OST
Ostschweizer
Fachhochschule

Collaborative Modeling Lesson 4: User Story Mapping and Splitting

ITBO Learning Lab 2, Initiative 1 (ZIOL, KAPS)

Spring Term 2025

Departement Informatik and ITBO

Agenda

- Motivation
- Concepts and activities
- Example

- Exercise
- Lab
- More information

Context and Motivation (1/3)

"Flat backlogs don't work for me"
(Jeff Patton)

One of the more troubling things, for me, about common Agile development practice is the idea of the flat user story backlog. It just seems like a dumb idea to me. Well not completely dumb, in that if I'm going to build software incrementally in small pieces, I'll need to figure out what piece to start with. The backlog puts those pieces in build order – from highest value to lowest value – which is great if you're a project manager – which I'm not.

I'm writing this from the plane as I fly back from a client site. We had what I feel like was a pretty successful couple days of story writing and release planning. I had fun – and I was pleased to see the group really get into it and feel good about what they came up with. I've been using a practice I [now] call story mapping. I first wrote about this approach in the article "How You Slice It." The article was published in January 2005 – but by the time it was written, I'd been using the practice for a couple years.

Source: The New User Story Backlog is a Map, Jeff Patton, jpattonassociates.com/the-new-backlog

Context and Motivation (2/3)

Why Story Splitting Matters

Working from a prioritized backlog of small user stories allows a team to get value and high-quality feedback on frequent intervals. Many teams struggle to split large user stories and features into good, small stories. Instead of ending up with small vertical slices through their architecture, they get stories that look more like tasks or architectural components and fail to experience the value or feedback small stories should provide.

Fortunately, story splitting is a skill that can be learned in a relatively short time. We've seen teams go from struggling to fluently splitting stories with just a couple hours of practice and some simple tools. Later, we'll look at how you can structure that practice.

https://www.humanizingwork.com/the-humanizing-work-guide-to-splitting-user-stories/

Context and Motivation (3/3)

- **Small user stories** allow a team to get **value** and **high-quality feedback**
  - On frequent intervals
- Teams often **struggle to split** large stories and features
- What **you want**:
  - small vertical slices through your architecture
- What you **don‘t want**:
  - stories that look like tasks or architectural components

Source: The Humanizing Work Guide to Splitting User Stories, humanizingwork.com/the-humanizing-work-guide-to-splitting-user-stories

Concept: INVEST Properties

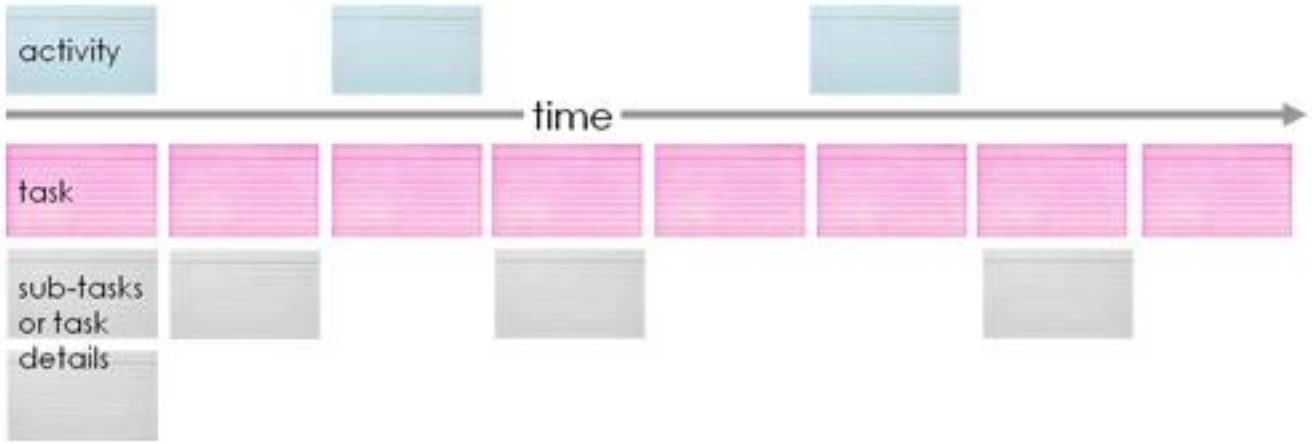
- https://agilealliance.org/glossary/invest/

