Agile usage centered software development

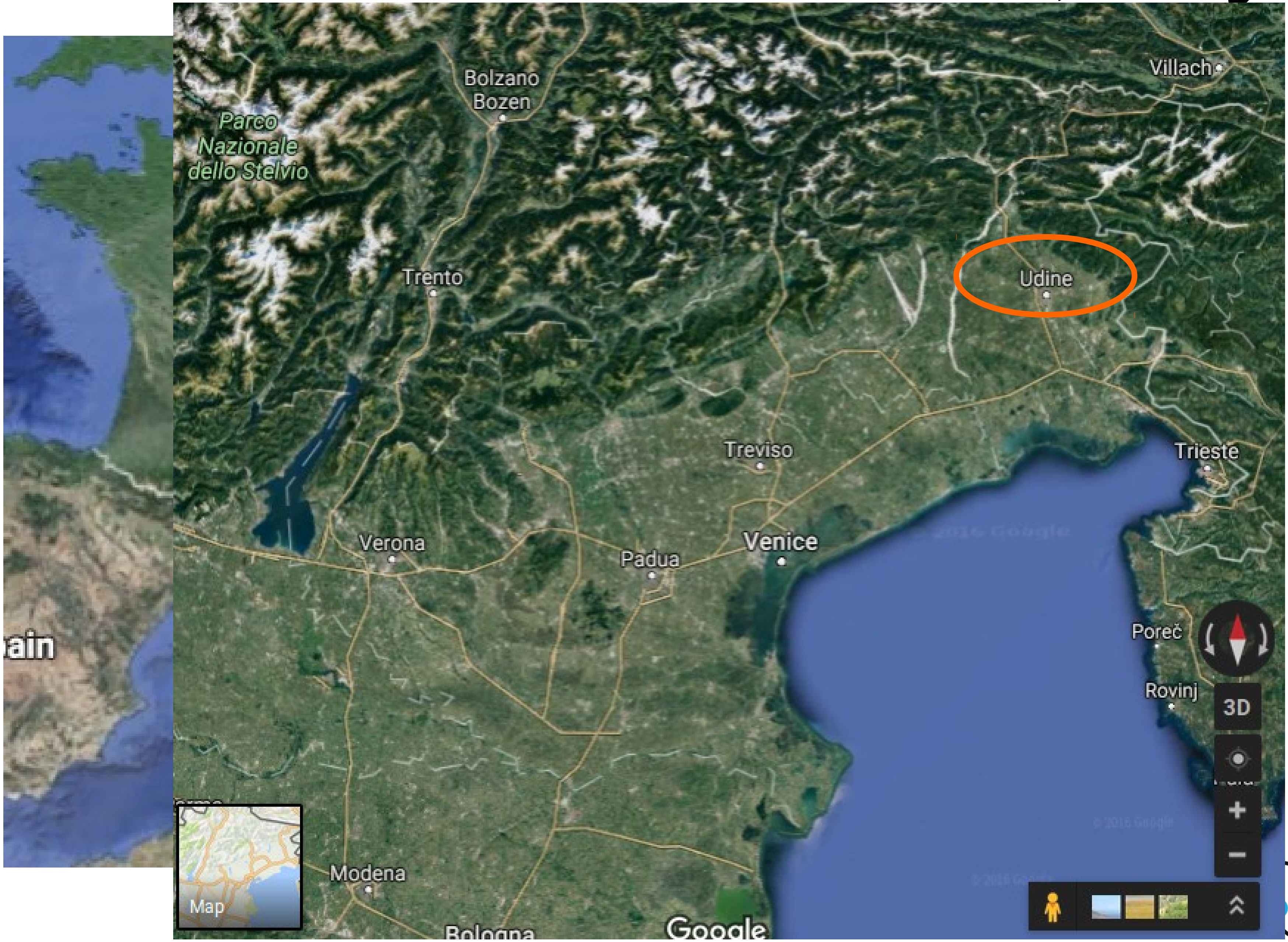
Giorgio Brajnik

Interaction Design Solutions Srl
giorgio.brajnik@designcoaching.net

October 2016

IDS
Design Coaching for Excellence

Udine, Italy



Bolzano
Bozen
Parco Nazionale dello Stelvio
Villach
Trento
Udine
Treviso
Trieste
Verona
Padua
Venice
Poreč
Rovinj
3D
Modena
Map
Bologna
Google
ain

Interaction Design Solutions S.r.l.

