Requirements Engineering im Überblick

Paul Grünbacher

The Challenge: Avoiding Requirements Mismatches

Stakeholder:
Ausgangspunkt jeder Anforderungsermittlung

- Wichtige Stakeholder
  - Kunden
  - Benutzer
  - Entwickler
- Beispiele
  - Content-Verantwortliche
  - Domänenexperten
  - Fachleute für Marketing
  - Usability-Experten
  - Verantwortliche für Markt- und Zielgruppenanalysen
  - …

Beispiele für Stakeholder-Ziele

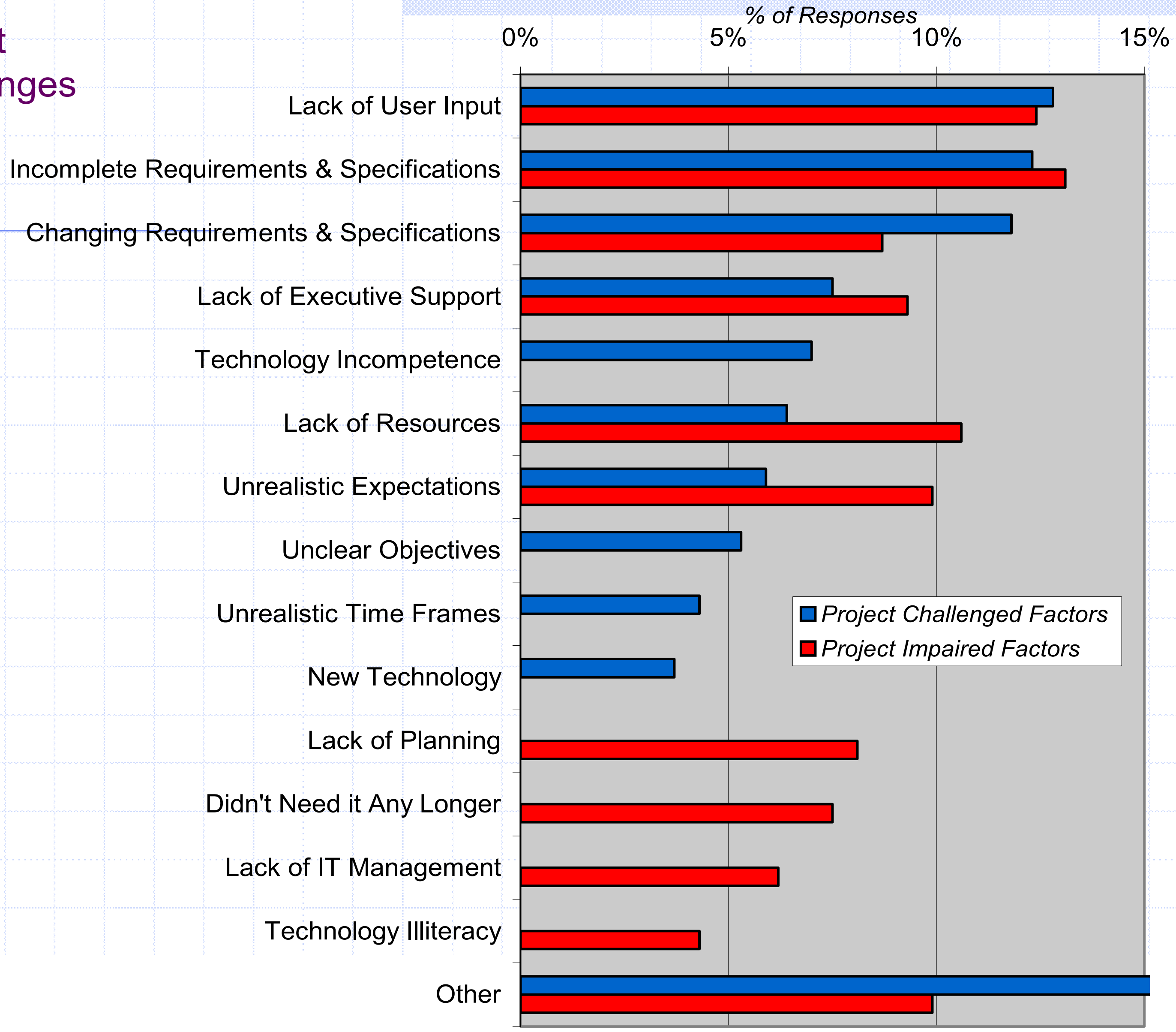
- Die Web-Anwendung muss mit 1. September 2004 online verfügbar sein *(Vorgabe des Auftraggebers).*
- Die Web-Anwendung muss gleichzeitig mindestens 2500 Benutzer unterstützen *(Qualitätsziel des Auftraggebers).*
- Als Entwicklungsplattform soll J2EE zum Einsatz kommen *(Technologieerwartung der Entwickler).*
- Die Übertragung sämtlicher Geschäftsdaten muss gesichert erfolgen *(Qualitätsziel des Benutzers).*
- Die Benutzerschnittstelle muss Layouts für unterschiedliche Kundengruppen ermöglichen *(Qualitätsziel des Auftraggebers).*
- Ein beliebiger Benutzer muss in der Lage sein, mit der Web-Anwendung ein gewünschtes Produkt innerhalb von drei Minuten zu finden *(Usability-Ziel des Auftraggebers).*
- Der Benutzer soll durch Auswahl eines Symbols jederzeit die im Warenkorb befindlichen Artikel anzeigen können *(Funktionsziel eines Benutzers).*

Software / IT Challenges

- 30% of software development projects fail.
- 70% of the remainder
  - Are over budget by 189%
  - Behind schedule by 222%
- More than 50% of this trouble is caused by inadequate requirements definition.

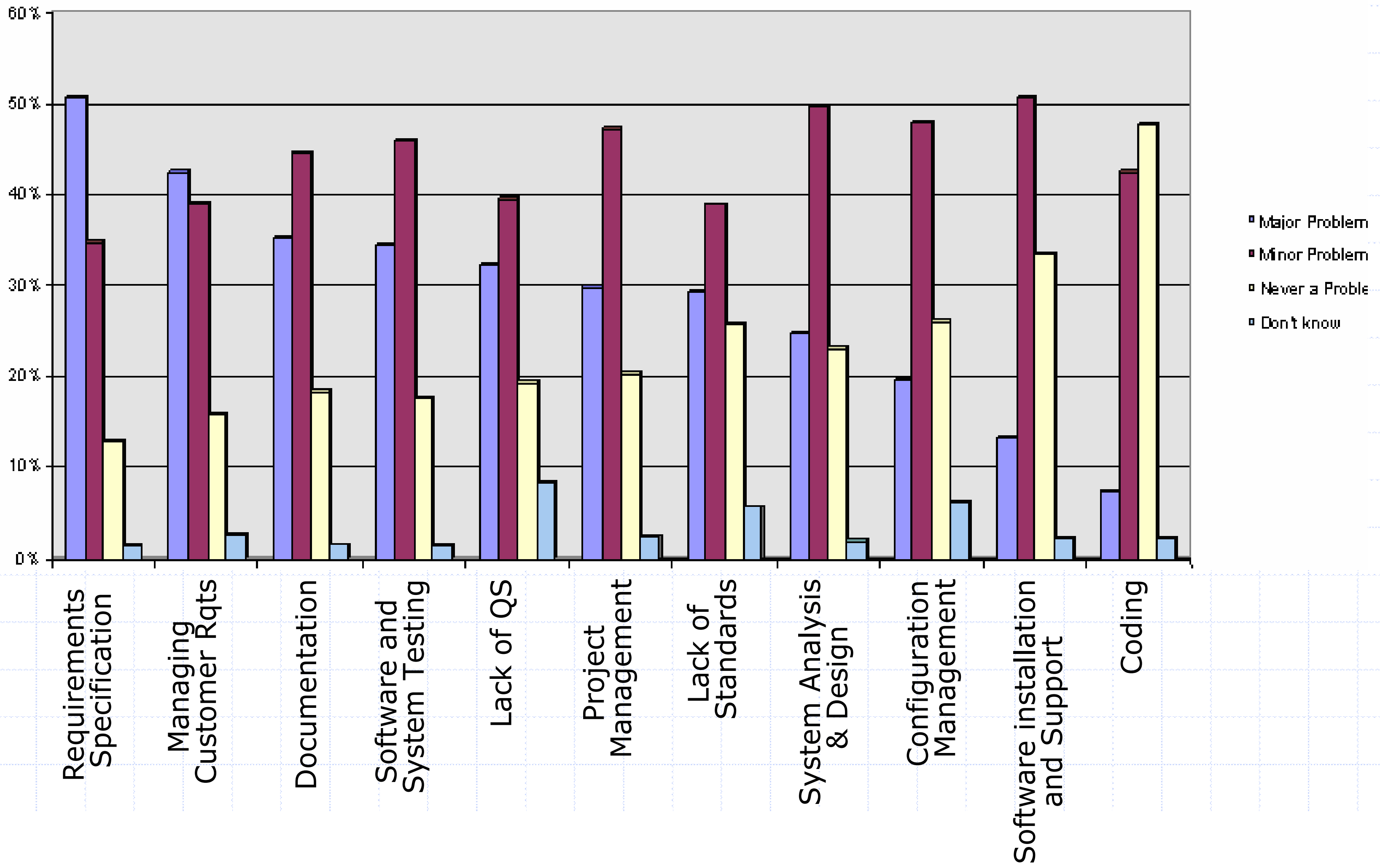
(352 companies, 8000 Projects. Source: The Standish Group, 1995)

Project Challenges



Source: 352 US Companies, 8000 Projects, The Standish Group, 1994

Perceived Relative Importance of Software Problems in Europe



Source: European Software Institute, TR 95104, www.esi.es, ESSI Project No. 11000, ESPITI, 3800 companies, 17 European countries

Einleitung

- Anforderungen spielen für die Qualität von Software-intensiven Systemen eine entscheidende Rolle.
- Nach wie vor Probleme mit Anforderungen
  - Unklar spezifiziert
  - Fehlerhaft
  - Unvollständig
  - Instabil
  - ...
- Die Folgen
  - Mangelnde Akzeptanz durch die Benutzer
  - Fehlplanungen
  - Inadäquate Softwarearchitekturen
  - ...

Woher kommen Anforderungen?

- Die Beteiligten und Betroffenen der Anwendungsentwicklung *(stakeholder)* mit ihren Zielen und Erwartungen bilden den Ausgangspunkt jeder Anforderungsermittlung.
- Stakeholder sind Personen oder Organisationen, die durch eine zu entwickelnde Anwendung betroffen sind und einen direkten oder indirekten Einfluss auf die Anforderungen nehmen.

Audience Survey - I

- What is your role in the product requirements process?
  - Sales & marketing
  - Engineering
  - Executive
  - Research & development
  - End user

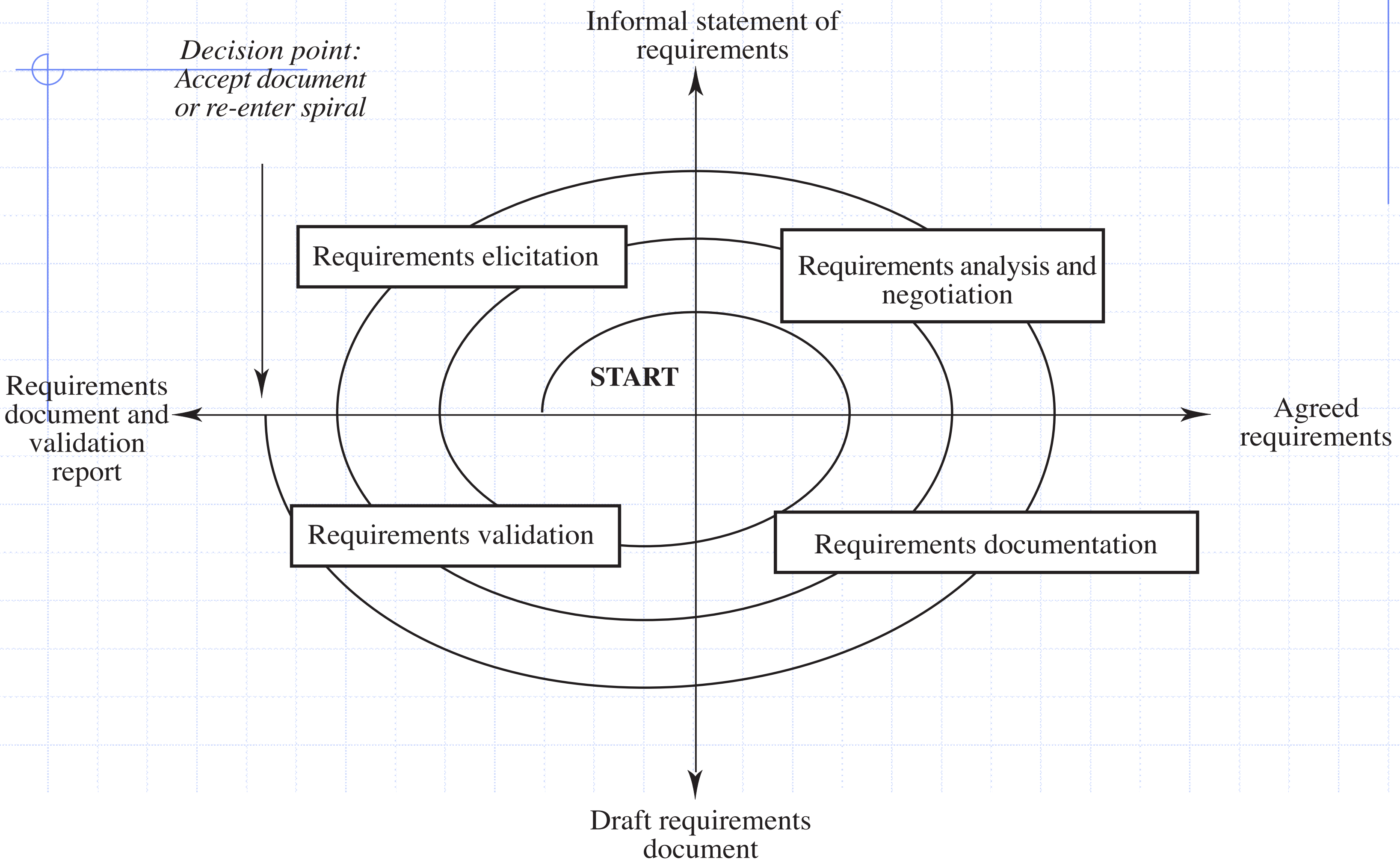
Audience Survey - II

- Which of these play a significant part in your organization’s typical requirements approach?
  - Prose documents
  - Requirements specification tools
  - Prototyping
  - Business case analysis
  - Stakeholder win-win concepts

Audience Survey - III

- What are your major concerns with your organization’s typical requirements approach?
  - Takes too long to do well
  - Too many defects
  - Too hard to keep up with change
  - Key stakeholders excluded
  - Too bureaucratic

Spiral RE Prozess



Informal statement of requirements

Decision point: Accept document or re-enter spiral

Requirements elicitation

Requirements analysis and negotiation

START

Requirements document and validation report

Agreed requirements

Requirements validation

Requirements documentation

Draft requirements document

(G. Kotonya and I. Sommerville 1998)

Wichtige Aufgaben des RE

- Ermitteln und Verhandeln der Anforderungen
- Beschreiben der Anforderungen
- Prüfen der Anforderungen
- Verwalten der Anforderungen

Ermitteln der Anforderungen

- Anforderungen können nicht durch simple Befragung der Stakeholder gewonnen werden.
- Anforderungen sind das Ergebnis eines gemeinsamen Lern- und Konsensbildungsprozesses.
- Für diese Aufgabe existieren zahlreiche Methoden:
  - Kreativitätstechniken
  - Szenariobasierte Methoden
  - Multikriterielle Entscheidungsverfahren
  - Moderationstechniken
  - Interviews
  - Dokumentenanalyse
  - …

Beschreiben der Anforderungen

- Die ermittelten Anforderungen müssen in einer projektgemäßen Form in einem **Anforderungsdokument** beschrieben werden.
- Es existieren zahlreiche Beschreibungsformen:
  - Informale (z.B. User Stories aus eXtreme Programming)
  - Semi-formale (z.B. Anwendungsfälle)
  - Formale (z.B. Z)
- Wahl der Beschreibungsform hängt ab von:
  - Projektrisiko
  - Adressatenkreis (Stakeholder und deren Vorwissen)

Prüfen der Anforderungen

- Dies umfasst
  - Validierung (»Haben wir das Richtige spezifiziert?«)
  - Verifikation (»Haben wir richtig spezifiziert?«)

- Verfahren
  - Reviews
  - Inspektionen
  - Prototyping
  - …

Verwalten der Anforderungen

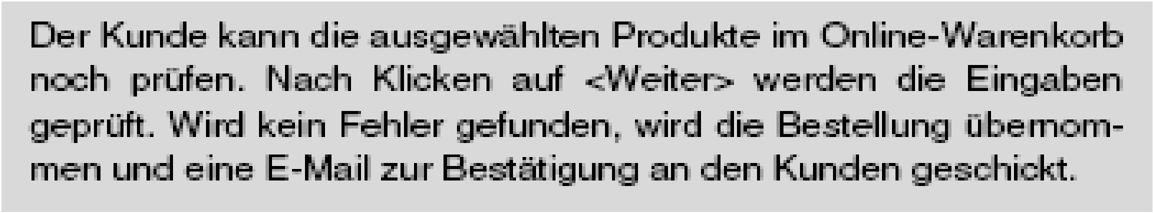
- Änderungen sind ein wesentliches Merkmal von Software-Projekten. Anforderungen sind in der Regel nicht statisch, sondern Änderungen unterworfen.
- Das Verwalten von Anforderungen (*Requirements Management*) unterstützt:
  - Einbringen neuer Anforderungen
  - Änderungen bestehender Anforderungen
  - Verwaltung der Abhängigkeiten zwischen Anforderungen sowie Beziehungen zu anderen Entwicklungsergebnissen (*traceability*)

Beschreibungsformen

- Stories (z.B. wie in Extreme Programming)
- Diagramme (z.B. Use Case Diagramme)
- Formatierte Spezifikationen (z.B. Use Cases)
- Formale Spezifikationen (z.B. Z)
- ...

