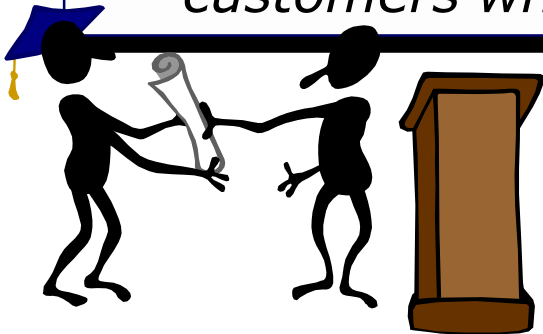
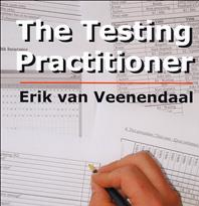


Requirements Engineering for Testers

*Quality makes products
which do not return and
customers who do*



Erik van Veenendaal
www.erikvanveenendaal.nl

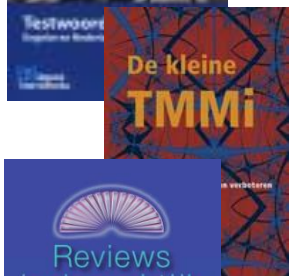
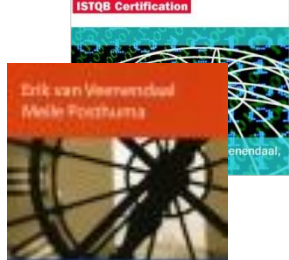
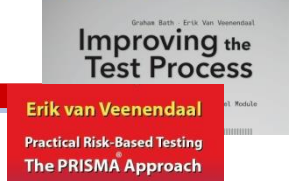
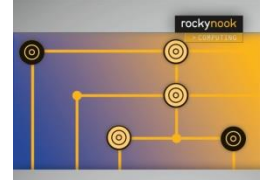
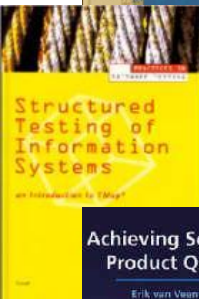
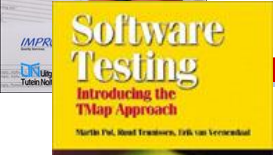


Erik van Veenendaal

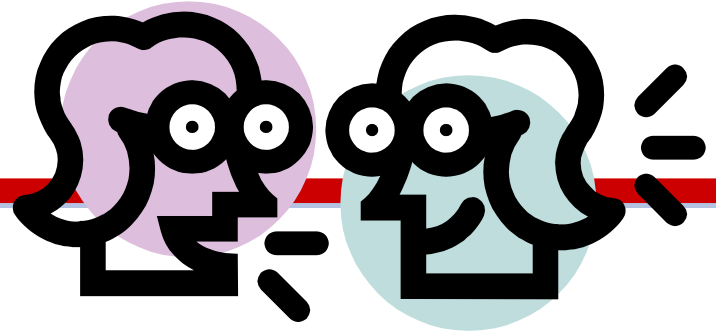


www.erikvanveenendaal.nl

- In IT & testing since 1985 working for many different clients and in many different roles
- Author “TMap”, “ISTQB Foundation” and “TMMi” and many other books and papers
- Former Vice-President International Software Testing Qualifications Board (ISTQB)
- Supporting member IREB board
- CEO TMMi Foundation
- Winner European - & ISTQB International Testing Excellence Award
- Living in Bonaire (Caribbean island)
- Table tennis player and coach



Recognize this?