IDS

Design Coaching for Excellence

Who we are

- Spinoff of the University of Udine, Italy
- People:
  - ISTQB certified employees
  - three partners
- Our focus is **Quality Engineering**
  - “We **help** you produce **usable** and **solid** software”
  - **Quality** of: user interfaces, software architectures, testing processes, requirements engineering

IDS
Design Coaching for Excellence

Interaction Design Solutions

- our clients
  - software houses
  - companies with internal development
  - companies that do outsourcing
- at the moment:
  - Danieli Automation, Overit, Tecnest, University of Udine, Teletronica, INAF, Phoenix

IDS
Design Coaching for Excellence

Software development

Software development

The goal is

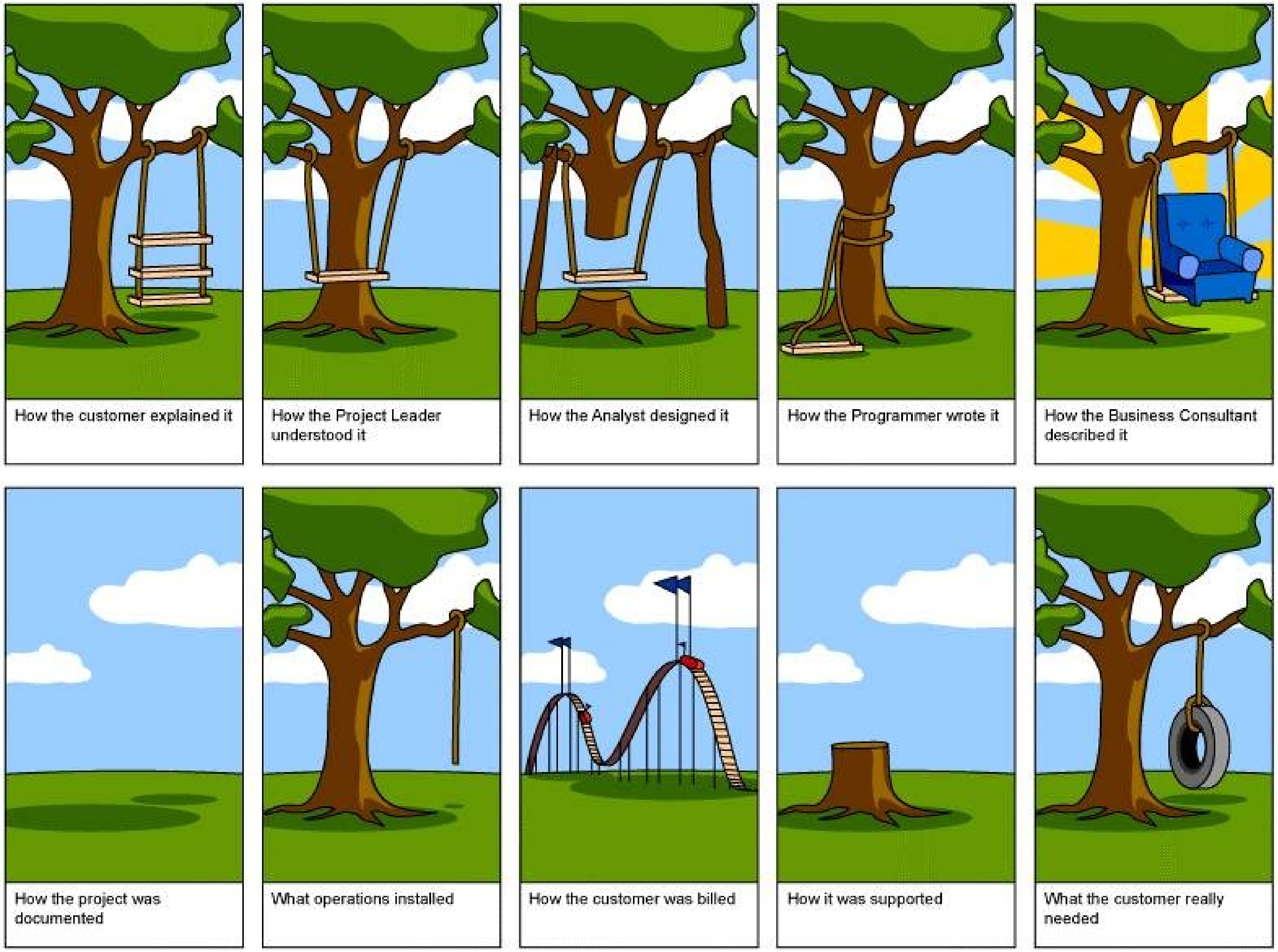
- to develop quality software, within allotted time and budget, such that it satisfies **stakeholders needs**

Stakeholder need:

- an expression of the business problem that has to be addressed in order to justify the product

IDS
Design Coaching for Excellence

Product risks


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Product risks

Due to uncertainty about the product to be delivered:

- is it the right one?
- what could be wrong?
- would users accept it?
- would users be able to use it?
- would they achieve what they need to?

Process risks

Due to uncertainty about the development process:

- what can go wrong?
- technologies that don't work well together?
- people that don't work together?
- misunderstanding?
- inefficiencies?

