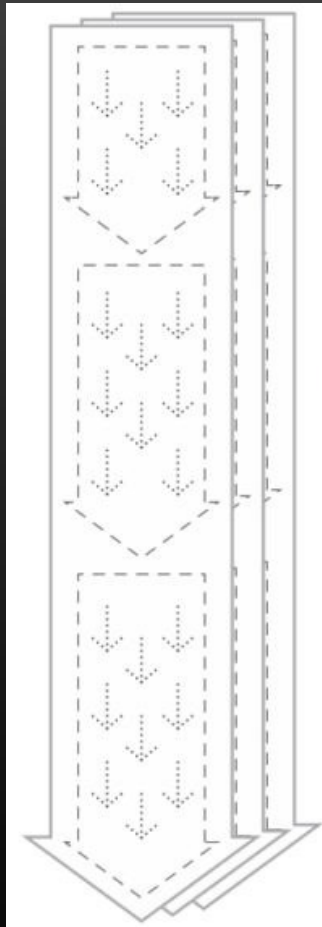
flows: BF, A9, A10,
A1, A12
test: select product,
provide information,
disconnect each
system in between₁₃

some slices from the
use case captured on
their own sticky notes

Sticky note metaphor



The story should clarify how to check if it is working



"As a [persona],
I want to [do something]
so that I can [realize a reward]"

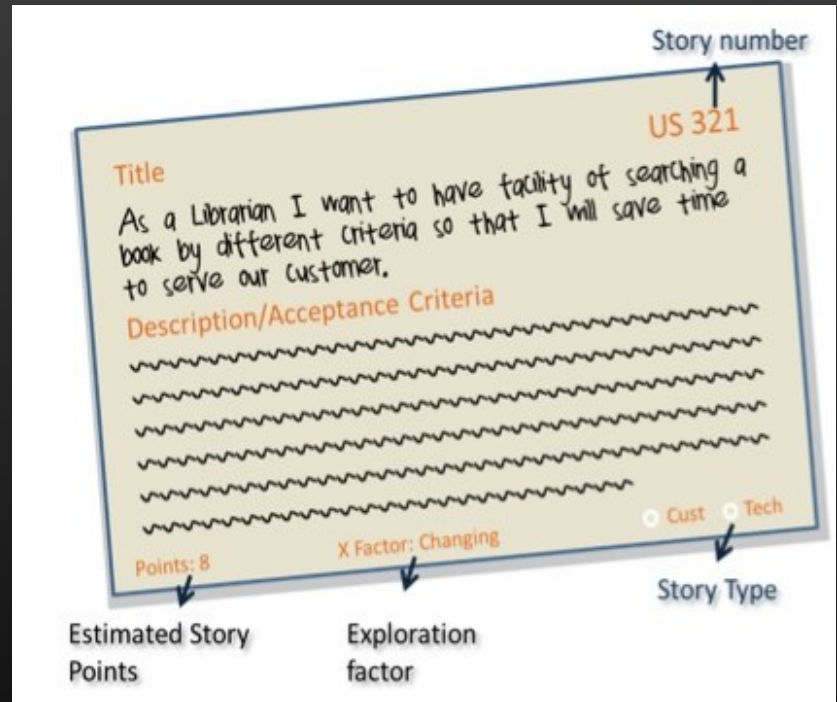
Who is this user?
What makes them tick?
Who's an example of such a person?

Why do they want to do this?
What's the benefit/reward?
How will we know of it's
working?

User stories in agile methods

The *backlog* is the prioritized list of user stories —requirements— for the product and their allocation to upcoming iterations (called sprints in the agile development method called Scrum.)

User story: a “short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system” (Cohn 2010)



User story != use case

→ See [examples](#)

user stories focus on functionality

User Story Title
As a <user role> I want to <goal> so that <benefit>.

Template

Find Reviews Near Address
As a typical user I want to see unbiased reviews of a restaurant near an address so that I can decide where to go for dinner.

FIGURE 5.2 A user story template and card

Automatic Builds
As a developer I want the builds to automatically run when I check in code so that regression errors are detected when they are introduced.

Undesirable technical story

The user story

A “short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system” (Cohn 2010).

User stories often are written according to the following structure (other styles also are used):

As a <type of user>, I want <some goal> so that <some reason>.

→ Advantages of the “As a user, I want” user story template.

As a customer, I want to add an item into shopping cart.

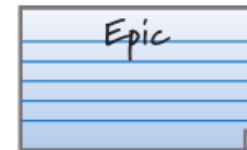
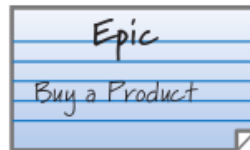
E.g. As a customer, I want to receive an SMS when the item is arrived so that I can go pick it up.

<role> represents the person, system, subsystem or any entity else who will interact with the system to be implemented to achieve a goal. He or she will gain values by interacting with the system.

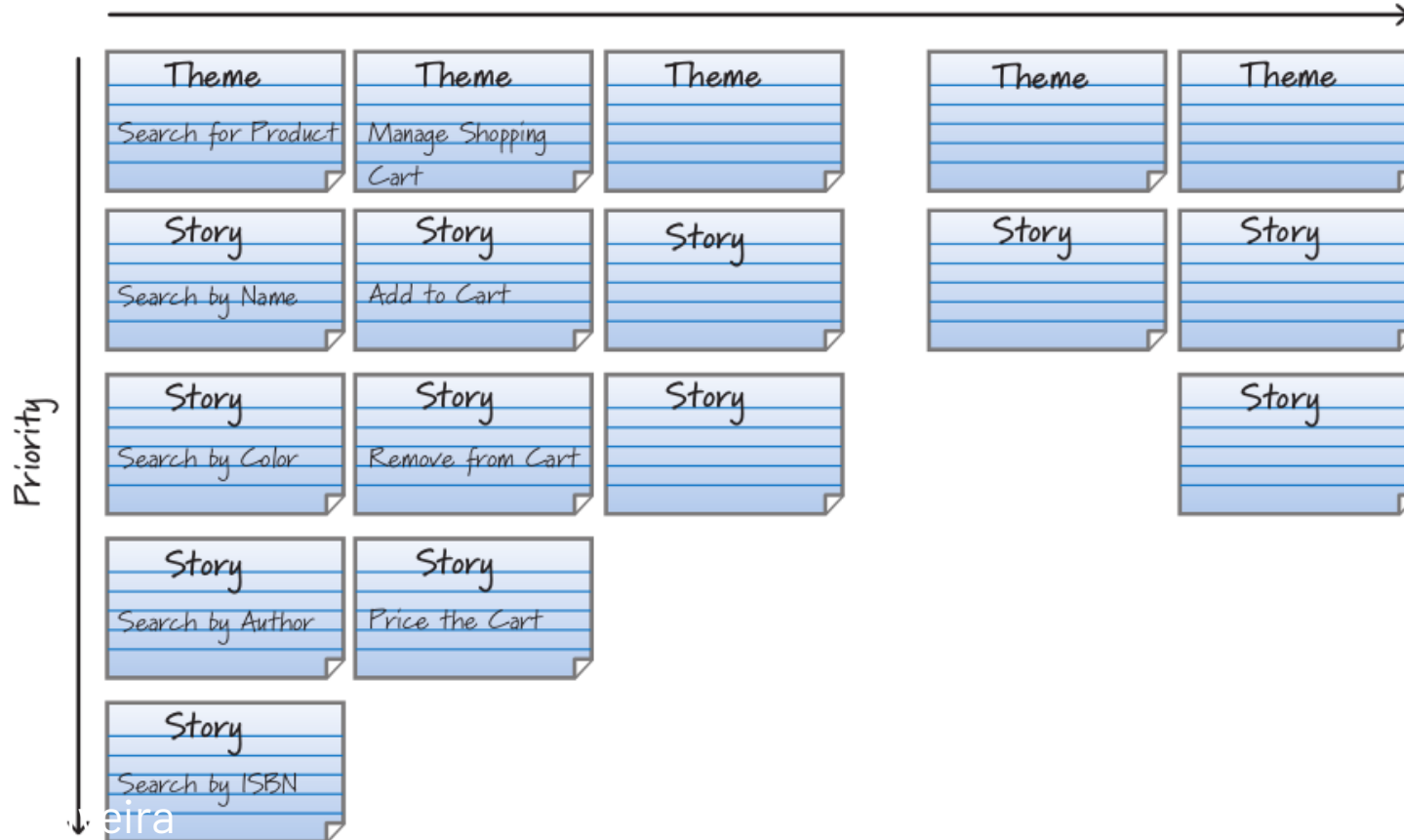
<business objective> represents a user's expectation that can be accomplished through interacting with the system.

