Capgemini

CONSULTING.TECHNOLOGY.OUTSOURCING

Software Specifications

TU Darmstadt,
13.11.2015,
Guido Schuh

People matter, results count.

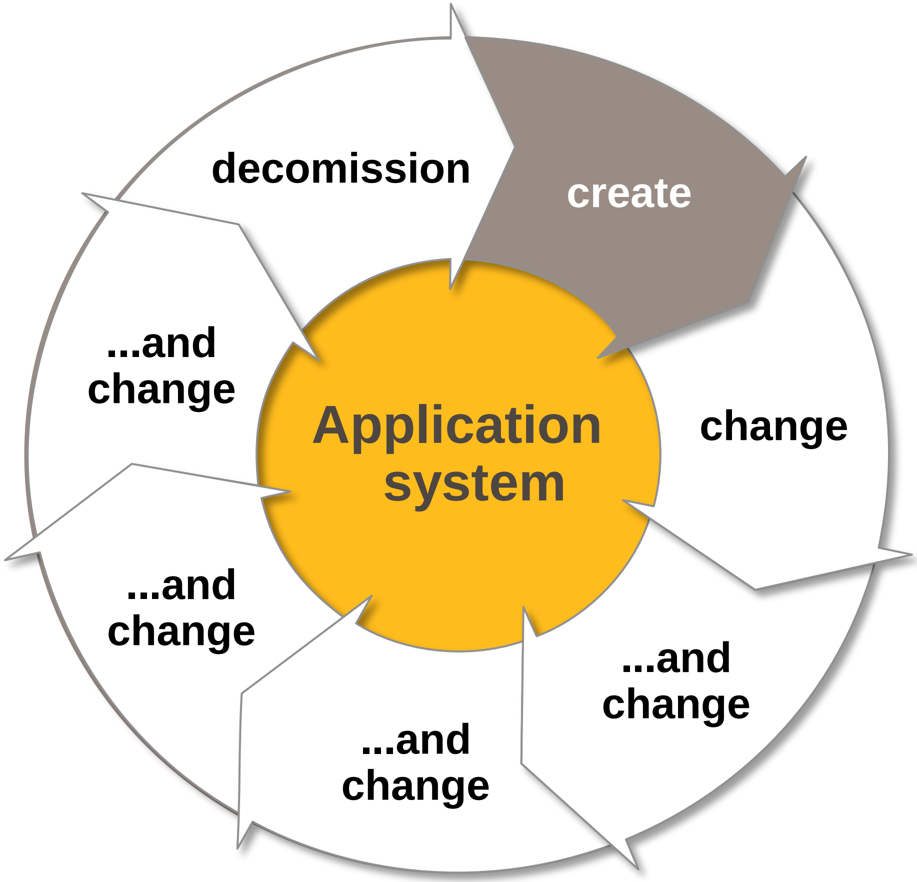
Agenda

- Why do we need software specification?
- Software specification in an agile context
- Software – then and now
- What does software specification look like?

Agenda

- **Why do we need software specification?**
- Software specification in an agile context
- Software – then and now
- What does software specification look like?

Business-critical application systems are operated and maintained for a long time



Specifications enable good system design and maintenance

New systems and changes can be described prior to implementation

- Basis for customer communications
- customer "sees" his system, even before it is built
- Reduces the risk of "building the wrong system"
- Reduces the risk of incomplete changes during maintenance

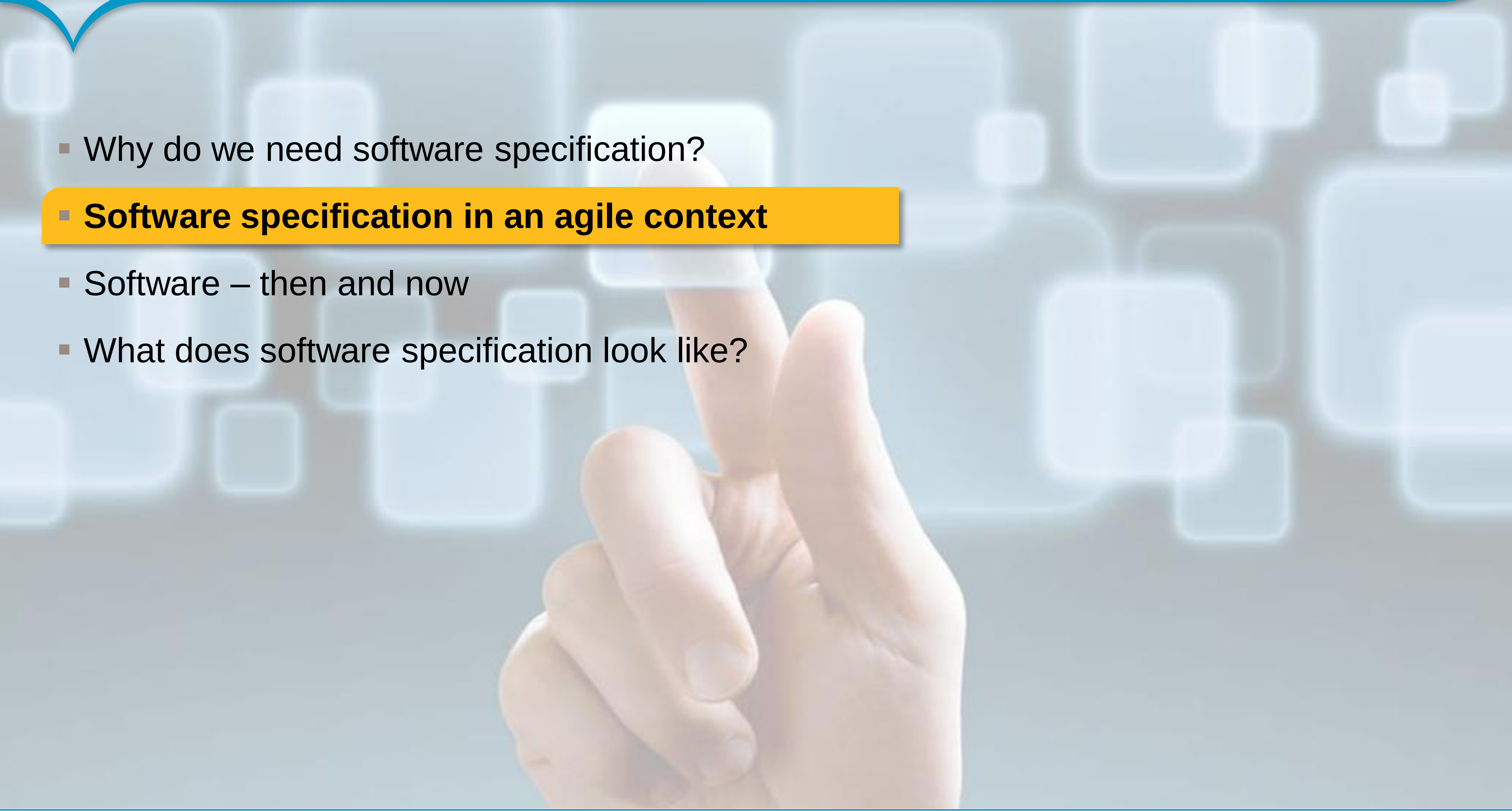
Goals of the software specification

The software specification describes an application system …

Comprehensive enough for customer

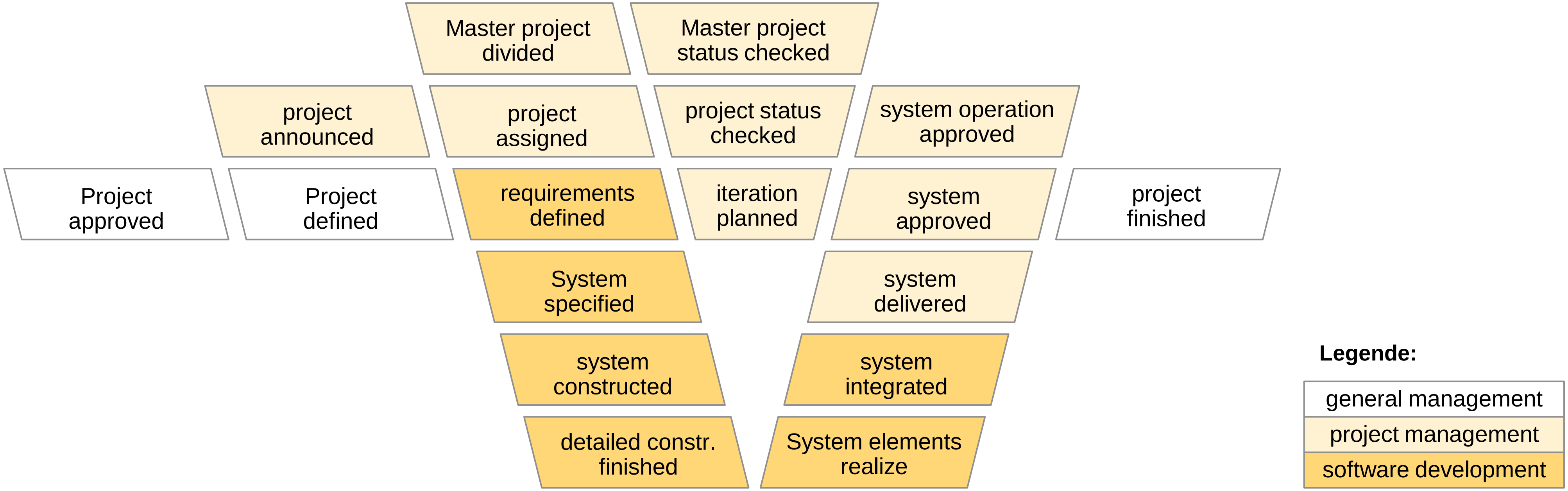
Detailed enough for developer

Agenda



- Why do we need software specification?
- **Software specification in an agile context**
- Software – then and now
- What does software specification look like?

V-Modell XT: for waterfall projects AND iterative projects



Master project divided

Master project status checked

project announced

project assigned

project status checked

system operation approved

Project approved

Project defined

requirements defined

iteration planned

system approved

project finished

System specified

system delivered

system constructed

system integrated

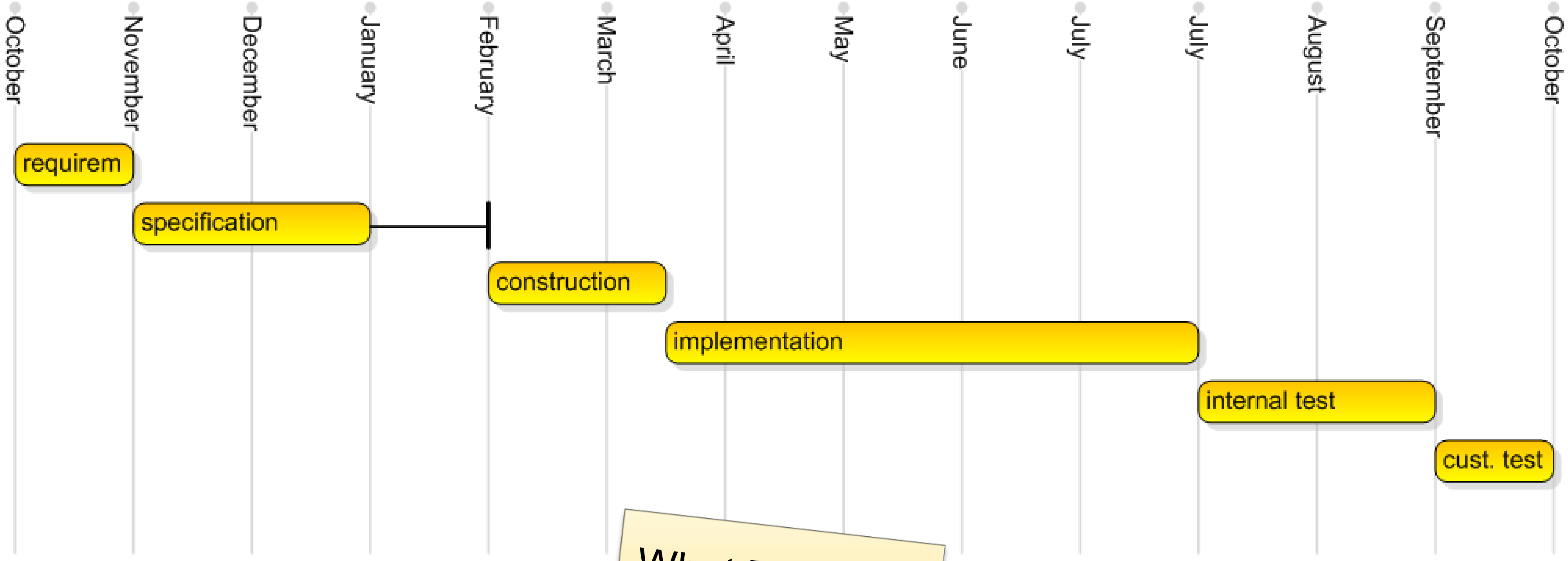
detailed constr. finished

System elements realize

Legende:

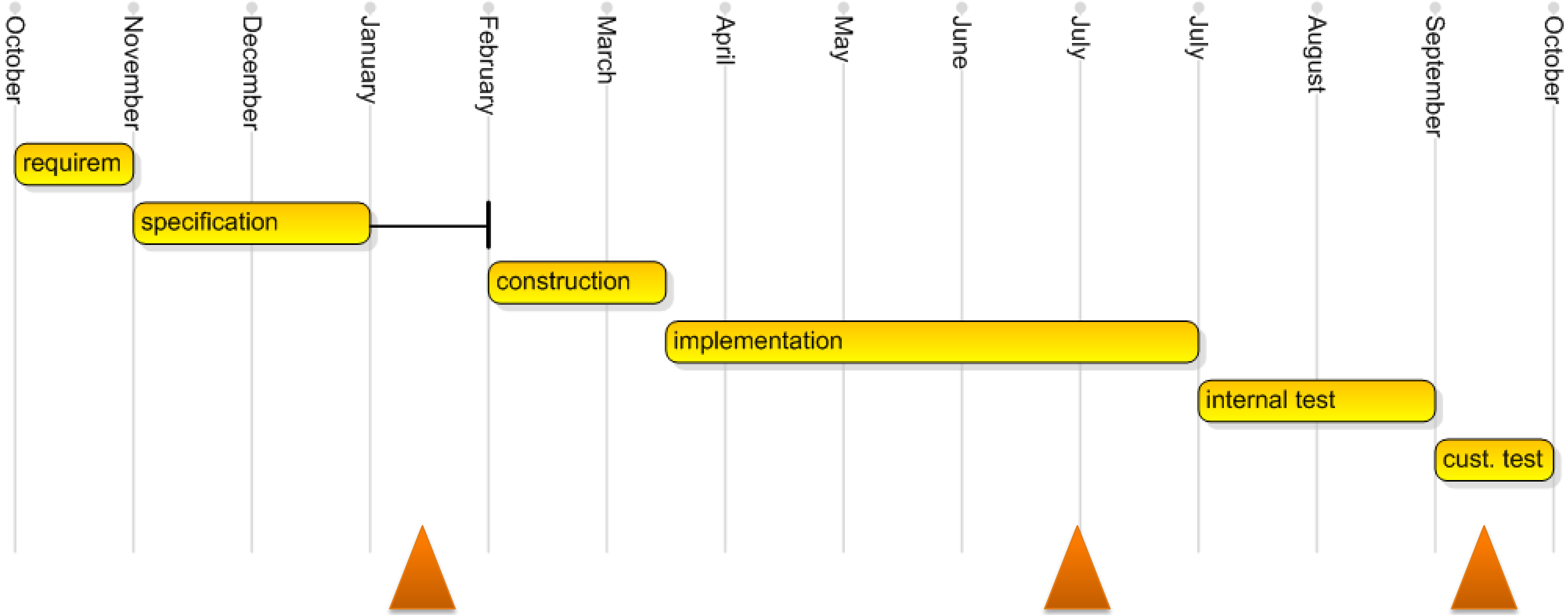
- general management
- project management
- software development

The project plan we dream of in a waterfall model



What Problems could we run into here?

