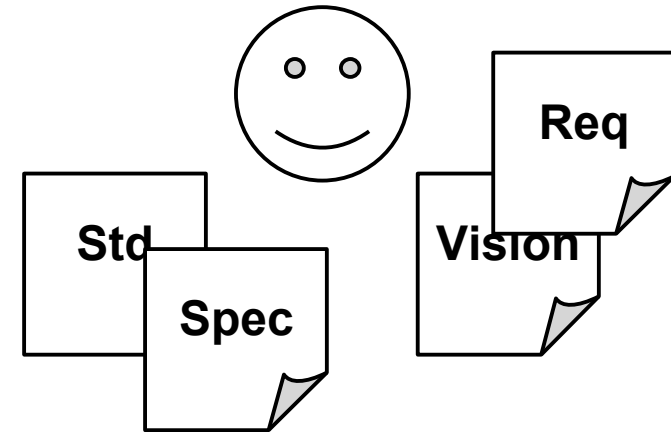# Beyond Documents:

Visualizing Informal Communication

Kurt Schneider, Eric Knauss, Kai Stapel

<firstname.lastname>@inf.uni-hannover.de

Software Engineering Group

Leibniz Universität Hannover

Overview

- **Motivation: beyond documents**

- **FLOW modeling concepts**
  - *Fluid information*
  - *Experience*
  - *Visualization*

- **Example: Security requirements**

- **Comparison with other notations**

The problem

- **Practitioners frustrated: RE is never perfect**

- **„All requirements are specified …"?**
  - *Conversations*
  - *Phone calls*
  - *E-mails*
  - *Meetings*
  - *Wikis*
  - *Blogs*
  - *Tradition*

> **Often ignored as „irrelevant"**
>
> **Why?**
> **Information forgotten**
> **No distribution**
> **No reconstruction**
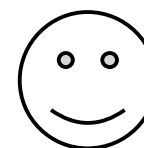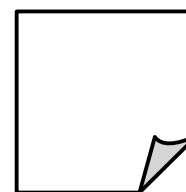>
> **Why popular?**
> **Fast**
> **Flexible**
> **Less effort, enjoyable**

Fluid and Solid information

- **Solid** representation:

  documents and stores

  (1) that can be retrieved by others

  (2) without the author or source

  (3) even after some time

  (4) in a form that supports dissemination.

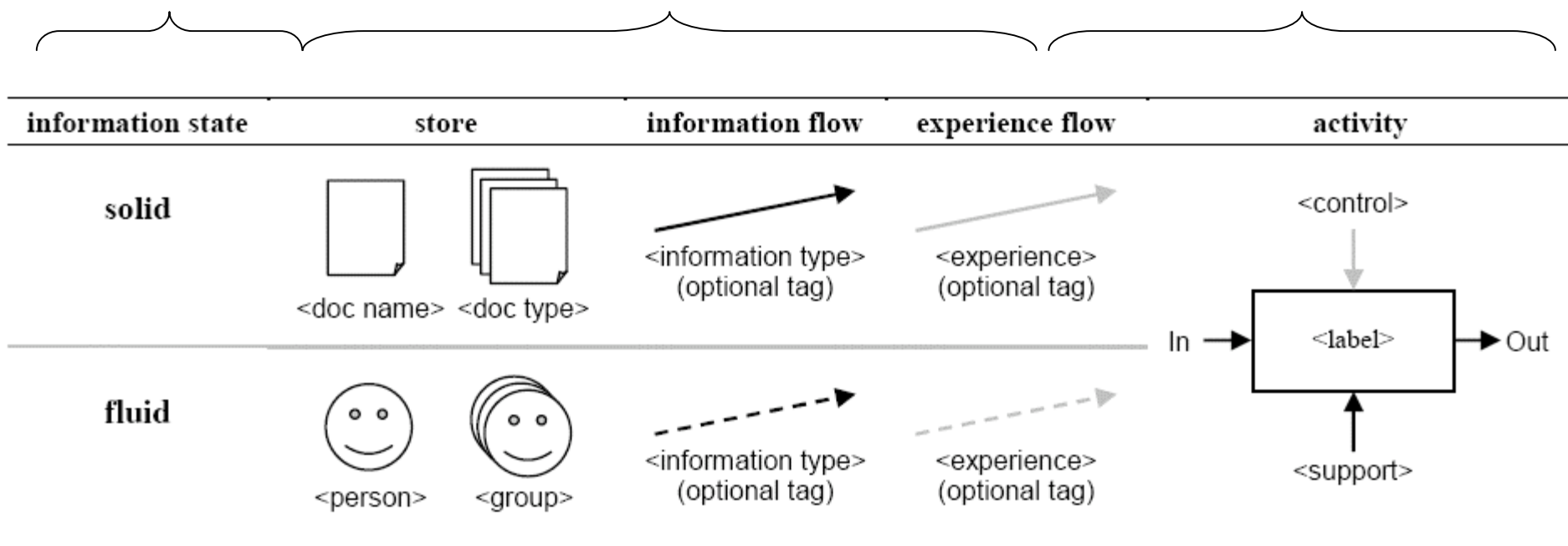- All other representations are called **fluid**.