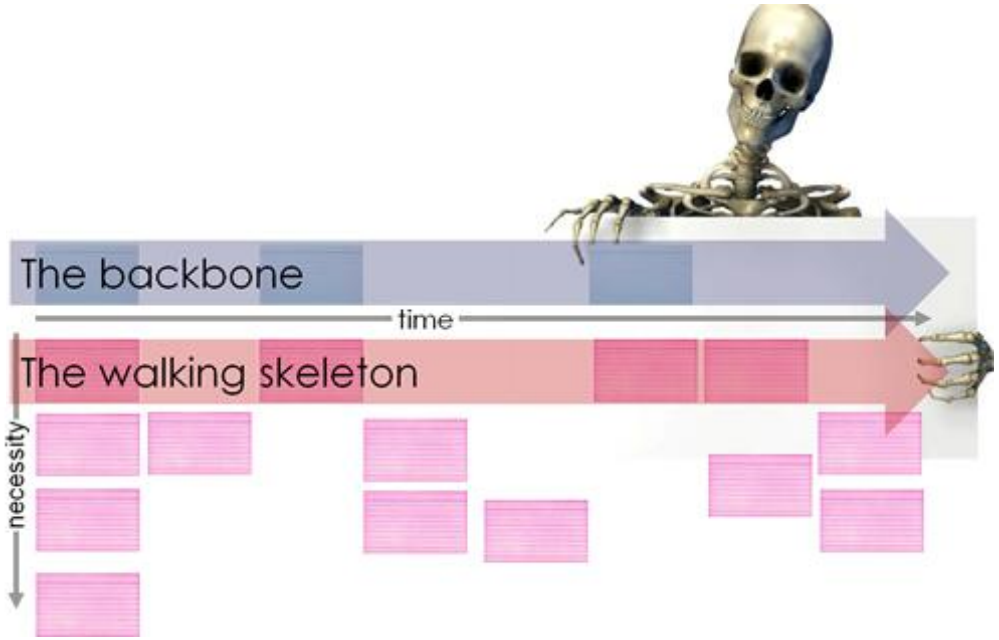
A good user story should be:

- "I" ndependent (of all others)
- "N" egotiable (not a specific contract for features)
- "V" aluable (or vertical)
- "E" stimable (to a good approximation)
- "S" mall (so as to fit within an iteration)
- "T" estable (in principle, even if there isn't a test for it yet)

Story Mapping

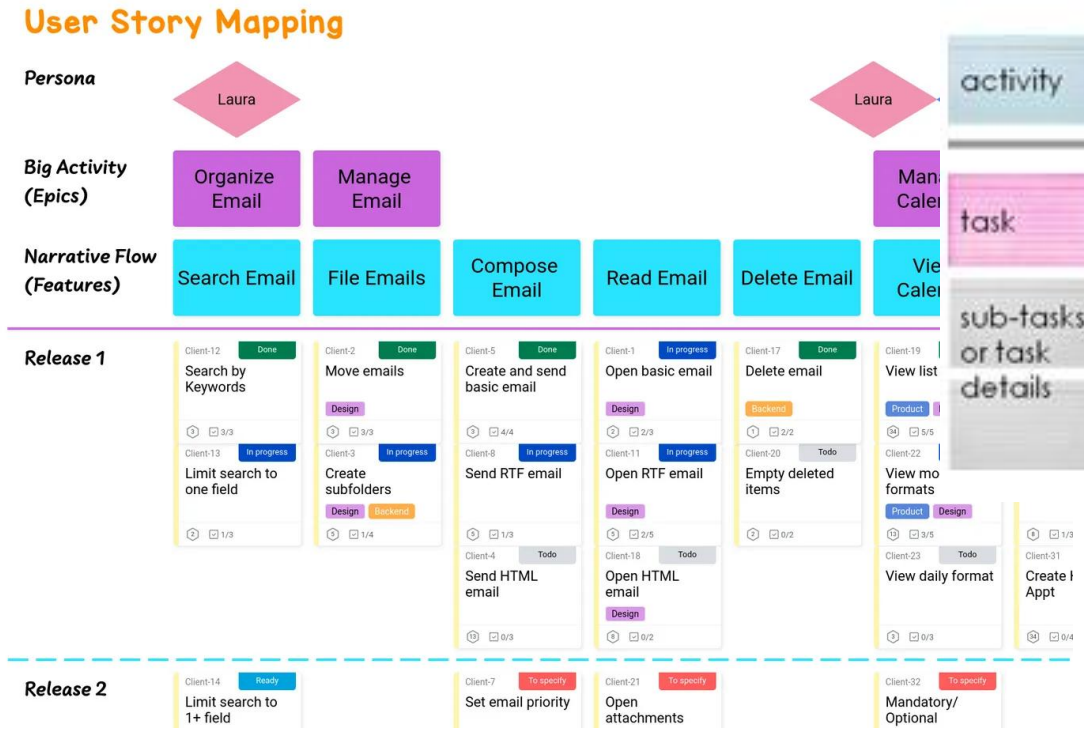
- Two dimensions: time and level of detail





