Agile RE with User Stories

Half-day tutorial at IEEE RE'18

Fabiano Dalpiaz Sjaak Brinkkemper

Requirements Engineering Lab (RE-Lab)
Dept. of Information and Computing Sciences
Utrecht University
f.dalpiaz@uu.nl

August 20, 2018

Utrecht University

Table of Contents

On this tutorial

Spoiler alert!

Four parts

1. Agile RE and user stories: fundamentals
2. The Quality User Story framework
3. Extracting conceptual models
4. Taming ambiguity and incompleteness

On this tutorial

Spoiler alert!

Four parts

1. Agile RE and user stories: fundamentals
2. The Quality User Story framework
3. Extracting conceptual models
4. Taming ambiguity and incompleteness

Interaction, also via hands-on activities

Literature

Key papers behind this tutorial

Garm Lucassen, Fabiano Dalpiaz, Jan Martijn E.M. van der Werf, and Sjaak Brinkkemper, *Improving agile requirements: the Quality User Story framework and tool*, Requirements Engineering 21 (2016), no. 3, 383–403

Garm Lucassen, Marcel Robeer, Fabiano Dalpiaz, Jan Martijn E. M. van der Werf, and Sjaak Brinkkemper, *Extracting conceptual models from user stories with Visual Narrator*, Requirements Engineering 22 (2017), no. 3, 339–358

Fabiano Dalpiaz, Ivor van der Schalk, and Garm Lucassen, *Pinpointing Ambiguity and Incompleteness in Requirements Engineering via Information Visualization and NLP*, Proceedings of the 24th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'18), 2018

Who are we?

Dr. Fabiano Dalpiaz

Assistant professor in Requirements Engineering at Utrecht University

- Artificial Intelligence (NLP and more) for RE
- Crowd Requirements Engineering
- Engaging the stakeholders via games
- Modeling languages

Local organizer of REFSQ 2018!

`http://www.staff.science.uu.nl/~dalpi001/`

Who are we?

Prof.dr. Sjaak Brinkkemper

Professor in Software Production
at Utrecht University

- Research group of 35 staff and PhDs
- Product Software: Methodology of Development, Implementation, and Entrepreneurship

`http://www.uu.nl/staff/SBrinkkemper/0`

Who are you?

- Name
- Organization
- Role
- Experience with user stories
- What do you expect to learn from this tutorial?

Credits

These slides are *partially based* on the slides by Garm Lucassen and Sjaak Brinkkemper presented at earlier tutorials and in professional courses.

Follow the slides

Download these slides from the following URL:
`https://bit.ly/2OPqiCL`

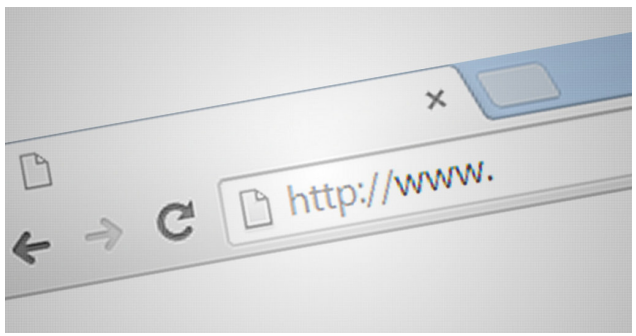


Table of Contents

Table of Contents

What is agile RE?

An informal definition

Agile RE [ISM+15]

The term "agile requirements engineering" is used to define the "agile way" of planning, executing and reasoning about requirements engineering activities.

Agile RE vs. Traditional RE

Results from a systematic literature review [ISM+15]

Seventeen practices of agile RE have been studied in the literature:

Practice	Freq.
1. Face-to-face communication	3
2. Customer involvement	3
3. User stories	2
4. Iterative requirements	3
5. Requirements prioritisation	5
6. Change management	2
7. Cross-functional teams	1
8. Prototyping	2
9. Testing before coding	4
10. Requirements modelling	2
11. Requirements management	2
12. Review meetings and acceptance tests	2
13. Code refactoring	1
14. Shared conceptualisations	1
15. Pairing for requirements analysis	1
16. Retrospectives	3
17. Continuous planning	1

Agile RE vs. Traditional RE

Challenges resolved by agile RE practices

- Communication issues
  - Frequent face-to-face meeting with the customer and among teams
  - Collocated teams for better collaboration
  - Onsite customer as opposed to contracts
  - Alternate customer representations (proxy customers)
  - Cross-functional agile teams
  - Integrated RE process, closer to development

Agile RE vs. Traditional RE

Challenges resolved by agile RE practices

- Overscoping
  - One continuous scope flow via continuous prioritization
  - Gradual detailing of requirements
  - Cross-functional teams that share responsibilities

Agile RE vs. Traditional RE

Challenges resolved by agile RE practices

- Overscoping
  - One continuous scope flow via continuous prioritization
  - Gradual detailing of requirements
  - Cross-functional teams that share responsibilities
- Requirements validation
  - Requirements prioritization done by the customer picking the most important requirements
  - Prototyping that provide a product blueprint

Agile RE vs. Traditional RE

Challenges resolved by agile RE practices

- Requirements documentation
  - User stories are precise, to-the-point, and prevent the need for long SRS documents that are hard to update
  - Face-to-face communication helps reduce ambiguities

Agile RE vs. Traditional RE

Challenges resolved by agile RE practices

- Requirements documentation
  - User stories are precise, to-the-point, and prevent the need for long SRS documents that are hard to update
  - Face-to-face communication helps reduce ambiguities
- Rare customer involvement
  - Requirements prioritization is done by the customer

Agile RE vs. Traditional RE

Open challenges (1/2)

- Minimal documentation: user stories and backlogs
  $\rightarrow$ Poor traceability
- High customer availability is demanded
  $\rightarrow$ When impossible, increased rework

Agile RE vs. Traditional RE

Open challenges (1/2)

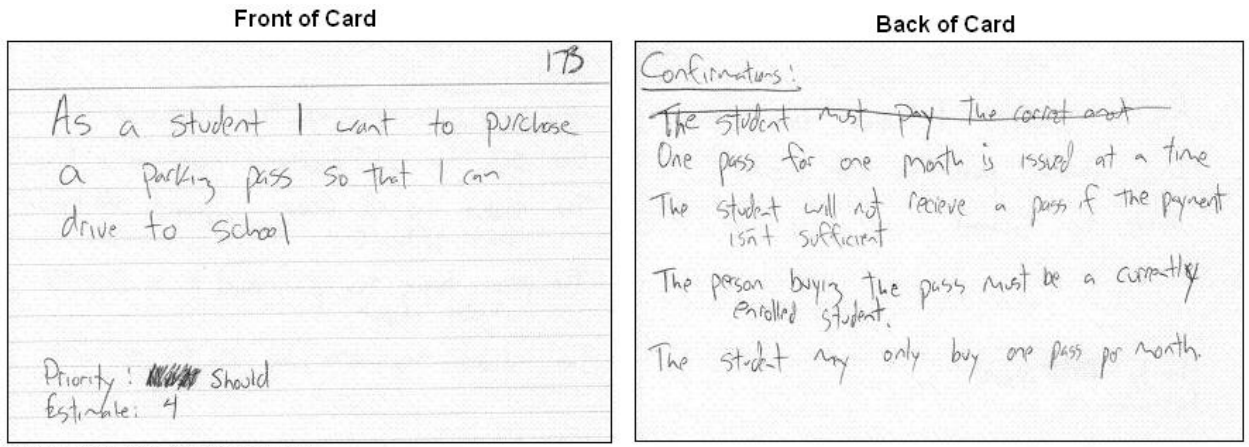
- Minimal documentation: user stories and backlogs
  → Poor traceability
- High customer availability is demanded
  → When impossible, increased rework
- Inappropriate architecture due to agility
  → Increased cost
- Budget and time estimations affected by continuous changes
  → Project delays

Agile RE vs. Traditional RE

Open challenges (2/2)

- Customer inability and lack of agreement
  → Increased rework
- Contractual limitations hindering change
  → Increased cost
- Requirements change and its consequence
  → Work delays

Your experience with user stories



How would you describe them as requirements?

What is a user story?

Some examples

- As a visitor, I want to purchase an event ticket

What is a user story?

Some examples

- As a visitor, I want to purchase an event ticket
- As a visitor, I want to search for new events by favorited organizers, so that I am the first to know of new events

What is a user story?

Some examples

- As a visitor, I want to purchase an event ticket
- As a visitor, I want to search for new events by favorited organizers, so that I am the first to know of new events
- As a visitor, I want to be notified when an event is close to becoming sold out, so that I do not miss the event

What is a user story?

Conceptualization

As a ⟨role⟩, I want to ⟨action⟩, (so that ⟨benefit⟩)

What is a user story?

Conceptualization

As a ⟨role⟩, I want to ⟨action⟩, (so that ⟨benefit⟩)

User stories only capture the essential elements of a requirement

- who it is for
- what s/he expects from the system
- why it is important (optional?)

What is a user story?

Guidelines

- Don't force a story into its format when unnatural
- Business/domain/application jargon
- No technical details

What is a user story?

Guidelines

- Don't force a story into its format when unnatural
- Business/domain/application jargon
- No technical details

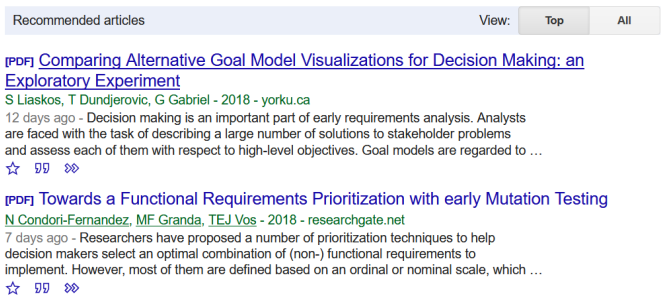
As a researcher,
I want to receive new paper notifications,
so that I can write a good literature review

What is a user story?

Guidelines

- Don't force a story into its format when unnatural
- Business/domain/application jargon
- No technical details

As a researcher,
I want to receive new paper notifications,
so that I can write a good literature review

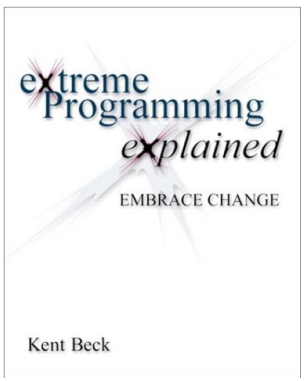


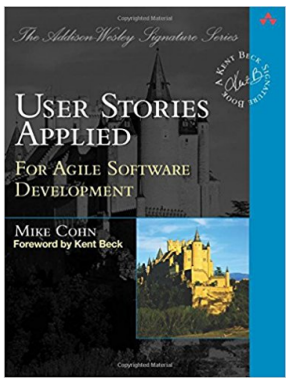
History

- First mention in Kent Beck's 1999 book "Extreme Programming Explained"
  - Unstructured text
  - Similar to use cases
  - Restricted in size
- Jeffries 2001: card, conversation, confirmation

History

- First mention in Kent Beck's 1999 book "Extreme Programming Explained"
  - Unstructured text
  - Similar to use cases
  - Restricted in size
- Jeffries 2001: card, conversation, confirmation
- Widespread popularity: Mike Cohn's "User Stories Applied" (2004)





Some evidence of their popularity

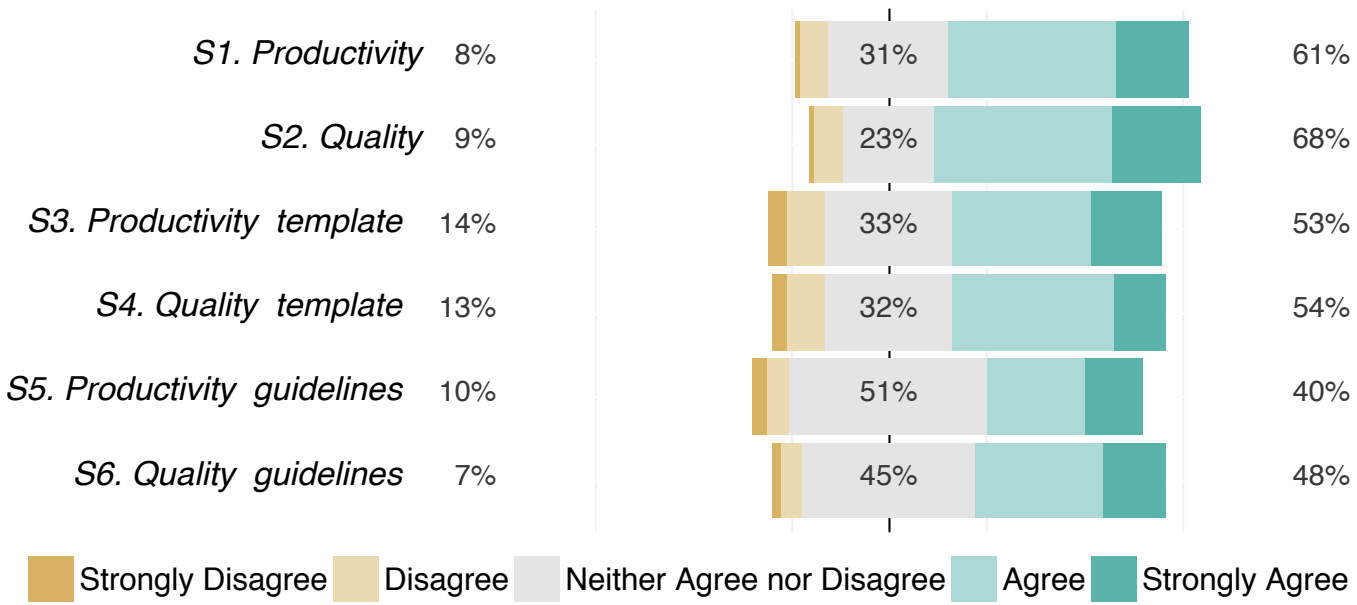
Results from academic studies

- 45% of practitioners employ user stories [Kas15]
- In agile development, adoption is up to 90% [WZWS14]

Some evidence of their popularity

Results from academic studies

- 45% of practitioners employ user stories [Kas15]
- In agile development, adoption is up to 90% [WZWS14]
- Practitioners' perception of impact is positive [LDvdWB16a]



Exercise #1

Your task

- Form groups of two
- Use a sheet of paper or a text editor
- Write at least 10 user stories for a conference management system
- About 10 minutes!

Exercise #1

Review

Exercise evaluation

- Let us discuss a few user stories!
- What are the key roles?

Exercise #1

Review

Exercise evaluation

- Let us discuss a few user stories!
- What are the key roles?

Discussion triggers

- Is the role the *actual* role?
- Did you specify the *why* part?
- Have you forced the text into the format?
- Did you use domain jargon?
- Are there technical details?