Syntax of FLOW notation

**Metaphor**   **Typical pros&cons**   **Transformations**

information state	store	information flow	experience flow	activity
solid		<information type> (optional tag)	<experience> (optional tag)	<control>
	<doc name> <doc type>			In → <label> → Out
fluid		<information type> (optional tag)	<experience> (optional tag)	<support>
	<person> <group>			

**There are many „fluid cultures"
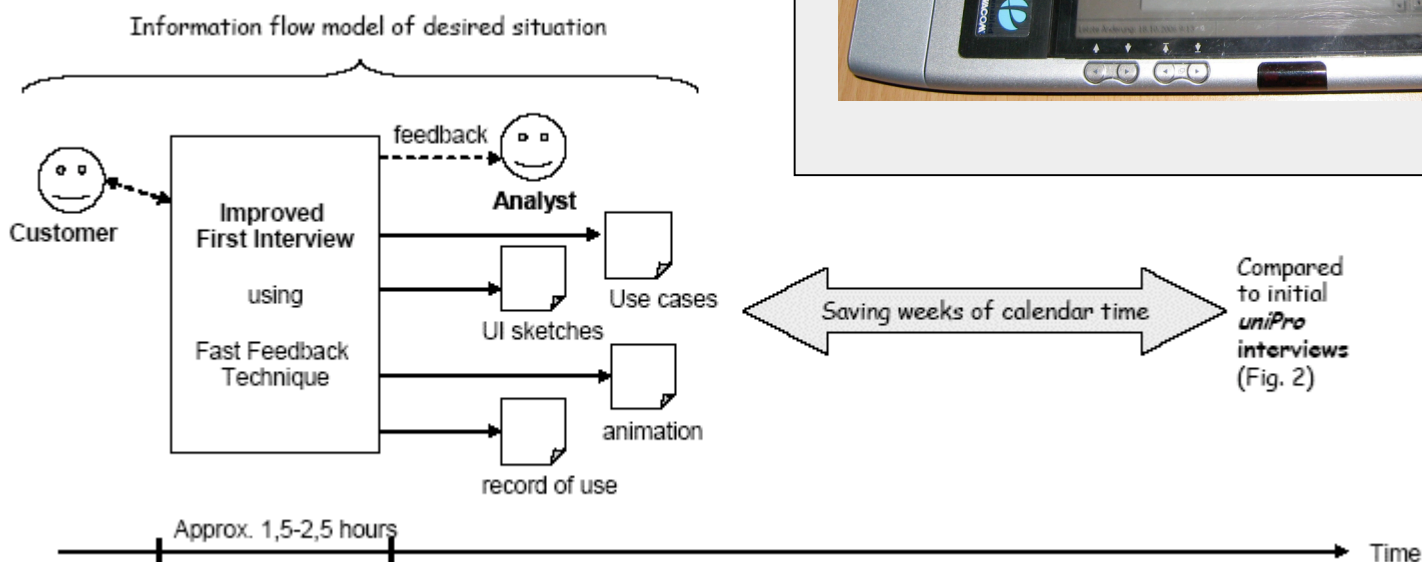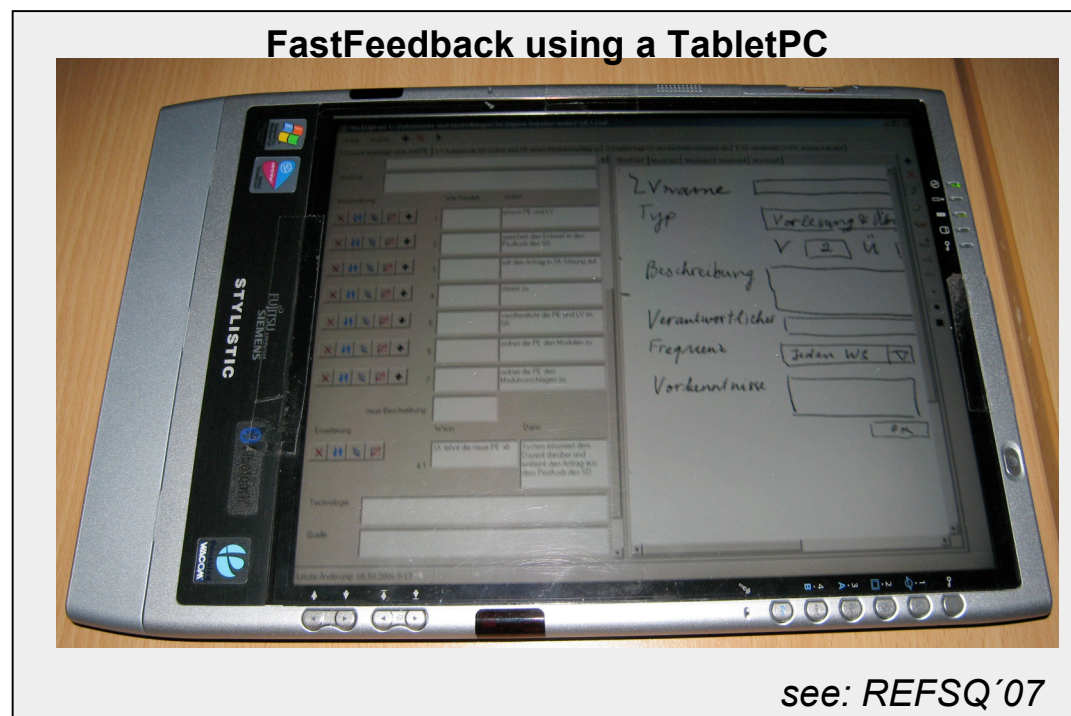(ok)**   **Explicit control of flows**

