# Features, scenarios, and stories

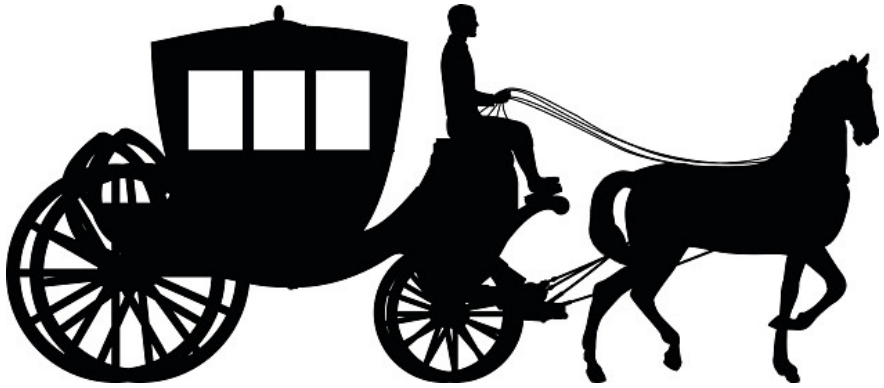
Antonio Brogi

Department of Computer Science
University of Pisa

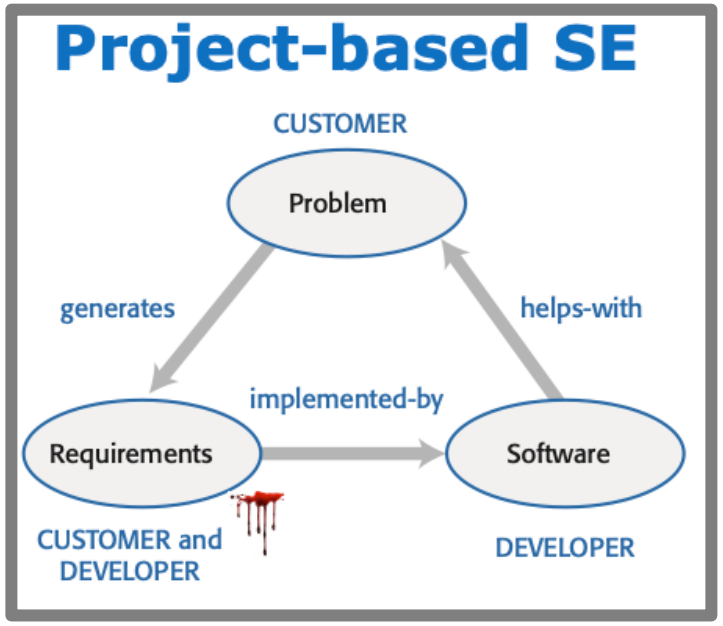
# Introduction

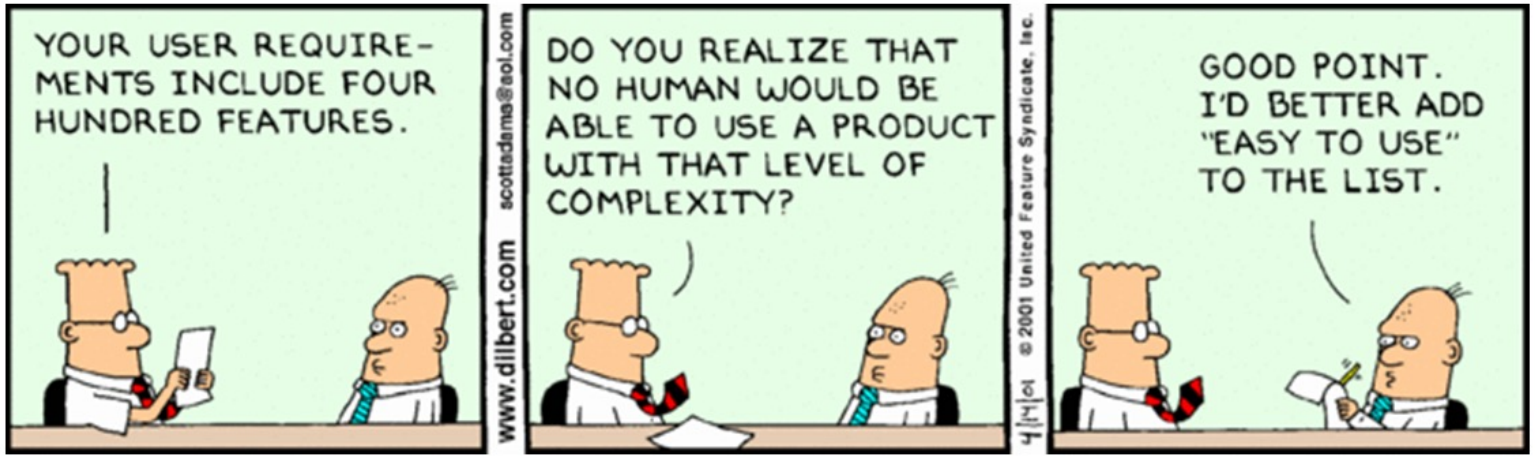
Factors driving the design of software products

- Inspiration
- Business/consumer needs not met by existing products
- Dissatisfaction with existing products
- Technical changes making new product types possible



# Requirements documentation





***¿Donde estamos?***
*(Where are we?)*

*Ma´anaatik* ***kat´ann***
*(We don't understand your language)*

# Introduction

Product-based software engineering needs less requirements documentation that project-based software engineering