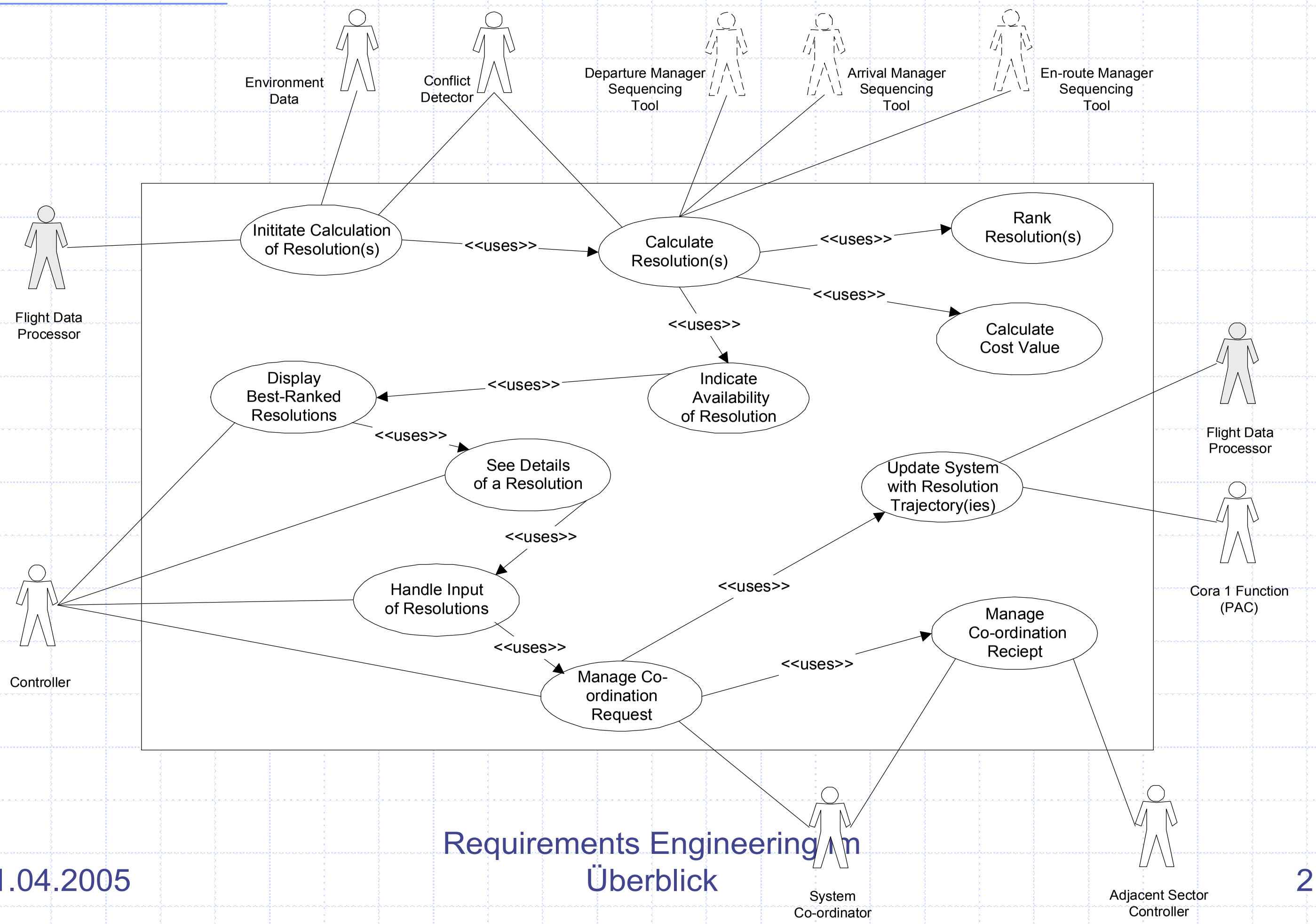
User Story

- Umgangssprachliche Beschreibungen gewünschter Eigenschaften, die eingesetzt werden, um ein gemeinsames Verständnis zwischen Kunden und Entwicklern herzustellen.



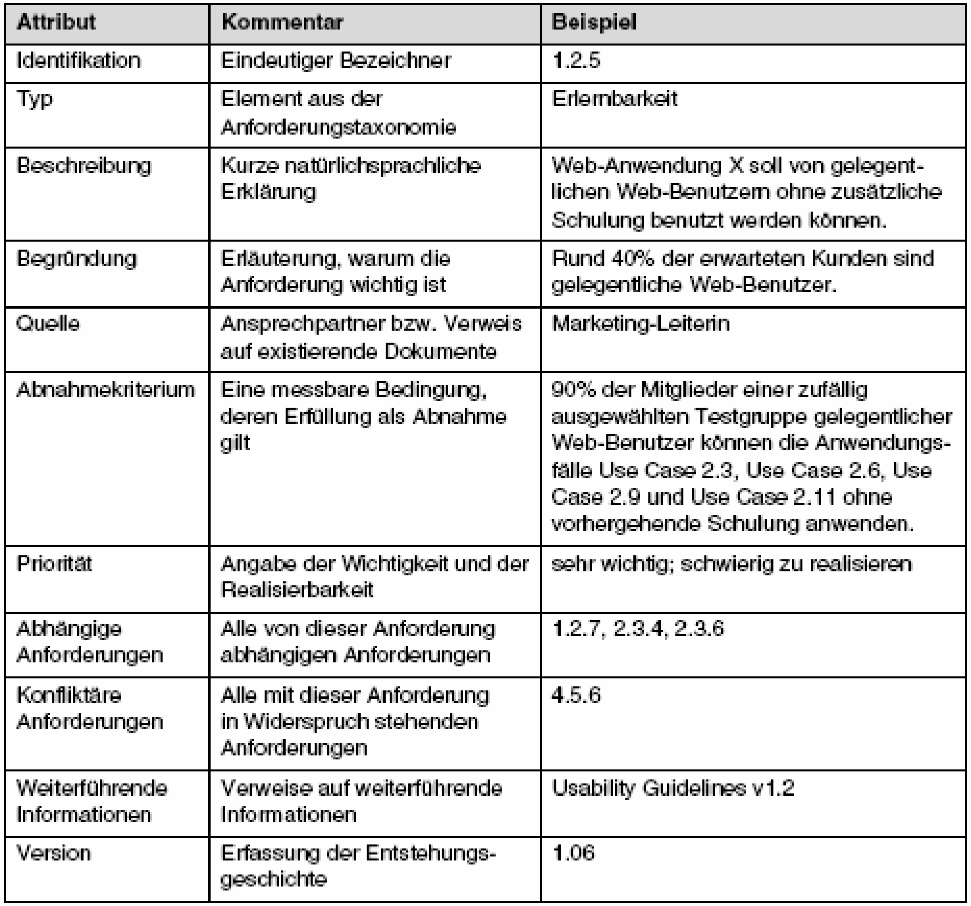
Der Kunde kann die ausgewählten Produkte im Online-Warenkorb noch prüfen. Nach Klicken auf <Weiter> werden die Eingaben geprüft. Wird kein Fehler gefunden, wird die Bestellung übernommen und eine E-Mail zur Bestätigung an den Kunden geschickt.

Diagramme



Environment Data
Conflict Detector
Departure Manager Sequencing Tool
Arrival Manager Sequencing Tool
En-route Manager Sequencing Tool
Flight Data Processor
Inititate Calculation of Resolution(s)
<<uses>>
Calculate Resolution(s)
<<uses>>
Rank Resolution(s)
<<uses>>
Calculate Cost Value
<<uses>>
Indicate Availability of Resolution
<<uses>>
Display Best-Ranked Resolutions
<<uses>>
See Details of a Resolution
<<uses>>
Handle Input of Resolutions
<<uses>>
Manage Co-ordination Request
<<uses>>
Update System with Resolution Trajectory(ies)
<<uses>>
Manage Co-ordination Reciept
Flight Data Processor
Cora 1 Function (PAC)
Controller
System Co-ordinator
Adjacent Sector Controller

Formatierte Spezifikationen



Attribut	Kommentar	Beispiel
Identifikation	Eindeutiger Bezeichner	1.2.5
Typ	Element aus der Anforderungstaxonomie	Erlernbarkeit
Beschreibung	Kurze natürlichsprachliche Erklärung	Web-Anwendung X soll von gelegentlichen Web-Benutzern ohne zusätzliche Schulung benutzt werden können.
Begründung	Erläuterung, warum die Anforderung wichtig ist	Rund 40% der erwarteten Kunden sind gelegentliche Web-Benutzer.
Quelle	Ansprechpartner bzw. Verweis auf existierende Dokumente	Marketing-Leiterin
Abnahmekriterium	Eine messbare Bedingung, deren Erfüllung als Abnahme gilt	90% der Mitglieder einer zufällig ausgewählten Testgruppe gelegentlicher Web-Benutzer können die Anwendungsfälle Use Case 2.3, Use Case 2.6, Use Case 2.9 und Use Case 2.11 ohne vorhergehende Schulung anwenden.
Priorität	Angabe der Wichtigkeit und der Realisierbarkeit	sehr wichtig; schwierig zu realisieren
Abhängige Anforderungen	Alle von dieser Anforderung abhängigen Anforderungen	1.2.7, 2.3.4, 2.3.6
Konfliktäre Anforderungen	Alle mit dieser Anforderung in Widerspruch stehenden Anforderungen	4.5.6
Weiterführende Informationen	Verweise auf weiterführende Informationen	Usability Guidelines v1.2
Version	Erfassung der Entstehungsgeschichte	1.06

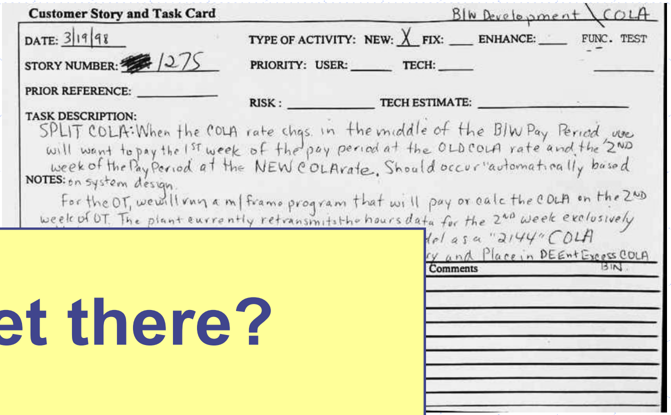
WinWin
Requirements Negotiation

Paul Grünbacher
Johannes Kepler University Linz

Paul.Gruenbacher@jku.at

Many approaches for modeling and describing requirements

- Stories, Prosa, Use Cases, Formatted Specs, etc.



Customer Story and Task Card BIW Development \ COLA

DATE: 3|19|98 TYPE OF ACTIVITY: NEW: X FIX: ___ ENHANCE: ___ FUNC. TEST

STORY NUMBER: [redacted] 1275 PRIORITY: USER: ______ TECH: ______ ______

PRIOR REFERENCE: __________ RISK : __________ TECH ESTIMATE: __________

TASK DESCRIPTION:
SPLIT COLA: When the COLA rate chgs. in the middle of the BIW Pay Period we will want to pay the 1ST week of the pay period at the OLD COLA rate and the 2ND week of the Pay Period at the NEW COLA rate. Should occur "automatically based

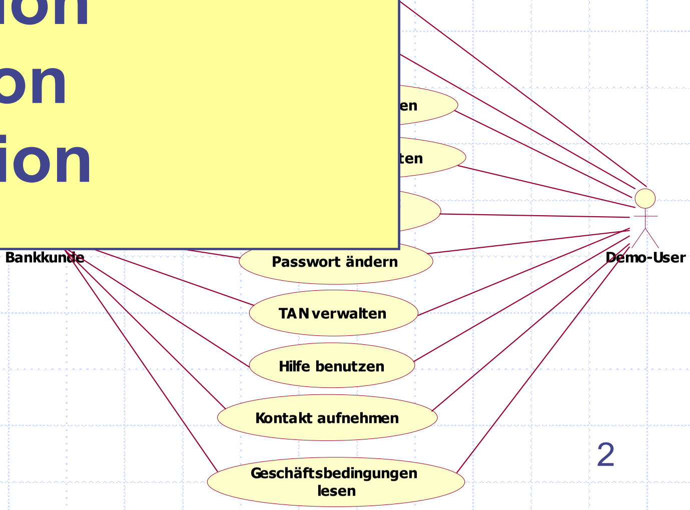
NOTES: on system design.
For the OT, we will run a m/frame program that will pay or calc the COLA on the 2ND week of OT. The plant currently retransmits the hours data for the 2ND week exclusively
…del as a "2144" COLA
…ry and Place in DEEnt Excess COLA BIN.

Comments

Attribut		
Identifikation	Ein	
Typ	Ele tax	
Beschreibung	Ku Erk	
Begründung	Erl An Sp	
Quelle	We die we Gru	
Abnahmekriterium	Ein bei An abg	
Priorität	An ggf Re An	
Abhängige Anforderungen	Alle von dieser Anforderung abhängigen Anforderungen	1.2.7, 2.3.4, 2.3.6
Konfliktäre Anforderungen	Alle Anforderungen die mit dieser Anforderung in Widerspruch stehen	4.5.6
Weiterführende Informationen	Verweise auf weiterführende Informationen	Usability Guidelines v1.2
Version	Dient zur Erfassung der Entstehungsgeschichte einer Anforderung	1.06

How do we get there?

Acquisition
Elicitation
Negotiation



WinWin

- Success-critical stakeholders jointly negotiate the requirements for a software development project to achieve mutually satisfactory solutions.

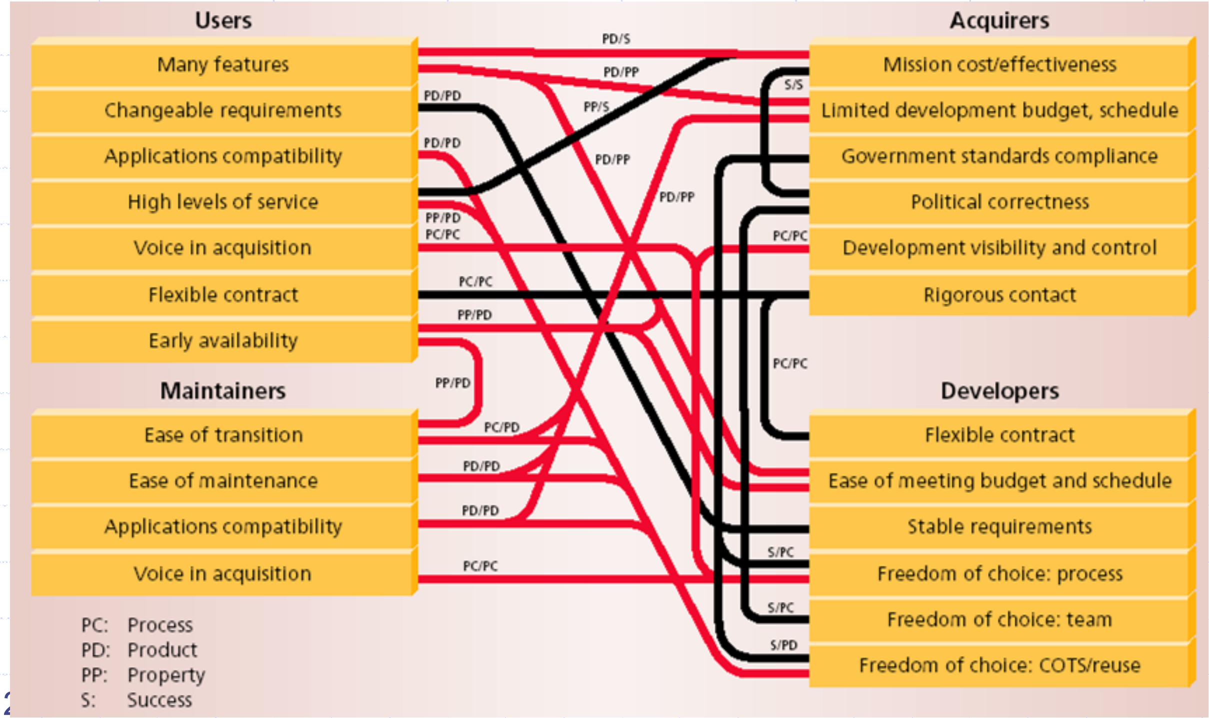
Who Are The Stakeholders?

- Customers
- Users
- Programmers
- Architects
- Domain Experts
- Analysts
- Marketing
- Sales
- Management
- …?

Why is negotiation so important?

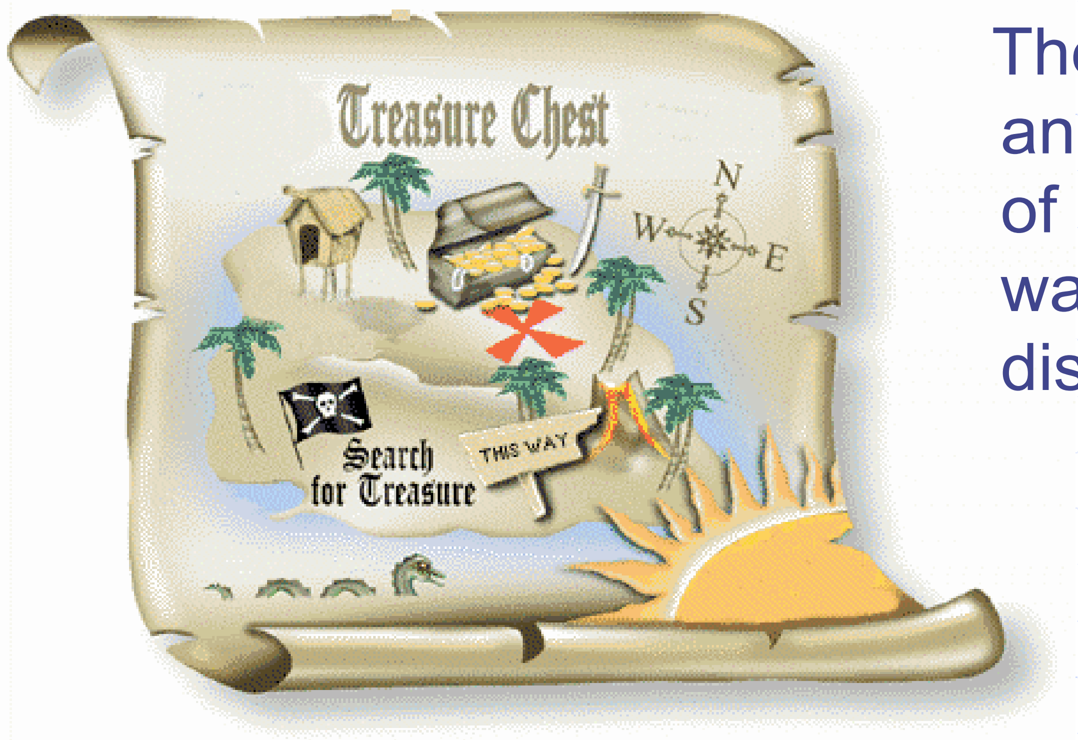
Understanding and Resolving Conflicts

Conflicts are inevitable. Achieving mutually satisfactory agreements is critical for building trust and managing expectations.



(Boehm, Port, Al Said, 2000)

Adapting to change



There is no complete and well-defined set of requirements waiting to be discovered.

Understanding and Meeting Constraints

Requirements depend on available resources and capabilities.

Sharing Knowledge

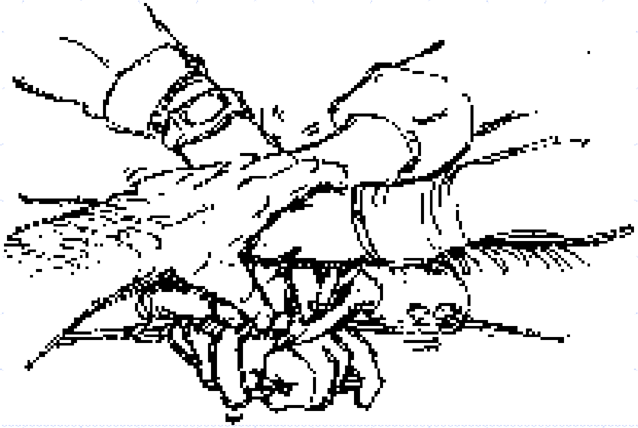
Users, customers, managers, domain experts, and developers have to share different skills, backgrounds, and expectations.

Fostering Team Learning

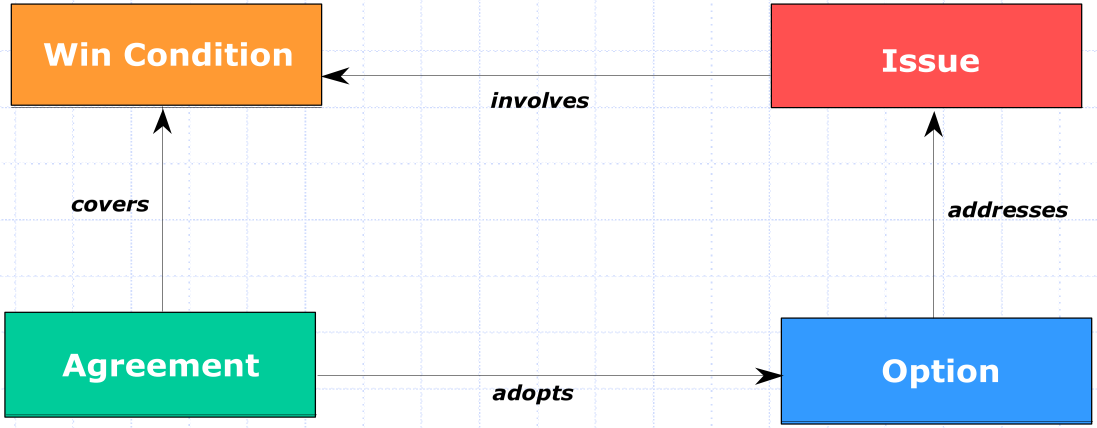
Requirements emerge from a process of co-operative learning in which they are explored, prioritized, negotiated, evaluated, and documented.