Visualization goals

- **Visualize solid and fluid**

- **Intuitive, use without training**
  - *Small set of simple symbols*
  - *on white-boards*
  - *in tools like Power-Point*
  - *In custom-built editors*

- **Link flow models to processes**
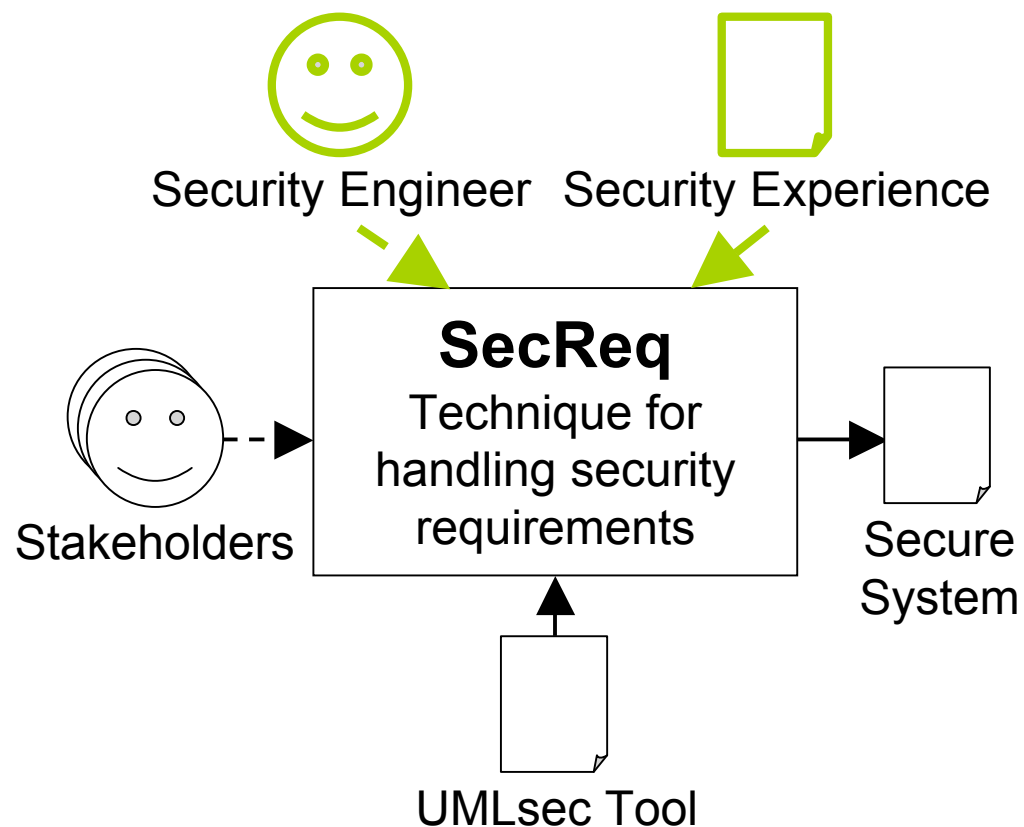 *Documents, Activities*