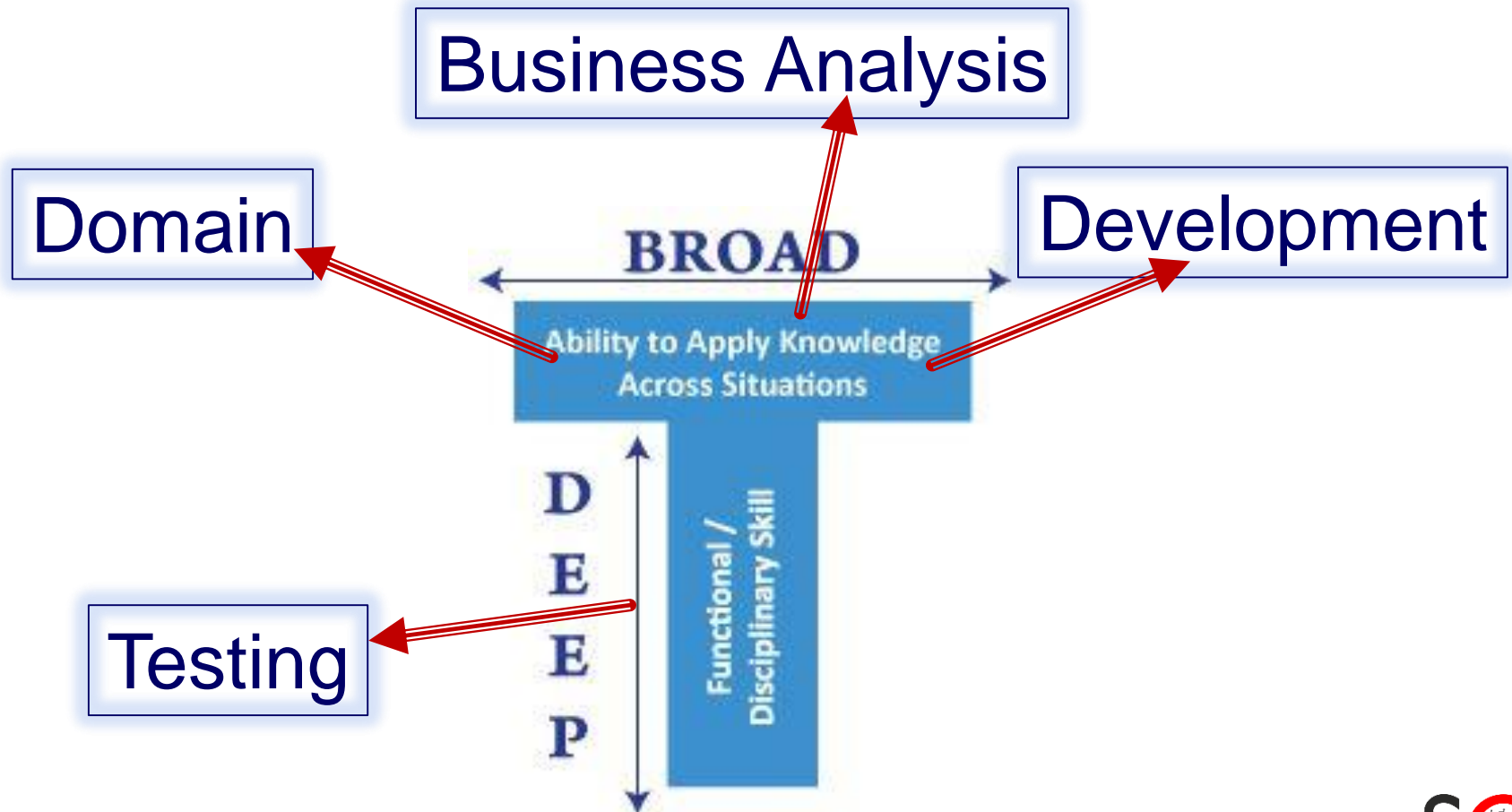


- **Tester:**
“Can’t test this, not clear, not unambiguous”
- Requirements engineer:
“What is a good testable requirement?”
- **Tester:**
“Uuuuhhhh.... SMART” (or INVEST)
- Requirements engineer:
“Let’s define ‘what are the requirements for requirements?’”

However....

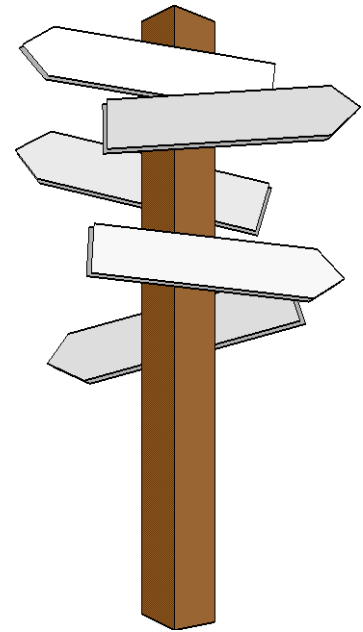
- We are one of the main stakeholders, planning, risk analysis, (test) designs and code are based upon **requirements**
- We are involved in **requirements** reviews, what level of quality is reasonable?
- (Test) designs may even be used as **requirements**
- Sometimes (in Agile) we identify and specify **requirements** and acceptance criteria
- We have a major interest in **requirements** and are heavily involved!

T-Shaped Testers



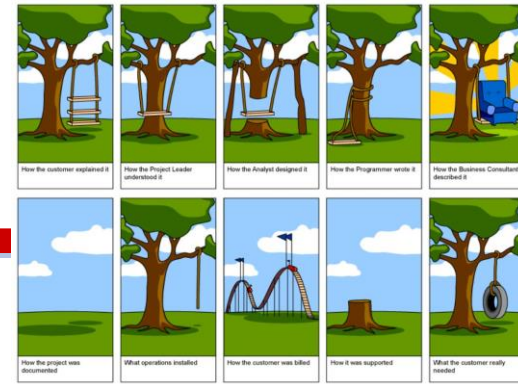
Requirements Engineering for Testers

- 30 min. - Introduction to Requirements Engineering
- 15 min. - What are good requirements?
- 15 min. - Reviewing requirements?
- 30 min. - Good requirements, ideas and options

