codecentric

Basel One

Veni, Modeli, Vici

Turning Domain Stories into
Testable Prototypes in Minutes




helvetia
KARAKUN
Kanton Basel-Stadt
ERNI
INNOFUSE
+++Ba
sel++
++Te+
+++ch
Basel
basel.ch
JAVA
USER
GROUP
CH
BLKB
optravis
company of .msg
cyon
CS2
CSS
IRIX
swiss made
software
ICT
Scouts
Wir fördern Zukunft
SD:>_
[SwissDevJobs.ch]
#BaselOne26
baselone.ch

Johannes Barop

Senior Consultant & Software Architect
codecentric AG • Hamburg

- 15 years of software development
- Frontend, backend and infrastructure
- Experience from start-ups to large corporations and from social networks to insurance



Christian Langmann

Solution Architect
codecentric AG • Hamburg

- 20 years of experience designing, developing and optimizing complex software systems
- Interests range from design with sketchnotes to delivering software in clusters



Dr. Annegret Junker

Chief Software Architect
codecentric AG • Munich

- 30 years of experience designing and optimizing complex applications
- DDD expert and host of the DDD Meetup in Munich
- Author of several books on DDD

Annegret is not here today. Her material forms the basis of this workshop.



We work on Miro

https://miro.com/app/board/uXjVEaGKLzM=/

Password:
baselone

Work in small groups and share devices.

The facilitator can support prototype generation.



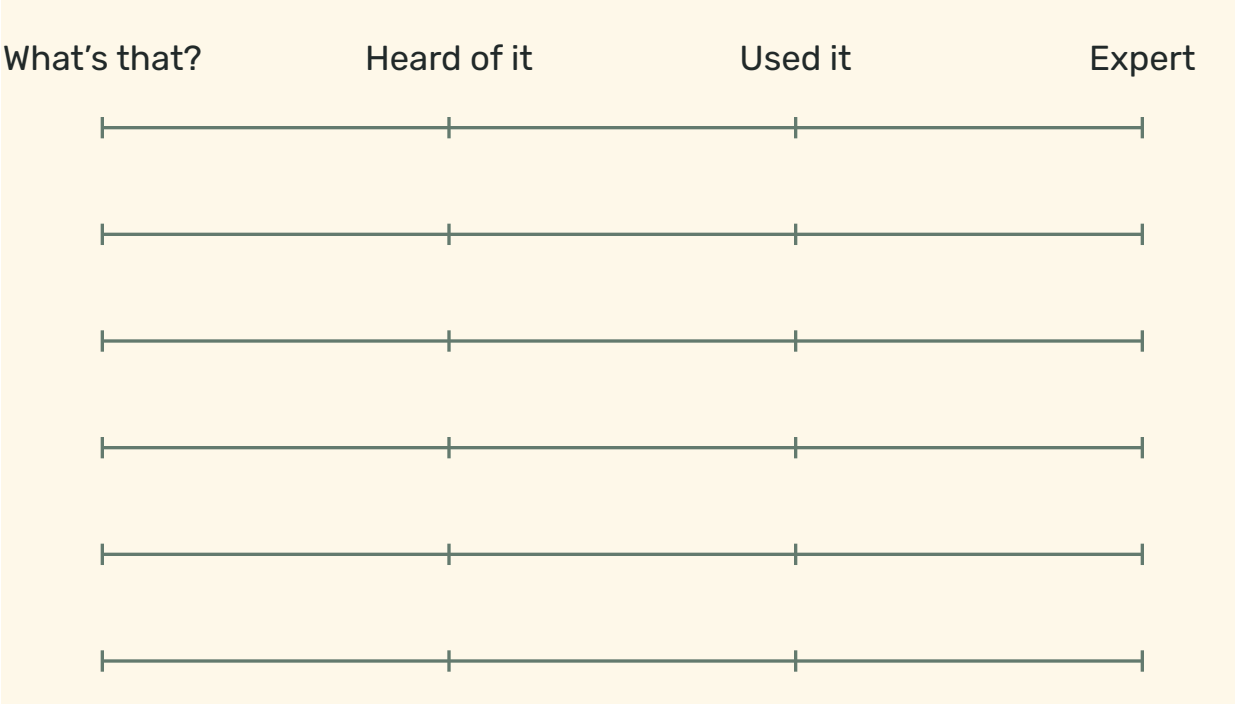
Check-in

Which book are you reading?