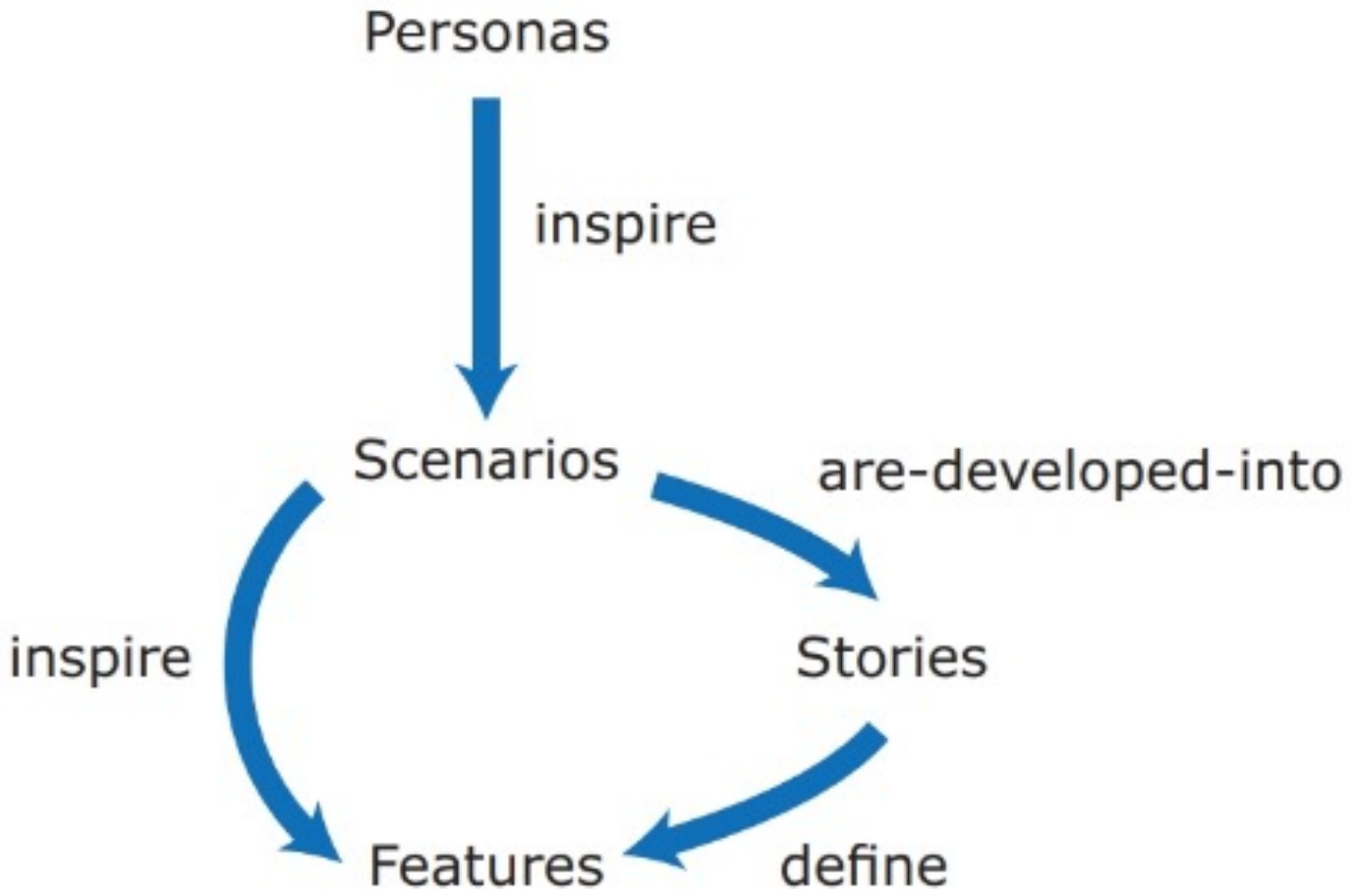
- Requirements not set by customers
- Requirements change

Identify product **features** (= fragments of functionality) instead

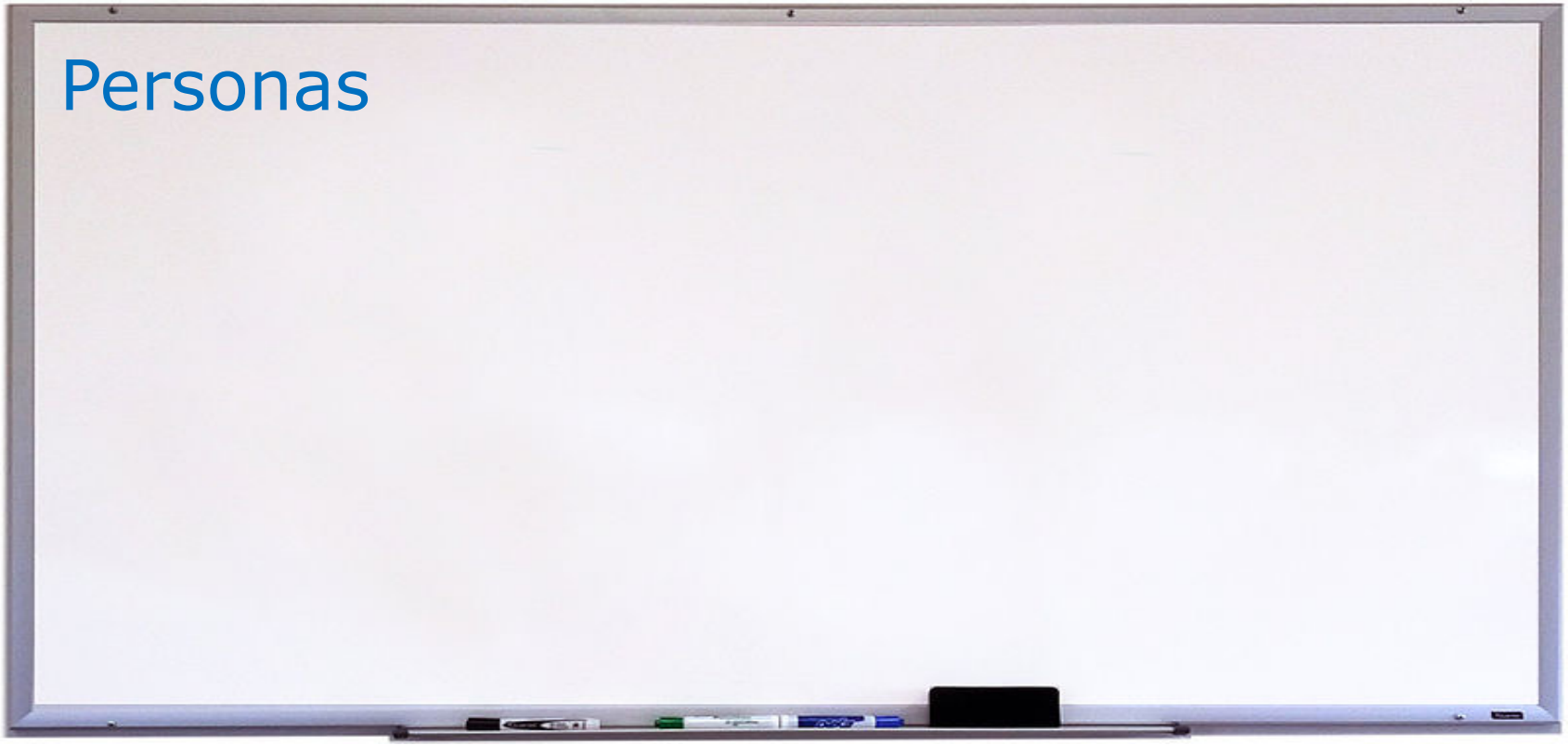
- Need to understand potential users
  - Interviews/surveys
  - Informal user analysis/consultations
- Warning: e.g. business managers buy product, employee (may not want to) use it

# Introduction

User representations (**personas**) and natural language descriptions (**scenarios** and **stories**) help identifying product **features**



# Personas



# Personas

Who are the target users for our product?

