

Lesson 3

Requirements 1

Requirements Introduction

Requirements



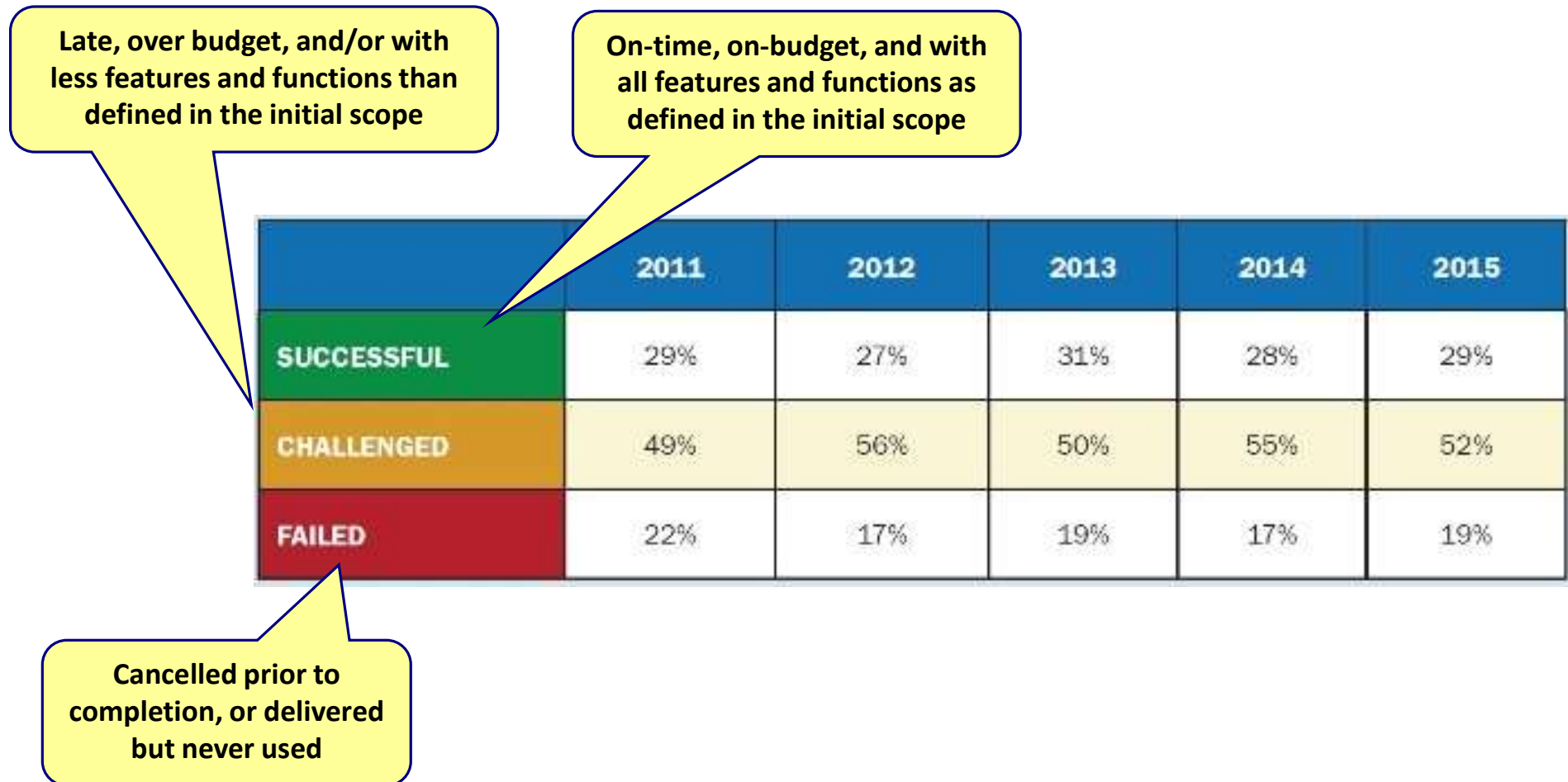
The hardest single part of building a software system is deciding precisely what to build.

Fred Brooks - "No Silver Bullet" 1987

If you don't get the requirements right, it does not matter how well you do anything else.

Karl Wieggers

Success in software development



Why do projects fail?