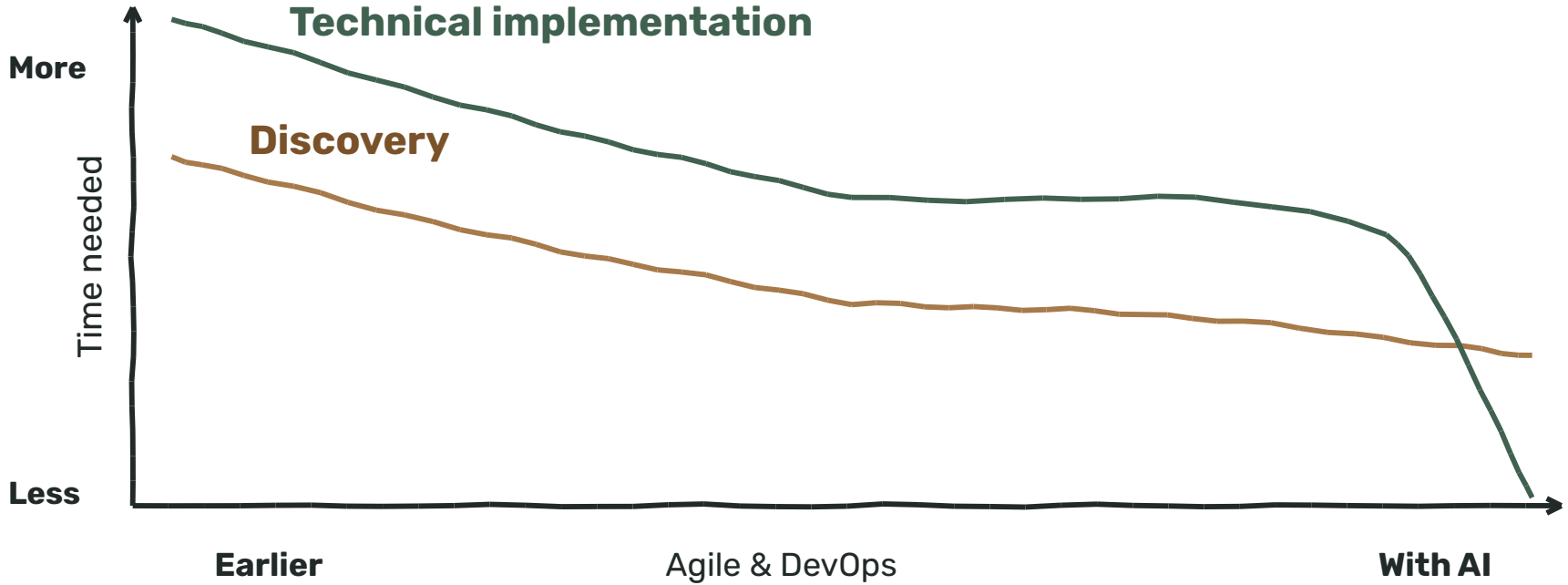
How familiar are you with these methods?



Where do you stand?



How the bottleneck shifts



What we'll do today

- Model an unfamiliar domain in small groups.
- Create Domain Stories and a Visual Glossary.
- Generate a testable prototype from your artifacts.
- Click through it and use what you learn to sharpen the model.



Basel One

A short introduction to DDD



Big Ball of Mud

- Foote & Yoder described the pattern and its trade-offs in a 1997 paper.
- Little upfront design can help teams deliver quickly.
- As the system grows, its structure can become hard to understand.



Foote & Yoder • Big Ball of Mud • PLoP '97

When changes are hard to predict

- Hidden dependencies spread the impact of a change.
- Understanding one part requires knowledge of much of the system.
- Explicit boundaries help us reason about change.