Developing Shared Commitments

- Conflicts are inevitable.
- Negotiation is critical to achieve mutually satisfactory agreements.



WinWin Negotiation Model

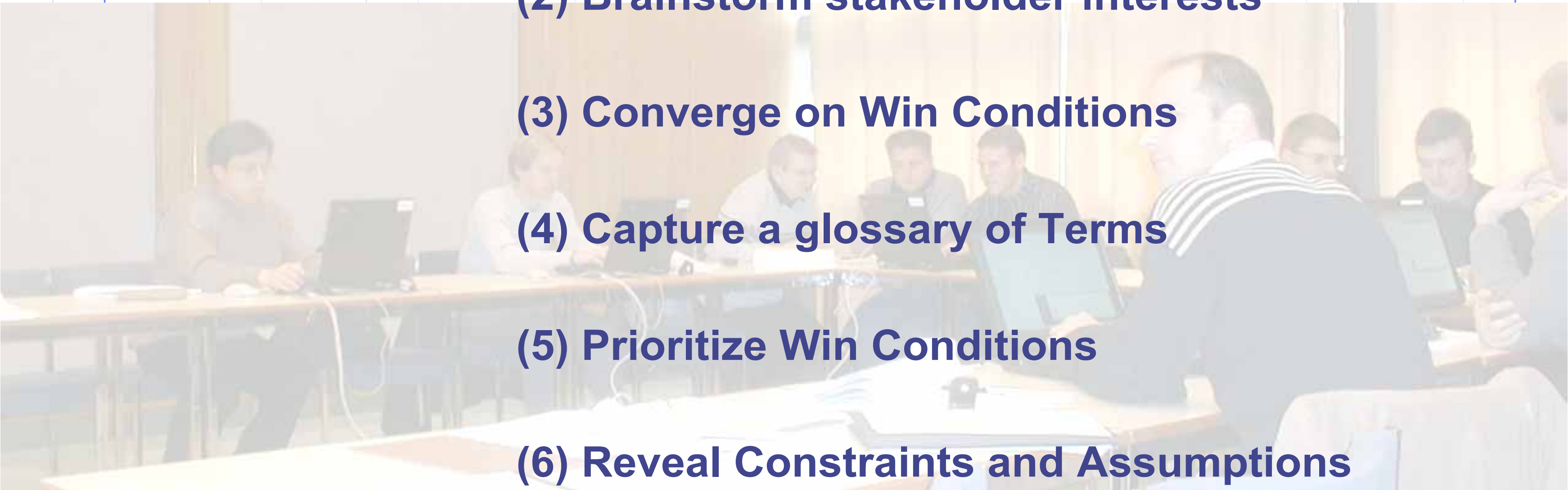


Win Condition:	Desired objective of an individual stakeholder
Issue:	Conflict, risk, uncertainty on a win condition
Option:	A way of overcoming an issue
Agreement:	A mutual commitment to an option or win condition

WinWin Equilibrium State
- All Win Conditions covered by Agreements
- No outstanding Issues

(Boehm et al. 1994)

EasyWinWin Process



(0) Identify Success-critical Stakeholders

(1) Review and expand negotiation topics

(2) Brainstorm stakeholder interests

(3) Converge on Win Conditions

(4) Capture a glossary of Terms

(5) Prioritize Win Conditions

(6) Reveal Constraints and Assumptions

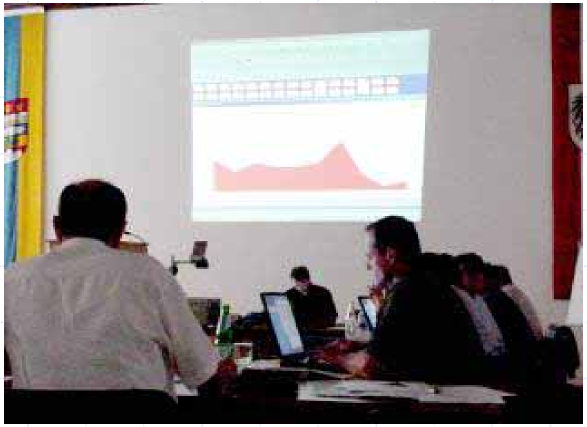
(7) Identify Issues and Options

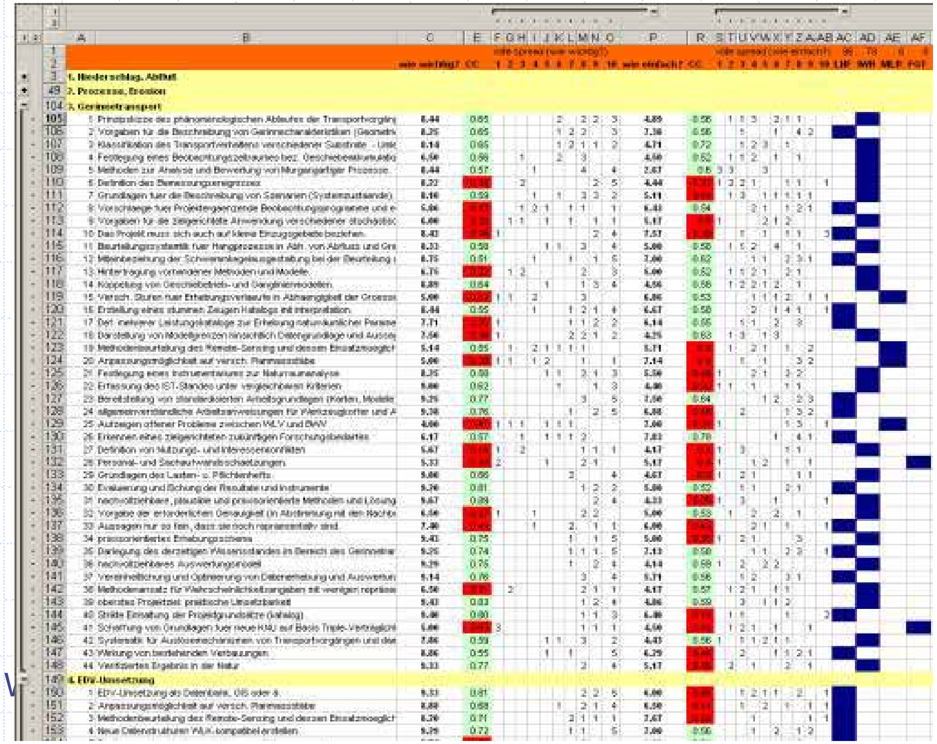
(8) Negotiate Agreements

Example 1: Flood and Avalanche Control

- Methods and Tools for Assessing Risks in Alpine Regions

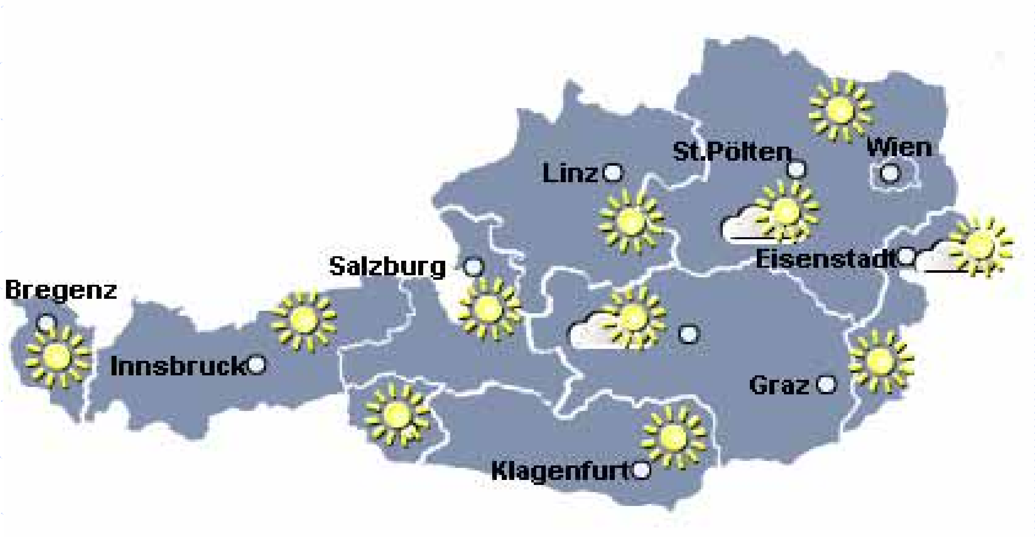
- One Day Kick-Off Meeting
  - 26 Stakeholders from different domains
  - ~400 brainstorming contributions
  - ~150 Win Conditions
  - Detailed analysis of priorities and conflicts



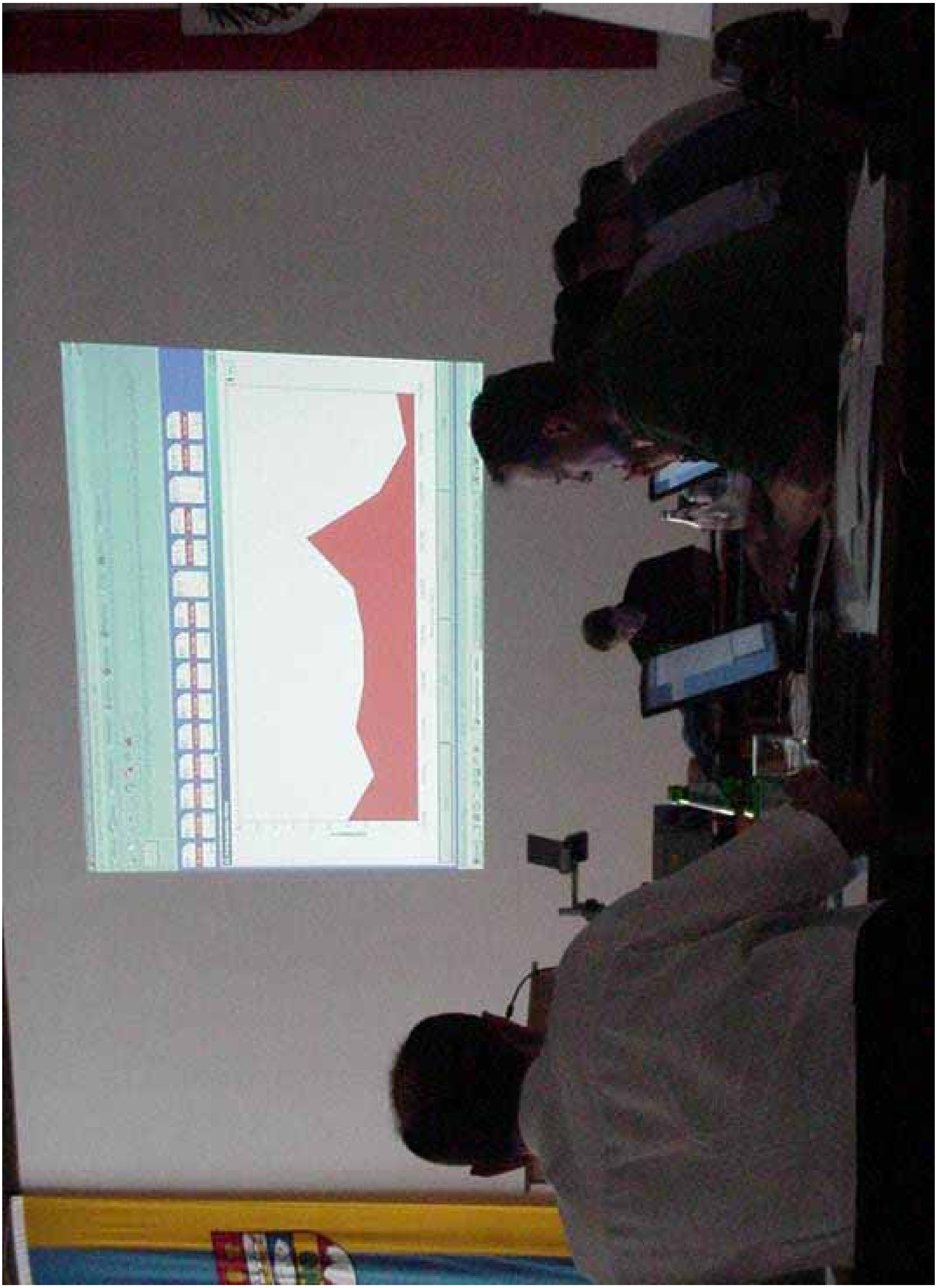


Example 2:
Graphical Weather Forecast Editor

- Complex graphical editor for the meteorologist



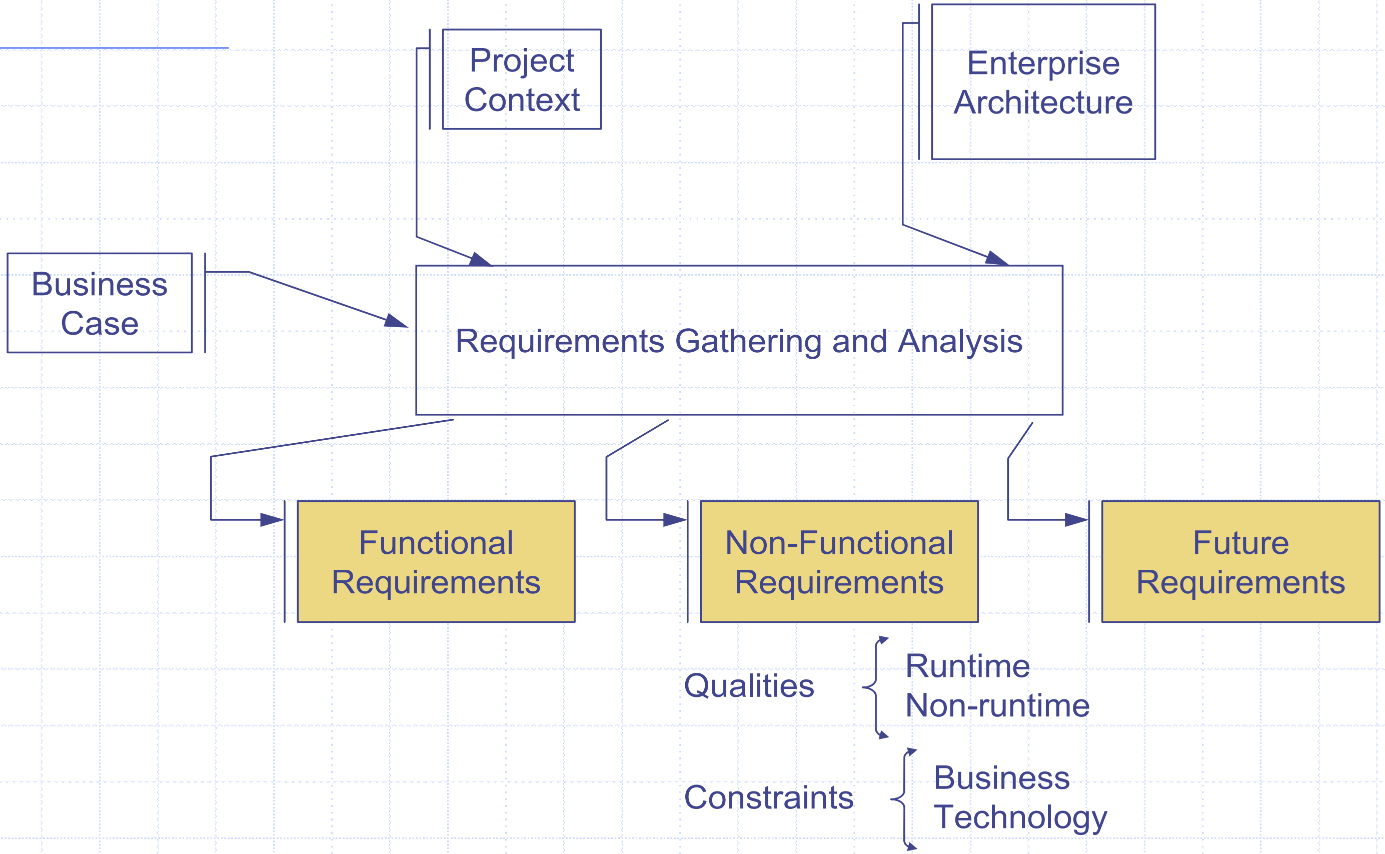
RISC
Software
GmbH



(1) Review and Expand Negotiation Topics

- Objective: refine, and customize the outline of negotiation topics.
- How: Could-be, Should-be
- Result: Shared Outline that helps to
  - stimulate your thinking,
  - organize your win conditions, and
  - serves as a completeness checklist for negotiations.

Stakeholders negotiate about …


Business Case

Project Context

Enterprise Architecture

Requirements Gathering and Analysis

Functional Requirements

Non-Functional Requirements

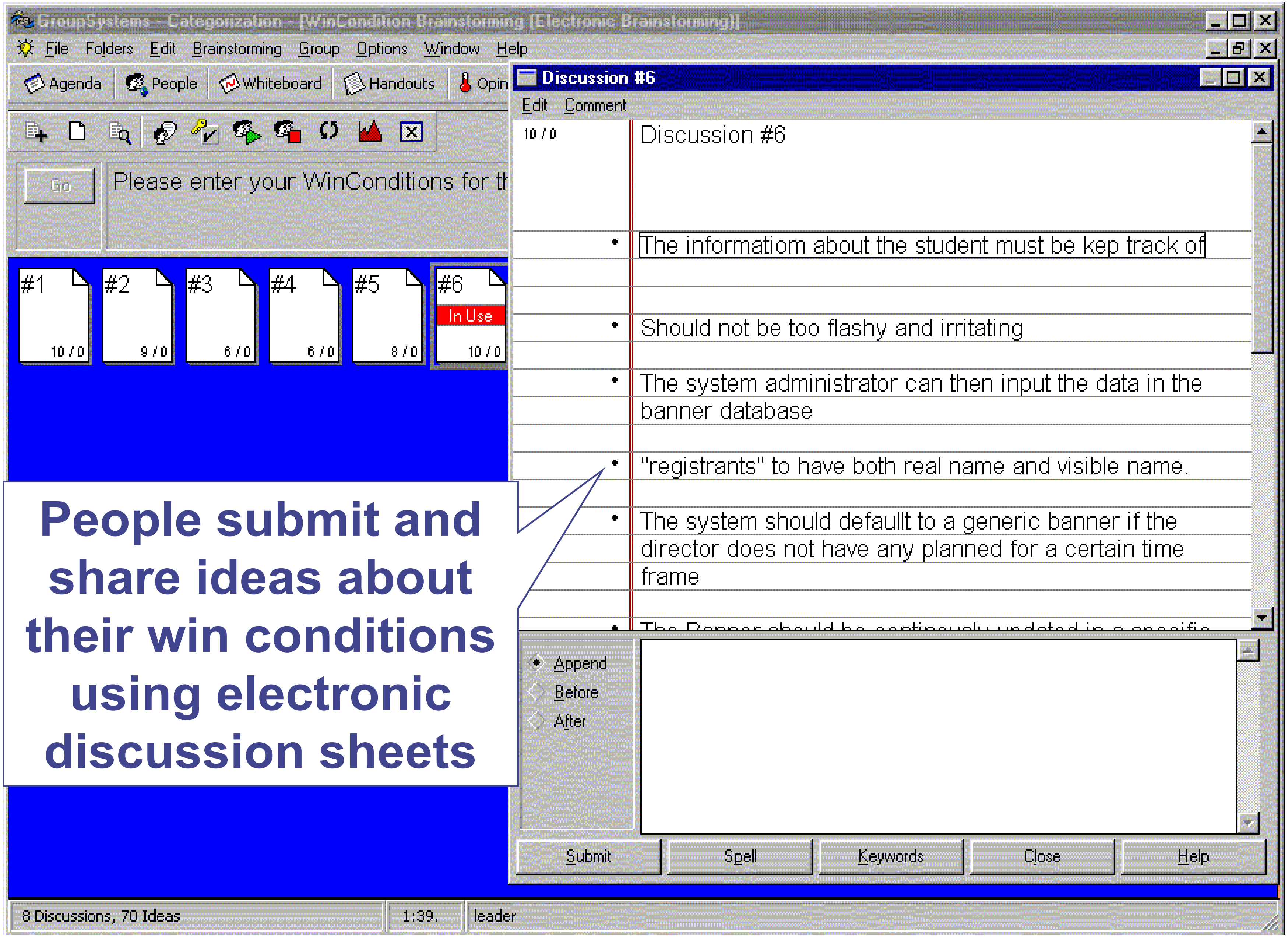
Future Requirements

Qualities { Runtime, Non-runtime }

Constraints { Business, Technology }

(2) Brainstorm Stakeholder Interests

- Objective: Share perspectives, views, background, expectations
- How: Anonymous, rapid brainstorming
- Result: An unstructured set of comments about their vested interests
  (win conditions)