Table of Contents

INVEST

The state of the practice

Several organizations use the INVEST framework – Bill Wake 2003

- Independent: minimize dependencies between user stories
- Negotiable: details are discussed in the iteration planning meetings
- Valuable to the customer
- Estimable: detailed enough to allow effort estimation
- Small in effort
- Testable with certain acceptance criteria

INVEST

The state of the practice

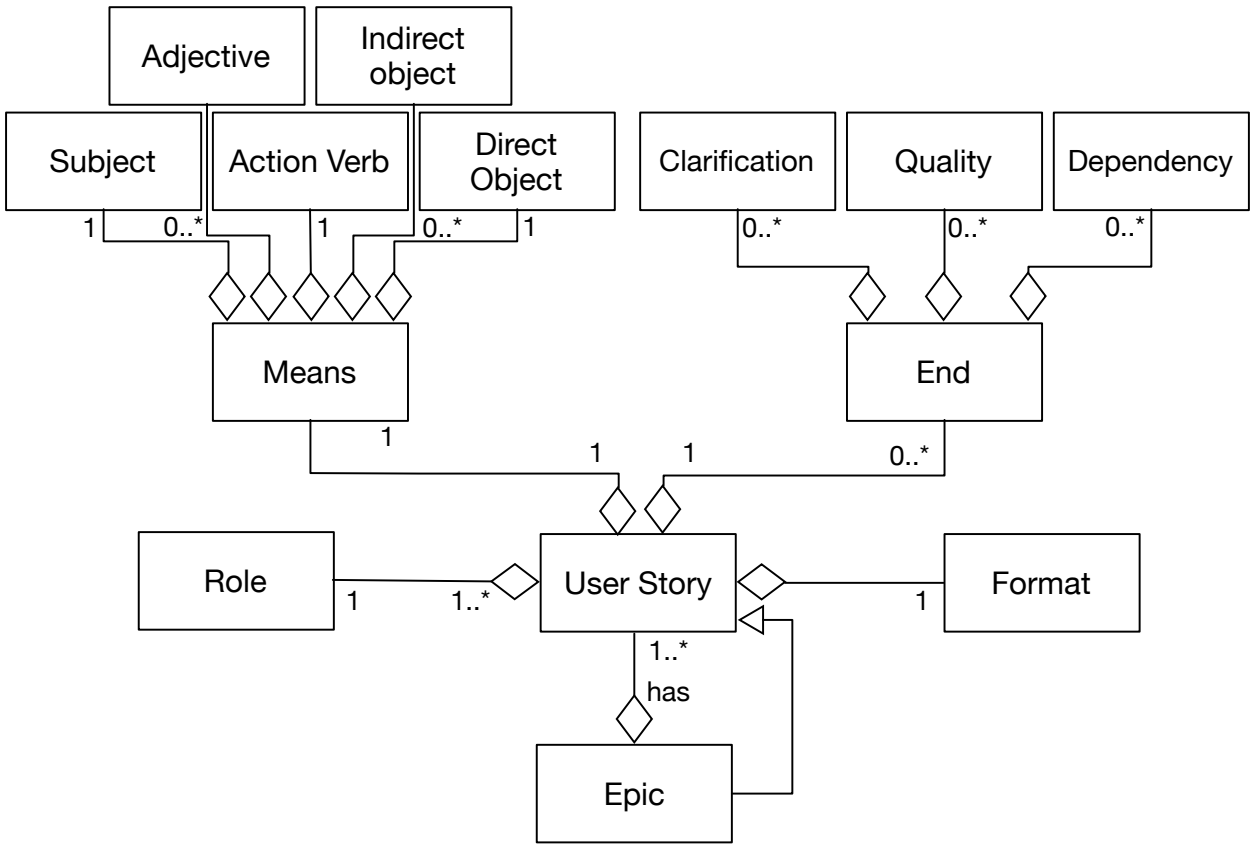
Several organizations use the INVEST framework – Bill Wake 2003

- Independent: minimize dependencies between user stories
- Negotiable: details are discussed in the iteration planning meetings
- Valuable to the customer
- Estimable: detailed enough to allow effort estimation
- Small in effort
- Testable with certain acceptance criteria

Useful, but hard to operationalize!

Understanding user stories

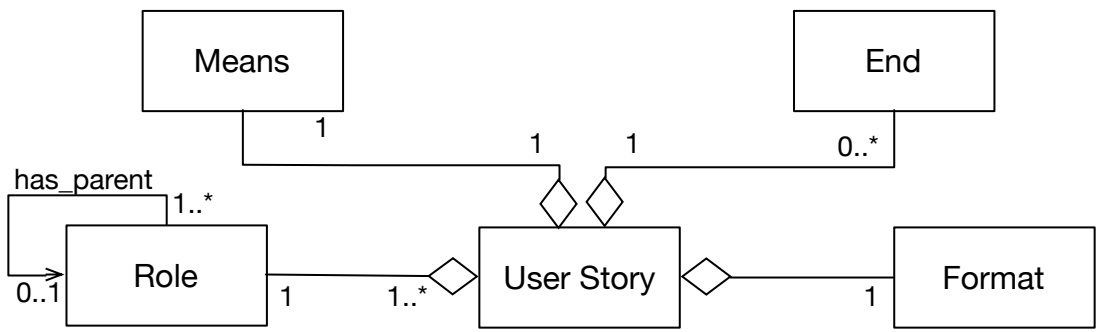
An approach based on linguistics [LDvdWB16b]



Understanding user stories

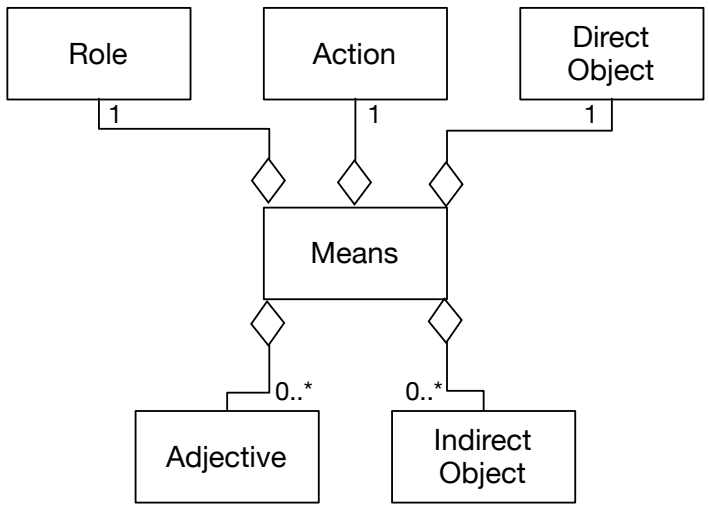
Applying the conceptual model, high-level analysis

As a $\langle$researcher$\rangle_{role}$,
I want to $\langle$receive new paper notifications$\rangle_{means}$,
so that $\langle$I can write a good literature review$\rangle_{end}$



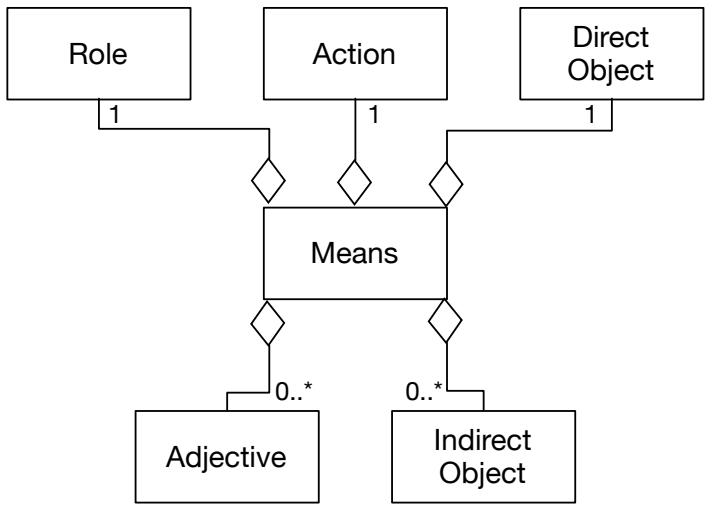
Understanding user stories

Applying the conceptual model, means



Understanding user stories

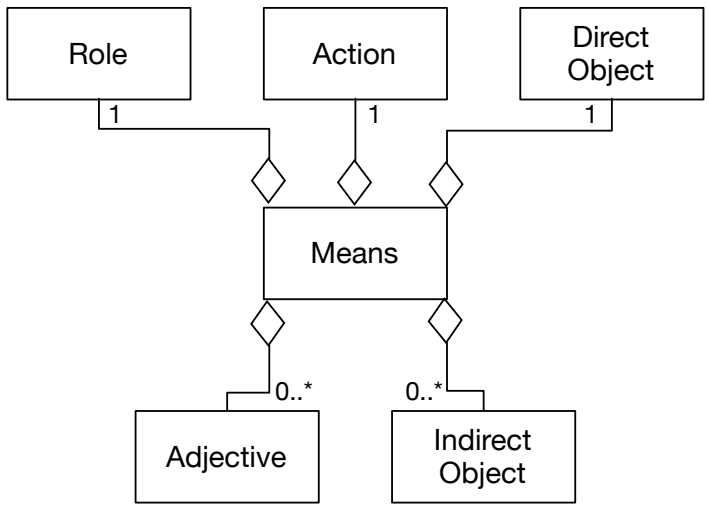
Applying the conceptual model, means



"I want to receive new paper notifications"

Understanding user stories

Applying the conceptual model, means



"I want to receive new paper notifications"

$$\Downarrow$$

$$\langle \text{I} \rangle_{role} \text{ want to } \langle \text{receive} \rangle_{action} \langle \text{new} \rangle_{adjective} \langle \text{paper notifications} \rangle_{d\text{-}object}$$

Understanding user stories

Applying the conceptual model, end

The *end* may represent one or more of the following:

- A *clarification* of the means
- A *quality aspect*
- A *dependency* on another user story

Understanding user stories

Applying the conceptual model, end

The *end* may represent one or more of the following:

- A *clarification* of the means
- A *quality aspect*
- A *dependency* on another user story

"so that I can write a good literature review"

Understanding user stories

Applying the conceptual model, end

The *end* may represent one or more of the following:

- A *clarification* of the means
- A *quality aspect*
- A *dependency* on another user story

"so that I can write a good literature review"

$$\Downarrow$$

$$\langle \text{I can write a } \langle \text{good} \rangle_{quality} \ \langle \text{literature review} \rangle_{dependency} \rangle_{clarification}$$

Quality problems in practice

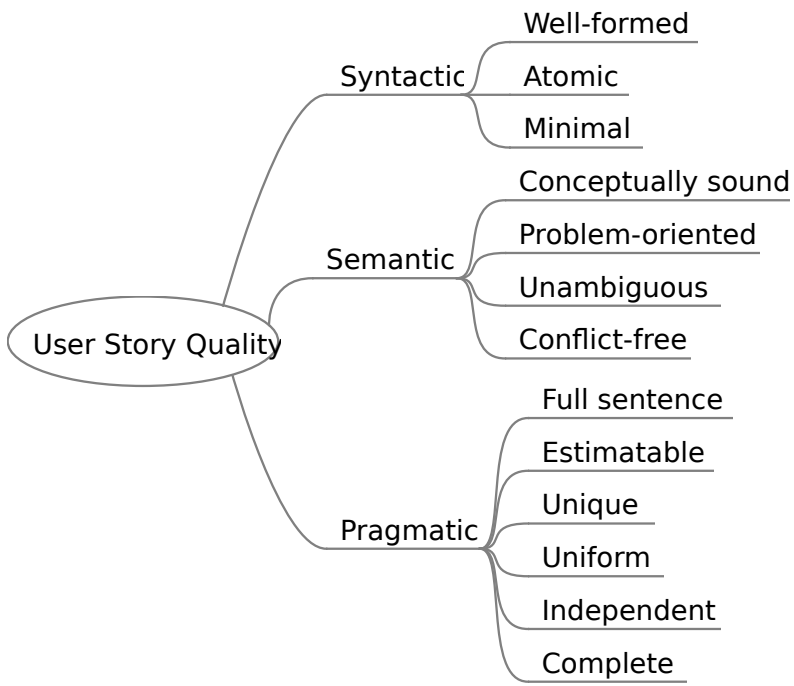
Regardless of INVEST

- The conceptual model captures correct stories
- In practice, however, stories
  - Are too long
  - Include unnecessary information
  - Include too little information
  - Are inconsistent
  - Are irrelevant for the software to-be
  - Contain ambiguity

The Quality User Story Framework

Overview

- Based on the critical analysis of hundreds of user stories
- Includes insights from other frameworks such as INVEST



The Quality User Story Framework

Quality of individual stories

Criteria	Description
Well-formed	Includes at least a role and a means
Atomic	Expresses a requirement for exactly one feature
Minimal	Contains nothing more than role, means and ends
Conceptually sound	The means expresses a feature and the ends a rationale
Problem-oriented	Only specifies the problem, not the solution to it
Unambiguous	Avoids terms that lead to multiple interpretations
Full sentence	Is a well-formed full sentence
Estimable	Does not denote an unrefined requirement that is difficult to plan and prioritize

The Quality User Story Framework

Quality of user story sets

Criteria	Description
Conflict-free	There should not be 2+ inconsistent user stories
Unique	Duplicates shall be avoided
Uniform	All user stories in a specification employ the same template
Independent	A user story is self-contained and has no inherent dependencies on other stories
Complete	Implementing a set of user stories creates a feature-complete application, no steps are missing

QUS in practice

A first set of criteria

- Don't consider all criteria upfront!
- Focus on
  1. Well-formed
  2. Atomic
  3. Minimal
  4. Conceptually sound
  5. Problem oriented
  6. Full sentence
  7. Uniform

QUS in practice

1. Well-formed

Well-formed

A user story includes at least a role and an action

Example (Violation)

I want to revoke access to problematic event organizers

QUS in practice

1. Well-formed

Well-formed

A user story includes at least a role and an action

Example (Violation)

I want to revoke access to problematic event organizers

$\Downarrow$ *(add role)*

As a TicketExpert Employee, I want to revoke access to problematic event organizers

QUS in practice

2. Atomic

Atomic

A user story expresses a requirement for exactly one feature/problem

Example (Violation)

As a Visitor, I want to register for an event and create a personal account, so that I can quickly register for future events

QUS in practice

2. Atomic

Atomic

A user story expresses a requirement for exactly one feature/problem

Example (Violation)

As a Visitor, I want to register for an event and create a personal account, so that I can quickly register for future events

⇓ *(split)*

1. As a Visitor, I want to register for an event, so that I am admitted to the event
2. As a Visitor, I want to create a personal account during event registration, so that I can quickly register for future events

QUS in practice

3. Minimal

Minimal

A user story contains nothing more than role, action and benefit

Example (Violation)

As an Event Organizer, I want to see the personal information of attendees (split into price levels). See: Mockup by Alice NOTE: - First create the overview screen

QUS in practice

3. Minimal

Minimal

A user story contains nothing more than role, action and benefit

Example (Violation)

As an Event Organizer, I want to see the personal information of attendees (split into price levels). See: Mockup by Alice NOTE: - First create the overview screen

⇓ *(remove unnecessary information)*

As an Event Organizer, I want to see the personal information of attendees

QUS in practice

4. Conceptually sound

Conceptually sound

The action expresses a feature and the benefit expresses a rationale

Example (Violation)

As an Event Organizer, I want to open the event page, so that I can see the personal information of attendees

QUS in practice

4. Conceptually sound

Conceptually sound

The action expresses a feature and the benefit expresses a rationale

Example (Violation)

As an Event Organizer, I want to open the event page, so that I can see the personal information of attendees

⇓ *(end becomes a separate means)*

1. As an Event Organizer, I want to open the event page, so that I can review event related information
2. As a User, I want to see personal information of attendees, so that I know the demographical distribution of the event

QUS in practice

5. Problem oriented

Problem oriented

A user story only specifies the problem, not the solution to it

Example (Violation)

As a Visitor, I want to download an event ticket. - Add download button on top right (never grayed out)

QUS in practice

5. Problem oriented

Problem oriented

A user story only specifies the problem, not the solution to it

Example (Violation)

As a Visitor, I want to download an event ticket. - Add download button on top right (never grayed out)

