Unit 2

UX Design Requirements

UX design requirements

- User stories –
- UX Design requirements –
- validating user stories and requirements

1. User Stories

- User stories are an agile way of capturing what is to be designed.
- User stories:

1. Involve user inputs.
2. Are stated in the perspective of usage.
3. Are stated from a customer perspective.
4. Were made popular by agile proponents.
5. Are now the de facto standard in industry.
6. Are focused on delivering chunks of meaningful capabilities to users

1. User Stories

- Choose user stories, especially for an agile process.
- UX mentality is all about value design over requirements.
- Rather than even thinking about requirements, we should think about how great products “require vision, perseverance, and the ability to do amazing things in the face of perceived impossibility”

1. User Stories

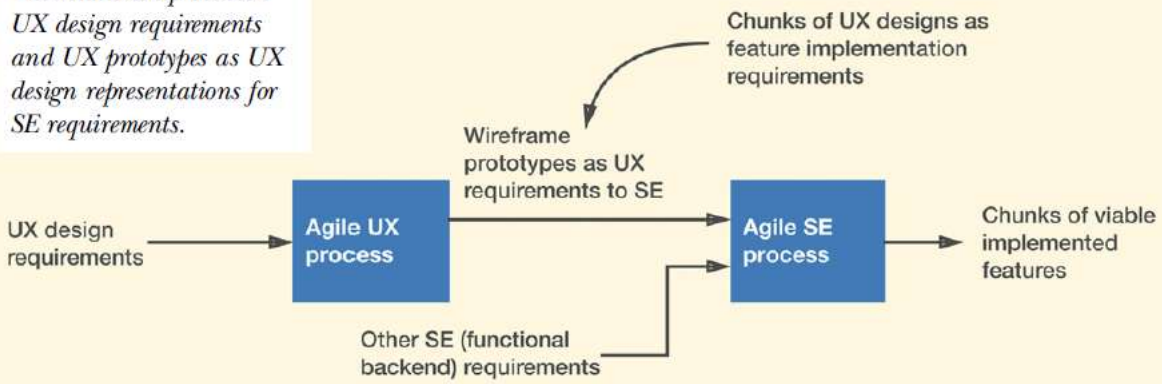
- Requirements for:

1. The UX parts.
2. The functional parts.

1. User Stories

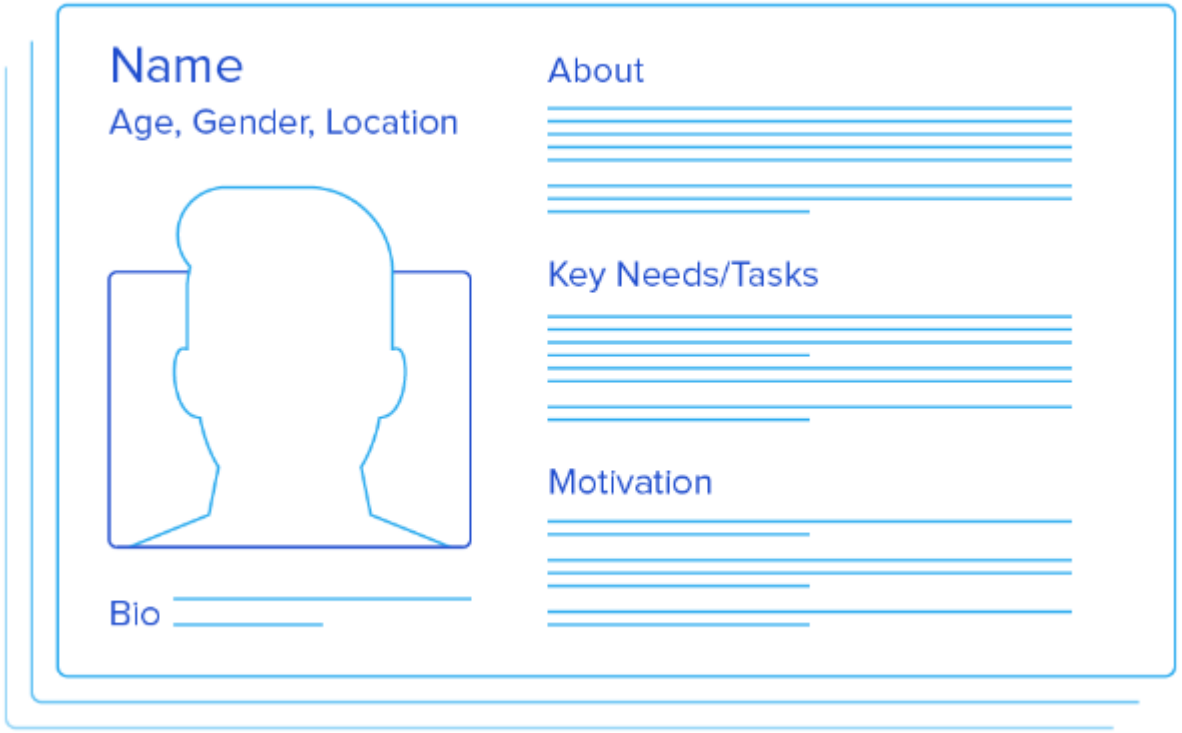
- UX prototypes—usually wireframe prototypes— are the vehicle for conveying requirements for the UX parts to the implementers.

