Tool-Guided, Domain-Specific, Systematic Requirements Management

R.Gerlich, R.Gerlich (BSSE)

DASIA'10

01.-04.06.2009, Budapest, Hungary

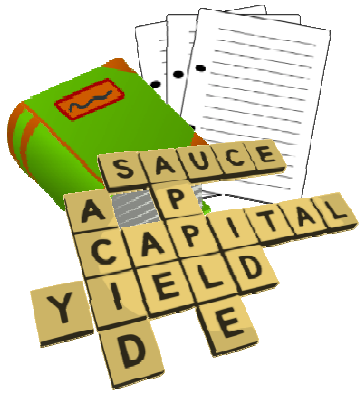
Dr. Rainer Gerlich BSSE Systen and Software Engineering
Auf dem Ruhbühl 181
88090 Immenstaad
Germany

Tel. +49/7545/91.12.58
Fax +49/7545/91.12.40
Mobil +49/171/80.20.659
email Rainer.Gerlich@bsse.biz

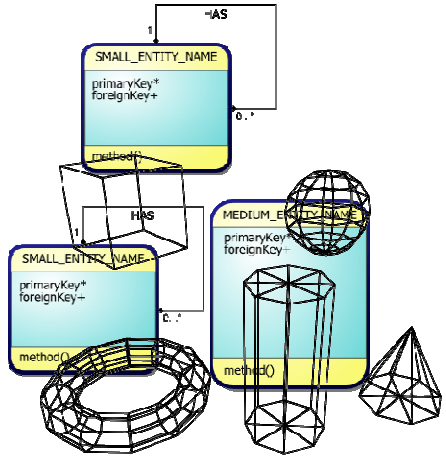
Overview

- Status
- Systematic Requirements Management
- Applications
- Conclusions

Text-based vs. Modelling Approaches



Text-based Approach



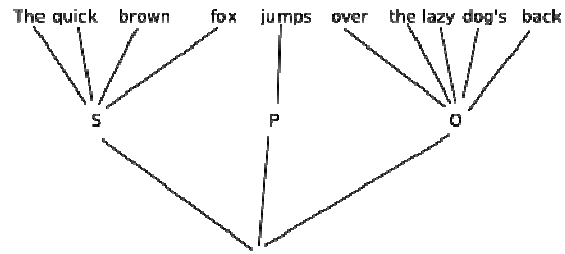
Model-based Approach

- Imprecise requirements (natural language)
- Tools only provide containers for requirements (free-text, black-box, no semantics)
- Manual management of dependencies and links
- Harmonisation requires discussion in the team
- Overhead due to manual maintenance
- No support for verification

- Formalised (i.e. defined form)
- Semantics = Model + Meta-Model
- Better support for verification and validation
- Non-specific: UML, SysML, ... (“universal”)
- domain-specific: more implicit information about domain

Improving (?) Text-based Specifications

Glossary
Knowledge Database (HOOD)



Limited Grammar (SOPHIST)

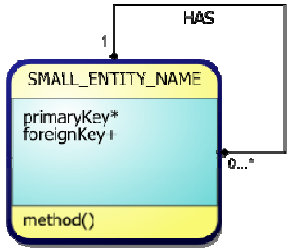
Interpretation, harmonisation, verification, validation and maintenance still manual labor

Administrative, contractual and planning aspects not supported

Universal Modelling Approaches

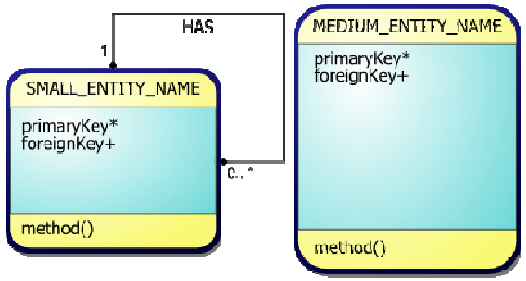
UML

- formal semantics only to a small degree; ambiguous
- ambiguity is part of the strategy (universality)
- major parts still as text
- no support of non-functional requirements



SysML

- formal semantics only to a small degree
- no support of non-functional requirements
- central element is text-based requirements
- formal requirements (e.g. OCL) have decidability issues