The project plan we dream of in a waterfall model

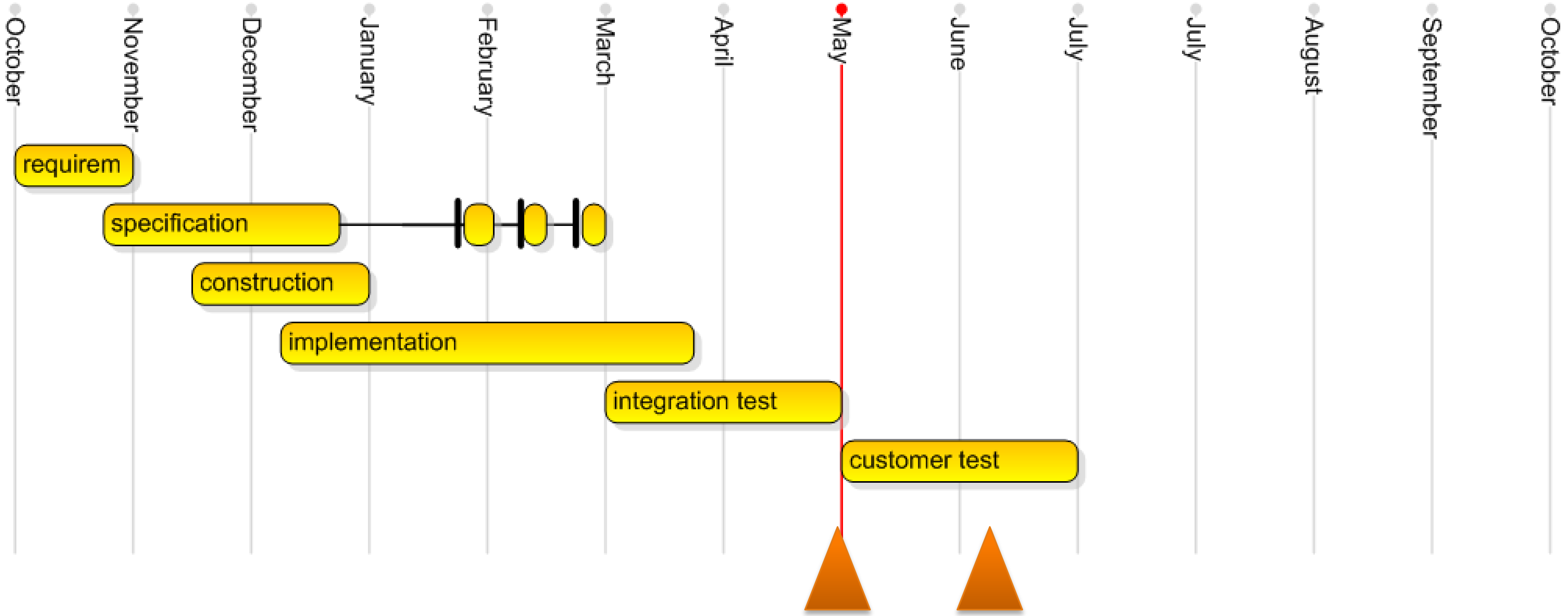


And you can be sure that some requirements change during implementation!

Wake up, we‘re going live at the end of July!!!

And the customer needs more time to test!

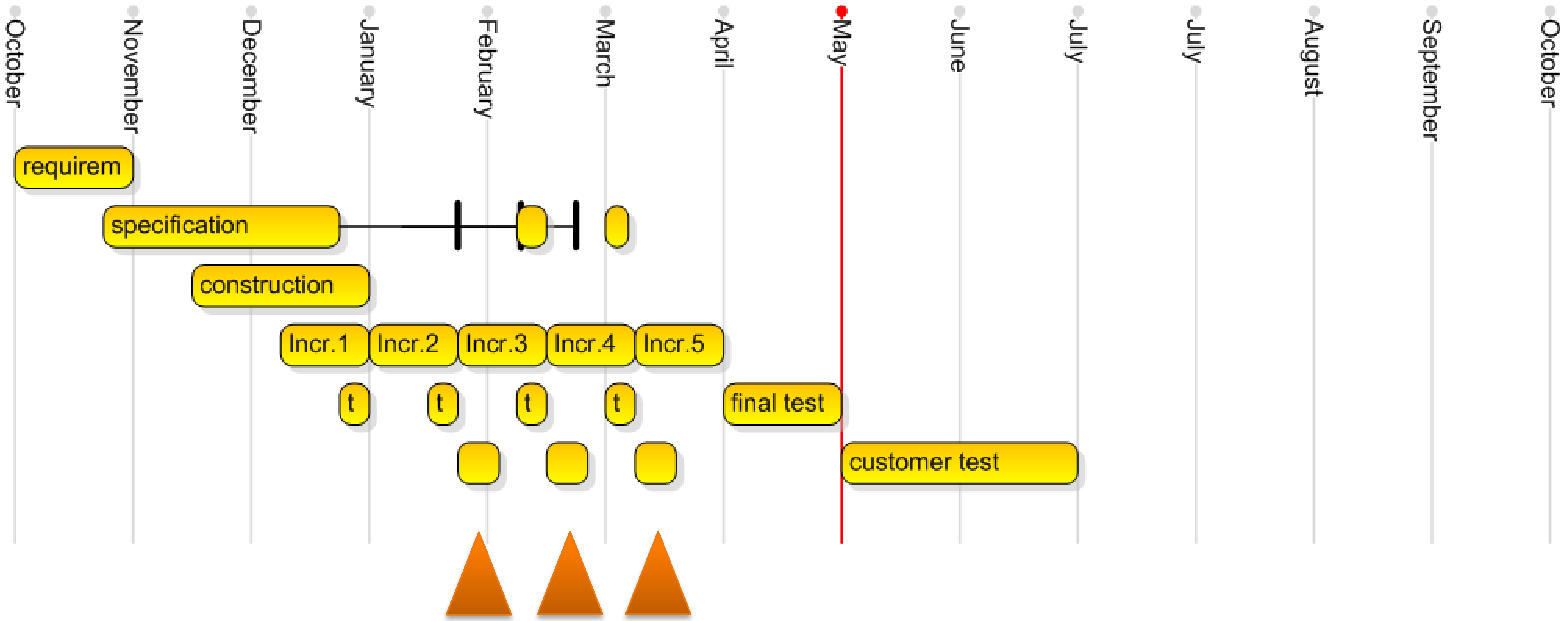
The usual project plan in a waterfall model



Yes! Me made it!
Need vacation NOW!

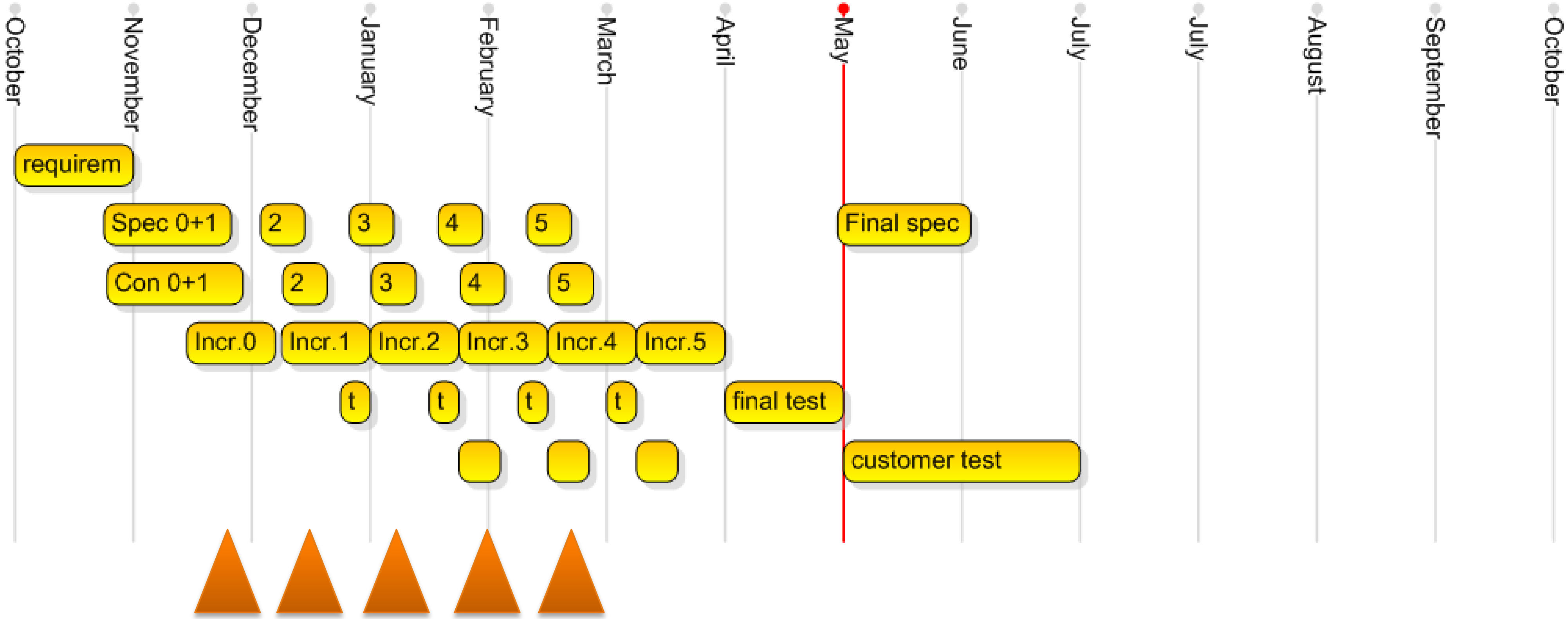
What do you mean,
you don‘t like the UI!?

Incremental implementation



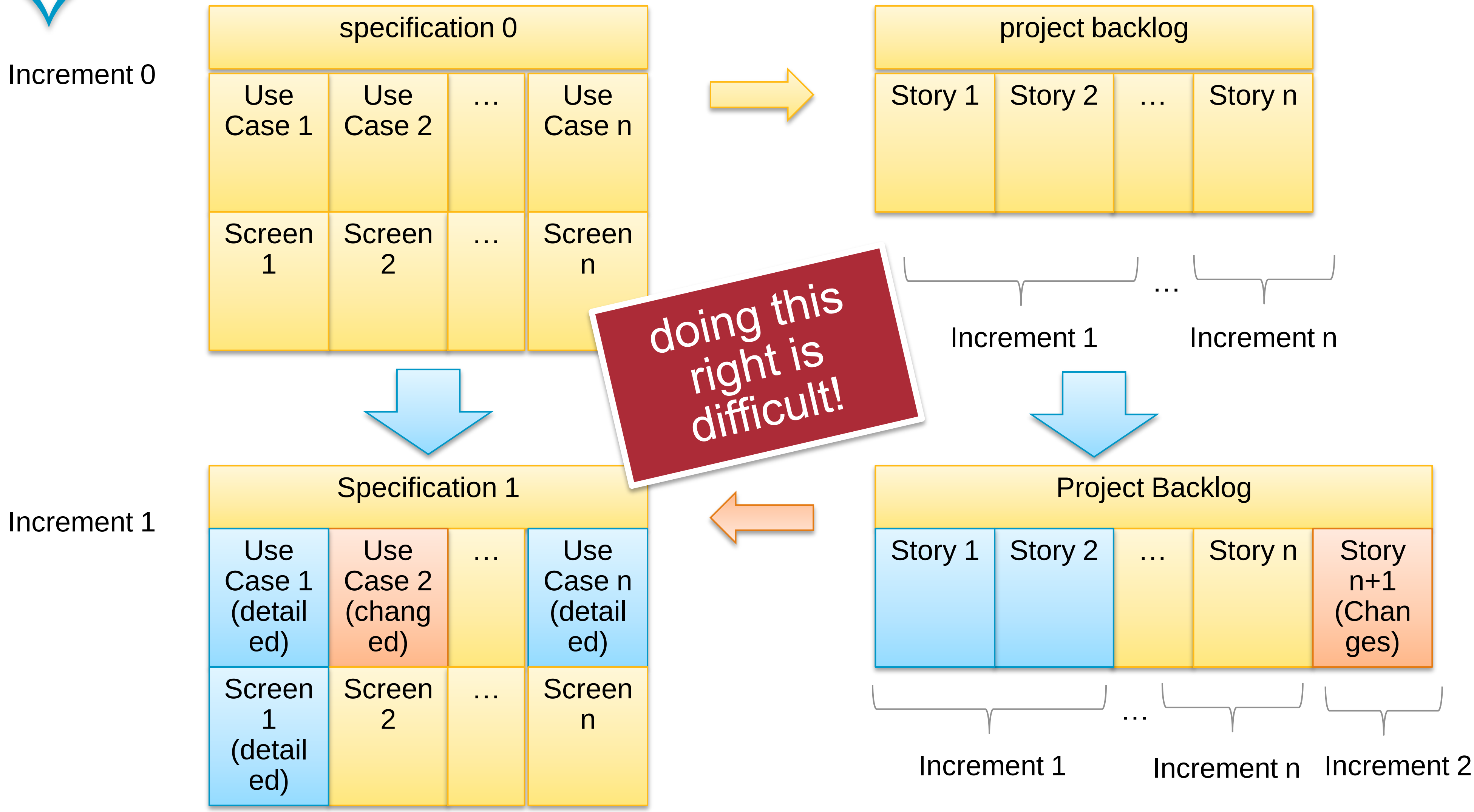
Customer, take a look at what we‘ve got and give us feedback!

Incremental specification, construction and implementation



Customer, take a
look at the current state
of the specificaton

specification vs. project backlog



Increment 0

specification 0

Use Case 1 | Use Case 2 | … | Use Case n

Screen 1 | Screen 2 | … | Screen n

project backlog

Story 1 | Story 2 | … | Story n

Increment 1 … Increment n

doing this right is difficult!

Increment 1

Specification 1

Use Case 1 (detailed) | Use Case 2 (changed) | … | Use Case n (detailed)

Screen 1 (detailed) | Screen 2 | … | Screen n

Project Backlog

Story 1 | Story 2 | … | Story n | Story n+1 (Changes)

Increment 1 … Increment n Increment 2

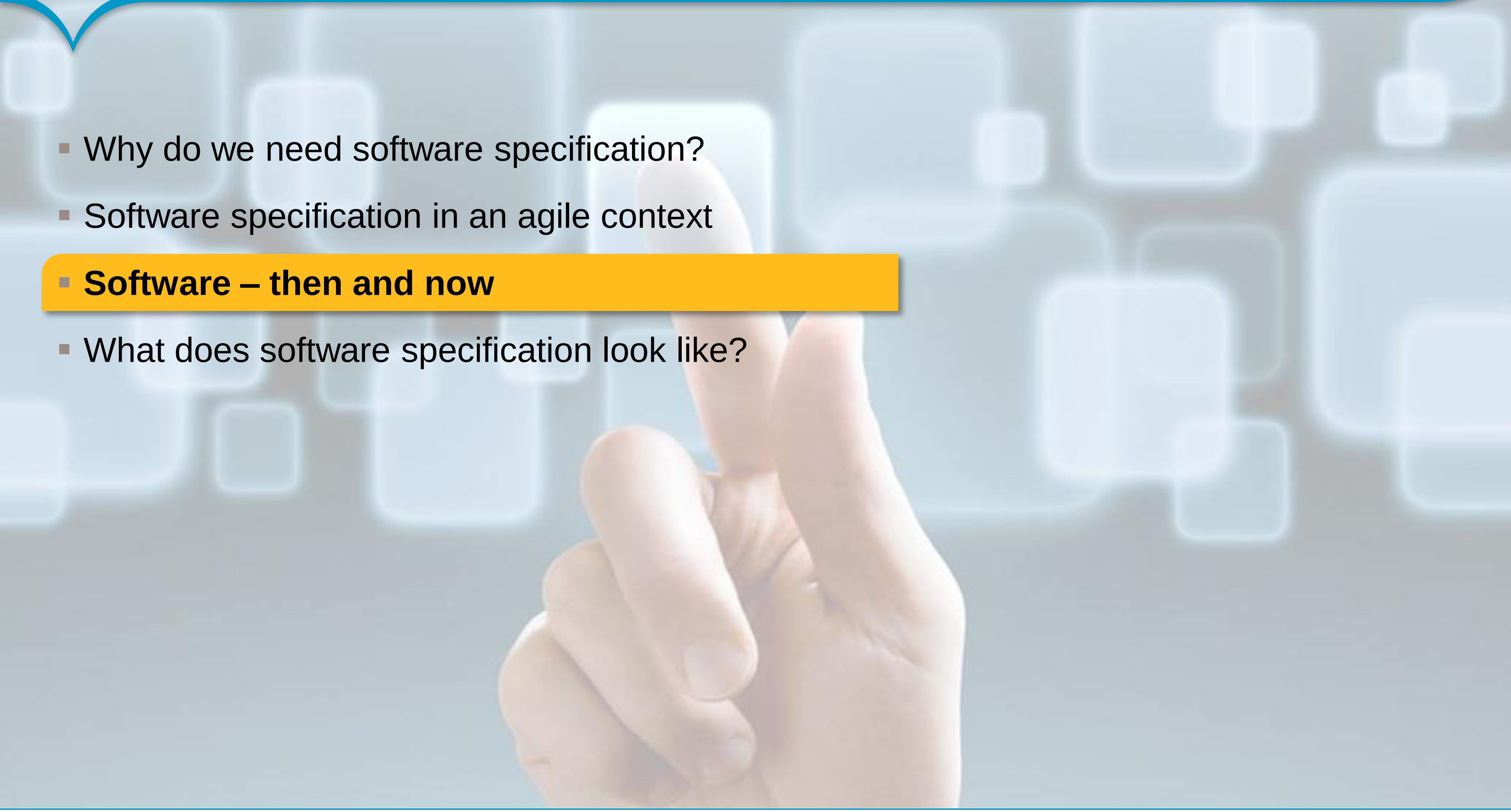
specification vs. project backlog

Typical agile projects don‘t do specification

But we need to enable long-term maintenance

Microsoft Word as a tool feels outdated here!