Basel One

Domain-Driven Design

- Start with the domain: the knowledge and activities the software supports.
- Domain experts and developers build a shared model.
- Let that model guide the software as understanding grows.

Eric Evans • Domain-Driven Design Reference



Focus the modelling effort

- The core domain is central to our business strategy.
- Supporting subdomains meet specific business needs.
- Generic subdomains address widely shared needs.



How this guides investment

Core

Invest deeply in understanding and distinctive solutions.

Supporting

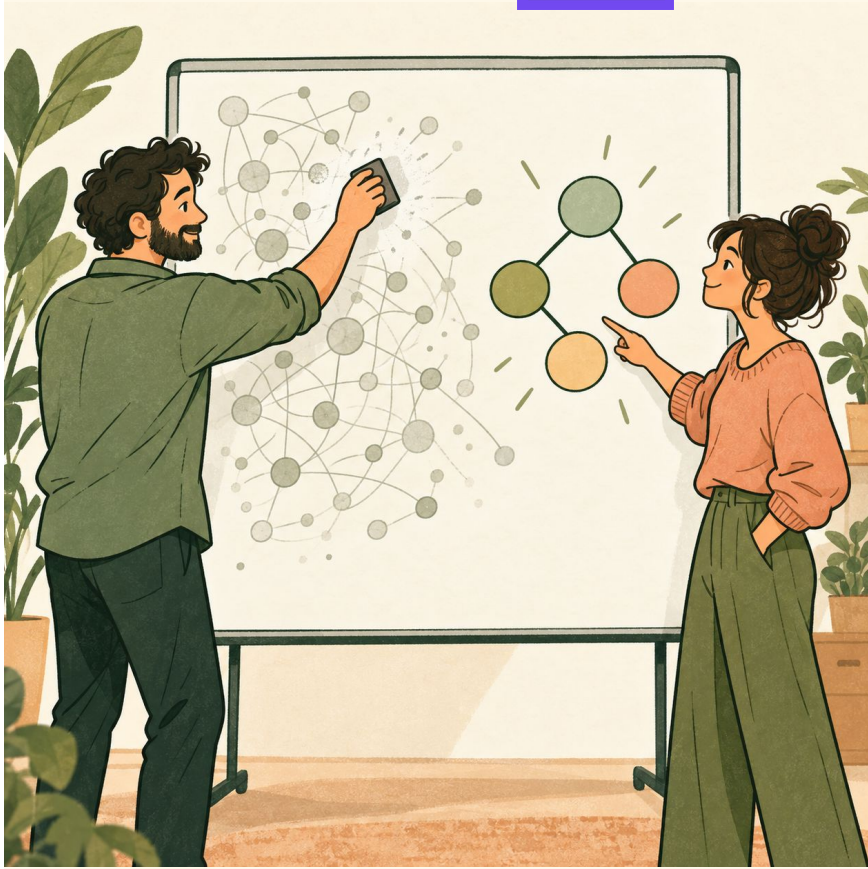
Build what is needed and keep the solution focused.

Generic

Evaluate established solutions before building your own.

Models have a purpose

- A model selects the concepts and rules that matter for a purpose.
- It leaves out detail that does not help that purpose.
- Different purposes may need different models.



Basel One

Bounded Contexts

- A bounded context defines where a model applies.
- Its terms and rules have a consistent meaning within that boundary.
- Relationships between contexts need explicit agreements.

Martin Fowler • Bounded Context

Basel One

Make the boundary explicit

Within a context

Keep the language, model and implementation aligned.

Between contexts

Agree how models connect and translate their meanings.

A model boundary does not prescribe a deployment unit.

Ubiquitous Language

- Domain experts and developers develop precise terms together.
- They use those terms in conversations, models, tests and code.
- They resolve ambiguity within each bounded context.



Strategic and tactical design

- Strategic design identifies priorities, model boundaries and relationships.
- Tactical design expresses concepts and rules within a model.
- Choose patterns that help express the domain clearly.