No information about domain

No inherent support for verification and validation

Administrative, contractual and planning aspects not supported

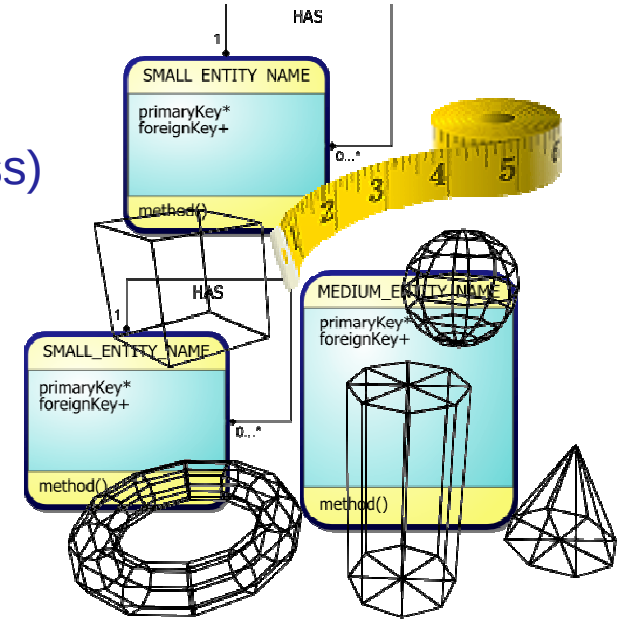
Systematic, Domain-Specific Approaches

Domain-Specific

- information about domain encoded in meta-model / semantics
- Still infinite set of applications
- covering all relevant requirement types of domain
- dependencies, links established / identified automatically
- verification by tool possible (completeness, consistency, correctness)

Systematic

- verification by tool implemented
- support validation by graphical / numerical feedback
- quality measurement by domain-specific metrics
- auto-tracking
- automatic test-case derivation
- exploration of problem / solution space (what-if, version comparison)
- connection to planning, software engineering, management, ...
- guide the user; “slap on the wrist”



What the Title Means

domain-specific

- The application domain is known, e.g. distributed and/or real-time systems, or communicating processes
- The number of supported applications out of this domain is infinite