Top reasons why projects fail

- Incomplete requirements
- Lack of user involvement
- Lack of resources
- Unrealistic expectations
- Lack of executive support
- Changing requirements and specifications

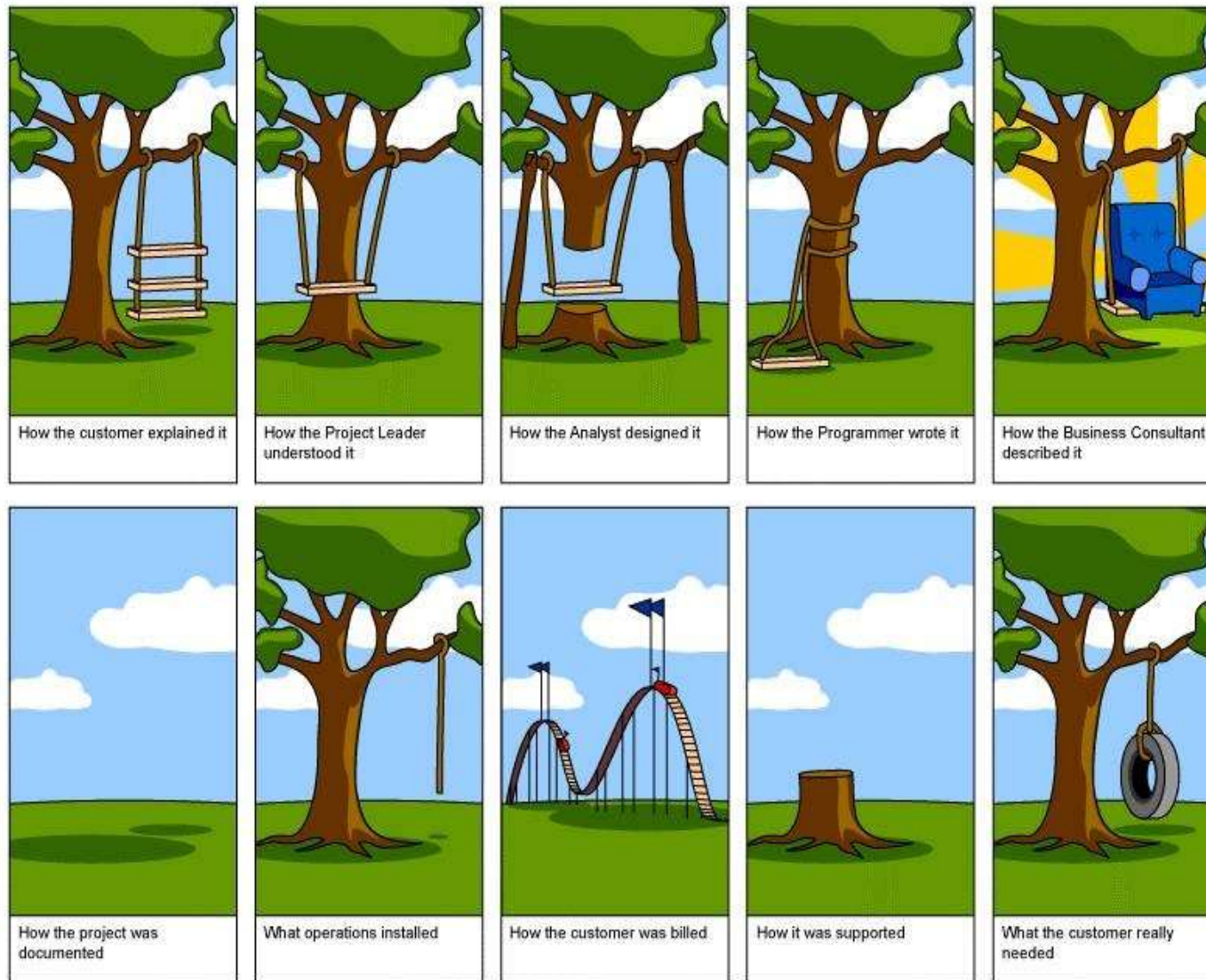
Standish Group CHAOS report (2010)

37% of all organizations reported inaccurate requirements as the primary reason for project failure.

- Only 49% of organizations have the resources in place to do requirements management properly.
- Only one-third of organizations' leaders value requirements management as a critical competency.
- Only 47% of organizations have a formal process to validate requirements.
- 51% of project and program dollars are wasted due to poor requirements management.
- 47% of unmet project goals were due to poor requirements management.