⇓ *(remove solution)*

As a Visitor, I want to download an event ticket

QUS in practice

6. Full sentence

Full sentence

A user story is a well-formed full sentence

Example (Violation)

update profile

QUS in practice

6. Full sentence

Full sentence

A user story is a well-formed full sentence

Example (Violation)

update profile

⇓ *(add 'want to')*

As a Visitor, I want to update my profile

QUS in practice

7. Uniform

Uniform

All user stories follow (roughly) the same template

Example (Violation)

1. As a Visitor, I want to create an account
2. As a Visitor, I want to reset my password
3. As a TicketExpert Manager, I receive an email notification when a new user is registered

QUS in practice

7. Uniform

Uniform

All user stories follow (roughly) the same template

Example (Violation)

1. As a Visitor, I want to create an account
2. As a Visitor, I want to reset my password
3. As a TicketExpert Manager, I receive an email notification when a new user is registered

⇓ *(add 'want to')*

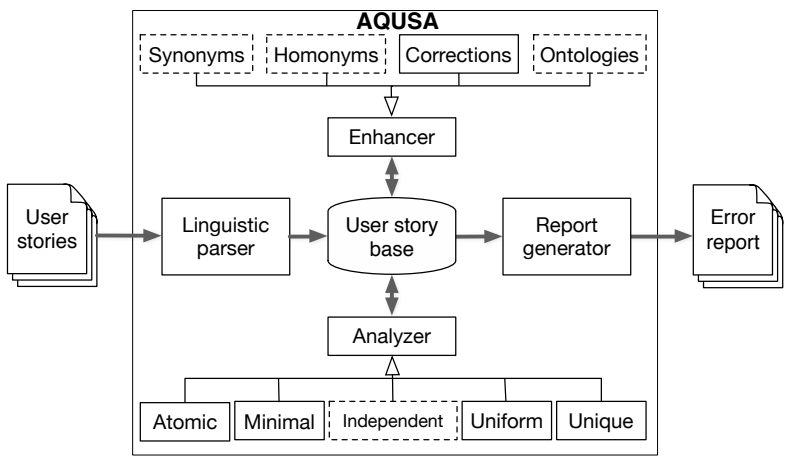
As an TicketExpert Manager, I want to receive an email notification when a new user is registered

The AQUSA tool

The Automatic Quality User Story Artisan

Tool developed at UU: `www.aqusa.nl`

- Automatically assesses user story quality according to QUS
- Focus on those criteria with potential for 100% recall
  - Well-formed
  - Atomic
  - Minimal
  - Explicit dependencies
  - Uniform
  - Unique



Exercise #2

Analyze a set of user stories

- Open `https://bit.ly/2nyvb7Q` with your browser
- Manually explore the output of the AQUSA tool
- Try to identify similar defects in the user stories that you wrote
- 10-15 minutes!

Exercise #2

Review

Exercise evaluation

- What were the most common mistakes?
- What is their impact?
- How do your own stories compare to the data set?
- Do you agree with the fixes that are suggested?

Estimating and developing

Applying the other criteria

- After initial fixes to sanitize the user stories, the other criteria become relevant
  8. Unambiguous
  9. Conflict-free
  10. Estimable
  11. Independent
  12. Unique
  13. Complete

Estimating and developing

Applying the other criteria

- After initial fixes to sanitize the user stories, the other criteria become relevant
  8. Unambiguous
  9. Conflict-free
  10. Estimable
  11. Independent
  12. Unique
  13. Complete

We focus only on some of these criteria today

QUS in practice: improving the user stories

8. Unambiguous

Unambiguous

A user story avoids terms that lead to multiple interpretations

Example (Violation)

As an Event Organizer, I want to edit the content that I added to an event's page

QUS in practice: improving the user stories

8. Unambiguous

Unambiguous

A user story avoids terms that lead to multiple interpretations

Example (Violation)

As an Event Organizer, I want to edit the content that I added to an event's page

$\Downarrow$ *(clarify the term "content")*

As an Event Organizer, I want to edit video and text content that I added to an event's page

QUS in practice: improving the user stories

8. Unambiguous

Unambiguous

A user story avoids terms that lead to multiple interpretations

Example (Violation)

As an Event Organizer, I want to edit the content that I added to an event's page

⇓ *(clarify the term "content")*

As an Event Organizer, I want to edit video and text content that I added to an event's page

More on ambiguity in the fourth part of the tutorial!

QUS in practice: improving the user stories

12. Independent

Independent

A user story is self-contained and has no inherent dependencies on other stories

Example (Violation)

1. As an Event Organizer, I am able to add a new event
2. As a Visitor, I am able to view an event page

QUS in practice: improving the user stories

12. Independent

Independent

A user story is self-contained and has no inherent dependencies on other stories

Example (Violation)

1. As an Event Organizer, I am able to add a new event
2. As a Visitor, I am able to view an event page

⇓ No solution here!

- It is not always possible for user stories to be fully independent
- Avoid dependencies as much as possible, but be flexible!

QUS in practice: improving the user stories

13. Complete

Complete

Implementing a set of user stories creates a feature-complete application, no steps are missing

Example (Violation)

1. As an Event Organizer, I want to update an event
2. As an Event Organizer, I want to delete an event

QUS in practice: improving the user stories

13. Complete

Complete

Implementing a set of user stories creates a feature-complete application, no steps are missing

Example (Violation)

1. As an Event Organizer, I want to update an event
2. As an Event Organizer, I want to delete an event

$\Downarrow$ *(add story)*

As an Event Organizer, I want to create an event

Exercise #3

Further improve your user stories

- Take a look at the user stories you have written
- Check them against the additional criteria
  1. Unambiguous
  2. Independent
  3. Complete
- 10 minutes!

Exercise #3

Review

Exercise evaluation

- Which was the most common defect?
- How simple was the task at hand?
- Share some examples!
- Any[1] doubts?

[1] . . . or many

Table of Contents

OK, so you've got a set of sanitized user stories

Additional obstacles

- As development goes on, the number of user stories increase

OK, so you've got a set of sanitized user stories

Additional obstacles

- As development goes on, the number of user stories increase
  - How to get a holistic view?

OK, so you've got a set of sanitized user stories

Additional obstacles

- As development goes on, the number of user stories increase
  - How to get a holistic view?
- Team members leave, and take away their know-how

OK, so you've got a set of sanitized user stories

Additional obstacles

- As development goes on, the number of user stories increase
  - How to get a holistic view?
- Team members leave, and take away their know-how
- Novices need to learn the jargon

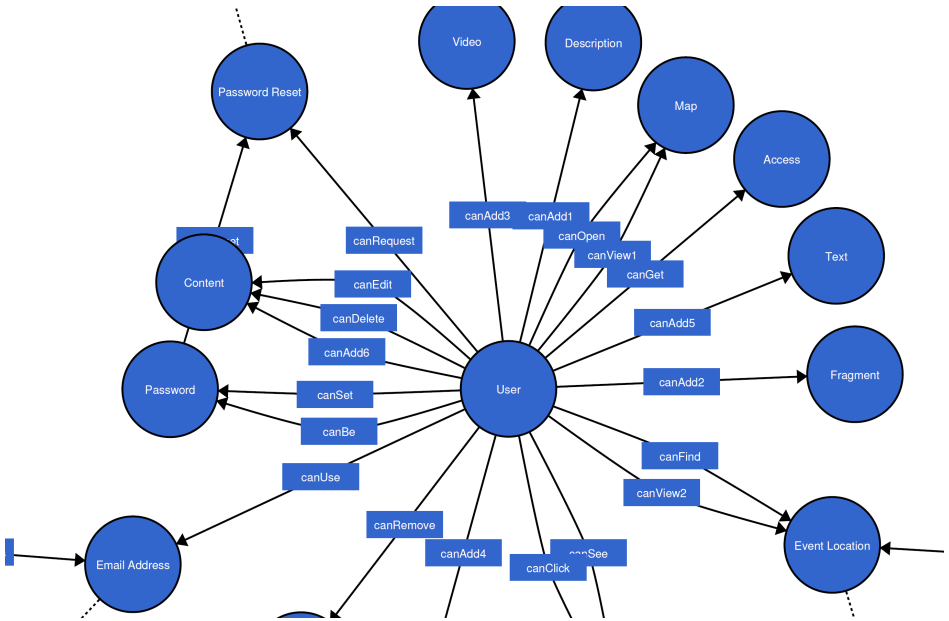
OK, so you've got a set of sanitized user stories

Additional obstacles

- As development goes on, the number of user stories increase
  - How to get a holistic view?
- Team members leave, and take away their know-how
- Novices need to learn the jargon
  - In agile development, sometimes without a glossary!

How about extracting a holistic overview?

Conceptual modeling to the rescue!

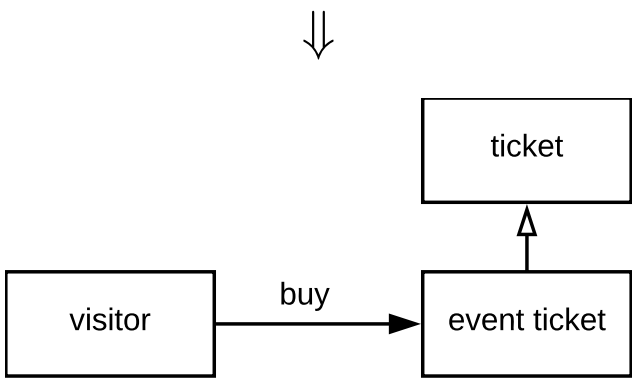


Garm Lucassen, Marcel Robeer, Fabiano Dalpiaz, Jan Martijn E. M. van der Werf, and Sjaak Brinkkemper, *Extracting conceptual models from user stories with Visual Narrator*, Requirements Engineering 22 (2017), no. 3, 339–358

Conceptual model extraction

Intuition

As a visitor, I want to buy an event ticket



Conceptual model extraction

1. Split on indicators

Role	As a visitor,
Means	I want to choose an event
End	so that I can book a ticket for that event

Conceptual model extraction

2. Functional role

Role	As a $\langle \text{visitor} \rangle_{ent}$,
Means	I want to choose an event
End	so that I can book a ticket for that event

Conceptual model extraction

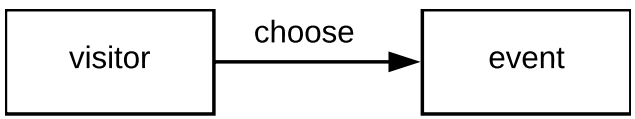
3. Simplify the means

Role	As a $\langle \text{visitor} \rangle_{ent}$,
Means	$\langle \text{I} \rangle_{=visitor}$ ~~want to~~ choose an event
End	so that I can book a ticket for that event

Conceptual model extraction

4. Main relationship

Role	As a $\langle \text{visitor} \rangle_{ent}$,
Means	$\langle \text{I} \rangle_{=visitor}$ ~~want to~~ $\langle \text{choose} \rangle_{rel}$ ~~an~~ $\langle \text{event} \rangle_{ent}$
End	so that I can book a ticket for that event



Conceptual model extraction

5. Simplify the end

Role	As a $\langle \text{visitor} \rangle_{ent}$,
Means	$\langle \text{I} \rangle_{=visitor}$ ~~want to~~ $\langle \text{choose} \rangle_{rel}$ ~~an~~ $\langle \text{event} \rangle_{ent}$
End	so that $\langle \text{I} \rangle_{=visitor}$ ~~can~~ book a ticket for that event

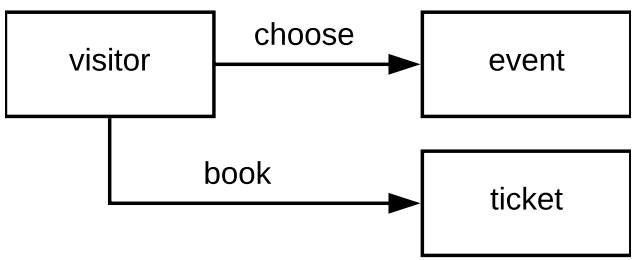
Conceptual model extraction

6. End relationship

Role As a $\langle \text{visitor} \rangle_{ent}$,

Means $\langle \text{I} \rangle_{=visitor}$ ~~want to~~ $\langle \text{choose} \rangle_{rel}$ ~~an~~ $\langle \text{event} \rangle_{ent}$

End so that $\langle \text{I} \rangle_{=visitor}$ ~~can~~ $\langle \text{book} \rangle_{rel}$ a $\langle \text{ticket} \rangle_{ent}$ for that $\langle \text{event} \rangle_{ent}$

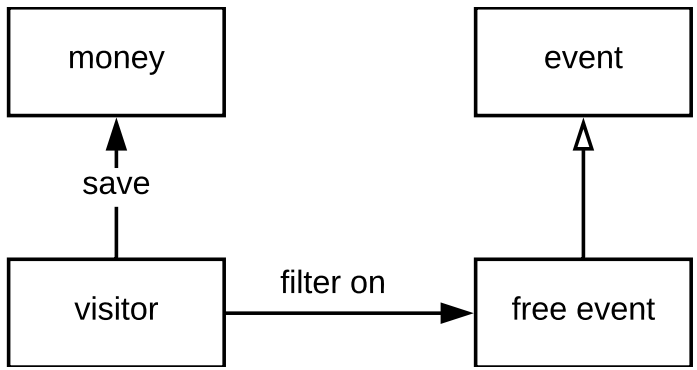


Conceptual model extraction

Create a holistic conceptual model

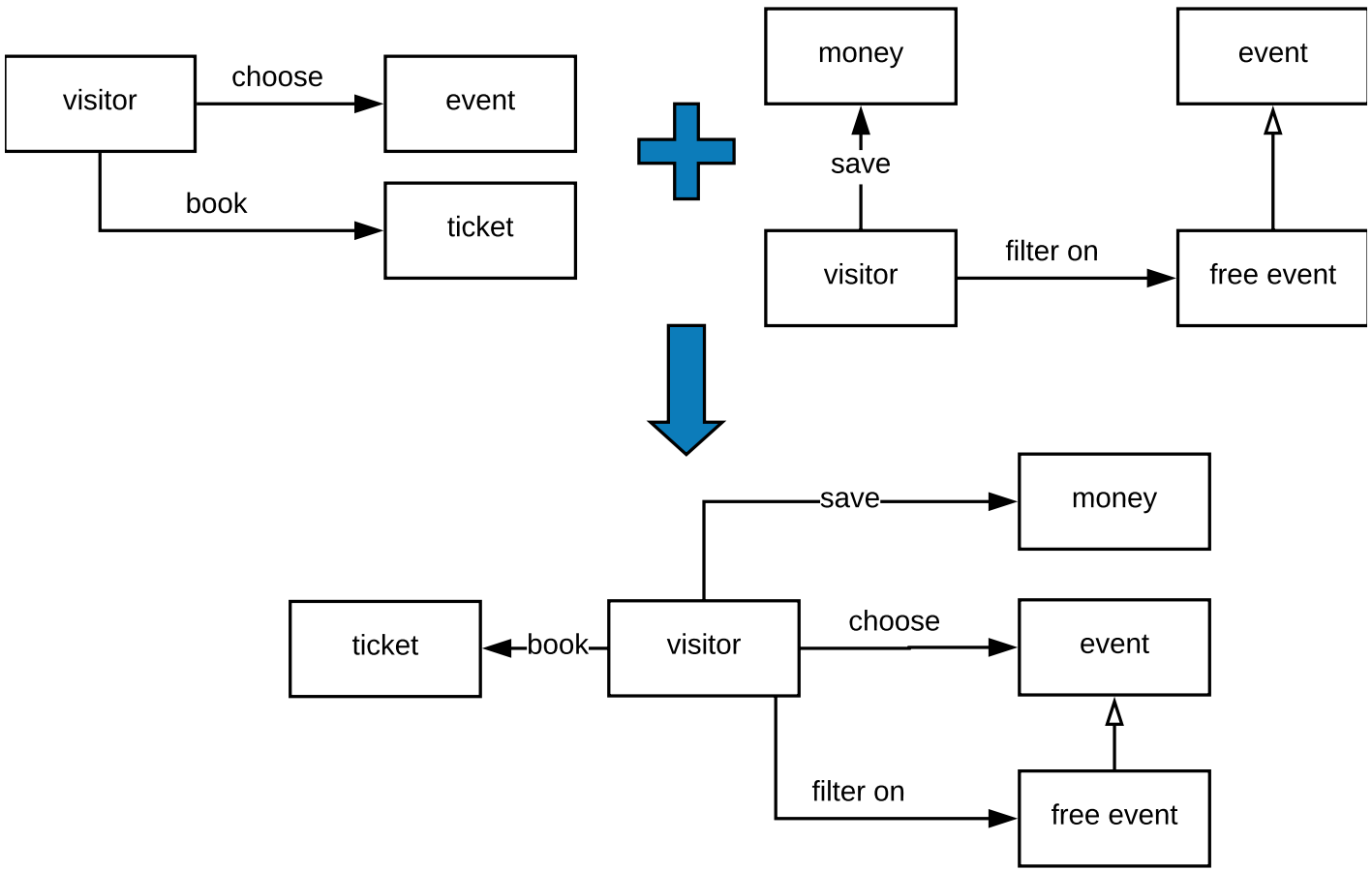
Repeat the described process for each story in the user story collection