Agenda

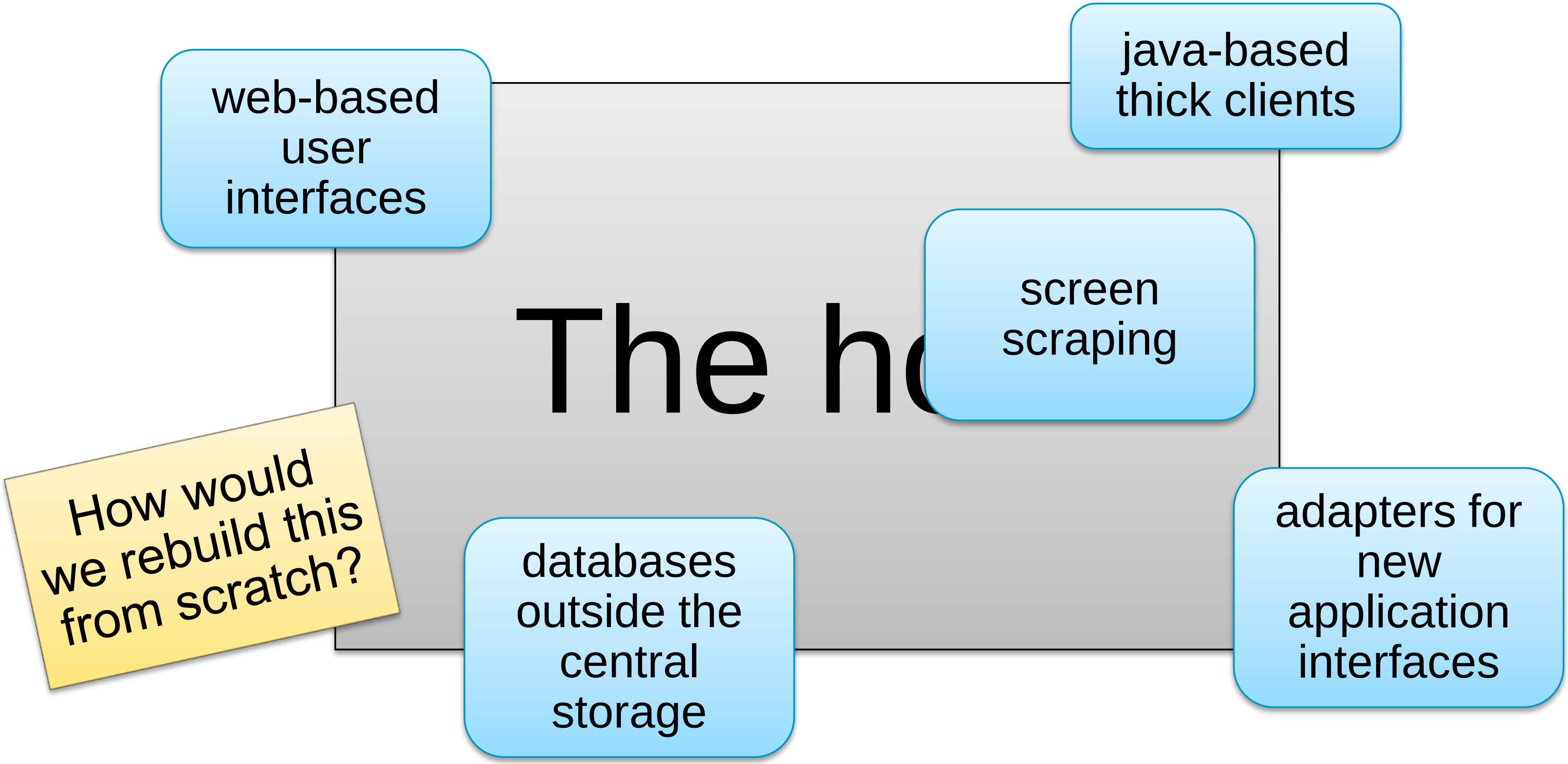


- Why do we need software specification?
- Software specification in an agile context
- **Software – then and now**
- What does software specification look like?

What software looked like in the 80s

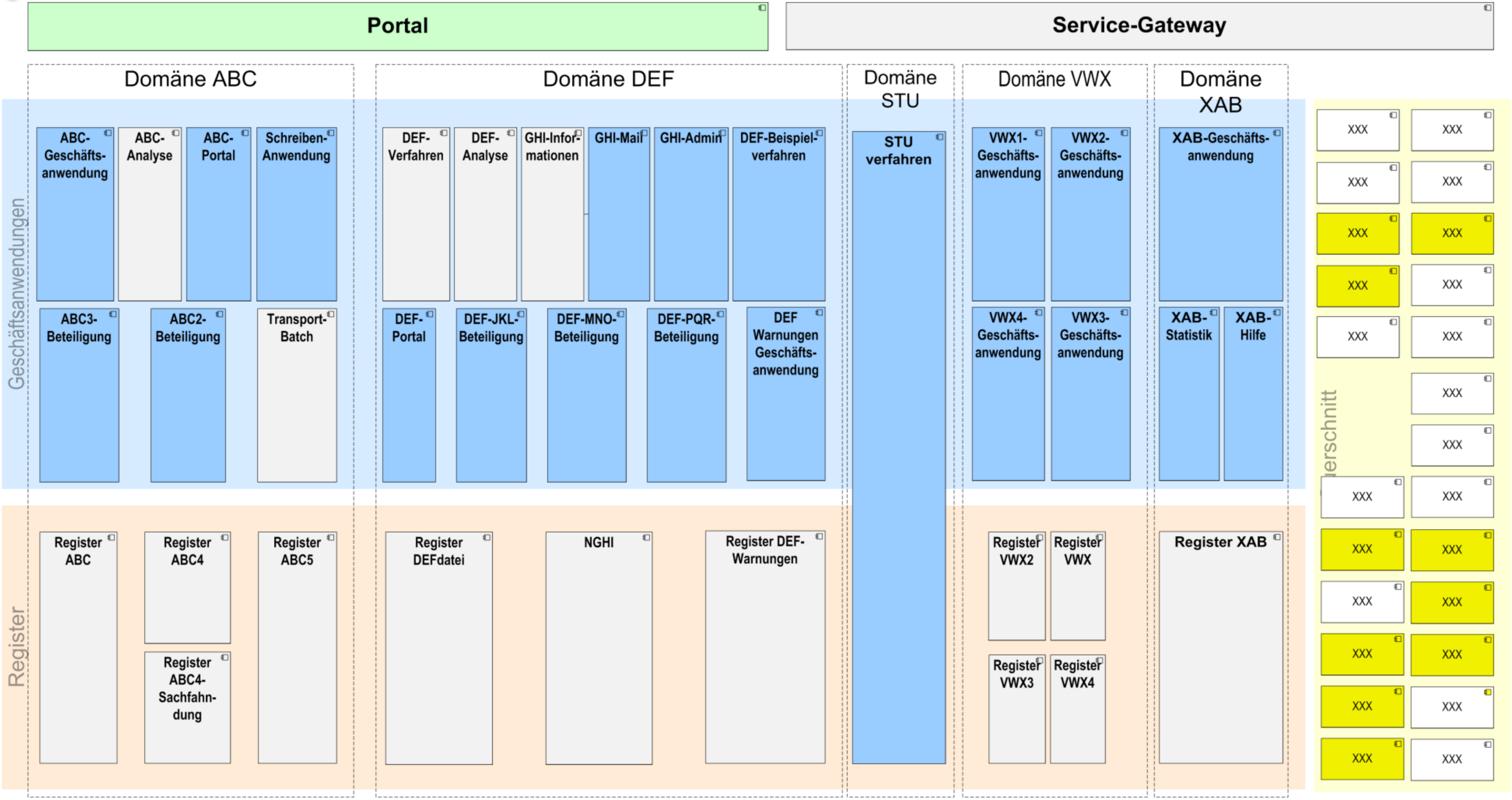
The host

The same software 20 years later



web-based user interfaces
java-based thick clients
The ho
screen scraping
How would we rebuild this from scratch?
databases outside the central storage
adapters for new application interfaces