<business value> represents the value behind the interaction with the system. May be omitted, if obvious from the business objective.

Finding good stories

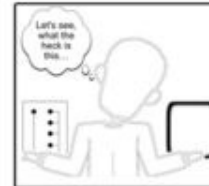


Workflow or usage sequence (over time)



Organizing the stories in priority stripes

STRIPE 0 TOPLINE NARRATIVE



...

TIME →

STRIPE 1 HIGH PRIORITY STORIES



STRIPE 2, ETC. LOWER PRIORITY STORIES

PRIORITY ↓

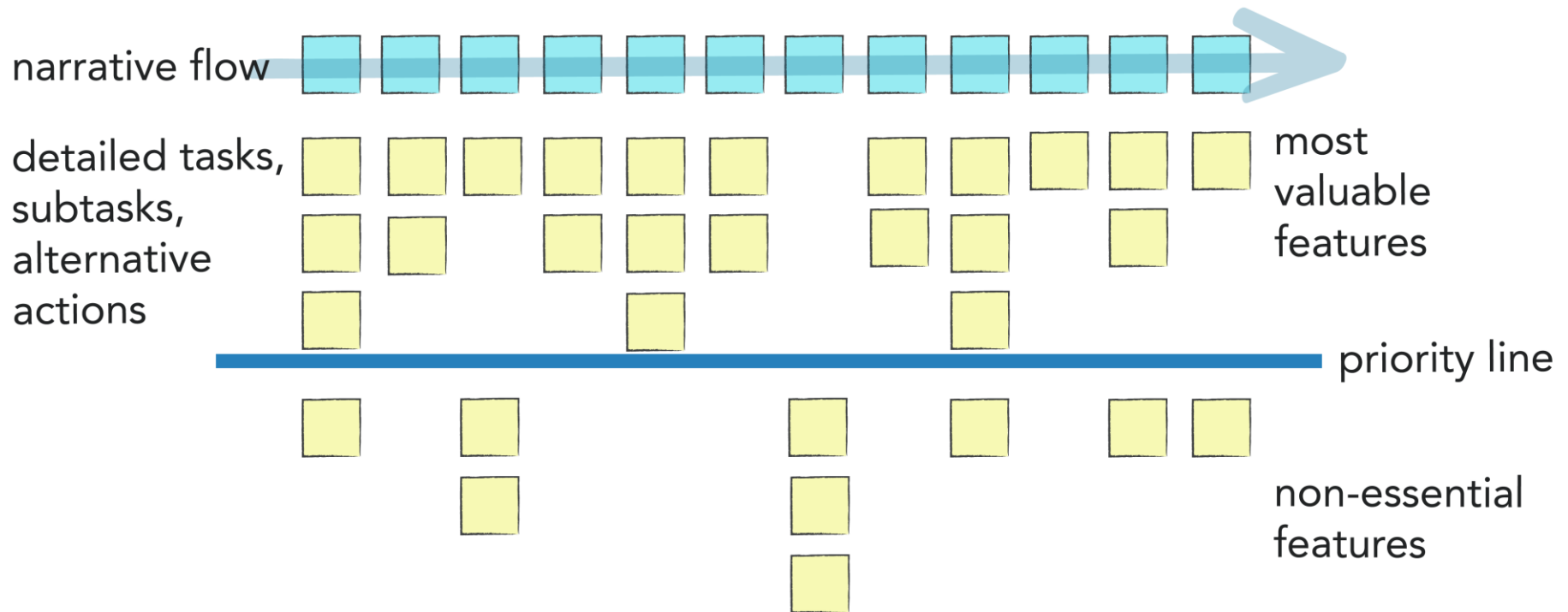


source: adapted from Jeff Patton's 'User Story Mapping'

© 2015 COWAN+

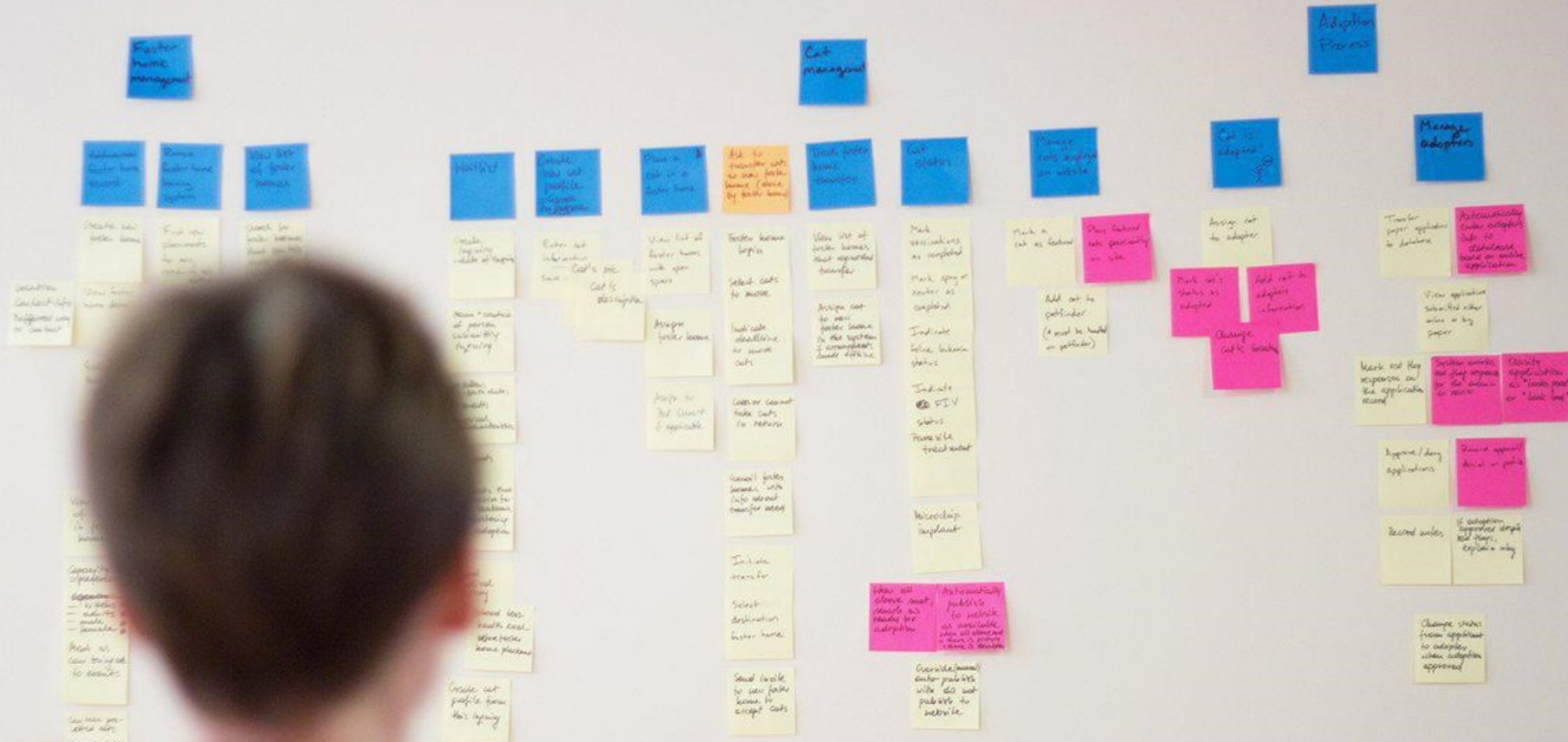
The user story map

User Story Map



<https://www.cactusgroup.com/blog/2017/07/31/user-story-mapping-high-level-release-plan/>

Physical boards or in software tools...



Acceptance criteria

Rather than specifying functional requirements, agile teams typically elaborate a refined user story into a set of acceptance tests that collectively describe the story's "conditions of satisfaction."

Add Prospect

As a property manager I want to add a new prospect to the lead management system so I can track my interactions with the prospect.

Conditions of Satisfaction

Capture name, email, phone #, contact date, contact format, lease type, and move-in date

Verify prospect is associated with an existing campaign

→ See [examples](#)

A structured language for writing the acceptance criteria

GIVEN [necessary context] WHEN [action] THEN [reaction].