Fig.
The relationship between UX design requirements and UX prototypes as UX design representations for SE requirements.



1. User Stories

- Close relationship of requirements in UX and SE



1. User Stories

- User stories are the de facto approach to both UX design and SE implementation requirements in late-funnel agile development.
- It’s unlikely that users would present us with a complete set of requirements. We, as UX research analysts, must write additional user stories to fill in the gaps.

User stories

Example 1: An online gamer

"As an online gamer, I want to have a multiplayer option so that I can play online with friends."

Example 2: A design team lead

"As a design team lead, I want to organize assets, so I can keep track of multiple creative projects."

Example 3: An e-commerce shopper

"As an e-commerce shopper, I want to filter my searches so I can find products quickly."

Now that you know what a user story looks like, you can get to work creating one.

1. User Stories

User Story Template

When writing a user story, remember that user stories follow a standard template:

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

1. User Stories

- What is a User Story?
- A user story is a short narrative describing a feature or capability needed by a user in a specific work role and a rationale for why it is needed, used as an agile UX design “requiremen
- A user story is a kind of small-scope requirement

  define a user story as a narrative in the user’s voice describing, “the smallest unit of work expressed as a benefit to the end user”
- A user story should fit on one 300 500 card.

1. User Stories

- Writing a User Story
- As a <relevant user work role> I want to <the desired small-scope capability in everyday product or system usage> So that <the reason why, the added value this capability will deliver>.

1. User Stories

- Example: User Stories for TKS
- raw data note:

1. I think of sports events as social events, so I like to go with my friends. The problem is that we often have to sit in different places, so it is not as much fun. It would be better if we could sit together.

- extract the essence of a user story

1. As a student ticket buyer I want an MU basketball ticket-buying option to get several seats together So I can sit with my friends.

1. User Stories

- Extrapolation Requirements in User Stories:
- Suppose ticket buyers expressed the need to search for events based on a predetermined criterion but said nothing about browsing events to see what is available.
- So you might write an extrapolation requirement to include the obvious need also to browse events as an extrapolation of the previous user story about being able to search for events by keywords.

As a ticket buyer
I want to be able to browse events with filters for category, description, location, time, rating, and price
So I can find relevant events in context.

1. User Stories

- Example: Inputs to User Stories as UX Design Requirements for the TKS

Transaction Flow

Existence of feature

I want to be able to pause, save, and later resume a ticket-buying transaction.

System implications: Will require accounts, including logging in and out, etc.

Transaction Flow

Timeouts

I want the system to have a timeout feature that removes my transaction from the screen if I haven't made any actions after a certain amount of time.

So it will protect my privacy.

Extrapolation: Ticket buyer shall be made aware of the existence and status of timeout, including progress indicator showing remaining time and audible beep near the end of the time-out period.

1. User stories

Revisiting previous steps (going back)

I want to be able to revisit a previous step easily.

So I don't have to go through multiple back actions.

Transaction progress awareness

I want to be able to track the progress of an entire transaction (what is done and what is left to do) using a "bread crumb trail."

