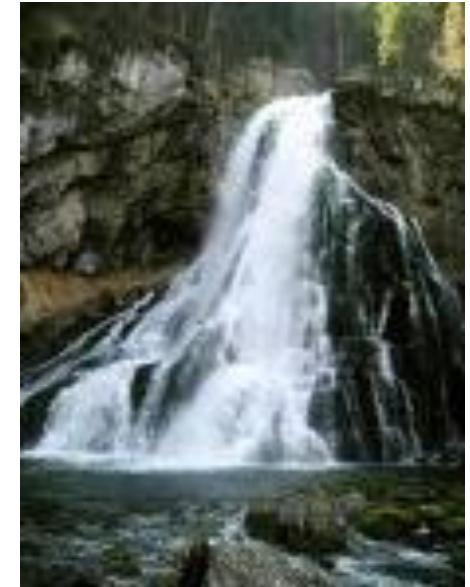


## **Die Bedeutung schriftlicher Dokumentation im Agilen Requirements Management**

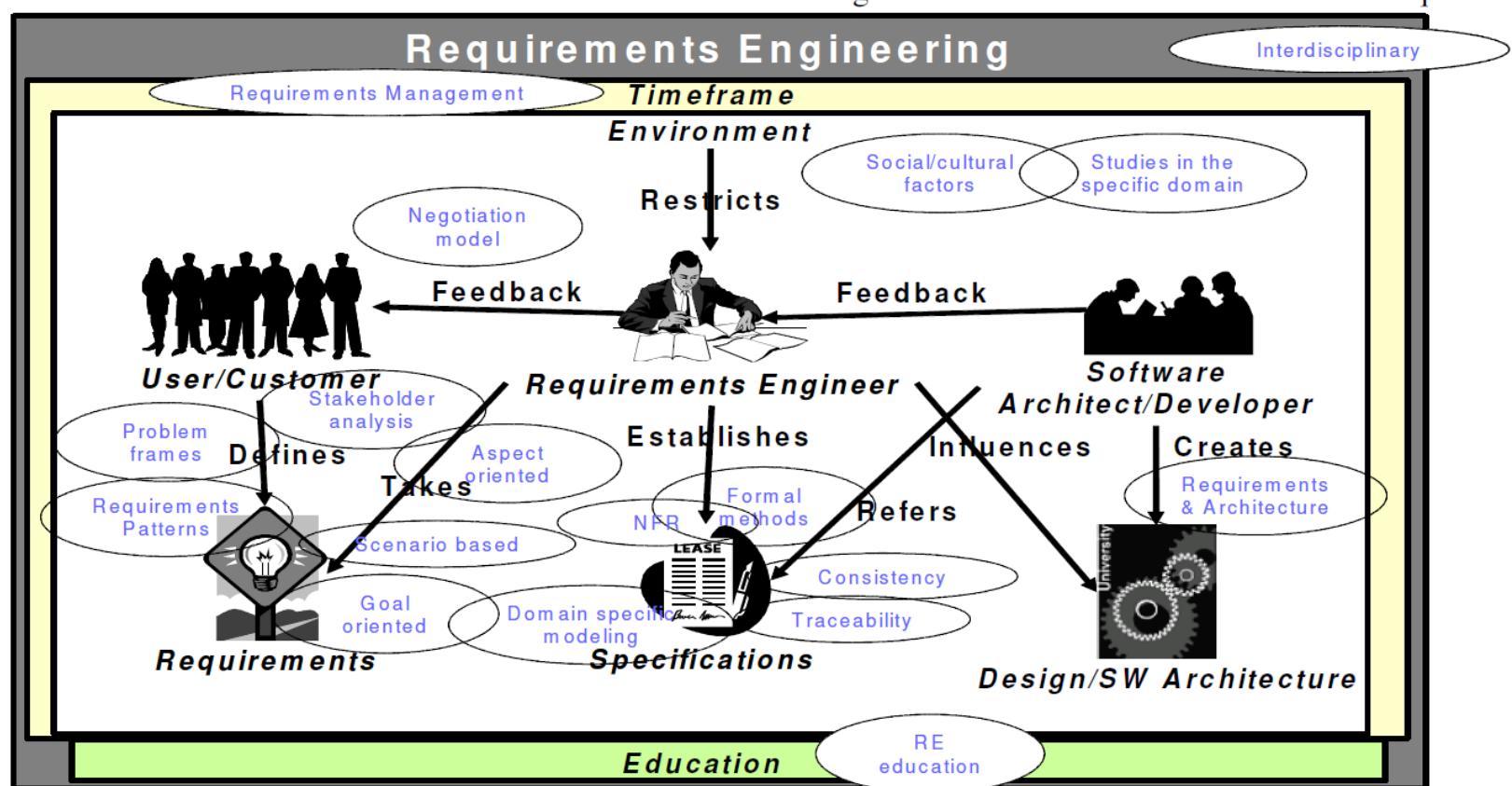
**Dr. Wolfgang Göbl**  
**Raiffeisen Solution**