- Need to understand potential users to design features useful for them
- Background, skills and experience of potential users are important

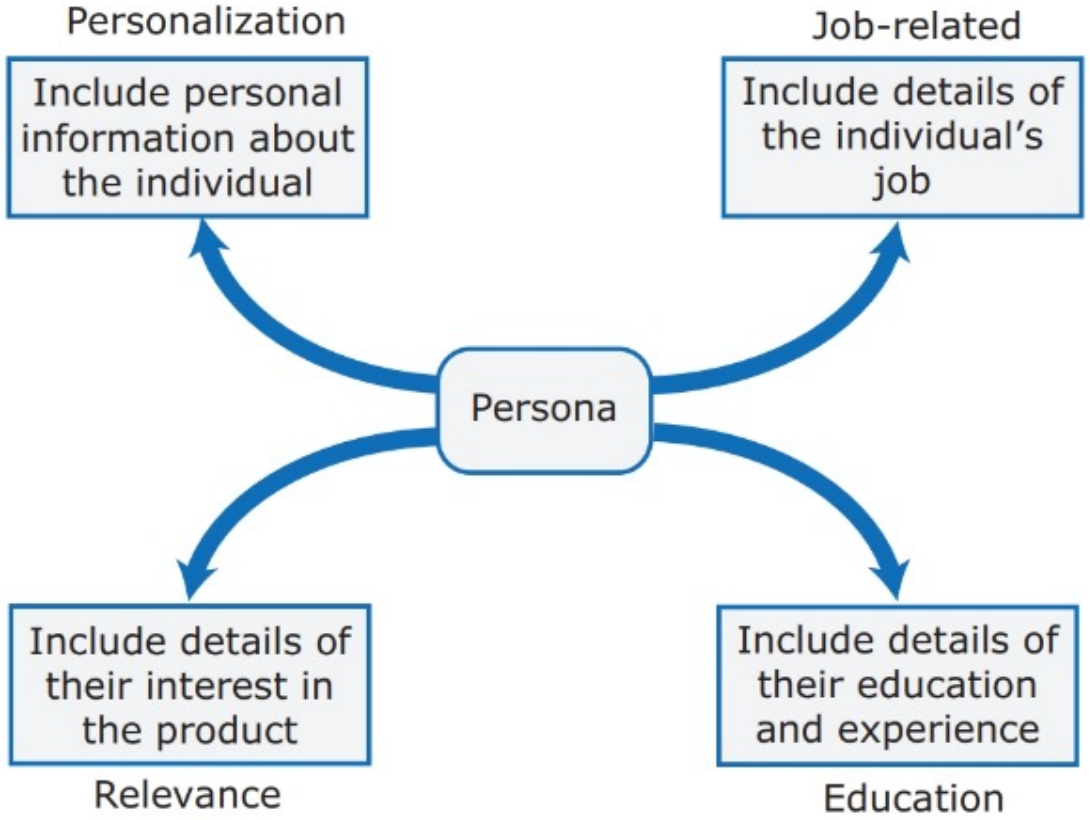
Personas are (imagined) types of product users

- e.g. for a dentist agenda: *dentist*, *receptionist*, *patient*
- Generally we need a few (1-2, max 5) personas to identify key product features

# Aspects of persona descriptions

You should give them a name and say something about their personal circumstances. It is sometimes helpful to use an appropriate stock photograph to represent the person in the persona. Some studies suggest that this helps project teams use personas more effectively.

If your product is targeted at business, you should say something about their job and (if necessary) what that job involves. For some jobs, such as a teacher where readers are likely to be familiar with the job, this may not be necessary.




If you can, you should say why they might be interested in using the product and what they might want to do with it.

You should describe their educational background and their level of technical skills and experience. This is important, especially for interface design.

# Using photos in persona descriptions?

*It is sometimes helpful to use an appropriate stock **photograph** to represent the person in the persona. Some studies suggest that this helps project teams use personas more effectively.*

Using photos is misleading. Personas shouldn't be about how people look, but what they do. (Steve Cable)

Detailed personas encouraged the team to assume that demographic information drove motivations. (Sara Wachter-Boettcher)

Persona profiles with a smiling photo result in an increase in willingness to use the personas. (Joni Salminen et al.)

Gender has a mediating influence on perceived attributes of males and females. (Francine M. Deutsch)

# Example of persona

*Personalization*
*Job-related*
*Education*
*Relevance*

**Emma, a history teacher**

Emma, age 41, is a history teacher in a secondary school (high school) in Edinburgh. She teaches students from ages 12 to 18. She was born in Cardiff in Wales, where both her father and her mother were teachers. After completing a degree in history from Newcastle University, she moved to Edinburgh to be with her partner and trained as a teacher. She has two children, aged 6 and 8, who both attend the local primary school. She likes to get home as early as she can to see her children, so often does lesson preparation, administration, and marking from home.

Emma uses social media and the usual productivity applications to prepare her lessons, but is not particularly interested in digital technologies. She hates the virtual learning environment that is currently used in her school and avoids using it if she can. She believes that face-to-face teaching is most effective. She might use the iLearn system for administration and access to historical films and documents. However, she is not interested in a blended digital/face-to-face approach to teaching.

→ Doesn’t have technical background

→ Wants product supporting administration

Personas allow developers to “step into the users’shoes”

When studying the users is not possible (e.g. for some new products) we can develop **proto-personas**