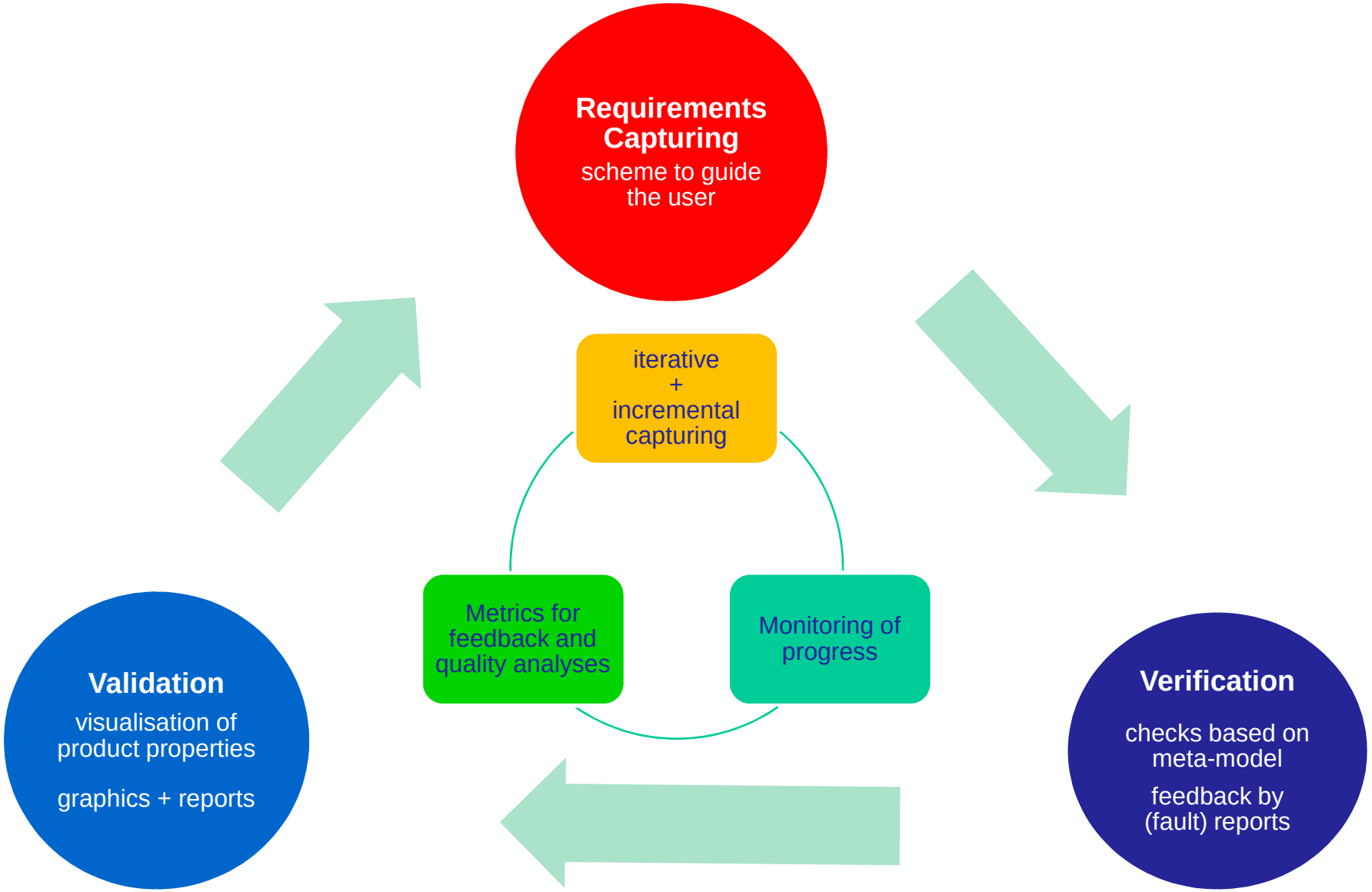
systematic

- All issues of requirements management are supported inherently and silently by the method and the related tool:
  - Requirements engineering: structuring, analysis, elicitation, verification and validation
  - Administration: linking and tracing
  - Organisation (multi-team, multi-site)
  - Issues of project management

tool-guided

- All requirements are correlated by rules and an underlying meta-model
- The method and tool can assess the quality of the requirements by concrete figures based on metrics applied to the requirements
- These figures are provided as direct feedback to the user
- The user is guided by this feedback towards a quality goal as specified by the meta-model

The Principal Approach for Systematic Requirements Management (SRM)



Some Domains

Communicating Processes

- Dynamic view on a system
- Any system operations based on message exchange
- Systematic Requirements Management (SRM)
- Specification of system operations (complete coverage of all requirement types)
- Distributed systems
- Client-server systems
- Checks based on meta-model
- Auto-reporting
- Correlation with project management issues
- “at a touch“

Project Management

- Systematic Project Management (SPM)
- Specification of work packages, personell, resources, effort, inputs/outputs, due dates, cost rates etc.
- Check of dependencies of work packages:
- via explicit dependencies
- Implicitly, via dependencies from input/putput coupling
- Checks based on meta-model
- Auto-reporting
- Bridge to MS-Project®
- “at a touch“

