# Métodos ágeis: o papel das *user stories*

Ilídio Oliveira

v2020/01/07, TP10

universidade de aveiro
departamento de eletrónica, telecomunicações e informática
deti

# Objetivos de aprendizagem

- Caracterizar os princípios da gestão do *backlog* em projetos ágeis
- Definir e escrever histórias para um determinado produto.
- Distinguir estimativa e priorização na gestão de histórias.
- Escreva os critérios de aceitação de uma história.
- Comparar as histórias e os caos de uso no que diz respeito a semelhanças e diferenças.
- Descreva o fluxo de trabalho de desenvolvimento baseado em histórias.

# Algumas ideias do desenvolvimento ágil

**QUICK LOOK**

**What is it?** Agile software engineering combines a philosophy and a set of development guidelines. The philosophy encourages customer satisfaction and early incremental delivery of software; small, highly motivated project teams; informal methods; minimal software engineering work products; and overall development simplicity. The development guidelines stress delivery over analysis and design (although these activities are not discouraged), and active and continuous communication between developers and customers.

**Who does it?** Software engineers and other project stakeholders (managers, customers, end users) work together on an agile team—a team that is self-organizing and in control of its own destiny. An agile team fosters communication and collaboration among all who serve on it.

**Why is it important?** The modern business environment that spawns computer-based systems and software products is fast-paced and ever-changing. Agile software engineering represents a reasonable alternative to conventional software engineering for certain classes of software and certain types of software projects. It has been demonstrated to deliver successful systems quickly.

**What are the steps?** Agile development might best be termed "software engineering lite." The basic framework activities—communication, planning, modeling, construction, and deployment—remain. But they morph into a minimal task set that pushes the project team toward construction and delivery (some would argue that this is done at the expense of problem analysis and solution design).

**What is the work product?** Both the customer and the software engineer have the same view—the only really important work product is an operational "software increment" that is delivered to the customer on the appropriate commitment date.

**How do I ensure that I've done it right?** If the agile team agrees that the process works, and the team produces deliverable software increments that satisfy the customer, you've done it right.

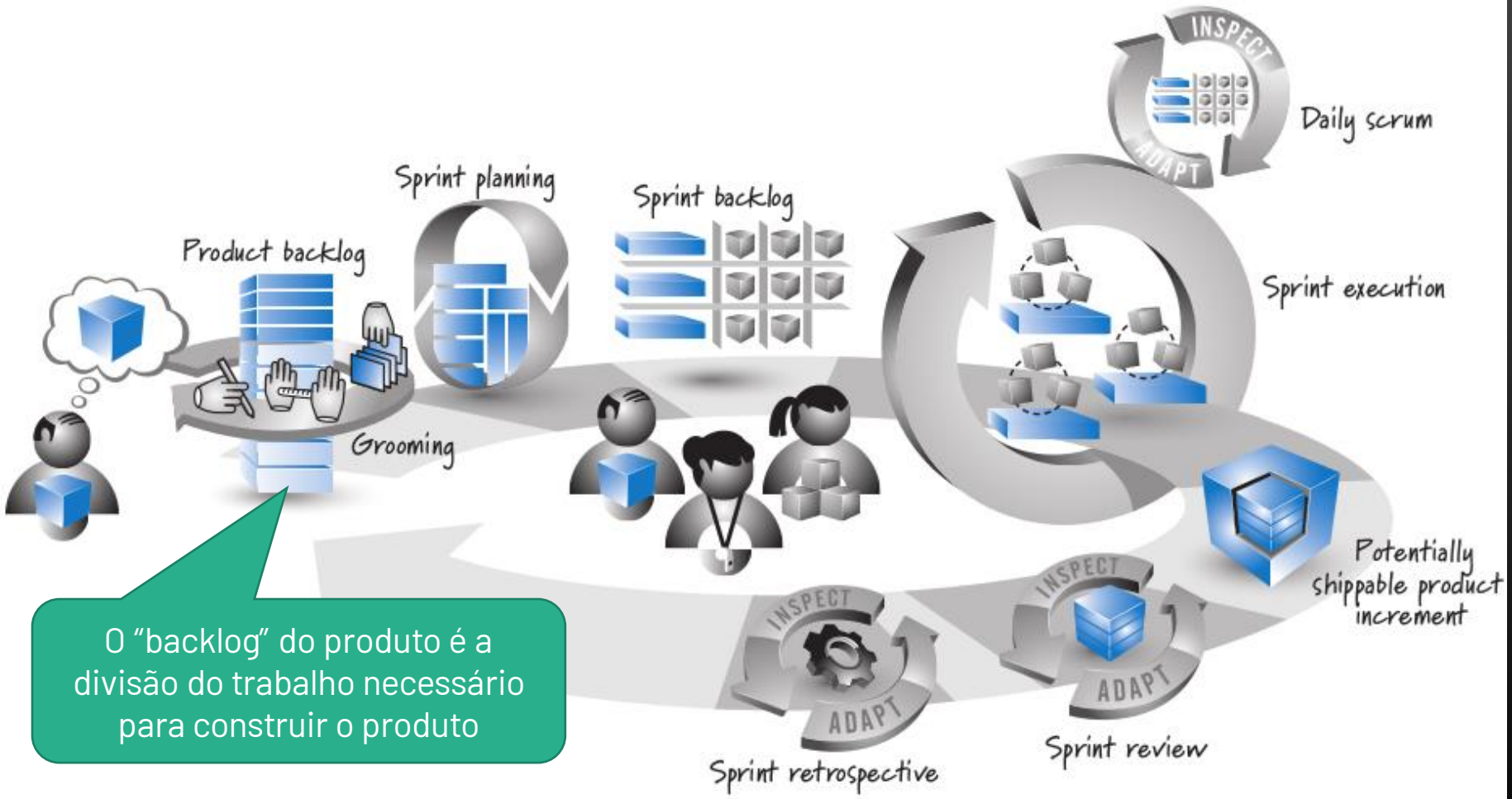
O dinamismo do mercado obriga a igual dinamismo das TIC/desenvolvimento. Especialmente quando os produtos do desenvolvimento passam a assumir um papel fundamental na criação das vantagens competitivas.

A transformação digital (competitiva) obriga a uma eng.a de software competitiva.

Credit: Pressmen'15
I Oliveira

# Atividades do Scrum

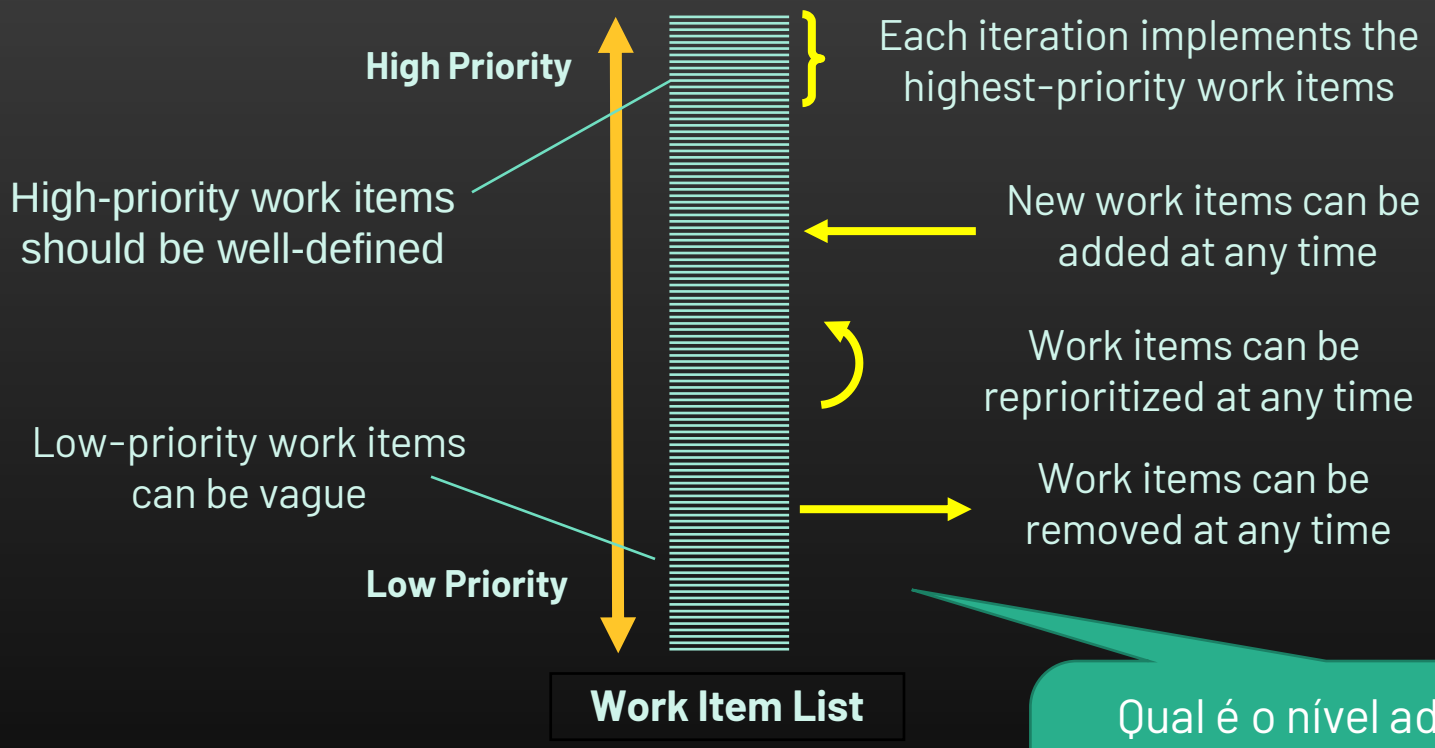
O Scrum não é um processo de desenv software, mas é frequentemente usado para a gestão de atividades da equipa




O "backlog" do produto é a divisão do trabalho necessário para construir o produto

FIGURE 2.3 Scrum framework

# "Regras" para gerir o *backlog* (à maneira ágil)




Qual é o nível adequado de detalhe para representar cada "item de trabalho"?
(a Scrum não impõe nenhum)

# Especificações evolutivas

Em vez de tomar um conjunto de decisões abrangente no início de um projeto, espalhamos a tomada de decisão ao longo da duração do projeto.

Para fazer isso, certificamo-nos de que temos um processo de trabalho que nos dá informação tão cedo e tão frequentemente quanto possível.