Jessica Williams, 35

**INCOME**
$92,000 per year

**FAMILY**
Married, 2 kids

**EDUCATION**
B.A.

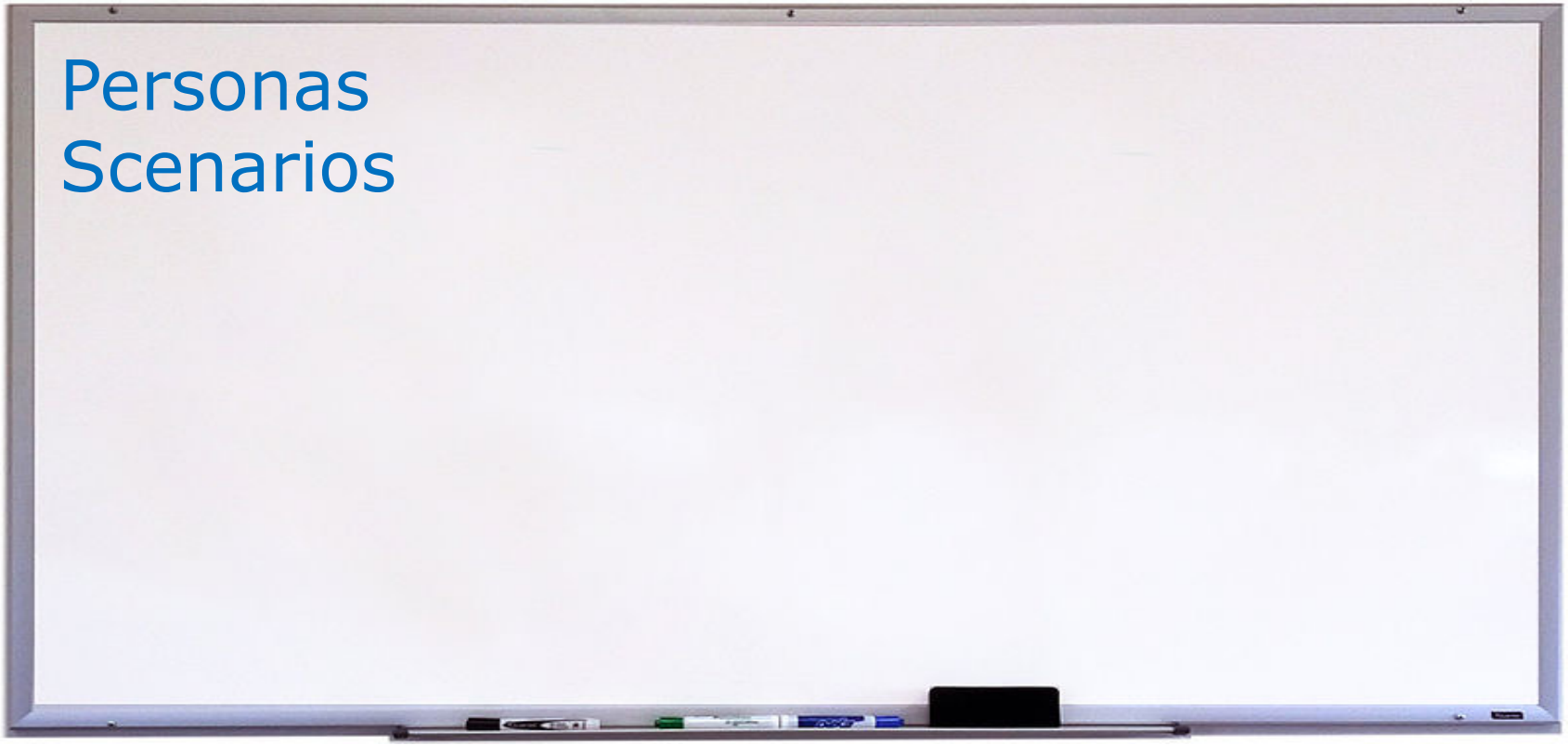
**CANCER RISK**
High. Mother has breast cancer.

**HOPES**

I want a clear path for me and my mom to follow, just tell me what to do.

Jessica has been identified at high-risk for breast cancer due to her family history. Her mother was recently diagnosed with breast cancer at the age of 65. Though Jessica has had several benign lumps, she is still concerned about her own risk and what the future holds for her mom’s health and treatment.

Jessica has done some research on her own around her genetic risk, and treatment for breast cancer. She has collected resources in a folder on her desktop, but is overwelmed with what the best next steps are, or small steps she can take day-to-day. Jessica also wants to be a means of support for her mom, who is not tech savvy, and provide her with accurate information.

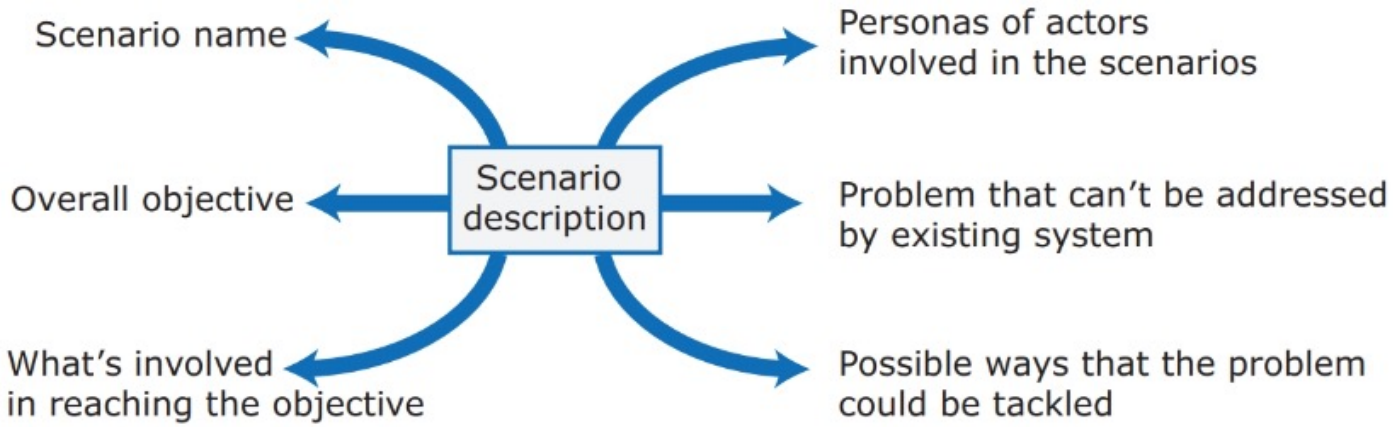

Personas
Scenarios

# Scenarios

To discover product features, we can define scenarios of user interactions with the product

Scenario = narrative describing a situation in which a user is using our product’s features to do something she wants to do

# Scenarios




**Fishing in Ullapool**

Jack is a primary school teacher in Ullapool, teaching P6 pupils. He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development, and economic impact of fishing.

As part of this, students are asked to gather and share reminiscences from relatives, use newspaper archives, and collect old photographs related to fishing and fishing communities in the area. Pupils use an iLearn wiki to gather together fishing stories and SCRAN (a history archive site) to access newspaper archives and photographs. However Jack also needs a photo-sharing site as he wants students to take and comment on each others' photos and to upload scans of old photographs that they may have in their families. He needs to be able to moderate posts with photos before they are shared, because pre-teen children can't understand copyright and privacy issues.

Jack sends an email to a primary school teachers' group to see if anyone can recommen an appropriate system. Two teachers reply and both suggest that he use KidsTakePics, a photo-sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account with KidsTakePics.

He uses the the iLearn setup service to add KidsTakePics to the services seen by the students in his class so that, when they log in, they can immediately use the system to upload photos from their phones and class computers.