PMI's *Pulse of the Profession*® (2014)

Problems with requirements



Problems with requirements



Problems with users and customers

Problems with finding requirements



Problems with documenting requirements

Problems with managing requirements



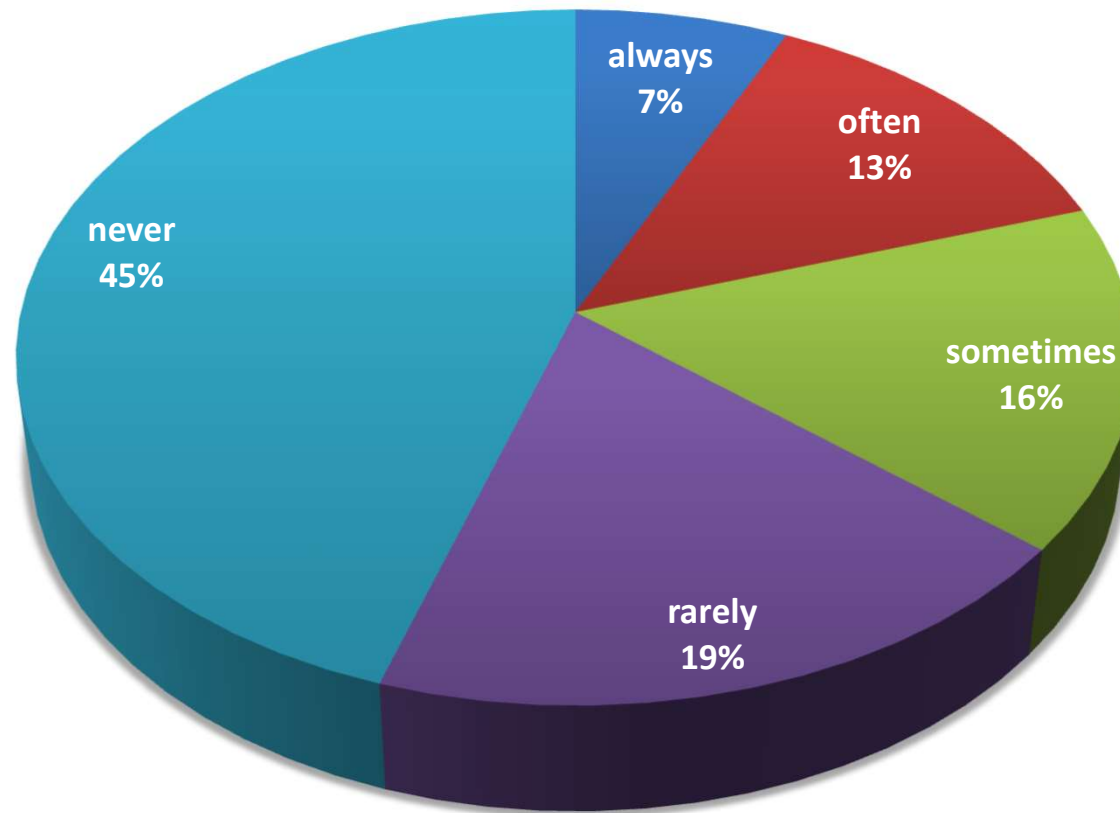
Problems with the users of the requirements

Problems with users/customers

- Customers don't know exactly what they want.
- Customers don't want to spend time on requirements elicitation.
- Customers don't want to make decisions.
- Customers ask for the solution, instead of their need.
- Customers don't understand technology.
- Customers assume they receive the right product they need.
- Users have difficulty with changes.
- Users think differently about what the system should do.
- Different users use the system for different tasks.
- Customers find it hard to prioritize requirements.
- Yes, but,... syndrome
- User and developer syndrome