Please enter your WinConditions for th

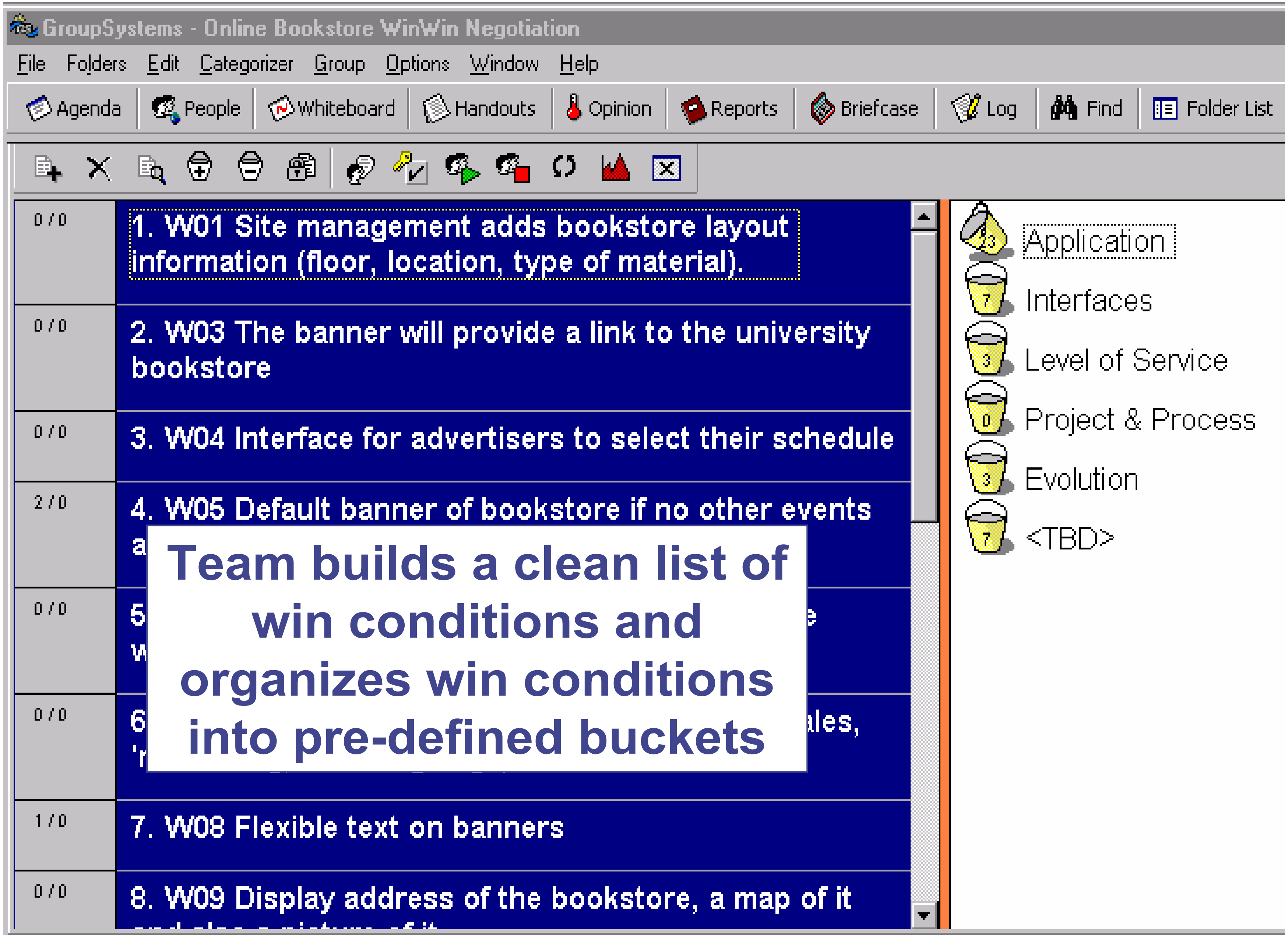
Discussion #6

- The informatiom about the student must be kep track of
- Should not be too flashy and irritating
- The system administrator can then input the data in the banner database
- "registrants" to have both real name and visible name.
- The system should defaullt to a generic banner if the director does not have any planned for a certain time frame
- The Banner should be continuously updated in a specific

People submit and share ideas about their win conditions using electronic discussion sheets

(3) Converge on Win Conditions

- Objective: Build and organize win conditions
- How: Structured discussion to converge on key win conditions
- Result: List of clearly stated, unambiguous win conditions


GroupSystems - Online Bookstore WinWin Negotiation
File Folders Edit Categorizer Group Options Window Help
Agenda
People
Whiteboard
Handouts
Opinion
Reports
Briefcase
Log
Find
Folder List

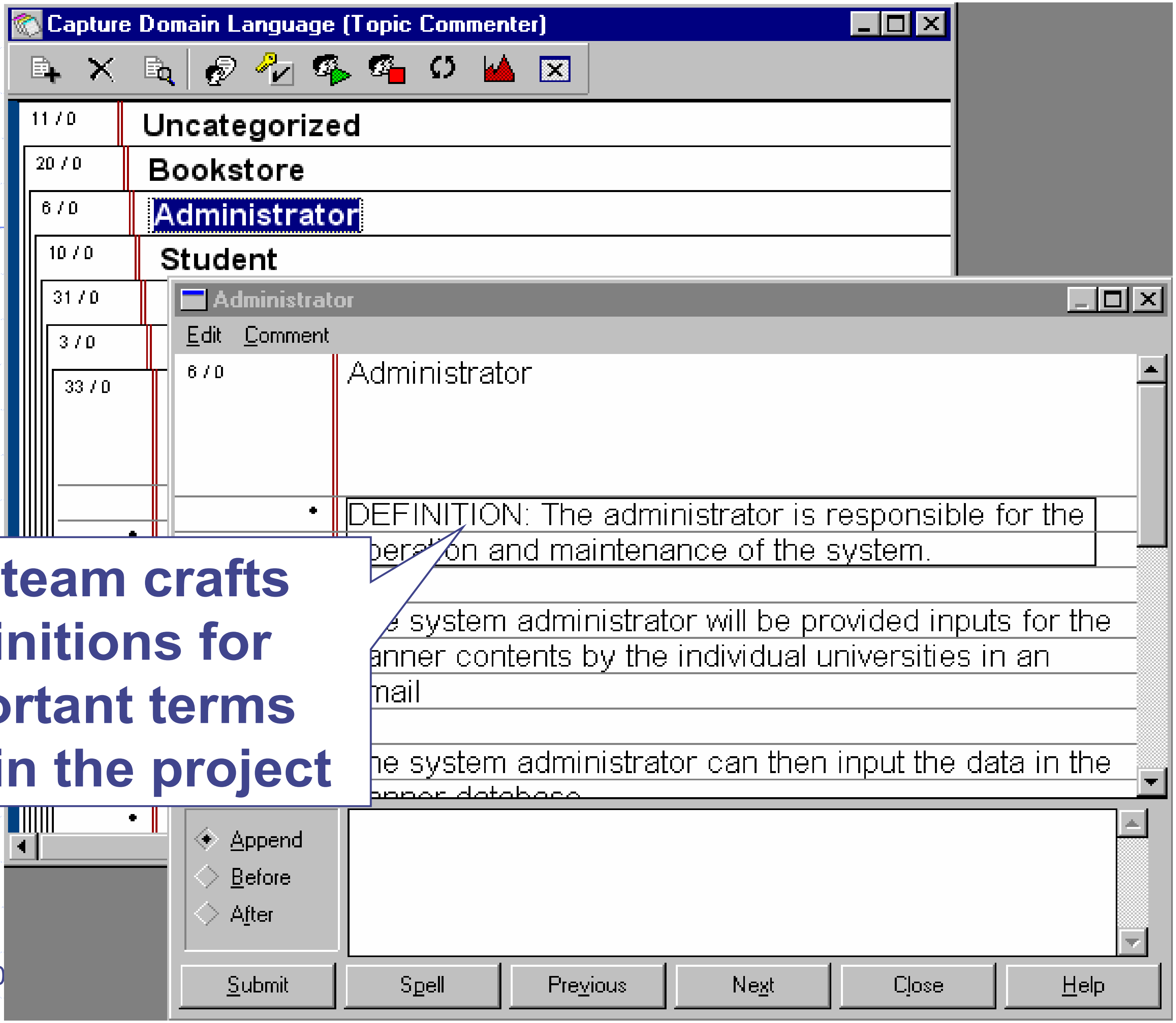
0 / 0	1. W01 Site management adds bookstore layout information (floor, location, type of material).
0 / 0	2. W03 The banner will provide a link to the university bookstore
0 / 0	3. W04 Interface for advertisers to select their schedule
2 / 0	4. W05 Default banner of bookstore if no other events a
0 / 0	5 w e
0 / 0	6 'r ales,
1 / 0	7. W08 Flexible text on banners
0 / 0	8. W09 Display address of the bookstore, a map of it

23	Application
7	Interfaces
3	Level of Service
0	Project & Process
3	Evolution
7	<TBD>

Team builds a clean list of win conditions and organizes win conditions into pre-defined buckets

(4) Capture a Glossary of Terms

- Objective: Define and share meaning of important terms.
- How: Initial definitions based on stakeholder statements; joint review
- Result: A glossary of terms with definitions and stakeholder statements showing usage of terms



11 / 0	Uncategorized
20 / 0	Bookstore
6 / 0	Administrator
10 / 0	Student
31 / 0	
3 / 0	
33 / 0	

6 / 0 Administrator

- DEFINITION: The administrator is responsible for the pera on and maintenance of the system.

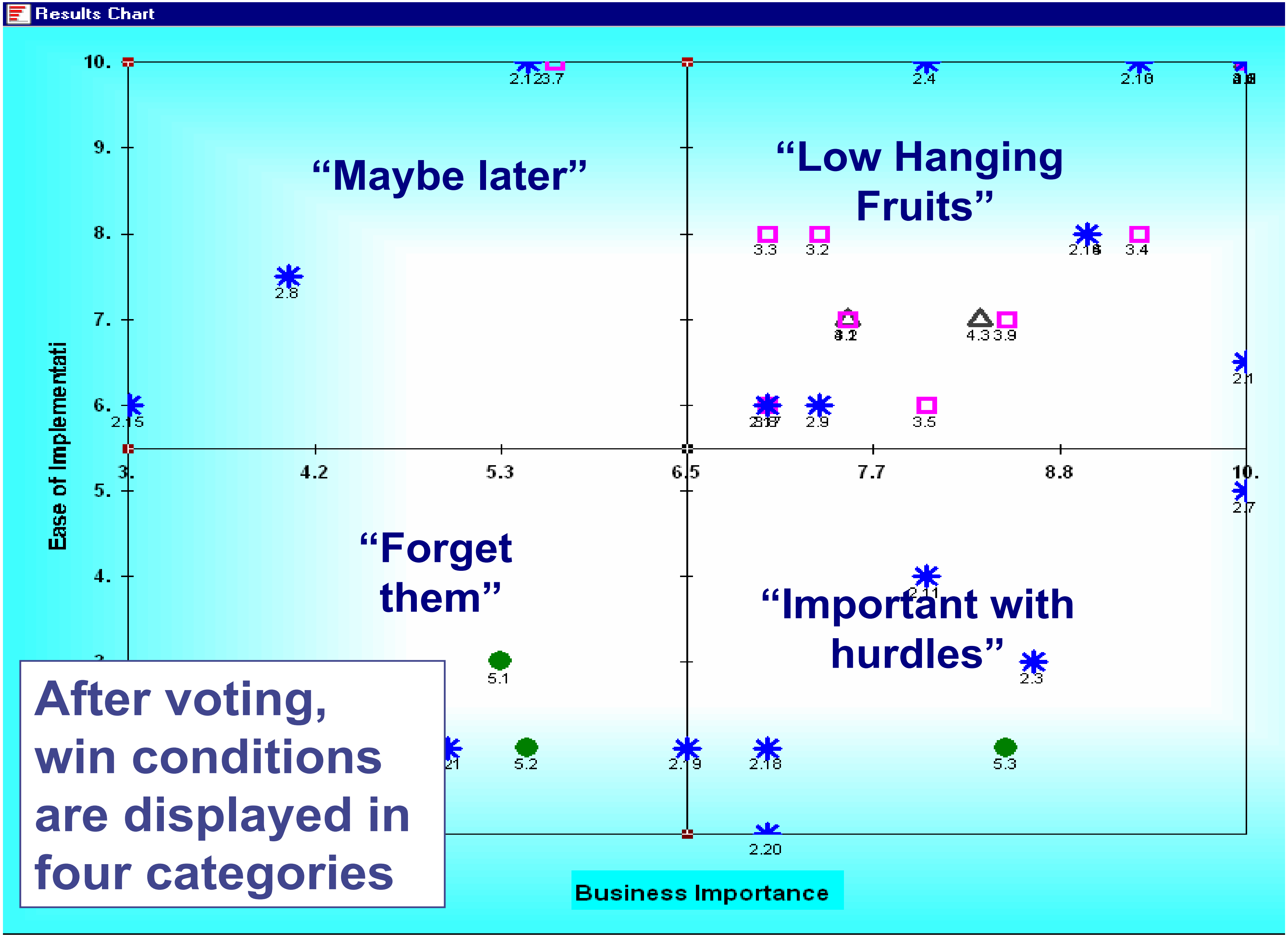
e system administrator will be provided inputs for the anner contents by the individual universities in an mail

he system administrator can then input the data in the anner database

The team crafts definitions for important terms used in the project

(5) Prioritize win conditions

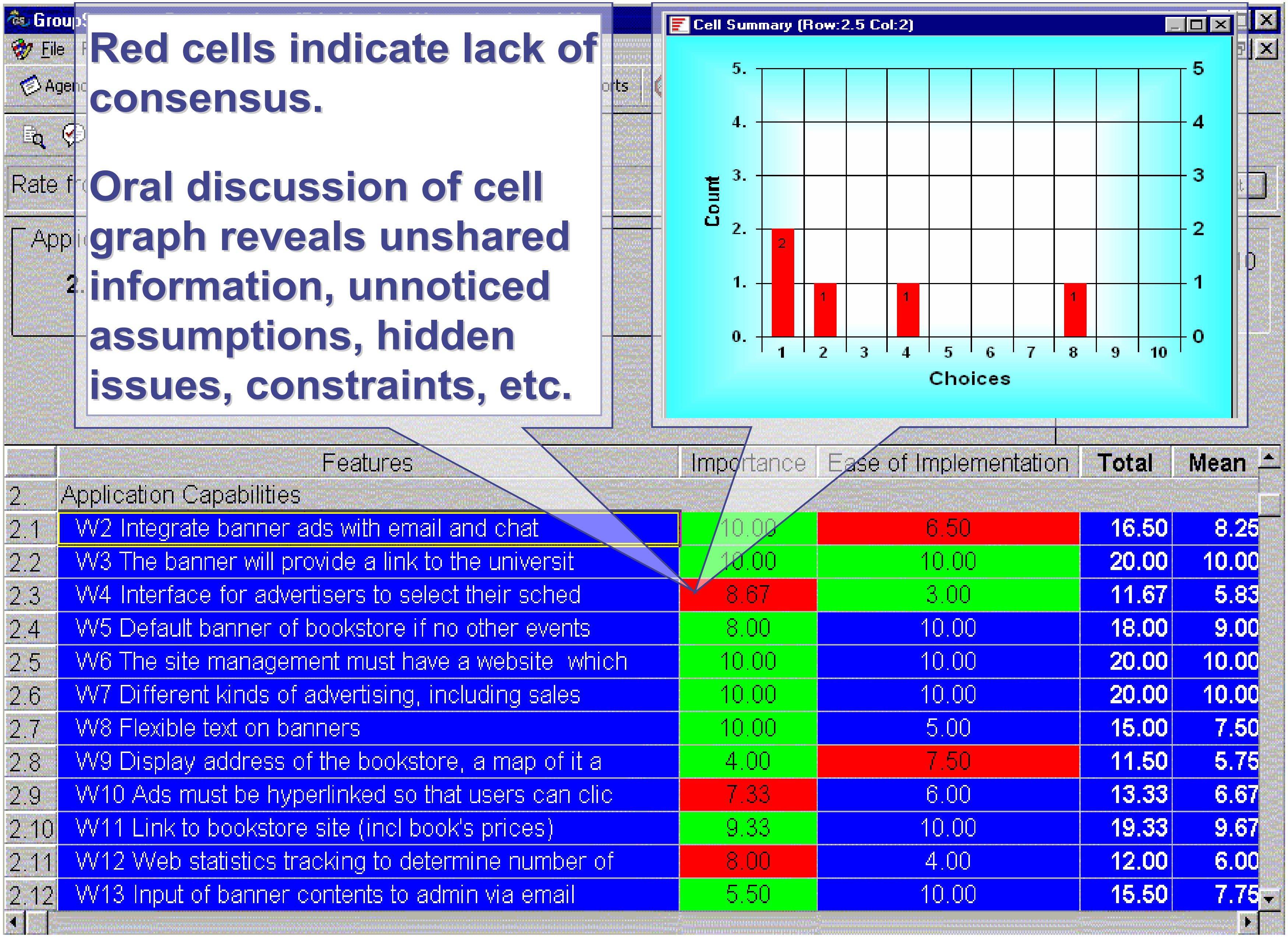
- Objective: Scope project, gain focus
- How: Vote on Business Importance & Ease of Realization
- Result: Prioritized win conditions



After voting, win conditions are displayed in four categories

(6) Reveal Constraints and Assumptions

- Objective: Surface and understand hidden assumptions
- How: Analyze prioritization poll to reveal hidden assumptions, different perceptions, …
- Result: Comments, Issues, sometimes Options



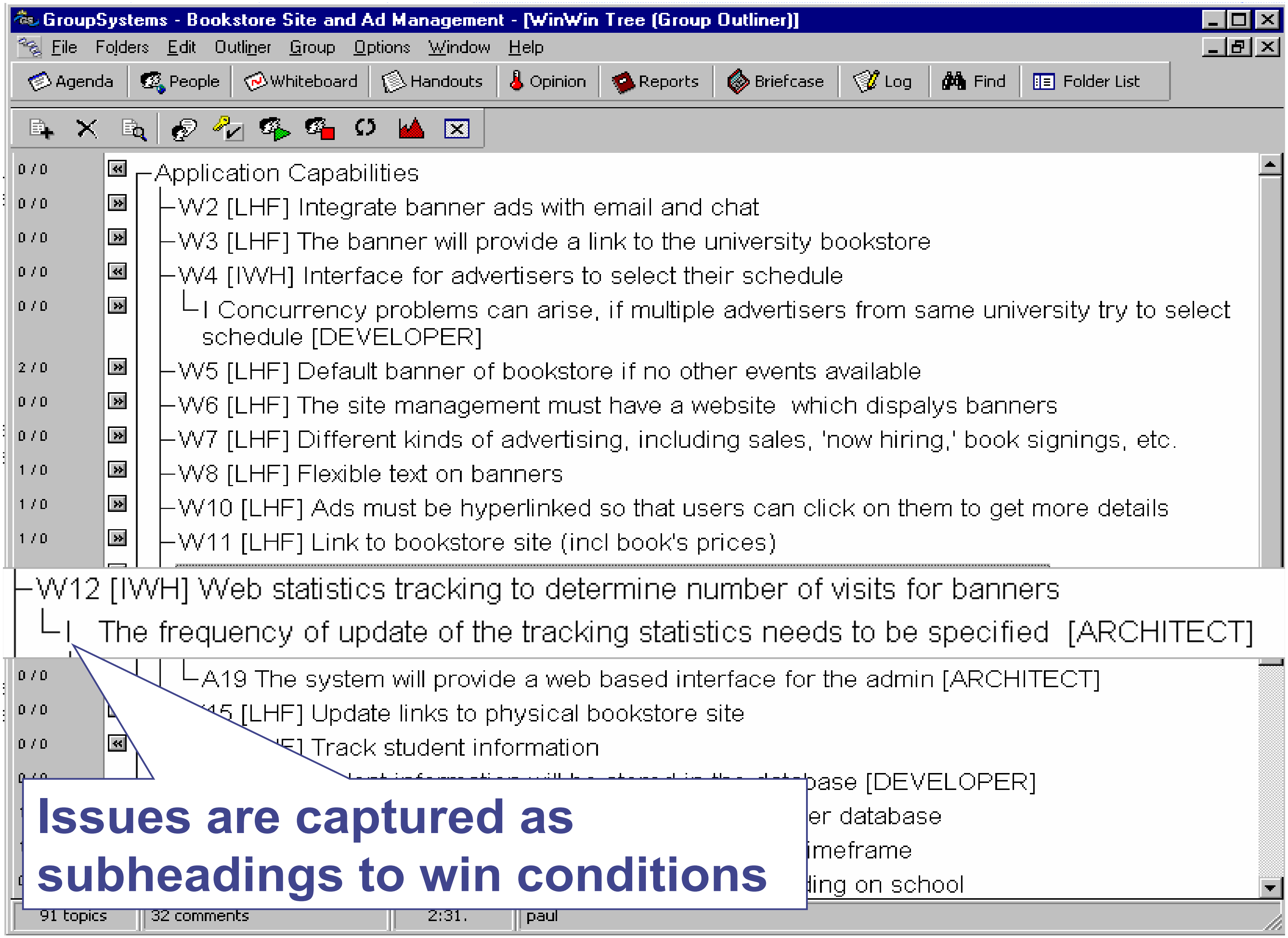
Red cells indicate lack of consensus.