a more modern approach:
a network of independent applications



Portal

Service-Gateway

Geschäftsanwendungen

Register

Domäne ABC

ABC-Geschäfts-anwendung

ABC-Analyse

ABC-Portal

Schreiben-Anwendung

ABC3-Beteiligung

ABC2-Beteiligung

Transport-Batch

Register ABC

Register ABC4

Register ABC4-Sachfahn-dung

Register ABC5

Domäne DEF

DEF-Verfahren

DEF-Analyse

GHI-Infor-mationen

GHI-Mail

GHI-Admin

DEF-Beispiel-verfahren

DEF-Portal

DEF-JKL-Beteiligung

DEF-MNO-Beteiligung

DEF-PQR-Beteiligung

DEF Warnungen Geschäfts-anwendung

Register DEFdatei

NGHI

Register DEF-Warnungen

Domäne STU

STU verfahren

Domäne VWX

VWX1-Geschäfts-anwendung

VWX2-Geschäfts-anwendung

VWX4-Geschäfts-anwendung

VWX3-Geschäfts-anwendung

Register VWX2

Register VWX

Register VWX3

Register VWX4

Domäne XAB

XAB-Geschäfts-anwendung

XAB-Statistik

XAB-Hilfe

Register XAB

erschnitt

XXX XXX

XXX XXX

XXX XXX

XXX XXX

XXX XXX

XXX

XXX

XXX XXX

XXX XXX

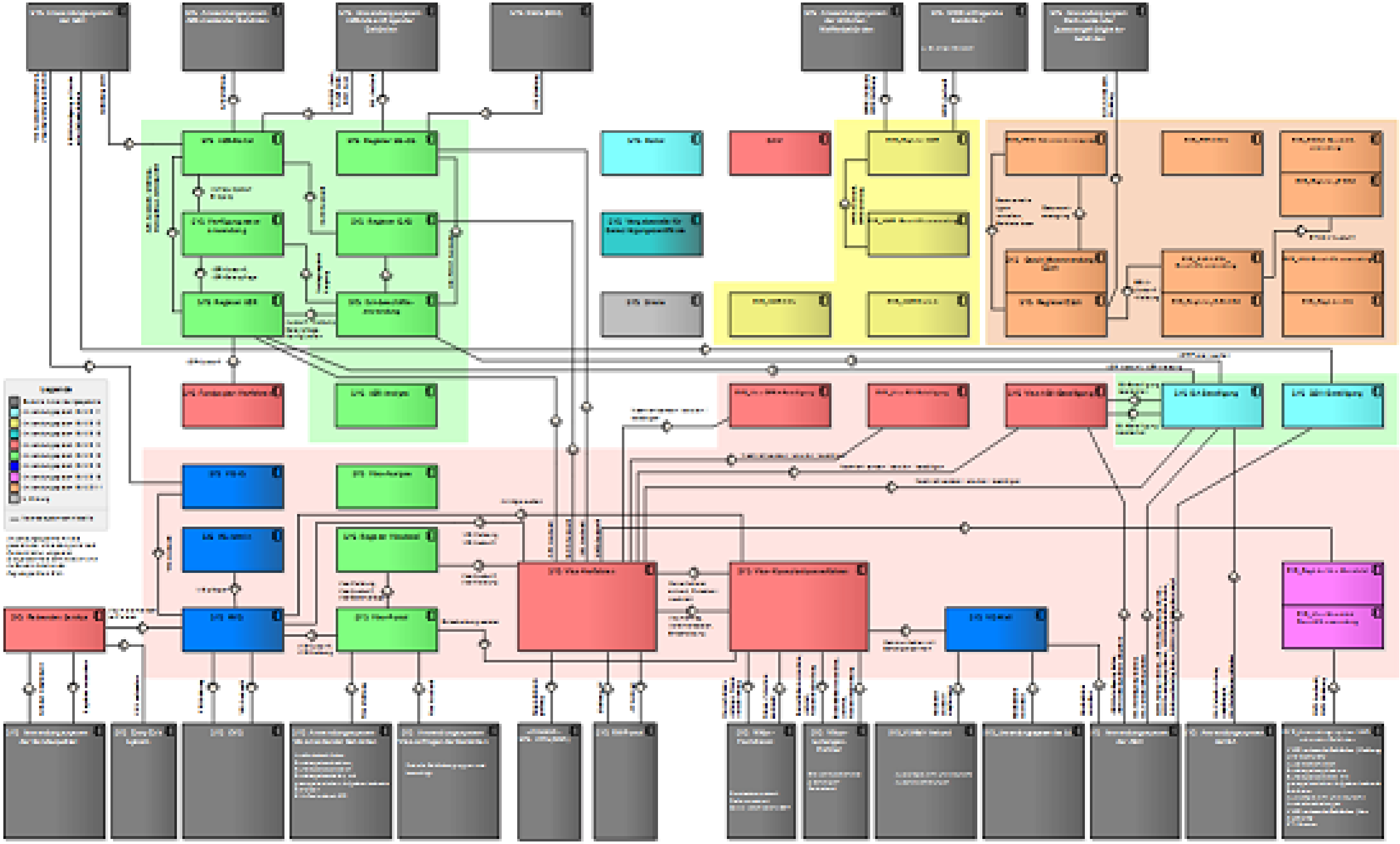
XXX XXX

XXX XXX

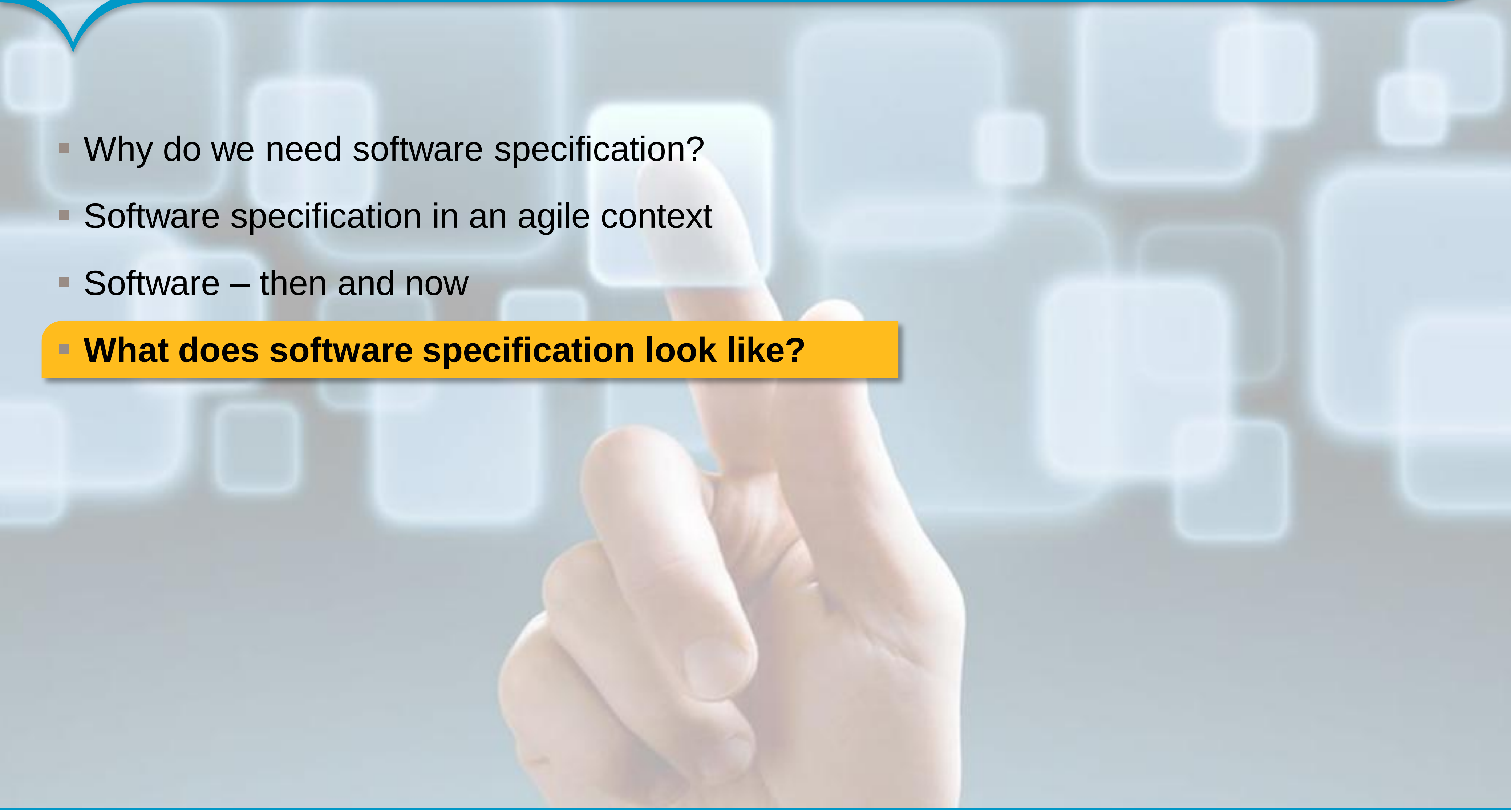
XXX XXX

XXX XXX

a more modern approach:
a network of independent applications

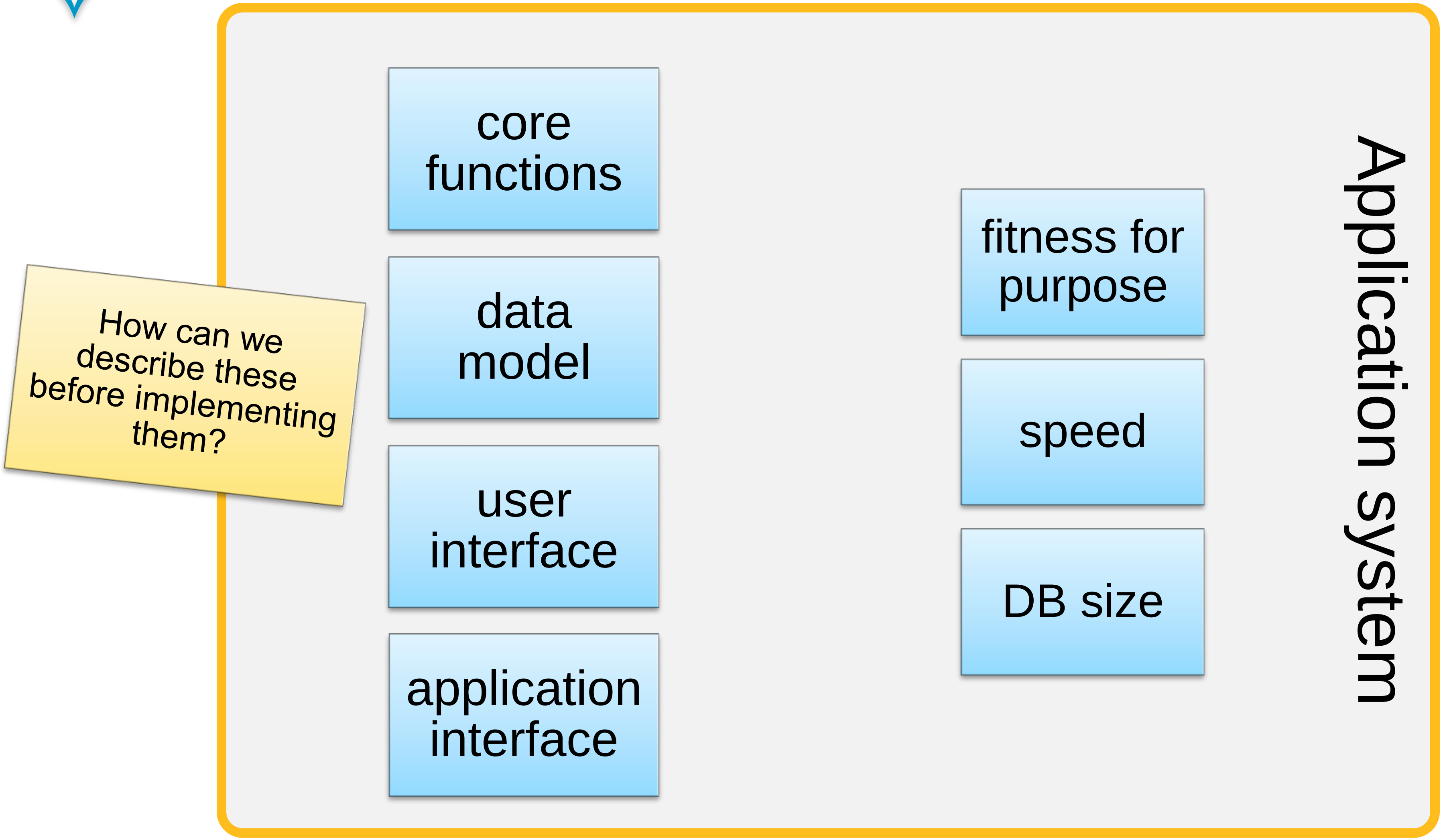


Agenda



- Why do we need software specification?
- Software specification in an agile context
- Software – then and now
- **What does software specification look like?**

Parts of an application system



How can we describe these before implementing them?

core functions

data model

user interface

application interface

fitness for purpose

speed

DB size

Application system

How can we describe these?

Description using…

- Pseudocode
- unstructured Plaintext
- Structured plaintext
- UML-modelling
- Screen prototyping

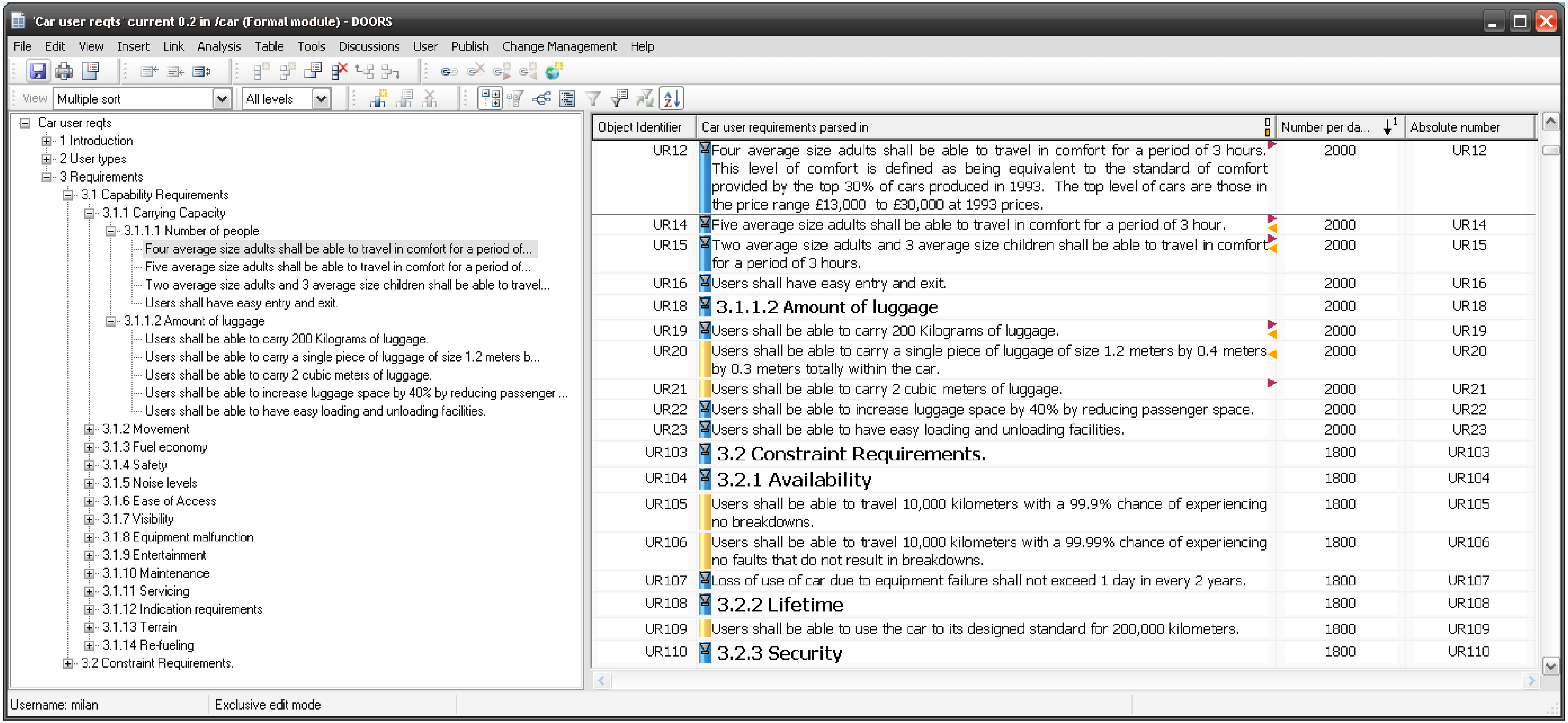
Methods

- Model-driven development
- Rational unified process (use cases)
- Agile / scrum (user stories)

Focus

- On core functions
- On application interface
- On user interface
- On data?

Requirements + Specification in IBM Rational Doors



'Car user reqts' current 0.2 in /car (Formal module) - DOORS

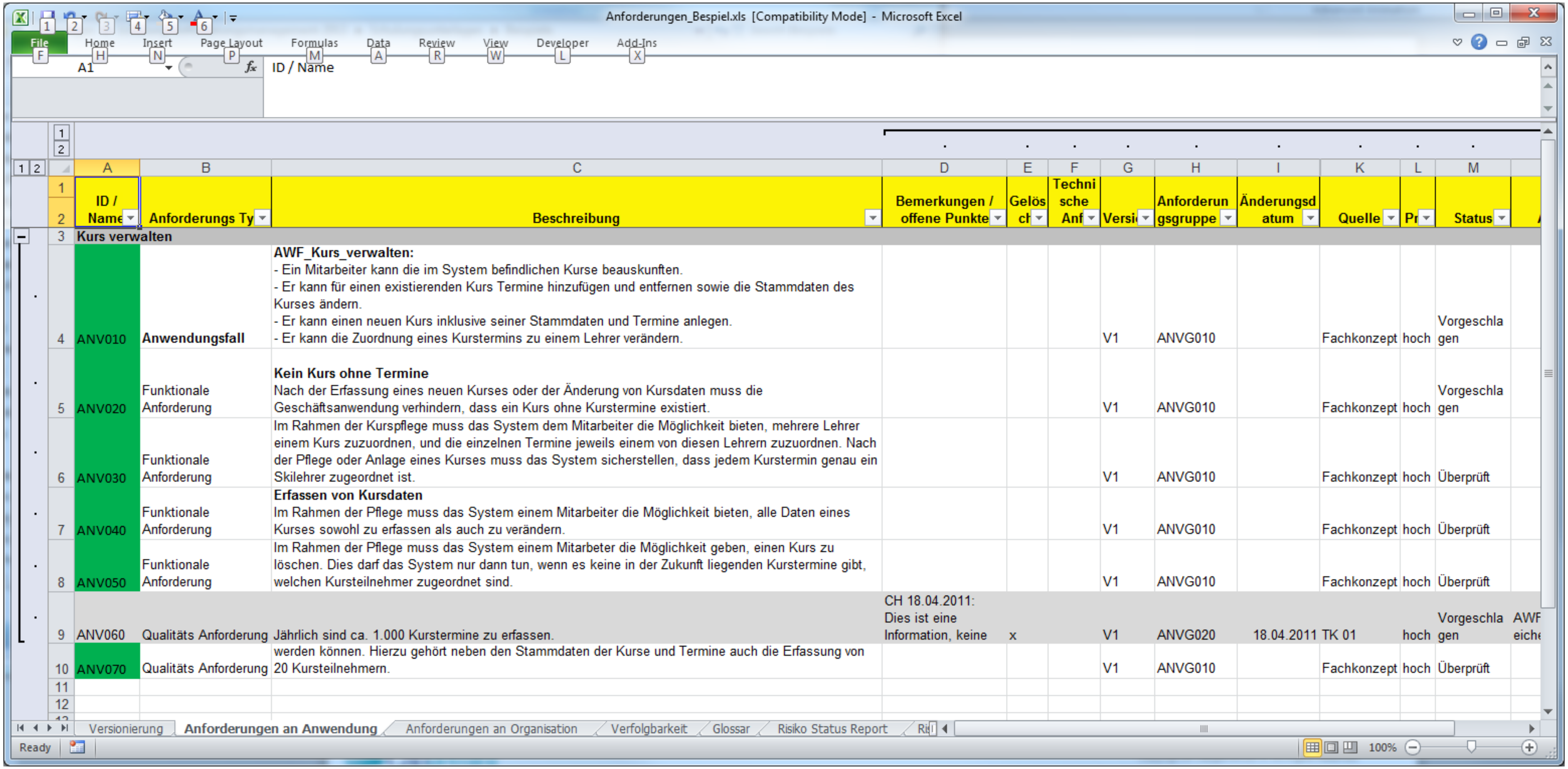
View: Multiple sort | All levels

- Car user reqts
  - 1 Introduction
  - 2 User types
  - 3 Requirements
    - 3.1 Capability Requirements
      - 3.1.1 Carrying Capacity
        - 3.1.1.1 Number of people
          - Four average size adults shall be able to travel in comfort for a period of...
          - Five average size adults shall be able to travel in comfort for a period of...
          - Two average size adults and 3 average size children shall be able to travel...
          - Users shall have easy entry and exit.
        - 3.1.1.2 Amount of luggage
          - Users shall be able to carry 200 Kilograms of luggage.
          - Users shall be able to carry a single piece of luggage of size 1.2 meters b...
          - Users shall be able to carry 2 cubic meters of luggage.
          - Users shall be able to increase luggage space by 40% by reducing passenger ...
          - Users shall be able to have easy loading and unloading facilities.
      - 3.1.2 Movement
      - 3.1.3 Fuel economy
      - 3.1.4 Safety
      - 3.1.5 Noise levels
      - 3.1.6 Ease of Access
      - 3.1.7 Visibility
      - 3.1.8 Equipment malfunction
      - 3.1.9 Entertainment
      - 3.1.10 Maintenance
      - 3.1.11 Servicing
      - 3.1.12 Indication requirements
      - 3.1.13 Terrain
      - 3.1.14 Re-fueling
    - 3.2 Constraint Requirements.