Distributed Real-Time Systems

- Executable specification (model-based)
- Support of functional, behavioural and non-functional requirements
- Verification on modelling level
- Auto-coding
- Model-based Testing
- Auto-reporting
- Validation support
- “at a touch“

Some Projects

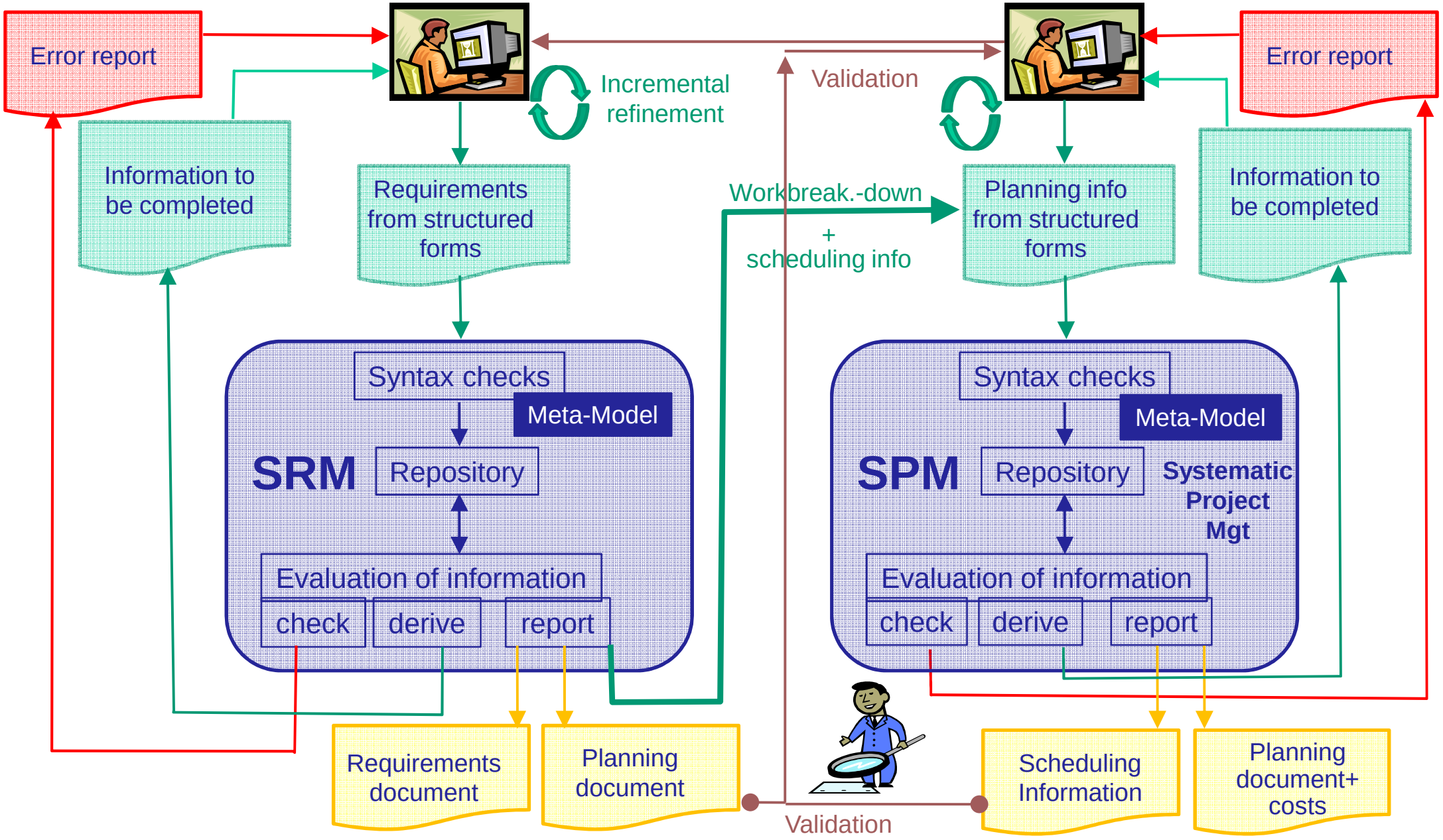
Product Lifecycle Management (PLM)	Shop	Bank Transfer	Quality Analysis	Project Management	Distributed Real-Time Systems
• Configuration of a product by an end-user • Specification of the product definition assembly • Specification of the product configuration assembly • Specification of interaction with ERP (Enterprise Resource Planning), CRM (Customer Relation Management) etc.	• Shop-portal for end-user • Specification of all activities from login, configuration, procurement, invoicong, delivery to payment	• Execution of a bank transfer • Specification of all activities from login, definition of the transfer elements, check of credit-wothiness to execution of transfer	• Analysis of several already existing specifications, established with MS-Word® or UML • The efficient SRM approach allowed to transfer the requirements at low costs within a short time period • Resulted in “poor quality“ conclusions • Neither the applied (universal) tools nor users could identify incompleteness, inconsistency and incorrectness of the requirements from the chosen (universal) notation	• Definition of all planning, management and cost elements as needed for a proposal • For several projects • Checks on feasibility of planning	• Experiment on-board ISS • In operation