# Requirements Management im Wasserfall

- Requirements Management fokussiert auf die Erstellung und Verwaltung einer funktional vollständigen, fehlerfreien Anforderungsdokumentation.
- Das Gebiet hat sowohl in der Forschung als auch der Praxis weit verbreitete Methoden (Use Cases, Objektmodelle, Szenarien,...) hervorgebracht.
- Der Analytiker spielt eine zentrale gestaltende Rolle in der Transformation unstrukturierter Anforderungen in strukturierte Spezifikationen.

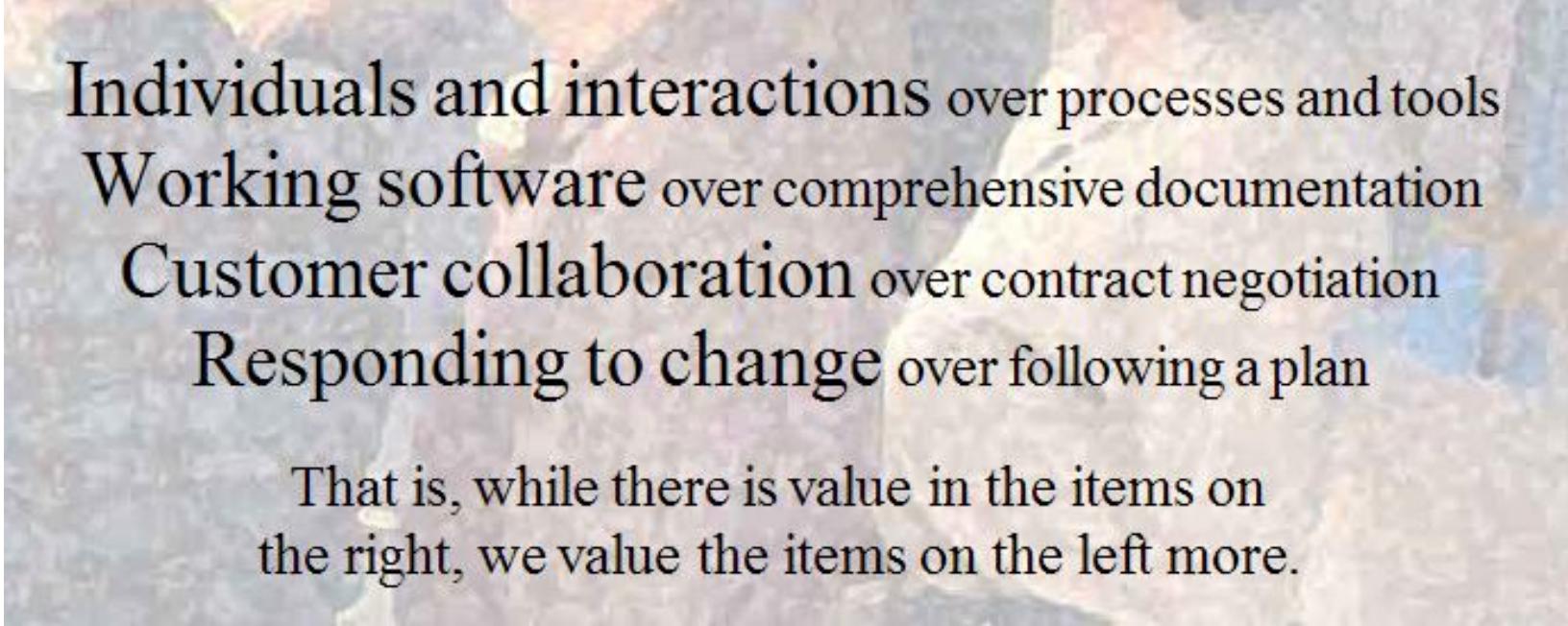


# Requirements Management im Wasserfall



Kamat et.Al „Figure Out the Current Software Requirements Engineering“,  
 14th Asia-Pacific Software Engineering Conference, 2007