Selected tactical building blocks

Entity

Has a continuing identity,
even when its attributes change.

Value object

Is defined by its values
and has no identity of its own.

Aggregate

Defines a consistency boundary
that protects business rules.

Learning the domain together

- Domain Stories reveal activities and handovers.
- A Visual Glossary clarifies concepts and relationships.
- Prototypes let us test assumptions and refine the model.

Domain Storytelling • Collaborative modelling



Coffee break



Our example today: bike rental



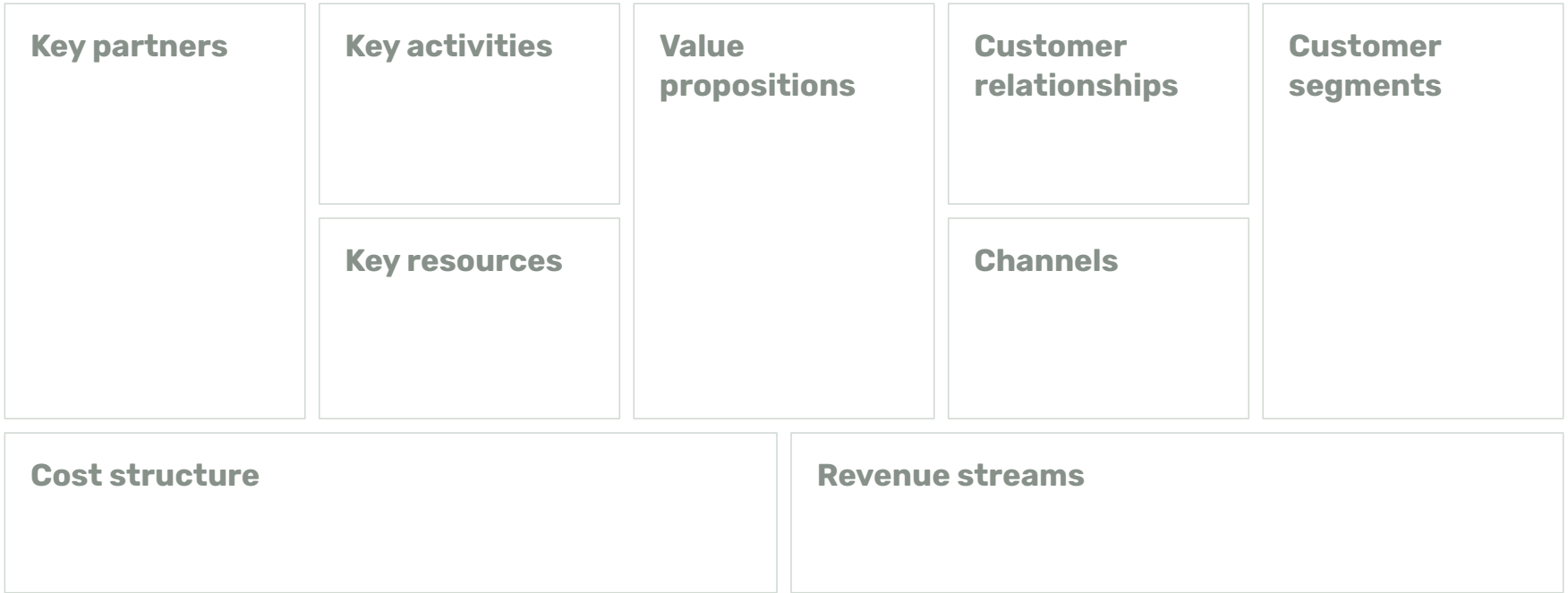
Meet BikeRent

- We will model and prototype a station-based bike-rental service.
- Riders unlock bikes at racks using the app.
- Commuters start near train stations.
- Tourists rent through partner hotels.



The Business Model Canvas

Nine building blocks describe our business model.



Adapted from Strategyzer AG • CC BY-SA 3.0

Who rents our bikes, and why?