Efficiency Figures (RE)

Example	Tool	# RQs	Effort* / m-h	Efficiency / (RQ / m-h)	
PLM (Analysis)	Word	1000	14000	0.07	**poor quality not completed**
PLM (Analysis)	UML	~ 5600	200000	~ 0.028	**poor quality not completed**
PLM (Specification) Product Lifecycle Management	SRM	1000	1000	1	
Shop (Specification)	SRM	300	100	3	
Bank Transfer (Specification)	SRM	400	50	8	
Embedded (Executable Specification)	SRM	5000	1000	5	

* effort roughly estimated, figures indicate a trend

Logic Flow of Operation



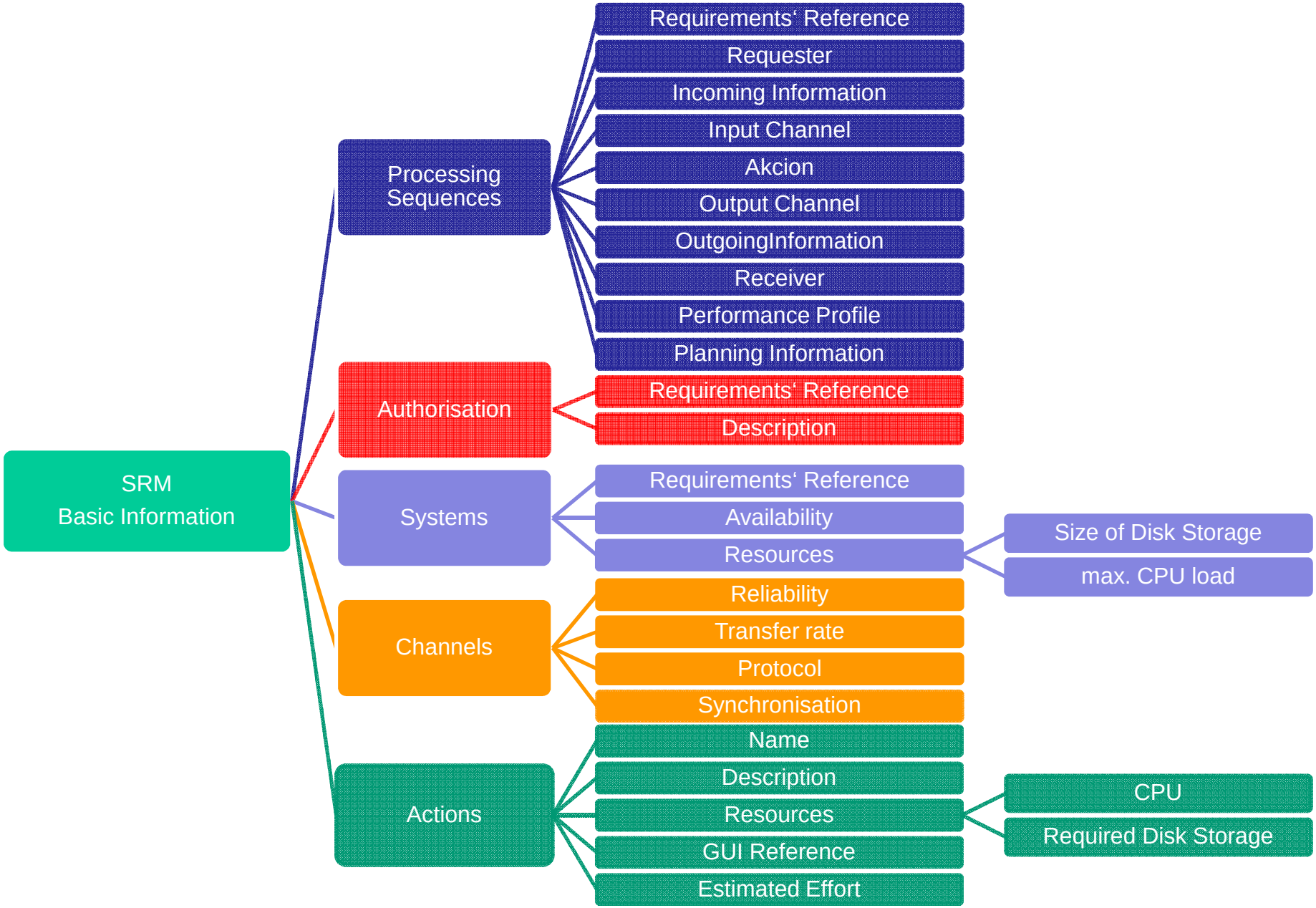
Derived Information: Performance Predictions