# Scenarios

Narrative, high-level scenarios facilitate communication and stimulate design creativity

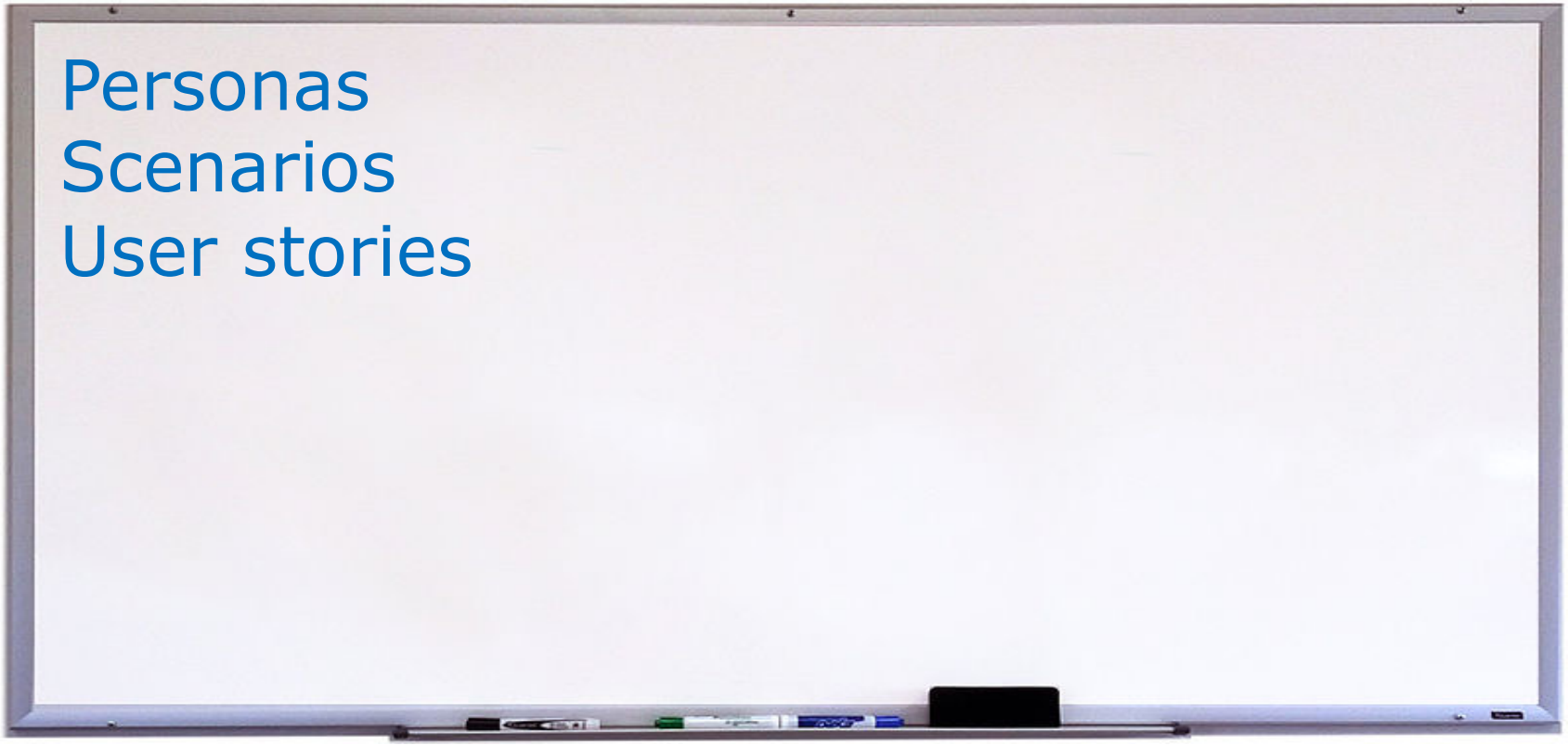
Scenarios are not specifications, though (they lack detail and may be incomplete)

# Writing scenarios

Several scenarios (e.g. 3-4) for each persona, covering main responsabilities of persona

Written from user’s perspective

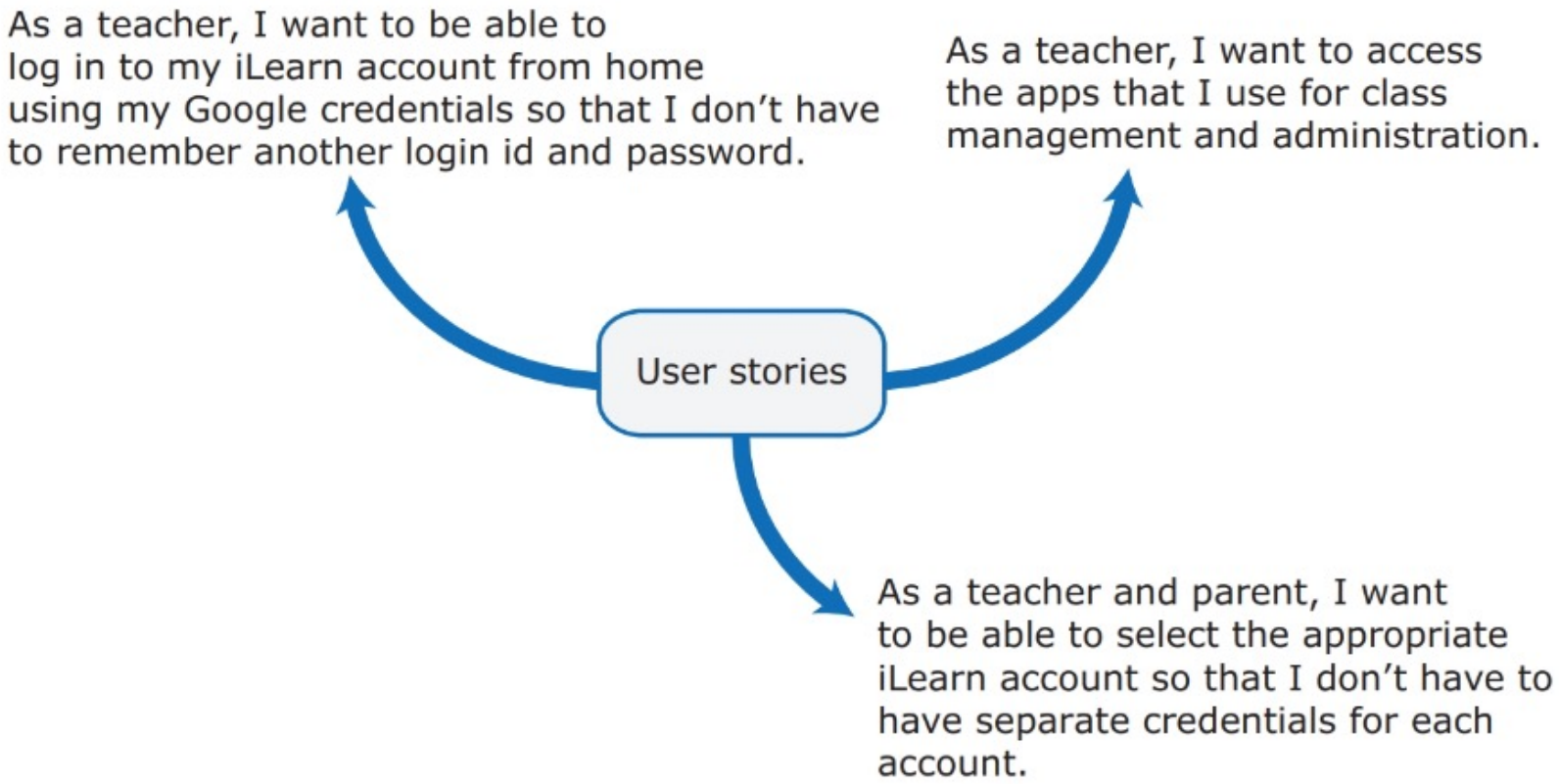
Each team member should (individually) create some scenarios, and then discuss them with rest of team and with users (if possible)