→ [Gerkin DSL](#).

→ Used to create AC that are meant to be automatize (feed the tests)

```
Title (one line describing the story)
```

```
Narrative:
```

```
As a [role]
```

```
I want [feature]
```

```
So that [benefit]
```

```
Acceptance Criteria: (presented as Scenarios)
```

```
Scenario 1: Title
```

```
Given [context]
```

```
    And [some more context]...
```


```
When [event]
```





```
Then [outcome]
```

```
    And [another outcome]...
```

```
Scenario 2: ...
```

User story as a colaboration context

▼ Frank Can Add Another Person as a Friend 

 ID #115218319    Close

STORY TYPE

★ Feature ▼

POINTS

⊖ Unestimated ▼

STATE

Start Unscheduled ▼

REQUESTER

RJ Ryan Jones ▼

OWNERS

<none> +

FOLLOW THIS STORY

(1 follower) ☒

Updated: less than a minute ago

DESCRIPTION [\(edit\)](#)

As Frank I want to add a friend I searched for to my friend network so that I can see their posts, they can see my posts and I can direct message them

GIVEN I have searched for a friend's name

WHEN I select "Add Friend" next to my friend's name

THEN my friend's name should appear in my friend list on my homepage

Dev Notes: The added friend needs to be added to the Frank's friends in database

Design Notes: Attached are mocks for the button and placement

LABELS

add friend | × individual user | × ▼

Scenario: Wilson posts to his own blog
Given I am logged in as Wilson
When I try to post to "Expensive Therapy"
Then I should see "Your article was published."

Scenario: Wilson fails to post to somebody else's blog
Given I am logged in as Wilson
When I try to post to "Greg's anti-tax rants"
Then I should see "Hey! That's not your blog!"

Scenario: Greg posts to a client's blog
Given I am logged in as Greg
When I try to post to "Expensive Therapy"
Then I should see "Your article was published."

Stories define your project

Every project starts with a story, no matter what you're building. Tracker helps your team better develop and keep track of them while they progress from start to delivered.

Start with a good story

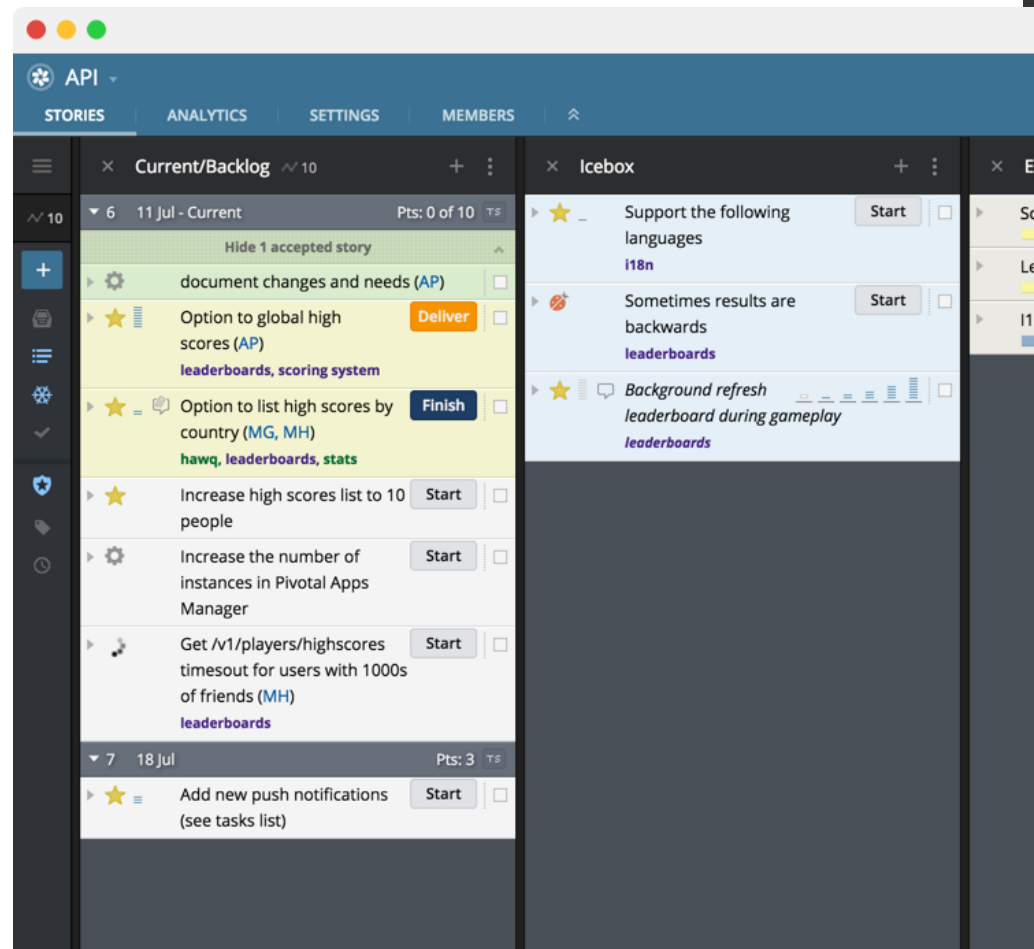
A story is a small, actionable bit of work that's either a placeholder for a future conversation or a reflection of one that already happened. Outlining what a user needs helps you focus on the what, not the how.

Define the story

Select among features, bugs, and chores to strike a healthy balance between building new features, staying ahead of technical debt, and keeping the bugs from piling up.

Estimate, then prioritize

Writing the story is just the beginning—now you get to rap about it. Estimate as a team to uncover the story's complexity. Choose among several point scales, then drag-and-drop to prioritize by iteration.



PivotalTracker style

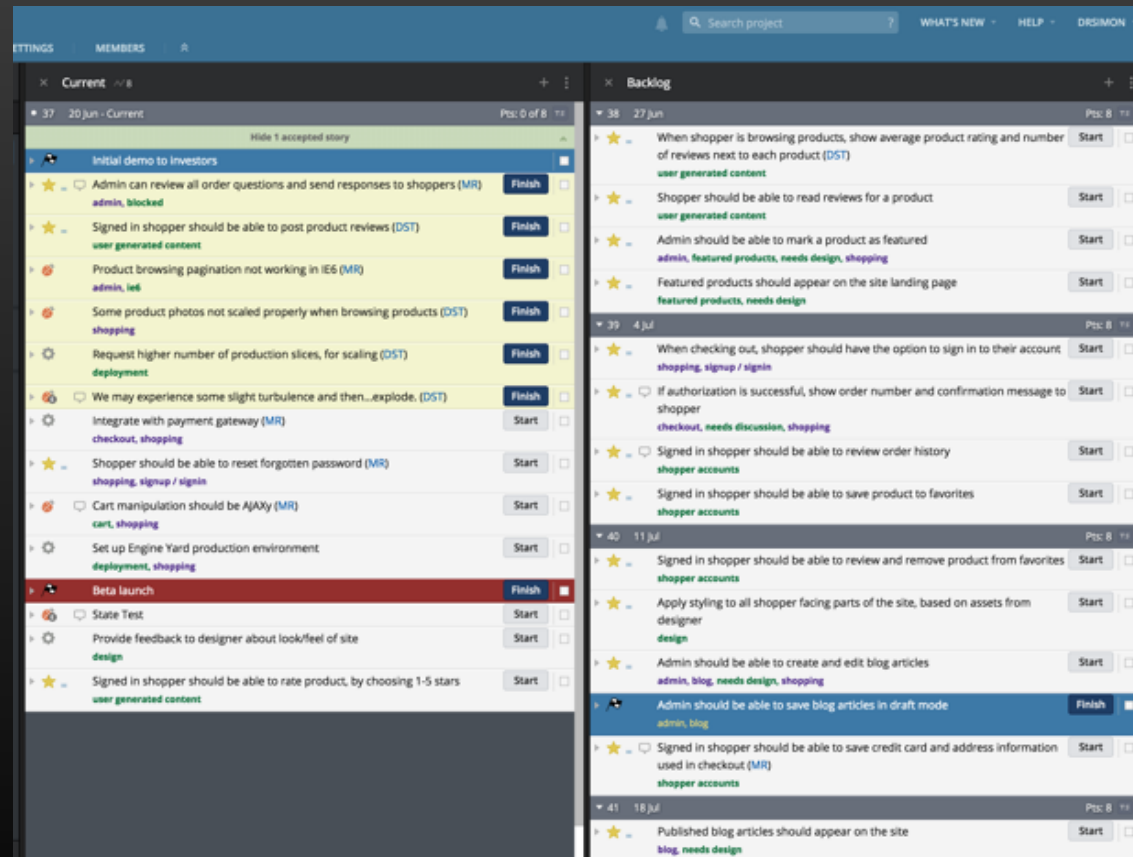
Tracker lists stories in a project's Current and Backlog [panels](#) in priority order.