System	max. CPU Utilisation (%)	CPU Time ms/d	Acutal Utilisation (%)
CRM	80	36028813	41,7
ERP	80	54259275	62,8
Shop	80	31536491	36.5
Store	80	74217643	85.9

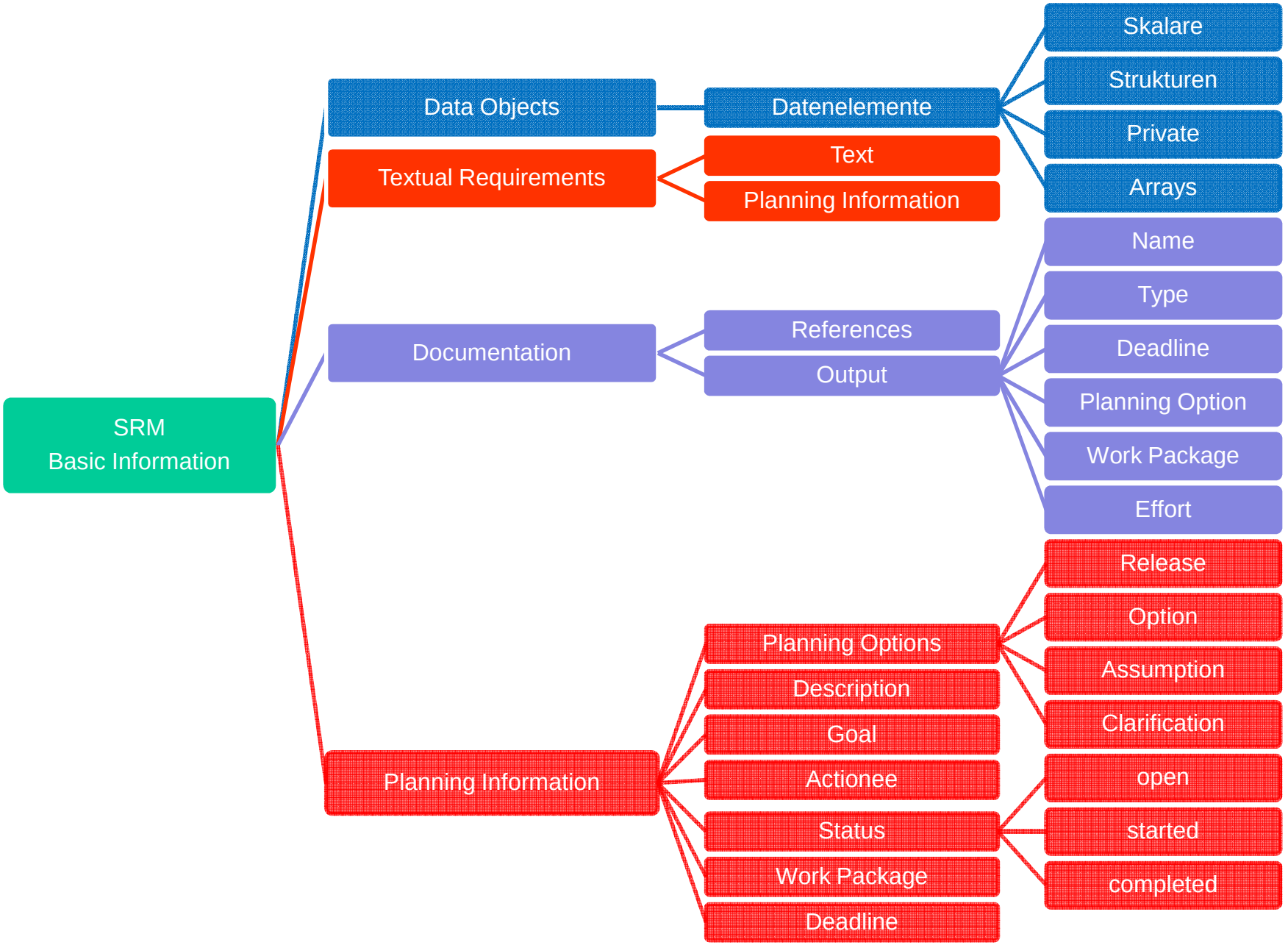
System	Available Mass Memory (MB)	Needed Amount (MB/d)
CRM	100000	23.50
ERP	100000	7.80
Shop	100000	3.60
Store	100000	14.70