Role	As a visitor,
Means	I want to filter on free events
End	so that I can save money



Conceptual model extraction

Merging the results



Analyzing conceptual models

Some possible uses

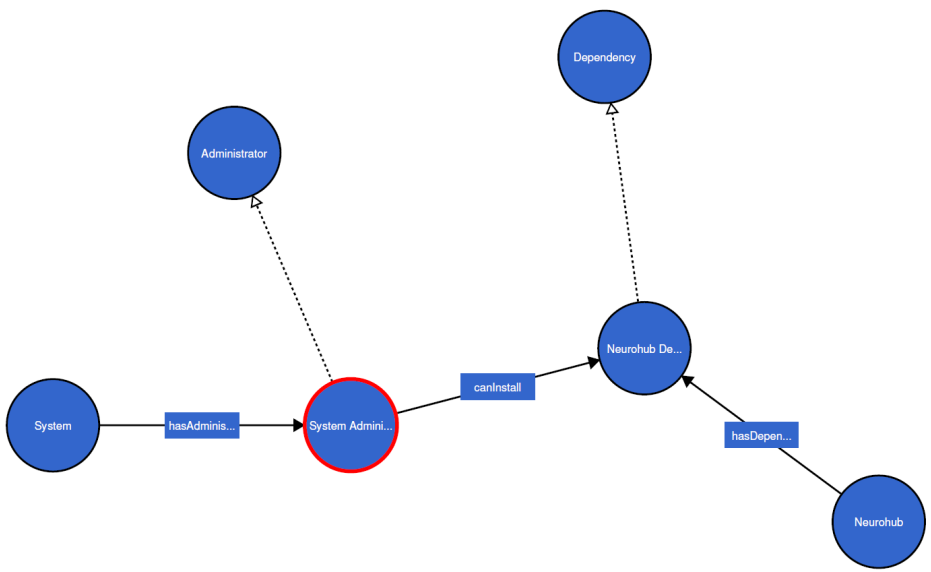
The resulting conceptual model can be used for different purposes:

- Possible inconsistencies
  - Conflict detection
  - Duplicate prevention
  - Ambiguity resolution
- Incompleteness mitigation

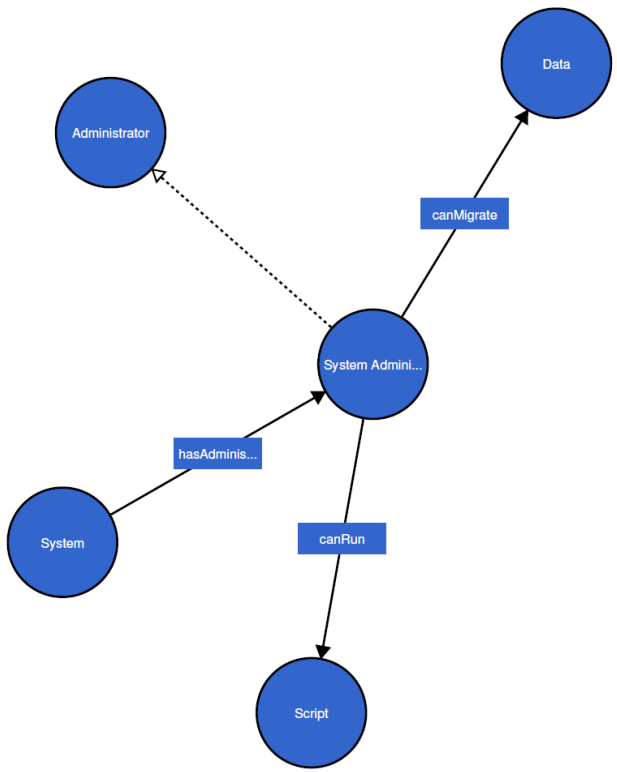
Analyzing conceptual models

Conflict detection

Systems administrator



System administrator

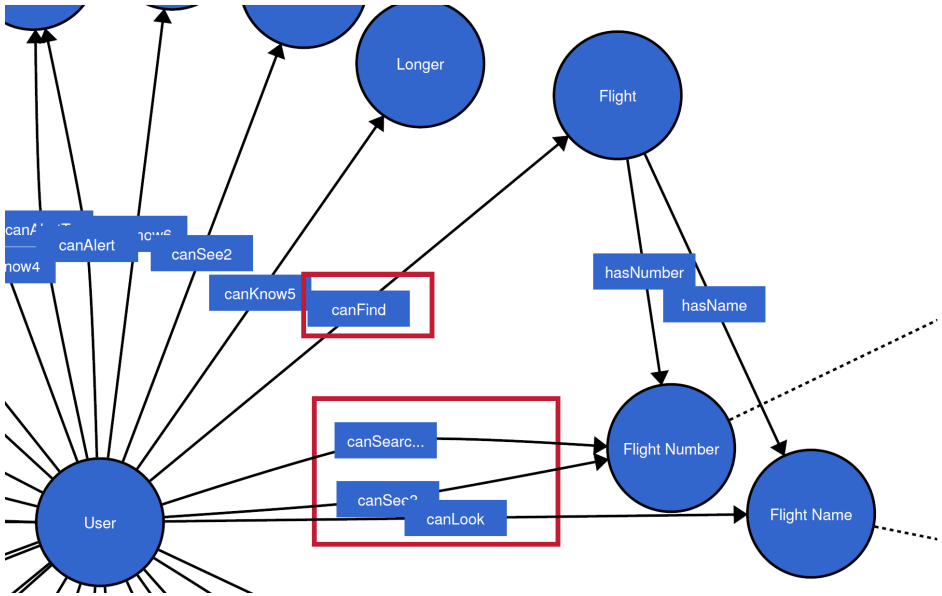


Analyzing conceptual models

Duplicate prevention

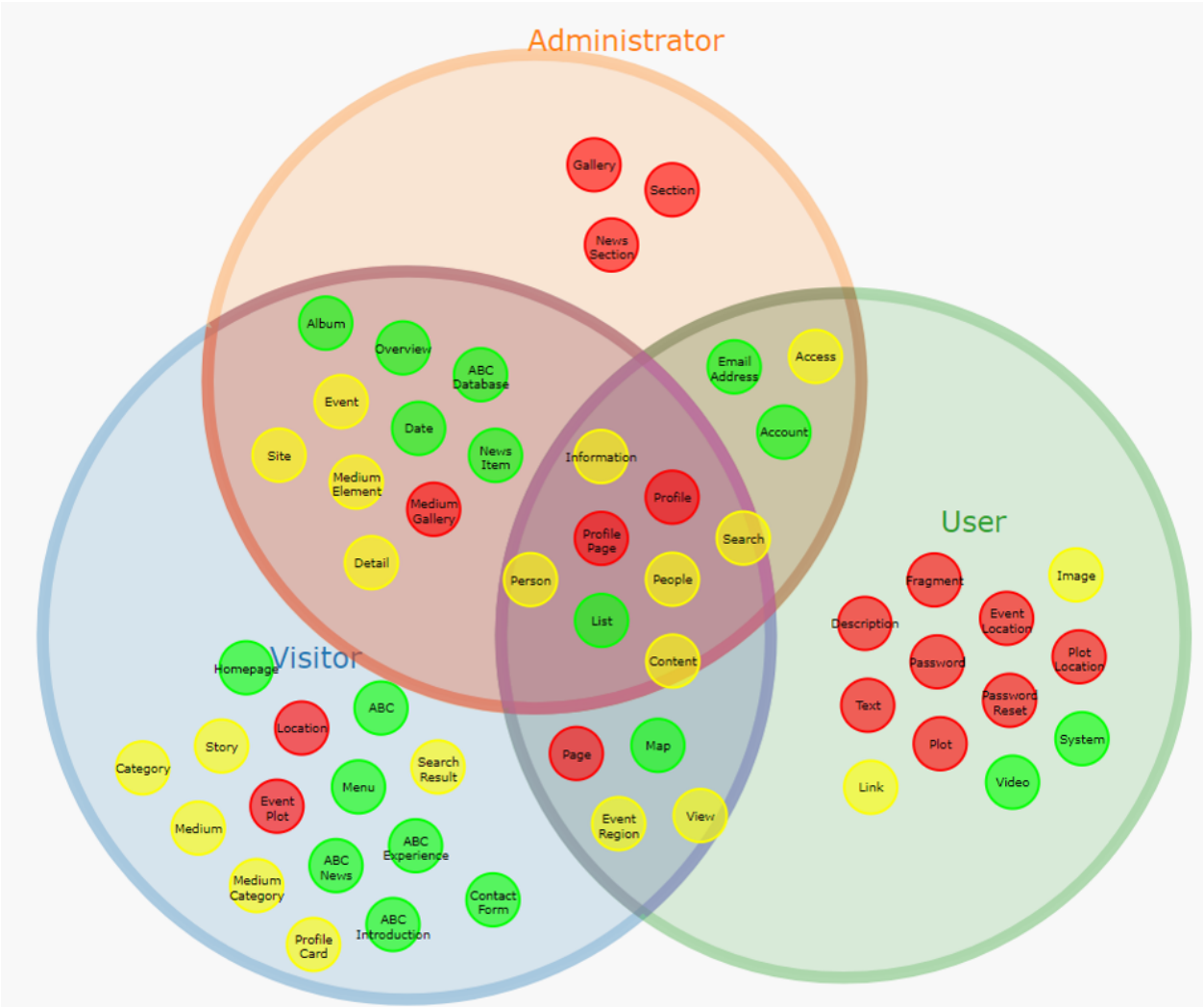
Separate stories for

- Find flight
- Search flight number
- Look for flight name



Analyzing conceptual models

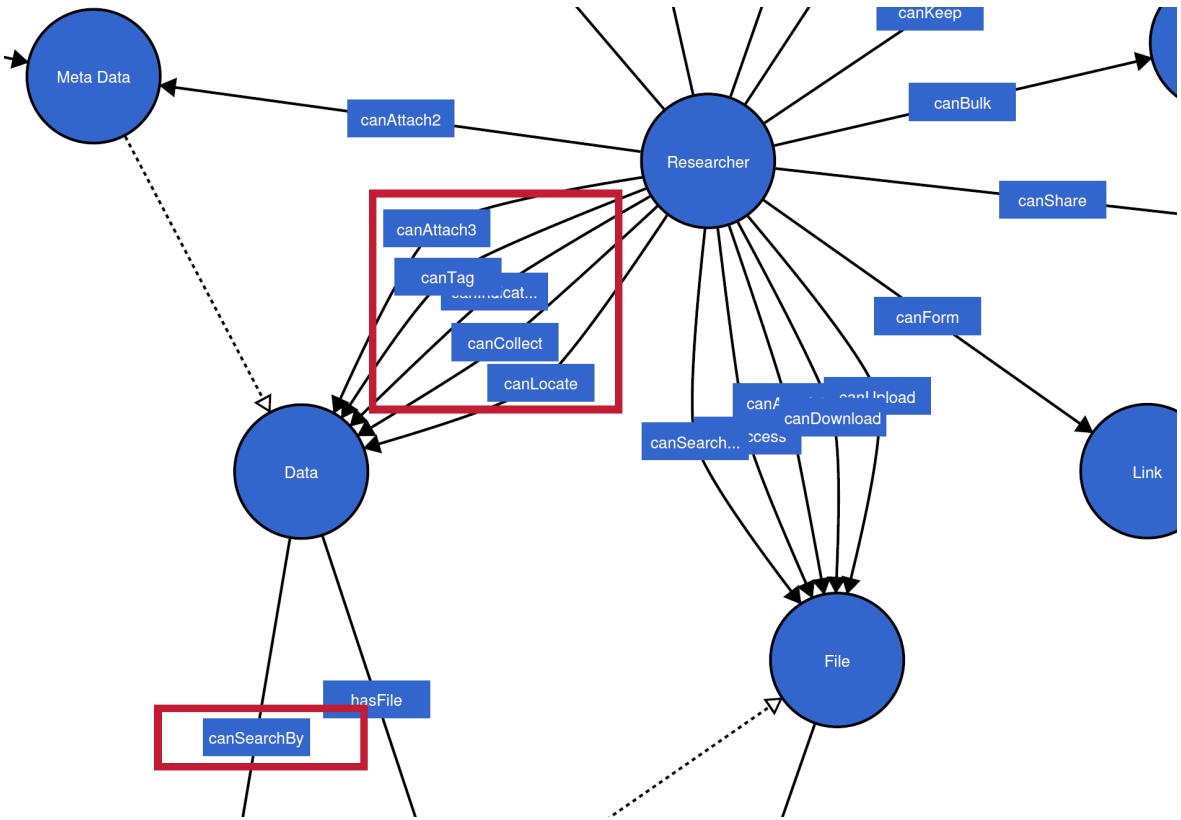
Ambiguity resolution → wait for part IV of this tutorial



Analyzing conceptual models

Incompleteness mitigation

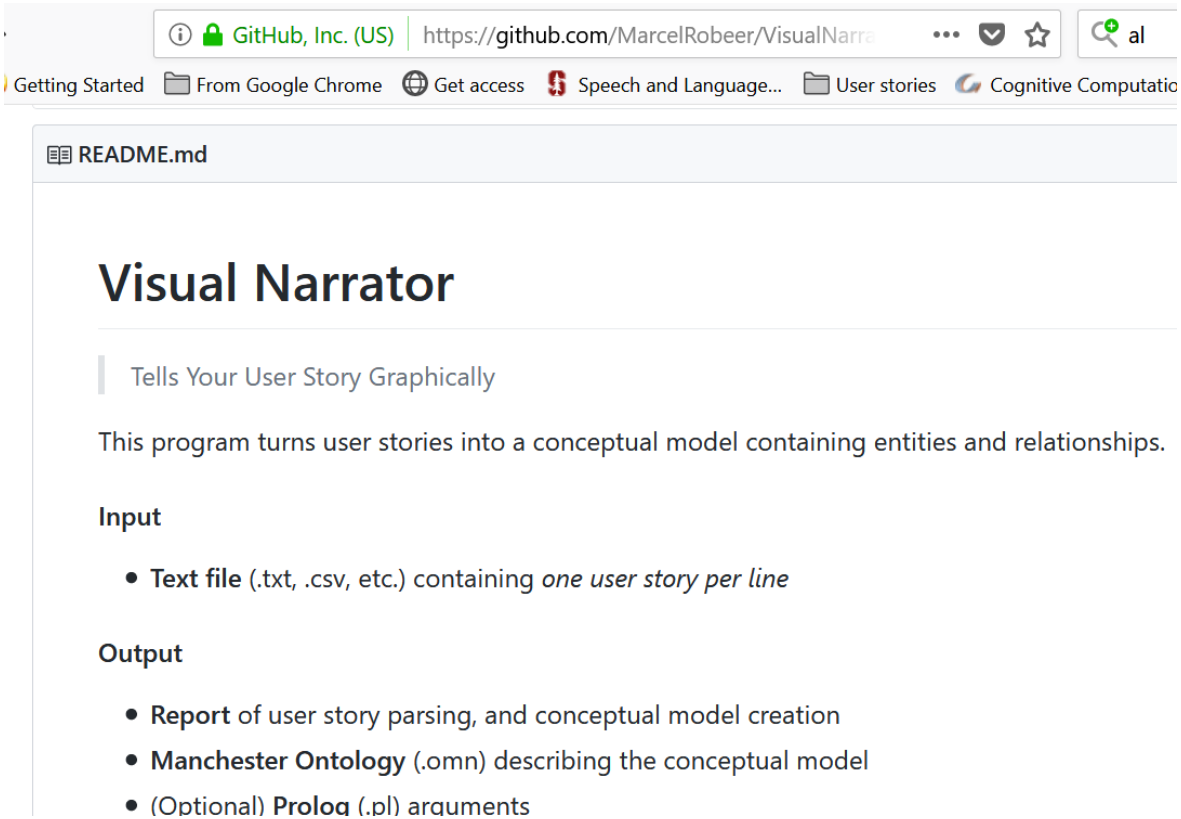
It seems that researcher cannot search by type



Analyzing conceptual models

Tool support: extraction

Conceptual models are automatically extracted from user stories with the Visual Narrator: `https://github.com/MarcelRobeer/VisualNarrator`



Analyzing conceptual models

Tool support: visualization

The outputs of the Visual Narrator can be visualized by

- the Interactive Narrator
- WebVOWL
- REVV-Light
- ...