Oral discussion of cell graph reveals unshared information, unnoticed assumptions, hidden issues, constraints, etc.

	Features	Importance	Ease of Implementation	Total	Mean
2.	Application Capabilities				
2.1	W2 Integrate banner ads with email and chat	🟩 10.00	🟥 6.50	16.50	8.25
2.2	W3 The banner will provide a link to the universit	🟩 10.00	🟩 10.00	20.00	10.00
2.3	W4 Interface for advertisers to select their sched	🟥 8.67	🟩 3.00	11.67	5.83
2.4	W5 Default banner of bookstore if no other events	🟩 8.00	10.00	18.00	9.00
2.5	W6 The site management must have a website which	🟩 10.00	10.00	20.00	10.00
2.6	W7 Different kinds of advertising, including sales	🟩 10.00	10.00	20.00	10.00
2.7	W8 Flexible text on banners	🟩 10.00	5.00	15.00	7.50
2.8	W9 Display address of the bookstore, a map of it a	🟩 4.00	🟥 7.50	11.50	5.75
2.9	W10 Ads must be hyperlinked so that users can clic	🟥 7.33	6.00	13.33	6.67
2.10	W11 Link to bookstore site (incl book's prices)	🟩 9.33	10.00	19.33	9.67
2.11	W12 Web statistics tracking to determine number of	🟥 8.00	4.00	12.00	6.00
2.12	W13 Input of banner contents to admin via email	🟩 5.50	10.00	15.50	7.75

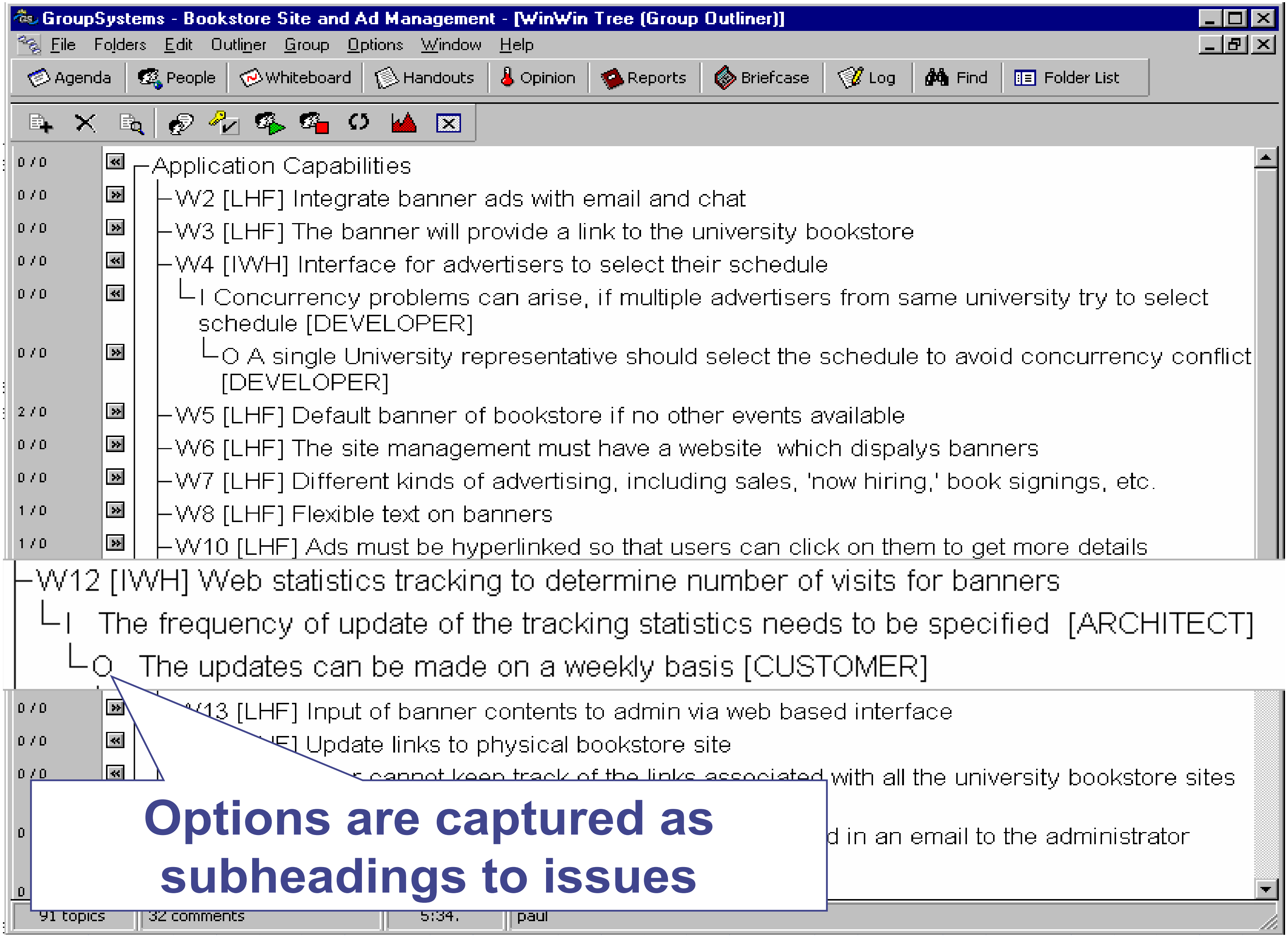
(7) Identify Issues and Options

- Objective: Explore candidate issues and options; Understand issues and options
- How: Develop/Review pass for Issues and Options
- Result: A WinWin Tree
  - Win conditions
  - Issues
  - Options


GroupSystems - Bookstore Site and Ad Management - [WinWin Tree (Group Outliner)]
File Folders Edit Outliner Group Options Window Help
Agenda People Whiteboard Handouts Opinion Reports Briefcase Log Find Folder List

```
0 / 0  Application Capabilities
0 / 0    W2 [LHF] Integrate banner ads with email and chat
0 / 0    W3 [LHF] The banner will provide a link to the university bookstore
0 / 0    W4 [IWH] Interface for advertisers to select their schedule
0 / 0      I Concurrency problems can arise, if multiple advertisers from same university try to select
           schedule [DEVELOPER]
2 / 0    W5 [LHF] Default banner of bookstore if no other events available
0 / 0    W6 [LHF] The site management must have a website  which dispalys banners
0 / 0    W7 [LHF] Different kinds of advertising, including sales, 'now hiring,' book signings, etc.
1 / 0    W8 [LHF] Flexible text on banners
1 / 0    W10 [LHF] Ads must be hyperlinked so that users can click on them to get more details
1 / 0    W11 [LHF] Link to bookstore site (incl book's prices)
W12 [IWH] Web statistics tracking to determine number of visits for banners
  I   The frequency of update of the tracking statistics needs to be specified  [ARCHITECT]
0 / 0      A19 The system will provide a web based interface for the admin [ARCHITECT]
0 / 0    15 [LHF] Update links to physical bookstore site
0 / 0    F] Track student information
           base [DEVELOPER]
           er database
           imeframe
           ling on school
91 topics   32 comments   2:31.   paul
```

Issues are captured as subheadings to win conditions


GroupSystems - Bookstore Site and Ad Management - [WinWin Tree (Group Outliner)]
File Folders Edit Outliner Group Options Window Help
Agenda People Whiteboard Handouts Opinion Reports Briefcase Log Find Folder List

0/0 Application Capabilities

0/0 W2 [LHF] Integrate banner ads with email and chat

0/0 W3 [LHF] The banner will provide a link to the university bookstore

0/0 W4 [IWH] Interface for advertisers to select their schedule

0/0 I Concurrency problems can arise, if multiple advertisers from same university try to select schedule [DEVELOPER]

0/0 O A single University representative should select the schedule to avoid concurrency conflict [DEVELOPER]

2/0 W5 [LHF] Default banner of bookstore if no other events available

0/0 W6 [LHF] The site management must have a website which dispalys banners

0/0 W7 [LHF] Different kinds of advertising, including sales, 'now hiring,' book signings, etc.

1/0 W8 [LHF] Flexible text on banners

1/0 W10 [LHF] Ads must be hyperlinked so that users can click on them to get more details

W12 [IWH] Web statistics tracking to determine number of visits for banners

I The frequency of update of the tracking statistics needs to be specified [ARCHITECT]

O The updates can be made on a weekly basis [CUSTOMER]

0/0 13 [LHF] Input of banner contents to admin via web based interface

0/0 F] Update links to physical bookstore site

0/0 cannot keep track of the links associated with all the university bookstore sites

0 d in an email to the administrator

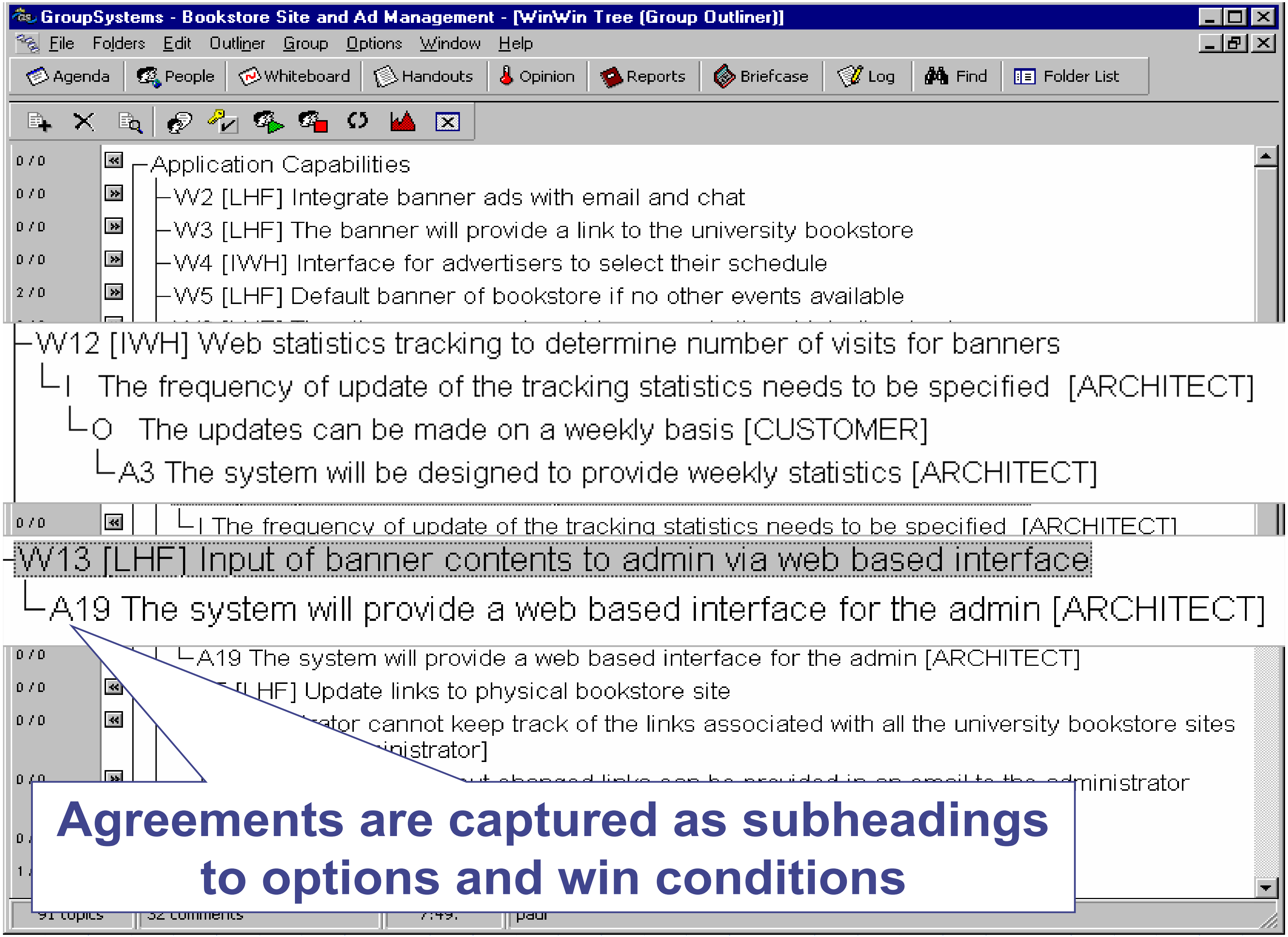
0

91 topics | 32 comments | 5:34. | paul

Options are captured as subheadings to issues

(8) Negotiate Agreements

- Objective: Negotiate agreements
- How:
  - Adopt win conditions that raised no issues as agreements;
  - Adopt options as agreements
- Result: A WinWin Tree:
  - Win conditions
  - Issues
  - Options
  - Agreements


GroupSystems - Bookstore Site and Ad Management - [WinWin Tree (Group Outliner)]
File Folders Edit Outliner Group Options Window Help
Agenda People Whiteboard Handouts Opinion Reports Briefcase Log Find Folder List
0 / 0
0 / 0
0 / 0
0 / 0
2 / 0
0 / 0
0 / 0
0 / 0
0 / 0
91 topics
32 comments
7:49.
paul

Application Capabilities
- W2 [LHF] Integrate banner ads with email and chat
- W3 [LHF] The banner will provide a link to the university bookstore
- W4 [IWH] Interface for advertisers to select their schedule
- W5 [LHF] Default banner of bookstore if no other events available

W12 [IWH] Web statistics tracking to determine number of visits for banners
- I The frequency of update of the tracking statistics needs to be specified [ARCHITECT]
  - O The updates can be made on a weekly basis [CUSTOMER]
    - A3 The system will be designed to provide weekly statistics [ARCHITECT]

I The frequency of update of the tracking statistics needs to be specified [ARCHITECT]

W13 [LHF] Input of banner contents to admin via web based interface
- A19 The system will provide a web based interface for the admin [ARCHITECT]

A19 The system will provide a web based interface for the admin [ARCHITECT]

[LHF] Update links to physical bookstore site

ator cannot keep track of the links associated with all the university bookstore sites

inistrator]

ministrator

Agreements are captured as subheadings to options and win conditions

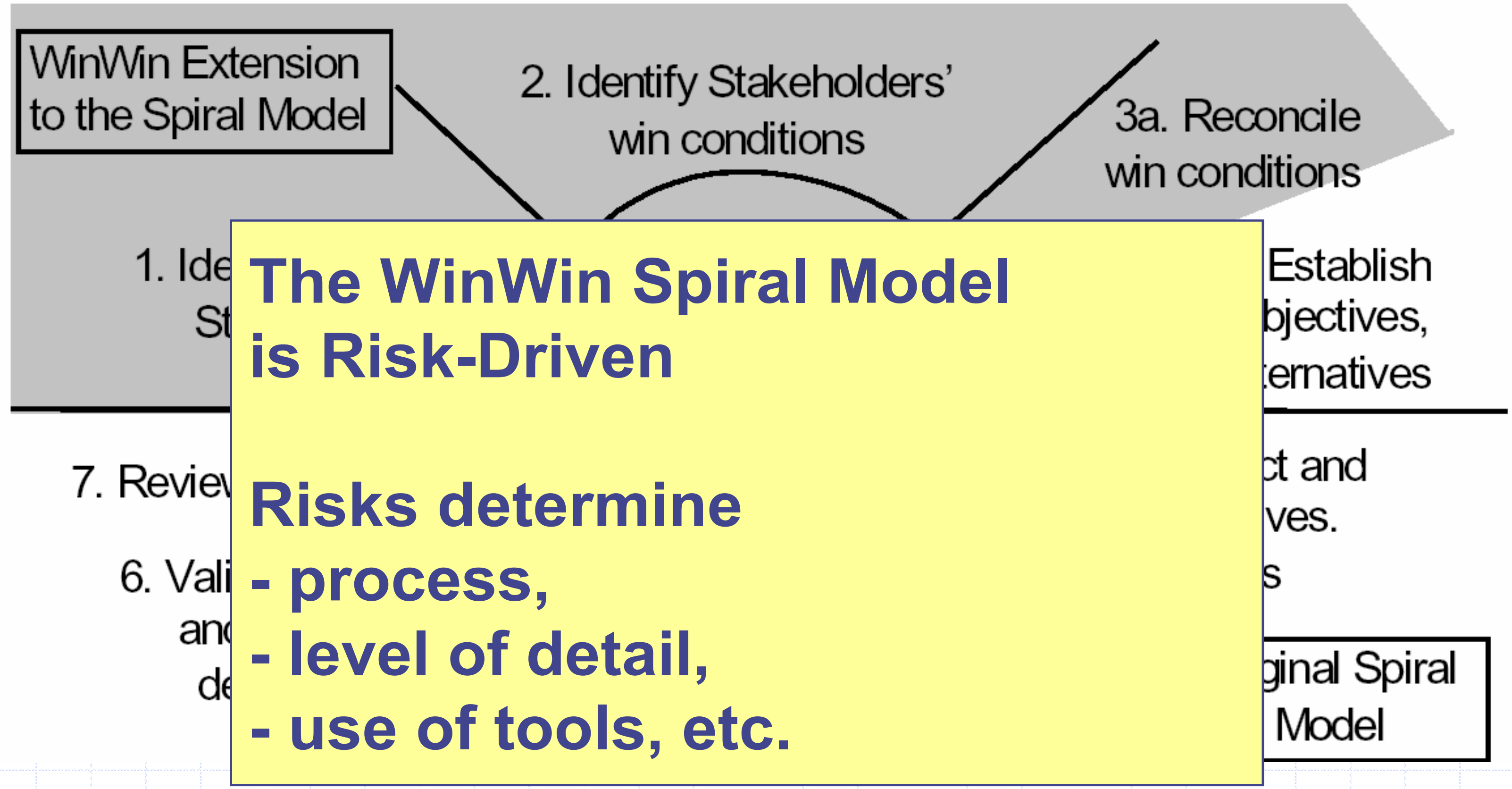
Next Generation Tool Support: EasyWinWin Cognito

- Emphasizes Distributed Requirements Negotiation
- Delivered as ActiveMethod on Cognito
- Based on EasyWinWin Process
- Internet-enabled
- Scalable Architecture

http://www.groupsystems.com

COGNITO

The WinWin Spiral Model


WinWin Extension to the Spiral Model
2. Identify Stakeholders' win conditions
3a. Reconcile win conditions
Establish
bjectives,
ernatives
ct and
ves.
s
ginal Spiral Model
1. Ide
St
7. Reviev
6. Vali
anc
de

The WinWin Spiral Model is Risk-Driven

Risks determine
- process,
- level of detail,
- use of tools, etc.