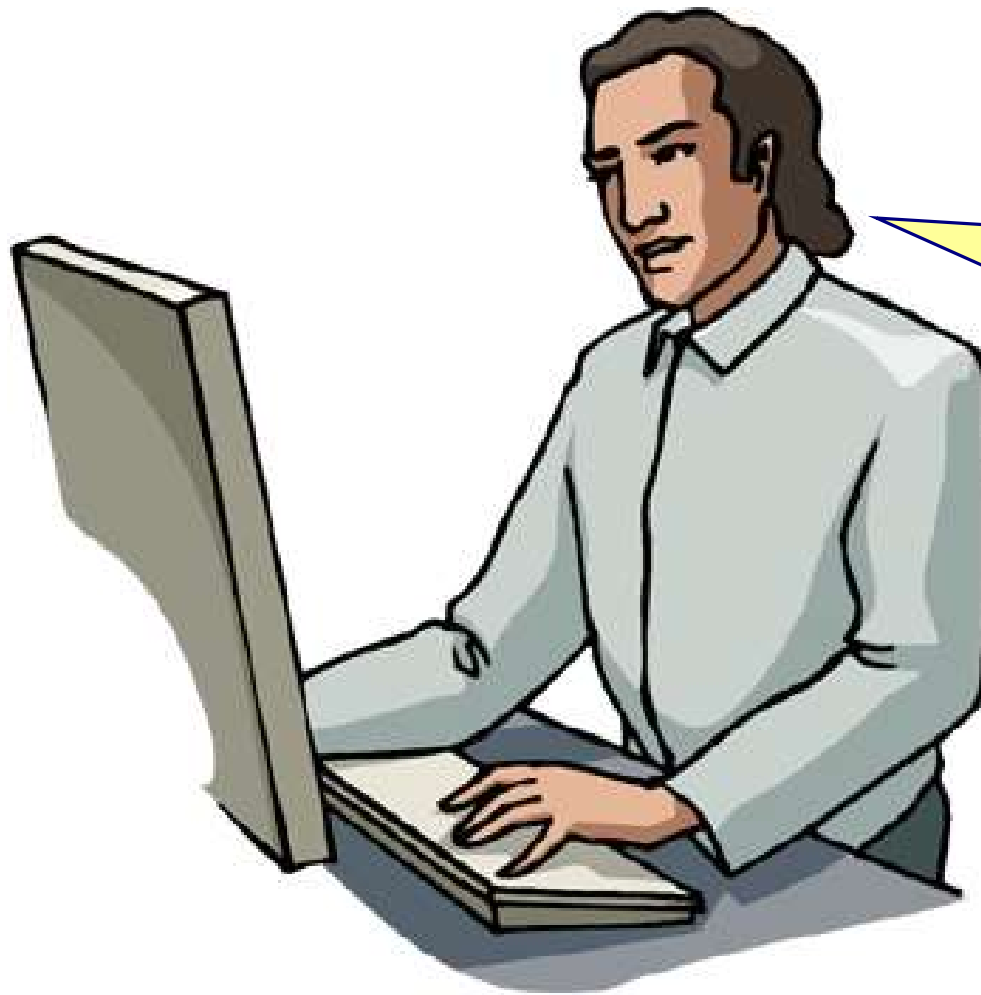


How much features are actually used?



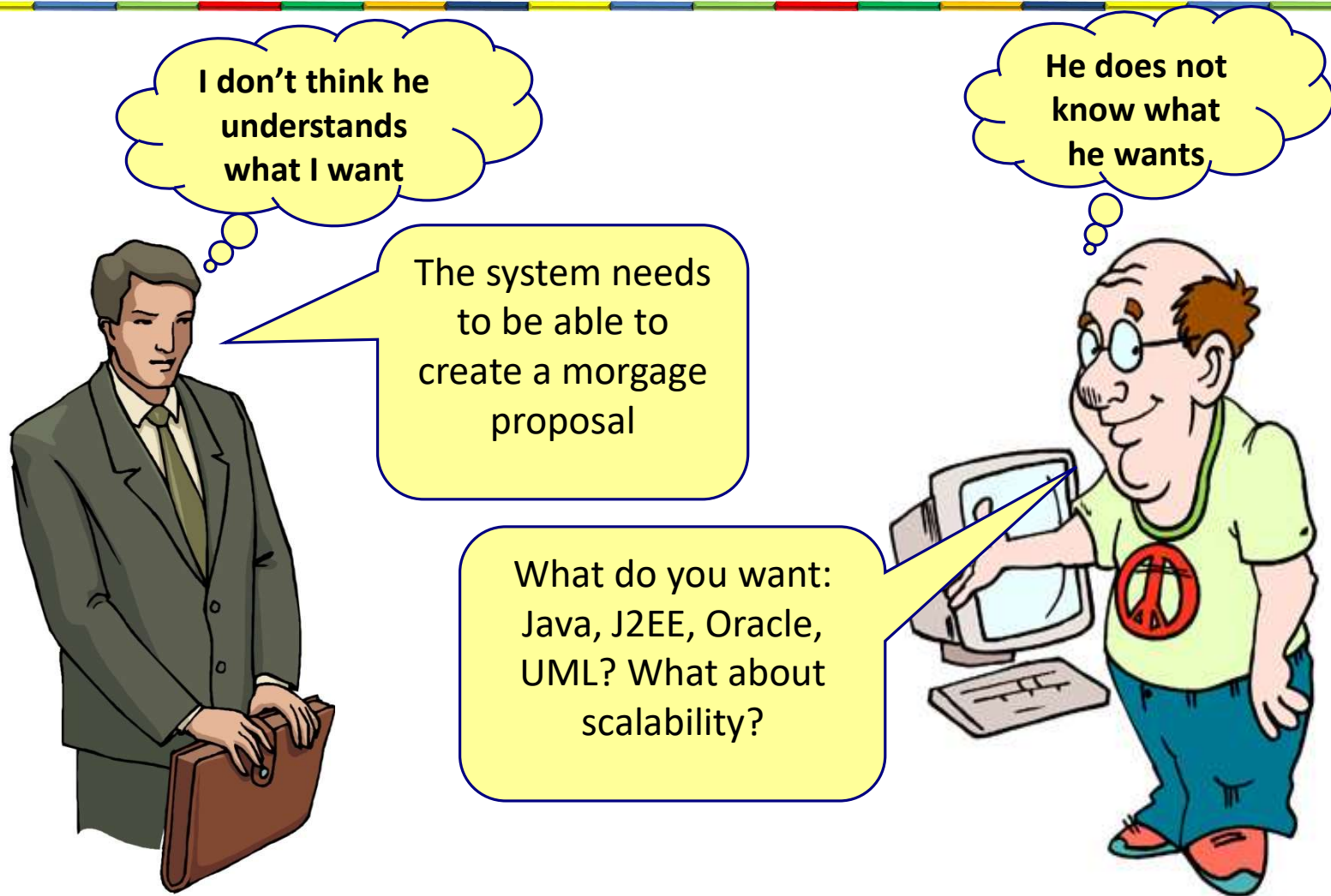
Source: the CHAOS report, Standish group (2004)