Source: The Humanizing Work Guide to Splitting User Stories, humanizingwork.com/the-humanizing-work-guide-to-splitting-user-stories

Different Naming for Concepts …

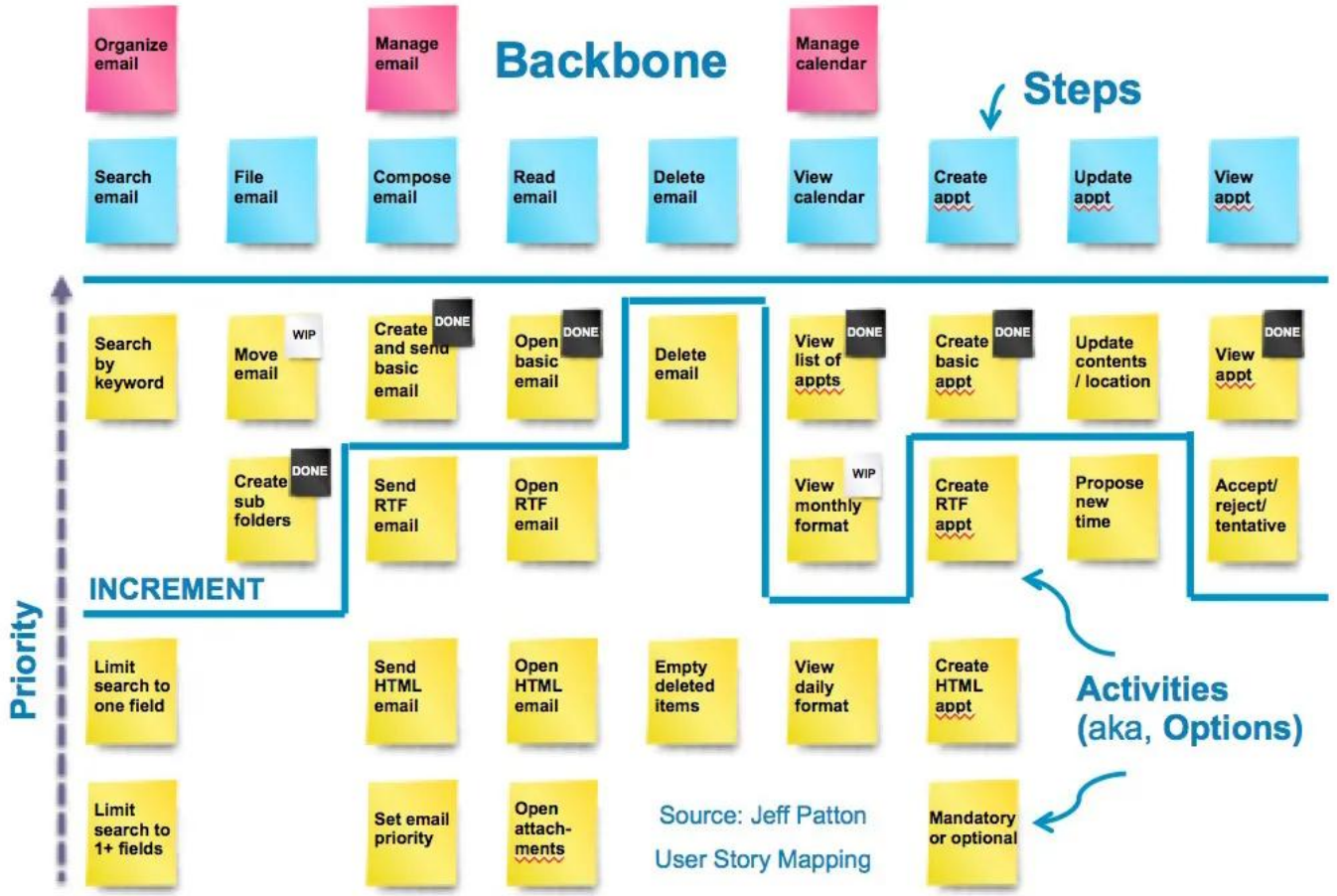


Source: draft.io/example/user-story-mapping

Source: humanizingwork.com/the-humanizing-work-guide-to-splitting-user-stories

- Activity = Epic ?
- Feature = Narrative = tasks = steps ?