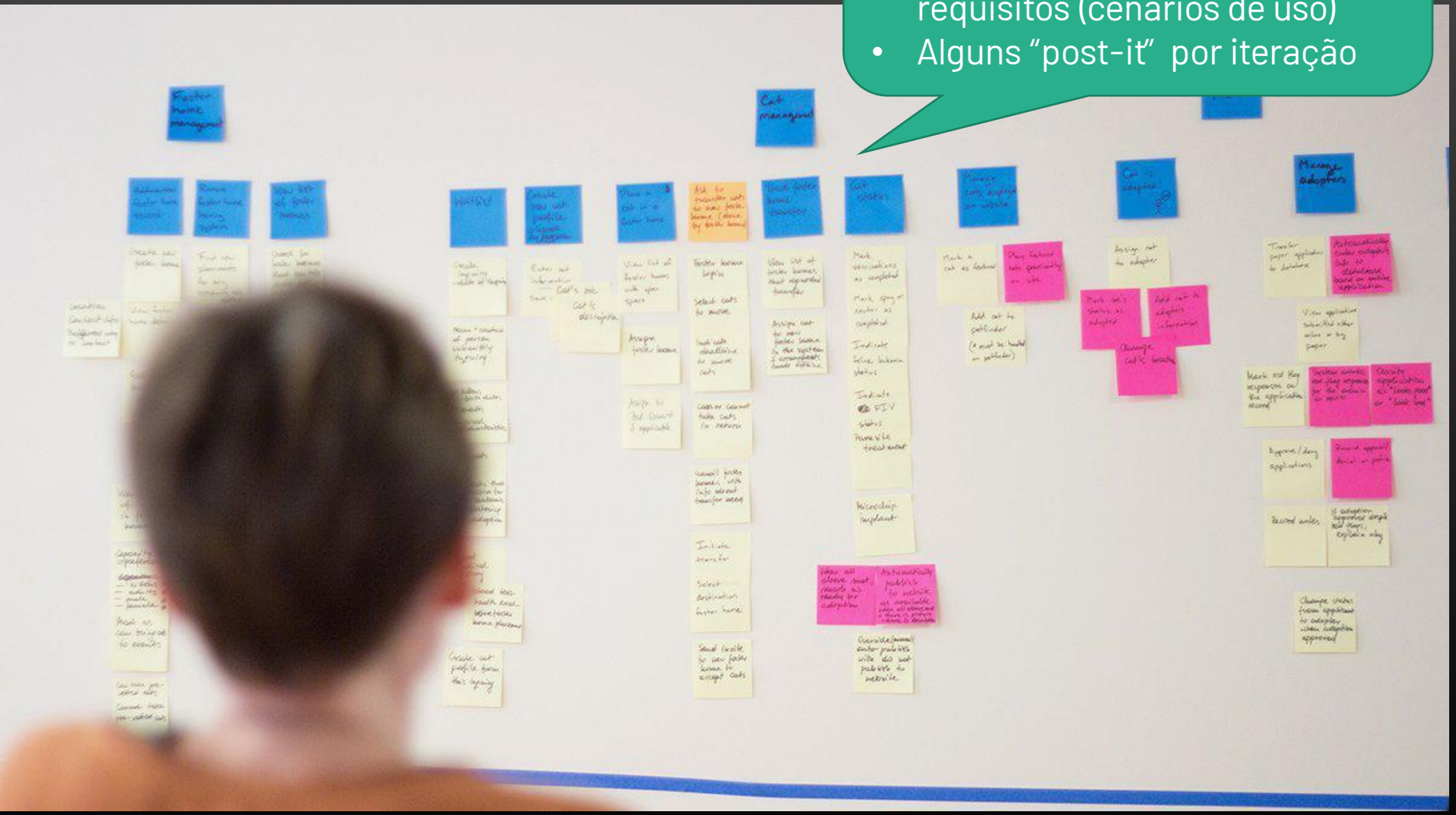
E é aqui que entram as histórias de utilização (user stories).

→ Podemos trabalhar com diferentes granularidades (versão "alto nível" e detalhar à media do necessário)

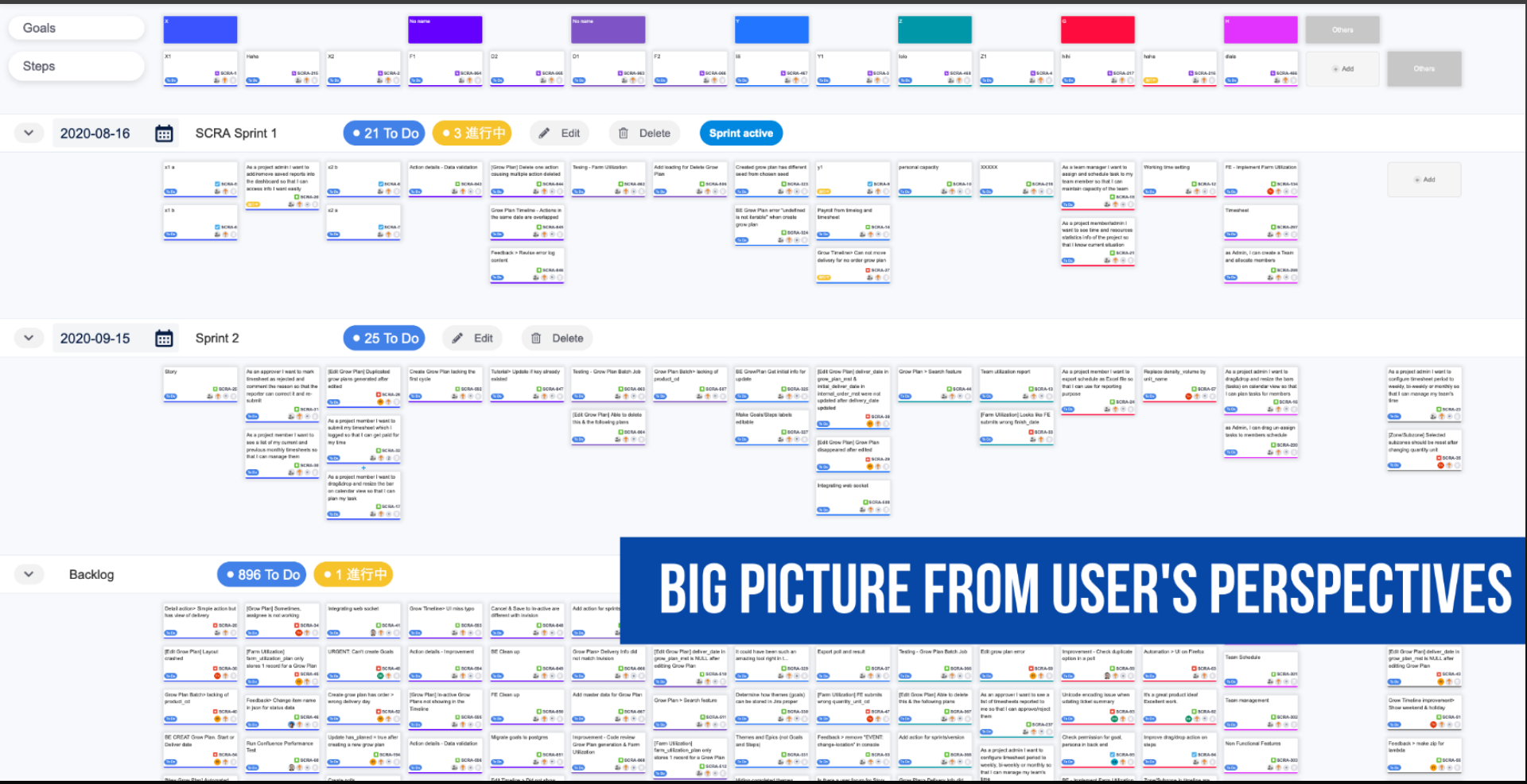
→ Há um sistema para recolher e incorporar informação.

# A metáfora do “post-it”

- Granularidade adequada para distribuir o trabalho
- Rastreabilidade para os requisitos (cenários de uso)
- Alguns “post-it” por iteração



# A metáfora do post-it numa ferramenta de planeamento ágil (Jira com plug-in)

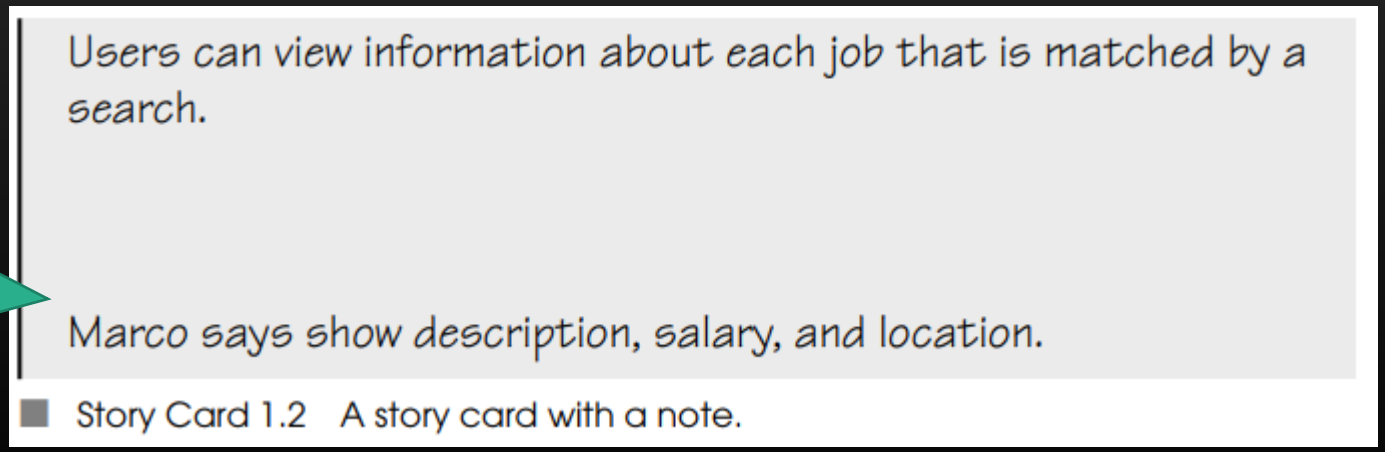



https://www.devsamurai.com/en/agile-user-story-mapping-for-jira/

# Proposta dos métodos ágeis: user stories

**User story** ("história"): uma "descrição curta e simples de uma funcionalidade do produto, contada do ponto de vista da pessoa que deseja essa nova capacidade, geralmente um utilizador ou promotor do sistema" (Cohn 2010)

Anotação informal do que é descoberto nas "conversas"




Story Card 1.2 A story card with a note.

# Exemplo

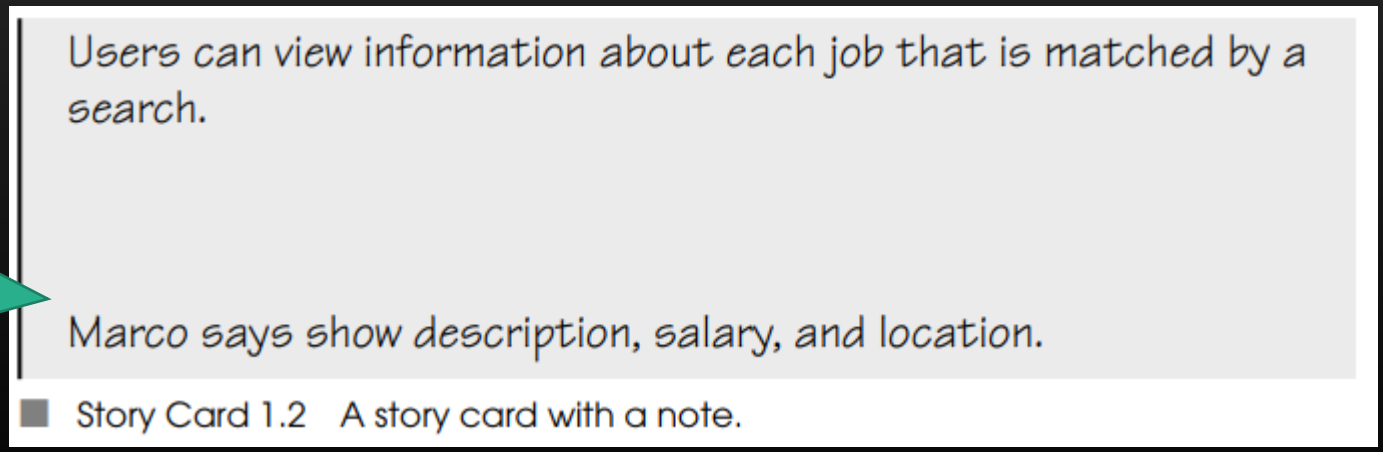
**Histórias adequada:**

- Uma empresa pode publicar novas oferta de emprego.
- Um Candidato pode limitar quem pode ver o seu currículo

**Histórias desadequadas:**

- O software será implementado em Python.
- O programa irá ligar-se à base de dados através de uma "*connection pool*" (reutilização de ligações já abertas)

Anotação informal do que é descoberto nas "conversas"




Story Card 1.2 A story card with a note.