Dragging a story to the top of your Backlog makes it the top priority

Tracker is a “pull” scheduling system — it pulls in stories to fill the team's WIP limit, based on [velocity](#).

Tracker automatically moves stories from the top of your project's Backlog into the current iteration according to the current [velocity](#). (depending on the individual [story estimates](#)).

Tracker projects use a linear point scale (i.e., 0, 1, 2, 3). You can change



PivotalTracker pannels

The image displays the PivotalTracker web application interface, divided into four main panels: Done, Current, Backlog, and Icebox. Numbered callouts (1-8) highlight specific features and actions:

- 1:** Icebox panel, "Signed in shopper should be able to review order details" story.
- 2:** Backlog panel, "Signed in shopper should be able to save card and address information used in checkout" story.
- 3:** Backlog panel, "Apply styling to all shopper facing parts of the site, based on assets from designer" story.
- 4:** Backlog panel, "Set up Cloud production environment" story.
- 5:** Current panel, "Cart manipulation should be AJAXy (MR)" story.
- 6:** Current panel, "Shopper should be able to reset forgotten password (MR)" story.
- 7:** Current panel, "Integrate with" story, showing a "REASON FOR REJECTION (OPTIONAL)" dialog box with the text "Integration fails using Alliance credits".
- 8:** Done panel, "Admin can review all order questions and send responses to shoppers" story.

Story Points

Funcionalidades (encomenda de comida online):

- F1: Início de sessão do utilizador (login)
- F2: Registo de novo utilizador na plataforma
- F3: Listar promoções em destaque do dia.
- F4: Colocar a encomenda (inclui pagamento)

Escala:

1pt: muito fácil. Direto de se implementar e âmbito reduzido.

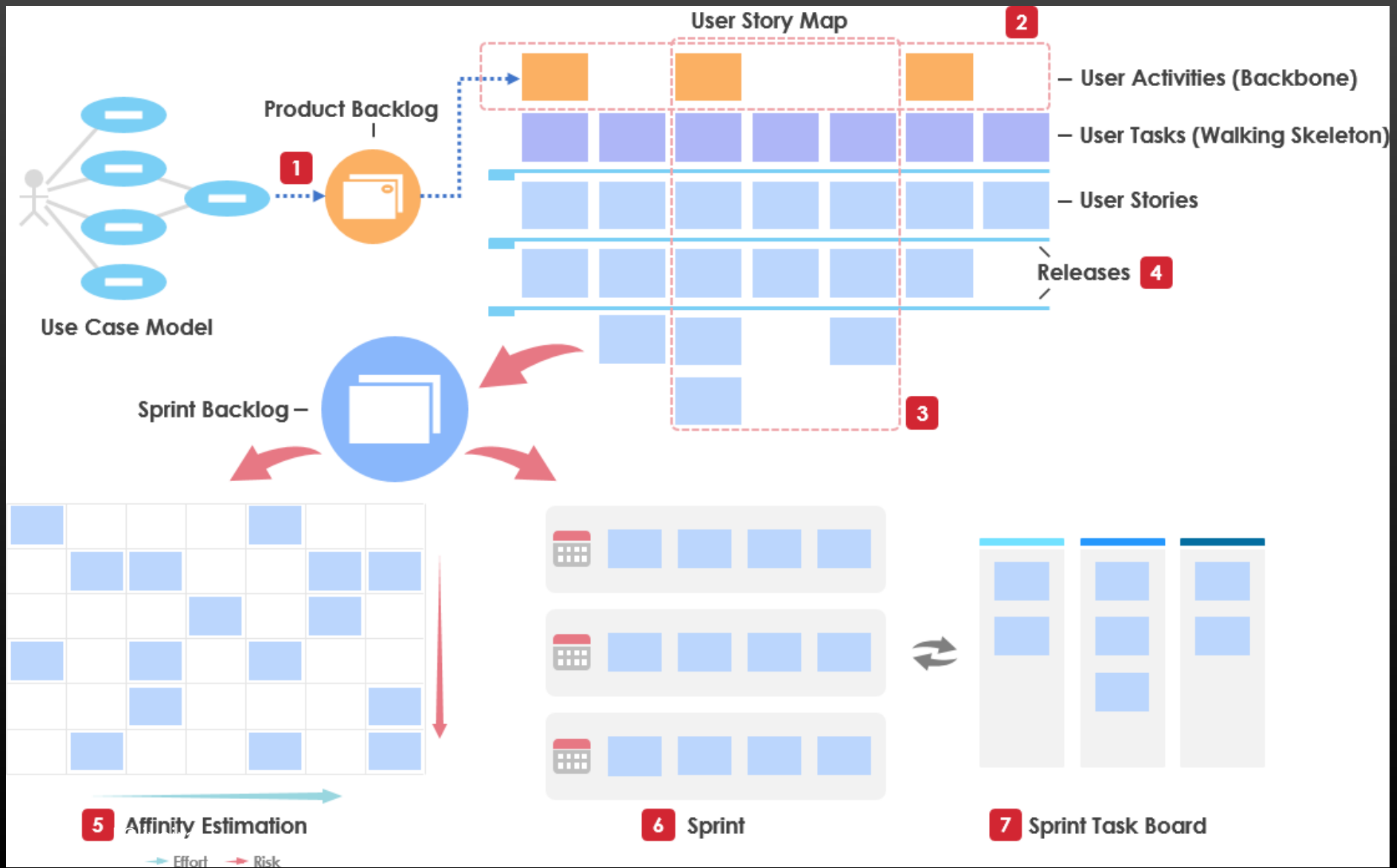
2pts: acessível; não oferece grande dificuldade.

4pts: complexo; tem várias interdependências (de outros módulos/serviços) ou um fluxo elaborado

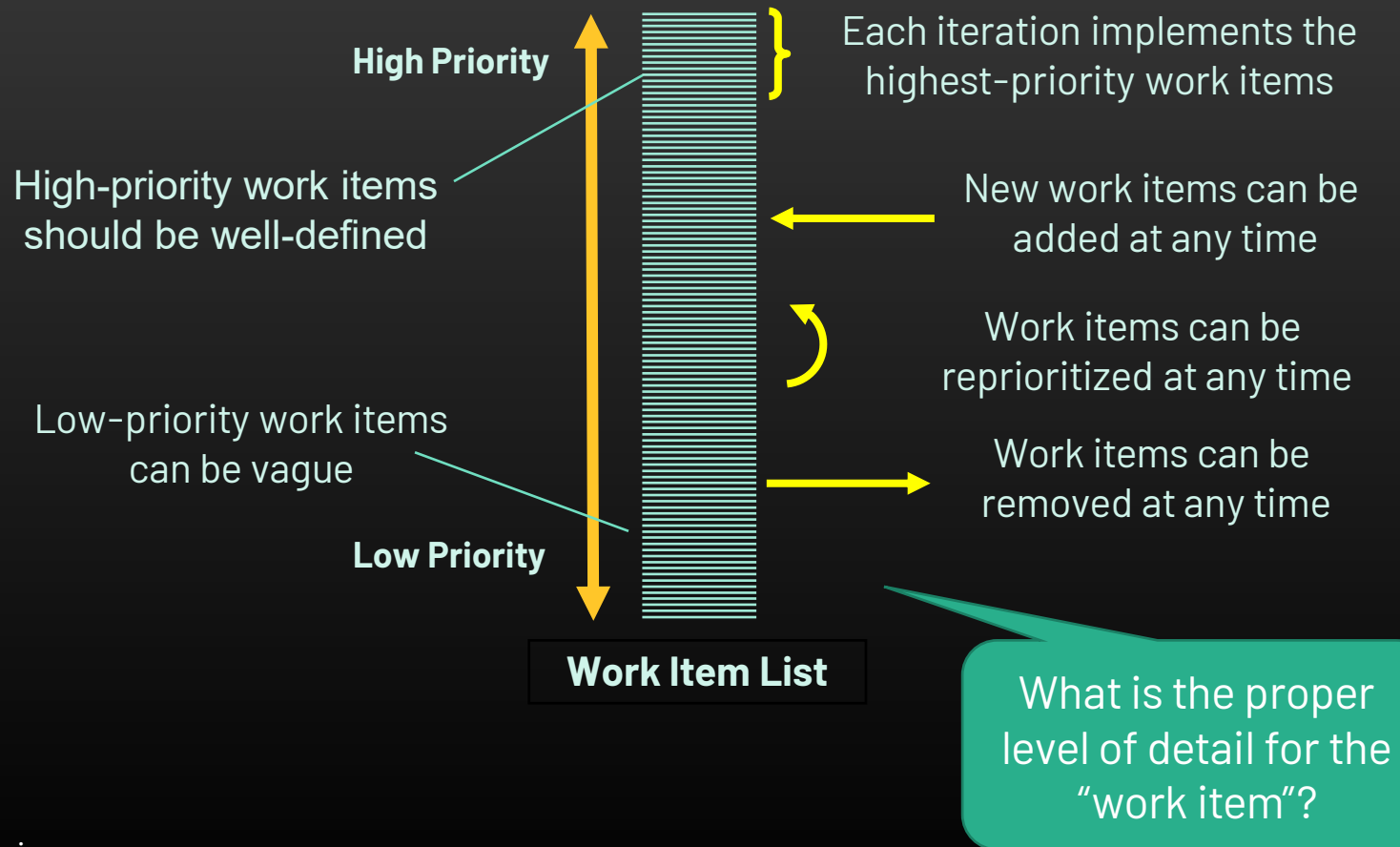
8pts: muito complexo; requer integrações, tecnologias ou conhecimentos que não são completamente dominados