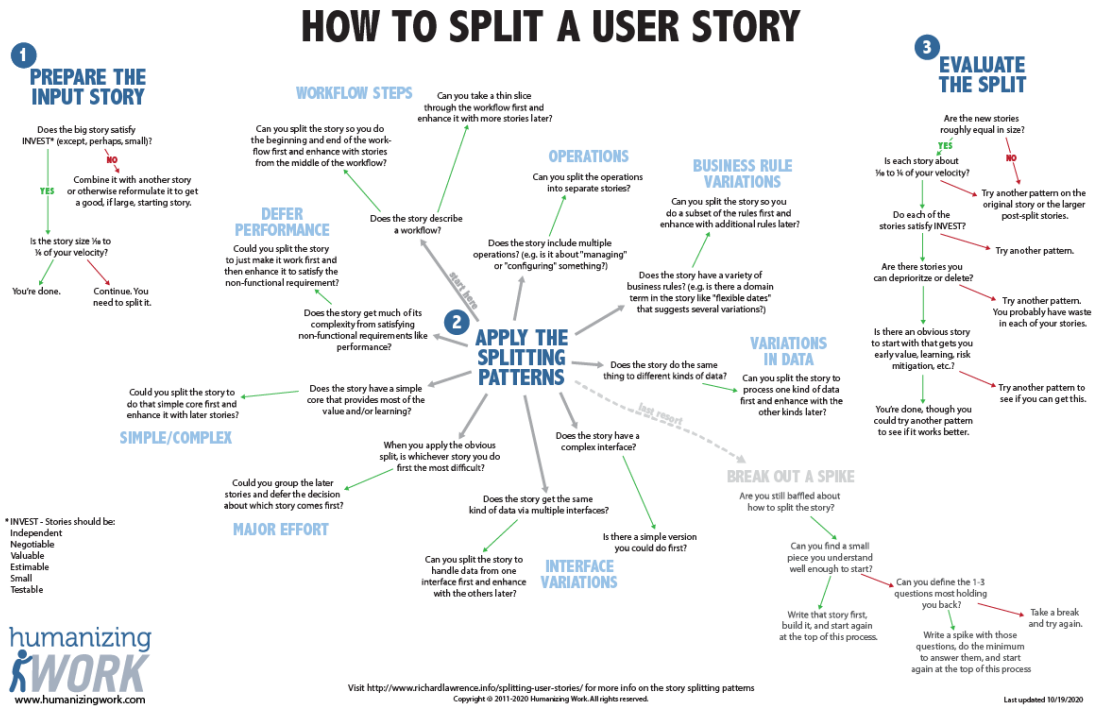
Sample Story Mapping



Source: User Story Mapping, Alin Mateescu, medium.com/i-want-to-be-a-product-manager-when-i-grow-up/user-story-mapping-dd7462ee78cf

Concept: Story Splitting Patterns

- Source: https://www.humanizingwork.com/the-humanizing-work-guide-to-splitting-user-stories/
  - Clicking on the image takes shows a larger version in the browser (Internet connection required)



HOW TO SPLIT A USER STORY

1 PREPARE THE INPUT STORY

Does the big story satisfy INVEST* (except, perhaps, small)?

NO → Combine it with another story or otherwise reformulate it to get a good, if large, starting story.

YES → Is the story size 1/10 to 1/4 of your velocity?

- You're done.
- Continue. You need to split it.

* INVEST - Stories should be:
Independent
Negotiable
Valuable
Estimable
Small
Testable

2 APPLY THE SPLITTING PATTERNS

start here

WORKFLOW STEPS

Does the story describe a workflow?

- Can you take a thin slice through the workflow first and enhance it with more stories later?
- Can you split the story so you do the beginning and end of the workflow first and enhance with stories from the middle of the workflow?