- **Reuse well-known notations**
  Comparison →
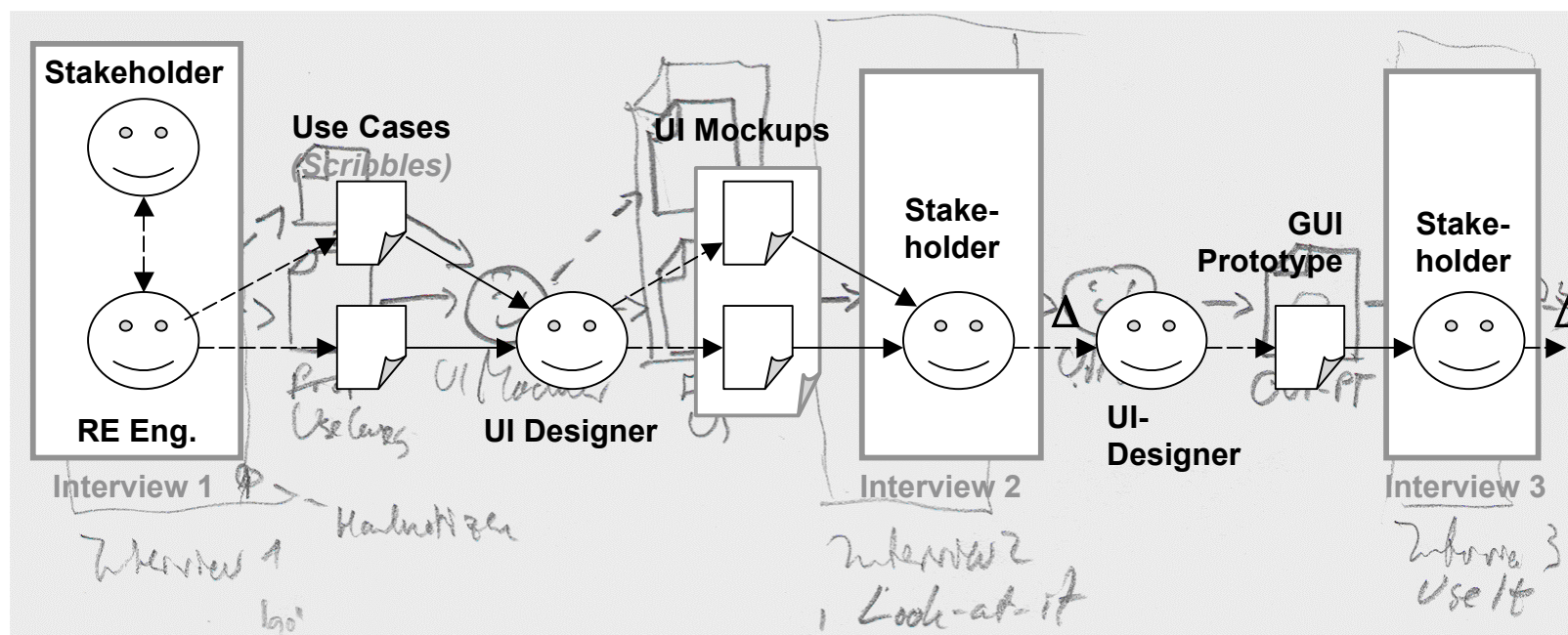
Leibniz
Universität
Hannover

- **Awareness and Overview**
- **Improving Requirements Processes and Practices**
- **Defining and Tailoring Communication**
- **Tools & Techniques**

FastFeedback using a TabletPC

see: REFSQ´07

Information flow model of desired situation

Customer — Improved First Interview using Fast Feedback Technique — feedback — **Analyst** — Use cases, UI sketches, animation, record of use

Saving weeks of calendar time — Compared to initial *uniPro* interviews (Fig. 2)

Approx. 1,5-2,5 hours — Time

For better visibility: green experience

Security Engineer   Security Experience

SecReq
Technique for
handling security
requirements

Stakeholders

Secure
System

UMLsec Tool

Initial elicitation: three alternative flows

(1)

Stakeholders   Initial SecReqs   Security Engineer   Improved SecReqs
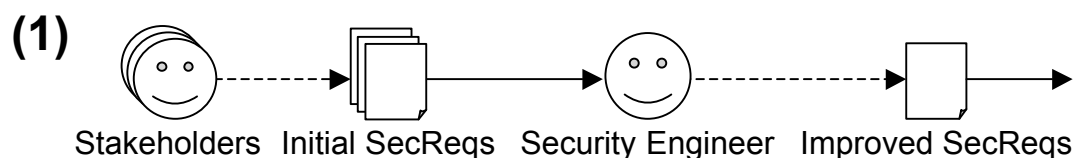
- **Stakeholders write**
- **Sec Eng reads**
- **Sec Eng integrates**

- **Stakeholders discuss**
- **Sec Eng moderates**
- **Sec Eng listens**
- **Sec Eng summarizes**

(2) Stakeholders

Security Engineer   Initial SecReqs

(3)