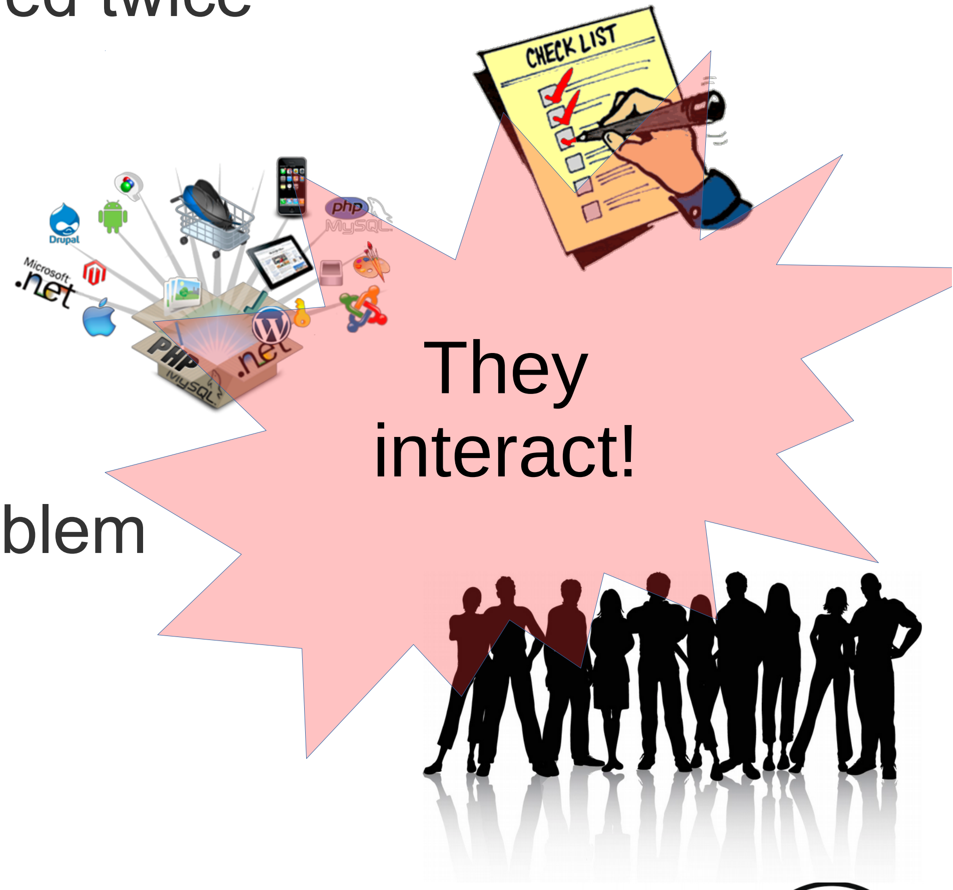
The problem

- **Requirements**

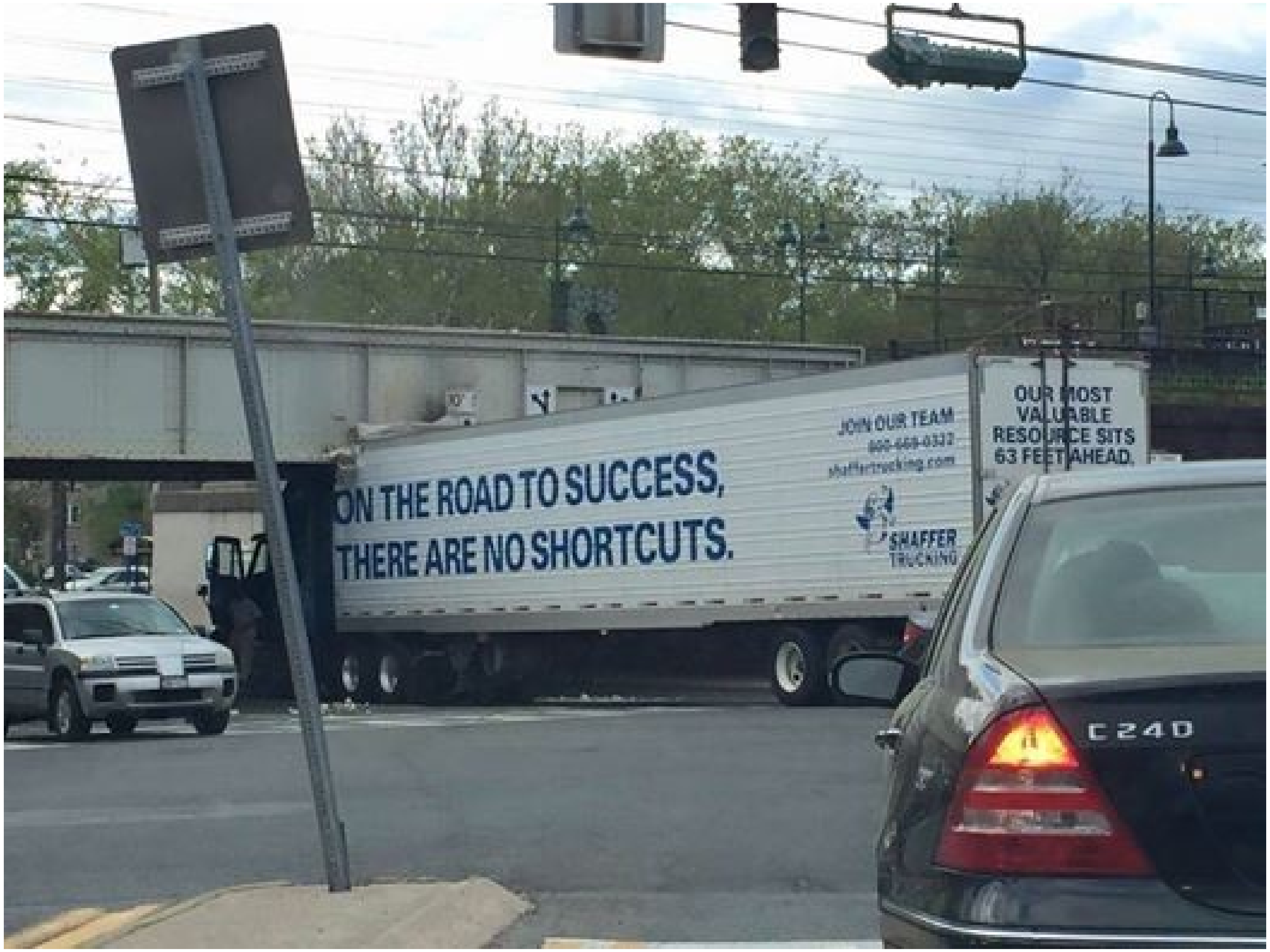
  - rarely the same problem is tackled twice
  - rarely they are stable
- **Technologies**