“yes, but...” syndrome



Yes, but, hmmm
Now that I see it, can we
change this into...?
Would it not be nice if...?
What happened with...?

The “user and developer” syndrome



Problems with finding requirements

- Not enough time is spend on finding requirements.
- Requirements are not tested
- Only one view is used
- Analysis paralysis, one takes too much time to find all requirements beforehand.
- Undiscovered ruins syndrome



How much effort do I spend on requirements?



Research on 15 bank and telecom projects:

- Most successful project : 28% effort is spent on requirements
- Average: 15.7% effort is spent on requirements

NASA projects:

Percentage spend on requirements	Average extra cost for too late delivery
Smaller than 5%	125%
5% to 10%	83%
More than 10%	30%

European research:

The fastest project teams spend twice as much of their time (17% vs. 9%) and effort (14% vs. 7%) on requirements compared to the slower teams

Spending more time on requirements can lower the duration of the project

“Undiscovered Ruins” syndrome

- The more you find, the more you realize still remains .
- You are never done with requirements analysis.



Problems with documenting requirements

- Too long
- Too technical
 - Design in requirement documents
- Not complete requirement documents
- There are a lot of different requirements
 - Business requirements
 - Functional requirements
 - Non-functional requirements



Problems with managing requirements



- Changing requirements
- Scope creep
- Requirements are hard to trace

Problems with users of the requirements (developers)



- Gold plating
- Implementation starts without clear requirements
- Developers don't understand the domain