Why Use WinWin ?

- The alternatives don’t work.
  - Win-lose often leads to lose-lose.
- Avoids costly rework.
  - 100X cost to fix requirements after delivery.
- Builds trust and manages expectations.
  - Looking out for other’s needs builds trust.
  - Balancing needs leads to realistic expectations.
- Helps stakeholders adapt to change.
  - Shared vision and the flexibility of quick re-negotiation.

Typen von Anforderungen

Paul Grünbacher

Es gibt aber noch andere Anforderungen ...

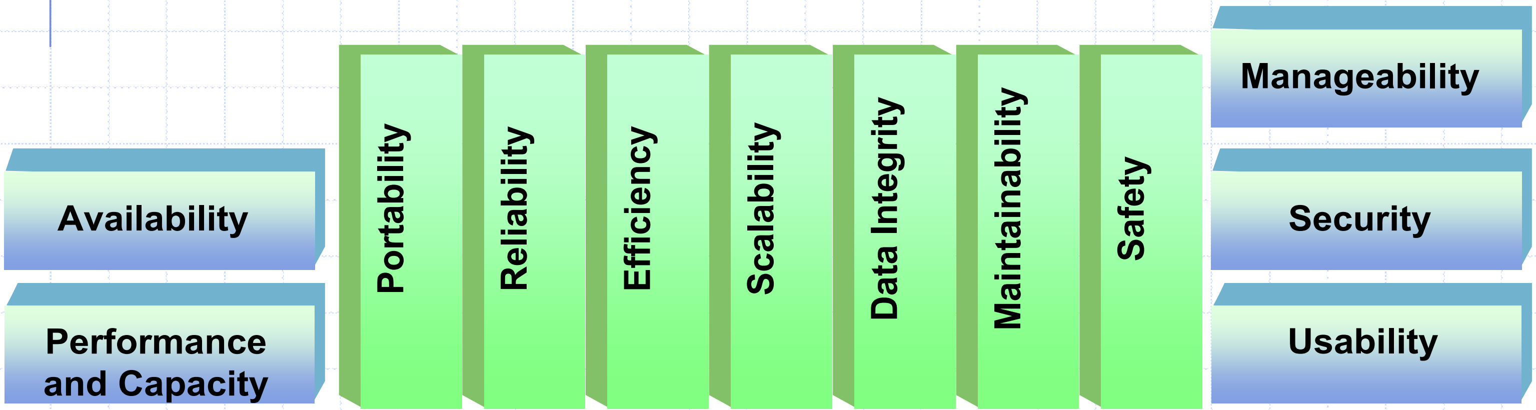
- Nicht-funktionale Anforderungen
  - Performance (Antwortzeit, Durchsatz)
  - Skalierbarkeit (Auslastung, Kapazität)
  - Verfügbarkeit (24 x 7)
  - Wartbarkeit, Flexibilität, Portabilität
  - Sicherheit
  - Verwaltbarkeit, Betreibbarkeit
  - Datenintegrität
- Anforderungen an die Benutzbarkeit
  - Interaktion
    - Anpassbarkeit, Effizienz, Integration, Navigation, Vorhersagbarkeit, Konzeptuelle Integrität, Konsistenz, Toleranz, Feedback, Antwortzeiten, Kontrolle
  - Anzeige
    - Lesbarkeit, Verständlichkeit, Hinweise, Layout, Konsistenz

Qualitäten

- Nur teilweise durch Service-Level Anforderungen beschreibbar
- Können sich aber auch auf Entwicklung, Wartung und Betrieb beziehen

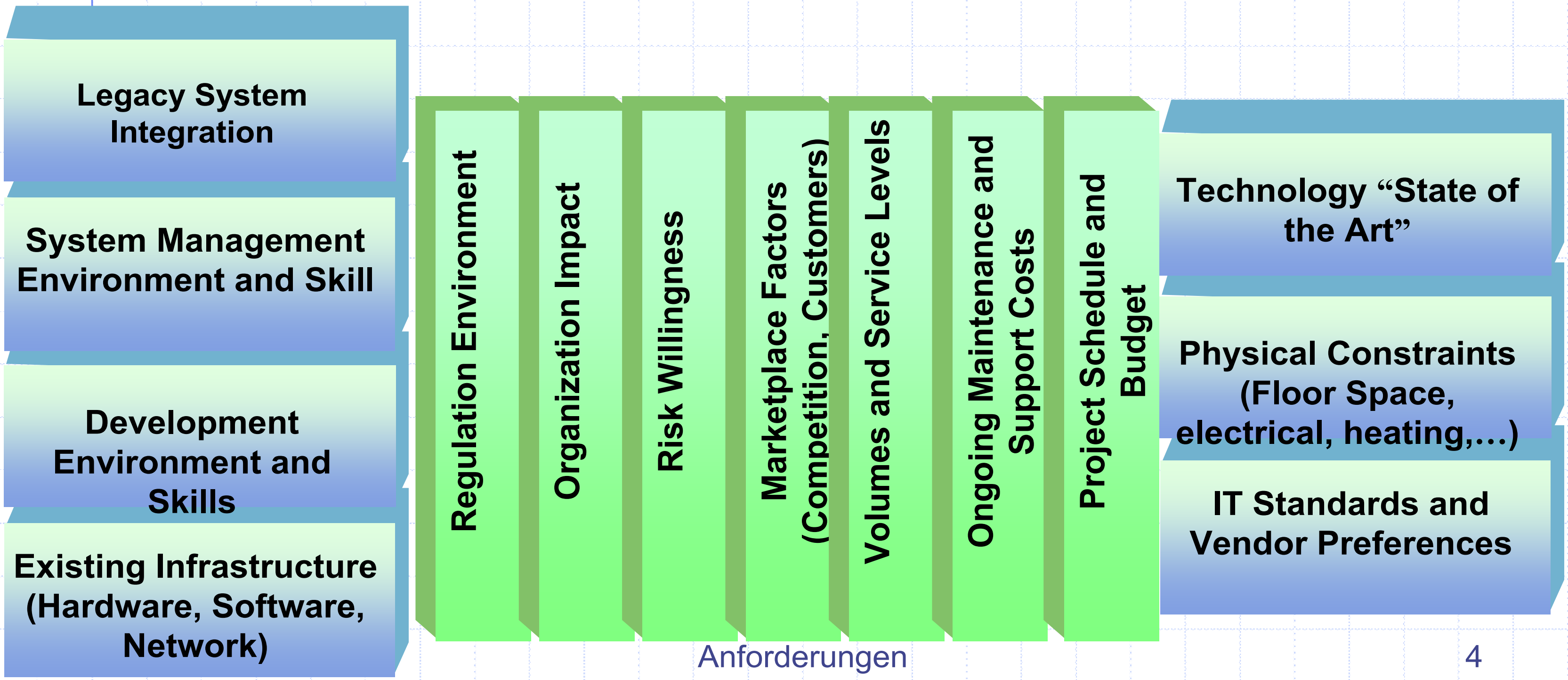
Runtime

Non Runtime



Constraints

- Entstehen aus geschäftlichen Aspekten, dem Geschäftsumfeld oder der IT-Organisation
- Entstehen auch aus dem technischem Umfeld, in dem das System betrieben wird.



Legacy System Integration

System Management Environment and Skill

Development Environment and Skills

Existing Infrastructure (Hardware, Software, Network)

Regulation Environment

Organization Impact

Risk Willingness

Marketplace Factors (Competition, Customers)

Volumes and Service Levels

Ongoing Maintenance and Support Costs

Project Schedule and Budget

Technology "State of the Art"

Physical Constraints (Floor Space, electrical, heating,…)

IT Standards and Vendor Preferences

Types of Requirements

(Example from Volere Template v9)

PROJECT DRIVERS:
1. The Purpose of the Product
2. Client, Customer, Stakeholders
3. Users of the Product

PROJECT CONSTRAINTS:
4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions

FUNCTIONAL REQUIREMENTS:
7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements

NON-FUNCTIONAL REQUIREMENTS:
10. Look and Feel
11. Usability
12. Performance
13. Operational
14. Maintainability and Portability
15. Security
16. Cultural and Political
17. Legal

PROJECT ISSUES:
18. Open Issues
19. Off-the-shelf Solutions
20. New Problems
21. Tasks
22. Cutover
23. Risks
24. Costs
25. User Documentation
26. Waiting Room
27. Ideas for Solutions

http://www.systemsguild.com/GuildSite/Robs/Template.html

Non-functional Requirements

10. Look and Feel Requirements
11. Usability Requirements
12. Performance Requirements
13. Operational Requirements
14. Maintainability and Portability Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements

Look and Feel Requirements

- The interface
- The style of the product

Usability Requirements

- Ease of use
- Personalization and internationalization requirements
- Ease of learning
- Accessibility requirements

Performance Requirements

- Speed and latency requirements
- Safety critical requirements
- Precision requirements
- Reliability and Availability requirements
- Robustness requirements
- Capacity requirements
- Scalability or extensibility requirements

Operational Requirements

- Expected physical environments
- Expected technological environment
- Partner applications
- Productization Requirements

Maintainability and Support Requirements

- How easy must it be to maintain this product?
- Are there special conditions that apply to the maintenance of this product?
- Supportability
- Portability requirements

Security Requirements

- Access requirements
- Integrity requirements
- Privacy requirements
- Audit requirements
- Immunity requirements

Cultural and Political Requirements

- Are there any special factors about the product that would make it unacceptable for some political reason?

Legal Requirements

- Does the system fall under the jurisdiction of any law?
- Are there any standards with which we must comply?

Volere

Requirements Specification Template

Edition 10.1

The first edition of the Volere Requirements Template was released in 1995. Since then organizations all over the world (see experiences of Volere users at http://www.volere.co.uk) have saved time and money by using the template as the basis for discovering, organizing and communicating their requirements.

You can download the template, try it and decide whether or not it's right for your project. If you like it, you pay a nominal **shareware** fee (Euro €40, US$50, GBP£30 AUD$70 or the equivalent) to entitle your project to continue using the template. Academic institutions and students are exempt from the shareware fee.

You can pay you shareware fee by sending a cheque (or check if you prefer) to:
The Atlantic Systems Guild Limited
11 St Mary's Terrace
London W2 1SU
United Kingdom

or in the United States to:
The Atlantic Systems Guild Inc.
353 West 12th Street
New York NY 10014
United States

This template developed by:
James & Suzanne Robertson
Principals of the Atlantic Systems Guild
London, Aachen & New York
Email james@systemsguild.com suzanne@systemsguild.com

Updates to this template are posted on our web sites
http://www.systemsguild.com   http://www.volere.co.uk

The .... System

Requirements Specification

Version ...

Table of Contents

PROJECT DRIVERS

PROJECT CONSTRAINTS

FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENTS

PROJECT ISSUES

Specification prepared by .... Date ....

Preamble

*This is a **template** for a requirements specification. Select all the sections that apply to your project, and replace the example entries with your own text. Delete any sections that are not relevant. Add any applicable new sections, and any facts that are specific to your product.*

Volere

Volere is the result of many years of practice, consulting and research in requirements engineering. We have packaged our experience in the form of a generic requirements process, requirements training, requirements consultancy, requirements audits, a variety of downloadable guides and this requirements template. We also provide requirements specification writing services.

The Volere requirements process is described in the book:

Mastering the Requirements Process by Suzanne Robertson and James Robertson, Addison-Wesley, London, 1999.

ISBN is 0-201-36046-2

Volere for managers, team leaders and advanced analysts is covered in the book:

Requirements-Led Project Management: Discovering David's Slingshot by Suzanne Robertson and James Robertson, Addison-Wesley, London, 2005.

ISBN is 0-321-18062-3

Public seminars on Volere are run on a regular basis in Europe, United States and Australia. For a schedule of courses, refer to http://www.systemsguild.com

In house seminars and consulting on Volere can be arranged on demand.

For further information contact: The Atlantic Systems Guild, 11 St Mary's Terrace, London, W2 1SU, United Kingdom.

email: suzanne@systemsguild.com james@systemsguild.com

web: http://www.systemsguild.com

web: http://www.volere.co.uk

Requirements Types

Functional requirements are the fundamental or essential subject matter of the product and are measured by concrete means like data values, decision-making logic and algorithms.

Non-functional requirements are the behavioral properties that the specified functions must have, such as performance, usability, etc. Non-functional requirements can be assigned a specific measurement. This template gives examples of quantified non-functional requirements.

Project constraints identify how the eventual product must fit into the world. For example the product might have to interface with or use some existing hardware, software or business practice, or it might have to fit within a defined budget or be ready by a defined date.

Project drivers are the business- related forces. For example the purpose of the project is a project driver, as are all of the stakeholders – each for different reasons.

Project issues define the conditions under which the project will be done. Our reason for including these as part of the requirements is to present a coherent picture of all the factors that contribute to the success or failure of the project and to illustrate how managers can use requirements as input to managing a project.

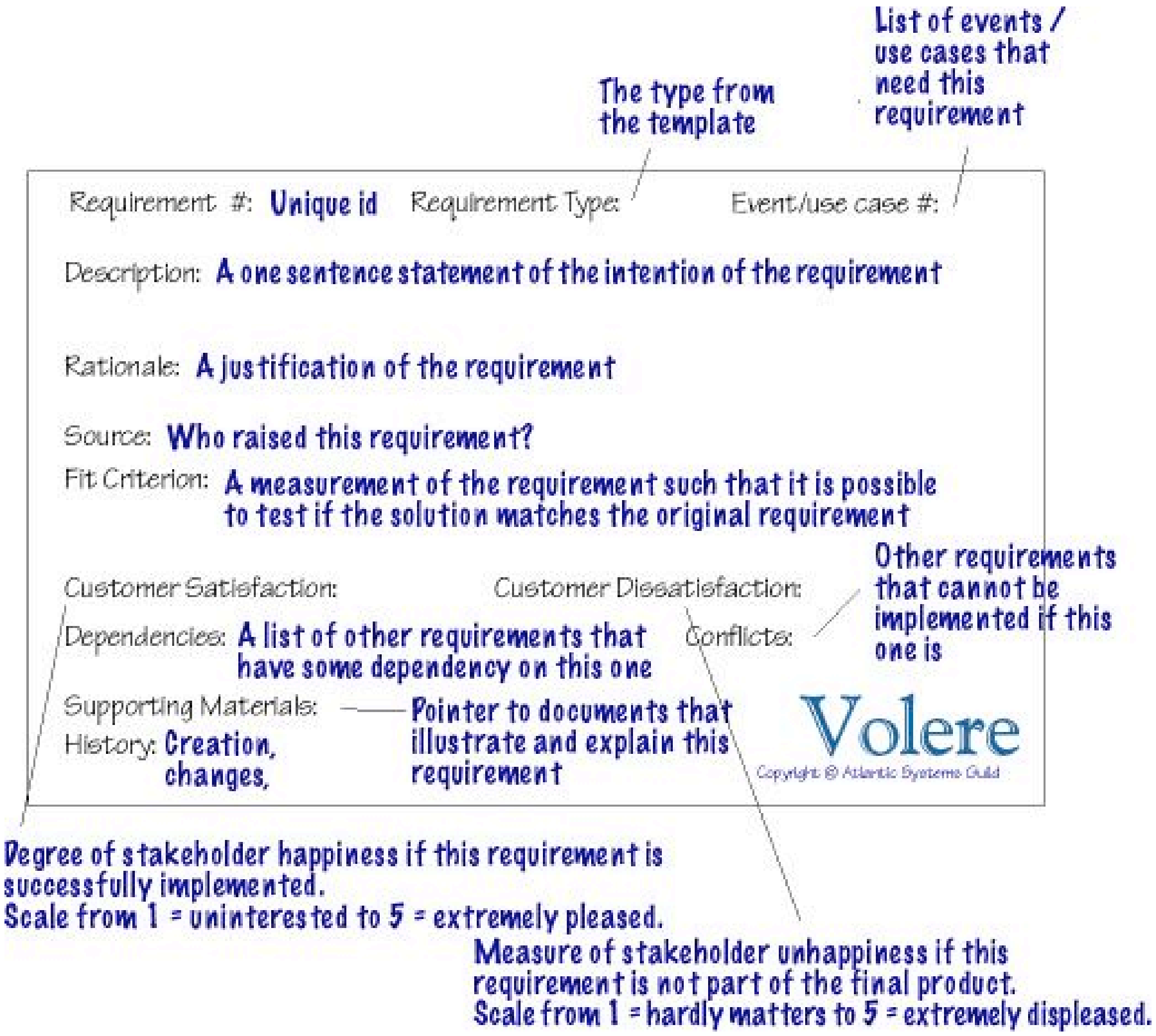
Testing requirements

You start testing requirements as soon as you start writing them.

Your first test is to determine if you can quantify the requirement by specifying its fit criterion. This *fit criterion* is an objective measure of the requirement's meaning; it is the criterion for evaluating whether or not a given solution fits the requirement. If a fit *criterion* cannot be adequately specified, then the requirement is ambiguous, or ill understood. If there is no fit *criterion*, then there is no way of knowing whether a solution meets the requirement.

Requirement Shell

Use this requirement shell as a guide for writing each atomic requirement.



List of events / use cases that need this requirement

The type from the template

Requirement #: Unique id Requirement Type: Event/use case #:

Description: A one sentence statement of the intention of the requirement

Rationale: A justification of the requirement

Source: Who raised this requirement?

Fit Criterion: A measurement of the requirement such that it is possible to test if the solution matches the original requirement

Customer Satisfaction: Customer Dissatisfaction:

Dependencies: A list of other requirements that have some dependency on this one

Conflicts:

Other requirements that cannot be implemented if this one is

Supporting Materials: Pointer to documents that illustrate and explain this requirement

History: Creation, changes,

Volere

Copyright © Atlantic Systems Guild

Degree of stakeholder happiness if this requirement is successfully implemented.
Scale from 1 = uninterested to 5 = extremely pleased.

Measure of stakeholder unhappiness if this requirement is not part of the final product.
Scale from 1 = hardly matters to 5 = extremely displeased.

Requirement Numbering

Give each requirement a unique identifier to make it traceable throughout the development process. The numbering scheme suggested in the requirement shell is:

Requirement # is the next unique requirement number

Requirement Type is the section number from the template that corresponds to this type of requirement

The inclusion of the section number is not absolutely necessary because each requirement has a unique requirement id. However it serves as a reminder of what this requirement relates to and helps to remind why the requirement is considered important. Also the ability to compare requirements of the same type makes it easier to identify contradictions and duplications.

For example:

A functional requirement is section 9, and the next unique number is 128.

Requirement #: 128 Requirement Type: 9

The product shall record the time when we are notified of a truck breakdown

A performance requirement comes from section 12, and the next unique number is 129.

Requirement #: 129 Requirement Type: 12

The product shall inform truck drivers of their schedule 30 minutes before leaving the depot.

Event/use case

Event # is the identifier of a Business Event/s whose response (or business use case) has this requirement.

Use Case # is the number of the Product Use Case/s that contain this requirement. There might be several Event/use case #'s for one requirement because the same requirement might relate to a number of events.

The terms business event and use case are already widely used in the systems development world.

The term business event means a business related happening that causes an event-response (or business use case) within the scope of the work that we are studying.