Action	Effort (man-hours)	CPU Time (ms)	# Executions / d	Consumed CPU Time (ms)
accept registration	15	1	1000	1000
check registration data	10	4	1000	4000
committment period exceeded	5	1	10	10
committment period valid	5	1	500	500
compile list of ordered products	20	5	100	500
initiate monitoring of delivery	10	1	150	150
load data of registered user	10	2	400	800
reject registration	5	1	10	10
send confirmation of purchase order to user	15	1	150	150
validity check	5	3	510	1530

Requirement Types on Operational Systems – SRM 1/2

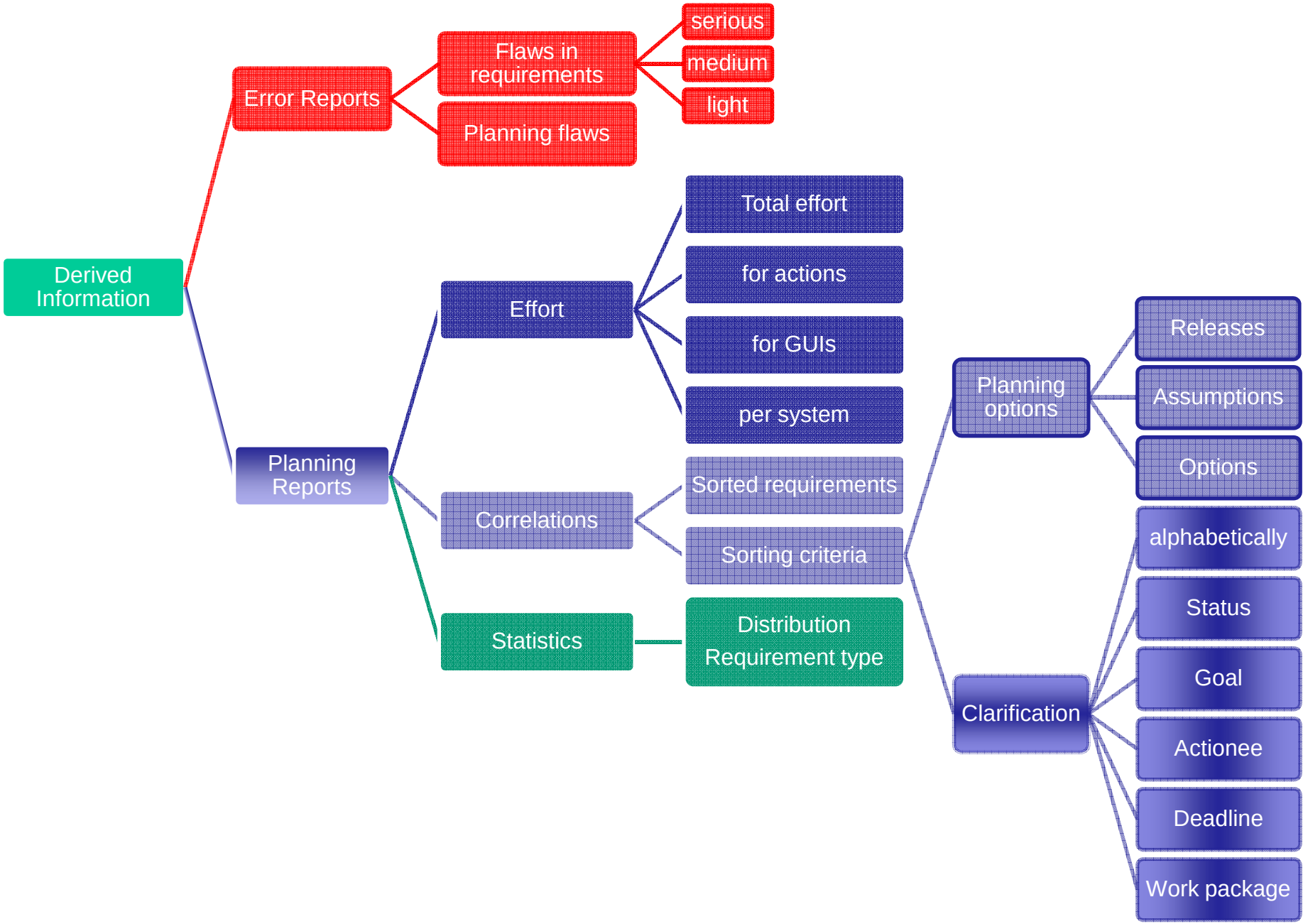


Requirement Types on Operational Systems – SRM 2/2



SRM – Derived Information 1/2

SRM – Derived Information 2/2

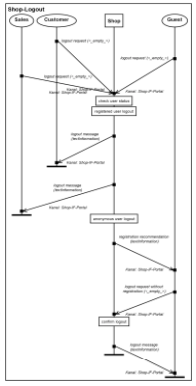


What-If and Comparison of Versions

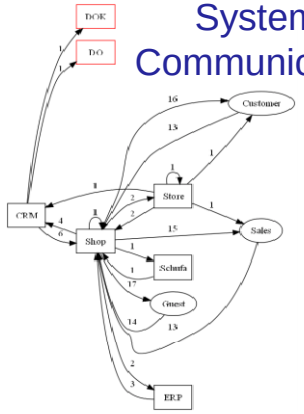
Qualification vs. Activity

Requirement-ID	Kommunikation
VVdemoANF-208	identification request
VVdemo-ANF-154	no delivery possible
VVdemo-ANF-156	price of configured product
VVdemo-ANF-180	update configuration
VVdemo-ANF-177	shopping basket restored
VVdemo-ANF-184	orderRejected

Actions vs. systems vs. Channels vs. Data



System-Communication



Doc vs. RQ

DOK-ID	Dokument	Typ	Betroffene Requirements
ApplDoc-Config	RD Config Procedure	doc	Shop-PA-41, Shop-PA-34, …
ApplDoc-Login	RD Login Procedure	doc	Shop-PA-9, Shop-PA-11, …
ApplDoc-Logout	RD Logout Procedure	doc	Shop-PA-16, Shop-PA-17, …
ApplDoc-Purchase	RD Purchase Order Procedure	doc	Shop-PA-56, Shop-PA-53, …
ApplDoc-System	RD Systems	doc	VVdemo-SYS-66, VVdemo-SYS-67

Requirement-ID	AP	Aktion	Termin	MA	Status
VVdemo-GR-322	WP1.1	Provide contents	31.01.00	RGs	offen
VVdemo-GR-332	WP1.1	clarify 2	15.01.00	TF	in_Arbeit
VVdemo-GR-333	WP1.1	clarify 3	31.01.00	WW	erledigt

Structured operational RQs
- sequences
- systems
- channels
- actions
- data objects
- communication

Correlation with Planning
- options
- releases
- clarification

SRM / SPM Repository + Meta-model

Textual Requirements

Derived information

Beschreibung	Bezug	Planungs-bezug
User specific configurations	Shop	
shall be preserved after the end of a session.	Shop	Release 1
shall be restored at the beginning of the next session.	Shop	Release 2
Following standards shall be applied:	Shop	
enabled entries in black color	Shop	Alle
disabled entries in gray color	Shop	Alle

Requirement-ID	Beschr	Ort	Rel	Klärung
VV-GR-329	GR8	c_GR.xls, Z16	2	Release 2
VV-GR-330	GR9	c_GR.xls, Z17	3	Release 3
VV-GR-331	GR10	c_GR.xls, Z18	4	Release 4