Security Instructor   Stakeholders   Security Requirements Elicitation   Security Engineer
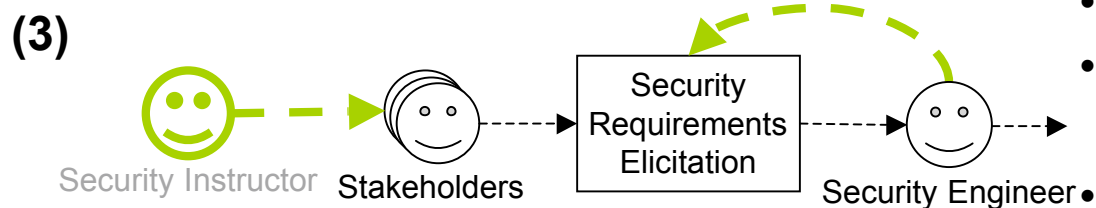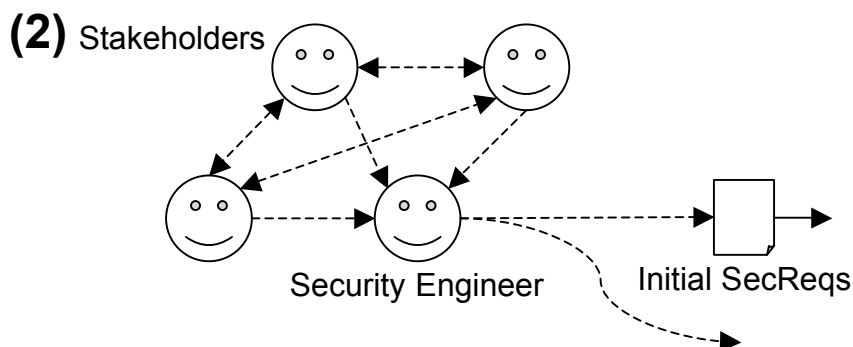
- **Instructor explains security**