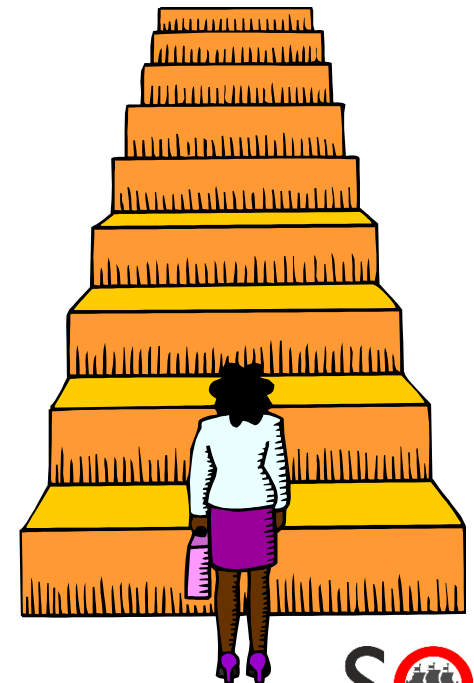


10:30 - 11:10 : Part 1
11:10 - 11:25 : Break
11:25 - 12:20 : Part 2

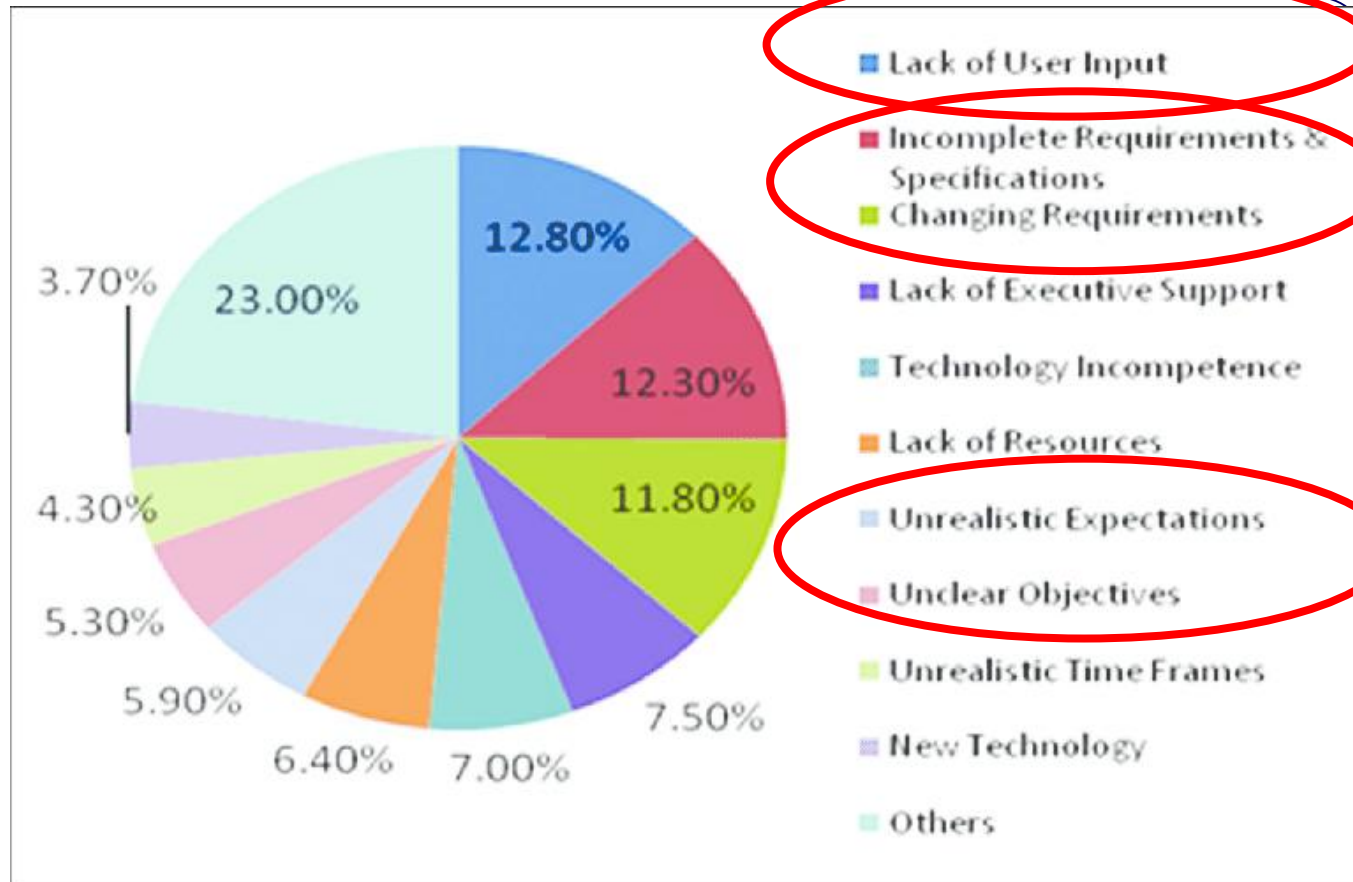
The Challenge



- To capture the need “completely” and “unambiguously” without resorting to specialist jargon, thus understandable by our stakeholders
- Requirements are the basis for:
 - Project (release) planning
 - Trade-off (priority setting)
 - Development
 - Testing

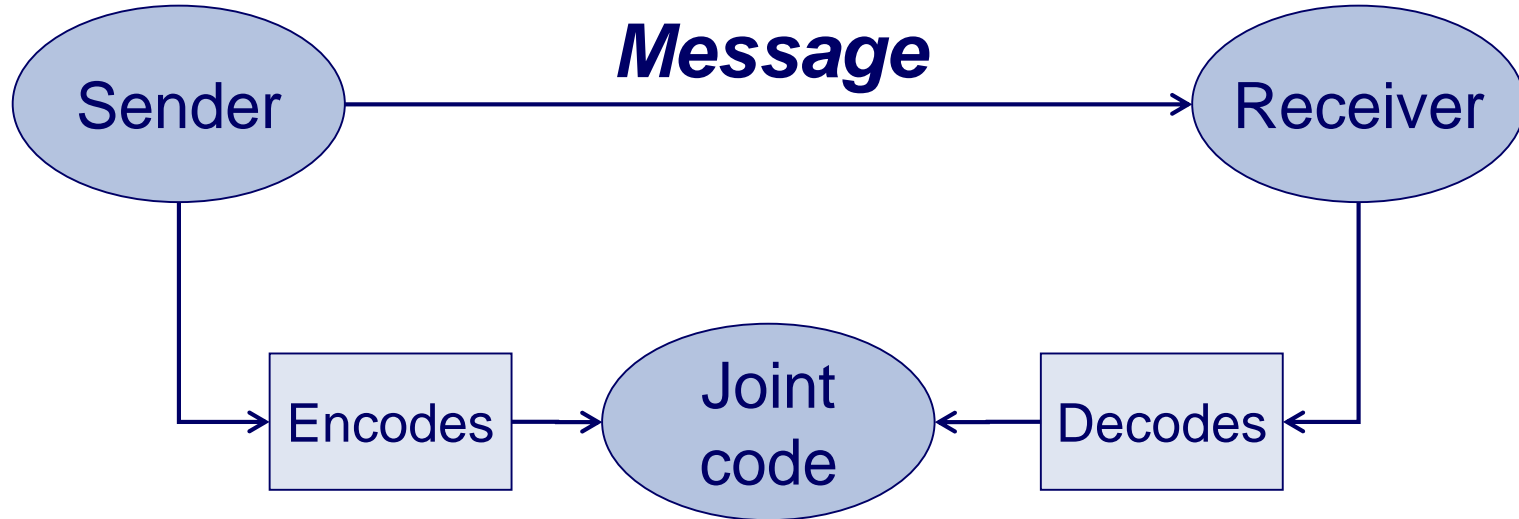


Project Success Factors....



Source: "Extreme Chaos" The Standish Group. www.standishgroup.com

Basic Communication Model



What is a Requirement?