  - complex
  - unexpected interactions
  - more or less suitable for the problem
- **People**

  - development team
  - deployment team
  - clients/users

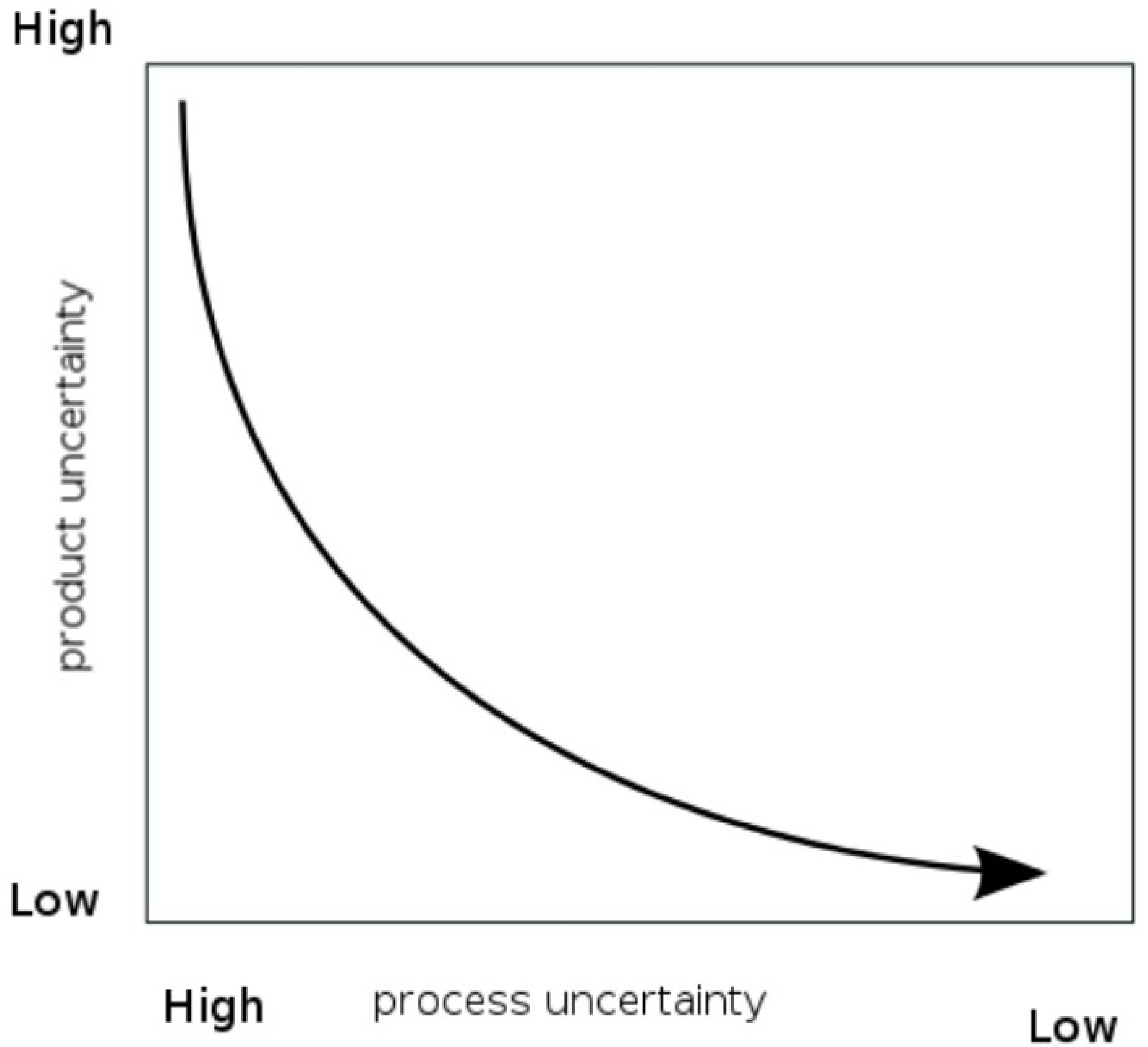


Risks have to managed

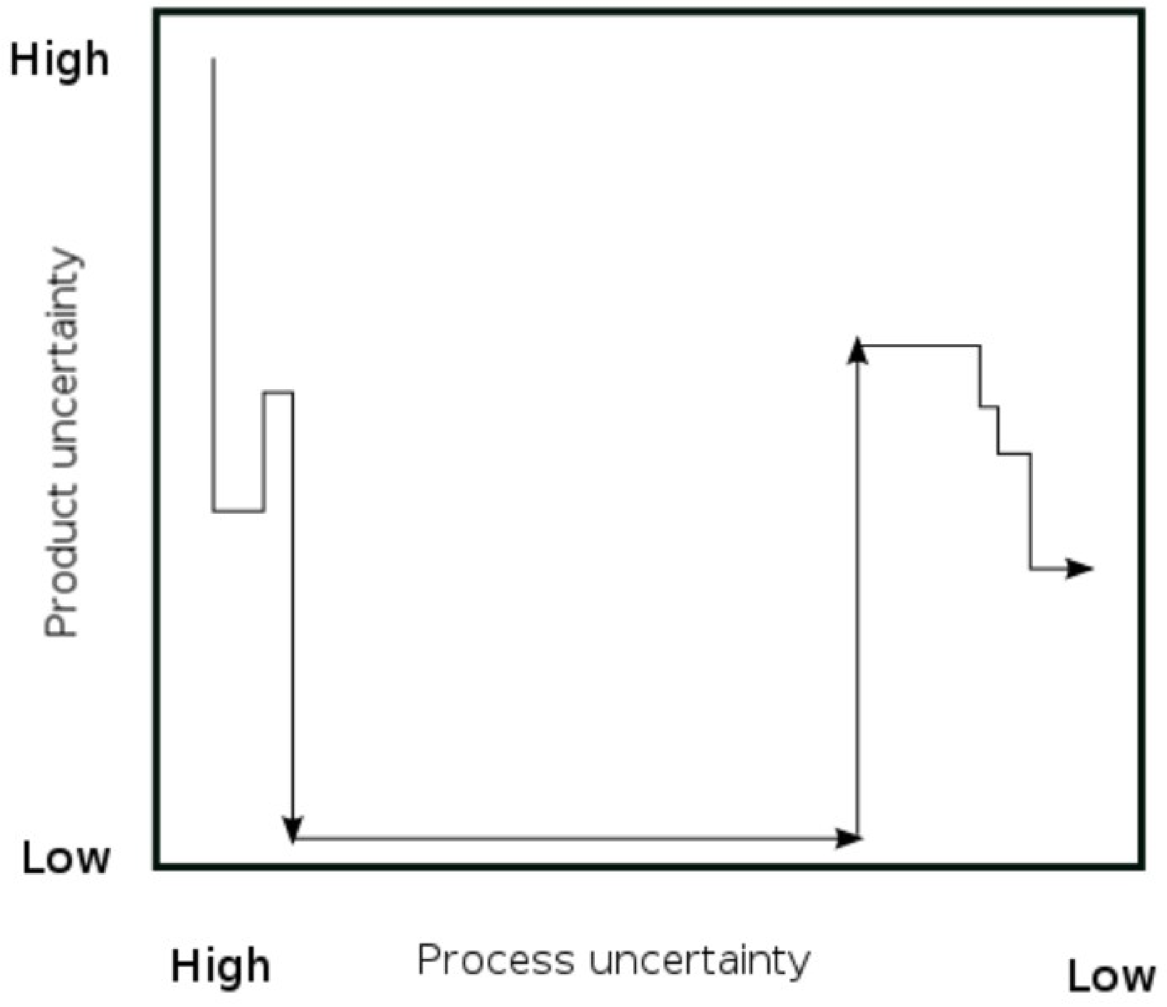


IDS
Design Coaching for Excellence

Ideal situation



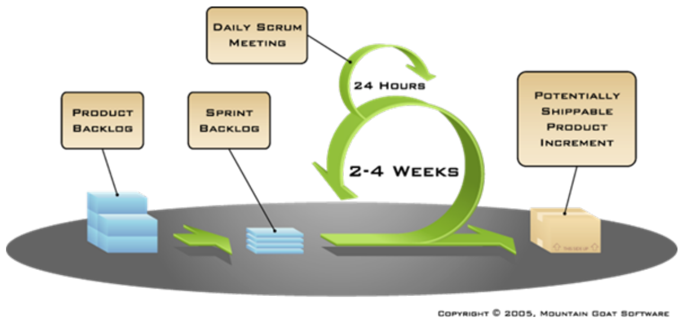
Waterfall approach



Agile management

Essential aspects of agile approaches

1) Development is **requirement-driven**
2) Planning is **distributed** over time