OPERATIONS

Does the story include multiple operations? (e.g. is it about "managing" or "configuring" something?)

Can you split the operations into separate stories?

BUSINESS RULE VARIATIONS

Does the story have a variety of business rules? (e.g. is there a domain term in the story like "flexible dates" that suggests several variations?)

Can you split the story so you do a subset of the rules first and enhance with additional rules later?

DEFER PERFORMANCE

Does the story get much of its complexity from satisfying non-functional requirements like performance?

Could you split the story to just make it work first and then enhance it to satisfy the non-functional requirement?

SIMPLE/COMPLEX

Does the story have a simple core that provides most of the value and/or learning?

Could you split the story to do that simple core first and enhance it with later stories?

VARIATIONS IN DATA

Does the story do the same thing to different kinds of data?

Can you split the story to process one kind of data first and enhance with the other kinds later?

MAJOR EFFORT

When you apply the obvious split, is whichever story you do first the most difficult?

Could you group the later stories and defer the decision about which story comes first?

INTERFACE VARIATIONS

Does the story get the same kind of data via multiple interfaces?

Can you split the story to handle data from one interface first and enhance with the others later?

Does the story have a complex interface?

Is there a simple version you could do first?

last resort

BREAK OUT A SPIKE

Are you still baffled about how to split the story?

Can you find a small piece you understand well enough to start?

Write that story first, build it, and start again.

Can you define the 1-3 questions most holding you back?

Take a break and try again.

3 EVALUATE THE SPLIT

Are the new stories roughly equal in size?

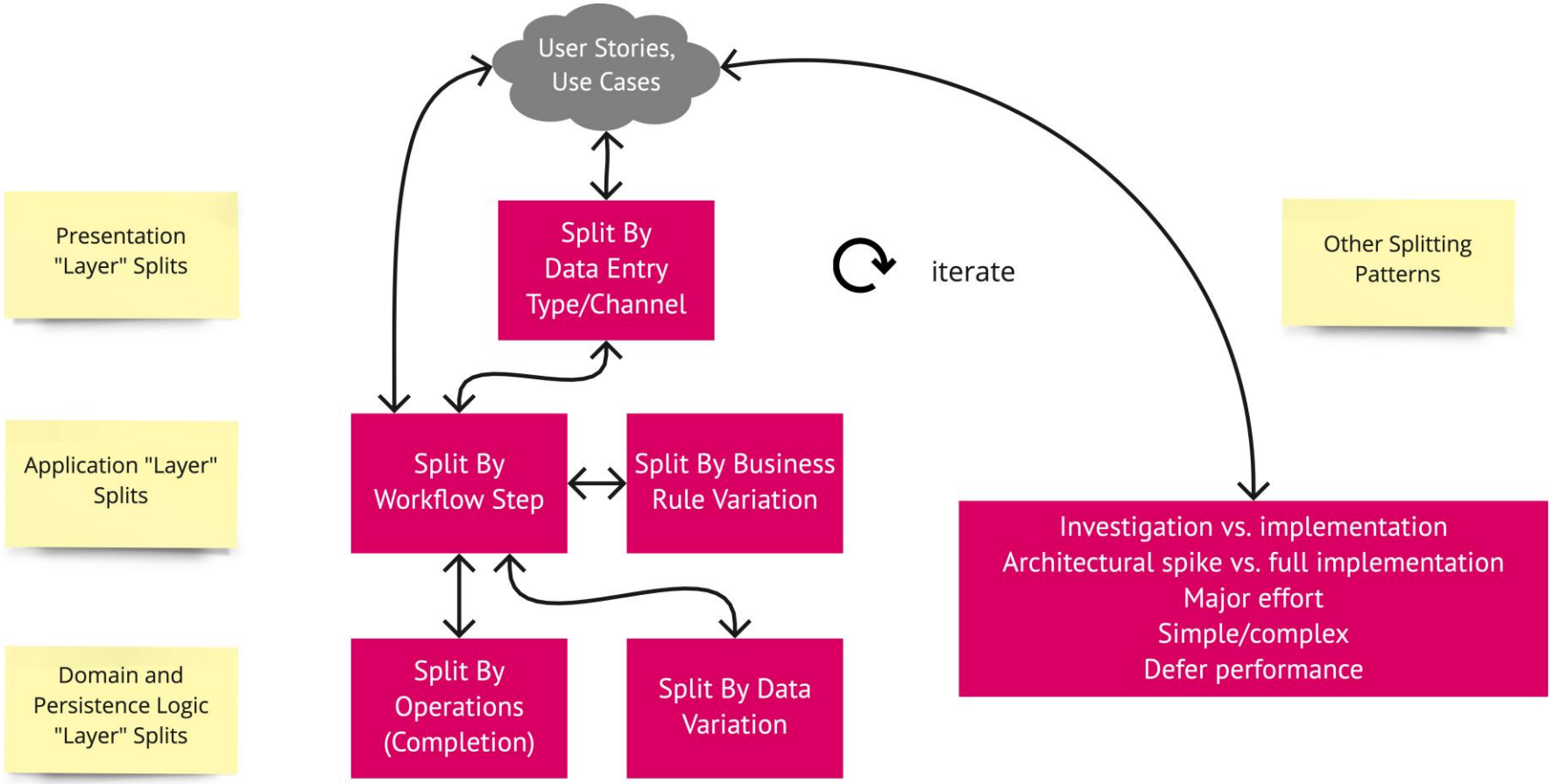
NO → Try another pattern on the original story or the larger post-split stories.