A requirement is a capability or condition to which the system must conform

- ... a capability needed by the user to solve a problem or achieve an objective
- ... a capability that must be met or possessed by a system to satisfy a contract, specification, standard or other formally imposed documentation
- ... a statement of intent that describes something the system needs to do for some user

Three Types of Requirements

- Functional requirements are things the product must do
 - *The product shall produce an updated schedule*
 - *As a <student>, I want <to be able to buy a parking pass> so I can <get to school quickly>*
- Non-functional requirements (e.g., ISO25001) are the properties that the product must have
 - *The product shall determine ... in less than 0.25 seconds*
 - *As a <member of the public> I want <the website to adequately cope with high loads> so I can <purchase a ticket quickly for a highly subscribed event>*
- A constraint is a restriction on the scope or design of the product
 - *The product shall run on the ... platform*

A main principle.....

How much documentation is enough?

- Requirements process is context dependent
 - User requirements – problem domain
 - State *what* the stakeholders want to achieve through use of the system. Avoid reference to any particular solution. “The user shall be able to.....”
 - System requirements – solution domain
 - State abstractly *how* the system will meet the stakeholder requirements. Avoid reference to any particular design. “The product shall
 - Agile / V-model / Outsourcing
 - Business / Project / Product / Human factors

Requirements Process (1)



1. Kick-off phase

- Objective, scope, stakeholders, business case
- Check: Are things clear enough to start?

2. Requirements gathering (quantity-based)

- Functional, Non-functional, Constraints
- Various gathering / elicitation techniques
- Building the backlog

3. Requirements specification (quality-based)

- Templates, rule set, level of detail needed
- Product backlog grooming / refinement

Requirements Process (2)



4. Verification and Validation

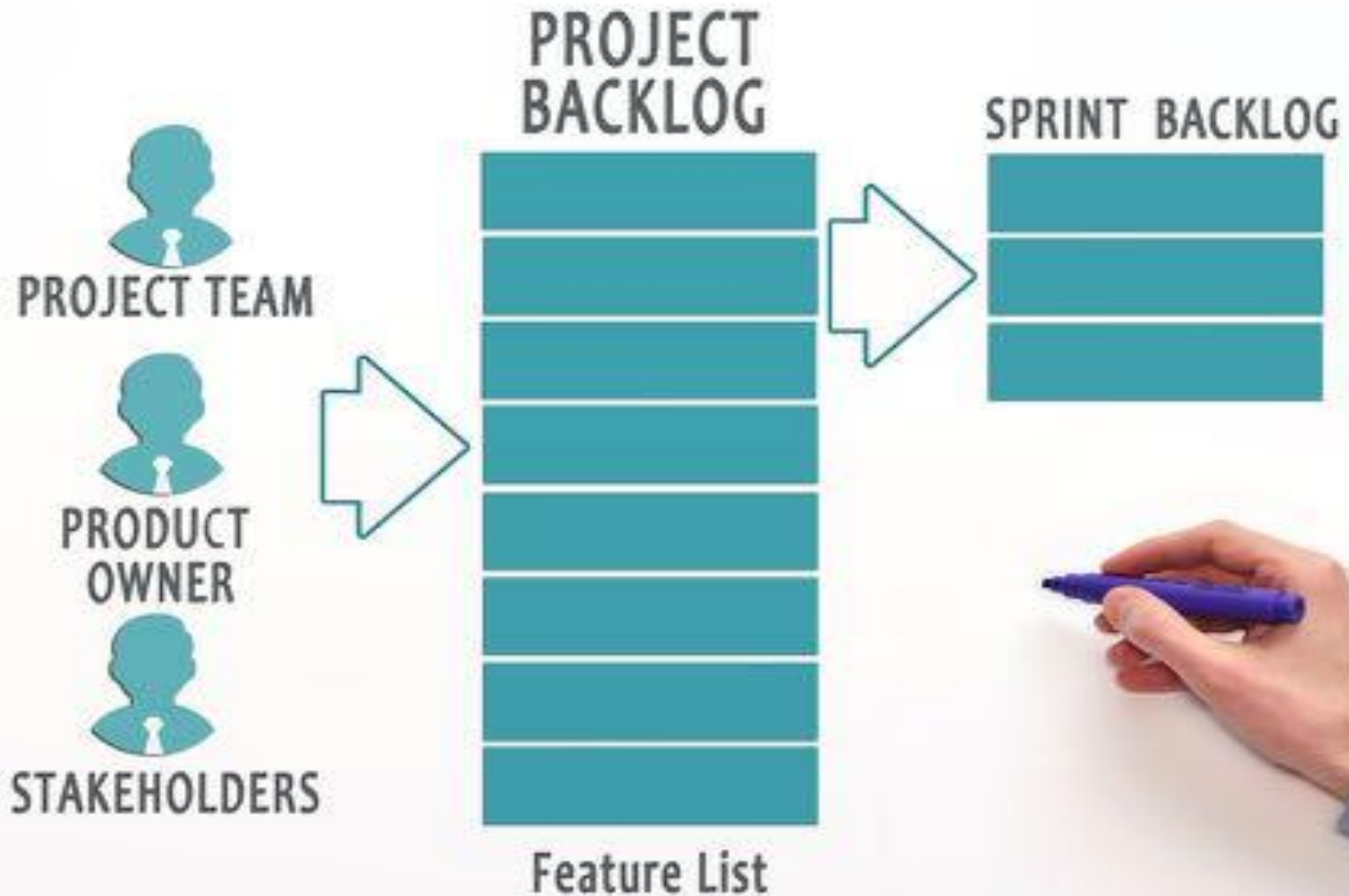
- Checking requirements
- Different types (walkthrough, pair-inspection)
- Use rules and checklists

5. Requirements management

- Identification and traceability
- Use attributes, e.g., source
- Change management
- Relates to the *entire* proces