→ <http://bit.ly/2IUrnMn>

Agile in Visual Paradigm



Managing the work items list (*backlog*)



Scrum framework activities

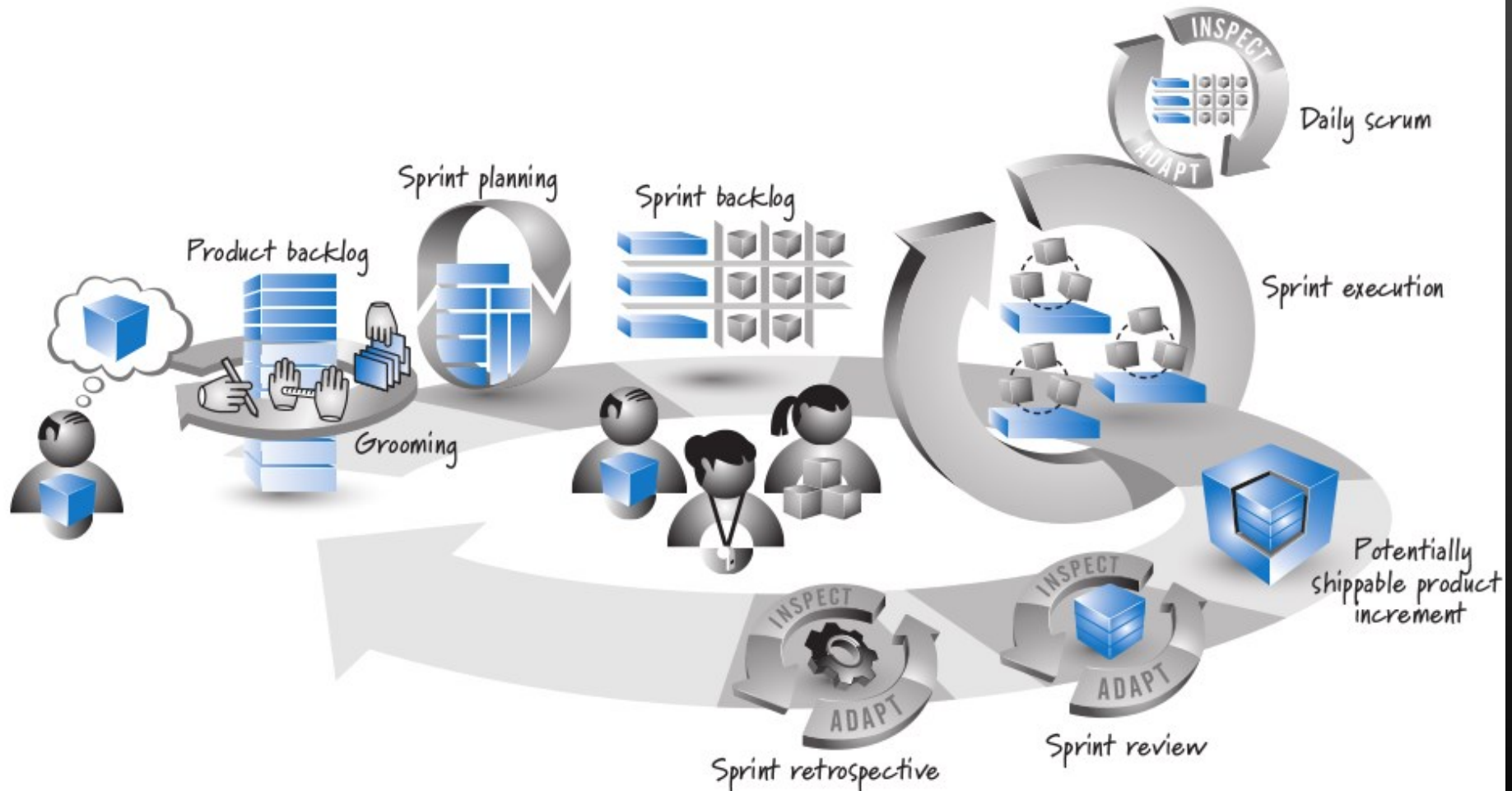


FIGURE 2.3 Scrum framework

Requirements elicitation by exploring user-centered scenarios

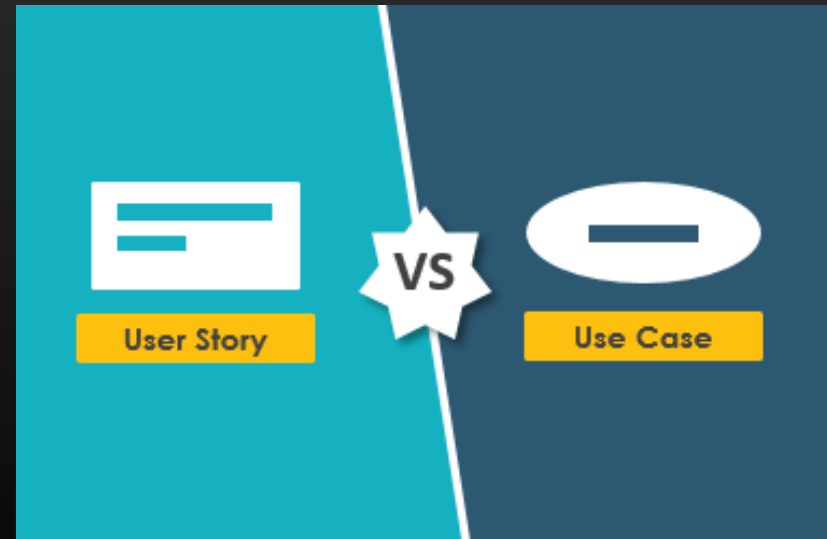
- A) Use cases
- B) User stories
- C) User-centered design (UCD)
- D) Customer Journey Map (Experience maps)

Recall: use cases and related assets

A *use case* describes a sequence of interactions between a system and an external actor that results in the actor being able to achieve some outcome of value.

The names of use cases are always written in the form of a verb followed by an object.