The term event-driven use case (or product use case) means a user-defined (or actor defined) piece of activity within the context of the product. Each product use case is connected to a business event. Business events and product use cases provide a way of grouping business-related requirements and tracing them through into implementation; they are used throughout the Volere development process.

Description

The requirement description is a one sentence summary of the requirement. The most common form of writing the description is:

The product shall do a specific thing for a specific person.

Rationale

The rationale explains why the requirement is considered to be important. The act of writing the rationale often serves as a tool for helping people to discover the real intention and hence the real requirement.

Fit Criterion

This *fit criterion* is an objective measure of the requirement’s meaning; it is the criterion for evaluating whether or not a given solution fits the requirement.

Customer Value

Customer Value is a measure of how much your client cares about each requirement.

Ask your stakeholders to grade each requirement for Customer Satisfaction on a scale from 1 to 5 where 1 means mild interest if this requirement is satisfactorily implemented, and 5 means they will be very happy if this requirement is satisfactorily implemented

The stakeholders also grade each requirement for Customer Dissatisfaction on a scale from 1 to 5 where 1 means that it hardly matters, and 5 means that they will be extremely displeased if this requirement is not satisfactorily implemented

The point of having a satisfaction and a dissatisfaction rating is that it guides your clients to think of the requirement from two different perspectives, and helps you to uncover what they care about most deeply. Another advantage is that you are managing expectations by reminding your clients that it might be necessary for them to prioritise requirements if you cannot implement all of them.

Dependencies

This keeps track of other requirements that have an impact on this requirement.

If the dependency exists because requirements use the same information, then use of standard naming conventions and definitions (see Section 5) will keep track of this dependency.

Other dependencies exist because a solution to this requirement has a positive or negative effect on solutions to other requirements. Another dependency occurs when the implementation of one requirement cannot be done without the implementation of other requirements. Capture these types of dependencies by cross-referencing the requirements.

Some requirements, especially project drivers and project constraints, have an impact on all the other requirements.

Conflicts

This keeps track of other requirements that disagree with this one. Conflicts that are caused by mistake are solved simply by bringing them to the surface and resolving them. Other conflicts are because of true differences in opinion/intention. These are the conflicts that might eventually need to be addressed using negotiation or mediation techniques. There is nothing wrong with having conflicting requirements providing you know that you have them. Then you are in a position to address the conflict.

History

We follow the requirement from the date that it was created, through all its changes. We minimize future confusion by recording the rationale for making major changes. When a requirement is deleted we record when, and the rationale behind the deletion. The date that the requirement passes its quality checks, and who passed it, is also recorded.

Definitions used in this template

Context of the Product

The boundaries between the product that we intend to build and the people, organizations, other products and pieces of technology that have a direct interface with the product.

Context of the Work

The subject matter, people and organizations that might have an impact on the requirements for the product. The context of study identifies the intersection of all the domains of interest.

Client

The person or organization for which the product is being built. The client is usually responsible for paying for the development of the product.

Customer

The person or organization who will buy the product (note that the same person/organization might play both the client, customer and sometimes user roles). In the case of internal customers we often say that they "buy into" the product. In other words they are not actually paying money but they support the product because it satisfies their needs.

Design or Systems Design

Crafting a solution to fit the requirements.

Developers

The people who specify and build the product.

Event

We use the term business event to mean a business related happening within a system adjacent to the work that we are studying. The happening causes the work to produce an event-response.

Fit Criterion

Objective measure for quantifying the meaning of a requirement, and eventually testing whether a given solution satisfies the original requirement.

Functional Requirement

An action that the product must be able to take, something that the product must do.

Global Constraint

Constraints that apply to the project as a whole.

Non-Functional Requirement

A property of quality that the eventual product must have.

Product

This is what we are attempting to deliver. This could be a piece of software, the installation of a package, a set of procedures, a piece of hardware, a piece of machinery, a new consumer product, a new organization, or almost anything.

Requirement

A measurable statement of intent about something that the product must do: or a property that the product must have: or a constraint on the system.

Stakeholder

A stakeholder is a person or organisation who has some demand on the product and/or is affected by its outcome/success.

System

The business system or work in the world, whose requirements are being studied.

Systems Analysis

Detailed study of the requirements, intended to prove their workability (usually through building models) as input to systems design.

Use case

We use the term product use case to mean a user-defined (or actor defined) piece of activity within the context of the product. We also use the term business use case to refer to the business' response to a business event.

User or Hands-on User

Someone who has some kind of direct interface with the product.

1 The Purpose of the Project

1a. The user problem or background of the project effort.

Content

A short description of the work context and the situation that triggered the development effort. It should also describe the work that the user wants to do with the delivered product.

Motivation

Without this statement, the project lacks justification and direction.

Considerations

You should consider whether or not the user problem is serious, and whether and why it needs to be solved.

1b. Goals of the project.

Content

This boils down to one, or at most a few, sentences that say "What do we want this product for?" In other words, the real reason that the product is being developed.

Motivation

There is a real danger of this purpose getting lost along the way. As the development effort heats up, and the customer and developers discover more and more what is possible, it may well be that the system as it is being constructed wanders away from the original goals. This is a bad thing unless there is some deliberate act by the client to change the goals. It may be necessary to appoint a person to be "custodian of the goals", but it is probably sufficient to make the goals public, and periodically remind the developers of it. *It should be mandatory to acknowledge the goals at every review session.*

Examples

"We want to give immediate and complete response to customers ordering our goods over the telephone."

"We want to be able to forecast the weather."

Fit Criterion

An objective measure that will enable testing to determine if the goal has been met by the product.

Some guideline for making goals measurable are:

- Specify each adverb and adjective so that everyone on the project understands the same meaning.

- Replace pronouns with the names of specific people or organisations.

- Ensure that the meaning of every noun is defined in one place in the specification (Section 5 of this template)

For instance the above example could be analysed and made less ambiguous as follows:

We - Employees of XYZ Corporation

want to give

immediate - during the course of a telephone call

and

complete - product availability and price

response - verbal information

to

customers - anyone who enquires about our products

our - supplied by XYZ Corporation

goods - products that we manufacture

over the telephone

Following the analysis, the goal could be restated as:

Employees of XYZ Corporation want to tell enquirers, during the course of a telephone call, the availability and price for any product manufactured by XYZ.

Whenever you analyze a goal using this technique you will find yourself going through several iterations. The discipline imposed by making the goal measurable guides you into asking more relevant questions about the meaning.

You can use the Volere Purpose, Advantage, Measurement questioning technique to help you make goals measurable.

2 Client, Customer and other Stakeholders

2a. The client is the person/s paying for the development, and owner of the delivered product.

Content

This item must give the name of the client. It is permissible to have several names, but more than three negates the point.

Motivation

The client has the final acceptance of the product, and thus must be satisfied with the product as delivered. You can think of the client as the person who is making the investment in the product. Where the product is being developed for in-house consumption, the roles of the client and the customer are often filled by the same person. If you cannot find a name for your client, then perhaps you should not be building the product.

Considerations

Sometimes, when building a package or a product for external users, the client is the marketing department. In this case, a person from the marketing department must be named as the client.

2b. The customer is the person/s who will buy the product.

Content

The name of the person who plays the role of the customer for the product. In the case of in house development the roles of the client and the customer are often played by the same person. In the case of the development of a mass market product there may be several people playing the role of customer. In the case of a product that is being developed for an international market, there might be a different customer (or customer profile) in each country.

Motivation

The customer role is ultimately responsible for deciding whether or not to buy the product from the client. *The product must be built to satisfy the aims of the customer/s whilst conforming to the constraints of the client*. Even if the customer/s are people who work for another part of the client's organization, they might still have the authority to decide whether or not to invest budget in the new product.

2c. Other stakeholders

Content

The roles and (if possible) names of other people and organizations who are affected by the product, or whose input is needed in order to build the product.

Examples of stakeholders include:

Users (detailed in section 3)

Sponsor

Testers

Business Analysts

Technology Experts

System Designers

Marketing Experts

Legal Experts

Domain Experts

Usability Experts

Representatives of external associations

For a complete checklist, download the stakeholder analysis template at http://www.volere.co.uk

For each type of stakeholder identify:

Stakeholder Identification (some combination of role/job title, person name, organization name),

Knowledge needed by the project,

Necessary degree of involvement for that stakeholder/knowledge combination,

Degree of influence for that stakeholder/knowledge combination,

Agreement on how to address conflict between stakeholders who have an interest in the same knowledge

Motivation

Failure to recognize stakeholders results in missing requirements.

3 Users of the Product

3a. The hands-on users of the product

Content

A list of the potential users of the product. For each category of user, provide the following information:

User name/category – This is most likely to be the name of a user group like: schoolchildren, road engineers, project managers.

User role – Summarizes the users' responsibilities.

Subject matter experience – Summarizes the users' knowledge of the business. Rate as novice, journeyman or master.

Technological experience – this describes the users' experience with relevant technology. Rate as novice, journeyman or master.

Other user characteristics – Describe any characteristics of the users that have an effect on the requirements and eventual design of the product. Describe things like:

Physical abilities/disabilities

Intellectual abilities/disabilities

Attitude to job

Attitude to technology

Education

Linguistic skills

Age group

Gender

Motivation

Users are human beings or other pieces of technology who interface with the product in some way. The role of the client is to pay for the development of the product and the role of the customer is to buy the product. The role of the user is to use the product to do work. You use the characteristics of the users to define the usability requirements for the product. You can also refer to Users as Actors.

Examples

Users can come from wide, and sometimes unexpected, sources. Consider the possibility of your users being clerical staff, shop workers, managers, highly-trained operators, general public, casual users, passers-by, illiterate people, tradesmen, students, test engineers, foreigners, children, lawyers, remote users, people using the system over the telephone or Internet connection, emergency workers, and so on.

3b. The priorities assigned to users

Content

Attach to each category of user a priority rating. This gives the importance and precedence of the user. Prioritize the users into:

Key users. These are critical to the continued success of the product. Give greater importance to requirements generated by this category of user.

Secondary users. They will use the product, but their opinion of it has no effect on its long-term success. Where there is a conflict between secondary users' requirements and those of key users the key users take precedence.

Unimportant users. This category of user is given the lowest priority. It includes infrequent, unauthorized and unskilled users, and people who misuse the product.

Percentage of this type of user – this is intended to assess the amount of consideration given to this category of user.

Motivation

If some users are considered to be more important to the product, or the organization, then this should be stated because it should affect the way that you design the product. For instance, you need to know if there is a large customer who has specifically asked for the product, and if they do not get what they want then the results could be a significant loss of business.

Some users may be listed as having no impact on the product. This means that the users will make use of the product, but have no vested interest in it. In other words, these users will not complain, nor will they contribute. Any special requirements from these users will have a lower design priority.

3c. User participation

Content

Where appropriate attach to the category of user, a statement of the participation that you think will be necessary for them to provide the requirements. Describe the contribution that you expect this user to provide – business knowledge, interface prototyping, usability requirements etc. If possible, assess the minimum amount of time that this user must spend for you to be able to determine the complete requirements.

Motivation

Many projects fail through lack of user participation, sometimes this is because the required degree of participation was not made clear. When people have to make a choice between getting their everyday work done and working on a new project, the everyday work takes priority. This requirement makes it clear, from the outset, that specified user resources must be allocated to the project.

3d. Maintenance users

Content

Maintenance users are a special type of hands-on user who have requirements that are specific to maintaining and changing the product.

Motivation

Many of these requirements will be discovered by considering all the different types of maintenance requirements detailed in section 14. However if we define the characteristics of the people who maintain the product it will help to trigger requirements that might otherwise be missed.

4 Mandated Constraints

This section describes constraints on the requirements that will affect the eventual design of the product. Note that constraint requirements will have a description, rationale and fit criterion. They are specified in the same form as functional and non-functional requirements.

4a. Solution design constraints

Content

This specifies constraints on the way that the problem must be solved. These are another type of requirement. You can think of them as mandated solutions. Carefully describe the mandated technology, include the appropriate version numbers, and the fit criterion for how you will test compliance. If possible, you should also explain the reason for using the technology.

Motivation

To identify constraints that must be part of the final product. Your client, customer or user may have design preferences. If these are not met then your solution is not acceptable.

Examples

Constraints are written using the same form as other atomic requirements (refer to the requirements shell for the attributes). One difference is that when you write a constraint you say the product *must* – in other words there is no negotiation. It is especially important for each constraint to have a rationale and a fit criterion as this helps to expose false constraints (solutions masquerading as constraints). Also, you will usually find that a constraint affects the entire product rather than one or more product use cases.

The product *must* use the current 2-way radio system to communicate with the drivers in their trucks.

The product *must* use the Windows NT operating system.

The product *must* be a hand-held device.

Considerations

We want to define the boundaries within which we can solve the problem. Be careful because anyone who has experience/exposure to a piece of technology tends to see requirements in terms of that technology. This tendency leads people to impose solution constraints for the wrong reason and it's very easy for false constraints to creep into a specification. If you impose untrue constraints the danger is that you do not have the creative freedom to come up with the best solution to the problem. The solution constraints should only be those that are absolutely non-negotiable. In other words, however you solve this problem you must use this particular technology. Any other solution would be unacceptable.

4b. Implementation environment of the current system

Content

This describes the technological and physical environment in which the product will be installed. This includes automated, mechanical, organizational and other devices. These include the non-human adjacent systems.

Motivation

To describe the technological environment into which the product must fit. The environment places design constraints on the product. This part of the specification provides enough information about the environment for the designers to make the product successfully interact with its surrounding technology.

The operational requirements are derived from this description.

Examples

This can be shown as a diagram, with some kind of icon to represent each separate device or person (processor). Draw arrows to identify the interfaces between the processors and annotate them with their form and content .

Considerations

All the component parts of the current system, regardless of their type, should be included in the description of the implementation environment.

If the product is to affect, or be important to the current organization, include an organization chart.

4c. Partner or collaborative applications

Content

This describes applications that are not part of the product but with which the product will collaborate. These can be external applications, commercial packages or pre-existing in-house applications.

Motivation

To provide information about design constraints that are caused by using partner applications. By describing or modeling these partner applications, you discover and highlight potential problems of integration.

Examples

This section can be completed by including written descriptions, models or references to other specifications. The descriptions must include a full specification of all interfaces that will have an effect on the product.

Considerations

Examine the work context model to determine if any of the adjacent systems should be treated as partner applications. It might also be necessary to examine some of the details of the work to discover relevant partner applications.

4d. Off-the-shelf software

Content

This describes applications that must be used to implement some of the requirements for the product.

Motivation

To identify and describe existing commercial, free, open source, etc. products that must be incorporated into the eventual product. The characteristics, behavior and interfaces of the package are design constraints.

Examples

This section can be completed by including written descriptions, models or references to supplier's specifications.

Considerations

The use of a specific package has been mandated. When gathering requirements you may discover requirements that are in serious conflict with the behavior and characteristics of the package. Keep in mind that the use of the package was mandated before the full extent of the requirements was known. In light of your discoveries you must consider whether the package is a viable choice when all the requirements are known. If the use of the package is not negotiable, then the conflicting requirements will have to be discarded.

Note that your strategy for discovering requirements is affected by the decision to use a COTS product. In this situation you investigate the work context in parallel with making comparisons with the capabilities of the COTS product. Depending on the comprehensibility of the COTS product, you might be able to discover the matches/mismatches without having to write each of the business requirements in atomic detail. The mismatches are the requirements that you will need to specify so that you can decide whether to satisfy them by either modifying the COTS product or modifying the business requirements.

You should also consider if there are any legal implications arising from your use of the software. You can cover this in section 17 – Legal Requirements.

4e. Anticipated workplace environment

Content

This describes the workplace in which the users will work and use the product. This should describe any features of the workplace that could have an effect on the design of the product.

Motivation

To identify characteristics of the physical workplace so that the product is designed to compensate for any difficulties.

Examples

The printer is a considerable distance from the user's desk. This constraint suggests that printed output should be de-emphasized.

The workplace is noisy, so audible signals might not work.

The workplace is outside so the product must be waterproof, have displays that are visible in sunlight and allow for the effect of wind on any paper output.

The user will be standing up or working in positions where he must hold the product. This suggests a hand-held product but only a careful study of the users' work and workplace will provide the necessary input to identifying the operational requirements.

Considerations

The physical work environment constrains the way that work is done. The product should overcome whatever difficulties exist, however you might consider a redesign of the workplace as an alternative to having the product compensate for it.

4f. How long do the developers have for the project?

Content

Any known deadlines, or windows of opportunity, should be stated here.

Motivation

To identify critical times and dates that have an effect on product requirements. If the deadline is short, then the requirements must be kept to whatever can be built within the time allowed.

Examples

To meet scheduled software releases.

There may be other parts of the business or other software products that are dependent on this product.

Windows of marketing opportunity.

Scheduled changes to the business that will use your product. For example the organization may be starting up a new factory and your product is needed before production can commence.

Considerations

State deadline limitations that exist by stating the date and describing why it is critical. Also identify prior dates where parts of your product need to be available for testing.

You should also ask questions about the impact of not meeting the deadline like:

What happens if we don’t build the product by ......?

What is the financial impact of not having, the product by…?

4g. What is the financial budget for the project?

Content

The budget for the project, expressed in money or available resources.

Motivation

The requirements must not exceed the budget. This may constrain the number of requirements that can be included in the product.

The intention of this question is to determine if the product is really wanted.

Considerations

Is it realistic to build a product within this budget? If the answer to this question is no, then either the client is not really committed to building the product or does not place enough value on the product. In either case you should consider whether it is worthwhile continuing.

5 Naming Conventions and Definitions

This section gives definitions of all terms, including acronyms, used in the project.

Content

A dictionary containing the meaning of all the names used within the requirements specification. Select names carefully to avoid giving a different, unintended meaning.

This dictionary should build on the standard names that your organization, or industry, uses. The names should also reflect the terminology in current use within the work area.

The dictionary should contain all terms that are used by the project. For each name write a succinct definition. The appropriate stakeholders must agree this definition.

Avoid abbreviations or acronyms, they introduce ambiguity: cause additional translations in the mind of anyone who is trying to understand your requirements and lead to misinterpretation.

Motivation

Names are very important. They invoke meanings that, if carefully defined, can save hours of explanations. Attention to names at this stage of the project helps to highlight misunderstandings.

The dictionary produced during requirements is used and added to throughout the project.

Examples

Gritter Truck -- a truck used for spreading de-icing substances on roads in winter.

Considerations

Make use of existing references and data dictionaries. Obviously it is best to avoid renaming existing items unless they are so ambiguous that they cause confusion.

From the start of the project emphasize the need to avoid homonyms and synonyms and explain how they increase the cost of the project.

Later on, as the analysis progresses, this description will be expanded to define all the elementary terms that describe a truck.

Truck = Truck Registration Number, Truck Capacity, Truck Service Status

As we progress through the requirements specification each of the elementary terms will be defined in detail

Truck Capacity - the number of tonnes of deicing material that can be carried by a truck. From 0.5 to 2 tonnes

The dictionary provides a link between the requirements analysts and the implementers. The implementers will add implementation details to the terms in the dictionary defining how the data will be implemented. Also, implementers add additional terms that are there because of the chosen technology and are independent of the business requirements.

6 Relevant Facts and Assumptions

6a. Factors that have an effect on the product, but are not mandated requirements constraints.

Content

Statements describing business rules, systems, and activities in the world that have an effect on this product.

Motivation

Relevant facts might contribute to requirements. They will have an effect on the eventual design of the product.

Examples

One ton of de-icing material will treat 3 miles of single lane roadway.

The existing application is 10,000 lines of C code.

6b. Assumptions that the team is making about the project

Content

A list of the assumptions that the developers are making. These might be about the intended operational environment, but can be about anything that has an effect on the product. As part of managing expectations, assumptions also contain statements about what the product will specifically *not* do.

Motivation

To make people declare the assumptions that they are making. Also to make everyone on the project aware of assumptions that have been made.

Examples

Assumptions about new laws or political decisions.