Problems with requirements



Problems with users and customers

Problems with finding requirements



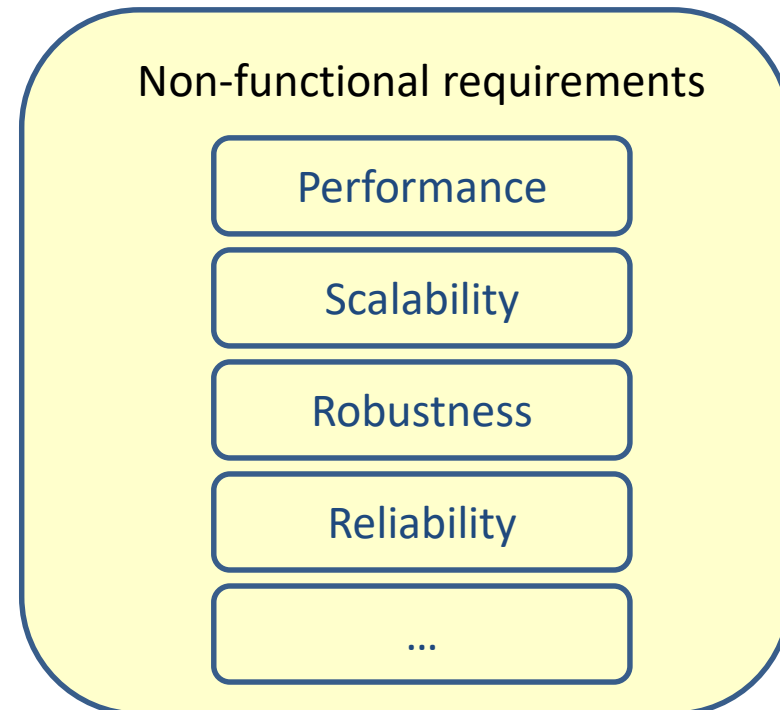
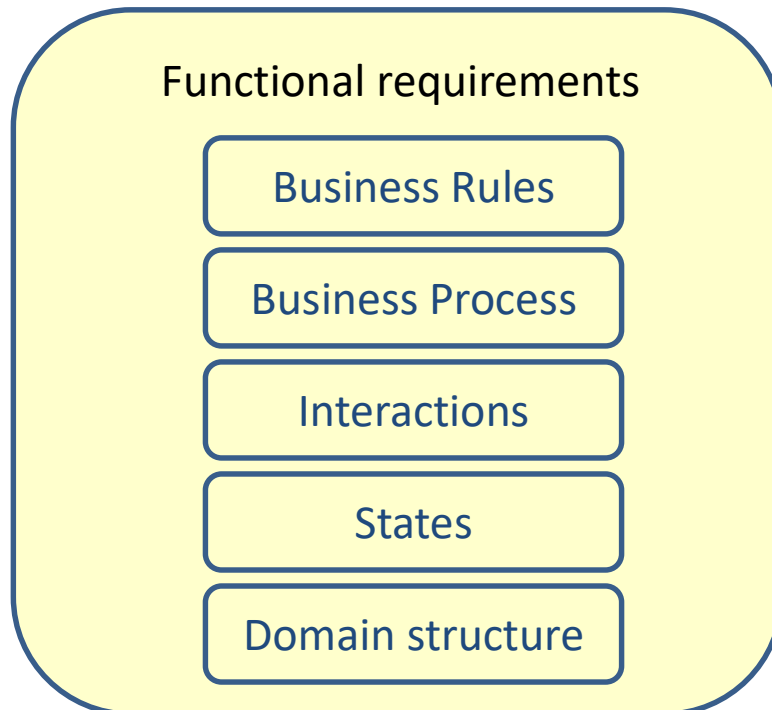
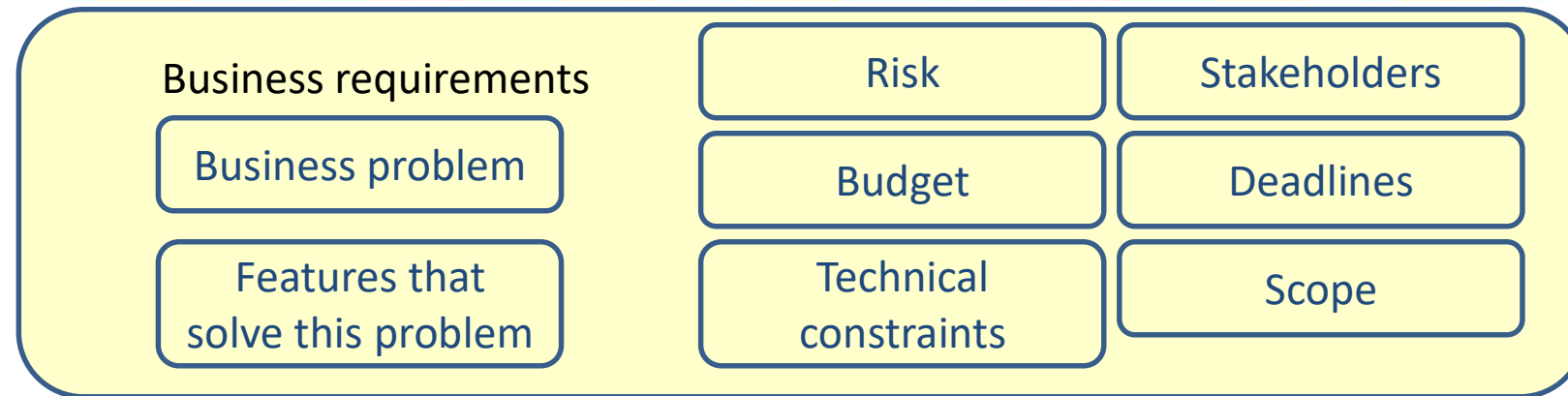
Problems with documenting requirements