Personas
Scenarios
User stories

# User stories

- Scenarios = high-level stories of product use
- User stories = finer-grained narratives

| | |
|---|---|
| **As a** | <role> |
| **I want to** | <do something> |
| **So that** | <reason/values> |



# Agile user stories


CLICK TO PLAY


1. *Our highest priority is to satisfy the customer through **early and continuous delivery of valuable software**.*

7. ***Working software** is the primary measure of progress.*

10. ***Simplicity**--the art of maximizing the amount of work not done--is essential.*

- organize and chunk work into units that represent value to the customer
  - distinguish more valuable from less important
  - visualize units of work, through **simple stories**
    - template: **As a** (who) + **I want to** (what) + **So I can** (why)
  - stories written from the user's viewpoint
  - good quality stories make rest of development process more efficient
  - good user stories create business value for the users
- build software from the users perspective

Homework

# Splitting user stories


CLICK TO PLAY


3. ***Deliver working software frequently****, from a couple of weeks to a couple of months, with a preference to the shorter timescale.*
10. ***Simplicity****--the art of maximizing the amount of work not done--is essential.*

- Think about application the way the user thinks about it
- Developers tend to see the application as a stack
- User thinks of an application in terms of valuable «slices of functionality»
- Build software incrementally, as slices of functionality
- A good story should take 1-2 days of work
- Split long stories into several shorter stories, which can be prioritized
- Example of how to split stories

# User stories for planning

- Scrum product backlog is often a set of user stories
- Long stories (epics) to be broken into simpler stories
- Stories are associated with **priorities** (and possibly also with an estimate of effort needed to implement the story) and **sorted** according to priority («requirements triage»)

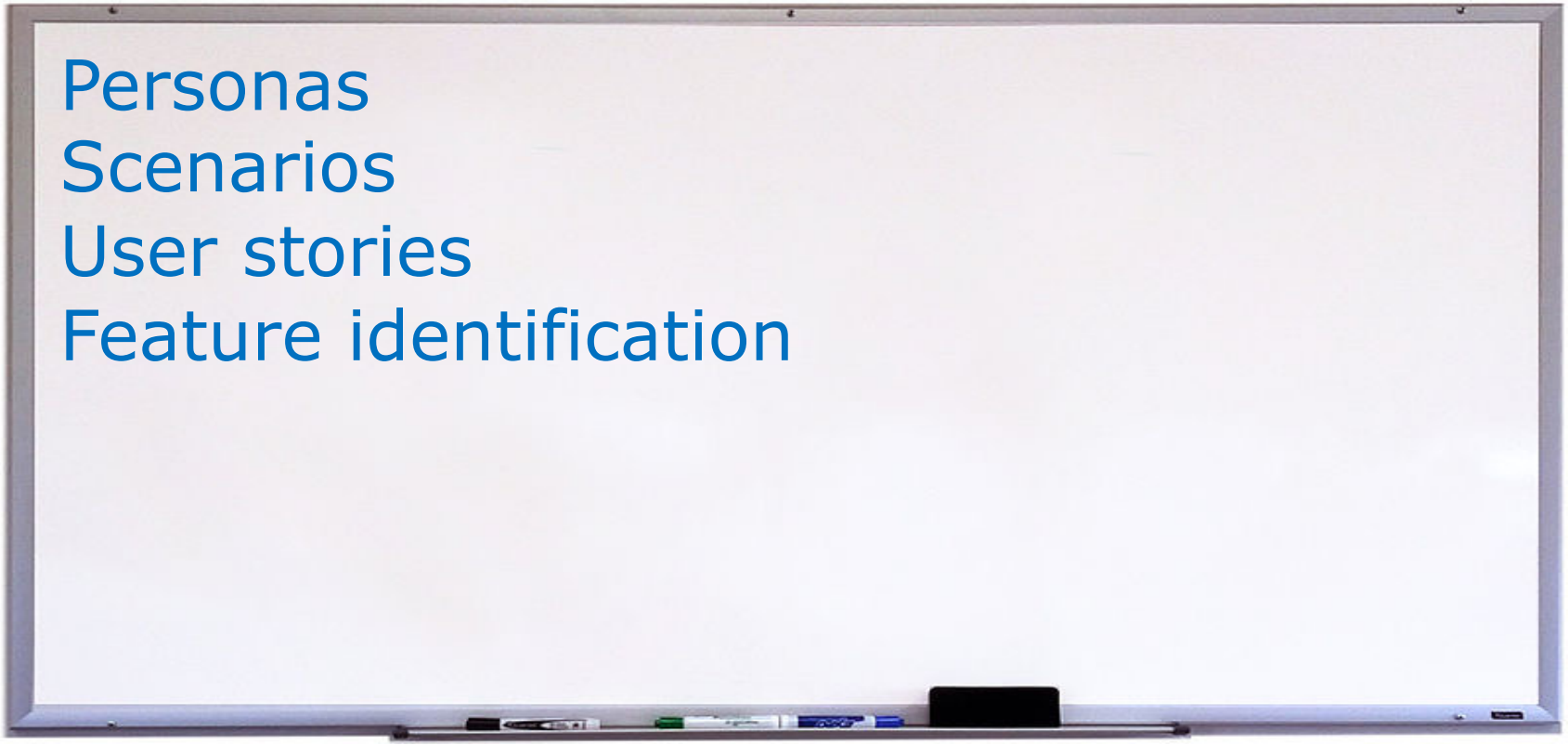
# User stories

It is possible to express all the functionalities described in a scenario as user stories

… so why bothering with scenarios?