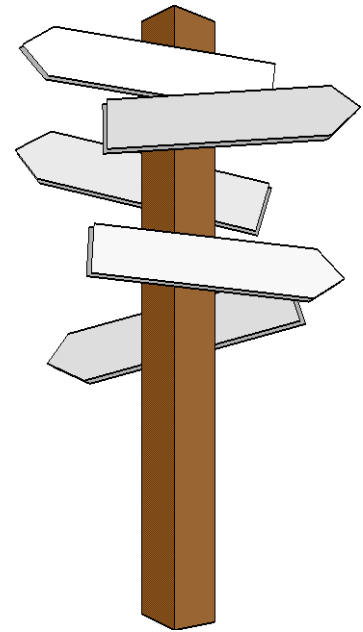
Note, in Agile
this is **not** a
linear process

From Quantity to Quality



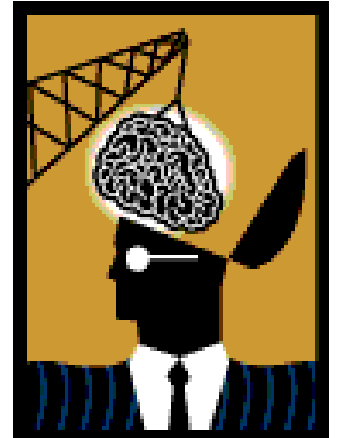
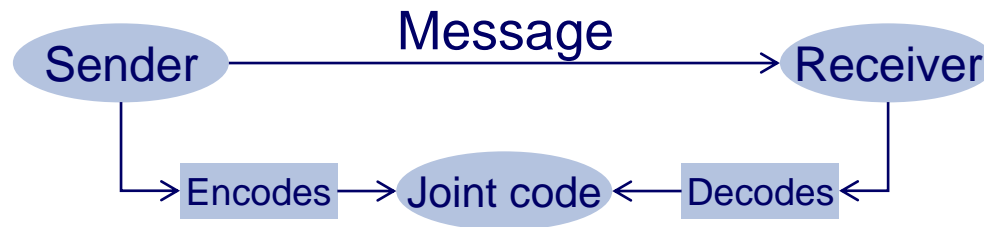
Requirements Engineering for Testers

- 30 min. - Introduction to Requirements Engineering
- **15 min. - What are good requirements?**
- 15 min. - Reviewing requirements?
- 30 min. - Good requirements, ideas and options



What is needed

- If we could look into each others brains, we wouldn't need documentation ...
- Documentation helps us to communicate



- Be careful, words will not be enough!
 - Formal / informal language
 - Different interpretations

"I didn't say he killed his wife"

"*I* didn't say he killed his wife"

"I ***didn't*** say he killed his wife"

"I didn't ***say*** he killed his wife"

"I didn't say ***he*** killed his wife"

"I didn't say he ***killed*** his wife"

"I didn't say he killed ***his*** wife"

"I didn't say he killed his ***wife***"



What are “good” requirements?

Identify at least five “rules” that determine whether a requirement is a good (or “poor”) requirement

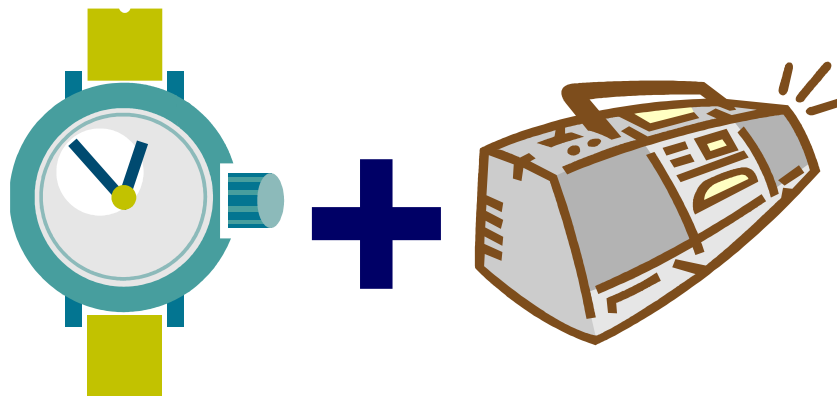


Consider !!

- Individual requirements (the “sentence”)
- Requirements attributes

Excercise Radio Watch (1)

- Study the requirement specification for the new **Radio Watch**
- Make comments (find defects) based on what you have learned so far e.g., attributes, rules,



Requirements Specification - New Radio Watch

The development of the innovative watch and music system shall raise the profit of the company by 10% within 6 months after delivery to the whole sale dealers.

The watch shall have no tic sound. Because people don't like to be waked up, the new radio watch shall wake up sleeping people more smoothly than any other watch.

The radio watch can operate with FM bandwidth only.

The radio watch will provide the user with date and time information.

The radio watch will provide the hours and minutes in digital format.

The watch shall be small and lightweight, because the users want to use it also on travel. Therefore the watch shall be usable with different power voltages. To achieve a minimum weight the watch must not contain any iron parts.

The watch shall be able to wake its environment with music or a wake-up call.

The watch shall be very easy to use. The snooze key shall be easy accessible.

The watch shall be highly reliable. It will resist a heavy knock of a sleeper and a fall from 50 cm.

The radio watch shall be very efficient in energy consumption. The watch system may only be operated with batteries of type 3434.

The radio watch must be able to save settings for wake up time.

Requirements Specification - New Radio Watch

Business objective

The development of the innovative watch and music system shall raise the profit of the company by 10% within 6 months after delivery to the whole sale dealers.

Vague No Req.

No ID's

The watch shall have no tic sound. Because people don't like to be waked up, the new radio watch shall wake up sleeping people more smoothly than any other watch.

shall, will, must, can?

Inconsistent

Ambiguous

The radio watch can operate with FM bandwidth only.

The radio watch will provide the user with date and time information.

Rationale

Compound

The radio watch will provide the hours and minutes in digital format.

The watch shall be small and lightweight, because the users want to use it also on travel. Therefore the watch shall be usable with different power voltages. To achieve a minimum weight the watch must not contain any iron parts.