Analyzing conceptual models

Tool support: visualization

The outputs of the Visual Narrator can be visualized by

- the Interactive Narrator
- WebVOWL
- REVV-Light
- ...

For simplicity, today we are going to use WebVOWL

Analyzing conceptual models

Four real-life data sets

- CamperPlus: turn camp management into a quick, easy and efficient experience
- Alfred: a personal interactive assistant for independent living and active ageing
- UniBath: an institutional data repository for the University of Bath
- Cornell: the Cornell Photos image library supports the university's marketing and communications needs

Analyzing conceptual models

Four real-life data sets

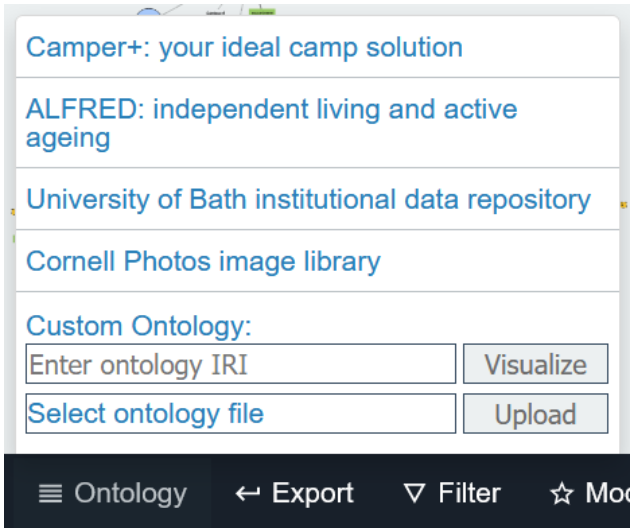
- CamperPlus: turn camp management into a quick, easy and efficient experience
- Alfred: a personal interactive assistant for independent living and active ageing
- UniBath: an institutional data repository for the University of Bath
- Cornell: the Cornell Photos image library supports the university's marketing and communications needs

Find them online: `https://bit.ly/2vH6sCO`

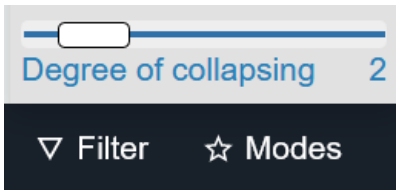
Analyzing conceptual models

Using WebVOWL

1. Launch the WebVOWL tool: `https://bit.ly/2MdZDmB`
2. Load one of the four ontologies



3. Use the degree of collapsing filter to see more/less elements

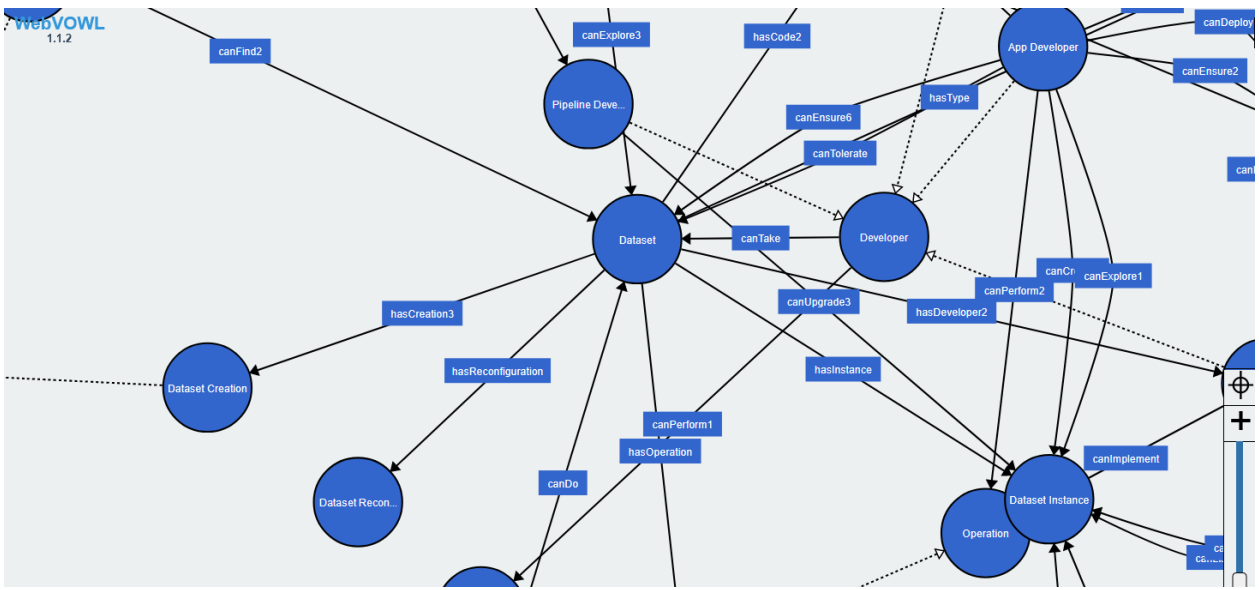


4. Tick "compact notation" in the "Modes" menu

Exercise #4

Explore models with WebVOWL

- Explore one of the four data sets
- Look for
  - Conflicts
  - Duplicates
  - Incompleteness



Exercise #4

Review

Exercise evaluation

- What were the main difficulties?
- How large are the models?
- Could you identify defects?

Table of Contents

Basic principle

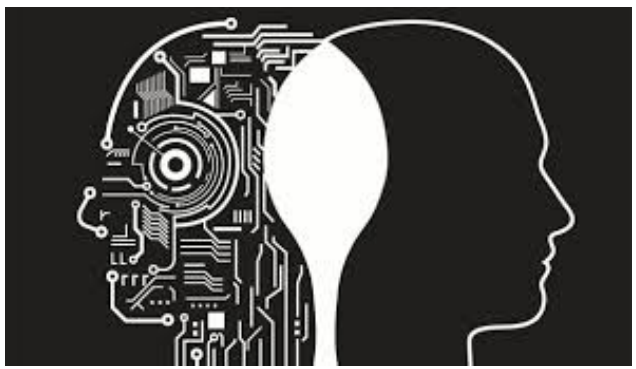
Identifying requirements defects is still hard!

- Natural language processing (NLP) tools do not deliver perfect accuracy in automated defect identification
- Human analysts are effective, but how do they scale?

Basic principle

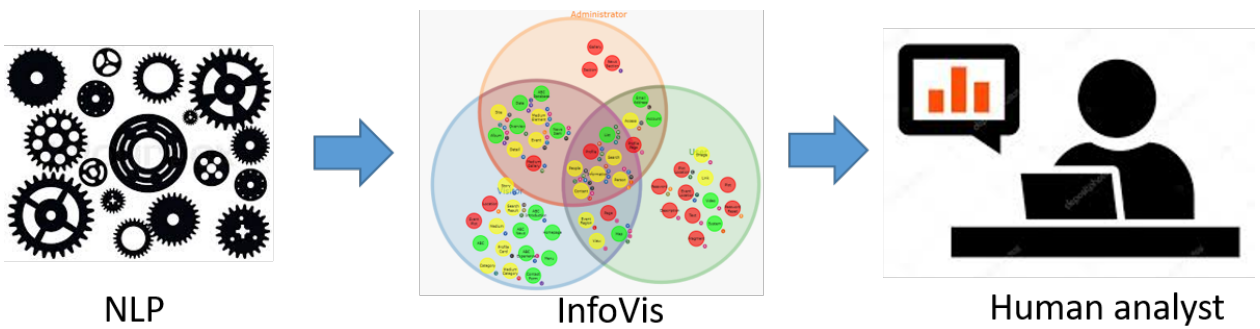
Identifying requirements defects is still hard!

- Natural language processing (NLP) tools do not deliver perfect accuracy in automated defect identification
- Human analysts are effective, but how do they scale?



Idea

To combine NLP with information visualization (InfoVis)
$\rightarrow$ automation to help humans



Terminological ambiguity

- Different stakeholders have their own viewpoints
- Including different terminologies!
  - Do *automobile* and *car* have the same meaning?

Terminological ambiguity

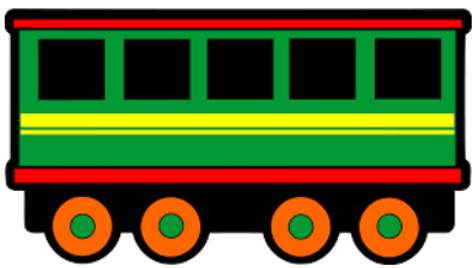
- Different stakeholders have their own viewpoints
- Including different terminologies!
  - Do *automobile* and *car* have the same meaning?
- Let $[\![\mathrm{t}]\!]^{V_1}$ be the denotation of term t according to viewpoint V_1

Terminological ambiguity

- Different stakeholders have their own viewpoints
- Including different terminologies!
  - Do *automobile* and *car* have the same meaning?
- Let $[\![t]\!]^{V_1}$ be the denotation of term t according to viewpoint V_1

$[\![car]\!]^{V_{Fabiano}}$

$[\![car]\!]^{V_{train\ engineer}}$



Terminological ambiguity

Viewpoints and conceptual systems [SG89]

Given two stakeholders with viewpoints V_1 and V_2,

1. *Consensus*: same terminology, same distinction
   $[\![\text{bank}]\!]^{V_1} = [\![\text{bank}]\!]^{V_2} = \text{a financial institution}$

Terminological ambiguity

Viewpoints and conceptual systems [SG89]

Given two stakeholders with viewpoints V_1 and V_2,

1. *Consensus*: same terminology, same distinction
   $[\![\text{bank}]\!]^{V_1} = [\![\text{bank}]\!]^{V_2} = \text{a financial institution}$
2. *Correspondence*: different terminology, same distinction
   $[\![\text{car}]\!]^{V_1} = [\![\text{automobile}]\!]^{V_2} = \text{road vehicle}$

Terminological ambiguity

Viewpoints and conceptual systems [SG89]

Given two stakeholders with viewpoints V_1 and V_2,

1. *Consensus*: same terminology, same distinction
   $[\![\text{bank}]\!]^{V_1} = [\![\text{bank}]\!]^{V_2} = \text{a financial institution}$
2. *Correspondence*: different terminology, same distinction
   $[\![\text{car}]\!]^{V_1} = [\![\text{automobile}]\!]^{V_2} = \text{road vehicle}$
3. *Conflict*: same terminology, different distinction.
   $[\![\text{bank}]\!]^{V_1} = \text{financial institution}$, $[\![\text{bank}]\!]^{V_2} = \text{land alongside a river}$

Terminological ambiguity

Viewpoints and conceptual systems [SG89]

Given two stakeholders with viewpoints V_1 and V_2,

1. *Consensus*: same terminology, same distinction
   $[\![\text{bank}]\!]^{V_1} = [\![\text{bank}]\!]^{V_2} = \text{a financial institution}$
2. *Correspondence*: different terminology, same distinction
   $[\![\text{car}]\!]^{V_1} = [\![\text{automobile}]\!]^{V_2} = \text{road vehicle}$
3. *Conflict*: same terminology, different distinction.
   $[\![\text{bank}]\!]^{V_1} = \text{financial institution}$, $[\![\text{bank}]\!]^{V_2} = \text{land alongside a river}$
4. *Contrast*: different terminology, different distinction.

Terminological ambiguity

Why are consensus, correspondence, . . . relevant to RE?

Example

As a student, I want to see my professors' research profile

As a head of department, I want to review the lecturers' research outputs, so that I can perform my yearly assessment.

Terminological ambiguity

Why are consensus, correspondence, . . . relevant to RE?

Example

As a student, I want to see my professors' research profile

As a head of department, I want to review the lecturers' research outputs, so that I can perform my yearly assessment.

$$[\![\text{professor}]\!]^{V_{Stud}} \stackrel{?}{=} [\![\text{lecturer}]\!]^{V_{HoD}}$$

Terminological ambiguity

Why are consensus, correspondence, . . . relevant to RE?