Object Identifier	Car user requirements parsed in	Number per da...	Absolute number
UR12	Four average size adults shall be able to travel in comfort for a period of 3 hours. This level of comfort is defined as being equivalent to the standard of comfort provided by the top 30% of cars produced in 1993. The top level of cars are those in the price range £13,000 to £30,000 at 1993 prices.	2000	UR12
UR14	Five average size adults shall be able to travel in comfort for a period of 3 hour.	2000	UR14
UR15	Two average size adults and 3 average size children shall be able to travel in comfort for a period of 3 hours.	2000	UR15
UR16	Users shall have easy entry and exit.	2000	UR16
UR18	**3.1.1.2 Amount of luggage**	2000	UR18
UR19	Users shall be able to carry 200 Kilograms of luggage.	2000	UR19
UR20	Users shall be able to carry a single piece of luggage of size 1.2 meters by 0.4 meters by 0.3 meters totally within the car.	2000	UR20
UR21	Users shall be able to carry 2 cubic meters of luggage.	2000	UR21
UR22	Users shall be able to increase luggage space by 40% by reducing passenger space.	2000	UR22
UR23	Users shall be able to have easy loading and unloading facilities.	2000	UR23
UR103	**3.2 Constraint Requirements.**	1800	UR103
UR104	**3.2.1 Availability**	1800	UR104
UR105	Users shall be able to travel 10,000 kilometers with a 99.9% chance of experiencing no breakdowns.	1800	UR105
UR106	Users shall be able to travel 10,000 kilometers with a 99.99% chance of experiencing no faults that do not result in breakdowns.	1800	UR106
UR107	Loss of use of car due to equipment failure shall not exceed 1 day in every 2 years.	1800	UR107
UR108	**3.2.2 Lifetime**	1800	UR108
UR109	Users shall be able to use the car to its designed standard for 200,000 kilometers.	1800	UR109
UR110	**3.2.3 Security**	1800	UR110

Username: milan | Exclusive edit mode

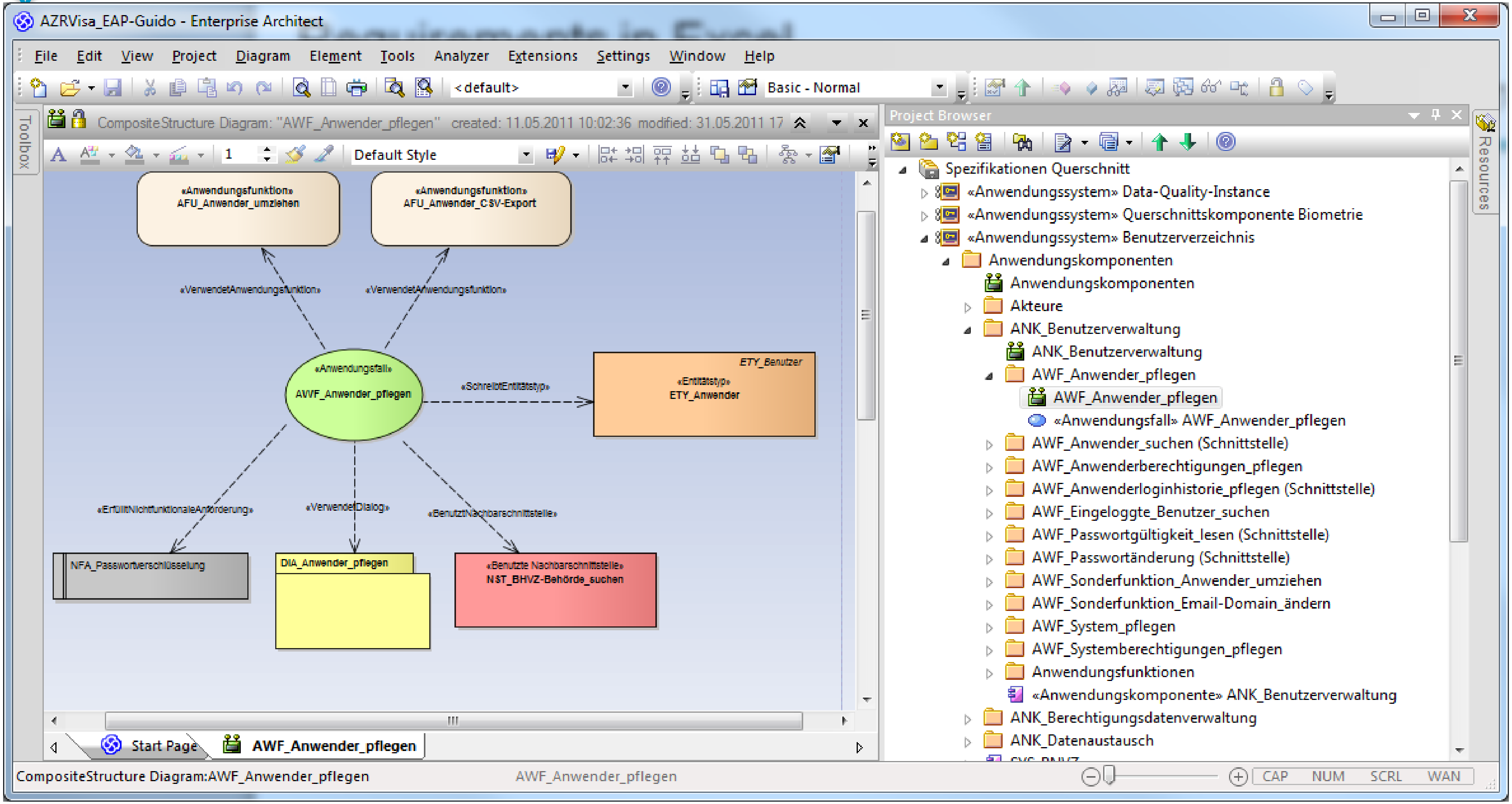
Requirements in Excel,
Specification in Sparx Enterprise Architect and MS Word



	A	B	C	D	E	F	G	H	I	K	L	M	
1–2	ID / Name	Anforderungs Ty	Beschreibung	Bemerkungen / offene Punkte	Gelösch	Technische Anf	Versi	Anforderungsgruppe	Änderungsdatum	Quelle	Pr	Status	
3	Kurs verwalten												
4	ANV010	Anwendungsfall	**AWF_Kurs_verwalten:** - Ein Mitarbeiter kann die im System befindlichen Kurse beauskunften. - Er kann für einen existierenden Kurs Termine hinzufügen und entfernen sowie die Stammdaten des Kurses ändern. - Er kann einen neuen Kurs inklusive seiner Stammdaten und Termine anlegen. - Er kann die Zuordnung eines Kurstermins zu einem Lehrer verändern.				V1	ANVG010		Fachkonzept	hoch	Vorgeschlagen	
5	ANV020	Funktionale Anforderung	**Kein Kurs ohne Termine** Nach der Erfassung eines neuen Kurses oder der Änderung von Kursdaten muss die Geschäftsanwendung verhindern, dass ein Kurs ohne Kurstermine existiert.				V1	ANVG010		Fachkonzept	hoch	Vorgeschlagen	
6	ANV030	Funktionale Anforderung	Im Rahmen der Kurspflege muss das System dem Mitarbeiter die Möglichkeit bieten, mehrere Lehrer einem Kurs zuzuordnen, und die einzelnen Termine jeweils einem von diesen Lehrern zuzuordnen. Nach der Pflege oder Anlage eines Kurses muss das System sicherstellen, dass jedem Kurstermin genau ein Skilehrer zugeordnet ist.				V1	ANVG010		Fachkonzept	hoch	Überprüft	
7	ANV040	Funktionale Anforderung	**Erfassen von Kursdaten** Im Rahmen der Pflege muss das System einem Mitarbeiter die Möglichkeit bieten, alle Daten eines Kurses sowohl zu erfassen als auch zu verändern.				V1	ANVG010		Fachkonzept	hoch	Überprüft	
8	ANV050	Funktionale Anforderung	Im Rahmen der Pflege muss das System einem Mitarbeter die Möglichkeit geben, einen Kurs zu löschen. Dies darf das System nur dann tun, wenn es keine in der Zukunft liegenden Kurstermine gibt, welchen Kursteilnehmer zugeordnet sind.				V1	ANVG010		Fachkonzept	hoch	Überprüft	
9	ANV060	Qualitäts Anforderung	Jährlich sind ca. 1.000 Kurstermine zu erfassen.	CH 18.04.2011: Dies ist eine Information, keine	x		V1	ANVG020	18.04.2011	TK 01	hoch	Vorgeschlagen	AWF eiche
10	ANV070	Qualitäts Anforderung	werden können. Hierzu gehört neben den Stammdaten der Kurse und Termine auch die Erfassung von 20 Kursteilnehmern.				V1	ANVG010		Fachkonzept	hoch	Überprüft	
11													
12													

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Requirements in Excel,
Specification in Sparx Enterprise Architect and MS Word



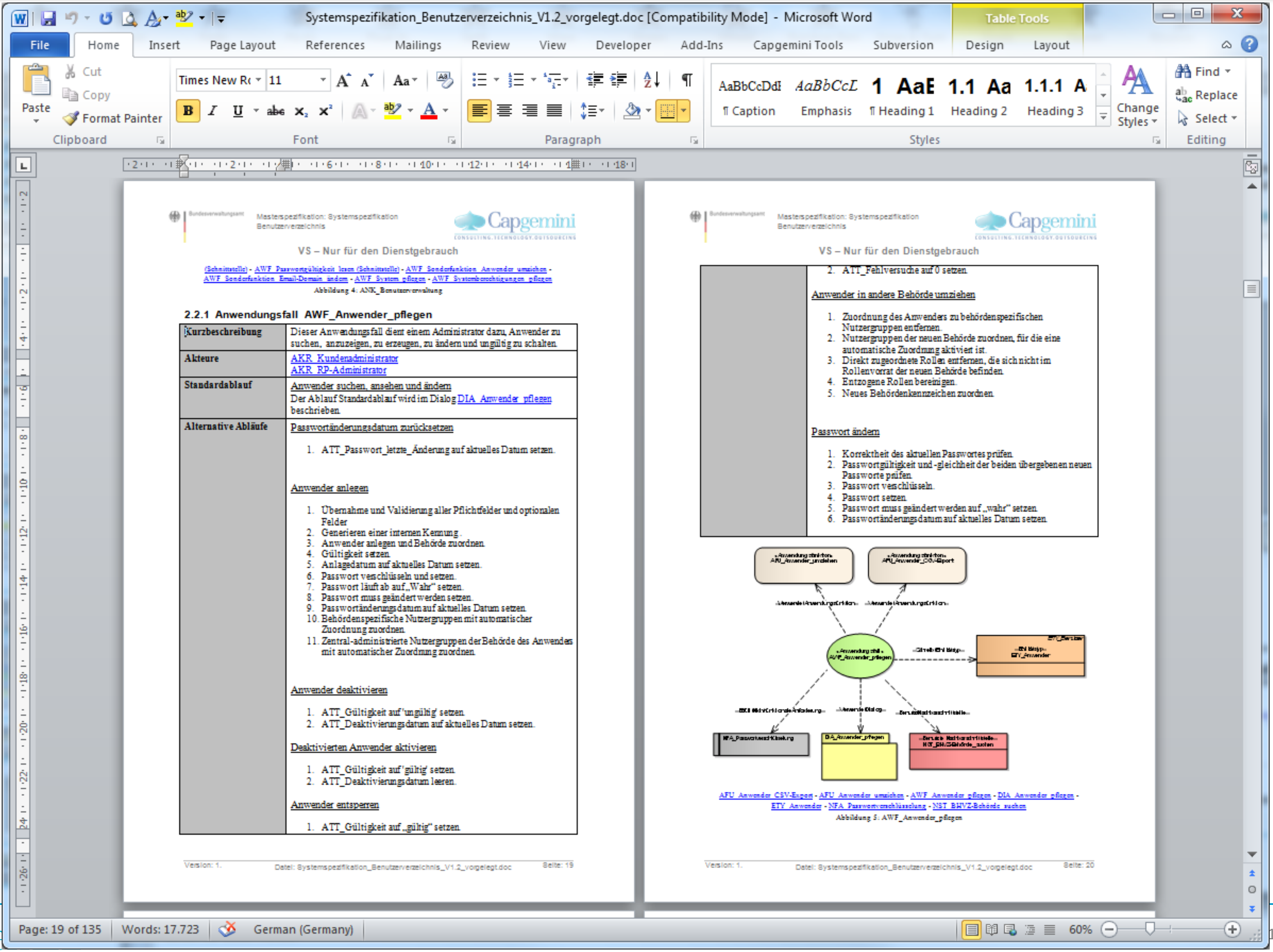
CompositeStructure Diagram: "AWF_Anwender_pflegen" created: 11.05.2011 10:02:36 modified: 31.05.2011 17

Project Browser:

- Spezifikationen Querschnitt
  - «Anwendungssystem» Data-Quality-Instance
  - «Anwendungssystem» Querschnittskomponente Biometrie
  - «Anwendungssystem» Benutzerverzeichnis
    - Anwendungskomponenten
      - Anwendungskomponenten
      - Akteure
      - ANK_Benutzerverwaltung
        - ANK_Benutzerverwaltung
        - AWF_Anwender_pflegen
          - AWF_Anwender_pflegen
          - «Anwendungsfall» AWF_Anwender_pflegen
        - AWF_Anwender_suchen (Schnittstelle)
        - AWF_Anwenderberechtigungen_pflegen
        - AWF_Anwenderloginhistorie_pflegen (Schnittstelle)
        - AWF_Eingeloggte_Benutzer_suchen
        - AWF_Passwortgültigkeit_lesen (Schnittstelle)
        - AWF_Passwortänderung (Schnittstelle)
        - AWF_Sonderfunktion_Anwender_umziehen
        - AWF_Sonderfunktion_Email-Domain_ändern
        - AWF_System_pflegen
        - AWF_Systemberechtigungen_pflegen
        - Anwendungsfunktionen
        - «Anwendungskomponente» ANK_Benutzerverwaltung
      - ANK_Berechtigungsdatenverwaltung
      - ANK_Datenaustausch

CompositeStructure Diagram:AWF_Anwender_pflegen AWF_Anwender_pflegen

Requirements in Excel,
Specification in Sparx Enterprise Architect and MS Word



Bundesverwaltungsamt

Masterspezifikation: Systemspezifikation
Benutzerverzeichnis

VS – Nur für den Dienstgebrauch

(Schnittstelle) - AWF_Passwortgültigkeit_lesen (Schnittstelle) - AWF_Sonderfunktion_Anwender_umziehen - AWF_Sonderfunktion_Email-Domain_ändern - AWF_System_pflegen - AWF_Systemberechtigungen_pflegen

Abbildung 4: ANK_Benutzerverwaltung

2.2.1 Anwendungsfall AWF_Anwender_pflegen

Kurzbeschreibung	Dieser Anwendungsfall dient einem Administrator dazu, Anwender zu suchen, anzuzeigen, zu erzeugen, zu ändern und ungültig zu schalten.
Akteure	AKR_Kundenadministrator AKR_RP-Administrator
Standardablauf	Anwender suchen, ansehen und ändern Der Ablauf Standardablauf wird im Dialog DIA_Anwender_pflegen beschrieben.
Alternative Abläufe	Passwortänderungsdatum zurücksetzen 1. ATT_Passwort_letzte_Änderung auf aktuelles Datum setzen. Anwender anlegen 1. Übernahme und Validierung aller Pflichtfelder und optionalen Felder 2. Generieren einer internen Kennung . 3. Anwender anlegen und Behörde zuordnen. 4. Gültigkeit setzen. 5. Anlagedatum auf aktuelles Datum setzen. 6. Passwort verschlüsseln und setzen. 7. Passwort läuft ab auf „Wahr" setzen. 8. Passwort muss geändert werden setzen. 9. Passwortänderungsdatum auf aktuelles Datum setzen. 10. Behördenspezifische Nutzergruppen mit automatischer Zuordnung zuordnen. 11. Zentral-administrierte Nutzergruppen der Behörde des Anwenders mit automatischer Zuordmng zuordnen. Anwender deaktivieren 1. ATT_Gültigkeit auf 'ungültig' setzen. 2. ATT_Deaktivierungsdatum auf aktuelles Datum setzen. Deaktivierten Anwender aktivieren 1. ATT_Gültigkeit auf 'gültig' setzen. 2. ATT_Deaktivierungsdatum leeren. Anwender entsperren 1. ATT_Gültigkeit auf „gültig" setzen. 2. ATT_Fehlversuche auf 0 setzen. Anwender in andere Behörde umziehen 1. Zuordnung des Anwenders zu behördenspezifischen Nutzergruppen entfernen. 2. Nutzergruppen der neuen Behörde zuordnen, für die eine automatische Zuordmng aktiviert ist. 3. Direkt zugeordnete Rollen entfernen, die sich nicht im Rollenvorrat der neuen Behörde befinden. 4. Entzogene Rollen bereinigen. 5. Neues Behördenkennzeichen zuordnen. Passwort ändern 1. Korrektheit des aktuellen Passwortes prüfen. 2. Passwortgültigkeit und -gleichheit der beiden übergebenen neuen Passworte prüfen. 3. Passwort verschlüsseln. 4. Passwort setzen. 5. Passwort muss geändert werden auf „wahr" setzen. 6. Passwortänderungsdatum auf aktuelles Datum setzen.

Version: 1. Datei: Systemspezifikation_Benutzerverzeichnis_V1.2_vorgelegt.doc Seite: 19

Bundesverwaltungsamt

Masterspezifikation: Systemspezifikation
Benutzerverzeichnis

VS – Nur für den Dienstgebrauch

AFU_Anwender_CSV-Export - AFU_Anwender_umziehen - AWF_Anwender_pflegen - DIA_Anwender_pflegen - ETY_Anwender - NFA_Passwortverschlüsselung - NST_BHVZ-Behörde_suchen

Abbildung 5: AWF_Anwender_pflegen

Version: 1. Datei: Systemspezifikation_Benutzerverzeichnis_V1.2_vorgelegt.doc Seite: 20

Requirements, tests, bugs, user stories in Bugtracking tool, Specification in a powerful Wiki

Idea: web based solution

Enables deep linking btw. all kinds of artifacts

Needs explicit versioning and branches

Use changes btw. versions

Difficult: Longform text editing in wiki source

Difficult: UML design in same versioning system as wiki text!?