Daily Scrum Meeting
24 Hours
Product Backlog
Sprint Backlog
2-4 Weeks
Potentially Shippable Product Increment
Copyright © 2005, Mountain Goat Software

3) Work is organized in **short iterations**
4) Each iteration releases a potentially **deliverable chunk** of the product
5) Development team **commits** to the delivery of an iteration

Usage Centered Development

IDS
Design Coaching for Excellence

Three basic ideas

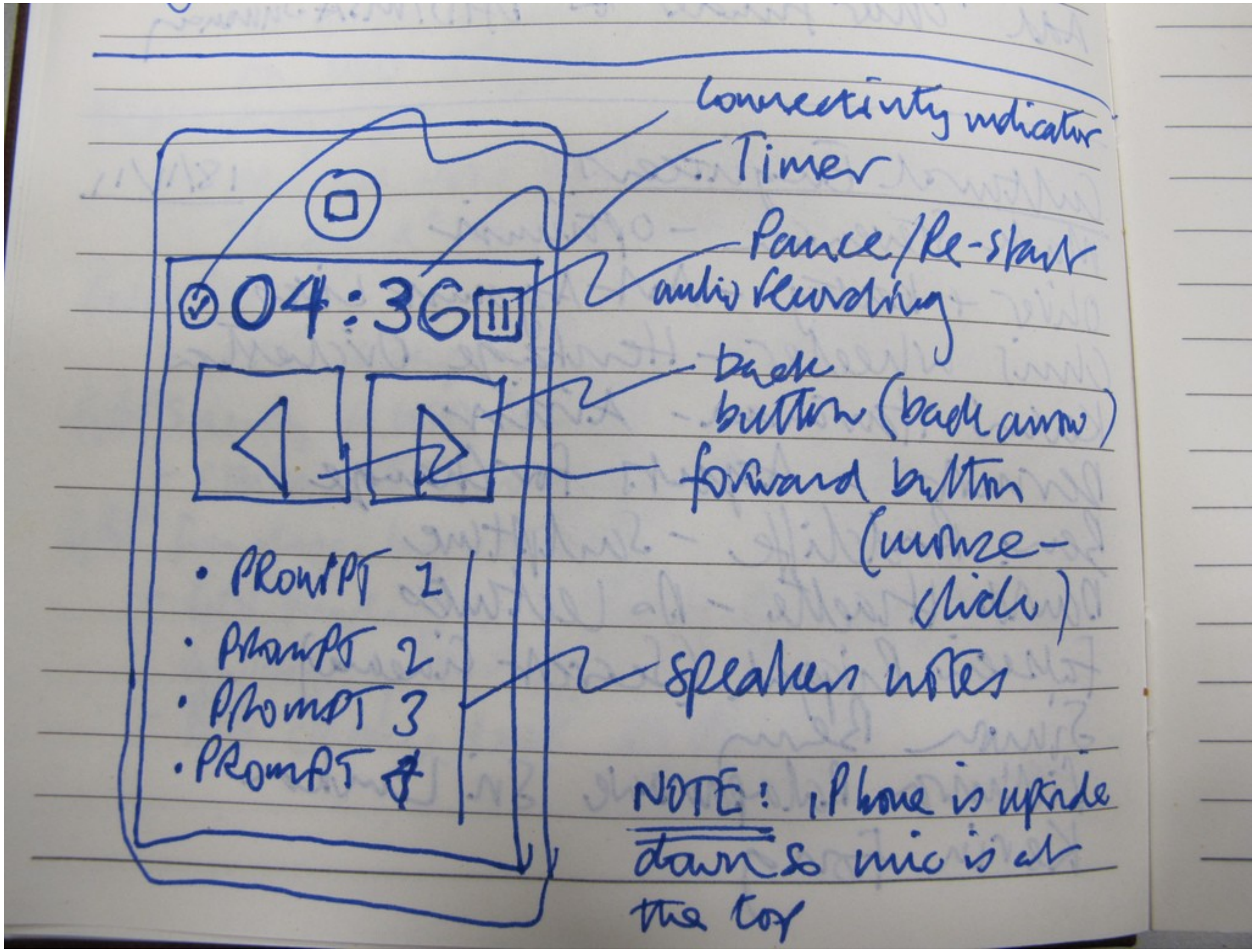
- Early focus on users and on understanding their needs
- Empirical evaluations of usability
- Iterative development
  - plan a prototype to investigate a risk
  - build it
  - evaluate it

Some UCD techniques

- definition of user profiles
- contextual enquiry
- interviews
- affinity diagrams
- definition of personas and scenarios
- sketching + storyboarding
- essential use cases
- task modeling
- conceptual design
- user testing