Design

The watch shall be able to wake its environment with music or a wake-up call.

Vague Not Testable

The watch shall be very easy to use. The snooze key shall be easy accessible.

The watch shall be highly reliable. It will resist a heavy knock of a sleeper and a fall from 50 cm.

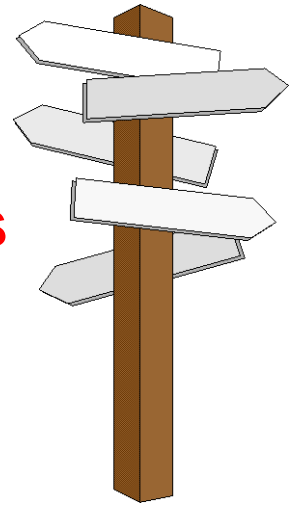
??

The radio watch shall be very efficient in energy consumption. The watch system may only be operated with batteries of type 3434.

Design

Requirements Engineering for Testers

- 30 min. - Introduction to Requirements Engineering
- 15 min. - What are good requirements?
- 15 min. - Reviewing requirements?
- **30 min. - Good requirements, ideas and options**



The purpose of the product

PRINCE-2 “Business Case”
Agile “Vision document”

- The user problem (*no more than 1 page*)
 - A short description of the situation that triggered the development effort
 - Describe the work that should be improved
- Goals of the project
 - What will the product do? (purpose)
 - What is the business advantage?
 - How will you measure the advantage?
 - Statement of needs on a high-level



Get stakeholders commitment on this !!

Purpose Example (A'dam Metro)

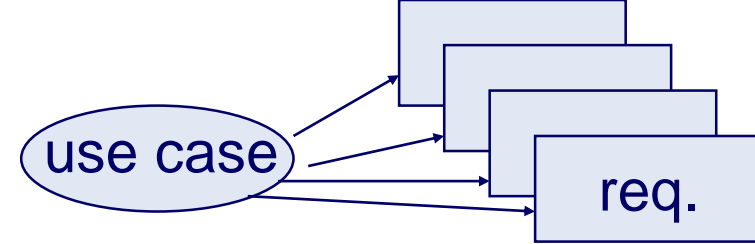
Business Requirements

Purpose: To sell metro tickets more efficiently (faster) than currently.

Rational: To increase sales and reduce cueing while buying metro tickets.

Acceptance Criteria: The product will hand out tickets 30% faster than the current system. This improvement shall be achieved on all priority 1 stations at peak hours.

Use Cases / EPICs

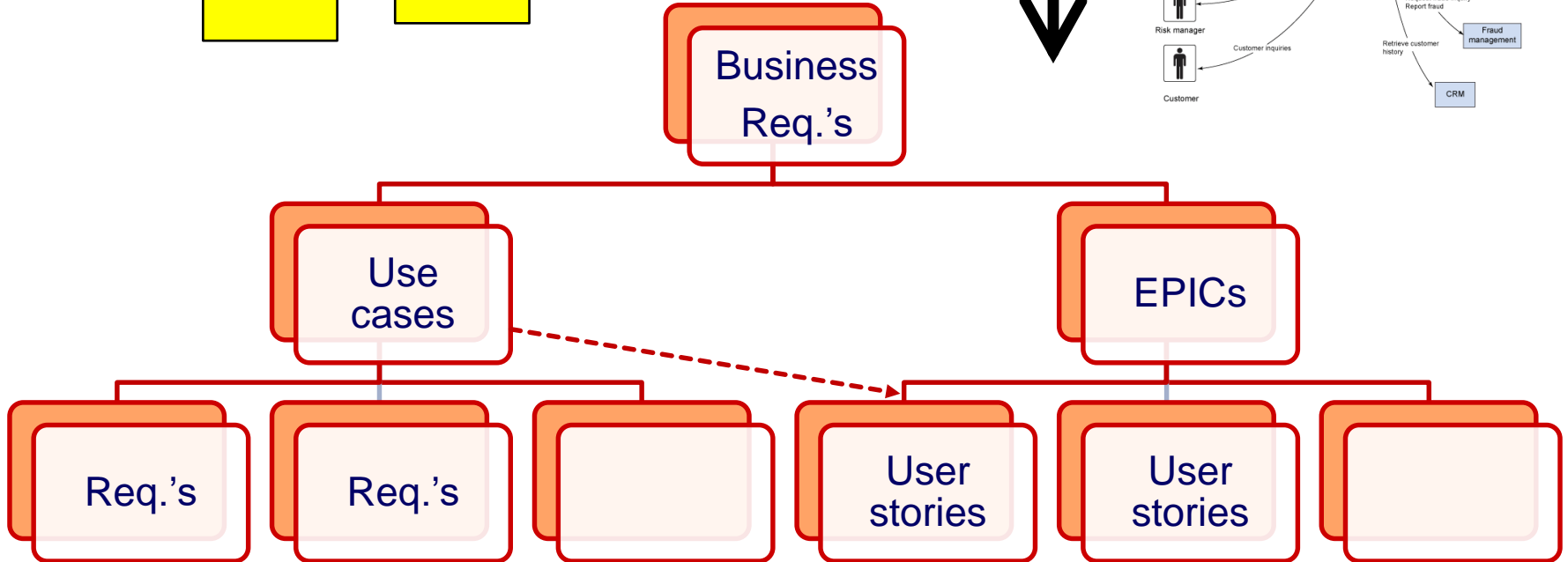
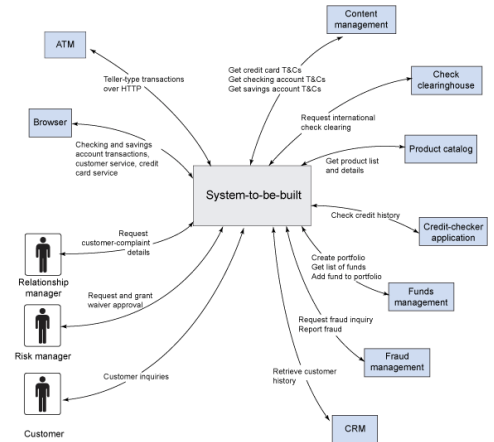
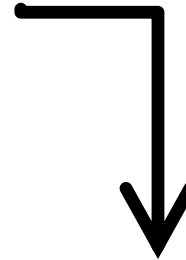
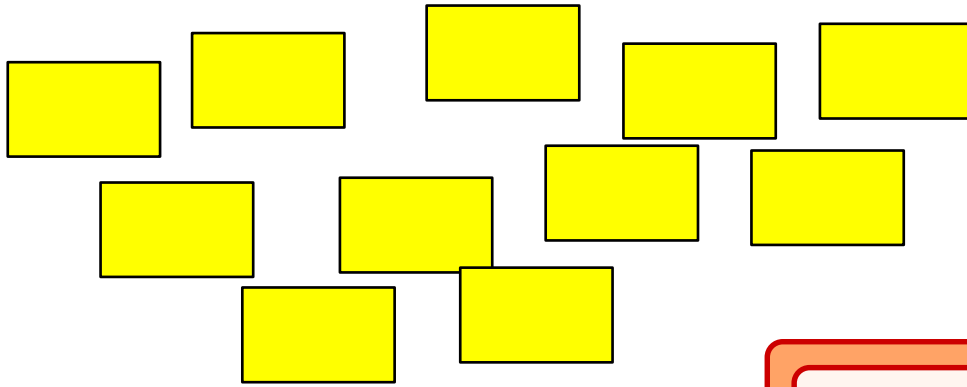