Scenarios

+ read more naturally
+ make it easier to understand stories
+ provide more context


Personas
Scenarios
User stories
Feature identification

# Feature identification

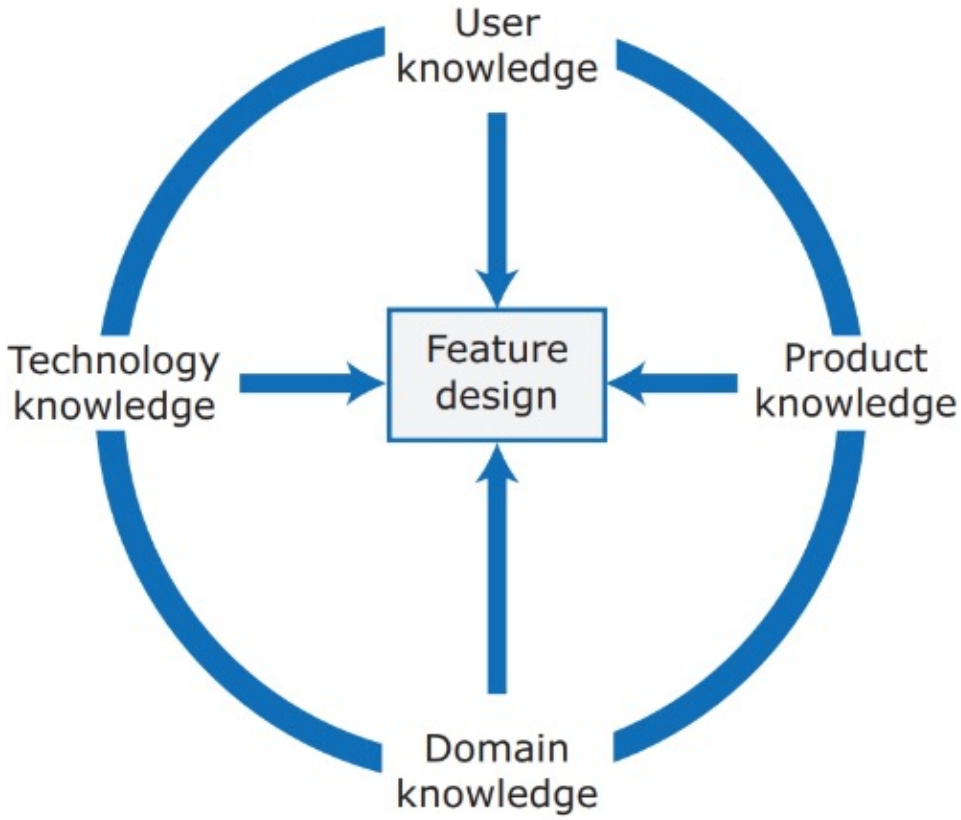
- Aim: Getting a list of features that define our product

- Properties
  - ***Independence*** - A feature should not depend on how other system features are implemented and should not be affected by the order of activation of other features.
  - ***Coherence*** - Features should be linked to a single item of functionality. They should not do more than one thing, and they should never have side effects.
  - ***Relevance*** - System features should reflect the way users normally carry out some task. They should not offer obscure functionality that is rarely required.

# Knowledge sources for feature design

You can use user scenarios and user stories to inform the team of what users want and how they might use the software features.

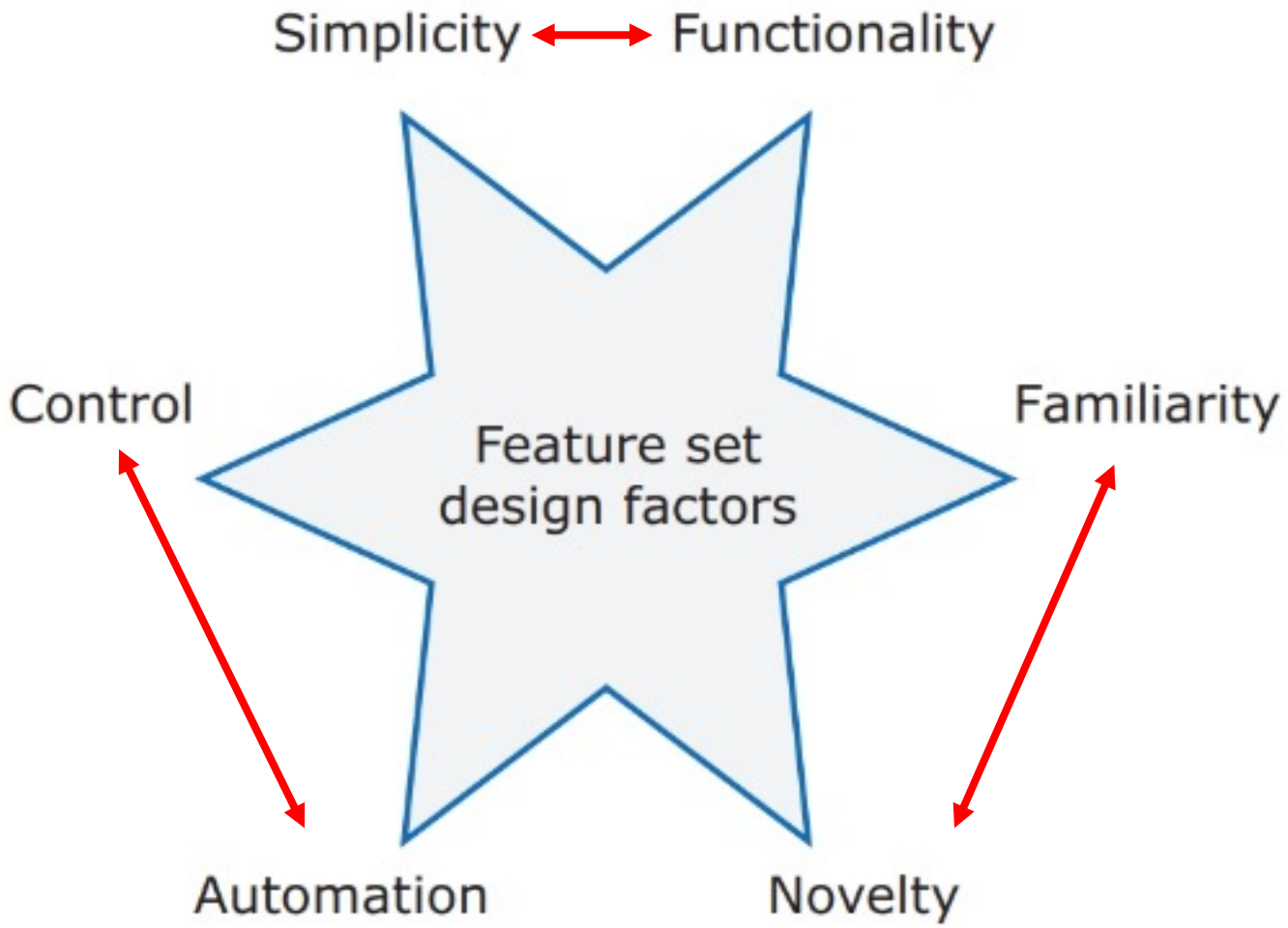
New products often emerge to take advantage of technological developments since their competitors were launched. If you understand the latest technology, you can design features to make use of it.




You may have experience of existing products or decide to research what these products do as part of your development process. Sometimes your features have to replicate existing features in these products because they provide fundamental functionality that is always required.

This is knowledge of the domain or work area (e.g., finance, event booking) that your product aims to support. By understanding the domain, you can think of new innovative ways of helping users do what they want to do.