1 of 3 • Customers

Key partners	Key activities	**Value propositions** Why choose BikeRent?	**Customer relationships** How do we interact?	**Customer segments** Who rents our bikes?
	Key resources		**Channels** How do we reach people?	
Cost structure			**Revenue streams**	

Adapted from Strategyzer AG • CC BY-SA 3.0

Who rents our bikes, and why?

1 of 3 • Customers

Key partners	Key activities	Value propositions	Customer relationships	Customer segments
		Easy to book Easy to pick up Easy to return	Self-service Support	Commuters Tourists
	Key resources		Channels App, Stations, Hotels,	

Cost structure	Revenue streams

Adapted from Strategyzer AG • CC BY-SA 3.0



What keeps BikeRent running?

2 of 3 • Operations

Key partners

Who helps us?

Key activities

What must we do?

Key resources

What do we need?

Value propositions

Customer relationships

Channels

Customer segments

Cost structure

Revenue streams

Adapted from Strategyzer AG • CC BY-SA 3.0

What keeps BikeRent running?

2 of 3 • Operations

Key partners

Public transport, Hotels

Key activities

Rent out bikes, Maintain bikes, Distribute bikes

Key resources

Bikes, Racks, Mechanics

Value propositions

Customer relationships

Channels

Customer segments

Cost structure

Revenue streams

Adapted from Strategyzer AG • CC BY-SA 3.0



How does BikeRent pay its way?

3 of 3 • Costs and revenue

Key partners	Key activities	Value propositions	Customer relationships	Customer segments
	Key resources		Channels	

Cost structure

What does the service cost?

Revenue streams

How do customers pay?

Adapted from Strategyzer AG • CC BY-SA 3.0

How does BikeRent pay its way?

3 of 3 • Costs and revenue

Key partners	Key activities	Value propositions	Customer relationships	Customer segments
	Key resources		Channels	

Cost structure

Bikes, racks, maintenance,
App and marketing

Revenue streams

Commuters: flat rate,
Tourists: pay per rental

Adapted from Strategyzer AG • CC BY-SA 3.0

From business model to stories

How does a commuter rent a bike at a station?

Key partners

Public transport, Hotels

Key activities

Rent out bikes
Maintain bikes
Distribute bikes

Key resources

Bikes, Racks, Mechanics

Value propositions

Easy to book
Easy to pick up
Easy to return

Customer relationships

Self-service
Support

Channels

App,
Stations,
Hotels,

Customer segments

Commuters
Tourists

Cost structure

Bikes, racks, maintenance,
App and marketing

Revenue streams

Commuters: flat rate,
Tourists: pay per rental

Adapted from Strategyzer AG • CC BY-SA 3.0

Introduction to Domain Storytelling



What is Domain Storytelling?

- Domain experts describe a concrete case.
- A moderator draws the story and reads it back.
- Everyone checks the picture and clarifies misunderstandings.
- The group learns who does what with whom.