## Agiles Manifest

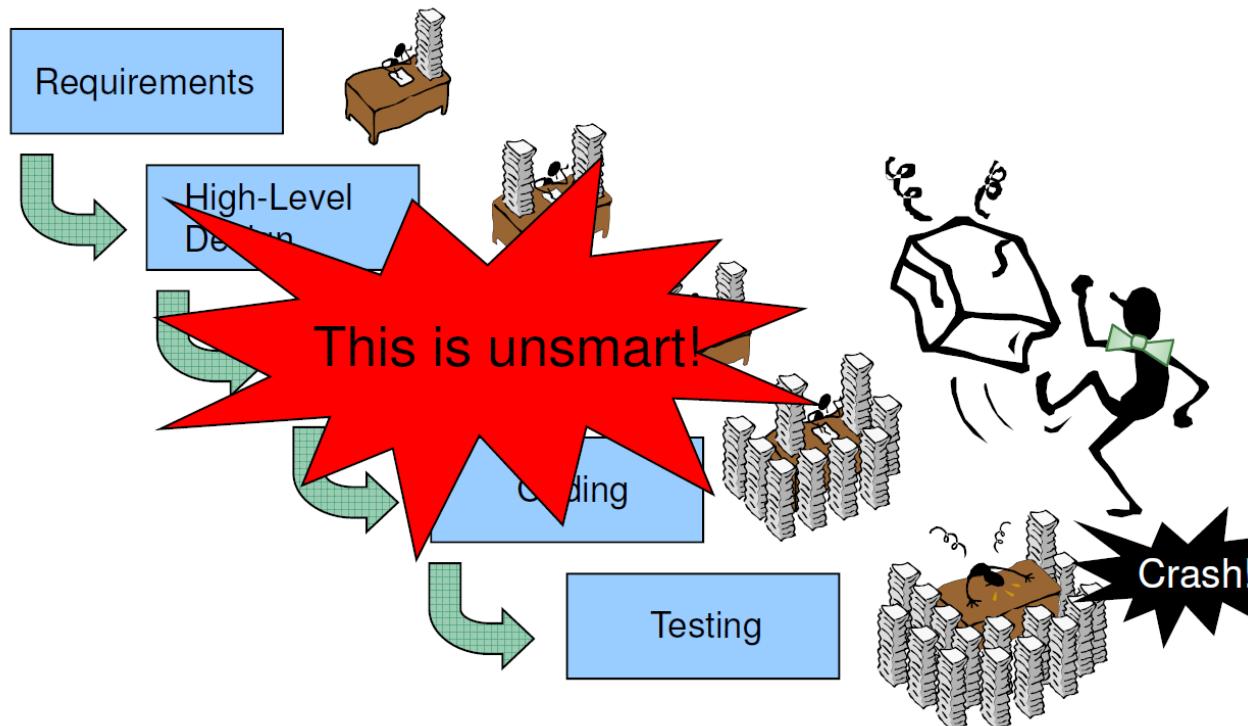


**Individuals and interactions** over processes and tools  
**Working software** over comprehensive documentation  
**Customer collaboration** over contract negotiation  
**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

*<http://agilemanifesto.org/>*

# Wasserfall ist „unsmart“



© 2008 Ivar Jacobson International

I. Jacobson, „Be Smart“, ReConf 2009

## Ist schriftliche Dokumentation im agilen Umfeld „unsmart“?



# Mündliche vs. schriftliche Kommunikation



- Rasch
- Informell
- Direkt

- Unverbindlich
- Wird vergessen
- Keine Modellbildung



- Verbindlich
- Nachvollziehbar
- Beständig
- Modellbildung

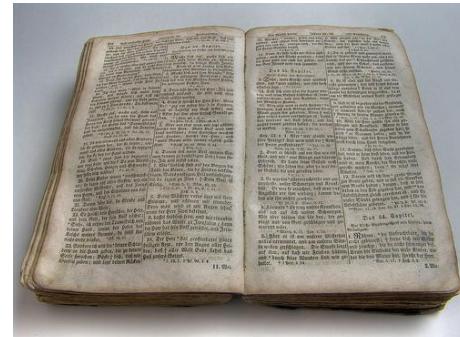
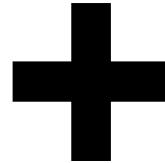
- Träge
- Aufwändig
- Indirekt