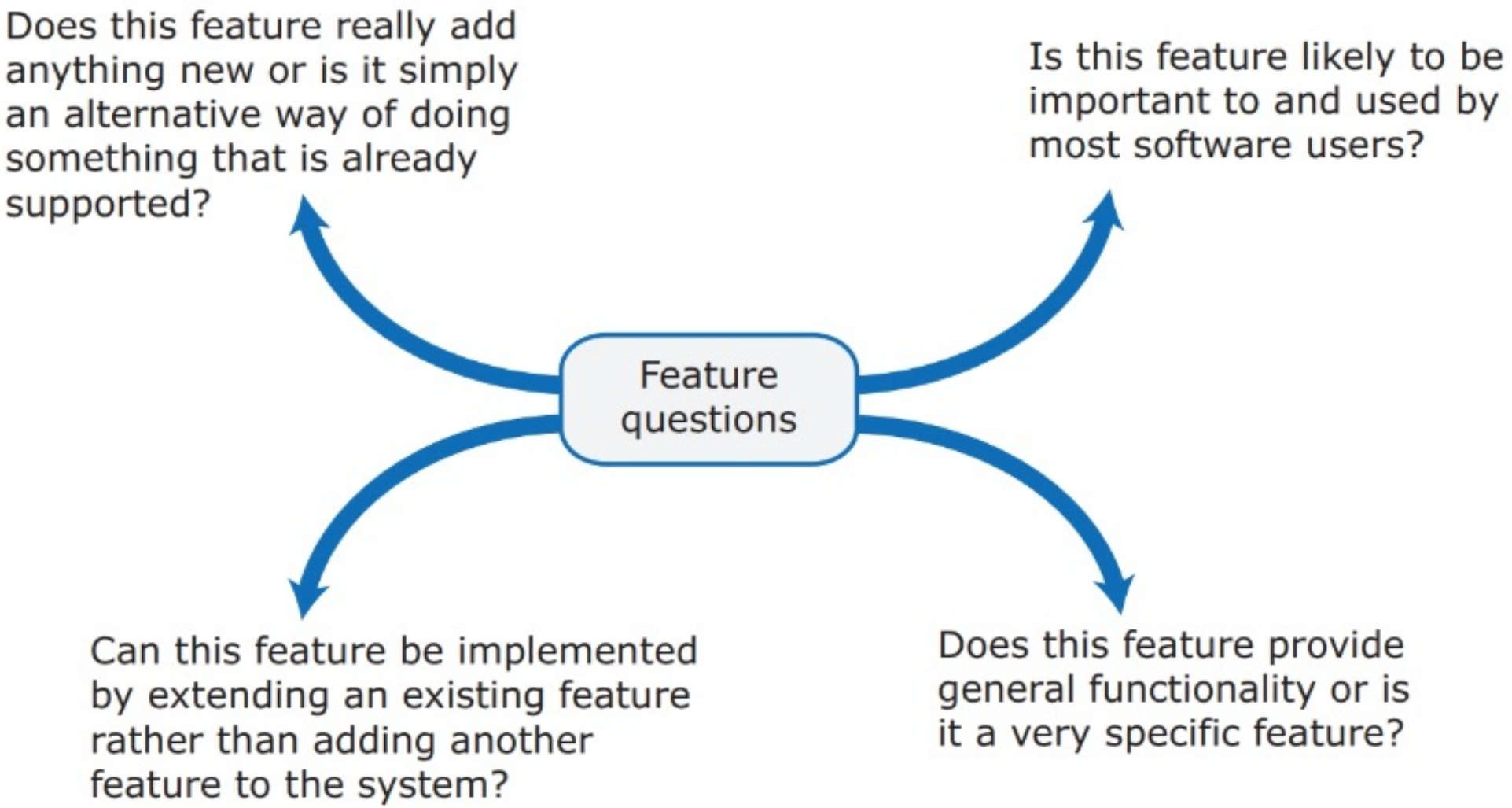
# Factors in feature set design



# Feature creep

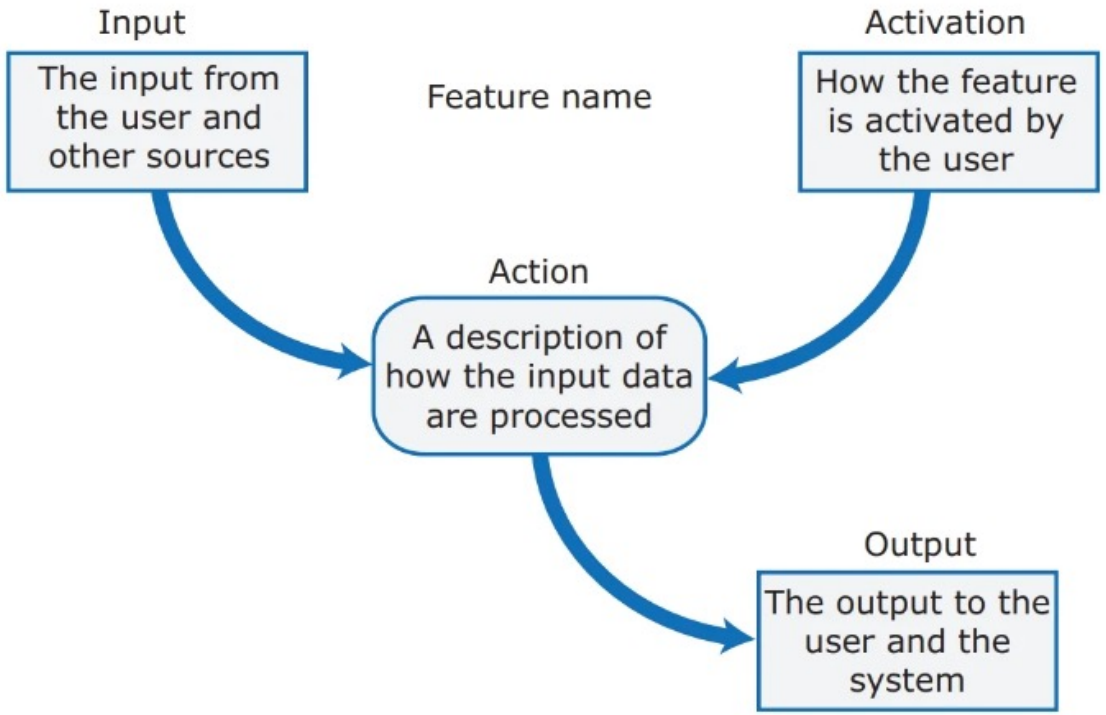
- Number of product features grows as new potential users are envisaged
  - Marketing executives want to meet all users’ demands
  - Marketing pressur to include competitors’features
  - Desire to support both experienced and inexperienced users

# Avoiding feature creep



# Feature derivation

Product development team must meet to discuss scenarios and stories to extract the list of product feature descriptions




Start prototyping to demonstrate (first novel and critical) features

# Reference


Chapter 3 – Features, Scenarios, and Stories