# “Fatiar” os cenários de uso para tornar o trabalho mais concreto, gerível  e segmentado

A equipa de projeto e o cliente/promotor começam a discutir requisitos sobre as motivações de uso:

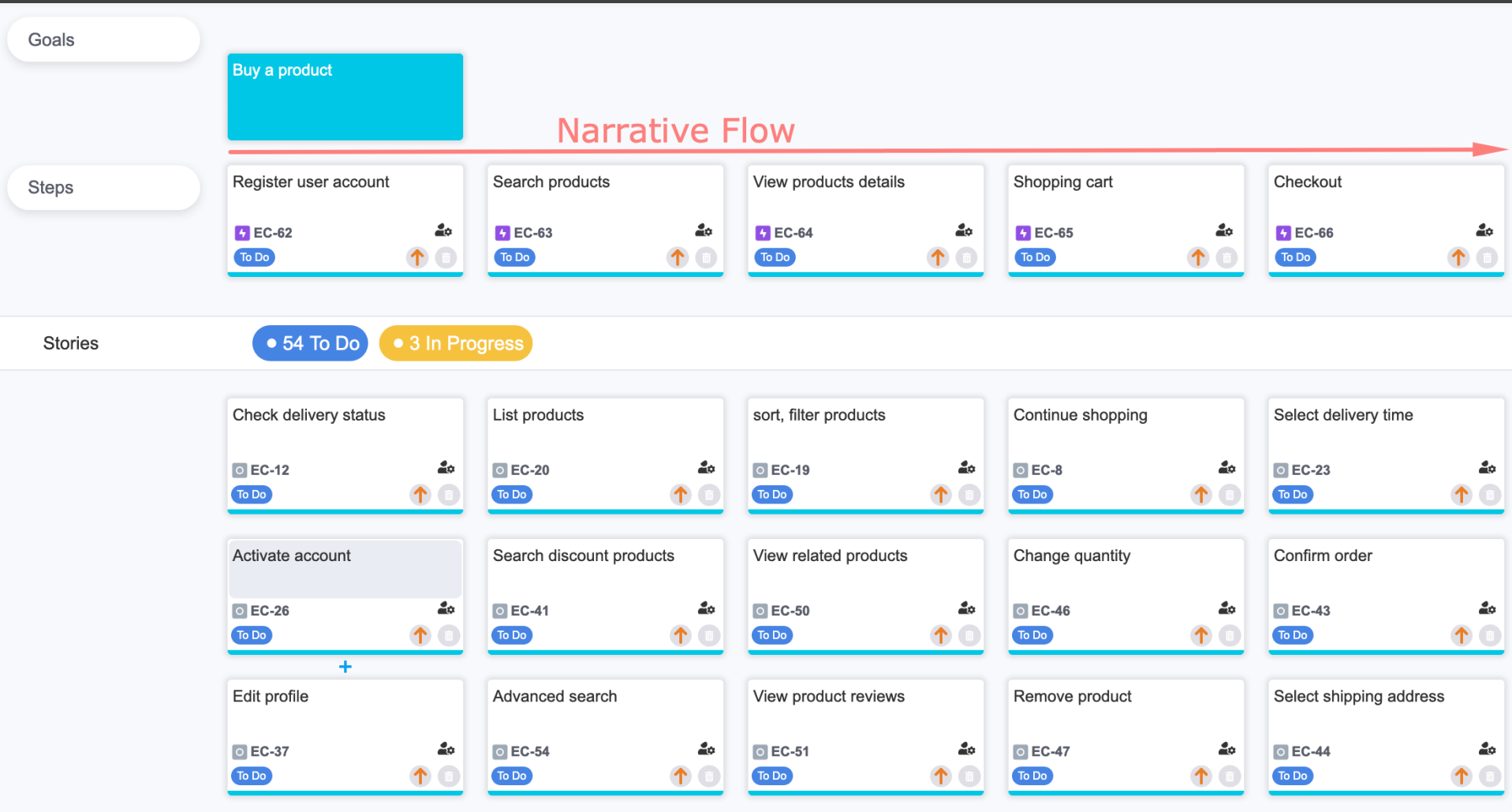
"Um Candidato (a um emprego) pode publicar um currículo (no site)".

Objetivo de alto nível ←→ caso de utilização.

Essa “história” será expandida à medida que os detalhes forem descobertos através de conversas / colaboração. →

Um possível desenvolvimento em histórias  (*user stories*):

- Um Candidato pode adicionar um novo currículo ao site.
- Um Candidato pode editar um currículo que já está no site.
- Um Candidato pode remover o currículo do local.
- Um Candidato pode marcar um currículo como inativo.
- Um Candidato pode marcar um currículo como escondido de certos empregadores.
- Um Candidato pode ver quantas vezes o seu currículo foi consultado




https://www.devsamurai.com/en/agile-user-story-mapping-for-jira/

# Boas ou más histórias?

a The user can run the system on Windows XP and Linux.

b All graphing and charting will be done using a third-party library.

c The user can undo up to fifty commands.

d The software will be released by June 30.

e The software will be written in Java.

f The user can select her country from a drop-down list.

g The system will use Log4J to log all error messages to a file.

h The user will be prompted to save her work if she hasn't saved it for 15 minutes.

i The user can select an "Export to XML" feature.

j The user can export data to XML.

# Pode-se usar um *template* para apresentar a história

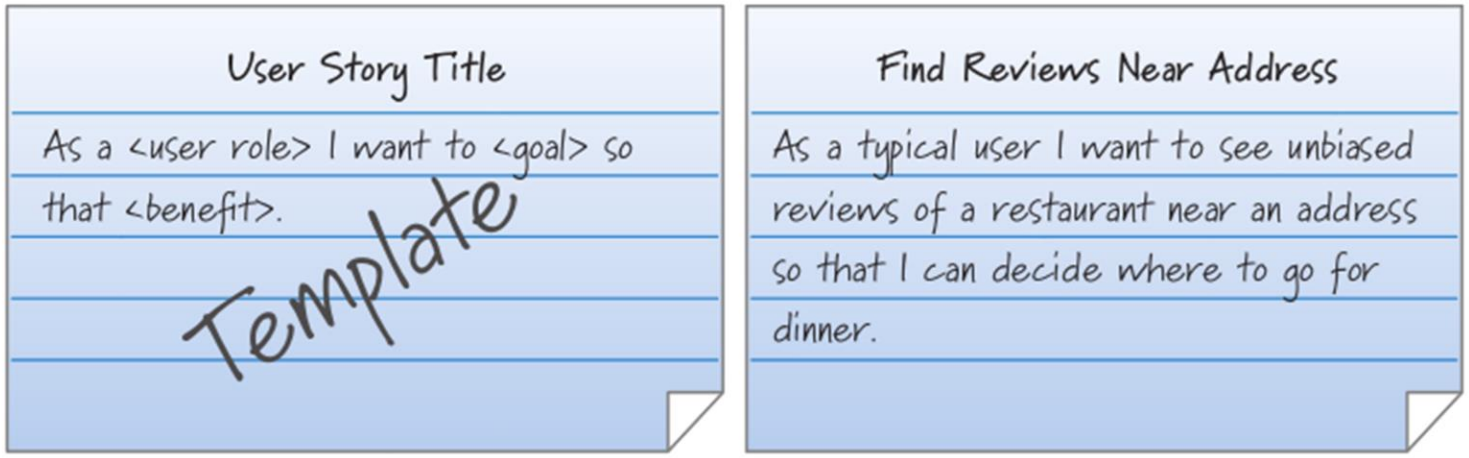
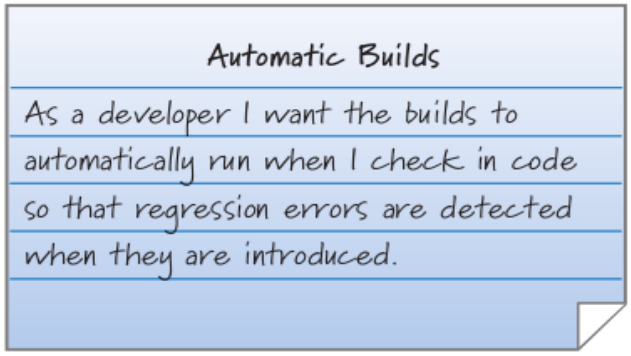



**FIGURE 5.2** A user story template and card

As histórias devem conter um benefício percetível para o utilizador!



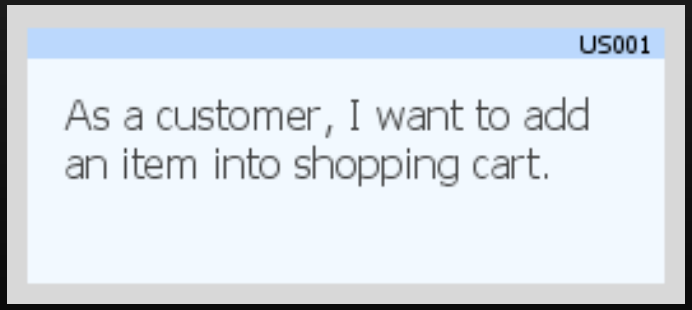

Undesirable technical story

# Estratégia para redigir a história

As histórias dos utilizadores são frequentemente escritas de acordo com a seguinte estrutura (mas há outros estilos):

Sendo <papel de utilizador>, quero <ação/funcionalidade pretendida> de modo a <satisfação obtida>

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




→ ver exemplos

E.g.:

Sendo um cliente, quero receber um SMS quando o artigo chegar de modo a que eu possa ir buscá-lo.

<role> representa a pessoa, o sistema, o subsistema ou qualquer outra entidade que interaja com o sistema a ser implementado para atingir um objetivo. É quem obtém valor da utilização do sistema.

<business objetive> representa uma expectativa de um utilizador sobre algo que pode realizar interagindo com o sistema.

<benefício> representa o valor resultante por da interação com o sistema. Pode ser omitido, se for óbvio (decorrente do ponto anterior).

# Epic (épico): “grande” objetivo do utilizador

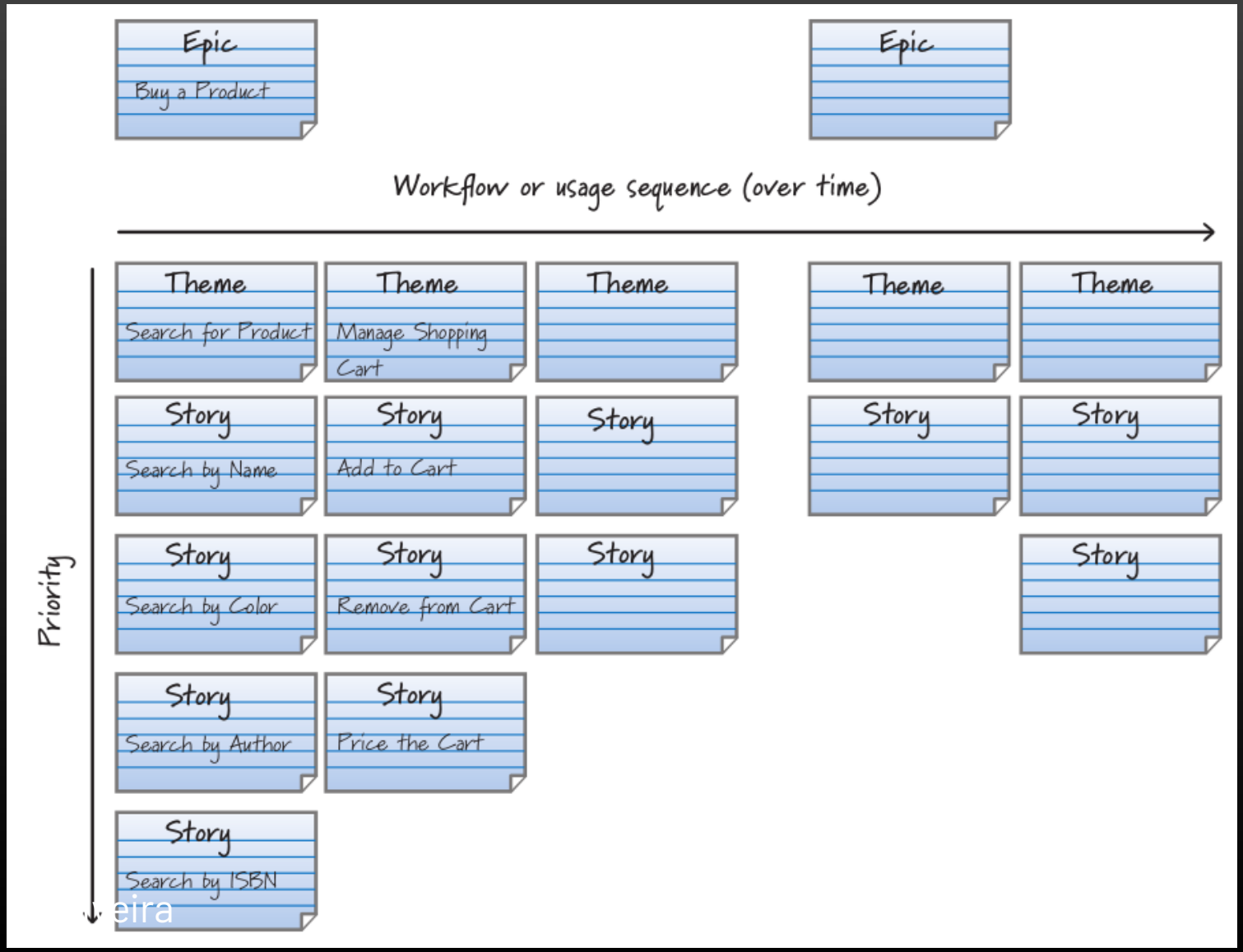
Quando uma história é muito “grande” (apresentada em alto nível), às vezes é referida como um épico.