I want this! ☺

Which is the right toolchain for our project?

customer standards

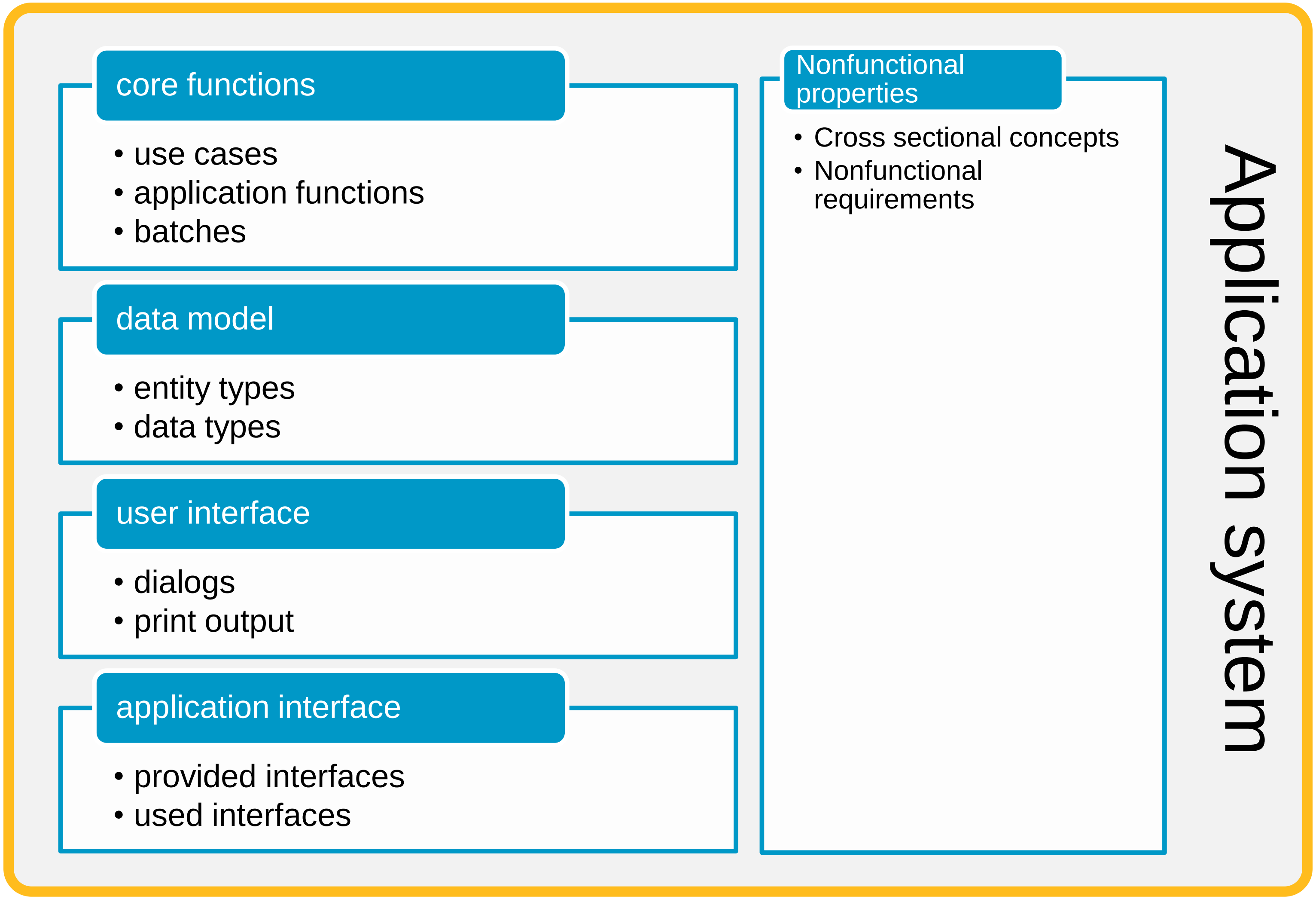
team know how

project size

licence fees

…

Parts of an application system: Using the Capgemini specification method



core functions

- use cases
- application functions
- batches

data model

- entity types
- data types

user interface

- dialogs
- print output

application interface

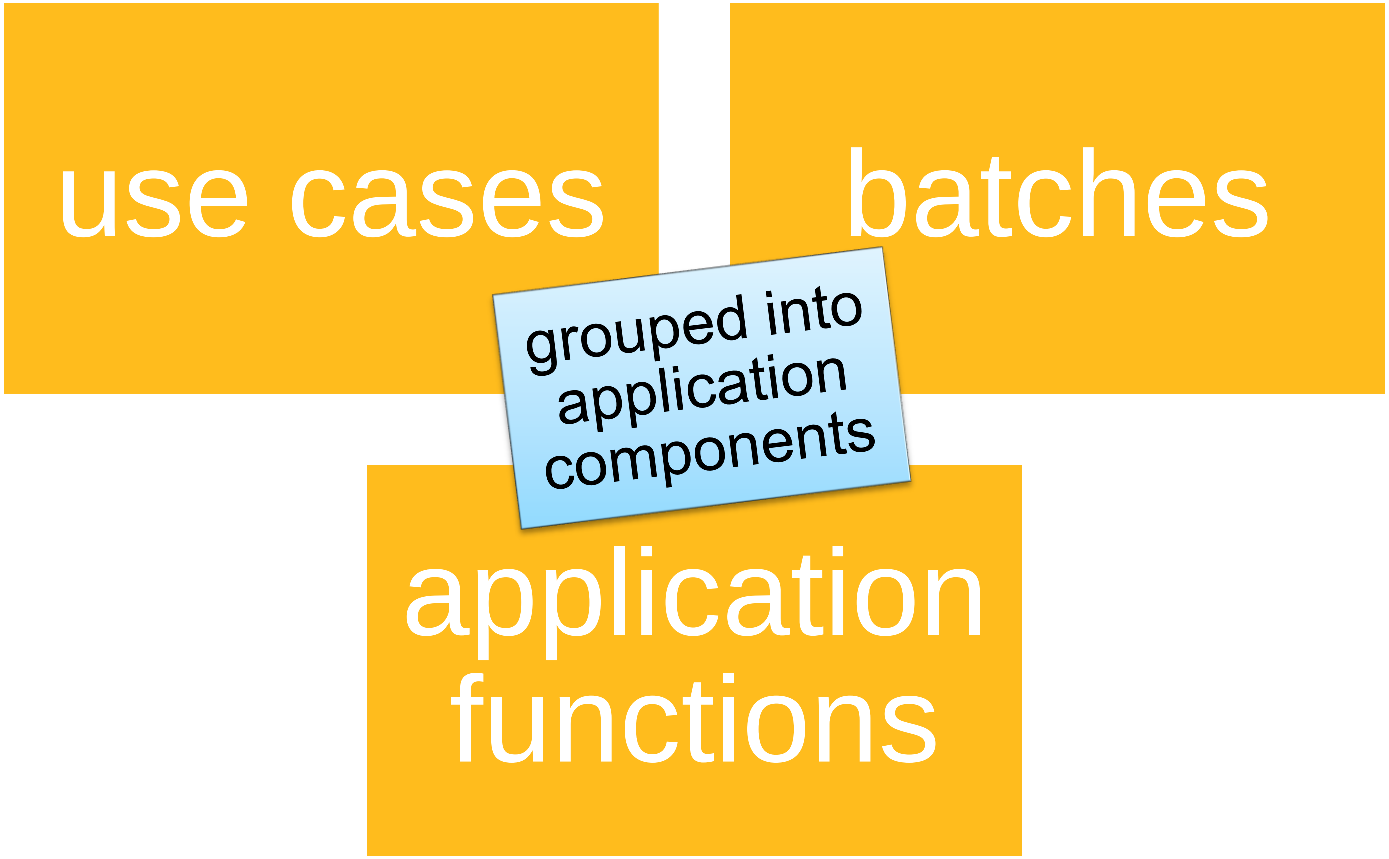
- provided interfaces
- used interfaces

Nonfunctional properties

- Cross sectional concepts
- Nonfunctional requirements

Application system

The system‘s core functions



use cases

batches

grouped into application components

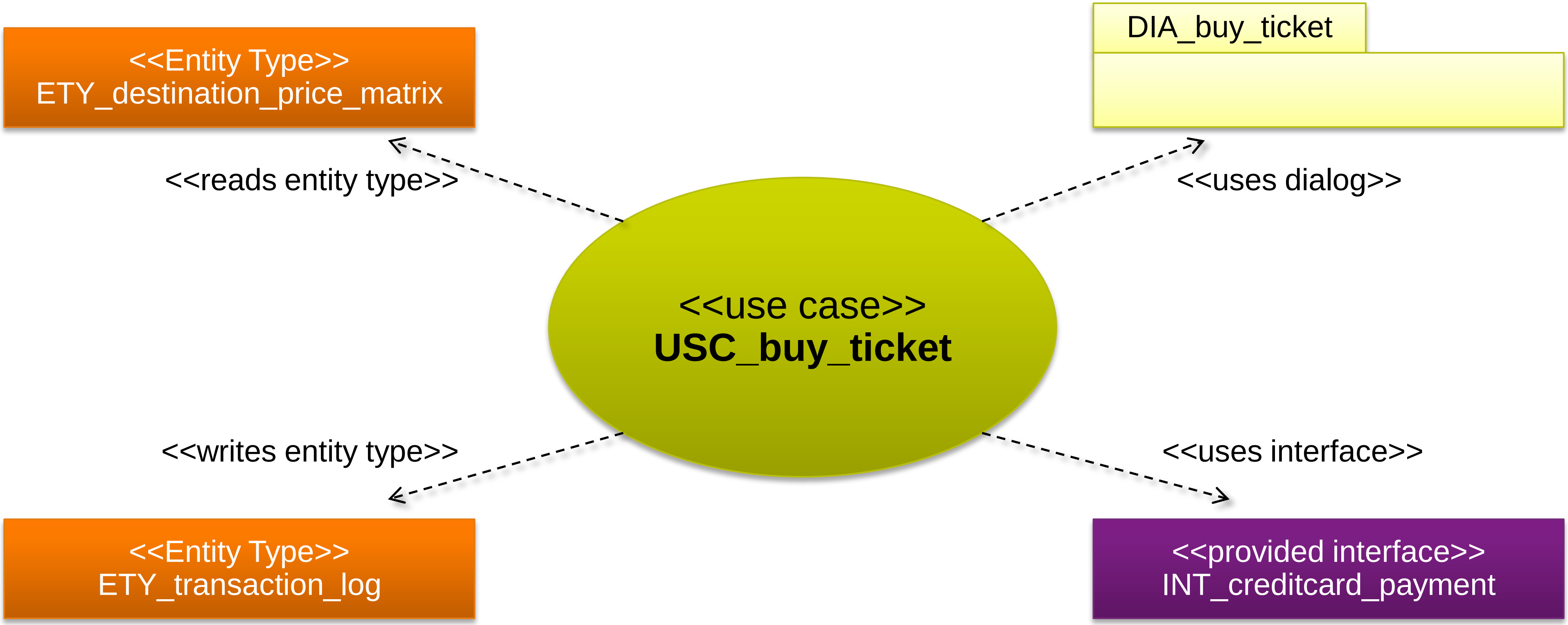
application functions

A use case describes a core functionality of the application system as a sequence of steps

Use Case USC_pay_ticket

description	The passenger pays his public transport ticket using money or a credit card.
actor	ACR_passenger
precondition	The passenger has selected his destination.
standard sequence	Buy ticket with cash 1. Vending machine shows the amount due. 2. Passenger inserts at least the amount due in coins and/or bills. If he cancels the transaction or a timeout occurs before enough money is inserted, the machine returns his money and the use case ends. 3. Vending machine prints the ticket with current date and destination (print output PRO_ticket)
alternative sequences	Buy ticket with credit card …

The use case diagram shows all artefacts connected to the use case



<<Entity Type>>
ETY_destination_price_matrix

DIA_buy_ticket

<<reads entity type>>

<<uses dialog>>

<<use case>>
USC_buy_ticket

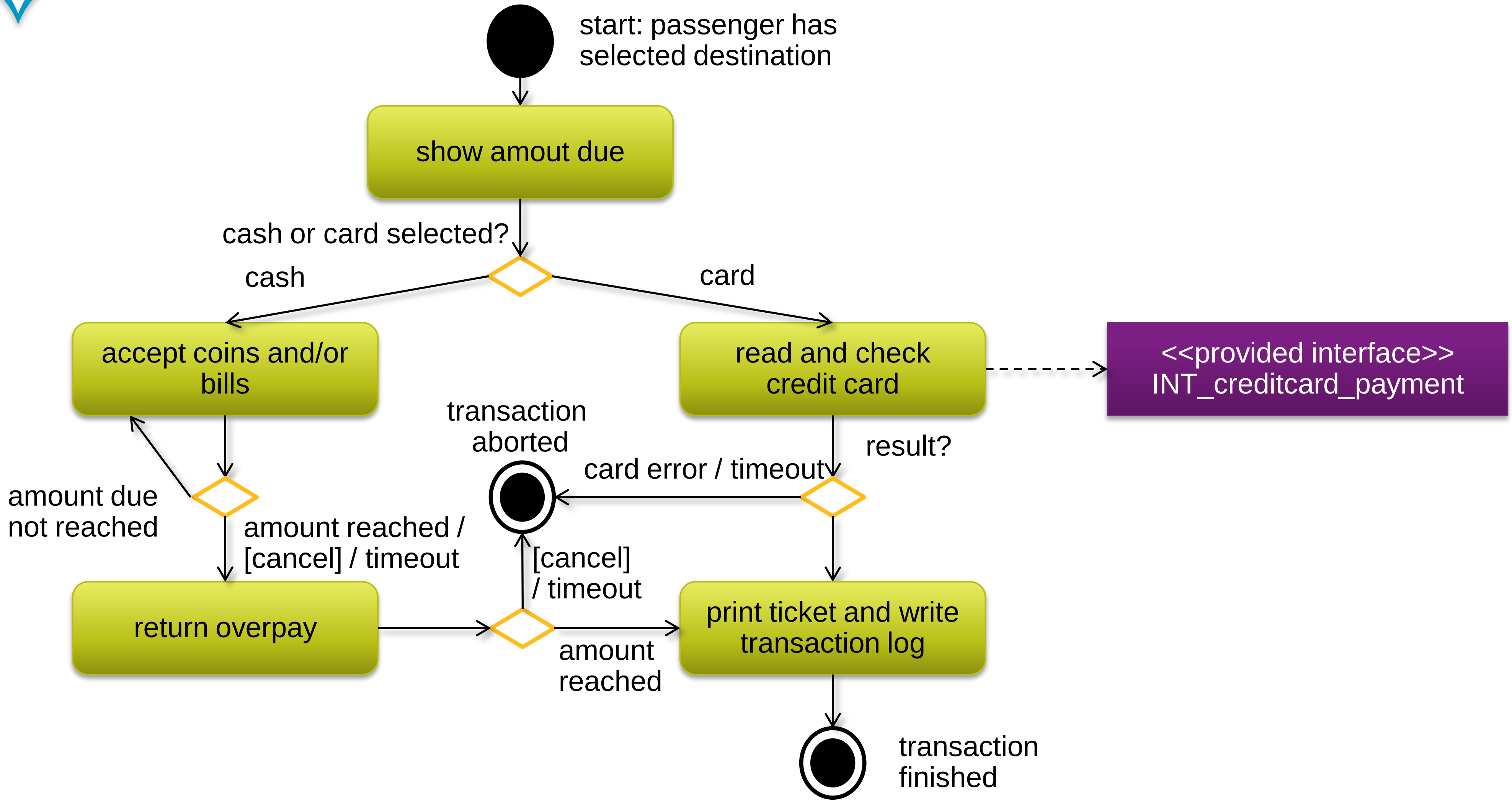
<<writes entity type>>

<<uses interface>>

<<Entity Type>>
ETY_transaction_log

<<provided interface>>
INT_creditcard_payment

the activity diagram shows flow of steps for all sequences and interface usage


start: passenger has selected destination
show amout due
cash or card selected?
cash
card
accept coins and/or bills
read and check credit card
<<provided interface>> INT_creditcard_payment
transaction aborted
result?
card error / timeout
amount due not reached
amount reached / [cancel] / timeout
[cancel] / timeout
return overpay
print ticket and write transaction log
amount reached
transaction finished