The use case is supplemented with a detailed description (following a template)



ID and Name:	UC-4 Request a Chemical		
Created By:	Lori	Date Created:	8/22/13
Primary Actor:	Requester	Secondary Actors:	Buyer, Chemical Stockroom, Training Database
Description:	The Requester specifies the desired chemical to request by entering its name or chemical ID number or by importing its structure from a chemical drawing tool. The system either offers the Requester a container of the chemical from the chemical stockroom or lets the Requester order one from a vendor.		
Trigger:	Requester indicates that he wants to request a chemical.		
Preconditions:	PRE-1. User's identity has been authenticated. PRE-2. User is authorized to request chemicals. PRE-3. Chemical inventory database is online.		
Postconditions:	POST-1. Request is stored in the CTS. POST-2. Request was sent to the Chemical Stockroom or to a Buyer.		
Normal Flow:	4.0 Request a Chemical from the Chemical Stockroom <ol style="list-style-type: none"> 1. Requester specifies the desired chemical. 2. System lists containers of the desired chemical that are in the chemical stockroom, if any. 3. System gives Requester the option to View Container History for any container. 4. Requester selects a specific container or asks to place a vendor order (see 4.1). 5. Requester enters other information to complete the request. 6. System stores the request and notifies the Chemical Stockroom. 		
Alternative Flows:	4.1 Request a Chemical from a Vendor <ol style="list-style-type: none"> 1. Requester searches vendor catalogs for the chemical (see 4.1.E1). 2. System displays a list of vendors for the chemical with available container sizes, grades, and prices. 3. Requester selects a vendor, container size, grade, and number of containers. 4. Requester enters other information to complete the request. 5. System stores the request and notifies the Buyer. 		
Exceptions:	4.1.E1 Chemical Is Not Commercially Available <ol style="list-style-type: none"> 1. System displays message: No vendors for that chemical. 2. System asks Requester if he wants to request another chemical (3a) or to exit (4a). 3a. Requester asks to request another chemical. 3b. System starts normal flow over. 4a. Requester asks to exit. 4b. System terminates use case. 		
Priority:	High		
Frequency of Use:	Approximately 5 times per week by each chemist, 200 times per week by chemical stockroom staff		

Use cases and similar user stories

TABLE 8-2 Some sample use cases and corresponding user stories

Application	Sample use case	Corresponding user story
Chemical tracking system	Request a Chemical	As a chemist, I want to request a chemical so that I can perform experiments.
Airport check-in kiosk	Check in for a Flight	As a traveler, I want to check in for a flight so that I can fly to my destination.
Accounting system	Create an Invoice	As a small business owner, I want to create an invoice so that I can bill a customer.
Online bookstore	Update Customer Profile	As a customer, I want to update my customer profile so that future purchases are billed to a new credit card number.

More often: user story \subset use case

Recall that user stories are concise statements of user needs, in contrast to the richer description that a use case provides. In the agile world, a user story sometimes covers the same scope as an entire use case, but in other cases a user story represents just a single scenario or alternative flow. If an agile development team were discussing requirements for the CTS, they might come up with user stories such as the following:

As a chemist, I want to request a chemical so that I can perform experiments.

As a chemist, I want to request a chemical from the Chemical Stockroom so that I can use it immediately.

As a chemist, I want to request a chemical from a vendor because I don't trust the purity of any of the samples available in the Chemical Stockroom.

The first of these three stories corresponds to the use case as a whole. The second and third user stories represent the normal flow of the use case and the first alternative flow, from Figure 8-3.

At this level, use cases look much like user stories. Both are focused on understanding what different types of users need to accomplish through interactions with a software system. However, the two processes move in different directions from these similar starting points, as illustrated in Figure 8-1. Both approaches can also produce other deliverables, such as visual analysis models, but Figure 8-1 illustrates the core distinction.

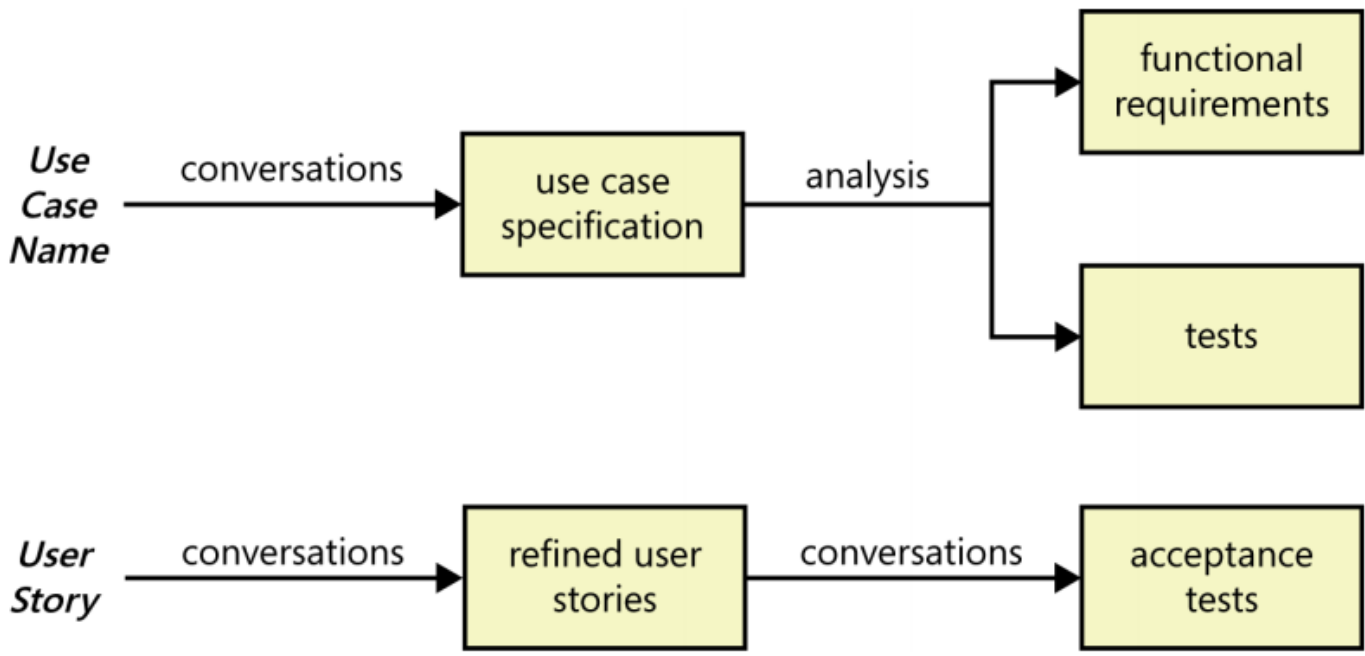


FIGURE 8-1 How user requirements lead to functional requirements and tests with the use case approach and the user story approach.

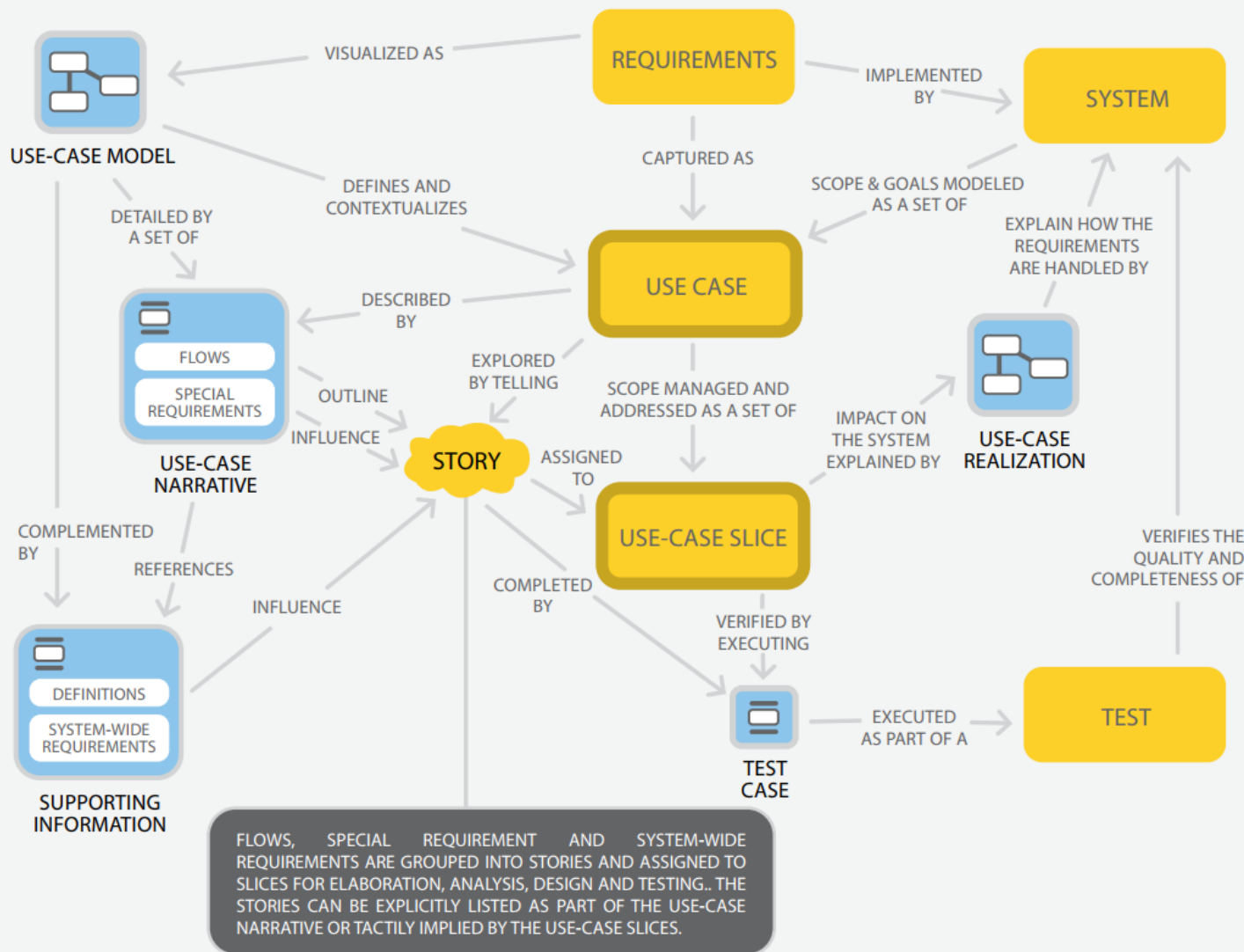
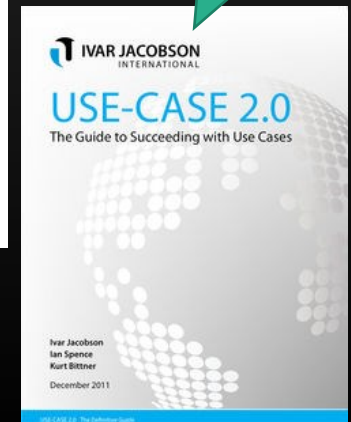