A commuter rents a bicycle

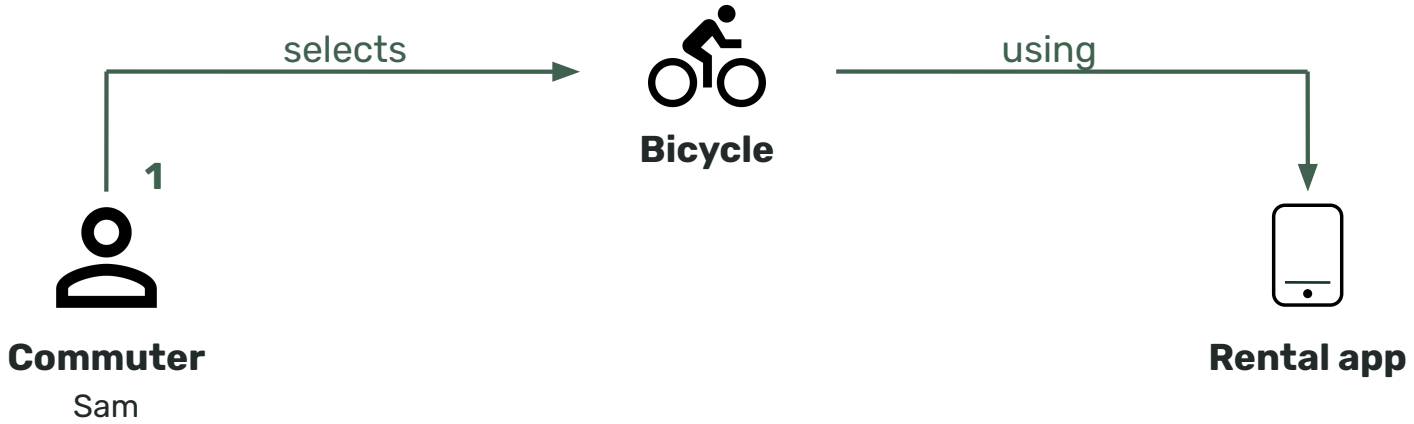


Commuter

Sam

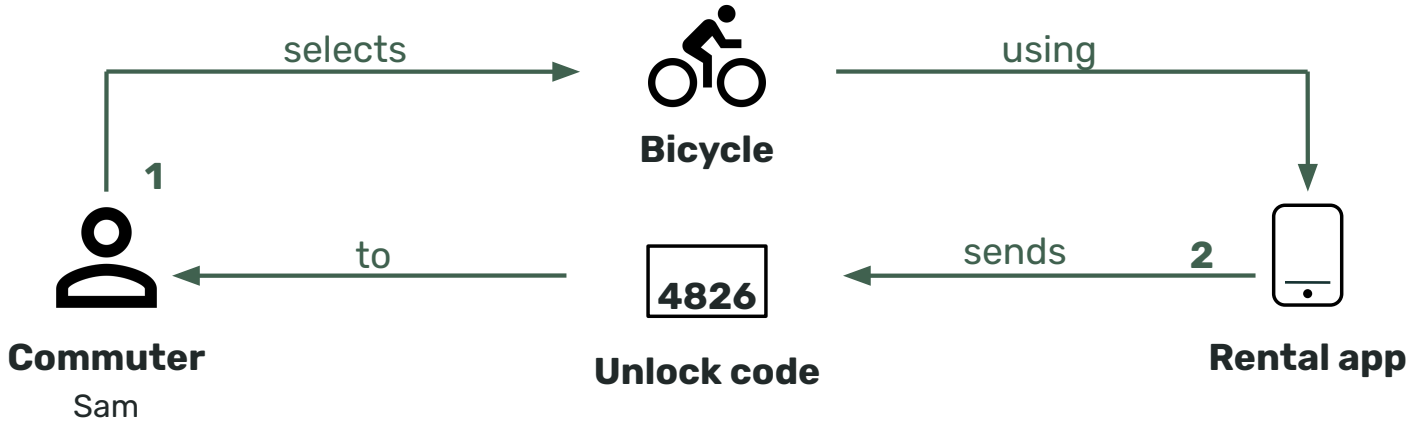
A commuter rents a bicycle

Sam selects a bicycle using the rental app.