Example

As a student, I want to see my professors' research profile

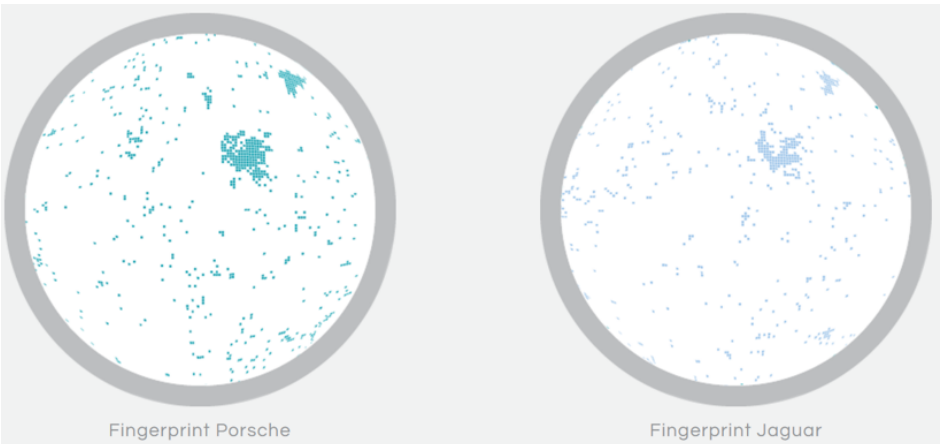
As a head of department, I want to review the lecturers' research outputs, so that I can perform my yearly assessment.

$$[\![\text{professor}]\!]^{V_{Stud}} \stackrel{?}{=} [\![\text{lecturer}]\!]^{V_{HoD}}$$
$$[\![\text{research profile}]\!]^{V_{Stud}} \stackrel{?}{=} [\![\text{research outputs}]\!]^{V_{HoD}}$$

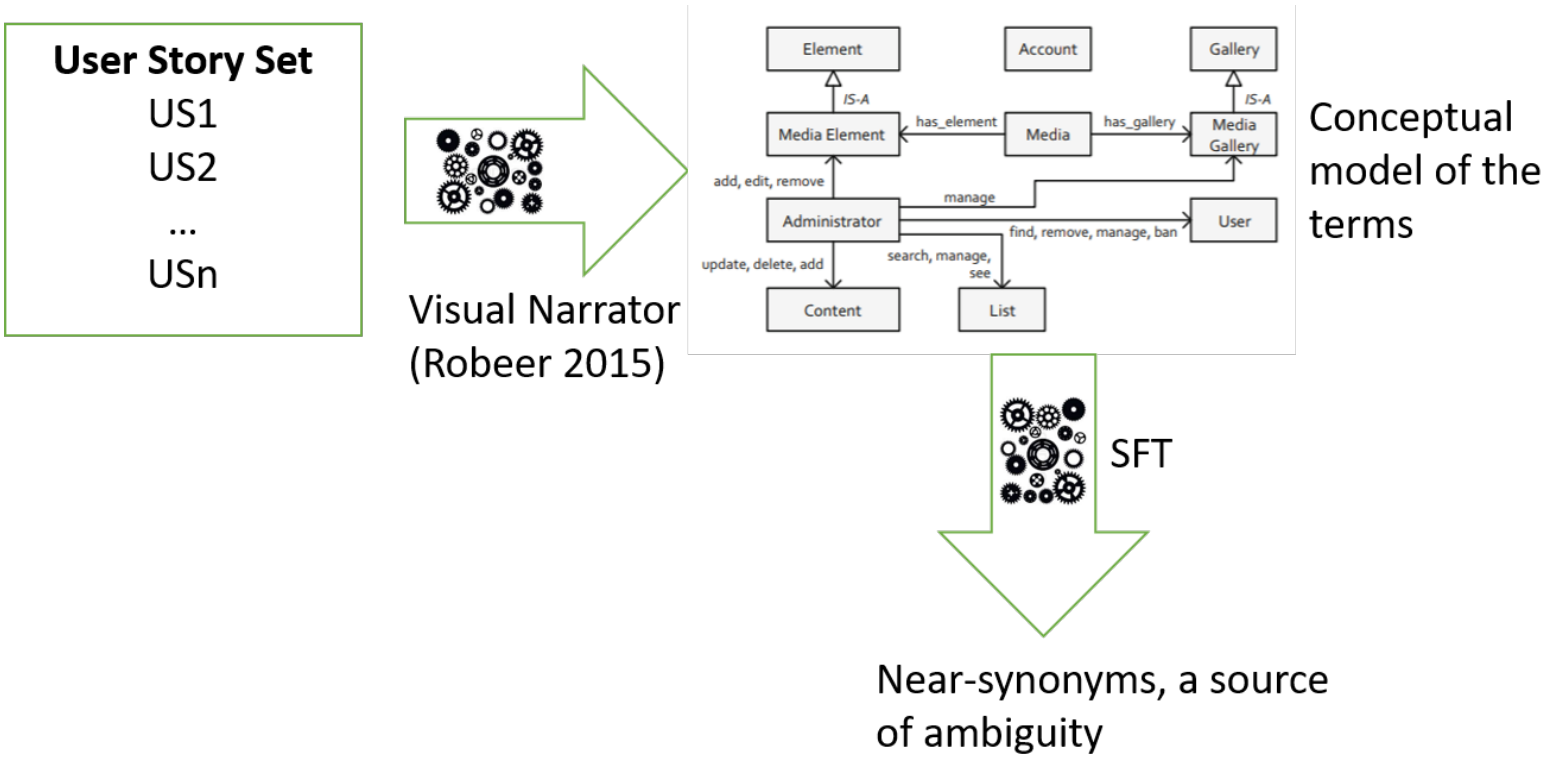
The intelligence, basics

We use Semantic Folding Theory (SFT):

- Every term is associated a semantic fingerprint
- Fingerprints are created by analyzing huge amounts of text
- Similar fingerprints indicate similar terms



The intelligence, applied



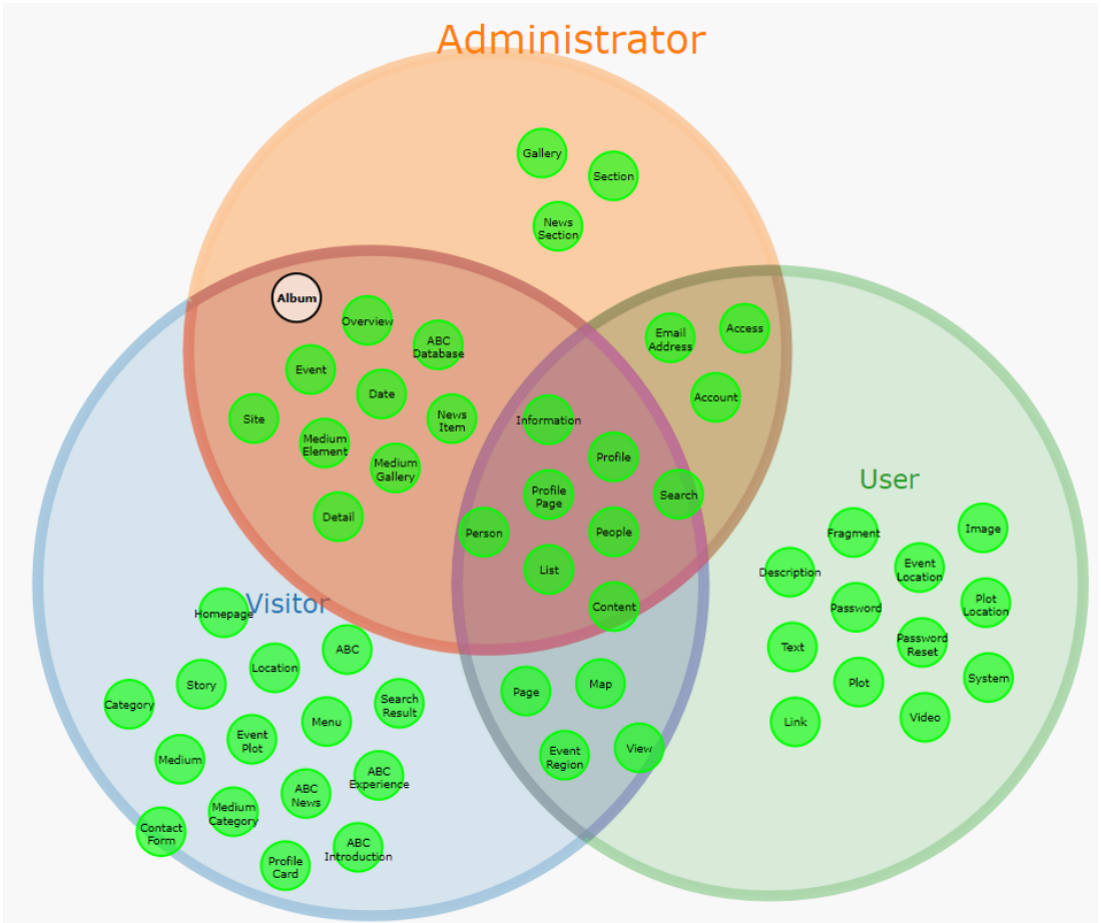
The intelligence, near-synonyms

- Given two terms t_1 and t_2

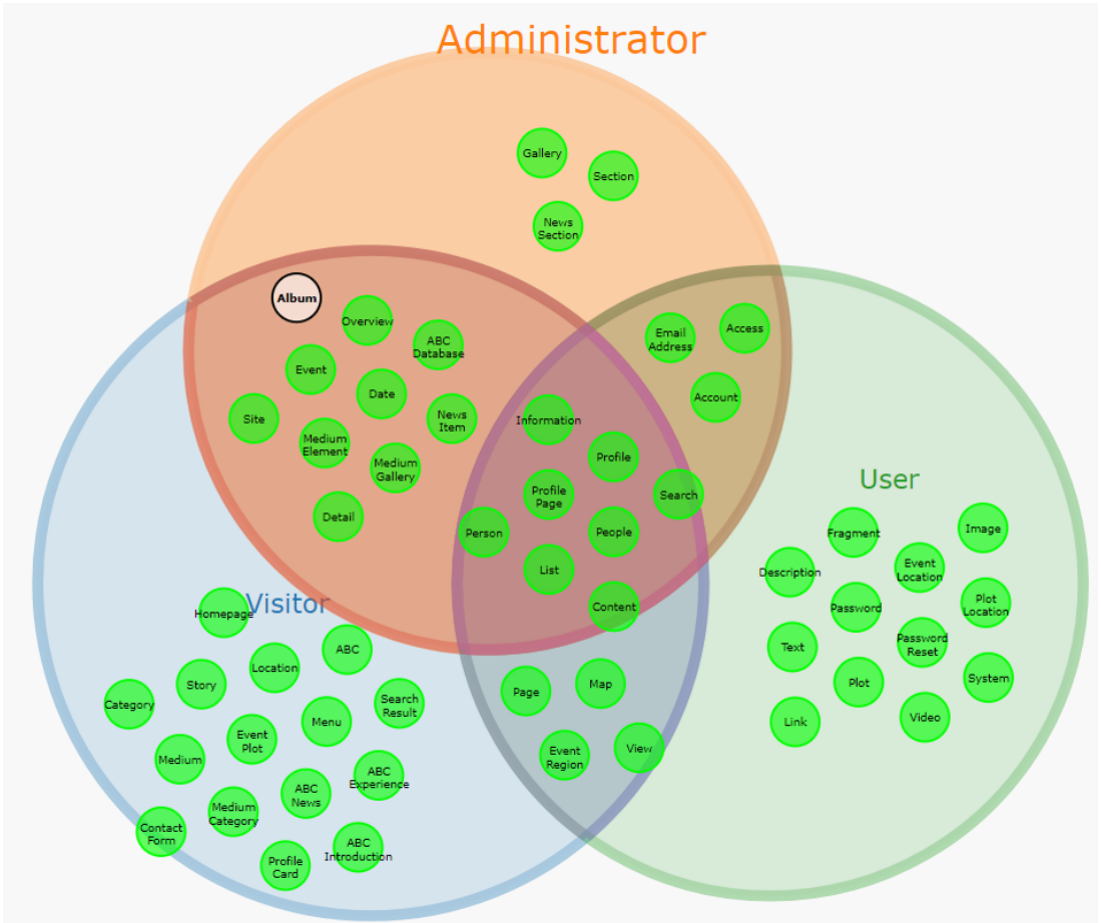
$$ambig_{t_1,t_2} = \frac{2 \cdot sim_{t_1,t_2} + simc_{t_1,t_2}}{3}$$

- A combination of term similarity and context similarity
  - 2/3 term similarity (car-automobile, etc.)
  - 1/3 context similarity: stories where exactly one of the terms appears
- Weights assessed via a correlation study with humans