Os Épicos podem ser divididos em várias histórias de tamanho menor.

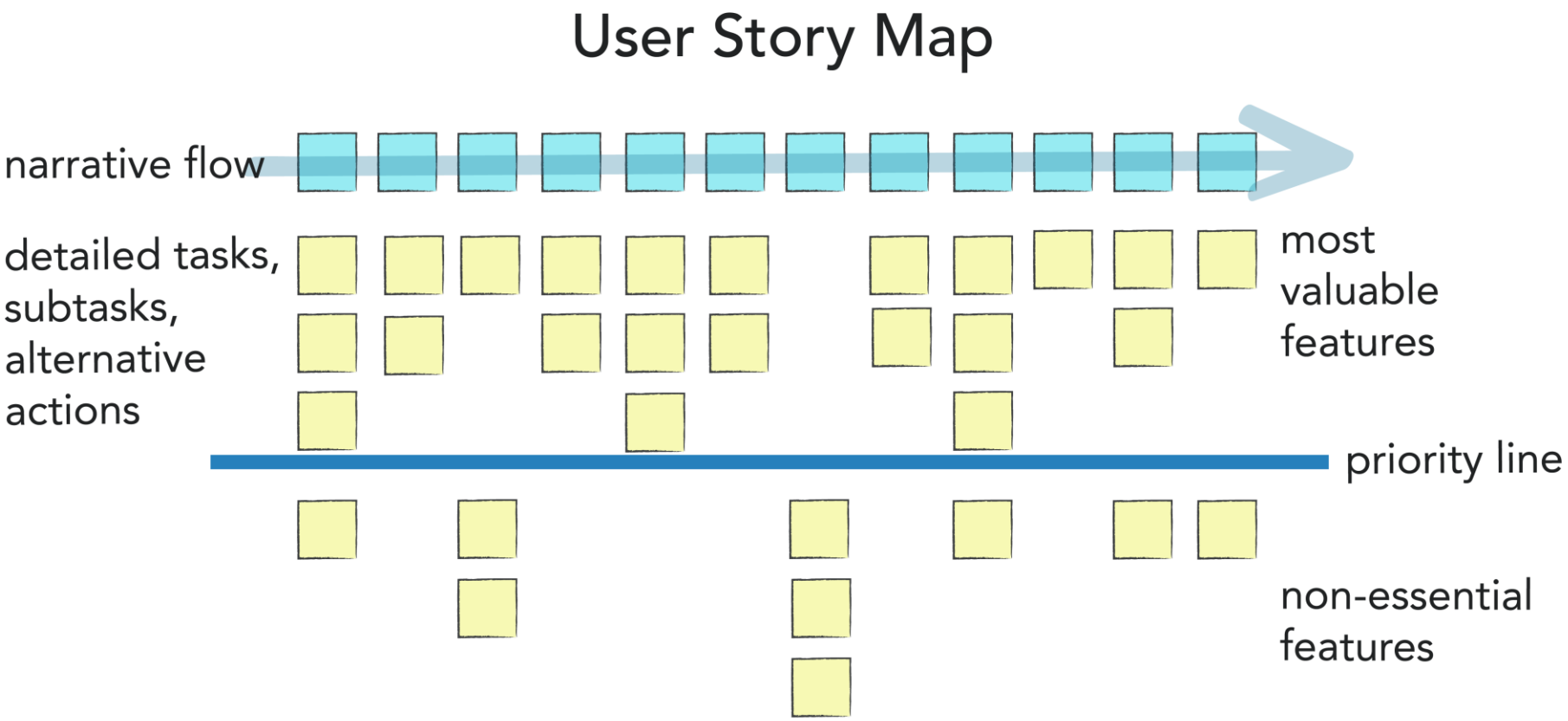
Por exemplo, o épico "Um utilizador pode usar o site para procurar um emprego" poderia ser dividido em várias histórias:

- Um utilizador pode procurar empregos por atributos como localização, intervalo salarial, designação da oferta, nome da empresa, e a data em que o trabalho foi postado.
- Um utilizador pode visualizar informações detalhadas sobre cada oportunidade  que seja encontrada numa pesquisa.
- Um utilizador pode ver informações detalhadas sobre uma empresa que publicou um trabalho.

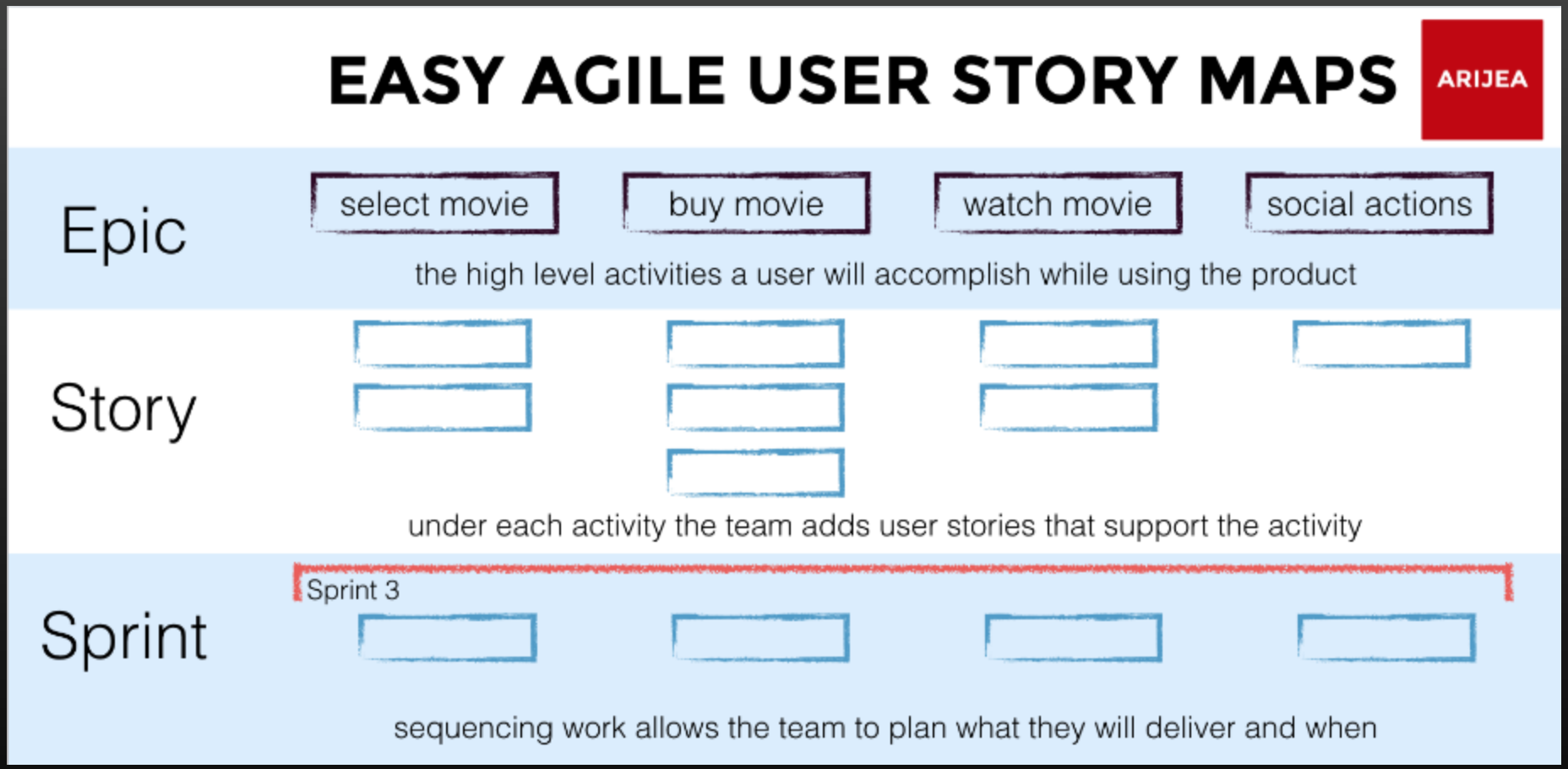
# À procura das histórias



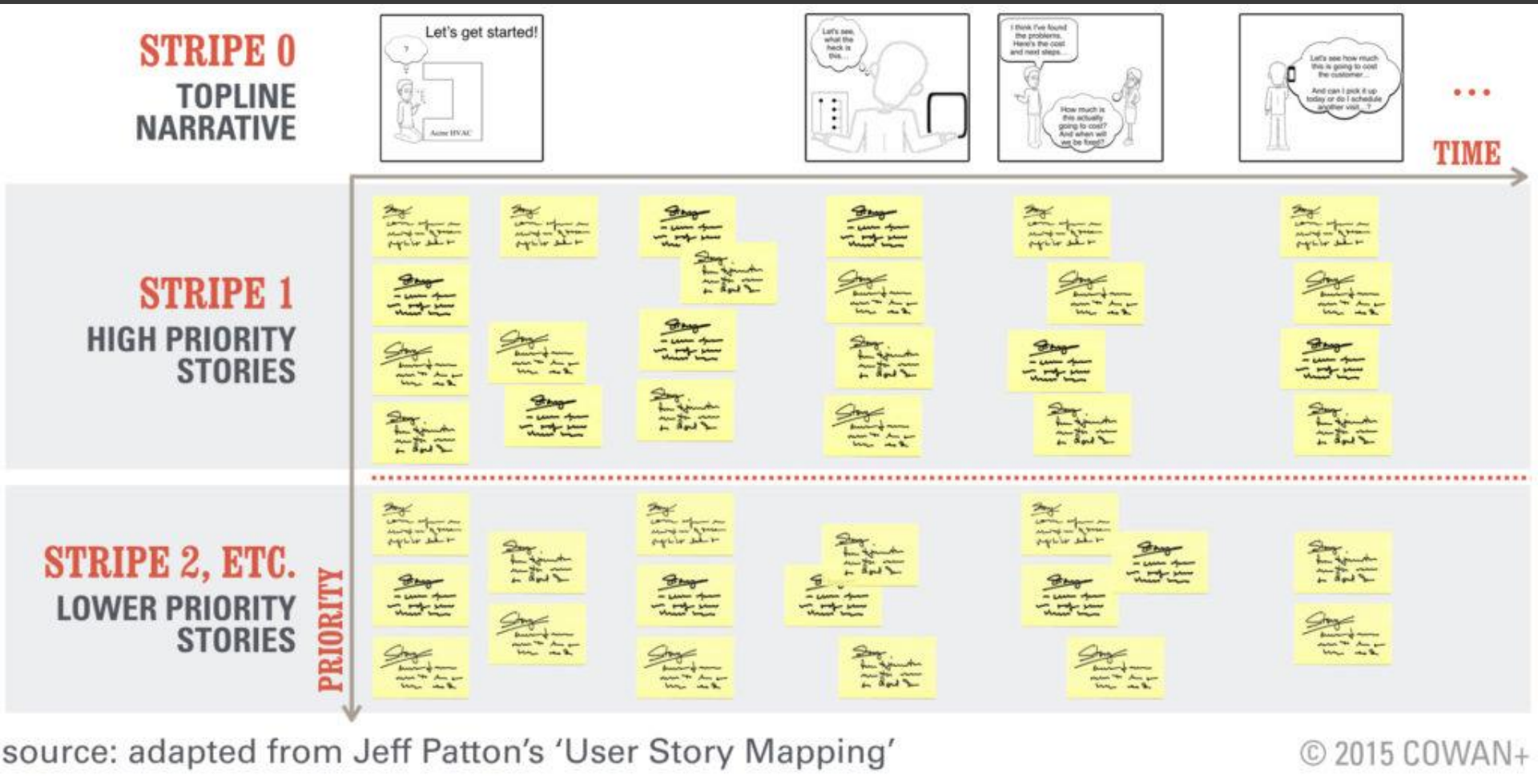
# O "mapa das histórias"