Use case functions encapsulate actions…

…that are needed by many use cases

or

…that cannot be described in a single sentence

Batches…

are like use cases

but called on command line or via timers by operations people

The system‘s data model

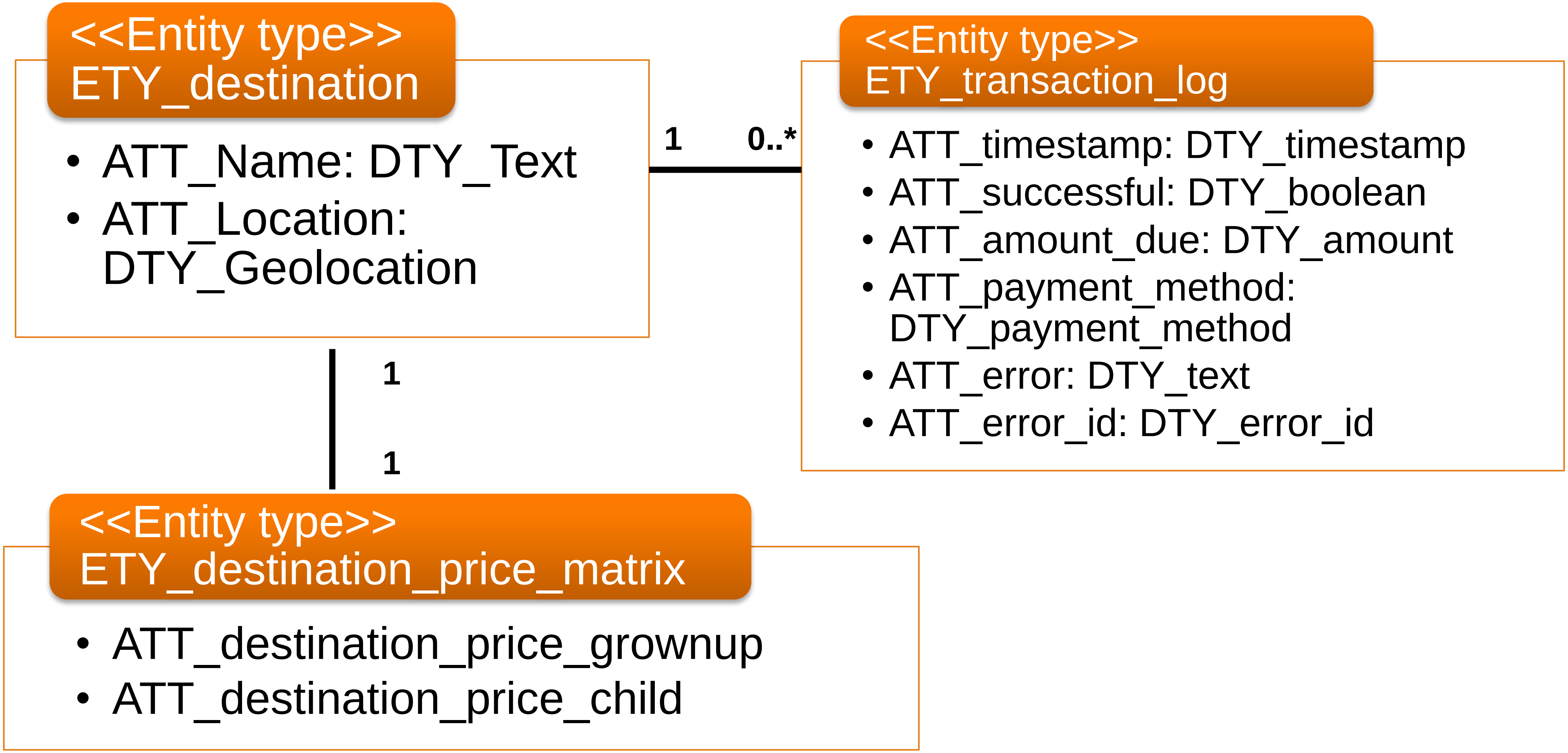
entity types

grouped into model components

data types

defined globally

Example: Data model of a ticket vending machine



<<Entity type>>
ETY_destination

- ATT_Name: DTY_Text
- ATT_Location: DTY_Geolocation

<<Entity type>>
ETY_transaction_log

- ATT_timestamp: DTY_timestamp
- ATT_successful: DTY_boolean
- ATT_amount_due: DTY_amount
- ATT_payment_method: DTY_payment_method
- ATT_error: DTY_text
- ATT_error_id: DTY_error_id

<<Entity type>>
ETY_destination_price_matrix

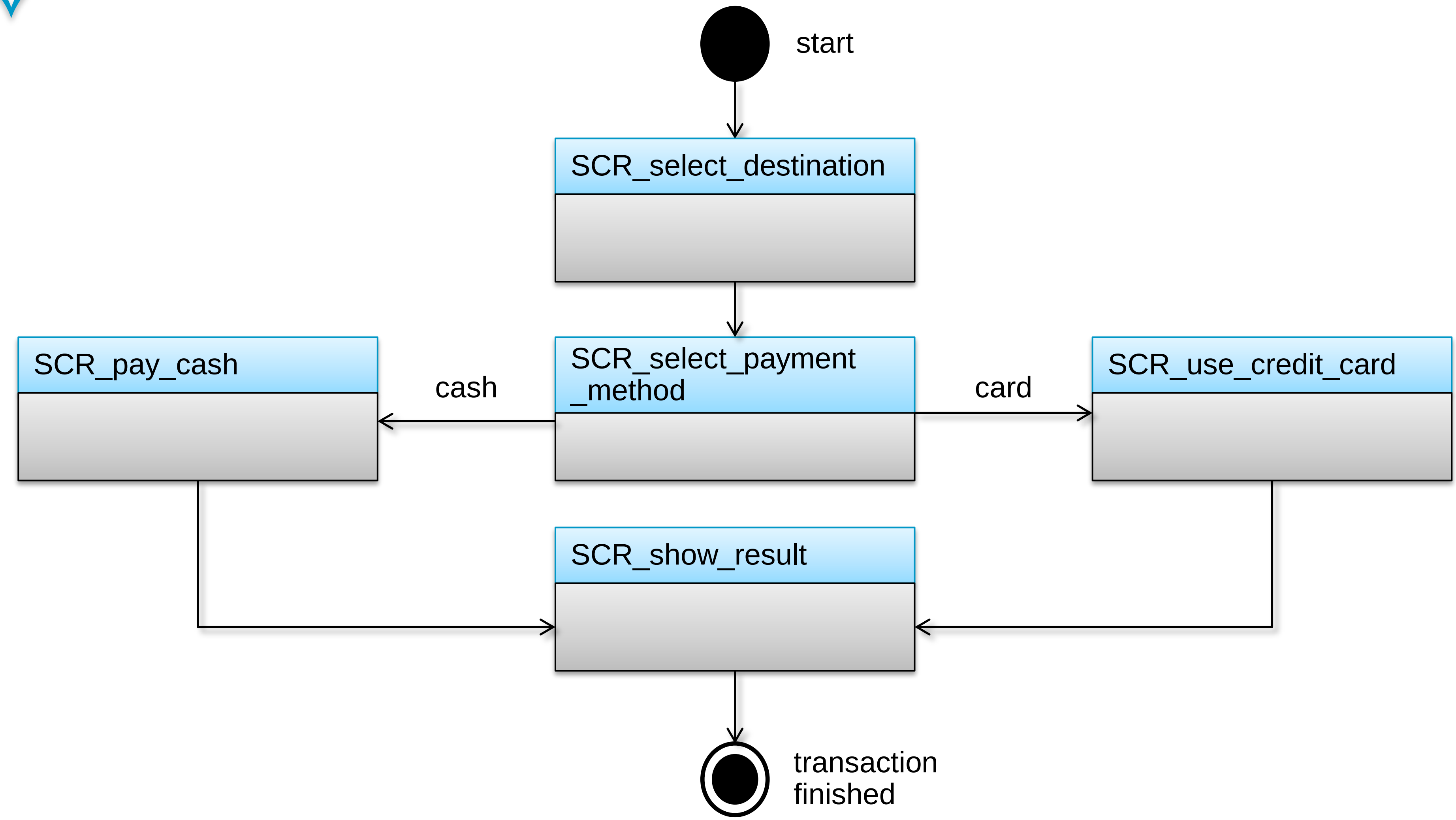
- ATT_destination_price_grownup
- ATT_destination_price_child

The system‘s user interface

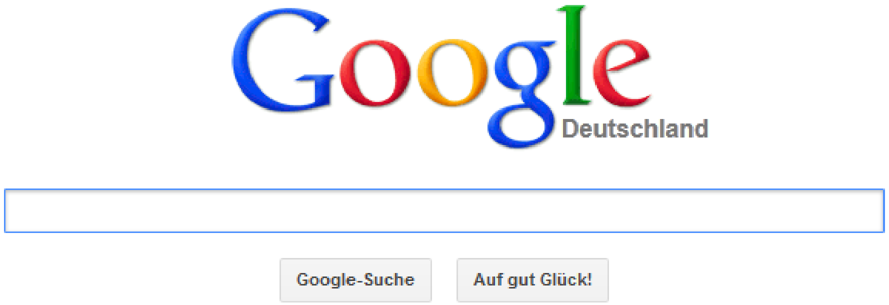
dialogs

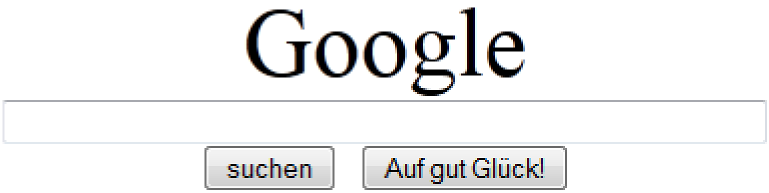
print outputs

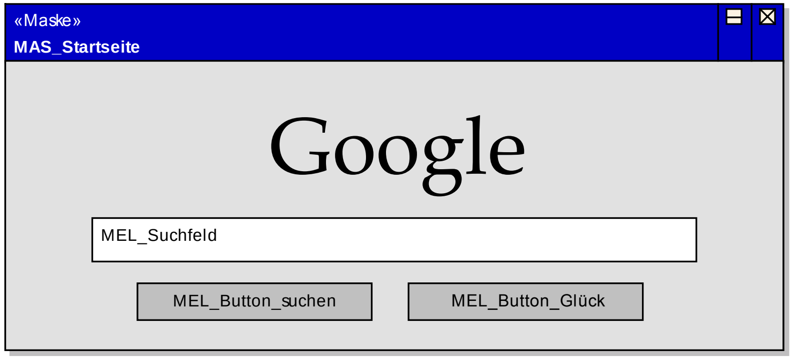
A dialog describes a sequence of screens. Example: DIA_buy_ticket

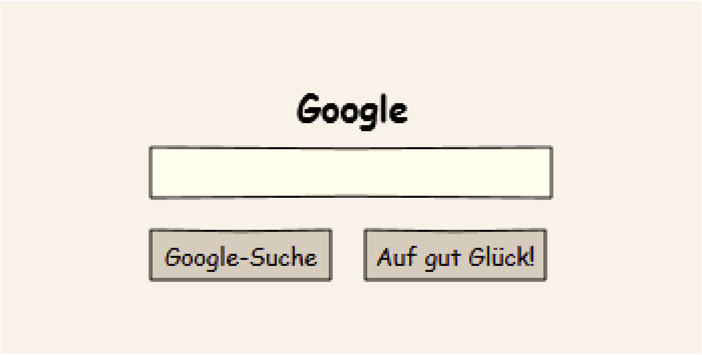


Screen mockups show a typical representation of the screen

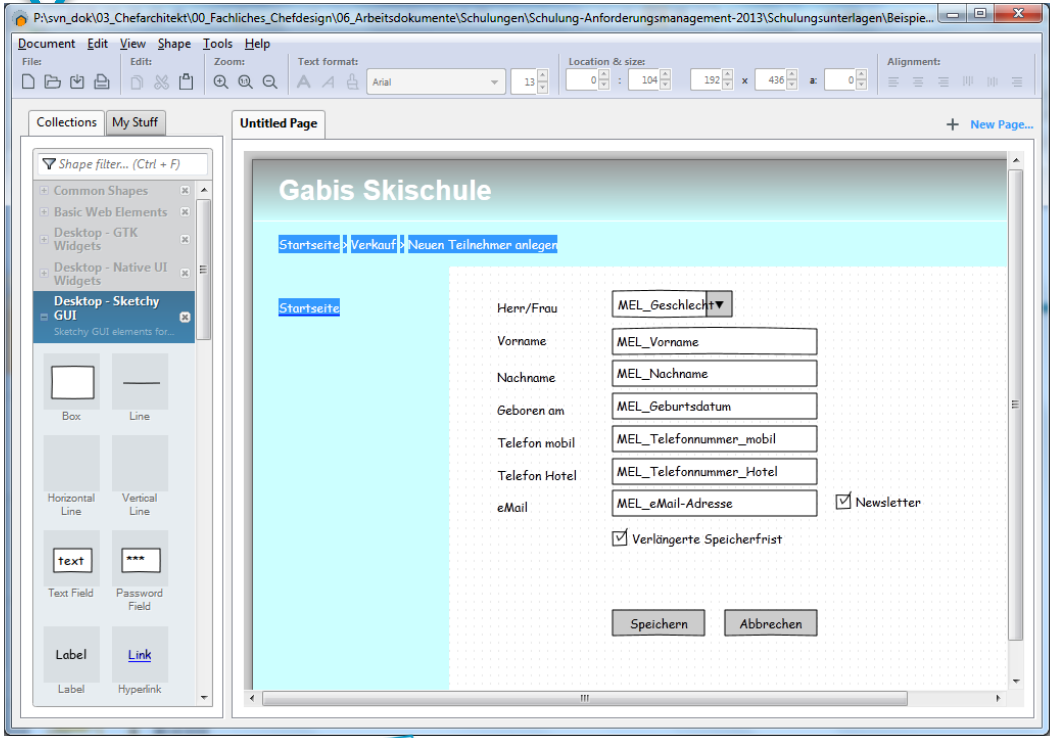








UI Prototyping tool “Evolus pencil”


P:\svn_dok\03_Chefarchitekt\00_Fachliches_Chefdesign\06_Arbeitsdokumente\Schulungen\Schulung-Anforderungsmanagement-2013\Schulungsunterlagen\Beispie...
Document Edit View Shape Tools Help
File: Edit: Zoom: Text format: Location & size: Alignment:
Arial 13 0 104 192 436 0
Collections My Stuff
Untitled Page
New Page...
Shape filter... (Ctrl + F)
Common Shapes
Basic Web Elements
Desktop - GTK Widgets
Desktop - Native UI Widgets
Desktop - Sketchy GUI
Sketchy GUI elements for...
Box
Line
Horizontal Line
Vertical Line
text
Text Field
***
Password Field
Label
Label
Link
Hyperlink

Gabis Skischule

Startseite > Verkauf > Neuen Teilnehmer anlegen

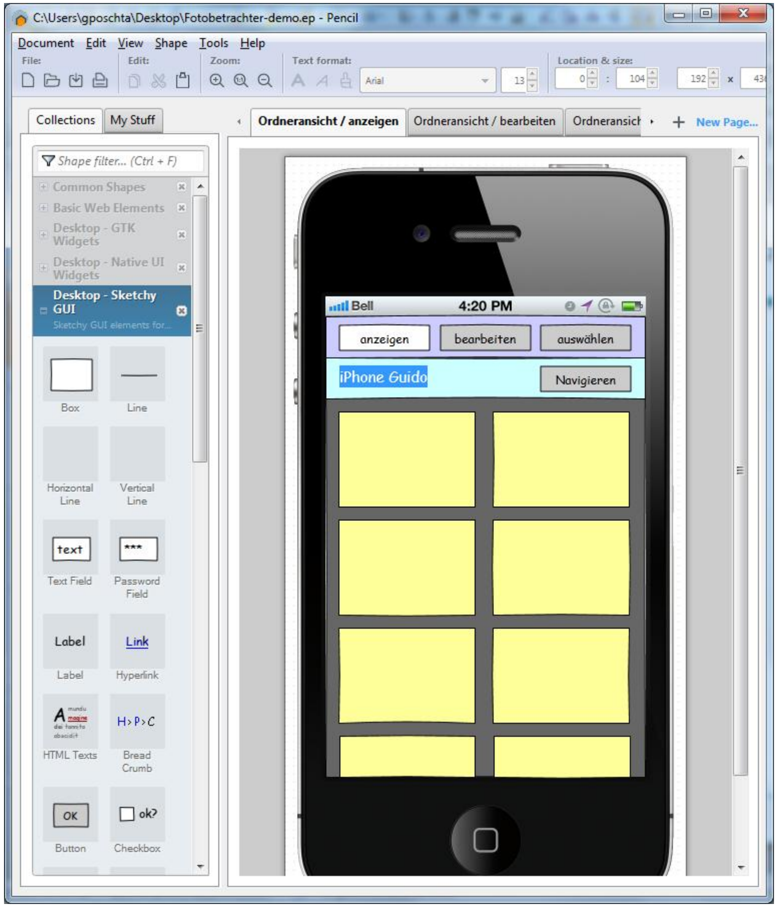
Startseite

Herr/Frau	MEL_Geschlecht ▼
Vorname	MEL_Vorname
Nachname	MEL_Nachname
Geboren am	MEL_Geburtsdatum
Telefon mobil	MEL_Telefonnummer_mobil
Telefon Hotel	MEL_Telefonnummer_Hotel
eMail	MEL_eMail-Adresse