- **Sec Eng elicits**
  - *one-by-one* or *all together*
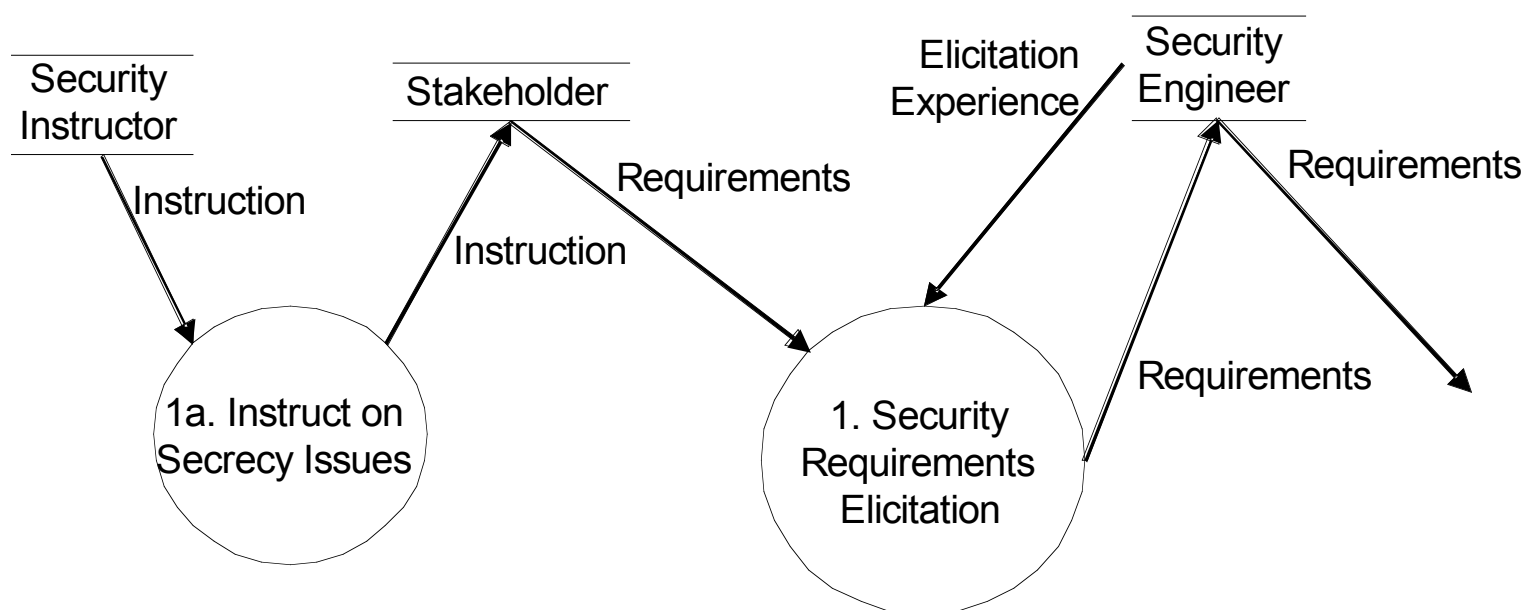- **Guided by experience**
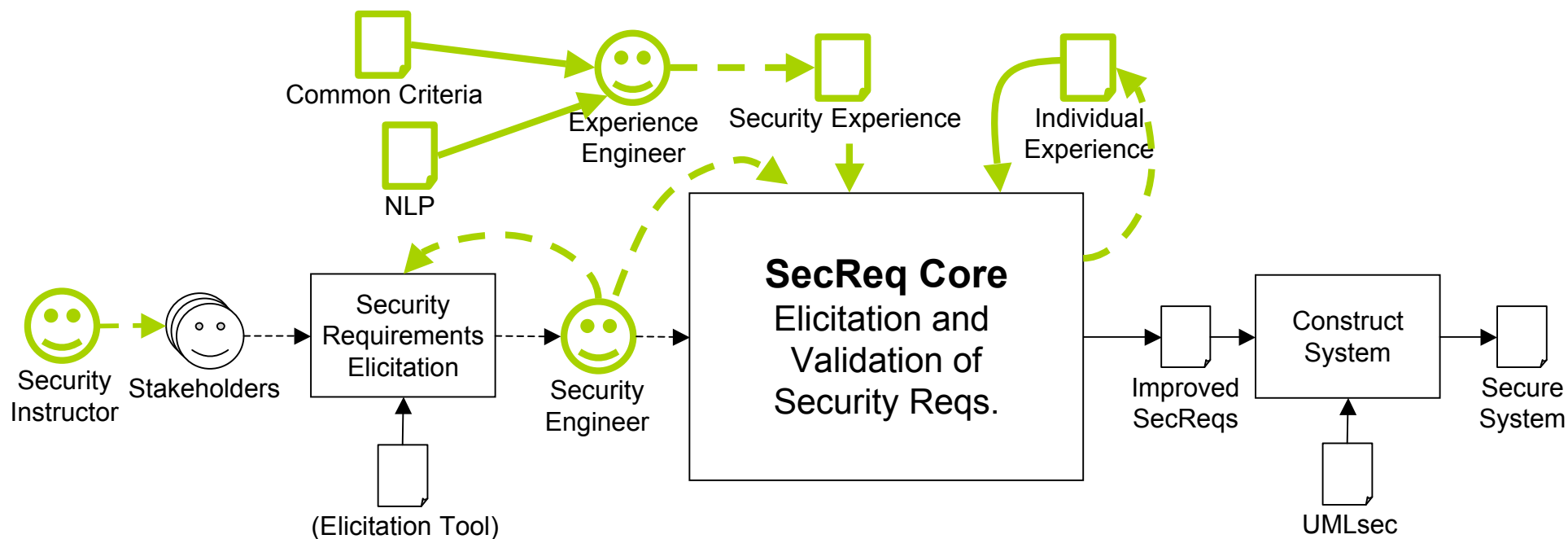
Related