So I don't have any confusion as to what I still have to do for a successful completion.

User reminders

I want to receive a reminder to take the ticket(s) and credit card at the end of each transaction.

So I don't forget and walk away, thinking I have completed the transaction.

Checkout

I want to see, before making a payment, a confirmation page showing exactly what is being purchased and the total cost.

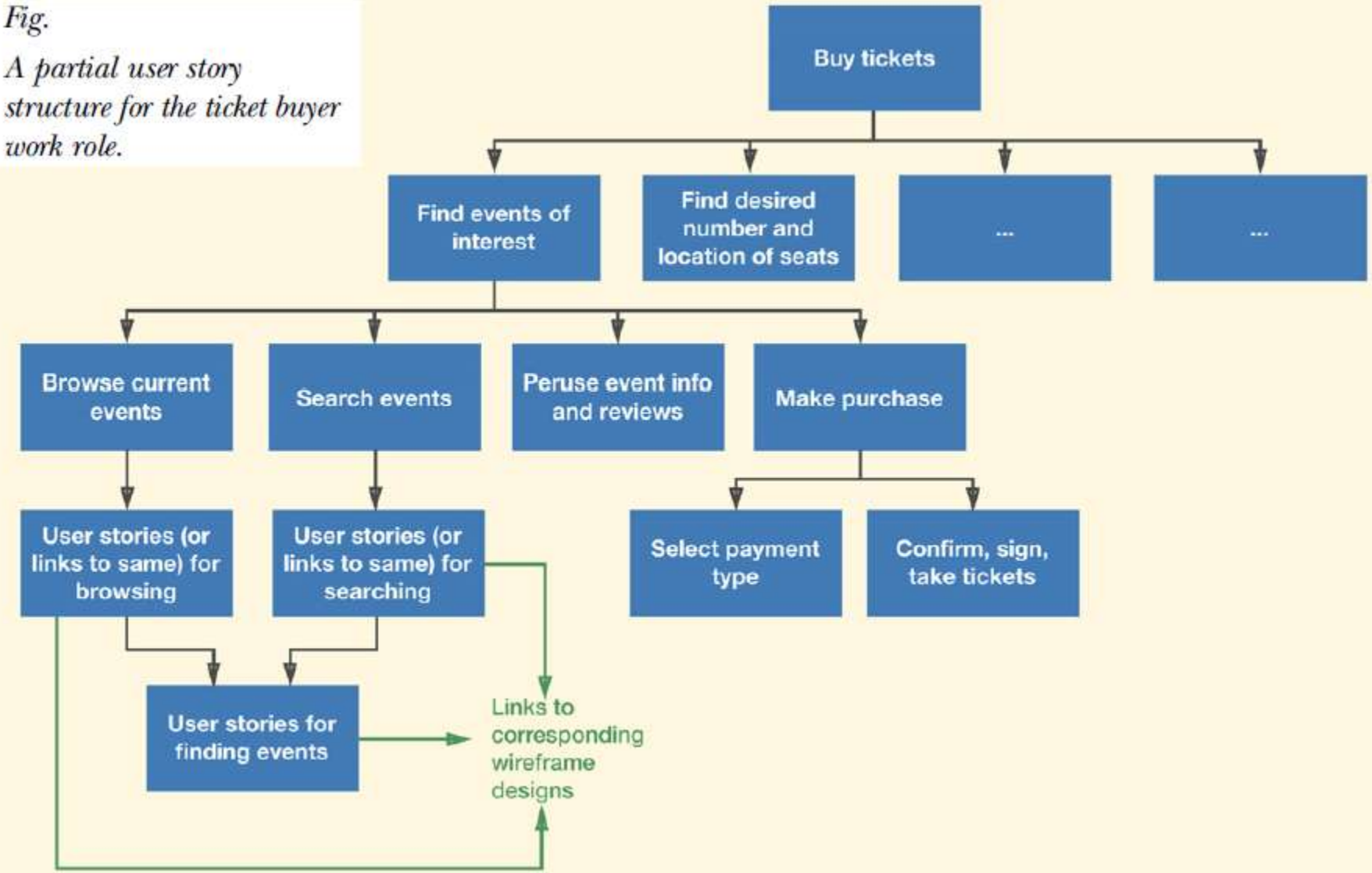
So I can be confident in what I am purchasing.