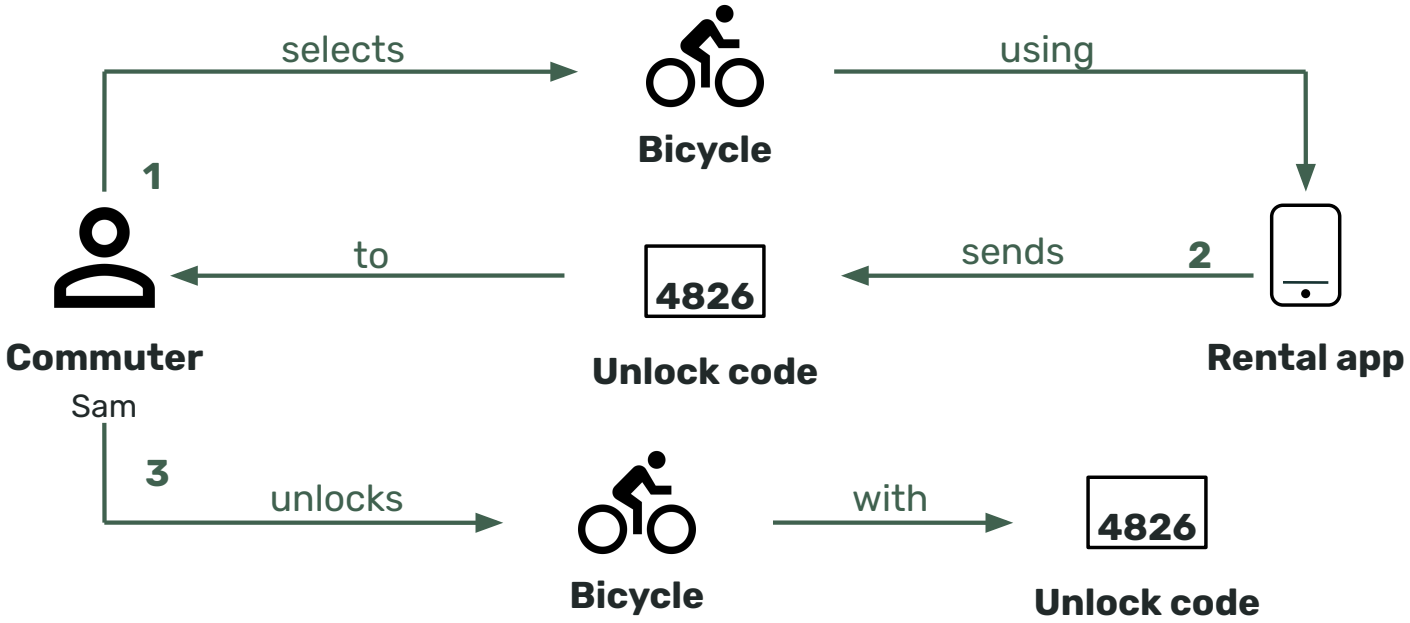
A commuter rents a bicycle

The rental app sends an unlock code to Sam.



A commuter rents a bicycle

Sam unlocks the bicycle with the code.



What makes a useful story?

- A story follows one concrete case.
- Precise actor names make responsibilities clear.
- The words in the picture match the domain language.
- Assumptions and open questions stay visible.
- A different outcome can become a separate story.

Domain Storytelling Commuter

- Model the commuter’s journey from registration to returning the bike.
- The commuter has just installed the app and has not registered yet.
- Choose a storyteller and a moderator. Everyone asks questions.
- Draw the story together. Record assumptions and open questions.



Lunch break



Presenting results



Domain Storytelling Tourist

- Model a bike rental at the hotel, through to the return.
- The tourist has just arrived at the hotel and seen an advert for bike rental.
- Choose a new storyteller and moderator. Everyone asks questions.
- Draw the story together. Record assumptions and open questions.



Basel One

Presenting results



Introduction to Visual Glossary





From stories to shared vocabulary

- A Visual Glossary captures domain terms and their relationships.
- The group agrees on what the words mean.
- Terms from both bicycle stories contribute to this understanding.
- These relationships help us ask more precise questions.



A visual glossary for BikeRent

We start with the term Rental.

Rental

Here, Rental means one use of a bicycle from pickup to return.

A visual glossary for BikeRent

A relationship connects two terms with a verb.



Read the relationship as a sentence: “A rental uses a bicycle.”

A visual glossary for BikeRent

The quantity makes our example rule explicit.



1 means exactly one bicycle per rental in this example.

A visual glossary for BikeRent

Each rental belongs to one commuter in this example.



Read each quantity in the direction of its arrow.

A visual glossary for BikeRent

A rack can hold zero or more bicycles at a given time.



0..* means zero or more. The rack's capacity is still an open question.

A visual glossary for BikeRent

We check what these example rules allow.



Can one rental include two bicycles?
Can a rack be empty?



Bring new understanding back

- Clearer definitions can change the words in a Domain Story.
- A relationship can reveal a missing step or assumption.
- The group revisits the story and checks the new understanding together.