☑ Newsletter

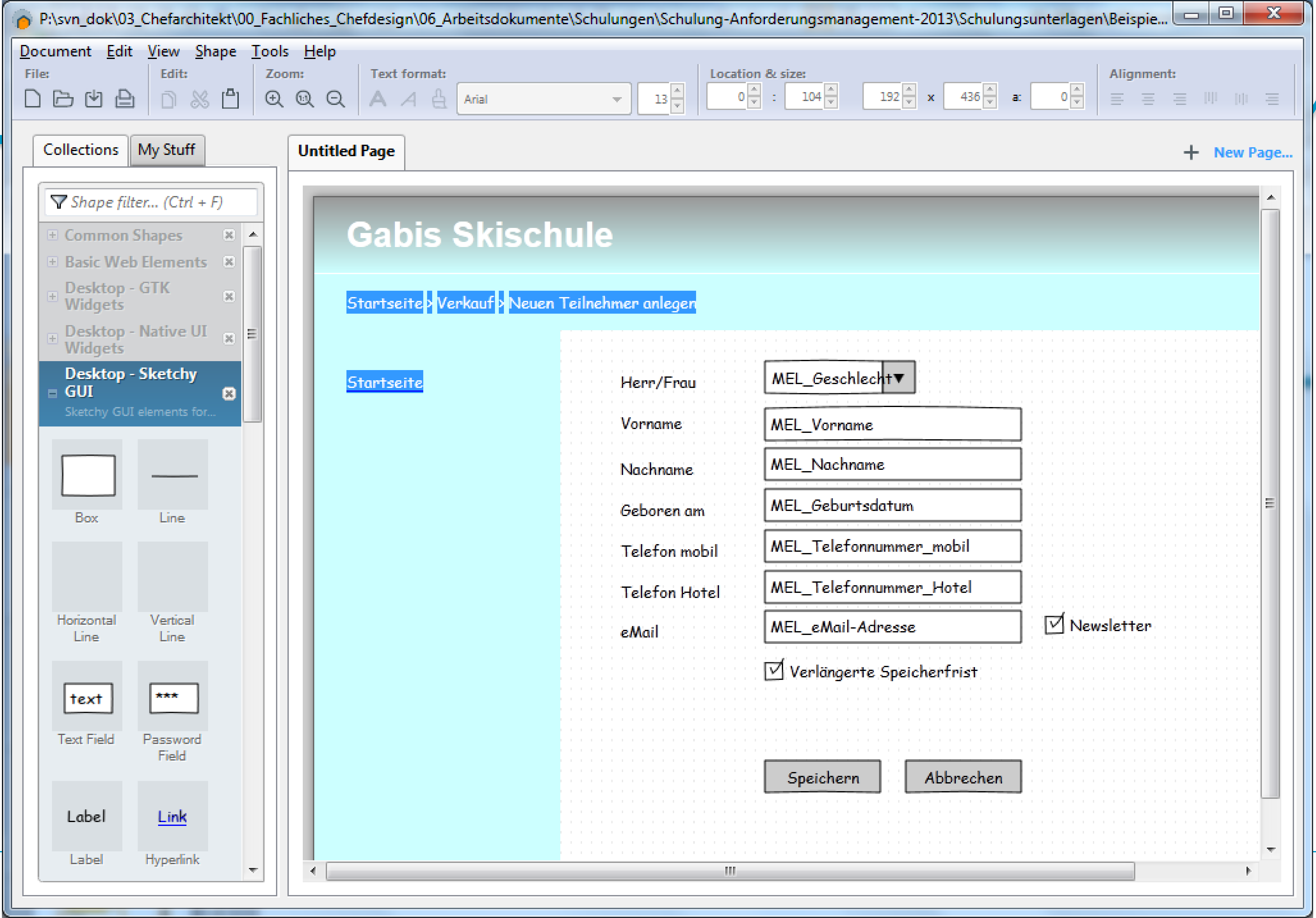
☑ Verlängerte Speicherfrist

Speichern Abbrechen



Great tool for simple prototypes

Not powerful enough for detailed description of modern UI



P:\svn_dok\03_Chefarchitekt\00_Fachliches_Chefdesign\06_Arbeitsdokumente\Schulungen\Schulung-Anforderungsmanagement-2013\Schulungsunterlagen\Beispie...

Arial 13

Location & size: 0 : 104 192 x 436 a: 0

Gabis Skischule

Startseite > Verkauf > Neuen Teilnehmer anlegen

Startseite

Herr/Frau: MEL_Geschlecht ▼

Vorname: MEL_Vorname

Nachname: MEL_Nachname

Geboren am: MEL_Geburtsdatum

Telefon mobil: MEL_Telefonnummer_mobil

Telefon Hotel: MEL_Telefonnummer_Hotel

eMail: MEL_eMail-Adresse

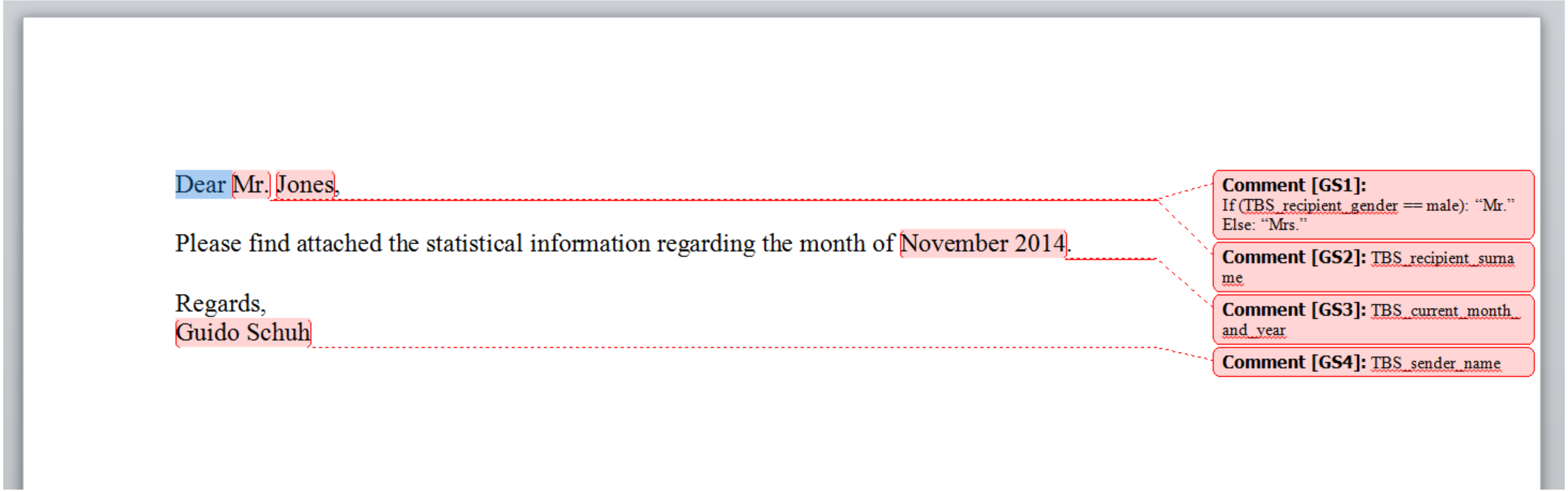
☑ Newsletter

☑ Verlängerte Speicherfrist

Speichern

Abbrechen

Print outputs



Dear Mr. Jones,

Please find attached the statistical information regarding the month of November 2014.

Regards,
Guido Schuh

Comment [GS1]:
If (TBS_recipient_gender == male): "Mr."
Else: "Mrs."

Comment [GS2]: TBS_recipient_surname

Comment [GS3]: TBS_current_month_and_year

Comment [GS4]: TBS_sender_name

The system‘s application interface

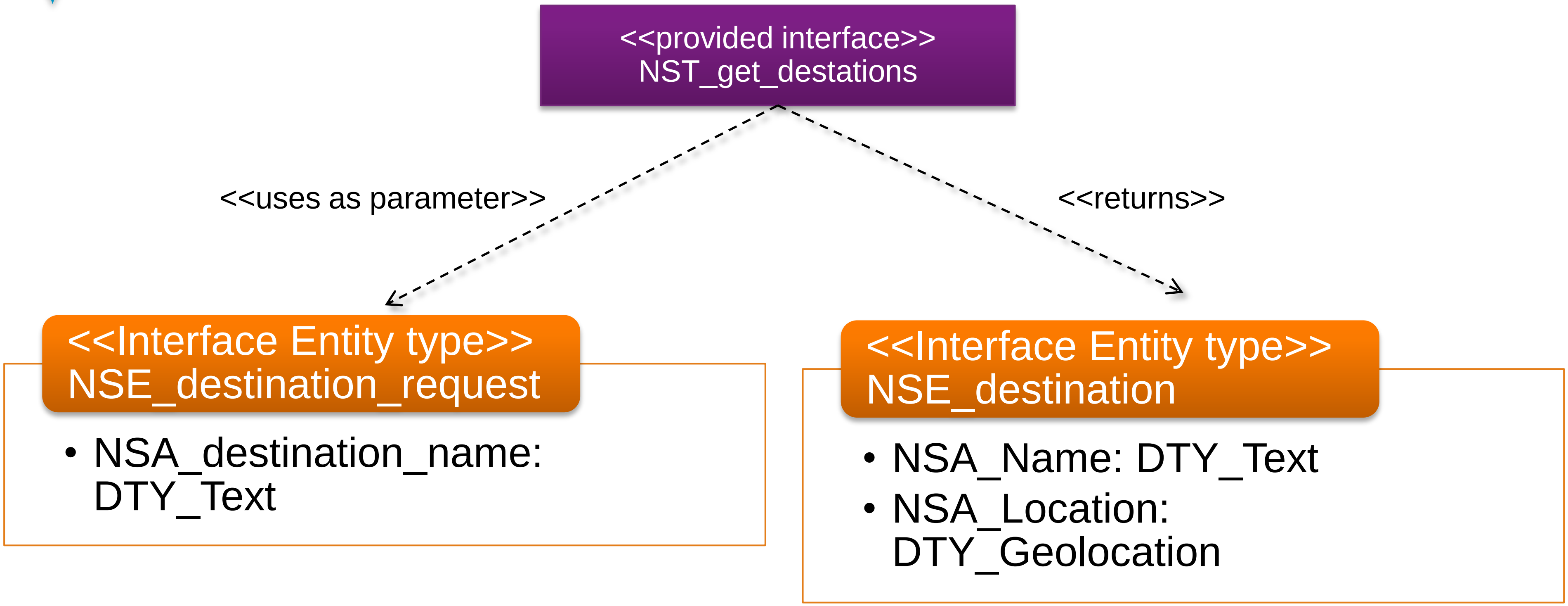
provided interfaces

Described in detail

used interfaces

reference to other system

provided interfaces



<<provided interface>>
NST_get_destations

<<uses as parameter>>

<<returns>>

<<Interface Entity type>>
NSE_destination_request

- NSA_destination_name: DTY_Text

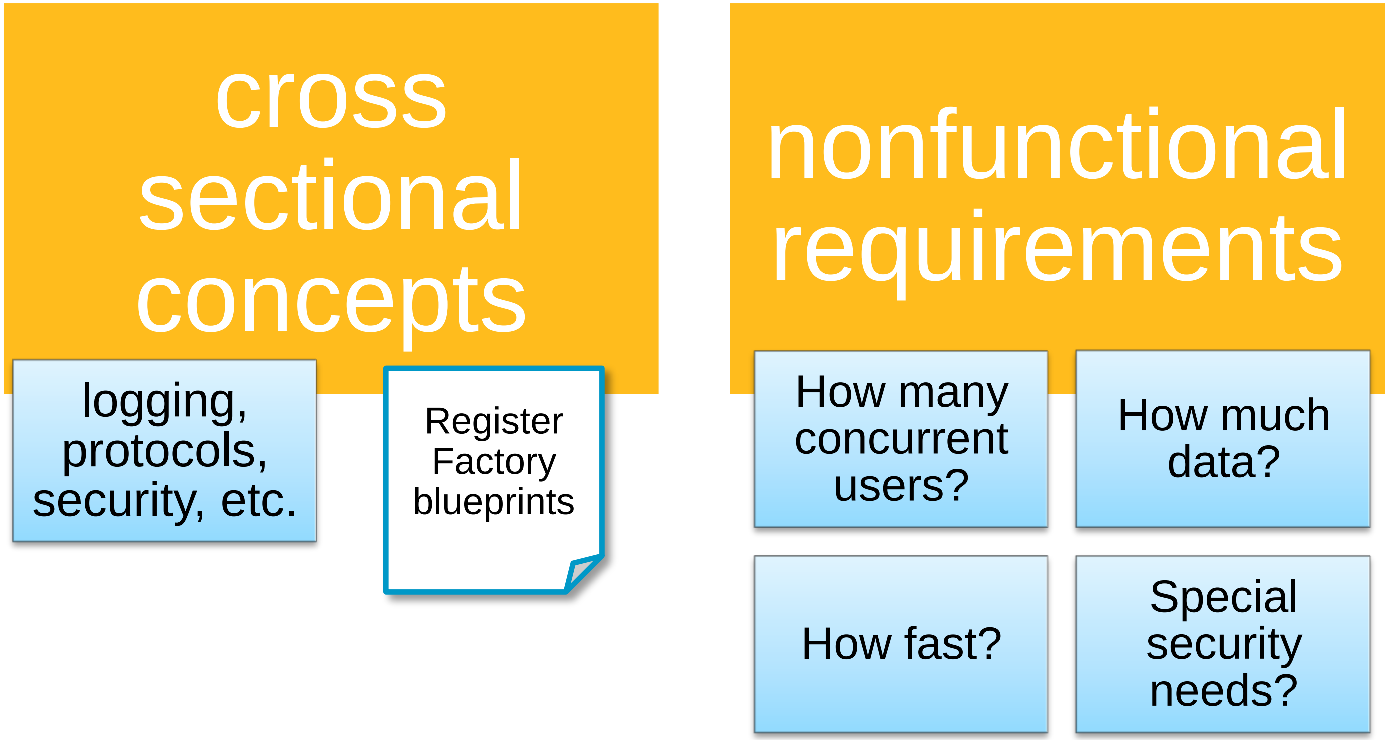
<<Interface Entity type>>
NSE_destination

- NSA_Name: DTY_Text
- NSA_Location: DTY_Geolocation

Plus text description: Which use cases are called depending on given parameters

Typically, entities are returned in a tree structure, designed like data model

The system‘s nonfunctional properties



cross sectional concepts

logging, protocols, security, etc.

Register Factory blueprints

nonfunctional requirements

How many concurrent users?

How much data?

How fast?

Special security needs?

Summary

Choose your weapons

- Depending on project context
- Describe only what you need
- Describe it in the way that works for you and the customer

Divide and conquer

- From system landscape down to application function
- Keep every level straightforward

Track your changes

- From requirements
- To specification
- To implementation
- To test
- To delivery

Questions?

Feel free to ask!

Capgemini

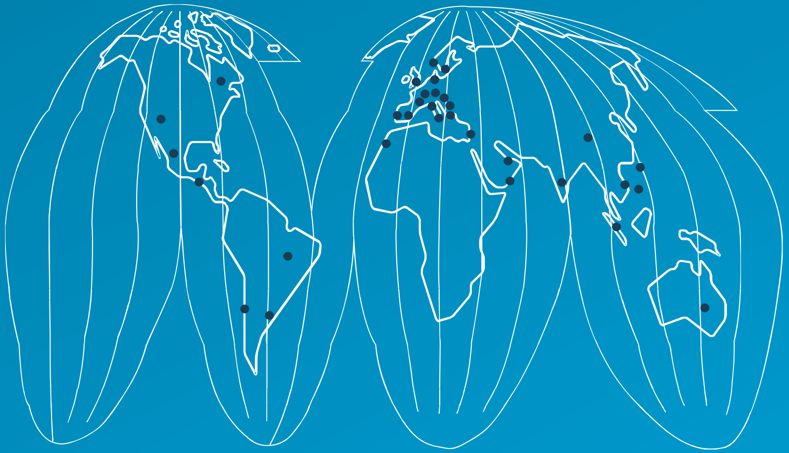
CONSULTING.TECHNOLOGY.OUTSOURCING

People matter, results count.

About Capgemini

With more than 130,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2012 global revenues of EUR 10.3 billion.
Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience^{TM}, and draws on Rightshore®, its worldwide delivery model.

Rightshore® is a trademark belonging to Capgemini



www.capgemini.com

Research and science live on the exchange of ideas, the clear arrangements are thereby useful:

Disclaimer:
Although this presentation and the related results were created carefully and to the best of author’s knowledge, neither Capgemini nor the author will accept any liability for it’s usage.

If you have any questions, please contact:
Capgemini | Offenbach
Dr. Martin Girschick
Berliner Straße 76, 63065 Offenbach, Germany

Telephone +49 69 82901-376
Email: martin.girschick@capgemini.com