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

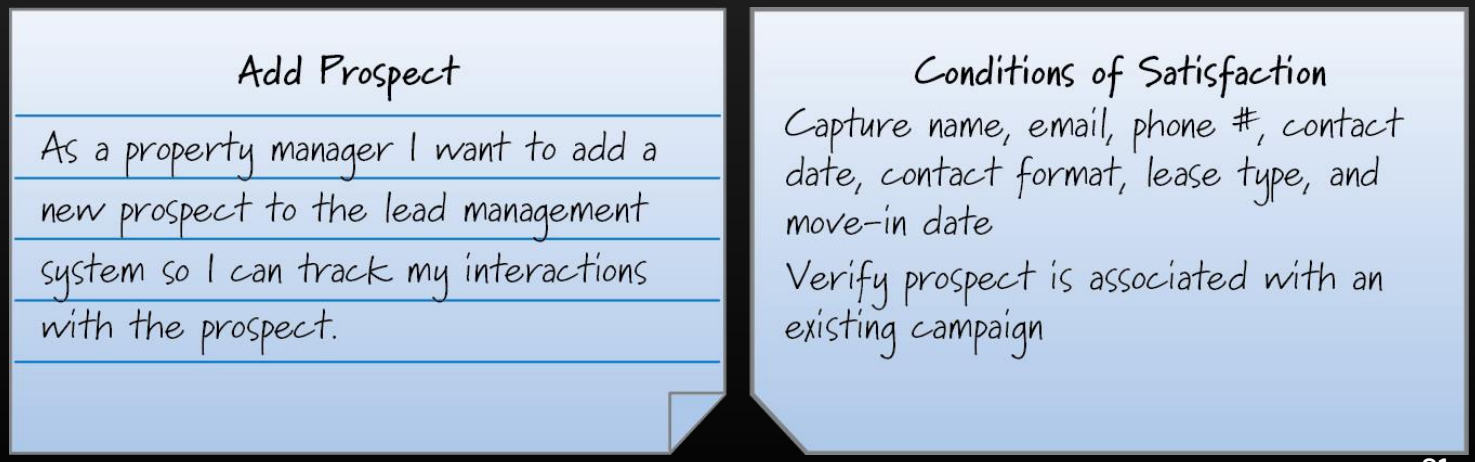

EASY AGILE USER STORY MAPS
ARIJEA
Epic
select movie
buy movie
watch movie
social actions
the high level activities a user will accomplish while using the product
Story
under each activity the team adds user stories that support the activity
Sprint
Sprint 3
sequencing work allows the team to plan what they will deliver and when

# Organização das histórias em níveis de prioridade (linhas de corte para as iterações)



# As histórias também fornecem um contexto para "anotar" as condições de aceitação/pontos de verificação

As equipas "ágeis" incluem na história num conjunto de condições que descrevem as "condições de satisfação" da história, isto é, o que tem de passar para a história poder ser aceite.



# As critérios de aceitação podem ser escritos seguindo um formato estruturado

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

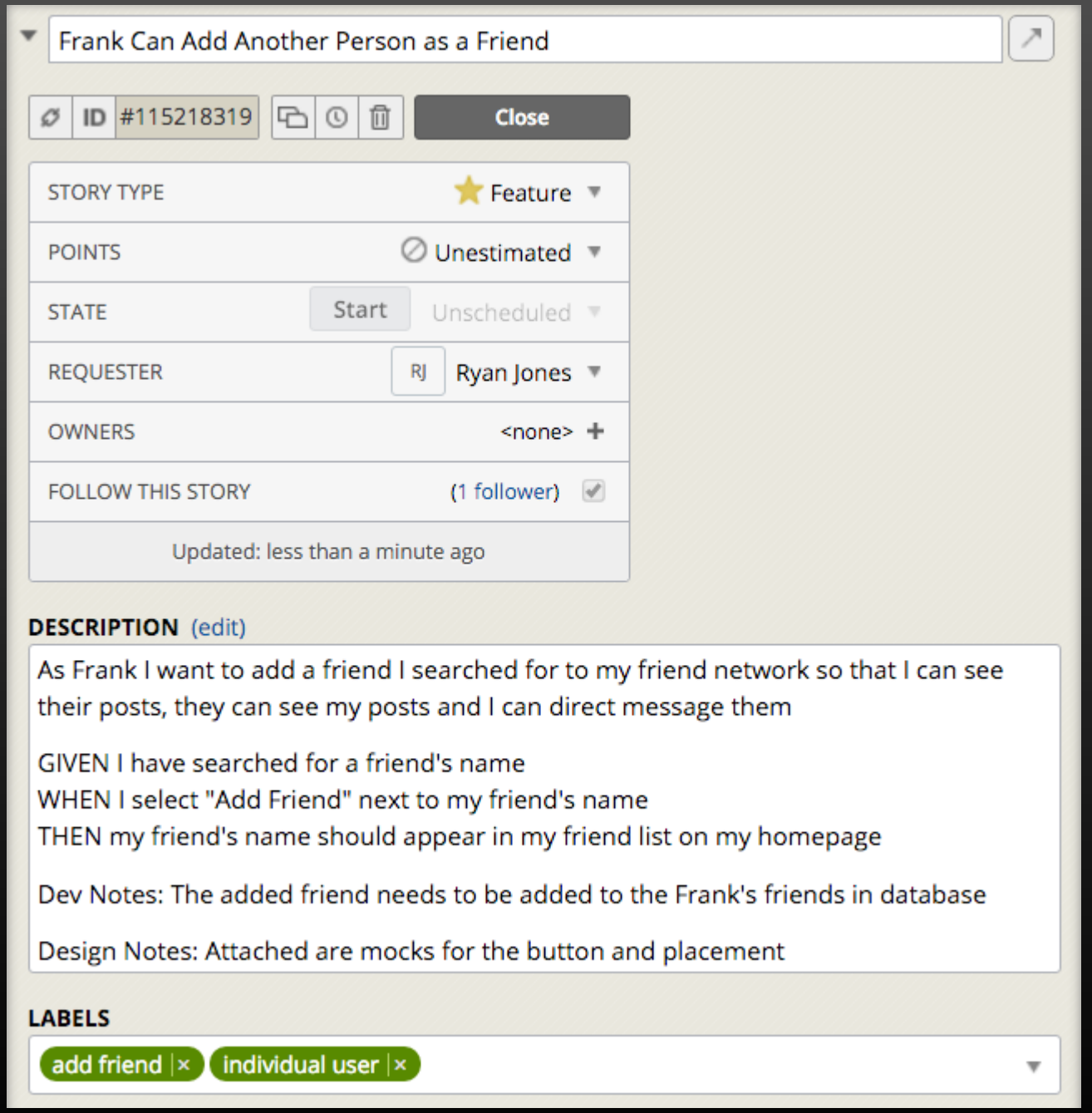
```
Title (one line describing the story)

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: ...
```


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

PivotalTracker

# 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

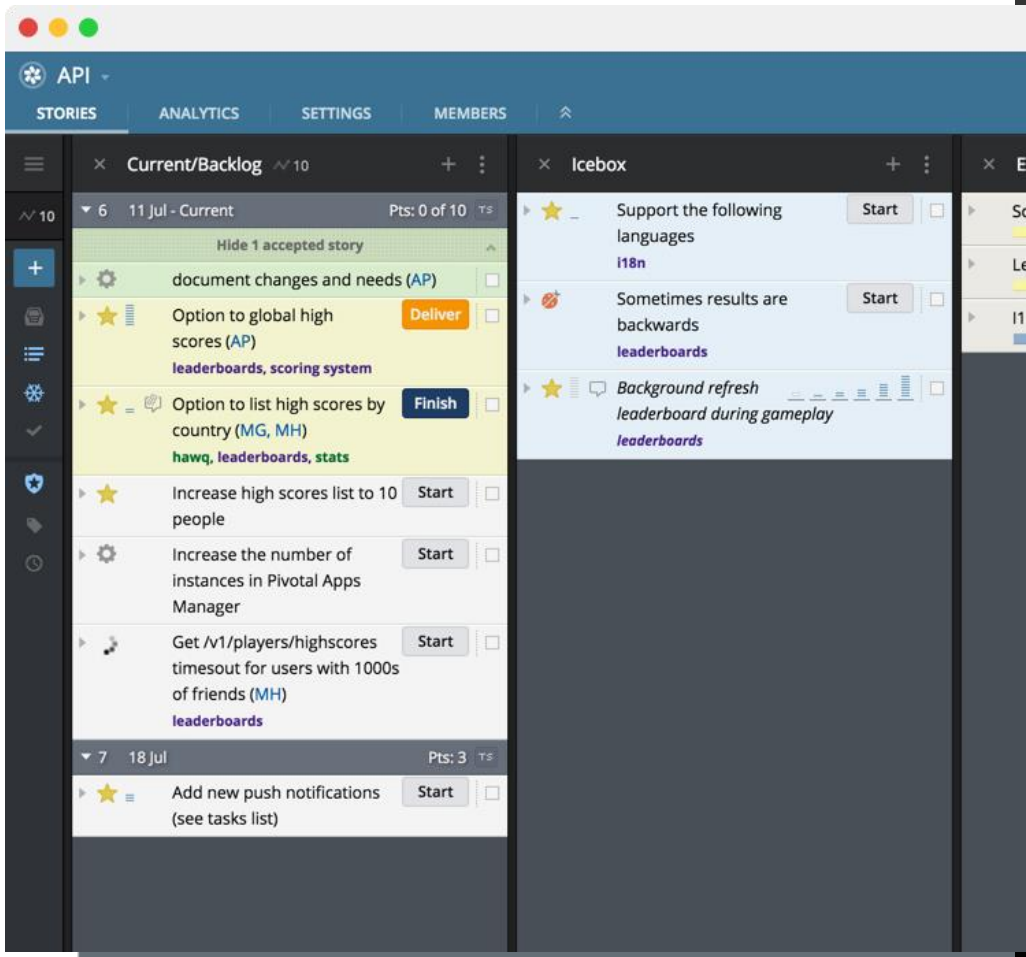
A story is a small, actionable bit of work that's either a placeholder for a future conversation or a reflection of one that already happened. Outlining what a user needs helps you focus on the what, not the how.