YES → Is each story about 1/10 to 1/4 of your velocity?

- Do each of the stories satisfy INVEST?
  - Try another pattern.
- Are there stories you can deprioritze or delete?
  - Try another pattern. You probably have waste in each of your stories.
- Is there an obvious story to start with that gets you early value, learning, risk mitigation, etc.?
  - Try another pattern to see if you can get this.
- You're done, though you could try another pattern to see if it works better.

Splitting Patterns in Action (1/2)

- https://socadk.github.io/design-practice-repository/activities/DPR-StorySplitting.html



Splitting Patterns in Action (2/2)

- https://socadk.github.io/design-practice-repository/activities/DPR-StorySplitting.html
- “As a Virtual Service Provider (VSP) and client of the PSTN (Public Switched Telephone Network) provider 'T’, I would like to create telephony orders on behalf of my end customers rapidly and reliably so that they are satisfied and stay with me rather than switch to T or another VSP.”

Splitting Criterion	Relevance in T	New Story Snippets (Subject to Component Identification)
Workflow Steps	High, order has 9 steps	Place order, Confirm order
Business Rule Variations	Number porting desired? Number porting possible?	Simulate change in customer database and billing system, reserve resources in telephony network
Major Effort	Relocation vs. new service	Create PSTN, move PSTN (to another address)
Simple/Complex	Visit of technician needed?	Schedule technician visit
Variations in Data	Mobile vs. fixed line telephony	Create new mobile service
Data Entry Methods	Browser UI vs. Rich Client App	Create new PSTN via Web API, Create new PSTN in browser
Defer Performance	Time to complete order	Respond to VSP (confirm order) within 4 hours (relaxed QA)
Operations (e.g., CRUD)	Create, Read, Update, Delete, Search (CRUDS) on orders	Request order status, Cancel order; Search order by id, by customer segment, by VSP, by VSP type
Break Out a Spike	20 backend systems	(Run a proof of Technology, mock each backend connection)

Application Example: SDD in Online Shop (1/3)

- *ChatGPT was happy to help*

✅ Epic (Business Level)

Epic:

As an online shopper, I want to have the option for same-day delivery so that I can receive my order faster and improve my shopping experience for urgent needs.

We'll use the typical e-commerce workflow:

1. Browse & Select Items
2. Add to Cart
3. Checkout & Choose Delivery
4. Payment
5. Order Processing
6. Pick & Pack
7. Dispatch
8. Delivery Confirmation

Application Example: SDD in Online Shop (2/3)