The human side, information visualization

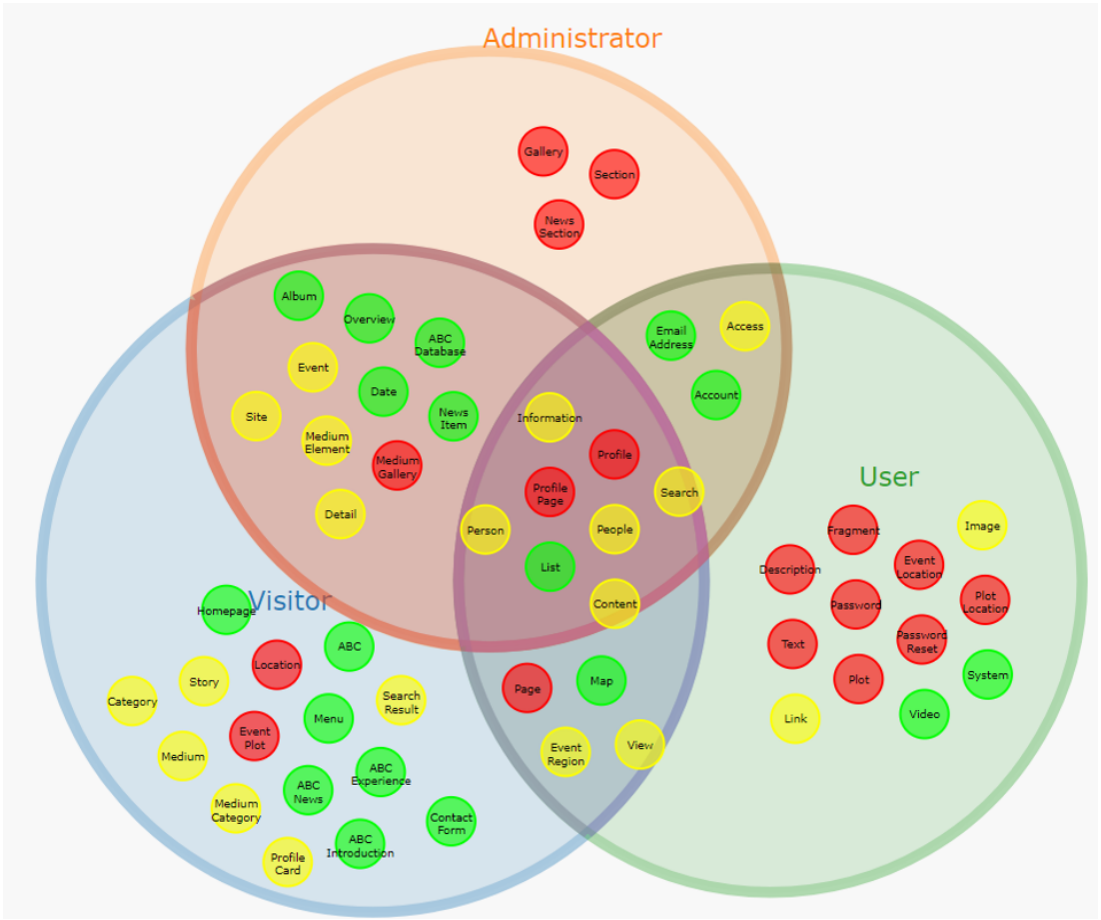


The human side, information visualization

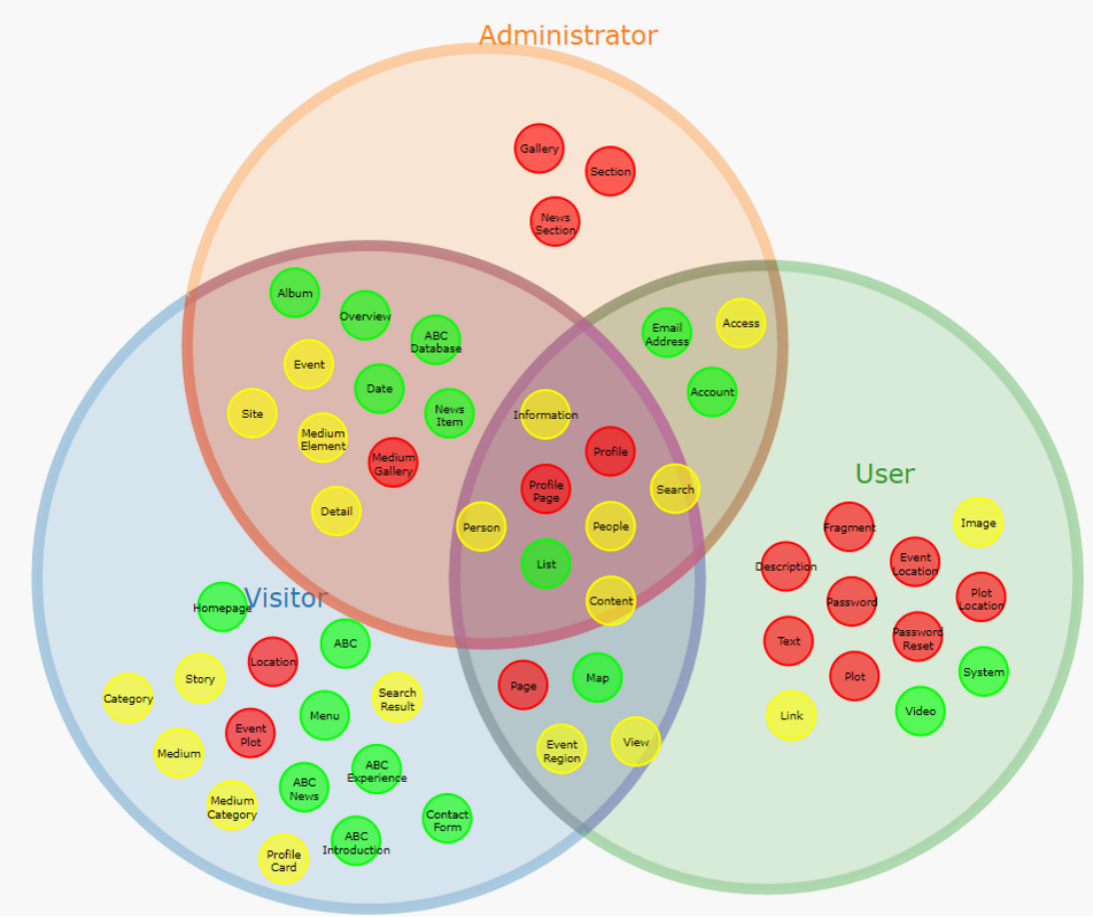


The intersecting areas show terms used by multiple roles

Highlighting *possible* ambiguity



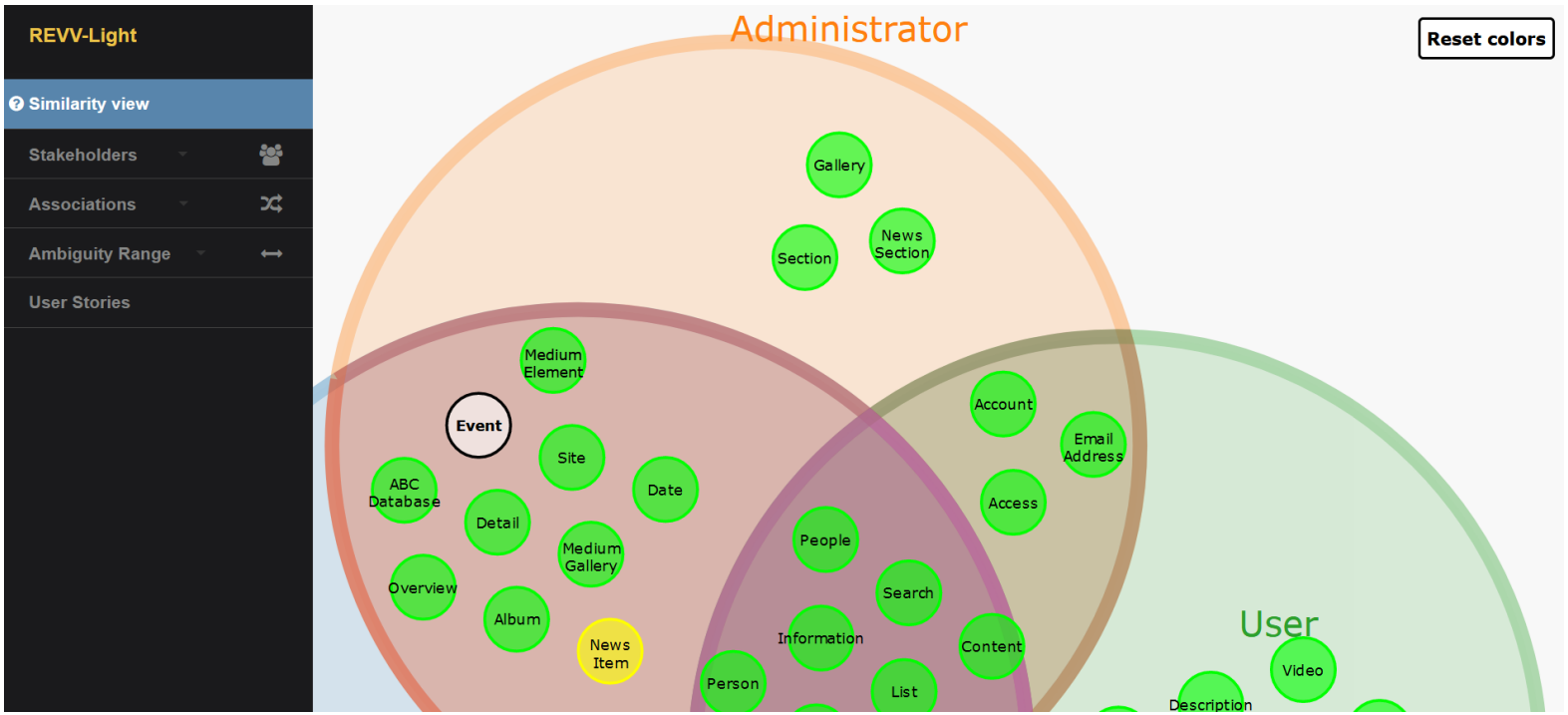
Highlighting *possible* ambiguity



Ambiguity between terms couples is calculated as described before

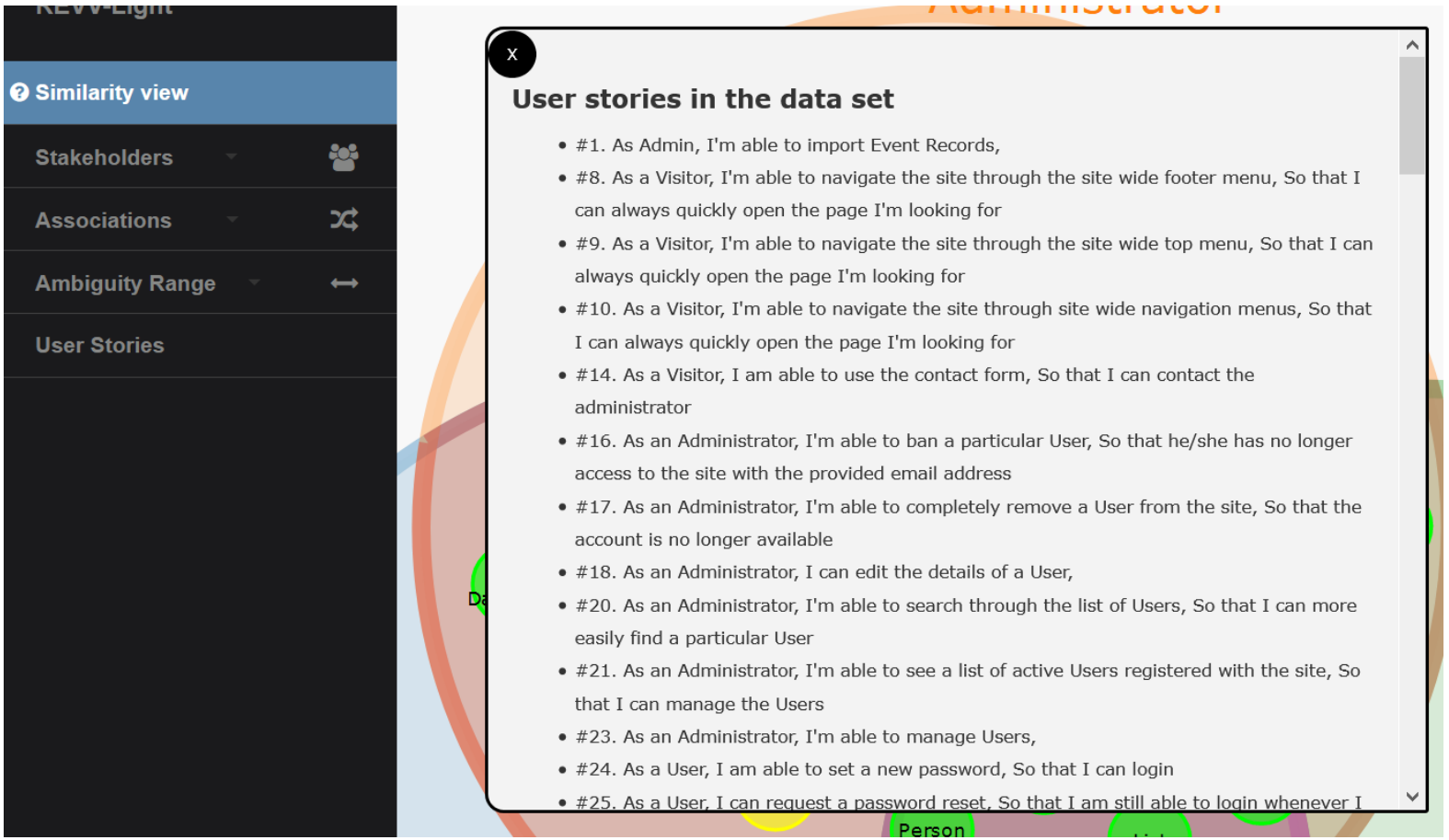
Tool support: REVV-Light

https://bit.ly/2Mn35dK



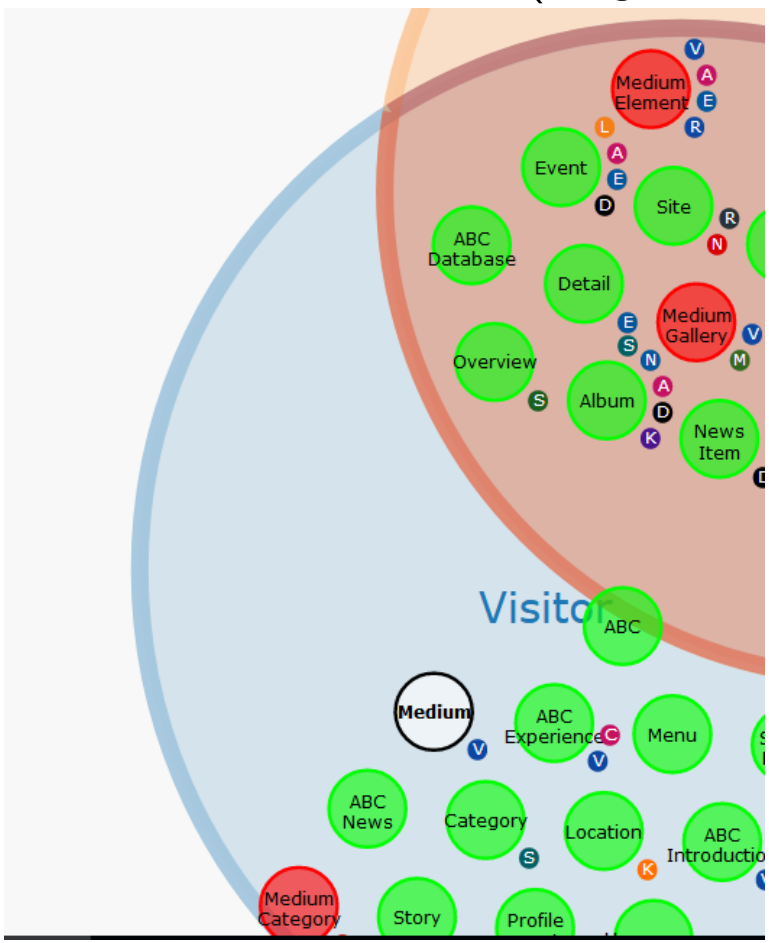
Scan the entire data set

. . . the user stories *correctly* parsed by the Visual Narrator



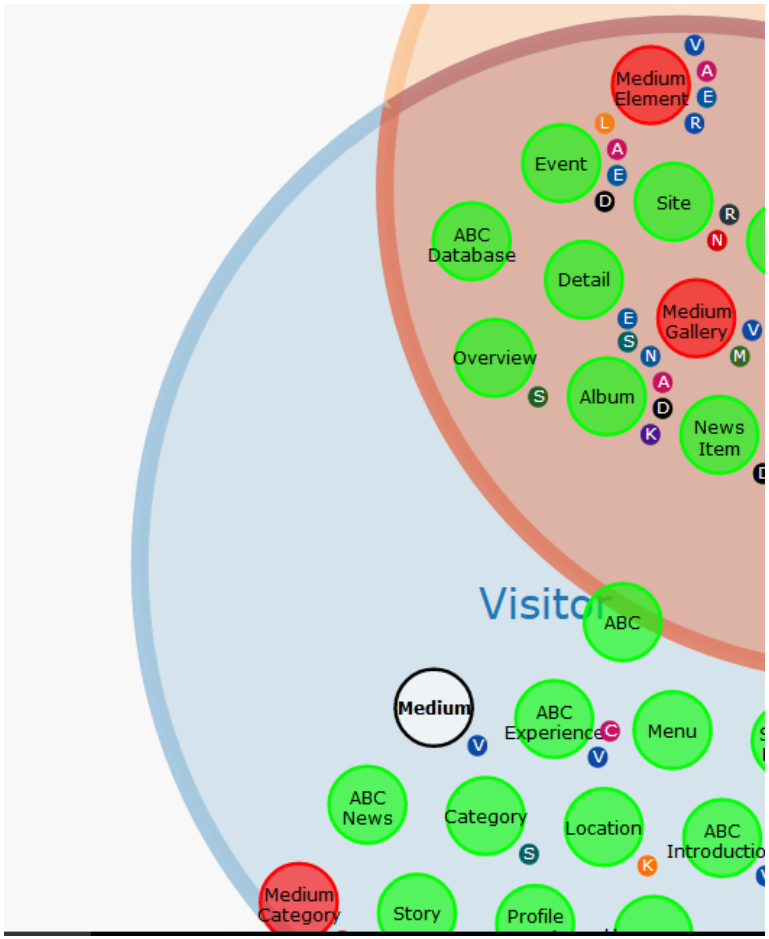
Analyzing ambiguity

Focus on one element (single click)

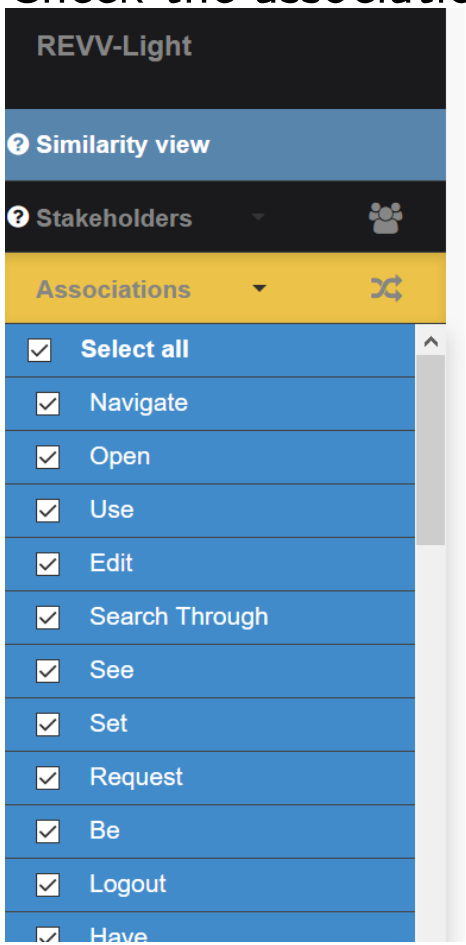


Analyzing ambiguity

Focus on one element (single click)