## Mündliche vs. Schriftliche Kommunikation

→ Erst wenig (substanzIELLE) Literatur in der Requirements Community

z.B. E. Joyeaux, „Agile Requirements(Engineering)“, ReConf 2010

Erfahrung Raiffeisen Solution: die richtige Mischung macht's



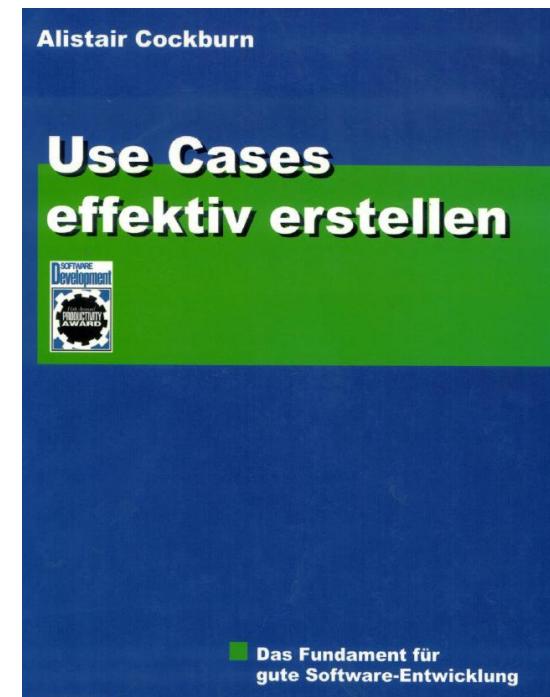
## Requirements Management im agilen Umfeld

- Agile Vorgehensmodelle ändern das Feld des Requirements Management radikal –
- die schriftliche Spezifikation tritt in den Hintergrund, möglichst direkte mündliche Kommunikation zwischen Fachbereich („Product Owner“) und Programmierern wird zu einem zentralen Element.

# Use Cases

„Ein Use Case ist ein

- in sich abgeschlossener Funktionsblock
- der einer Benutzerrolle hilft,
- **genau ein vollständiges Benutzerziel zu erreichen.“**



# Use Cases (i)

## Use Case 22 Einen Verlustfall Registrieren

**Umfang:** »System« meint das Computersystem, das Forderungen erfasst .

**Ebene:** Blau (Anwenderziel) 

**Version:** 2

**Status:** Geprüft

**Revision:** Läuft

**Anwendungskontext:** Verlustfall voll

**Primärakteur:** Angestellter

**Vorbedingungen:** Angestellter hat si

**Trigger:** Angestellter hat mit der Ein

**Nachbedingung:** Informationen zur

**Invariante:** Es geschieht nichts.

**Stakeholder und Interessen:** Wie zu

**Standardablauf:**

Um die Arbeit des Angestellten zu b  
ihm arbeiten, sobald die erforderlichen Daten erfasst sind. Der Angestellte kann die  
Daten in beliebiger Reihenfolge eingeben, um den momentanen Erfordernissen  
nachzukommen. Die folgende Sequenz gilt als die wahrscheinlichste.

- i. Angestellter gibt Policennummer des Versicherten ein oder alternativ Namen und Datum des Vorfalls. System greift auf verfügbare Informationen zur Police zu und zeigt an, dass die Forderung mit der Police abgeglichen wird.



id. Angestellter will Verlustfall neu öffnen, der unterbrochen oder gespeichert wurde oder vervollständigt werden muss.

idr. Angestellter findet Verlustfall mit Hilfe der Suchdetails von »Verlustfall«.

oderungslimit, ohne dass Daten des Forderungslimits relevant sind:  
als an, die für die Festlegung des  
ller abweichendes Forderungsl-

derungslimit und es sind Daten in  
legung des Forderungslimits rele-

ind und fordert Angestellten auf,  
mit neuem Forderungslimit fort-

: System fährt mit dem Verlustfall

legung des Forderungslimits: Sys-  
derungslimit relevante Daten (alle  
bleiben gespeichert).

erung:  
Forderungsduplikate aus der Ver-  
ste aus und untersucht sie. Dieser  
n.