Describing the bigger picture

- Start with the context diagram
- Use cases give users a convenient way to partition the product
- One or more use cases per business event
 - also consider 'misuse cases', e.g., for security req.
- Six step scenario's are a great starting point
 - Name
 - Actor (user)
 - Short description ('happy day scenario')
 - Pre conditions
 - Post conditions



Hierarchy and Traceability



Use Case Example (A'dam Metro)

Use Case: Traveller buying a ticket.

Actor: Traveller

1. The traveller offers destination, type of ticket and payment to the product
2. The product checks whether the payment is ok for the chosen destination and type of ticket
3. The product checks whether the network is operational for the chosen destination.
4. The product submits ticket and if necessary change.
5. The product stores the transaction

Requirements Example (A'dam Metro)

Use Case step 2.: The product checks whether the payment is ok for the chosen destination and type of ticket

Requirements:

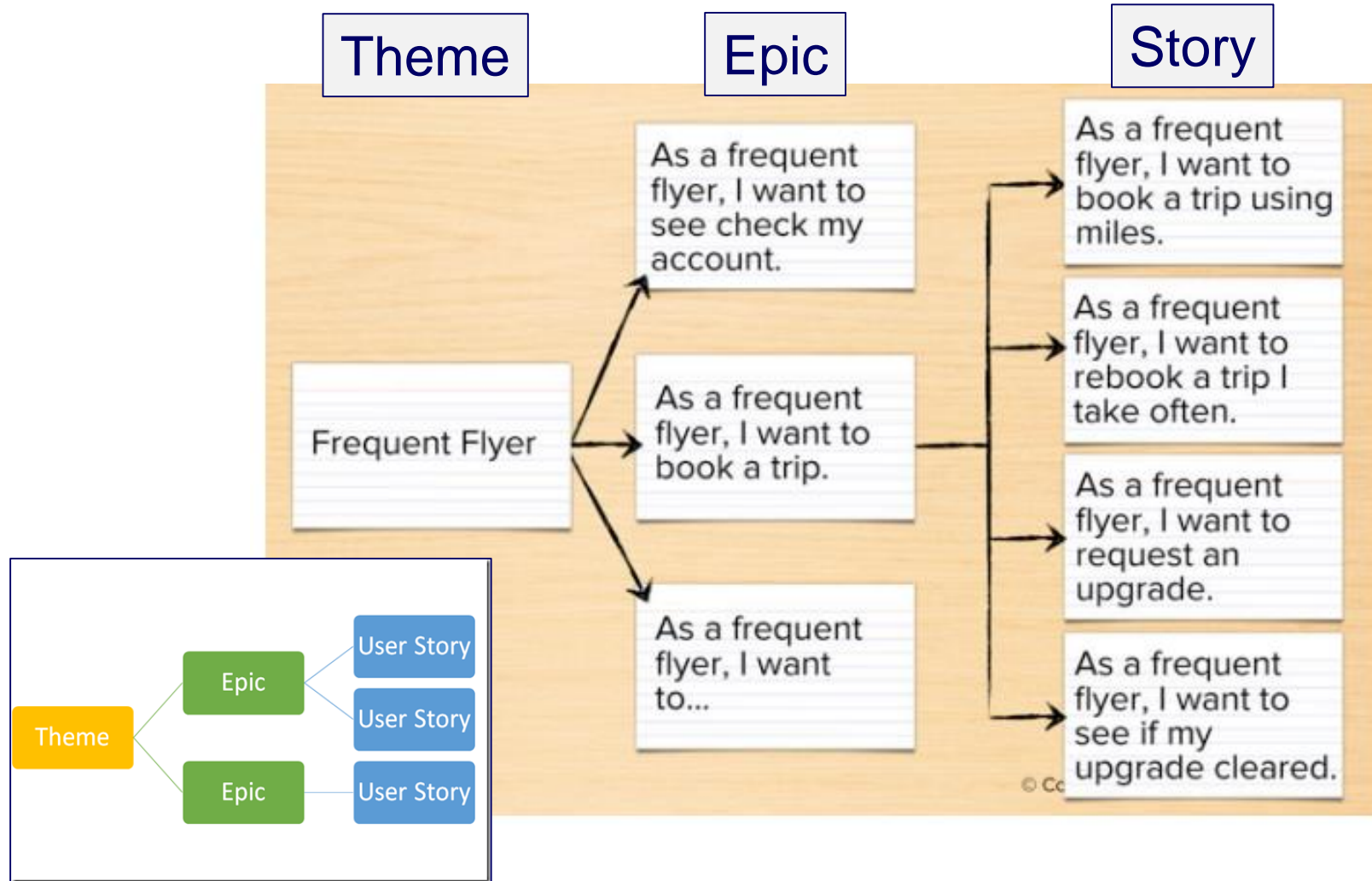
2.1 The product shall establish that the payment consists of legally valid money