✅ User Story Map Table

Workflow Step	Basic (MVP)	Enhanced	Optimized
Browse & Select Items	*As a shopper, I want to see which items qualify for same-day delivery so that I know what's eligible.*	*As a shopper, I want filters for same-day delivery items so that I can quickly find them.*	*As a shopper, I want to see same-day availability per ZIP code so that I can plan better.*
Add to Cart	*As a shopper, I want to be notified if my cart is eligible for same-day delivery so that I can decide quickly.*	*As a shopper, I want to see estimated delivery time in my cart so that I know before checkout.*	*As a shopper, I want suggested add-ons that still keep same-day eligibility so that I can maximize my order.*
Checkout & Choose Delivery	*As a shopper, I want to choose same-day delivery at checkout so that I can receive my order today.*	*As a shopper, I want to see a delivery time window (e.g., 2–6 PM) so that I can plan my day.*	*As a shopper, I want dynamic pricing based on cutoff time so that I understand cost trade-offs.*
Payment	*As a shopper, I want the same-day delivery fee clearly shown so that I can make an informed decision.*	*As a shopper, I want multiple payment options for same-day delivery fees so that it's convenient.*	*As a shopper, I want loyalty points or discounts applied for same-day delivery so that I feel rewarded.*
Order Processing	*As a system, I want to flag same-day delivery orders so that the warehouse prioritizes them.*	*As a system, I want an alert system for approaching cutoff times so that orders are not delayed.*	*As a system, I want automated order batching for same-day routes so that operations are efficient.*
Pick & Pack	*As a picker, I want same-day orders highlighted so that I can prioritize them.*	*As a picker, I want optimized picking routes for same-day orders so that I work faster.*	*As a picker, I want real-time packing instructions so that I reduce errors.*
Dispatch	*As a dispatcher, I want to see all same-day orders so that I can plan deliveries efficiently.*	*As a dispatcher, I want to auto-assign drivers based on route optimization so that delivery is on time.*	*As a dispatcher, I want predictive traffic-based routing so that delays are minimized.*
Delivery Confirmation	*As a shopper, I want real-time tracking so that I know when my package will arrive.*	*As a shopper, I want SMS/Push updates on delivery status so that I stay informed.*	*As a shopper, I want to reschedule or reroute delivery in real-time so that I stay in control.*

Application Example: SDD in Online Shop (3/3)

	Browse & Select It	Add to Cart	Checkout & Choose	Payment	Order Processing	Pick & Pack	Dispatch	Delivery
Workflow (Basic)	Show eligibility	Notify eligibility	Choose same-day option	Show delivery fee	Flag same-day orders	Highlight same-day	List same-day orders	Real-time trac
Operations	Filter by same-day	Show est. delivery time	Edit before payment	Multiple payment options	Auto status updates	Optimize picking routes	Auto-assign drivers	One-click con
Business Rule	Eligibility by cutoff	Enforce rules	Enforce cutoff times	Dynamic pricing rules	Max orders per hour	Apply security rules	Traffic-based routing	PIN for high-v
Data Variation	Large/fragile handling	Partial eligibility	Some items same-day	Promo discounts	Handle high-value items	Handle fragile items	Different item sizes	Multi-package
Interface Variation	Mobile-friendly badge	Mobile cart UI	Mobile toggle	Mobile payment flow	Dashboard view	Mobile picker app	Driver mobile UI	SMS + Email

✅ Story Splitting Patterns Applied

We already used **Workflow Steps** for the horizontal axis. Now we'll add vertical splitting based on:

- **Operations** – Split by CRUD or operational steps.
- **Business Rule** – Split by complexity of rules or constraints.
- **Data Variations** – Split by differences in data (e.g., item size, ZIP codes).
- **Interface Variations** – Split by platforms or interface differences (e.g., desktop vs mobile).

Preview: Lab Exercise for This Lecture Lesson/Topic

Lab 4: Story Mapping and Splitting

Table of Content

OST
Ostschweizer
Fachhochschule

Collaborative Modeling Lesson 4: WORK IN PROGRESS (V2)

ITBO Learning Lab 2, Initiative 1 (ZIOL, KAPS)

Spring Term 2025

Departement Informatik and ITBO

Pros and Cons of Story Mapping

- Advantages (source: book, page nn):
  - TODO
- Disadvantages, liabilities:
  - TODO

Pros and Cons of Story Splitting

- See ChatGPT chat

When (not) to Use, Alternatives

- When to use:
  - TODO
- When not to use:
  - TODO
- Alternatives:
  - See chapter nn of the book

Critical Success Factors (Do's and Don'ts)

- TODO
  - Resulting from lab construction and prototypical use

Discussion: Opinions, Tips and Tricks

- TODO
  - Resulting from lab construction and prototypical use

Summary

- TODO

Resources (Links, Books, etc.)

- Also see https://www.mountaingoatsoftware.com/blog/five-simple-but-powerful-ways-to-split-user-stories

Questions (Repetition, Practice)

- TODO