Problems with managing requirements



Problems with the users of the requirements

Different types of requirements



Specifying functional requirements

- Text and diagrams
- Requirement statements
- Use Cases
- User stories

Text and diagrams

- Everyone writes the requirements in their own specific format
- Box and line diagrams are often not clear



- We don't know if we are complete
- The focus is on the system
 - Not on the interaction with the system

Requirement statements

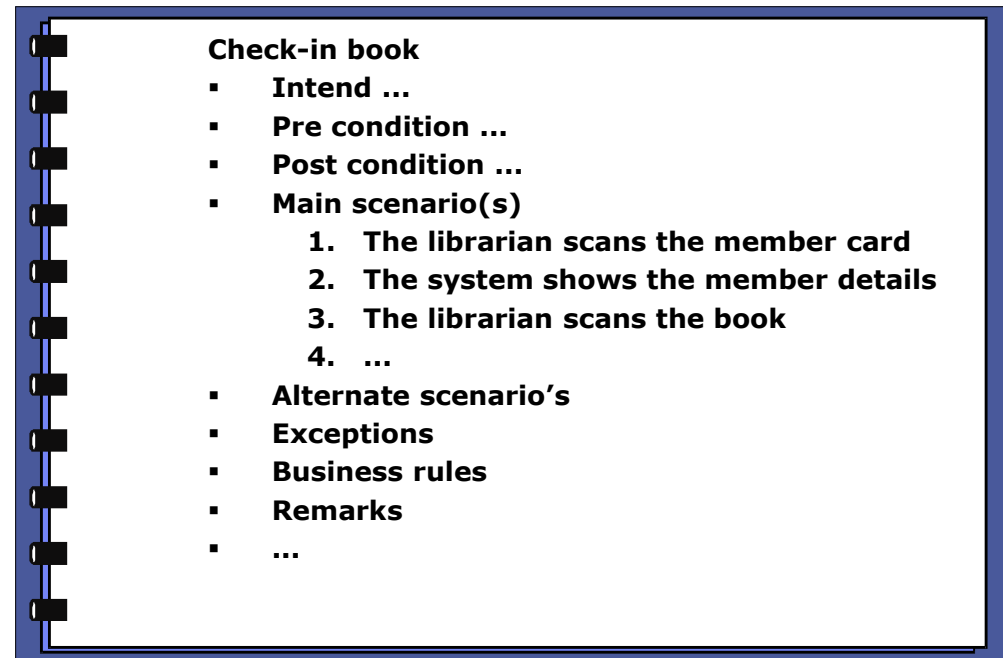
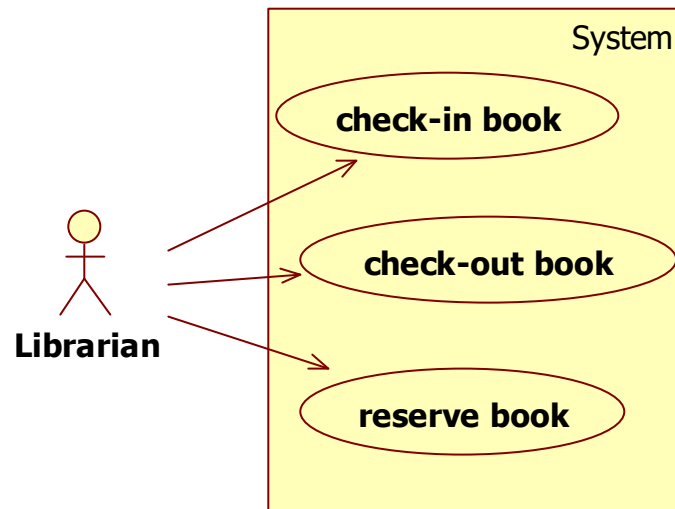
- The system shall allow users to search books in the library. Users can search on the following items: *isbn* number, *title*, *author(s)*, *year* (of publication).
- The system shall contain information about each book in the library: *isbn* number, *title*, *author(s)*, *year* (of publication), *copy number*, *shelfid*, current *status* (on-loan, on-shelf, on-hold, on-loan-and-on-hold), the date on which the book entered loan and/or hold status.
- The system shall allow users to reserve books. Users can reserve maximal 5 books.
- The system shall record all loan information.
- The system shall compute the fee if a book is returned too late
- ...