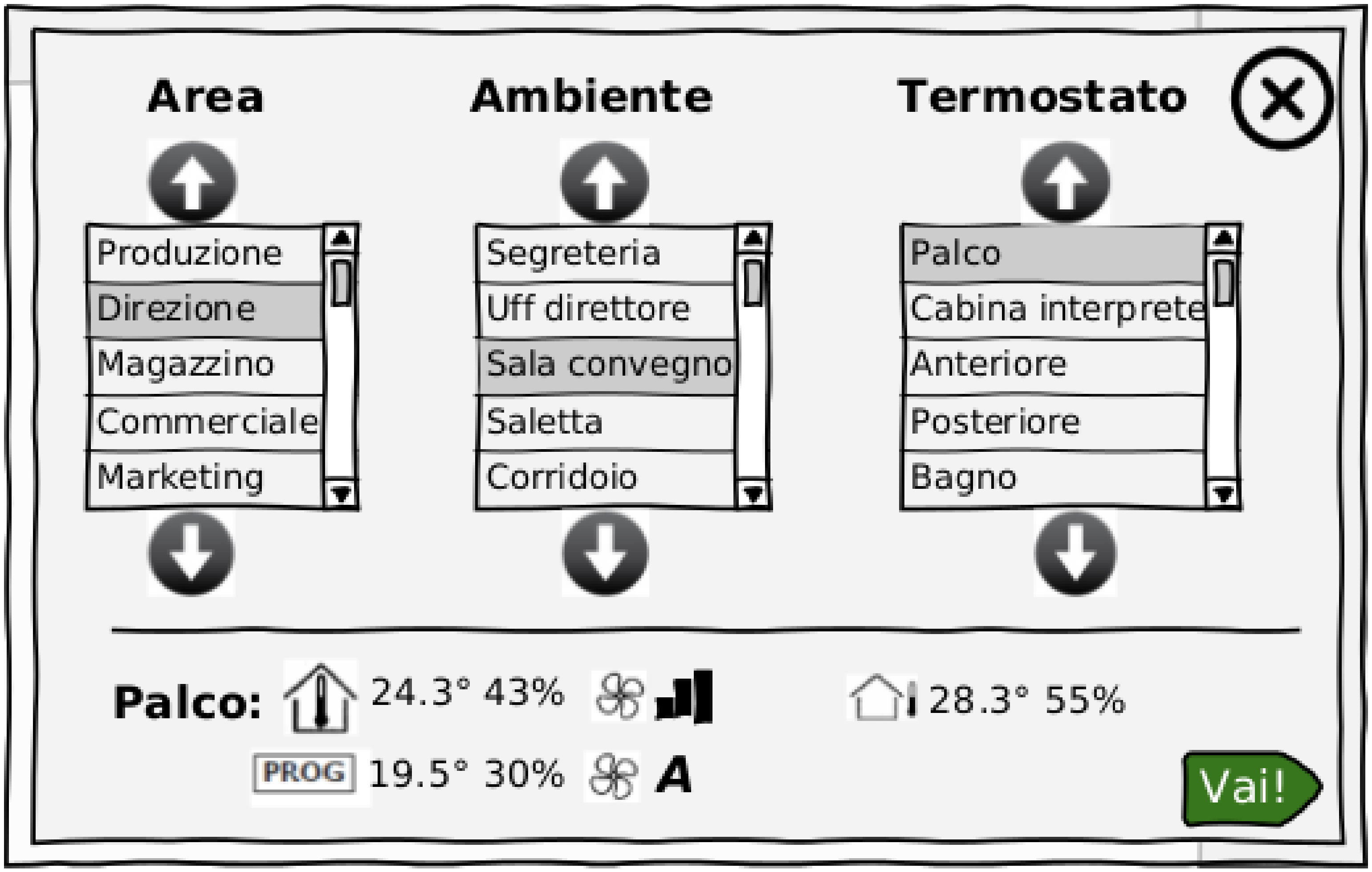
Sketching e storyboarding

Sketches



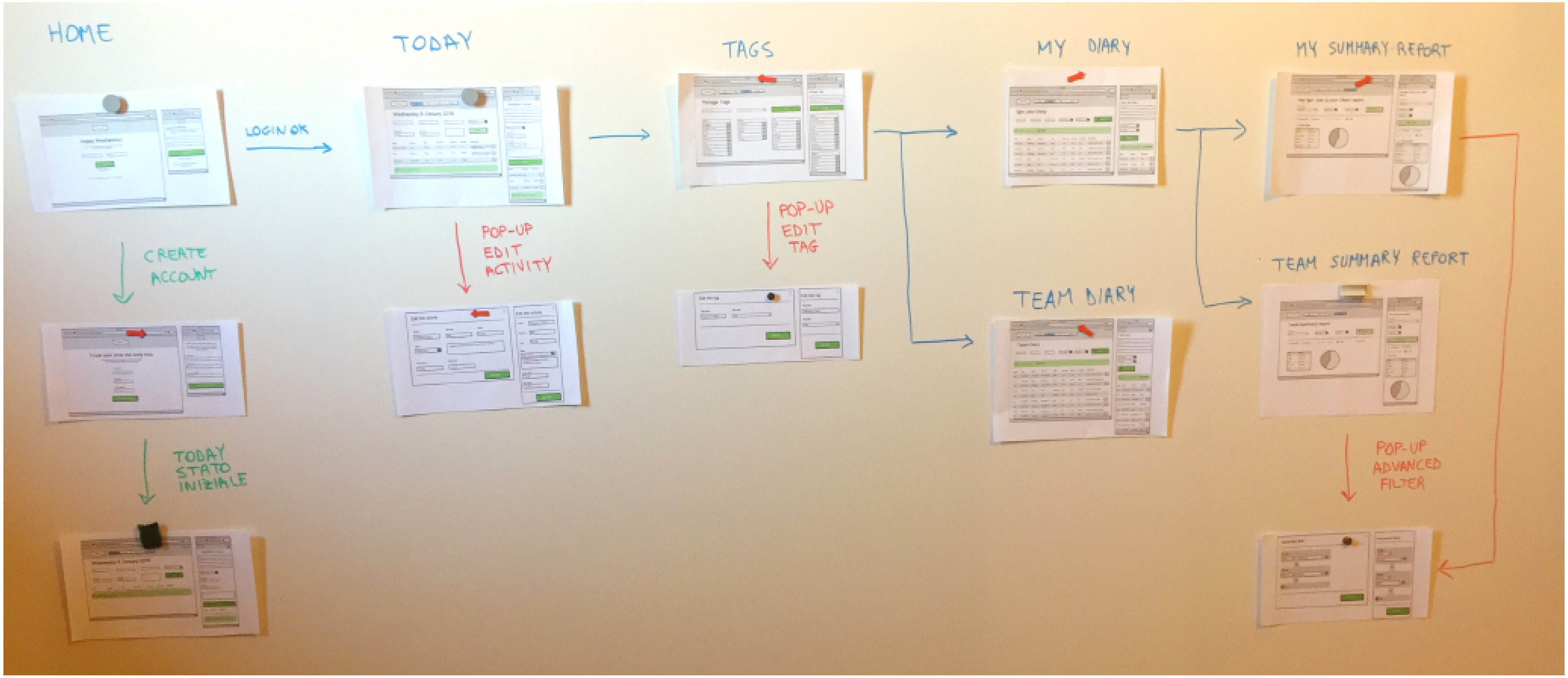
IDS Design Coaching for Excellence

Sketches



IDS
Design Coaching for Excellence

Storyboard



(I. Odorico, 2016)

IDS Design Coaching for Excellence

Conceptual design

Conceptual model

IDS Design Coaching for Excellence

Materials, tools and interaction spaces

IDS
Design Coaching for Excellence

Materials, tools and interaction spaces

IDS Design Coaching for Excellence

Testing a sw system

Functional testing

- required to ensure high quality
  - to find bugs
  - to estimate quality
  - to identify critical areas
  - to estimate customer-support costs
  - ...
- at different levels: unit/integration/system/intra-system tests
- different channels: developers, QA team, beta-users
- different mechanisms: scripted, exploratory, automated
- different moments: while developing, regression, before releasing

The problem

- Insufficient testing
  - → low quality
    - → poor customer satisfaction
    - → brand damage
    - → customer support costs
    - → fixing costs
- Inefficient development process
  - → made it worse by short deadlines
    - → poor results
    - → overtime
    - → increase of technical debt
    - → turn-over

IDS

Design Coaching for Excellence

The problem (2)

- Poor quality control
  - → no indicators on quality of a release
  - → unknowns:
    - → how many bugs
    - → their impact on users
    - → which part of the product is affected
  - → no visibility on quality trends

IDS

Design Coaching for Excellence

121-T (one-to-one testing)
What & Why

IDS
Design Coaching for Excellence

Fast & High quality

- **End-to-end tests** of web apps
  - through the user interface