2c2a. Angestellter stellt fest, dass Forderung ein Duplikat ist:

Angestellter öffnet Forderungsduplikat aus der Liste der Forderungen zur  
Bearbeitung, wenn sie noch nicht als abgeschlossen ausgezeichnet wurde  
(auf der Grundlage der Zugriffsberechtigung des Angestellten). Angestell-  
ter darf alle Daten aus vorher gespeicherter Datei löschen.

2c2b. Angestellter stellt fest, dass Forderung kein Duplikat ist: Er kehrt  
zum Verlustfall zurück und beendet ihn.

2d. Vorherige Informationen zum Verlustfall werden verändert, nachdem die erste  
Prüfung über Forderungsduplikate durchgeführt wurde:

2dr. System führt erneut Prüfung über Forderungsduplikate durch.

## Use Cases vs. User Stories

*„A Use Case is a way of describing requirements.  
A User Story is a way of prioritizing work.“*

*„A story is a promise to have a conversation  
A use case is the record of the conversation. If you think you need one.“*

<http://alistair.cockburn.us/Blog>

# Requirements Management im agilen Umfeld

- Verfolgt man z.B. die vielen Diskussionen der immer größer werdenden Scrum-Community, kann man leicht den Eindruck gewinnen, dass die Rolle des Analytikers obsolet wird und schriftliche Dokumentation eher ein lästiges Beiwerk als ein wertvolles Asset ist.



# Schriftliche Dokumentation



Chris Rupp

General Manager  
SOPHIST GmbH

Irgendeine Art der Dokumentation des Gelernten ist wichtig, denn durch Dokumentation kann Wissen asynchron mitgeteilt werden. Nicht immer sind alle Teammitglieder im War-Room vertreten oder nehmen an den täglichen Stand-Up-Meetings teil, denn es gibt auch verteilt arbeitende Projekte. Auch müssen - bei vorhandener Dokumentation - während der nächsten Neuentwicklung in fünf Jahren nicht alle Businessprozesse und -Artefakte wieder von Neuem ermittelt werden. Dafür muss nicht zwingend eine dicke Prosaspezifikation entstehen. Wie wäre es mit ein paar BPNM- oder UML-Klassendiagrammen, die das Scrum-Team pflegt?

# Schriftliche Dokumentation

- Strukturierung der Anforderungen (Use Cases)
- Vermeidung von Missverständnissen
- Basis für Benutzerhandbuch
- Aufteilung auf mehrere Analytiker

<u>UC 057000 AM SYSTEM ANMELDEN/STARTSEITE ANZEIGEN (GWG, LUN) .....</u>	13
<u>UC 057001 KONTO ANLEGEN .....</u>	18
<u>UC 057002 KONTO ÄNDERN .....</u>	26
<u>UC 057003 KLIENT ZU KONTO ZUORDNEN .....</u>	30
<u>UC 057004 KONTO SUCHEN/ANZEIGEN (OEL, GWG: FINAL BIS AUF SUCHLISTE) .....</u>	45
<u>UC 057006 ZAHLSCHEIN DRUCKEN (GWG) .....</u>	60
<u>UC 057007 KONTOSCHEINABSCHLUß DURCHFÜHREN (OEL) .....</u>	65
<u>UC 057008 KONTO LÖSCHEIN .....</u>	69
<u>UC 057009 KONTOAUSWERTUNG ANZEIGEN (GWG) .....</u>	70
<u>UC 057010 KLIENTEN VERWALTEN .....</u>	74
<u>UC 057011 KONTOINFORMATION DRUCKEN .....</u>	80
<u>UC 057030 PASSWORT ÄNDERN (LUN) .....</u>	81
<u>UC 057031 BENUTZEROPTIONEN ÄNDERN .....</u>	84
<u>057032 BENUTZER UMMELDEN (LUN) .....</u>	85
<u>UC 057050 AM SYSTEM ANMELDEN-NOTARBANK APPLIKATION (LUN, GWG) .....</u>	94

# Beispiel Use Case

## 8 UC 057006 Zahlschein drucken

### 8.1 Kurzbeschreibung

Der Benutzer kann einen SEPA Zahlschein für ein bestimmtes Kontos ausdrucken.