Characteristics of requirement statements



- By reading the requirement statements it is not always clear
 - What problem the system solves
 - How the system is used
- We don't know if we are complete
- Lots of statements
 - Difficult to understand
 - Difficult to manage
- The focus is on the system
 - Not on the interaction with the system

Use cases



Characteristics of use cases



- Describe how the user uses the system
 - The focus is on the interaction with the system, not the system itself
- Work very well for requirements
- Much easier to see if we are complete

User stories

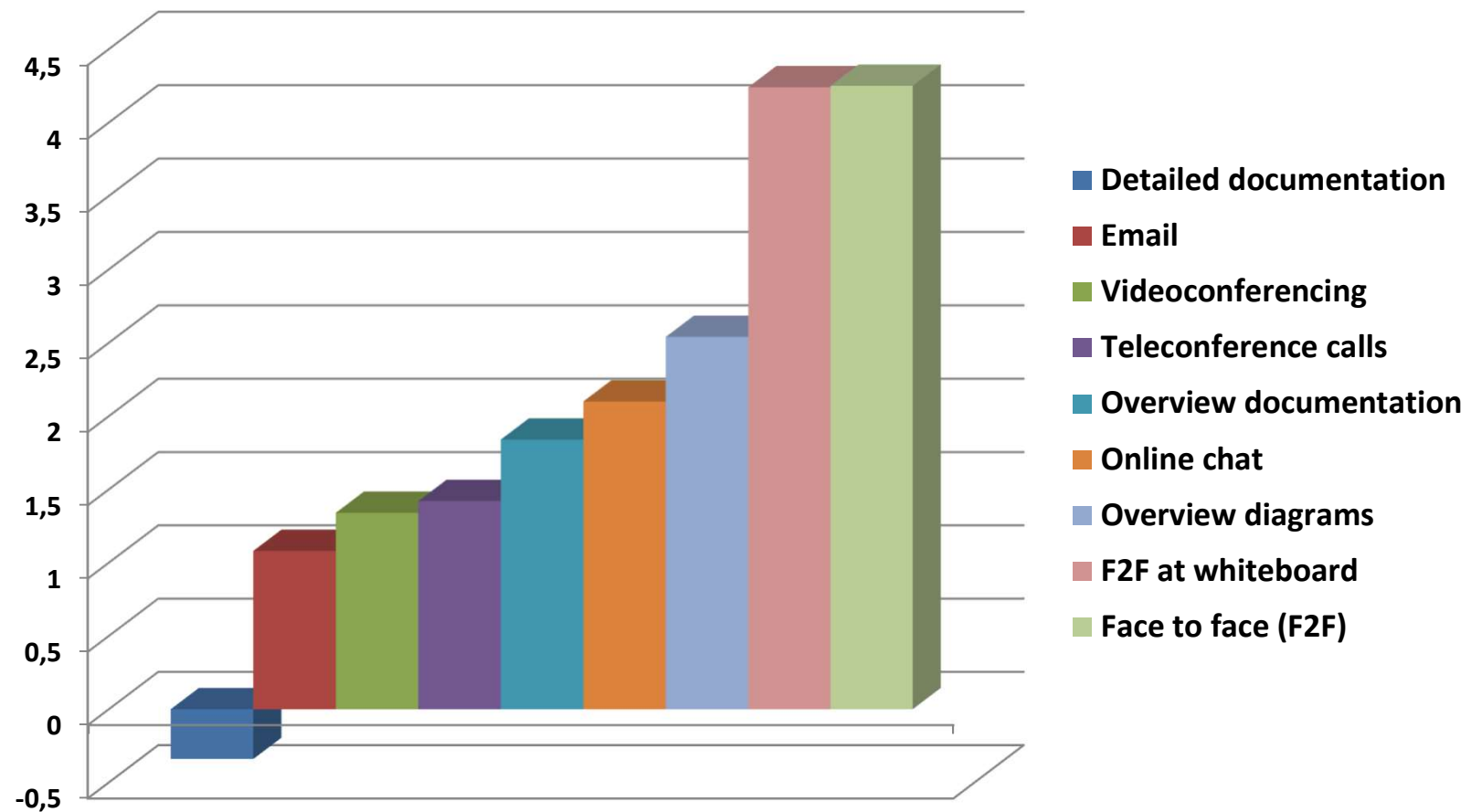
As a librarian I can check-out books so that I can record when customers take books out of the library

As a librarian I can check-in books so that I can record when books are returned

As a librarian I can see the customer details so that I know all necessary details of our customers



Effectiveness of communication



Characteristics of user stories



- Act as a reminder of requirements
 - The focus is on the conversation with the user
 - Not on writing down the requirements
- Effective way of communication
- You need the availability of a user all the time