### Define the story

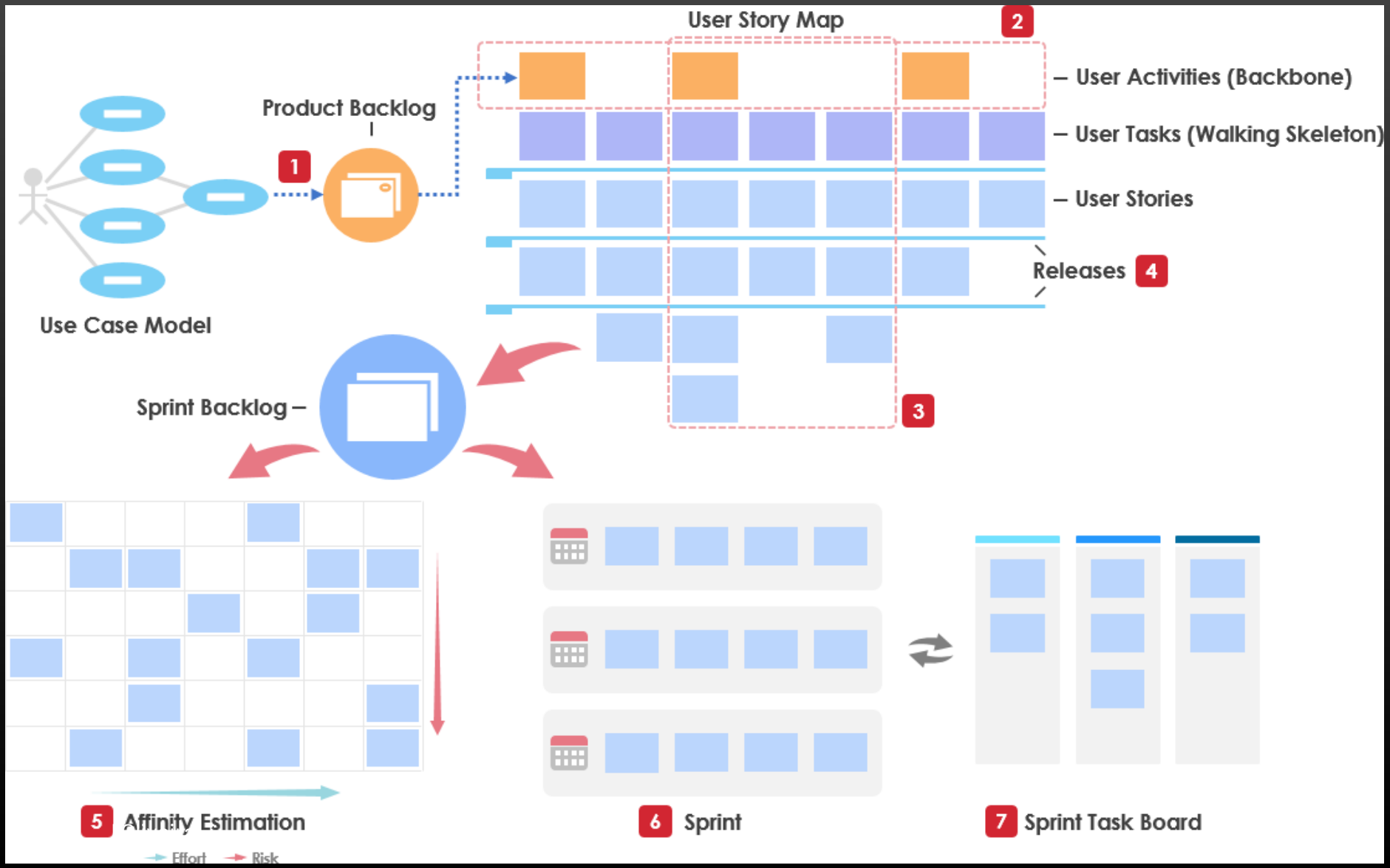
Select among features, bugs, and chores to strike a healthy balance between building new features, staying ahead of technical debt, and keeping the bugs from piling up.

### 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.



# Agile in VisualParadigm



# Determinação de requisitos explorando cenários centrados no utilizador

A) *Use cases* (casos de utilização)

B) *User stories* (histórias)

C) *User-centered design, UCD* (Desenho centrado no utilizador)

D) *Customer Journey Map/Experience maps* (Mapas de experiência)

Não falámos de todos, em MAS.

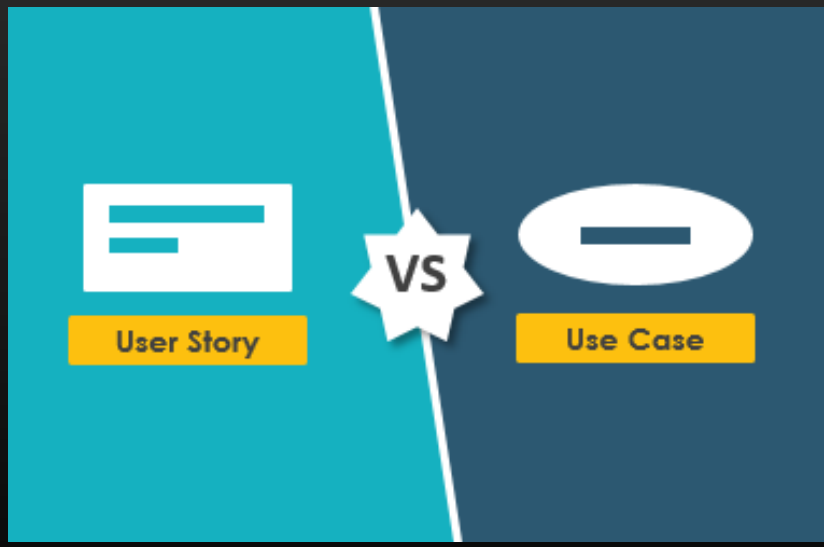
# Relembrar: casos de utilização e resultados associados

Um caso de utilização descreve uma sequência de interações entre um sistema e um ator externo da qual o ator obtém resultado de valor (para as suas motivações).

Os nomes dos casos de uso são sempre escritos a forma de um verbo seguido por um objeto.

O caso de utilização é complementado com uma descrição detalhada (seguindo um padrão/narrativa estruturada)

Um caso de utilização inclui um fluxo principal e variantes.



| | | | |
|---|---|---|---|
| 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.<br>PRE-2. User is authorized to request chemicals.<br>PRE-3. Chemical inventory database is online. | | |
| Postconditions: | POST-1. Request is stored in the CTS.<br>POST-2. Request was sent to the Chemical Stockroom or to a Buyer. | | |
| Normal Flow: | **4.0 Request a Chemical from the Chemical Stockroom**<br>1. Requester specifies the desired chemical.<br>2. System lists containers of the desired chemical that are in the chemical stockroom, if any.<br>3. System gives Requester the option to View Container History for any container.<br>4. Requester selects a specific container or asks to place a vendor order (see 4.1).<br>5. Requester enters other information to complete the request.<br>6. System stores the request and notifies the Chemical Stockroom. | | |
| Alternative Flows: | **4.1 Request a Chemical from a Vendor**<br>1. Requester searches vendor catalogs for the chemical (see 4.1.E1).<br>2. System displays a list of vendors for the chemical with available container sizes, grades, and prices.<br>3. Requester selects a vendor, container size, grade, and number of containers.<br>4. Requester enters other information to complete the request.<br>5. System stores the request and notifies the Buyer. | | |
| Exceptions: | **4.1.E1 Chemical Is Not Commercially Available**<br>1. System displays message: No vendors for that chemical.<br>2. System asks Requester if he wants to request another chemical (3a) or to exit (4a).<br>3a. Requester asks to request another chemical.<br>3b. System starts normal flow over.<br>4a. Requester asks to exit.<br>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 | | |

# As histórias podem ser apresentadas num nível de abstração próximo do caso de utilização

**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. |

# Mais frequentemente, a história é um desdobramento do caso de utilização

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.

# 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.

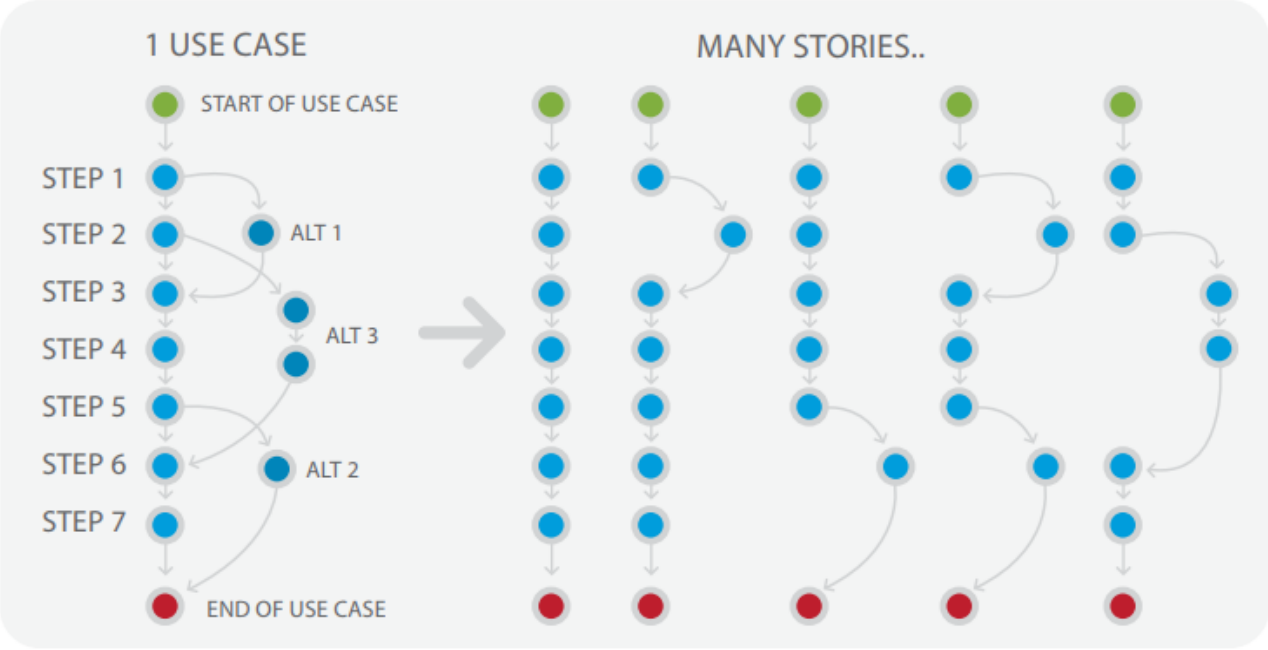
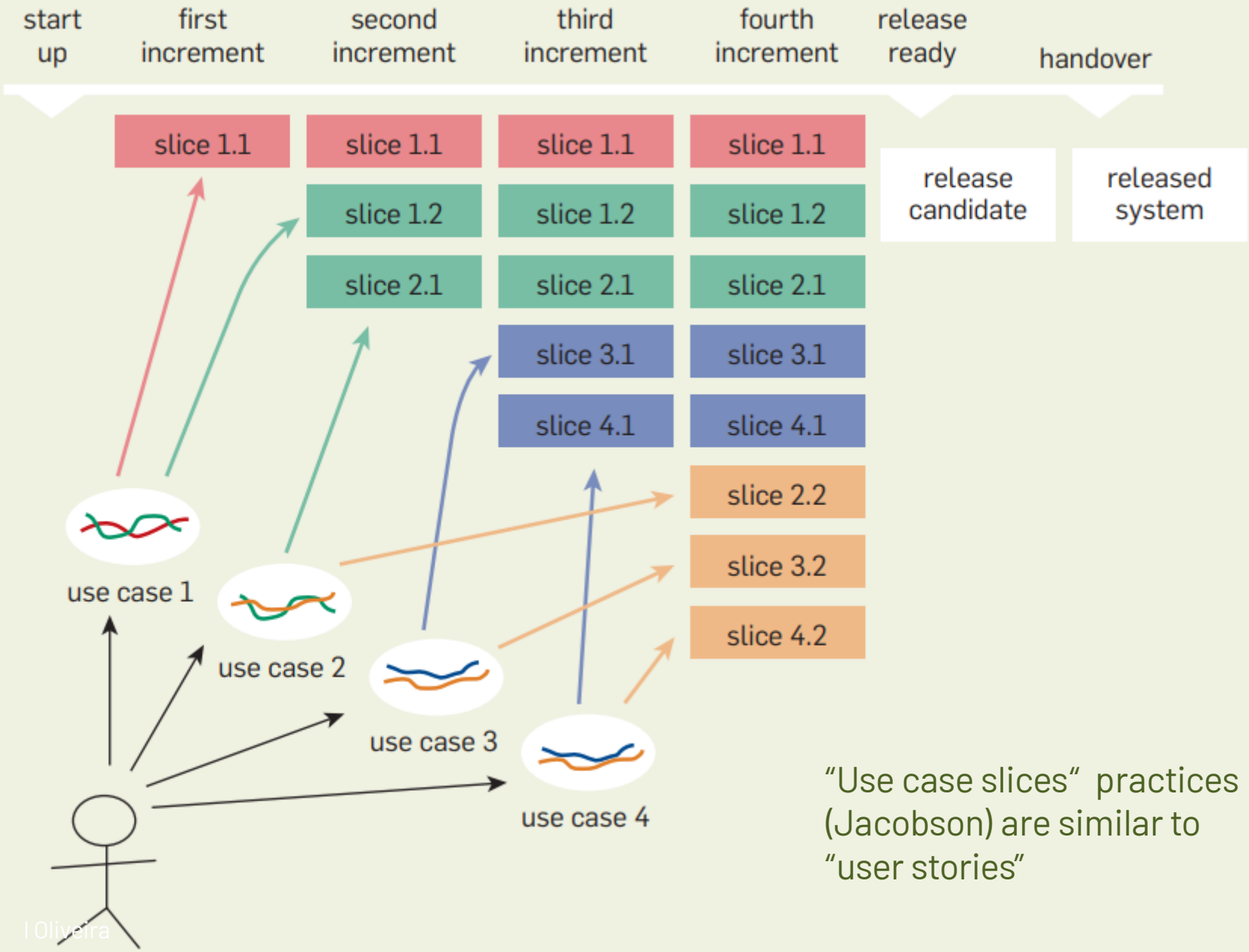