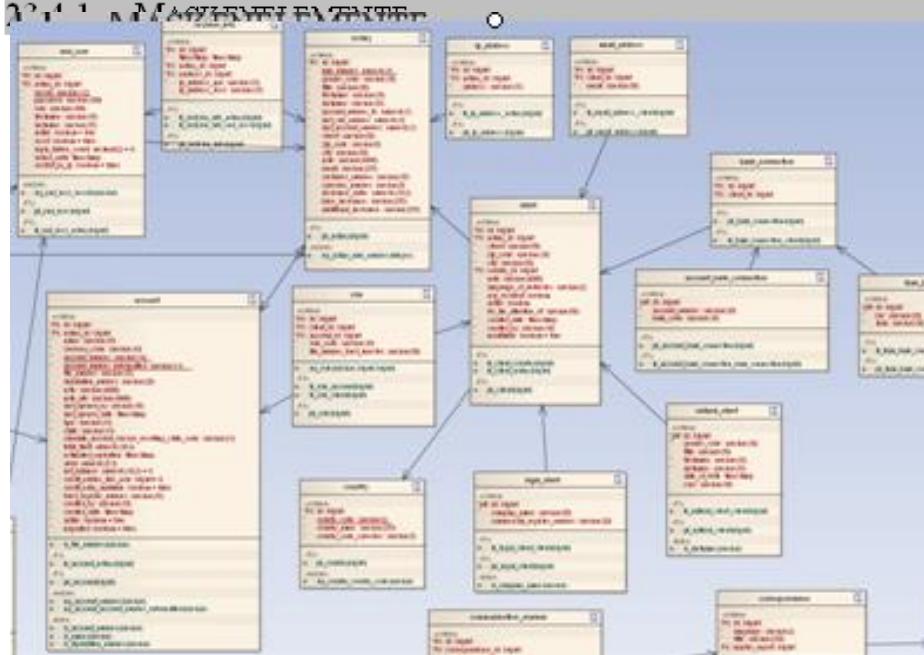
### 8.2 Standardablauf

1. Benutzer öffnet die Detailansicht eines Kontos → Maske UC057004 – M3 Kontodetailansicht (Beispiel angelehnt an Seite 51)
2. Benutzer wählt das Icon „Zahlschein drucken“  beim Konto
3. System öffnet →Maske UC 057006-M1 Zahlscheindaten erfassen, Seite 61 und übernimmt dabei die für den Zahlscheindruck relevanten Daten
  - Empfängername
  - Empfänger IBAN/BIC
  - Verwendungszweck
4. Benutzer hat optional die Möglichkeit die Daten zu editieren und Auftraggeberdaten zu ergänzen
5. Benutzer wählt „Zahlschein drucken“
6. System druckt → Druckstück Sepa Zahlschein, Seite 59

# Schriftliche Dokumentation

## 23 UC057020 NOTARIATE VERWALTEN (MANAGE NOTARY'S OFFICE).....105

<b>23.1</b>	<b>SHORT DESCRIPTION.....</b>	<b>105</b>
<b>23.2</b>	<b>CREATE NOTARY'S OFFICE .....</b>	<b>105</b>
<b>23.2.1</b>	<b>MASKE UC057020 – M1 NOTARIAT ANLEGEN .....</b>	<b>106</b>
<b>23.3</b>	<b>NOTARIAT SUCHEN/ANZEIGEN (SEARCH/DISPLAY NOTARIES OFFICE) .....</b>	<b>111</b>
<b>23.3.1</b>	<b>MASKE UC057020 – M2 NOTARIAT ANZEIGEN/ÄNDERN .....</b>	<b>112</b>
<b>23.4</b>	<b>NOTARIAT ÄNDERN (MODIFY NOTARIES OFFICE) .....</b>	<b>113</b>
<b>23.4.1</b>	<b>MASKE UC057020 – M3 NOTARIAT ÄNDERN .....</b>	<b>113</b>



Menü KV Huber / Maier x

Zahlschein drucken

Empfänger	Notariat Tades	Auftraggeber	Max Mustermann
Empfänger IBAN	ATNG32001234567890	Auftraggeber IBAN	ATNG32001234567890
Empfänger BIC	ATXYZZZE	Auftraggeber BIC	ATWERMNOEME
Betrag	300.000,00 EUR		
Verwendungszweck	KV Huber/Maier		

Zahlschein drucken Fertig

## Schriftliche Dokumentation

Die Kunst ist zum richtigen Zeitpunkt die richtige Anzahl an Details zu beschreiben.