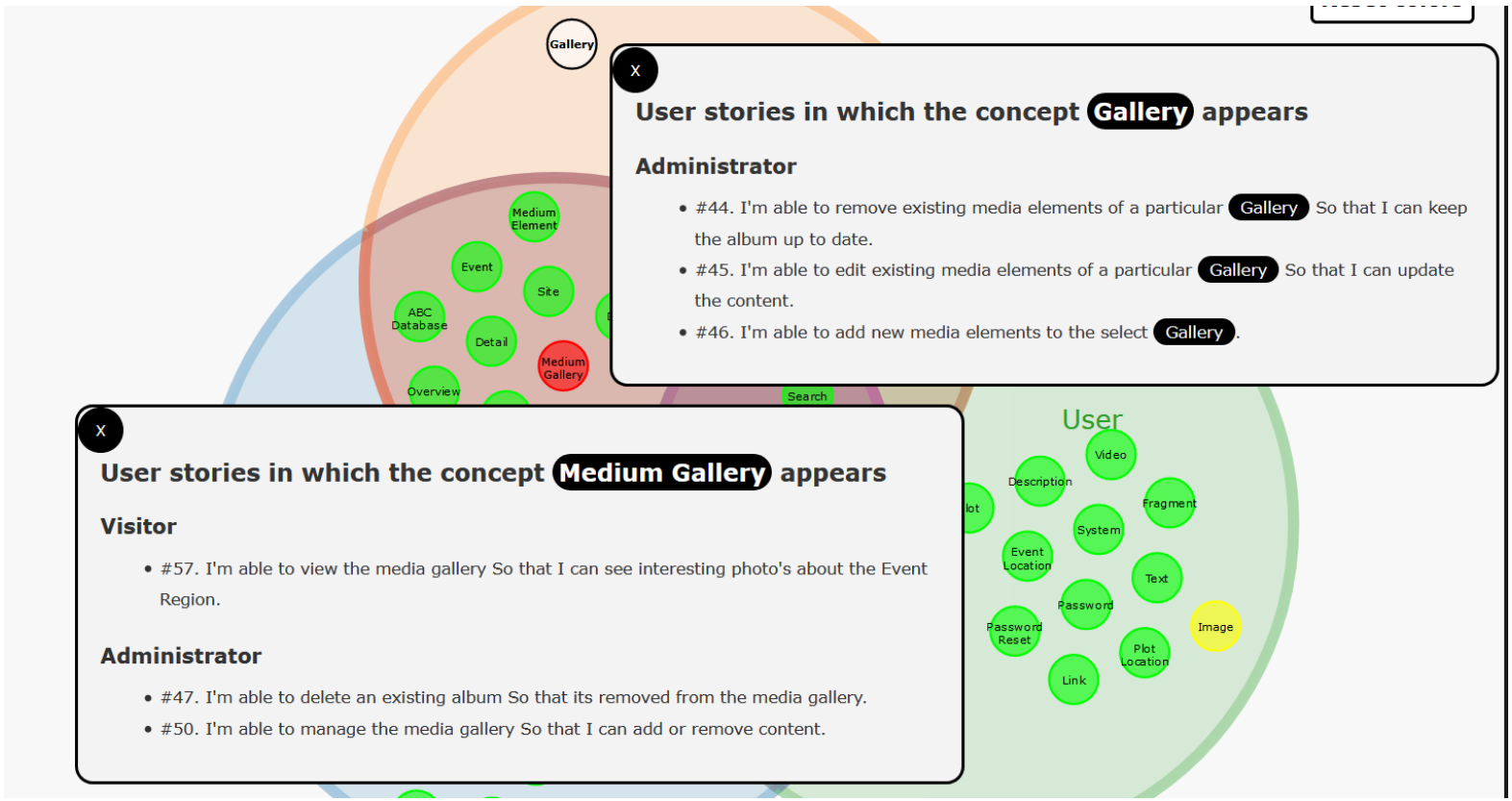


Check the associations



Check the user stories for a term

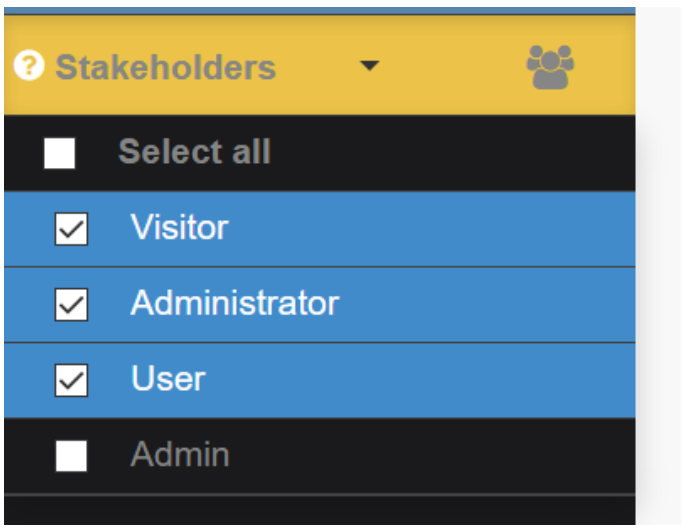
Is an ambiguity real?



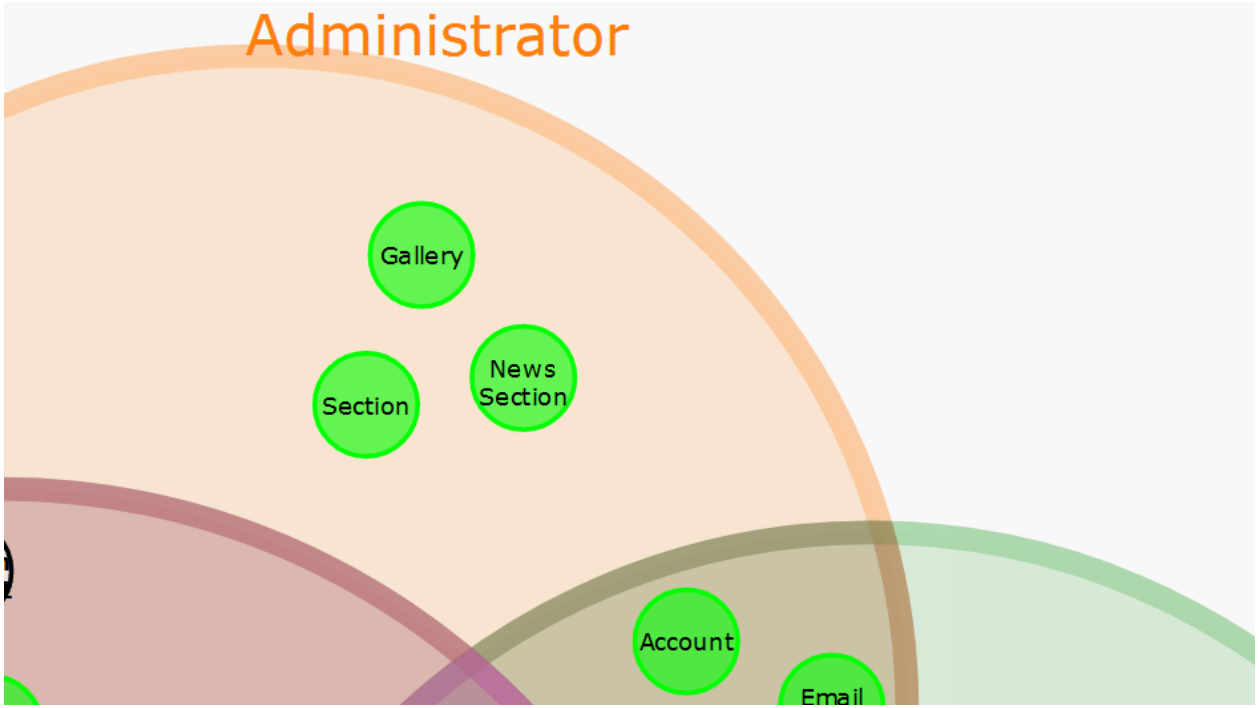
Double click on a term!

Analyzing ambiguity

Focus only on certain roles



Side-product: analyzing incompleteness



Possible incompleteness: no user stories about *Gallery*, *Section*, *News Section* for roles "User" and "Visitor"?

Exercise #5

Analyze ambiguity with REVV-Light

- Explore one of the four data sets
- Use the functions of the tool to examine ambiguity between
  - Nouns and compound nouns
  - Verbs / associations
- You can use the printed user stories as a help
- Optionally, look at incompleteness too!

RE 2018 Tutorial

Camper+: your ideal camp solution
ALFRED: independent living and active ageing
University of Bath institutional data repository
Cornell Photos image library

Exercise #5

Review

Exercise evaluation

- What were the main difficulties?
- Was the tool useful?
- What function of the tool was mostly useful?
- What did you miss?

Table of Contents

Summary

What have I learned today?

Agile RE with user stories

- User stories are concise, to-the-point, popular

Summary

What have I learned today?

Agile RE with user stories

- User stories are concise, to-the-point, popular
- Writing good user stories is not that simple

Summary

What have I learned today?

Agile RE with user stories

- User stories are concise, to-the-point, popular
- Writing good user stories is not that simple
- Semi-automated exist to support the task, for example

Summary

What have I learned today?

Agile RE with user stories

- User stories are concise, to-the-point, popular
- Writing good user stories is not that simple
- Semi-automated exist to support the task, for example
  - The QUS framework and the AQUSA tool [LDvdWB16b]

Summary

What have I learned today?

Agile RE with user stories

- User stories are concise, to-the-point, popular
- Writing good user stories is not that simple
- Semi-automated exist to support the task, for example
  - The QUS framework and the AQUSA tool [LDvdWB16b]
  - The Visual Narrator tool [LRD+17]

Summary

What have I learned today?

Agile RE with user stories

- User stories are concise, to-the-point, popular
- Writing good user stories is not that simple
- Semi-automated exist to support the task, for example
  - The QUS framework and the AQUSA tool [LDvdWB16b]
  - The Visual Narrator tool [LRD+17]
  - The REVV-Light tool [DvdSL18]

Summary

What have I learned today?

Agile RE with user stories

- User stories are concise, to-the-point, popular
- Writing good user stories is not that simple
- Semi-automated exist to support the task, for example
  - The QUS framework and the AQUSA tool [LDvdWB16b]
  - The Visual Narrator tool [LRD+17]
  - The REVV-Light tool [DvdSL18]
- Tools assist human requirements engineers, do not replace them!

Contribute to our research!

What can I do?

We aim to improve the tooling to make impact on agile RE practices

What can I do?

- Provide us with user story sets
- Use the tools in your practice
- Adapt and extend the tools (open source)
- Tell us your problems

Discussion and thank you

Q&A

Open questions or suggestions?

Contact me at f.dalpiaz@uu.nl

Discussion and thank you

Q&A

Open questions or suggestions?

Contact me at f.dalpiaz@uu.nl

References I


Fabiano Dalpiaz, Ivor van der Schalk, and Garm Lucassen, *Pinpointing Ambiguity and Incompleteness in Requirements Engineering via Information Visualization and NLP*, Proceedings of the 24th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'18), 2018.


Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, and Shahaboddin Shamshirband, *A systematic literature review on agile requirements engineering practices and challenges*, Computers in Human Behavior 51 (2015), 915–929.


Mohamad Kassab, *The changing landscape of requirements engineering practices over the past decade*, Proceedings of the International Workshop on Empirical Requirements Engineering (EmpiRE), IEEE, 2015, pp. 1–8.

References II


Garm Lucassen, Fabiano Dalpiaz, Jan Martijn van der Werf, and Sjaak Brinkkemper, *The Use and Effectiveness of User Stories in Practice*, Proceedings of the 22nd International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'16), LNCS, vol. 9619, 2016, pp. 205–222.


Garm Lucassen, Fabiano Dalpiaz, Jan Martijn E.M. van der Werf, and Sjaak Brinkkemper, *Improving agile requirements: the Quality User Story framework and tool*, Requirements Engineering 21 (2016), no. 3, 383–403.


Garm Lucassen, Marcel Robeer, Fabiano Dalpiaz, Jan Martijn E. M. van der Werf, and Sjaak Brinkkemper, *Extracting conceptual models from user stories with Visual Narrator*, Requirements Engineering 22 (2017), no. 3, 339–358.

References III


Mildred LG Shaw and Brian R Gaines, *Comparing conceptual structures: consensus, conflict, correspondence and contrast*, Knowledge Acquisition 1 (1989), no. 4, 341–363.


Xinyu Wang, Liping Zhao, Ye Wang, and Jie Sun, *The role of requirements engineering practices in agile development: An empirical study*, Proceedings of the Asia Pacific Requirements Engineering Symposium (APRES), vol. 432, 2014, pp. 195–209.

Behavior-Driven Development (BDD)

Making user stories testable

BDD promotes to write acceptance tests that

- Complement the who, what, and why parts
- Determine when a user story is fulfilled
  - Given some context
  - When some action is carried out
  - Then a set of observable consequences occurs

Behavior-Driven Development (BDD)

Making user stories testable

BDD promotes to write acceptance tests that

- Complement the who, what, and why parts
- Determine when a user story is fulfilled
  - Given some context
  - When some action is carried out
  - Then a set of observable consequences occurs

Example

Given the user is interested in the RE field,
When a new paper is published in the RE conference
And the user is not an author,
Then the user is notified of such paper.

QUS in practice: improving the user stories

9. Conflict-free

Conflict-free

A user story should not be inconsistent with any other user story

Example (Violation)

1. As an Event Organizer, I'm able to edit any event
2. As an Event Organizer, I'm able to delete only the events that I added

QUS in practice: improving the user stories

9. Conflict-free

Conflict-free

A user story should not be inconsistent with any other user story

Example (Violation)

1. As an Event Organizer, I'm able to edit any event
2. As an Event Organizer, I'm able to delete only the events that I added

⇓ *(change 1)*

As an Event Organizer, I'm able to edit events that I added

QUS in practice: improving the user stories

10. Estimable

Estimable

A user story does not denote an unrefined requirement that is difficult to plan and prioritize

Example (Violation)

As an Event Organizer, I want to see my task list during the event, so that I can prepare myself (e.g., I can see when I should leave home)

QUS in practice: improving the user stories

10. Estimable

Estimable

A user story does not denote an unrefined requirement that is difficult to plan and prioritize

Example (Violation)

As an Event Organizer, I want to see my task list during the event, so that I can prepare myself (e.g., I can see when I should leave home)

$\Downarrow$ *(split)*

1. As an Event Employee, I want to see my task list during the event, so that I can prepare myself
2. As an Event Organizer, I want to upload a task list for event employees

QUS in practice: improving the user stories

11. Unique

Unique

Every user story is unique, duplicates are avoided

Example (Violation)

1. As a Visitor, I'm able to see new events, so that I stay up to date
2. As a Visitor, I'm able to see new events, so that I stay up to date

QUS in practice: improving the user stories

11. Unique

Unique

Every user story is unique, duplicates are avoided

Example (Violation)

1. As a Visitor, I'm able to see new events, so that I stay up to date
2. As a Visitor, I'm able to see new events, so that I stay up to date

⇓ *(remove one)*

1. As a Visitor, I'm able to see new events, so that I stay up to date