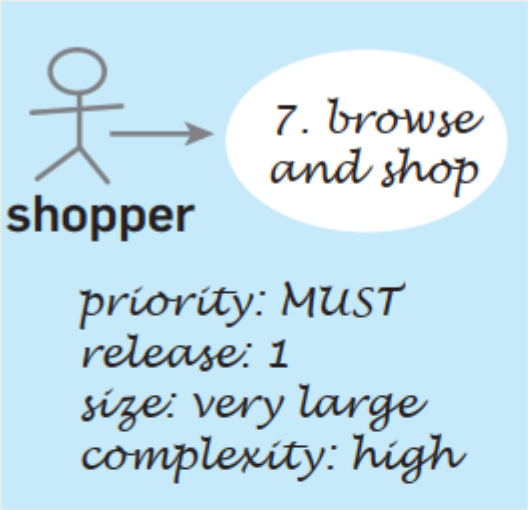

**FIGURE 8:**
**THE RELATIONSHIP BETWEEN THE FLOWS AND THE STORIES**

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

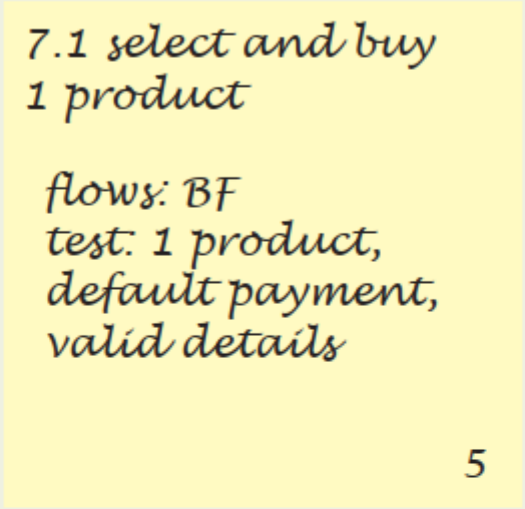


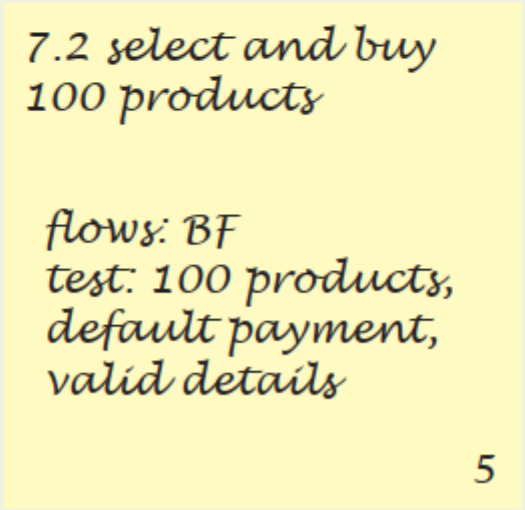

"Use case slices" practices (Jacobson) are similar to "user stories"

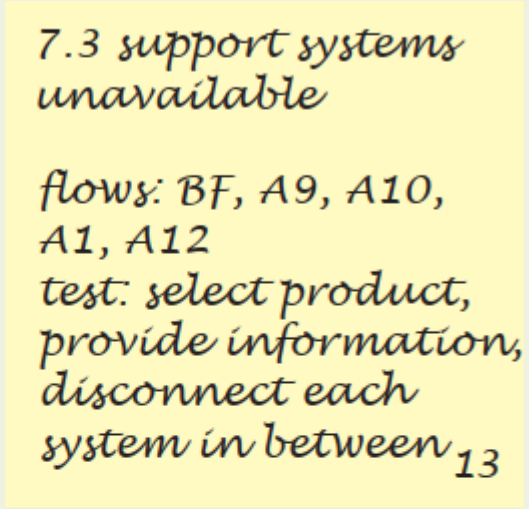
**Figure 5. Capturing the properties of a use case and its slices using sticky notes.**




a use case and its properties captured on a sticky note



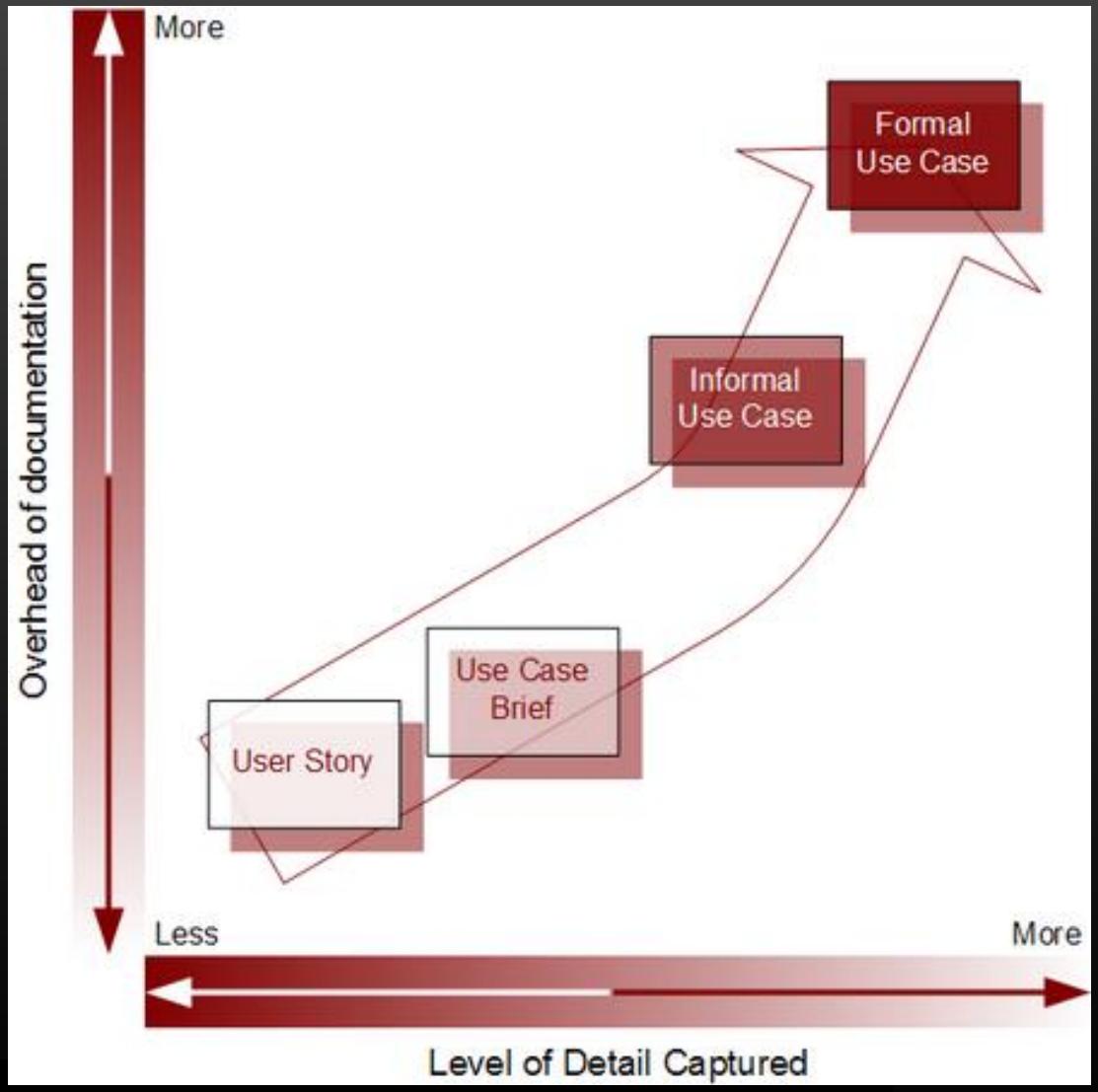



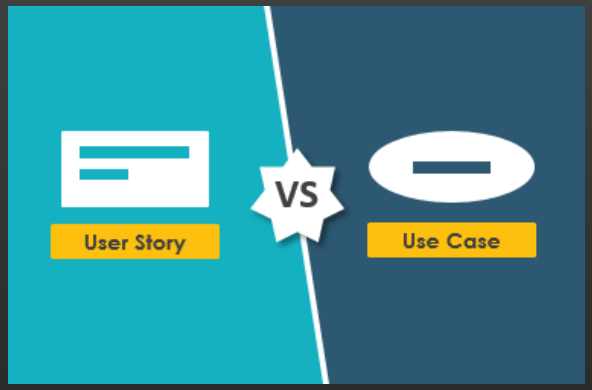




some slices from the use case captured on their own sticky notes

| Em comum | Próprio dos casos de utilização | Próprio das histórias |
|---|---|---|
| • Ambos seguem uma abordagem centrada na utilização<br>• Ambos são contextos para descrever o diálogo utilizadores/sistema<br>• Ambos resultam em casos de teste que representam os critérios de aceitação<br>• Ambos podem ser estimados | • Visão geral para ajudar a entender a extensão do sistema e o seu valor<br>• Descreve como o utilizador imagina a interação com o sistema para atingir os seus objetivos.<br>• Fornecer à equipa do projeto uma estrutura e contexto que falta à coleção das histórias<br>• Pode examinar cada elemento do caso de utilização (fluxos, pré-condições, pós-condições, e assim por diante) para procurar requisitos funcionais e não funcionais pertinentes e para definir testes (ajuda a evitar que se ignorem requisitos.) | • Declaração concisa das necessidades de um utilizador<br>• Existe um acesso facilitado a especialistas do domínio (refinar a história conforme necessário)<br>• Mais adequado para funcionar como um item do *backlog* para o dia-a-dia (Scrum, Kanban)<br>• Critérios de aceitação explícitos |