  - for user acceptance testing
  - for system testing
- Run **automatically**
  - fast
  - convenient
  - reliable
- For **each release**
  - as often as practical
  - with understandable reports

Innovation factor:

- **Model-driven Techniques** that enable us to be quick and agile

IDS
Design Coaching for Excellence

With 121-T

- Test cases can be **run as often** as needed

  → regression testing
- Test cases can fully cover **realistic usage scenarios**
  - based on simulating users of the UI

  → use case testing
- Test cases can be easily **extended**
- Test reports are **automatically** generated
- Test cases can be **quickly adapted** to changes in the User Interface

  → to cope with **frequent refactoring**

  → to support **agile development**

IDS Design Coaching for Excellence

Benefits

- Quick results: **control quality risks** as often as needed
- Fast setup: **change the UI** as often as needed
- Custom test cases: **easily understand** quality risks
- High level test cases: **easily specify new** tests
- Structural coverage metrics: **understand what to test**
- Requirements coverage metrics: **understand impact** of failures
- **Save time-for-testing** (→ 97%)
- **Save costs** (→ 80%)

IDS
Design Coaching for Excellence

Conclusions

- Usage-centered development reduces **product risks**
  - It is **cost-effective**
  - It can be paired to agile approaches to reduce **process risks**

- Appropriate and cost-effective **test automation** processes can be put in place

IDS
Design Coaching for Excellence

Contact

Giorgio Brajnik

Interaction Design Solutions Srl
giorgio.brajnik@designcoaching.net
www.designcoaching.net
+39 340 2722173

IDS
Design Coaching for Excellence