2.2 The product shall calculate the lowest fare for the destination considering day of week and time

2.3 The product shall compare the travellers' payment with the calculated payment

2.4 The product shall provide feedback in case the payment is not sufficient.

Example



Requirements cards

Requirement # :

Priority :

Requirement Type :

Use case :

Description :

Rationale :

Source :

Acceptance Criteria :

Size :

Supporting material : annotation, conversation, examples

Requirements attributes (1)

- ID

- To allow traceability

- Requirements Type

- Allows req.'s to be sorted, grouping allows the requirements to be checked on completeness and for conflicts, e.g., by non-functional, by business process

- Use case

- For traceability and change control purposes
- Again for grouping etc.

- Description

- The intent of the requirement (may initially be ambiguous)
- the stakeholders' wishes & needs

Requirements attributes (2)

- **Rationale**

- Reason behind the requirement's existence. Helps to clarify and understand the requirement and to identify 'gold plating' req.'s.

- **Priority**

- Measure of the business value and importance. For negotiation, but also for risk-based testing

- **Acceptance criteria**

- To make the requirement measurable and testable

- **Source**

- Name of the person who raised the requirement , or document.

- **Size**

- Number of story points

Acceptance Criteria

involve tester here

- We have to be able to tell whether a solution completely satisfies, or fits, a requirement
- To make requirements measurable / testable
- To define boundaries
- To reach consensus (with the stakeholder)
- To allow accurate planning & estimation
- In practice very important for non-functional requirements
 - 90% of the customers must be able to get the correct ticket from the product in no more than 25 seconds



Example Acceptance Criteria

Requirement / User Story

- As a student, I want to be able to buy a parking pass so I can get to school quickly

Acceptance Criteria (5 to 8)

- The student will not receive the parking pass if the payment is insufficient
- One can only buy a parking pass to the school parking lot if the person is a registered student
- The student can only buy one parking pass each month
-

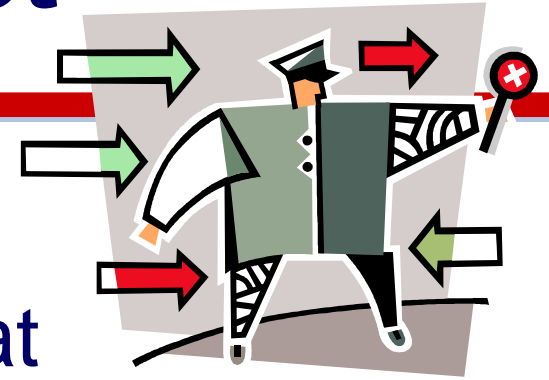
given <context> *when* <the event occurs>
then <expected outcome>

Use Templates

- *The <stakeholder> shall be able to <capability>*
 - The order clerk shall be able to raise an invoice
- *As a <role>, I want <activity> so that <business value>*
 - As a job seeker, I want to search for a job, so I can advance my career
- *The <product> shall be able to <action> <entity>*
 - The launcher shall be able to launch missiles
- *The <product> shall <function> <object> every <performance> <unit>*
 - The coffee machine shall produce a hot drink every 10 seconds

Requirements Rule Set

- Usefull set of agreements
- Specify the contents and format of a requirement (and requirements document)
- More objective, less discussion
- Applied during specification and reviewing
- Organization specific
- Rules for tracing, format and content



Examples of Rules (1)

- Identification
- Valuable / purpose
- Changes
- Grouping
- Uniqueness
- Consistency
- Annotation
- Language
- No design
- Knowledge responsible (source)

All forms of annotation, comments, notes, suggestions, examples, or other items not part of the formal requirement shall be clearly indicated as such. This will be documented by using the attribute 'additional information'.

Examples of Rules (2)

- Detail
- Brief / Small
- Unambiguous
- Priority
- Rationale
- Compound
- Independent
- Technically achievable
- Testable

Req.'s shall be unambiguous to the intended readership. Req.'s shall have only one interpretation. For example the word shall is used and not the word should. Words like can shall only be used when more than one option is available. Directive language (active voice) shall be used, e.g., specifies and not can specify.

INVEST

Things to do consider tomorrow

1. Start with a purpose statement & context diagram
2. Introduce requirements attributes (cards)
3. Use acceptance criteria to fine-tune req.'s
4. Use templates
5. Define and use requirements rules
6. First quantity, than quality
7. Spread the word; share the knowledge
– whole team approach





www.erikvanveenendaal.nl

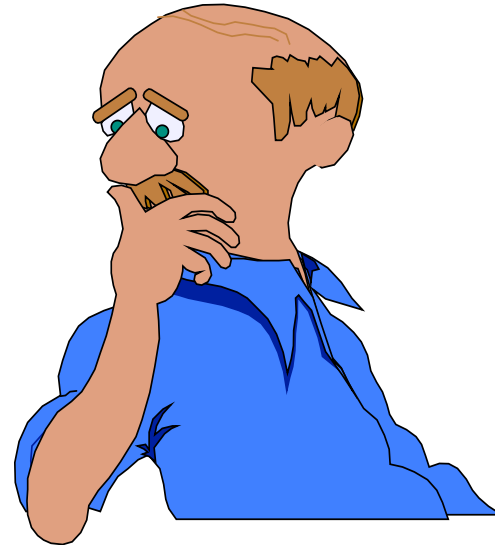


erik@erikvanveenendaal.nl



@ErikVveenendaal

Any questions.....?



Thank you !!