1. User stories

- Organize Sets of User Stories for Use in UX Design
- HTI is also a way to manage the collection of user stories and helps us maintain completeness of the collection.

Fig.

A partial user story structure for the ticket buyer work role.



1. User Stories

- Example requirement notes

Searching

83: I would like to be able to search for events by price, artist, venue, and/or date.

104: If I am looking for a specific band, I expect a Google-style search where I type in the name of the band.

112: It would be really nice to be able to search for interesting events say within two blocks from where I am currently.

Browsing

74: I would like to be able to browse all events on the kiosk.

106: I would like to be able to browse by different topics (type of event). For example, in big cities, even in music there will be different genres.

107: I would like to be able to browse by locations.

Sorting

75: I would like to be able to sort the results I find using some criteria.

The corresponding user stories are easy to visualize as:

1. User stories

Searching

User Story TKS 83:

As a ticket buyer
I would like to be able to search for events by price, artist, venue, and/or date
Because I want maximum flexibility in searching.

User Story TKS 104:

As a ticket buyer
I expect a Google-style search where I type in, for example, the name of a band
Because I don't want to be constrained by special formats in searches.

User Story TKS 112:

As a ticket buyer
I want to be able to search for events by proximity to a given location
Because I seek maximum power in searching.

1. User stories

- Prioritizing User Stories for Design and Development:
- Prioritization is guided by a business-oriented concept, namely of delivering minimum viable product (MVP) releases—that is, to produce something initially, however limited, that can be deployed immediately and evaluated to get feedback from actual use.

2. UX DESIGN REQUIREMENTS

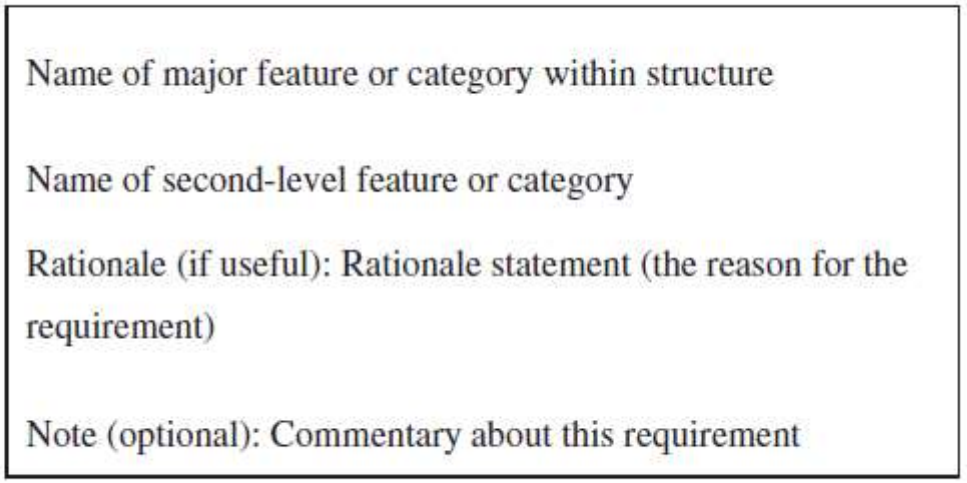
- The Requirements Structure Evolves:
- The hierarchical categories will evolve as needed to accommodate each new elemental data note.
- Compose Requirements Statements:
  - Ask what user needs are reflected by the note.
  - Write it as a requirement statement.
  - Find where it fits into the evolving requirements structure

2. UX design requirements

- The Requirement Statement and Requirements Document

Fig.

Generic structure of a requirement statement.



2. UX design requirement

Security

Privacy of ticket-buyer transactions

Shall protect security and privacy of ticket-buyer transactions.

Note: In design, consider timeout feature to clear screen between customers.

Fig.
Example requirement statement.

3. VALIDATING USER STORIES AND REQUIREMENTS

- Coordinating Requirements, Terminology, and Consistency
- Take User Stories and Requirements Back to Customers and Users for Validation
- Resolve Organizational, Social, and Personal Issues Arising Out of Work Practice Changes