FIGURE 9: THE USE-CASE 2.0 WORK PRODUCTS

Copy in Moodle...



Jacobson: flows in a use case match stories

A story is described by part of the use-case narrative, one or more flows and special requirements, and one or more test cases. The key to finding effective stories is to understand the structure of the use-case narrative. The network of flows can be thought of as a map that summarizes all the stories needed to describe the use case. **Figure 8** illustrates the relationship between the flows of a use-case narrative and the stories it describes.

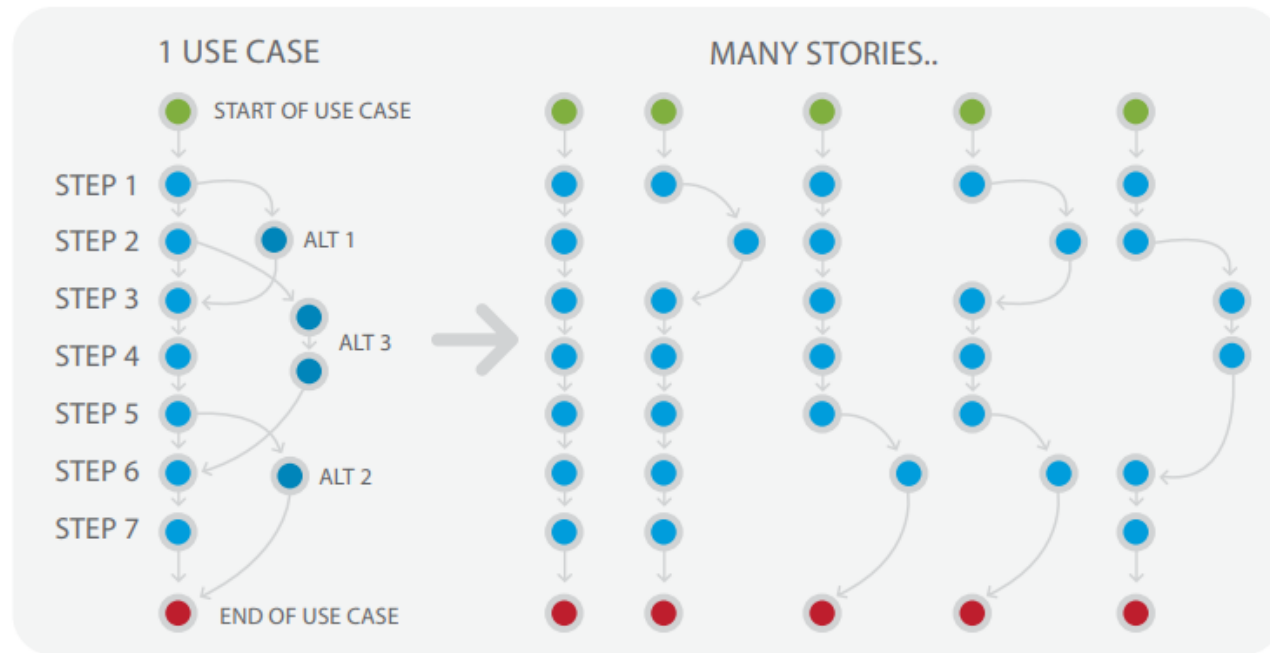


FIGURE 8:
THE RELATIONSHIP BETWEEN THE FLOWS AND THE STORIES

Figure 4. Use cases, use-case slices, increments, and releases.

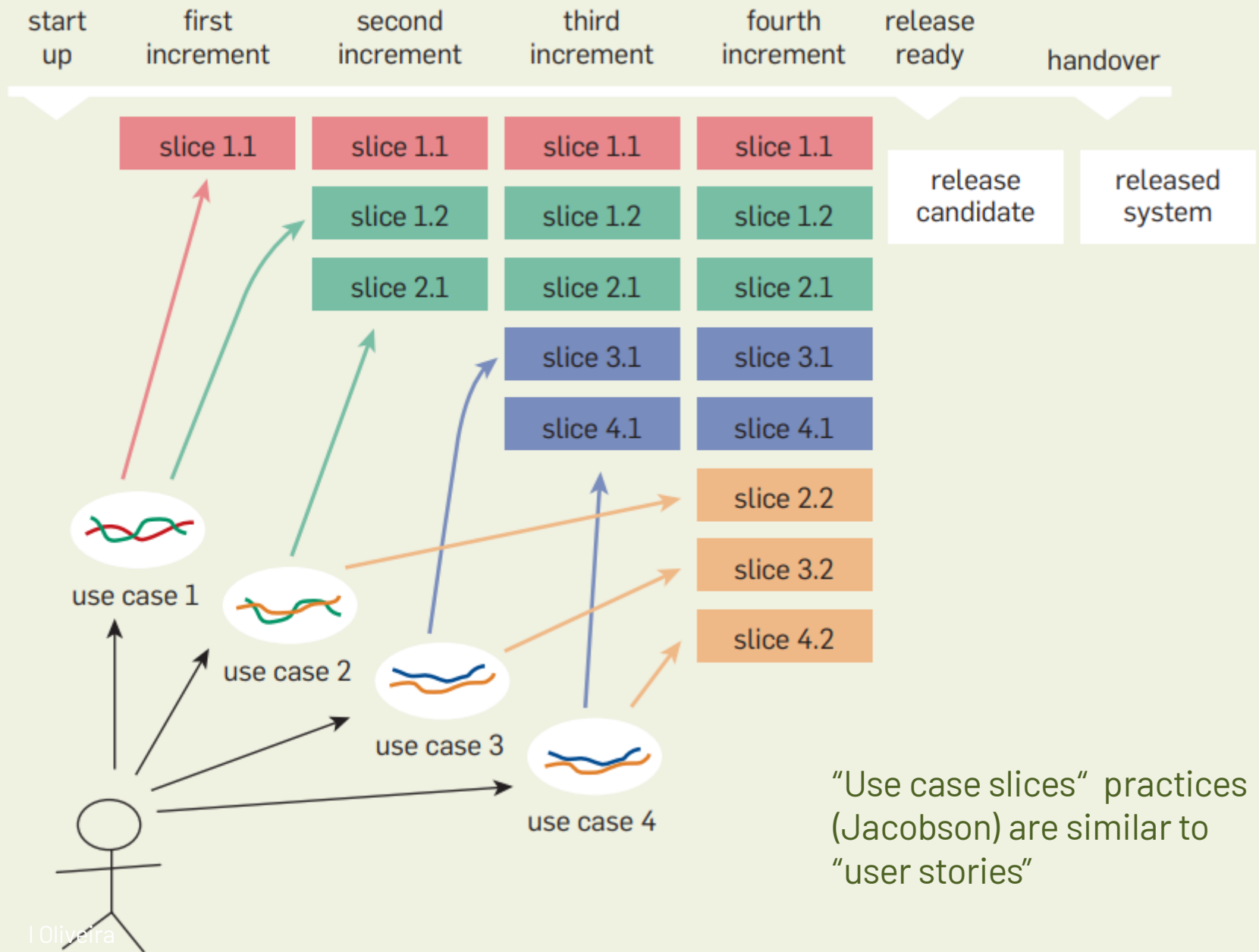
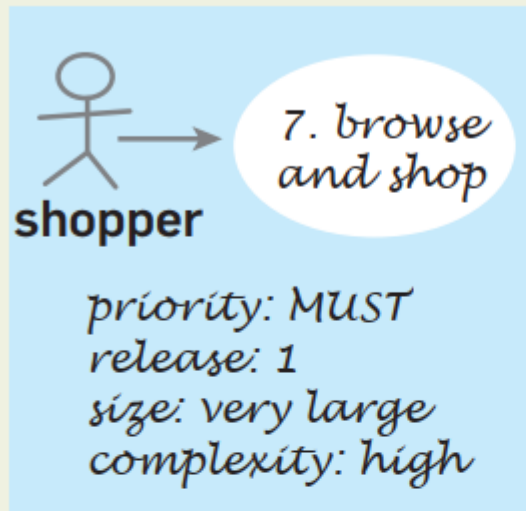