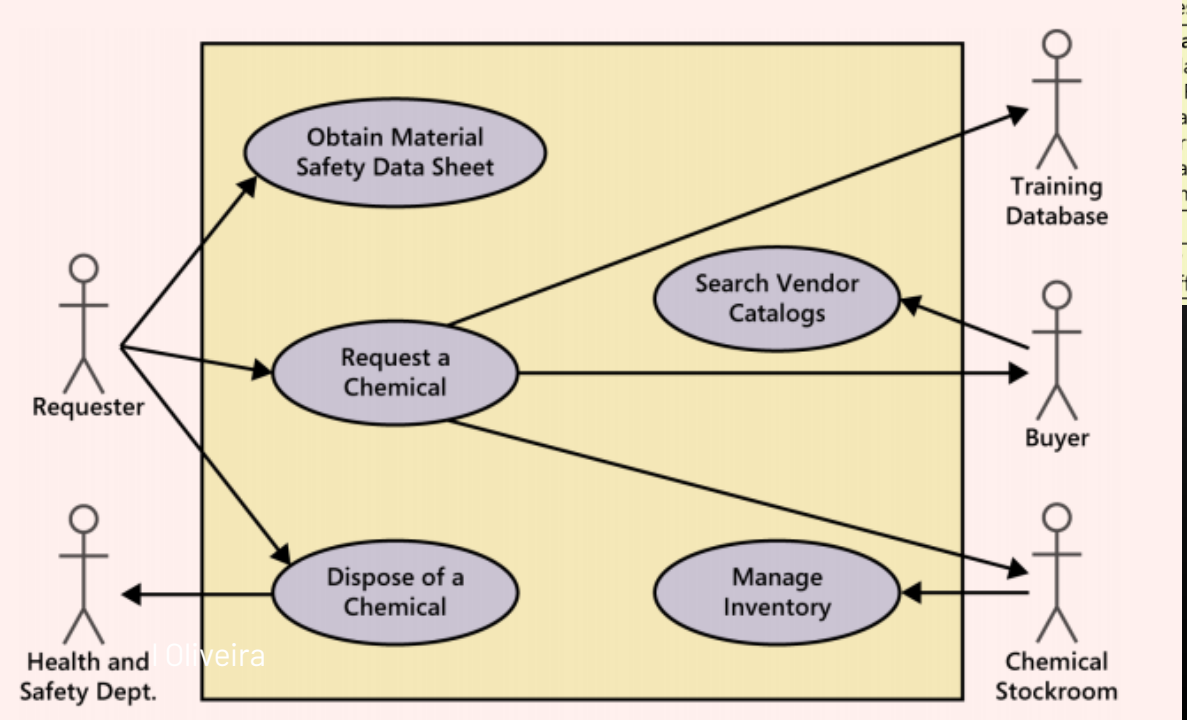
# Posicionamento relative dos casos de utilização e histórias



# Casos de utilização

| | | | |
|---|---|---|---|
| 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.<br>PRE-2. User is authorized to request chemicals.<br>PRE-3. Chemical inventory database is online. | | |
| Postconditions: | POST-1. Request is stored in the CTS.<br>POST-2. Request was sent to the Chemical Stockroom or to a Buyer. | | |
| Normal Flow: | **4.0 Request a Chemical from the Chemical Stockroom**<br>1. Requester specifies the desired chemical.<br>2. System lists containers of the desired chemical that are in the chemical stockroom, if any.<br>3. System gives Requester the option to View Container History for any container.<br>4. Requester selects a specific container or asks to place a vendor order (see 4.1).<br>5. Requester enters other information to complete the request.<br>6. System stores the request and notifies the Chemical Stockroom. | | |
| Alternative Flows: | **4.1 Request a Chemical from a Vendor**<br>1. Requester searches vendor catalogs for the chemical (see 4.1.E1).<br>2. System displays a list of vendors for the chemical with available container sizes, grades, and prices.<br>3. Requester selects a vendor, container size, grade, and number of containers.<br>4. Requester enters other information to complete the request.<br>…s the request and notifies the Buyer. | | |
| | **…al Is Not Commercially Available**<br>…ays message: No vendors for that chemical.<br>…Requester if he wants to request another chemical (3a) or to exit (4a).<br>…asks to request another chemical.<br>…rts normal flow over.<br>…asks to exit.<br>…minates use case. | | |
| | …5 times per week by each chemist, 200 times per week by chemical<br>…f | | |



# Histórias

Users can view information about each job that is matched by a search.

Marco says show description, salary, and location.

Story Card 1.2 A story card with a note.

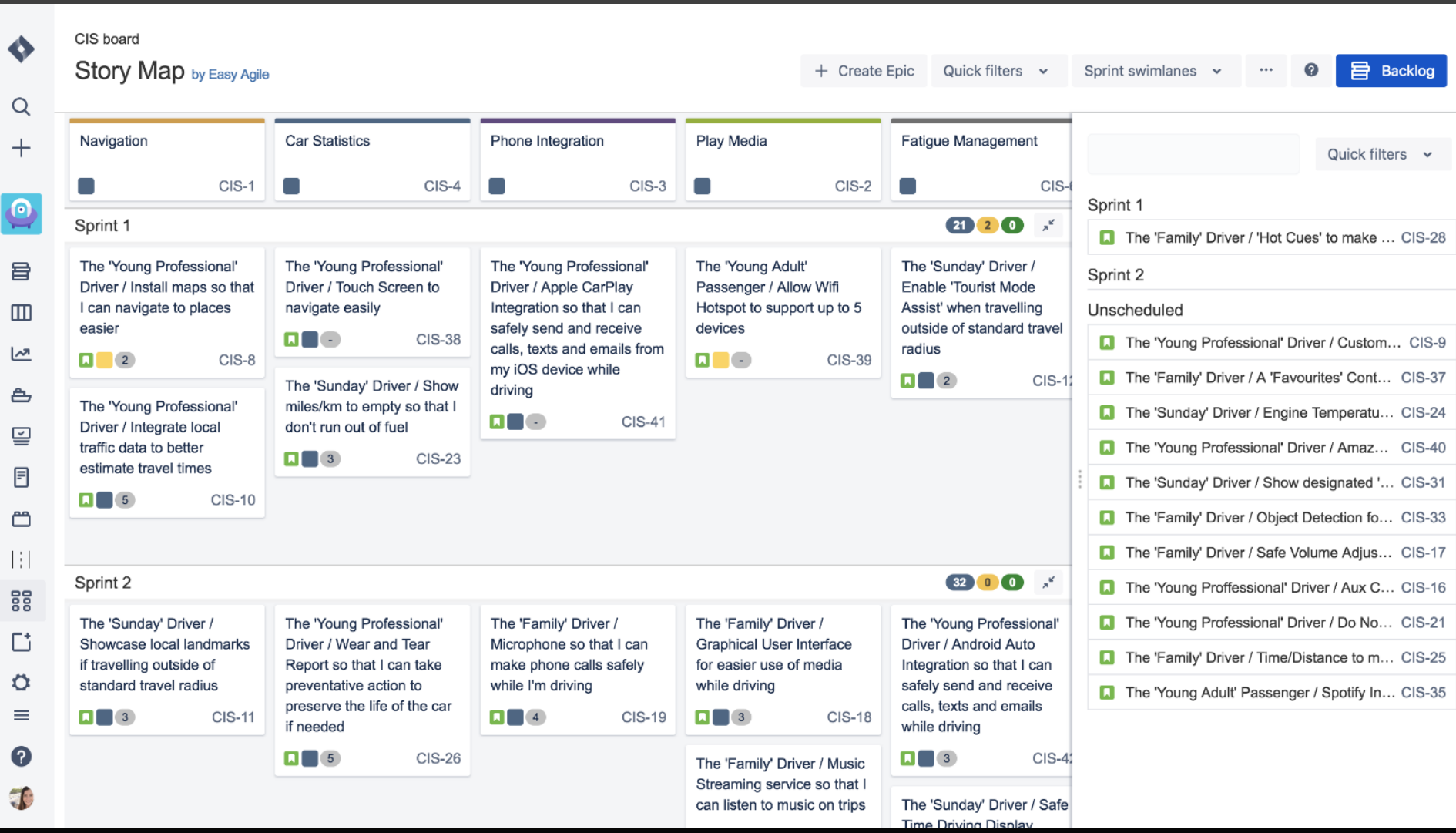
Try it with an empty job description.
Try it with a really long job description.
Try it with a missing salary.
Try it with a six-digit salary.

Story Card 1.3 The back of a story card holds reminders about how to test the story.

**Table 1.3** *Splitting a story to create a better release plan.*

| Iteration | Stories | Story Points |
|---|---|---|
| Iteration 1 | A, B, C | 13 |
| Iteration 2 | D, E, F | 12 |
| Iteration 3 | G, H, Y | 13 |
| Iteration 4 | J, Z | 4 |

# Histórias


CIS board
Story Map by Easy Agile
+ Create Epic
Quick filters
Sprint swimlanes
Backlog
Navigation
Car Statistics
Phone Integration
Play Media
Fatigue Management
CIS-1
CIS-4
CIS-3
CIS-2
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 / Integrate local traffic data to better estimate travel times
CIS-10
The 'Young Professional' Driver / Touch Screen to navigate easily
CIS-38
The 'Sunday' Driver / Show miles/km to empty so that I don't run out of fuel
CIS-23
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
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 'Family' Driver / Music Streaming service so that I can listen to music on trips
The 'Young Professional' Driver / Android Auto Integration so that I can safely send and receive calls, texts and emails while driving
The 'Sunday' Driver / Safe
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 Proffessional' 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

# Os *story points* são usados para contruir o *burndown chart*

# Benefits of usage-centric requirements

The power of both use cases and user stories comes from their user-centric and usage-centric perspective. The users will have clearer expectations of what the new system will let them do than if you take a feature-centric approach. The customer representatives on several Internet development projects found that use cases clarified their notions of what visitors to their websites should be able to do. Use cases help BAs and developers understand the user's business. Thinking through the actor-system dialogs reveals ambiguity and vagueness early in the development process, as does generating tests from the use cases.

Overspecifying the requirements up front and trying to include every conceivable function can lead to implementing unnecessary requirements. The usage-centric approach leads to functionality that will allow the user to perform certain known tasks. This helps prevent "orphan functionality" that seems like a good idea but that no one uses because it doesn't relate directly to user goals.

# Algumas ideias a reter

- Os projetos ágeis (especialmente os da Scrum) utilizam um *backlog* do produto, que é uma lista prioritária da funcionalidade a desenvolver.
- Os itens do *backlog* do produto podem ser o que a equipa quiser, mas as histórias surgiram como a forma mais comum de representar os itens do *backlog* do produto (em software).
- Ambos os casos de utilização e as histórias focam-se em conversas e uso do sistema por pessoas.
- Os casos de utilização fornecem mais estrutura e uma forma de documentar os detalhes recolhidos em análise.
- As histórias dos utilizadores são refinadas conforme necessário. Os detalhes são acrescentados, em colaboração regular com os especialistas do domínio.
- A histórias recorrem a exemplos curtos para definir condições de aceitação.

# References

| Core readings | Suggested readings |
| --- | --- |
| • Jacobson, I., Spence, I., & Kerr, B. (2016). Use-case 2.0. *Communications of the ACM*, *59*(5), 61–69.<br>• "User Story vs Use Case for Agile Software Development", Visual Paradigm | • Jacobson, I., Spence, I., & Bittner, K. (2011). *Use-Case 2.0 The Guide o Succeeding with Use Cases*. [e-Book]<br>• User story (VisualParadigm handbook)<br>• EasyAgile training materials |