- **Data/information flow focus**
- **Persistent storage modeled**
- **Dependencies via data only**
- **Context diagrams**

Different

- **Stores do not transform**
- **Data dictionary rules vs. FLOW**
- **No intuitive symbols**
- **No concept of fluid**
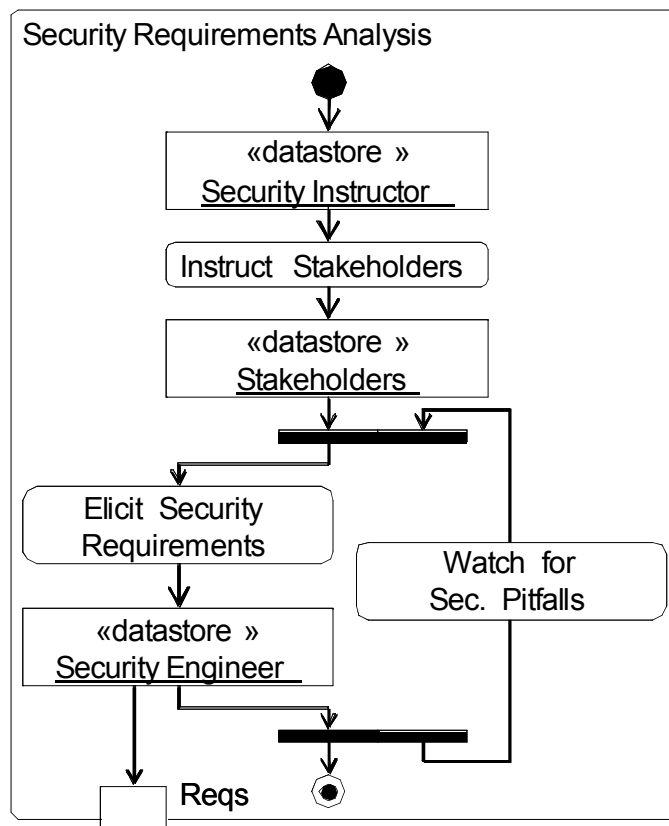
UML Activity Diagrams: Control flow
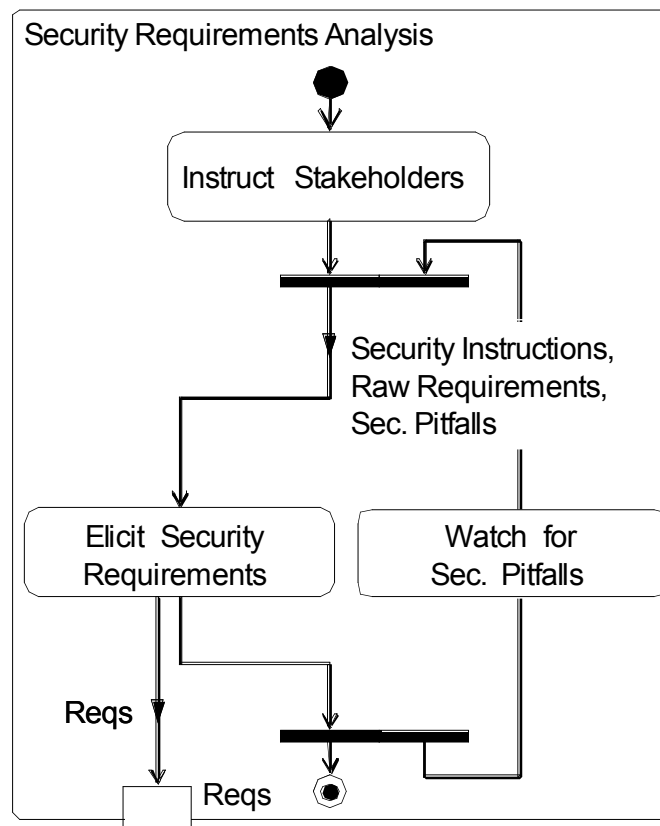
Related