Figure 5. Capturing the properties of a use case and its slices using sticky notes.



a use case and its properties
captured on a sticky note

7.1 select and buy
1 product

flows: BF
test: 1 product,
default payment,
valid details

5

7.3 support systems
unavailable

flows: BF, A9, A10,
A1, A12
test: select product,
provide information,
disconnect each
system in between¹³

7.2 select and buy
100 products

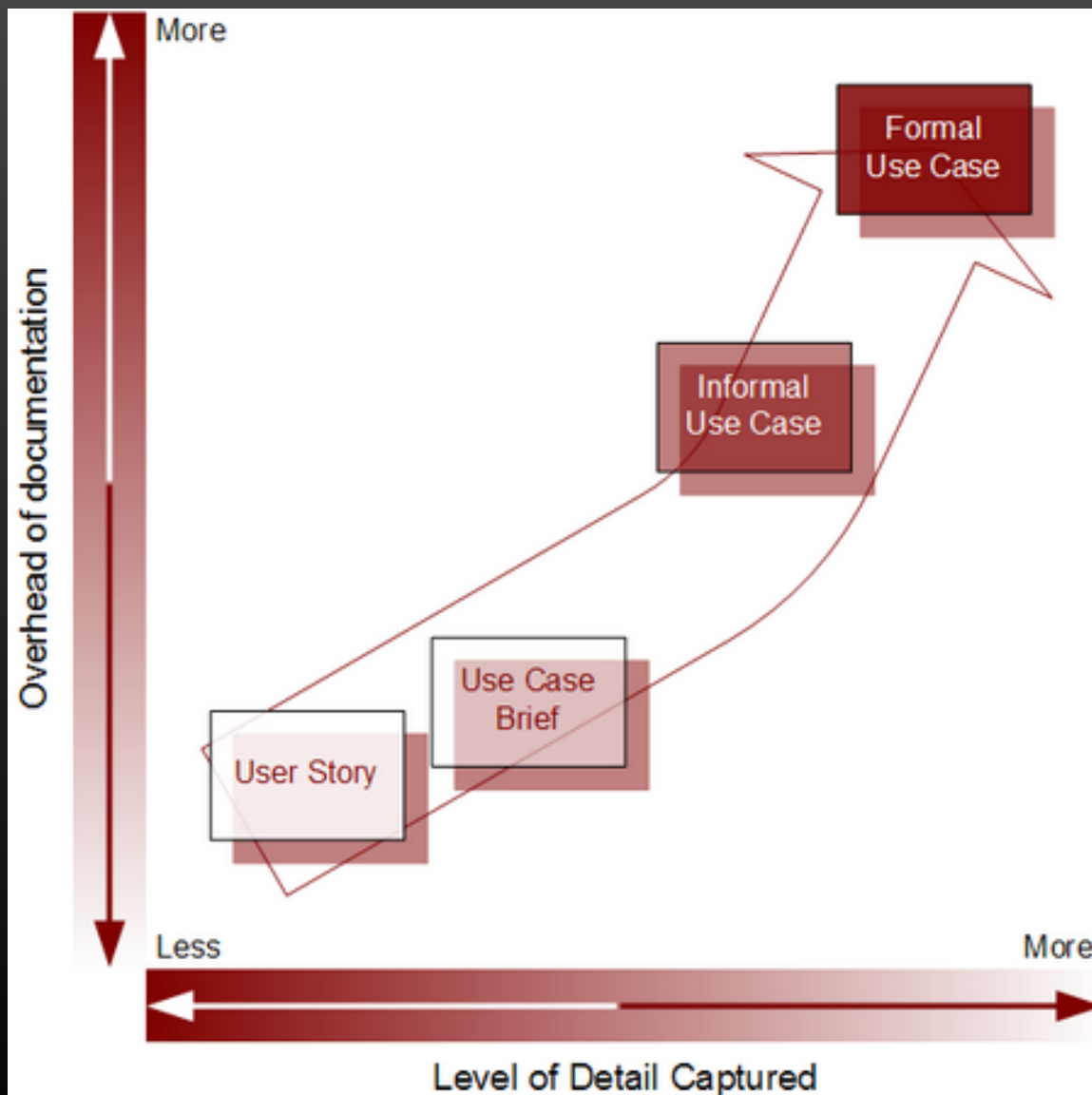
flows: BF
test: 100 products,
default payment,
valid details

5

some slices from the
use case captured on
their own sticky notes

User stories and use cases: what is the difference?

Common	Use-Cases	User-story
<ul style="list-style-type: none">• Both take a usage-centric approach• Both are placeholders for a conversation• Both result in test cases that represent the acceptance criteria• Both can be estimated	<ul style="list-style-type: none">• Big picture to help people understand the extent of the system and its value• Dive further into describing how the user imagines interacting with the system to accomplish his objective.• Provide project participants with a structure and context that a collection of user stories lacks• You can examine each element of a use case (flows, preconditions, postconditions, and so on) to look for pertinent functional and nonfunctional requirements and to derive tests. This helps you avoid overlooking any requirements• Active scope management	<ul style="list-style-type: none">• Concise statement of a user's needs• Easy access to domain experts available (refine the story as needed)• More suited to act as a backlog item for daily activities (Scrum, Kanban, specification by example)• Explicit acceptance criteria



Usage-centric approaches to requirements: benefits and limitations

Both use cases and user stories shift from the product-centric perspective of requirements elicitation to discussing what users need to accomplish, in contrast to asking users what they want the system to do.

The intent of this approach is to describe tasks that users will need to perform with the system, or user-system interactions that will result in a valuable outcome for some stakeholder.

Use cases and user stories work well for exploring the requirements for business applications, websites, kiosks, and systems that let a user control a piece of hardware.

However: they are inadequate for understanding the requirements of certain types of applications. Applications such as batch processes, computationally intensive systems, business analytics, and data warehousing might have just a few use cases. The complexity of these applications lies in the computations performed, the data found and compiled, or the reports generated, not in the user-system interactions.

Nor are use cases and user stories sufficient for specifying many embedded and other real-time systems.

Take away messages

- Agile projects (especially Scrum ones) use a product backlog, which is a prioritized list of the functionality to be developed.
- Product backlog items can be whatever the team desires, but user stories have emerged as the best and most popular form of product backlog items.
- Both use cases and user stories focus on conversations and system usage by people.
- Use cases provide more structure and a way to document details collected in analysis.
- User stories are refined as needed; focus on giving an example for a small feature.

References

Core readings	Suggested readings
<ul style="list-style-type: none">• Jacobson, I., Spence, I., & Kerr, B. (2016). Use-case 2.0. <i>Communications of the ACM</i>, 59(5), 61–69.• “User Story vs Use Case for Agile Software Development”, Visual Paradigm	<ul style="list-style-type: none">• Jacobson, I., Spence, I., & Bittner, K. (2011). Use-Case 2.0 <i>The Guide o Succeeding with Use Cases</i>. [e-Book]• User story (VisualParadigm handbook)