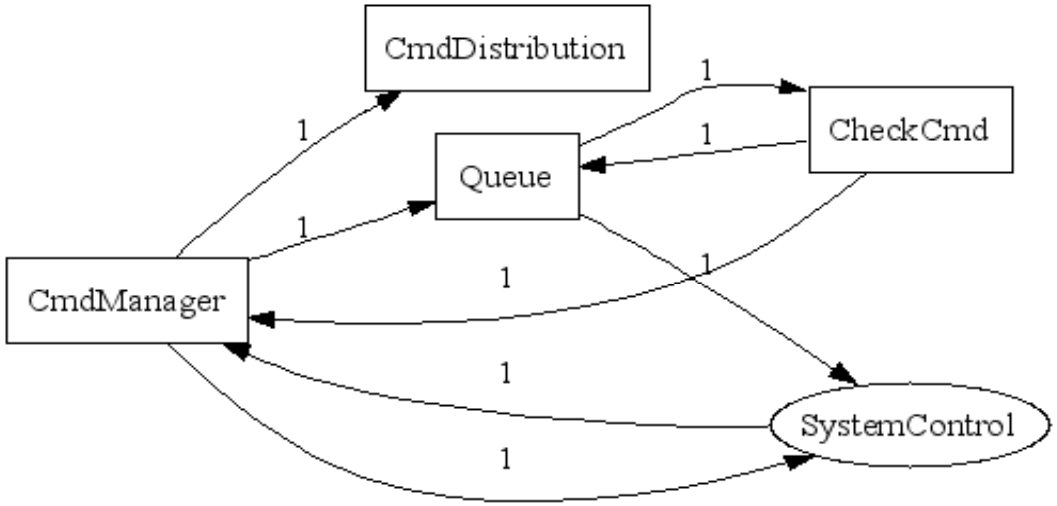
Kanal	Daten-rate (MB/s)	#Pak/d	Volumen MB/d
Shop-IF-CRM	10	2000	49.3
Shop-IF-ERP	10	3000	78.5
Shop-IF-Extern	10	1000	17.6
Shop-IF-Intern	10	5000	234.8
Shop-IF-Portal	10	11000	443.9
Shop-IF-Store	10	4000	356.2

Requirement-ID	System	abgeleitet aus
VVdemoProject-SYS-66.syscom1	Shop	Shop-PA-56, Shop-PA-11, Shop-PA-53, …
VVdemoProject-SYS-66.syscom2	Store	Shop-PA-59, Shop-PA-60, Shop-PA-61, …

Testfall	System	Ablauf	Daten	Ergebnisse	Bezug zu RQ
committment check	CRM	Shop-PurchaseOrder	keine	purchase order rejected, purchase order accepted	Shop-PA-53, Shop-PA-54
rejected configuration	Shop	Shop-Configuration	commentedConfiguration	update configuration	Shop-PA-31
finalisePurchaseOrder	Shop	Shop-PurchaseOrder	keine	committment check	Shop-PA-51

Requirements Capturing

Element Type	Occurrence
Authorisations	1
Process Sequences	1
Documents	1
Non-functional Requirements	1
Actions	11
Processings Steps	10
Systems	5
Channels	1
Data Objects	14
TOTAL	59



Input notation may be adapted !

#	Requirement-ID	Sender	Incoming Request	Channel	Actionee	Action	Outgoing Request	Receiver	Channel
1	CmdDemo-PA-2	SystemControl	ExternalCmd	STD	CmdManager	Verify command	insertCmd	Queue	STD
2	CmdDemo-PA-3	CmdManager	insertCmd	STD	Queue	sendACK	inqueue	SystemControl	STD
3	CmdDemo-PA-4					storeCmdInBuffer	NOP	NOP	NOP
4	CmdDemo-PA-5					validate	validateCmd	CheckCmd	STD
5	CmdDemo-PA-6	Queue	validateCmd	STD	CheckCmd	verifyCmd	checkResult	CmdManager	STD
6	CmdDemo-PA-7					NOP	getNextCmd	Queue	STD
7	CmdDemo-PA-8	CheckCmd	getNextCmd	STD	Queue	checkOnCmdLoss	AL(lost) validateCmd	CheckCmd	STD
8	CmdDemo-PA-9	CheckCmd	checkResult	STD	CmdManager	distribute	AL(isValid) ValidCmd	CmdDistribution	STD
9	CmdDemo-PA-10					NOP	AL(isInvalid) NAK	SystemControl	STD
10	CmdDemo-PA-11	CmdManager	ValidCmd	STD	CmdDistribution	distributeValidCmd	NOP	NOP	NOP

Conclusions

Universal vs. specific

- Universal approaches give poor support to users, cannot conclude on the quality, and increase the overhead
- Cause a lot of discussions on harmonisation of requirements coming from different teams, possibly on multiple sites
- Limitation to a specific domain allows to make use of information for optimisation
- Systematic organisation allows to guide a user towards high quality of specification and planning
- Domain-specific approaches can support a large number of individual applications

Efficiency

- Systematic organisation saves a lot of effort due to synergies enabled by the meta-model
- Due to the inherent, integrated capabilities on quality assessments systematic approaches avoid a lot of human intervention
- Due to the significantly reduced effort a user can concentrate on application issues rather than on maintaining links manually and looking for completeness, consistency and correctness of requirements

Verification and Validation

- The underlying meta-model defines inherently the outmost quality goal for any application of the chosen domain
- When the tool cannot identify an error anymore, the quality goal is reached
- Due to the feedback from the tool a user can validate the requirements easily and immediately