- **Storage can be modeled**

Different

- **Synchronisation of control**
  - **No intuitive symbols**
    - **Nothing fluid**
    - **No data flow**

With «datastore » Objects

Security Requirements Analysis

«datastore »
Security Instructor

Instruct Stakeholders

«datastore »
Stakeholders

Elicit Security
Requirements

Watch for
Sec. Pitfalls

«datastore »
Security Engineer

Reqs

With InformationFlows

Security Requirements Analysis

Instruct Stakeholders

Security Instructions,
Raw Requirements,
Sec. Pitfalls

Elicit Security
Requirements

Watch for
Sec. Pitfalls

Reqs

Reqs

**Stereotypes
can do almost
everything -
even emulate
FLOW**

*Kurt Schneider: **How FLOW can be used to discuss SecReq***

Related

- **Dependencies modeled**
- **Information represented**

Different

- **Complex, detailed notation**
- **Flows only implicit**
- **Fluid not defined**

SecReq Analysis

agent: Security Instructor
↑Instructions:Experience

Instruct Stakeholders

SecReq Elicit.

agent: Stakeholders
↑rawReqs:Requirements

agent: Security Engineer
↓ rawReqs:Requirements
↑ secExp:Experience

Tell Reqs

Watch for Sec. Pitfalls
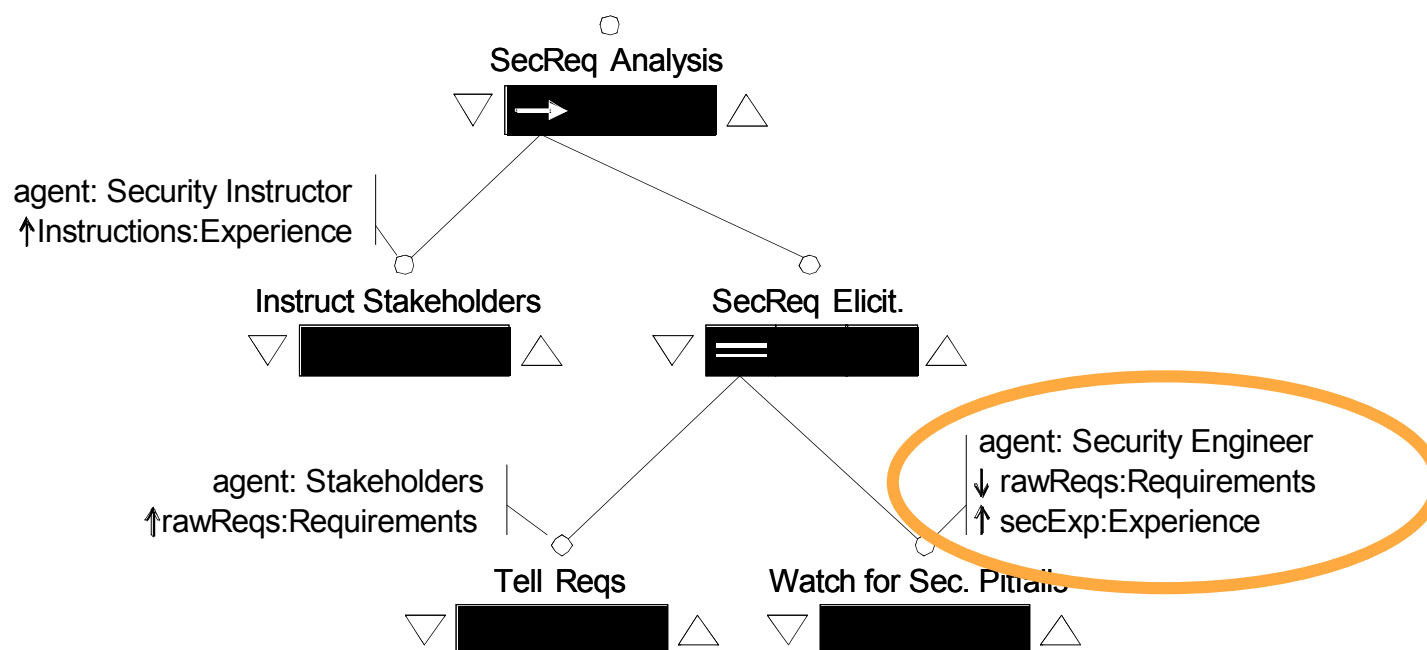
Related

- **Information flow**
- **Including fluid/informal flows**

Different

- **Automated collection**
- **Description, not presription**
- **Observing, not designing flows**
- **Proposal: not yet applied (2006)**

2008-06-20
Phone call (x1)

Tim

2008-06-25
Meeting (x2)

Tina

John

2008-06-21
E-mail (x1)

Christine

2008-06-25
Meeting (x2)
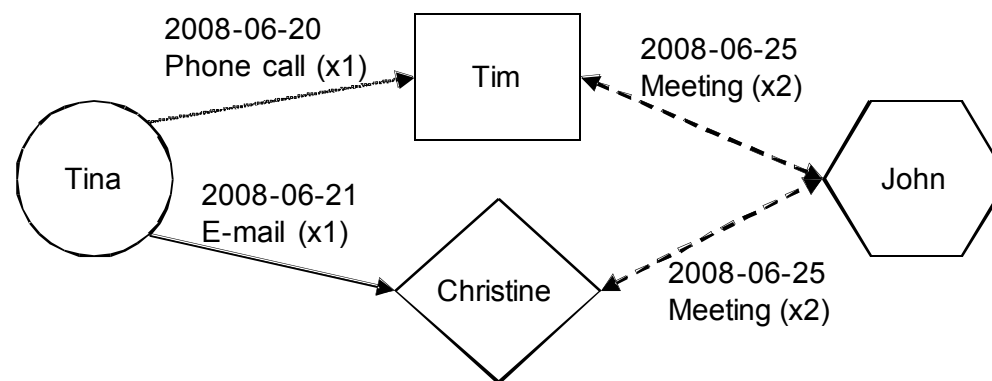
Legend:
Tina is a Security Instructor
Tim and Christine are Stakeholders
John is a Security Engineer

Summary of related notations

- **We do not rank notations!**

- **We try to express our key concepts**
  - *Information flow*
  - *Experience*
  - *Fluid information*

- **Surprise: Often difficult to express in "related notation"**
  Different purpose → subtle differences → hard to express concepts

- **Conclusion: It is worth-while considering FLOW**

Leibniz
Universität
Hannover

Conclusions

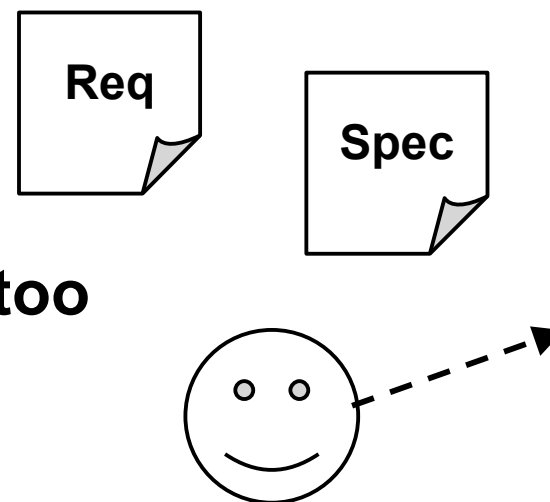- **Let's face it: Not all requirements are documented**

- **That's fine!**

Req

Spec

- **If f we model fluid representations, too**

- **Relax: No formal notation required**

- **Modeling is a crucial learning process**

We propose: FLOW!

- **It requires good (=simple) visualizations**

Modeling purpose and applications

Awareness and Overview

Improving Requirements Processes and Practices

Defining and Tailoring Communication

see: Software Engineering (SE'07)