Coffee break



Visual Glossary

- Build one glossary from your commuter and tourist stories.
- Agree on meanings. Write each term in the singular.
- Connect terms with verbs and add quantities.
- Test the rules. Record assumptions and open questions.
- Revisit both stories and update them together.



Basel One

Presenting results



Skills and prototyping

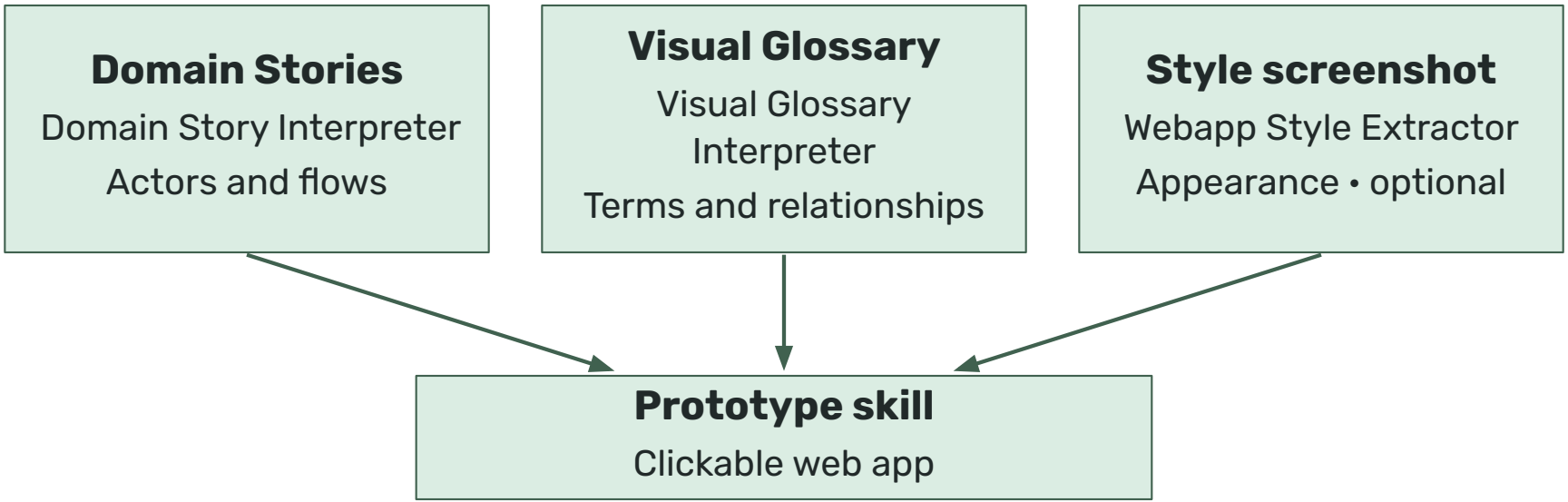


Make the model testable

- Turn your Domain Stories and Visual Glossary into a clickable web app.
- Try the journeys together and check whether they match your understanding.
- Spot missing steps, unclear terms and assumptions made by the AI.
- Use what you learn to refine the model and the prototype.

Four skills, one prototype

A skill gives the AI reusable instructions for a specific task.



Generate, try, refine

- Export readable images of both stories and the glossary.
- Provide the prompt and skills. Add a style screenshot if you want one.
- Check that the agent can read the inputs and has loaded the required skills.
- Generate the prototype, walk through the stories and feed back corrections.

Build your prototype

- Export both stories and your glossary as readable images.
- Use the Miro prompt and add your images to the agent.
- Generate a prototype and try both rental journeys.
- Record one mismatch or open question to share.

https://miro.com/app/board/uXjVEaGKLzM=/



Presenting results



Discussion



DDD in the age of AI

Is DDD more or less important today than it was in 2003?



Less important Equally important **More important**

What in today's workshop influenced your view?

Check-out and goodbye



Check-out

What will you try on Monday?



How was the workshop?

Thanks for joining us!

Johannes & Christian