- 1. Use Case Name
- 2. Kurzbeschreibung
- 3a. Maskenskizzen und Datenattribute
- 3b. Geschäftsregeln
- 4. Iteratives Verfeinern

## Beispiel Use Case

### 1 UC 057006 Geld beheben

#### 1.1 Kurzbeschreibung

Der Bankkunde kann mit seiner Bank-Karte beim Bankomat Geld beheben

#### 1.2 Standardablauf

1. Benutzer führt Karte in den Schlitz ein
2. System öffnet →Maske UC 057006A PIN eingeben
3. Benutzer gibt PIN ein.
4. Benutzer wählt „Geld beheben“
5. System öffnet →Maske UC 057006B Geld beheben
6. Benutzer wählt einen vorgeschlagenen Betrag aus oder gibt einen Betrag ein.
7. System gibt den gewünschten Betrag aus.

## Warum Use Cases verwenden

1. Die Use Case Namen bilden ein „Raster“ für die Gliederung des Systems
2. Der Standardablauf ermöglicht es allen Beteiligten auf einen Blick zu erkennen was das System tut (und was nicht)
3. Dokumentation für Software Wartung (Near-Shoring) erforderlich.

*Mehr Gründe unter:*

<http://alistair.cockburn.us/Why+I+still+use+use+cases>

## Schnorcheln vs. Tauchen

- Zu Beginn muss man sich rasch einen Überblick über den Fachbereich und die Systemgrenzen verschaffen.
- Use Cases sind das optimale Mittel zur Strukturierung
  - ▶ Nur Kurzbeschreibung erfassen



## Schorcheln vs. Tauchen

- Im Sprint muss der Analytiker an den richtigen Stellen tief tauchen
- Use Cases bleiben das optimale Mittel um Überblick zu bewahren werden aber detaillierter beschrieben
  - ▶ Standardabläufe
  - ▶ Masken
  - ▶ Detailattribute
  - ▶ Geschäftsregel



## Schorcheln vs. Tauchen

- Die Kunst dabei ist zu erkennen wann die schriftliche Dokumentation ausreicht und welche Teile man als „Freiheitsgrade“ in der Implementierung lässt



## Erfahrungen RSO

- Requirements Management ändert sich in Scrum Radikal
- Analytiker zu Beginn Treiber der Requirements (Grobanalyse)
- User Stories alleine KEIN ausreichendes Strukturierungsmittel
  - Viel zu feingranular
- Verhältnis User Stories: Use Cases ca. 10:1
- Ab ca. Hälfte des Projekts läuft Detailanalyse direkt über Entwickler.
  - „Nachdokumentation“ durch Analytiker
- Rolle des Product Owners problematisch (Information Bottleneck)
  - Direkten Kontakt mit Benutzern suchen!

## Fazit&Lessons learned

- Agile Methoden lassen bei Entwicklern das Gefühl der „Freiheit“ von den Fesseln der schriftlichen Dokumentation aufkommen.
- Agile Methoden sind aber in Wirklichkeit viel „strenger“ und überwachender als Wasserfall (daily scrum, Task Board)
- Schriftliche Dokumentation unentbehrlich
  - ▶ Die Frage ist nur was wann dokumentiert wird

## Literatur

*Chris Rupp „Eine Streitschrift, eine Provokation“ (SQ Magazine), 2010*

*I. Jacobson, „Be Smart“, ReConf 2009*

*E. Joyeaux, „Agile Requirements(Engineering)“, ReConf 2010*

<http://agilemanifesto.org/>

Alistair Coburn: „Use Cases effektiv erstellen“, ISBN-10: 3826613449, 2003

<http://alistair.cockburn.us>

# Vielen Dank!

***Dr. Wolfgang Göbl***

*Raiffeisen Solution*

*Mooslackengasse 23, 1190 Wien*

*Tel. +43 1 99366 2456*

*wolfgang.goebel@r-solution.at*