Assumptions about what your developers expect to be ready in time for them to use. For example, other parts of your products, the completion of other projects, software tools, software components, etc.

Assumptions about the technological environment in which the product will operate. These assumptions should highlight areas of expected compatibility.

The software components that will be available to the developers.

Other products being developed at the same time as this one.

Availability and capability of bought-in components.

Dependencies on computer systems or people external to this project

The requirements that will specifically *not* be carried out by the product.

Considerations

We often make unconscious assumptions. It is necessary to talk to the members of the project team to discover any unconscious assumptions that they have made. Ask stakeholders (both technical and business-related) questions like “What software tools are you expecting to be available, will there be any new software products, are you expecting to use a current product in a new way, are there any business changes you are assuming we will be able to deal with....?” It is important to state these assumptions up front. You might also consider the probability of whether or not the assumption is correct, and where relevant, a list of alternatives if something that is assumed does not happen.

The assumptions are intended to be transient. That is, they should all be cleared by the time the specification is released. In other words, the assumption should have become either a requirement or a constraint. For example, if the assumption was about the capability of a product that is intended to be a partner product to yours, then the capability should have been proven satisfactory, and thus it becomes a constraint to use it. On the other hand, if the bought-in product is not suitable, then it becomes a requirement for the project team to construct the needed capability.

7 The Scope of the Work

7a. The current situation

Content

This is an analysis of the existing business processes, and the manual and automated processes that might be replaced or changed by the new product. This investigation might already have been done by business analysts as part of the business case analysis for the project.

Motivation

If your project intends to make changes to an existing manual/automated system then you need to understand the effect of proposed changes. The study of the current situation provides the basis for understanding the effect of proposed changes and choosing the best alternatives.

7b. The context of the work.

Content

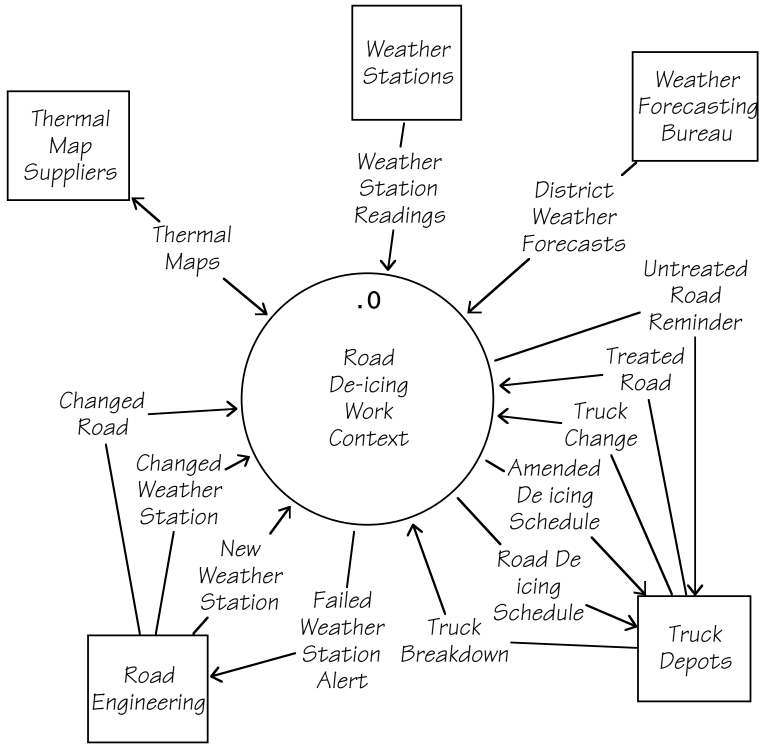
The work context diagram identifies the work that we need to investigate in order to be able to build the product. Note that this includes more than the intended product. Unless we understand the work that the product will support, there is little chance of building a product that will fit cleanly into its environment.

The adjacent systems on the example context diagram e.g. Weather Forecasting Bureau, indicate other subject matter domains (systems, people and organizations) that need to be understood. The interfaces between the adjacent systems and the work context indicate why we are interested in the adjacent system. In the case of Weather Forecasting Bureau, we can say that we are interested in the details of when, how, where, who and why they produce the District Weather Forecast information.

Motivation

To clearly define the boundaries for the work study and requirements effort. Without this definition, there is little chance of building a product that will fit seamlessly into its environment.

Examples



Considerations

The names used on the context diagram should be consistent with the naming conventions discussed in section 5.

7c. Work partitioning

Content

An event list, identifying all the business events to which the work responds. The business events are user-defined. The response to each event (the business use case) represents a portion of work that contributes to the total functionality of the work.

The event list includes:

Event Name

Input from other systems (identical with name on context diagram)

Output from other systems (identical with name on context diagram)

Internal objects/entities that are connected to this business event. For example, both events 8 and 9 would be connected to an internal object called road. In other words there is a need within the context to remember information about roads and that information is relevant to events 8 and 9 (and many other events as well). It is this identification of common internal objects that provides a link between events.

Motivation

To identify logical chunks of the system that can be used as the basis for discovering detailed requirements. These business events also provide the subsystems that can be used as the basis for managing detailed analysis and design.

Example

Business Event List

Event Name	Input & Output
1. Weather Station transmits reading	Weather Station Readings (in)
2. Weather Bureau forecasts weather	District weather Forecast (in)
3. Road engineers advise changed roads	Changed Road (in)
4. Road Engineering installs new weather station	New Weather Station (in)
5. Road Engineering changes weather station	Changed Weather Station (in)
6. Time to test Weather Stations	Failed Weather Station Alert (out)
7. Truck Depot changes a truck	Truck Change (in) Amended De-icing Schedule (out)
8. Time to detect icy roads	Road De-icing Schedule (out)
9. Truck treats a road	Treated Road (in)

10 Truck Depot reports problem with truck	Truck Breakdown (in) Amended Gritting Schedule (out)
11. Time to monitor road gritting	Untreated Road Reminder (out)

Considerations

Attempting to list the business events is a way of testing the work context. This activity uncovers uncertainty and misunderstanding about the project and helps with precise communications. When you do an event analysis it will usually cause you to make some changes to your work context diagram.

8 The Scope of the Product

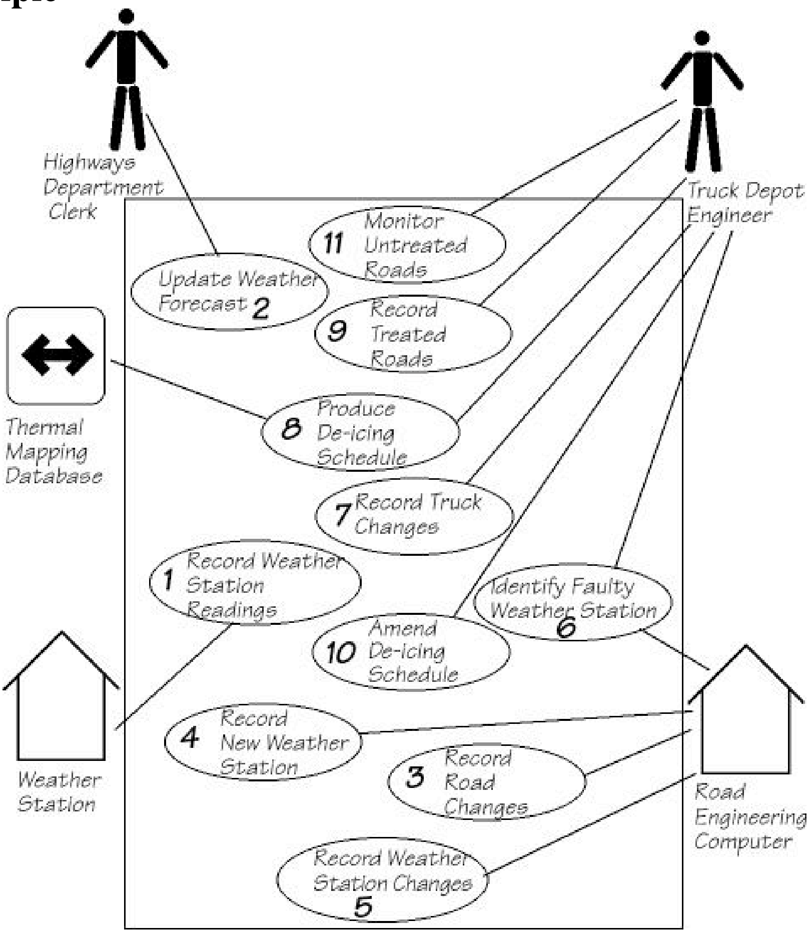
8a Product Boundary

Use case diagram identifies boundaries between the users and product. You define the boundary of each product use case by analyzing the business events and the business event response (the business use case.

For each business use case you consider: your knowledge of the business use case (this might be in many different forms – models, notes, manuals etc.), together with your knowledge of the constraints that affect this business use case, the stakeholders who are affected by this business use case, and the relevant facts and assumptions that affect this business use case.

Taking all these factors into account, you derive the product use cases by deciding where the product boundary should be in order to make the greatest contribution to the project purpose.

Example


Highways Department Clerk
Truck Depot Engineer
11 Monitor Untreated Roads
Update Weather Forecast 2
9 Record Treated Roads
Thermal Mapping Database
8 Produce De-icing Schedule
7 Record Truck Changes
1 Record Weather Station Readings
Identify Faulty Weather Station 6
10 Amend De-icing Schedule
Weather Station
4 Record New Weather Station
3 Record Road Changes
Road Engineering Computer
Record Weather Station Changes 5

You derive the product use cases by deciding where the product boundary should be for each one of the business events. These decisions are based on your knowledge of the work and the requirements constraints.

8b Product use case list

The use case diagram is a graphical way of summarizing all the use cases relevant to the product. If you have a large number of use cases, we find 15-20, is around the limit, then it is better to list the use cases and model each one individually. For each use case on the list you should have: use case number, user/actor name, use case description and use case fit criterion. Also if you have built a use case description and/or any scenario models for this use case then this list can point to them.

Use Case 8

User/actor name

Truck Depot Engineer

Description

Produce road de-icing schedule

Fit Criterion

Sensor readings shall be used to prepare a schedule for the de-icing trucks.

Use Case Scenarios

The description for this use case describes the normal way that it operates. Scenario models 8.1, 8.2, 8.3 illustrate exception cases for this use case.

Each of the individual requirements that relates to this use case will contribute to meeting the fit criterion of the use case. Each individual requirement will also have its own detailed fit criterion.

9 Functional and Data Requirements

9a. Functional Requirements.

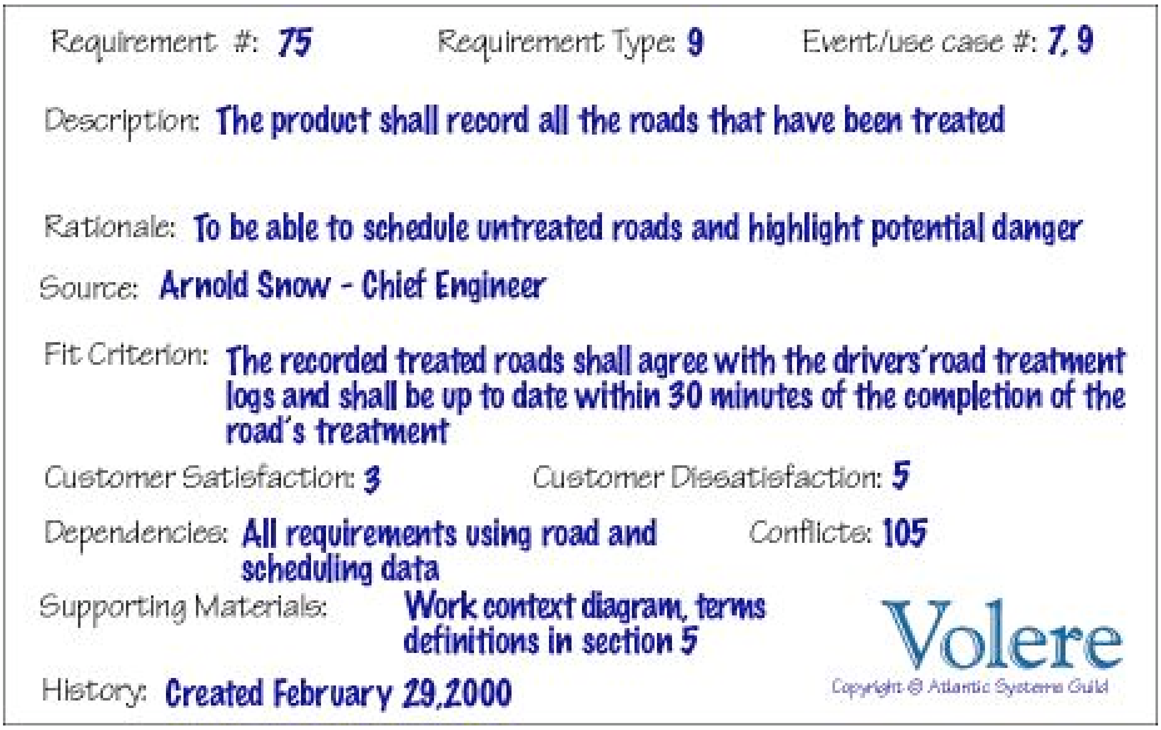
Content

A specification for each individual functional requirement. As with all types of requirements, use the *Requirements Shell.* A full explanation is included in this template's introductory material and further details are in the *Mastering the Requirements Process* textbook and course.

Motivation

To specify the detailed functional requirements that must be supported by the product.

Examples



Requirement #: 75 Requirement Type: 9 Event/use case #: 7, 9

Description: The product shall record all the roads that have been treated

Rationale: To be able to schedule untreated roads and highlight potential danger

Source: Arnold Snow - Chief Engineer

Fit Criterion: The recorded treated roads shall agree with the drivers'road treatment logs and shall be up to date within 30 minutes of the completion of the road's treatment

Customer Satisfaction: 3 Customer Dissatisfaction: 5

Dependencies: All requirements using road and scheduling data Conflicts: 105

Supporting Materials: Work context diagram, terms definitions in section 5

History: Created February 29,2000

Volere

Copyright © Atlantic Systems Guild

Fit Criterion

Each functional requirement must have a fit criterion. The fit criterion depends on the required action. For example, if the requirement is to record some data, then the fit criterion would say that the data must be able to be retrieved and must match certain standards. For calculations, the resulting data must conform to predicted results.

Considerations

If you have produced an event/use case list (see 7b & 8a) you can use them to help you trigger the functional requirements for each event/use case. If you have not produced an event/use case list, give each functional requirement a unique number and, to help with traceability, they can be partitioned into event/use case-related groups later in the development process.

9b. Data requirements.

Content

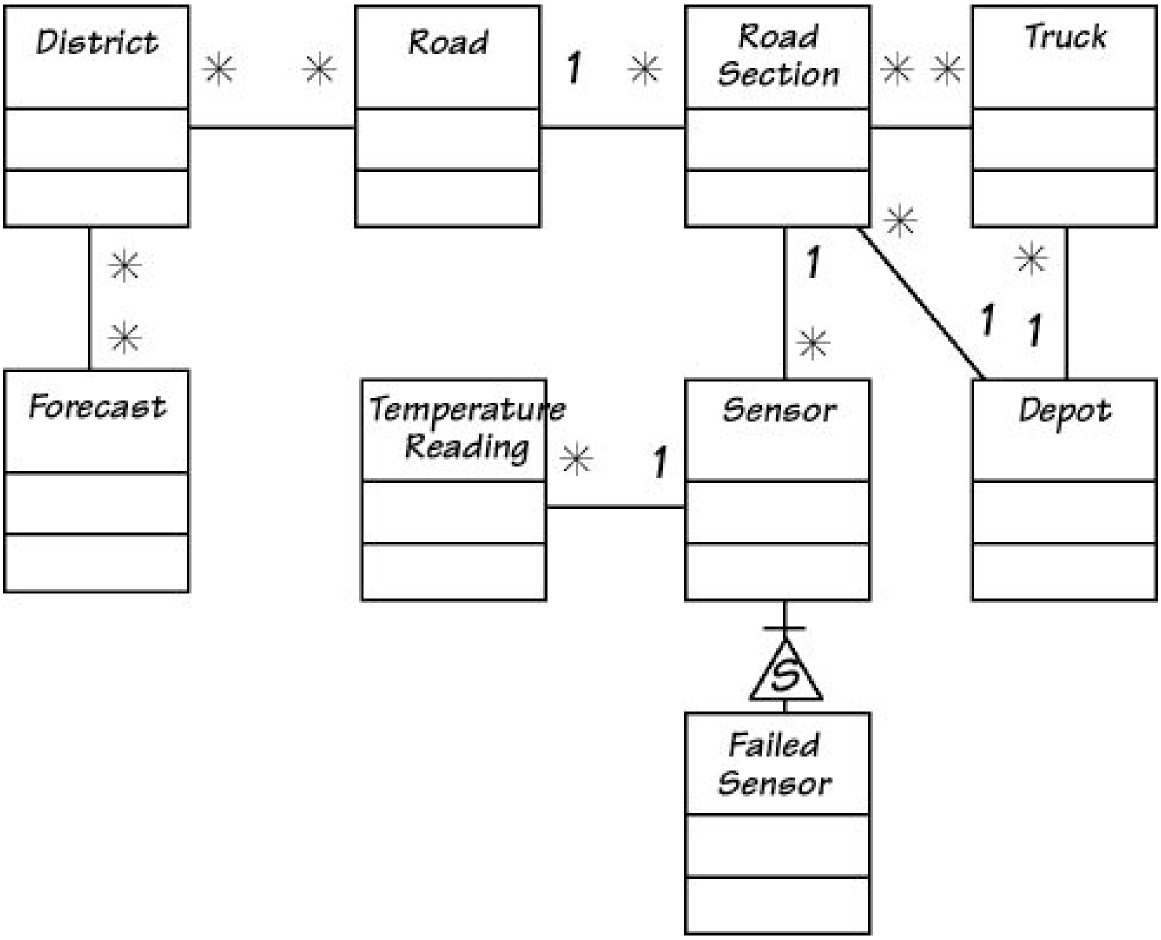
A specification of the essential subject matter/business objects/entities/classes that are germane to the system. This might take the form of a first-cut data model, an object model or a domain model. Or it might be adequately dealt with by defining the terms in the dictionary described in section 5. We have included examples of two notations for modeling business data, there are many others.

Motivation

To clarify the system's subject matter and thereby trigger requirements that have not yet been thought of.

Example 1

The following is a model of the system's business subject matter using the Unified Modeling Language (UML) class model notation.



Fit Criterion

You can use any type of data or object model to capture this knowledge. The issue is to capture the meaning of the business subject matter and the connections between the individual parts and that you are consistent within your project. If you have an established company standard notation, then use that as it will help you to reuse knowledge between projects.

To support your data model you would also define:

Name of business object/entity (use naming convention from 5)

Statement of the purpose of the class/entity

Description of relationships between classes/entities

Attributes of the object/entity (use conventions from 5)

Considerations

Are there any data/object models for similar/overlapping systems that might be a useful starting point? Is there a domain model for the subject matter dealt with by this system?

10 Look and Feel Requirements

10a. The interface

Content

The section contains requirements relating to spirit of the interface. Your client may have given you particular demands such as corporate branding, style, colors to be used, degree of interaction and so on. This section captures the requirements for the interface rather than the design for the interface.

Motivation

To ensure that the appearance of the product conforms to the organization's expectations.

Examples

"The product shall comply with corporate branding standards."

"The product shall be attractive to a teenage audience."

"The product shall appear authoritative."

Considerations

Interface design may overlap the requirements gathering process. This particularly true if you are using prototyping as part of your requirements process. As prototypes develop it is important to capture the requirements that relate to the look and feel. In other words, be sure that you understand your client's intentions for the product's look and feel. Record these as requirements instead of merely having a prototype to which the client has nodded his approval.

10b. The style of the product

Content

A description of salient features of the product that are related to the way a potential customer will see the product. For example, if your client wants the product to appeal to the business executive, then a look and feel requirement is that the product has a conservative and professional appearance. Similarly if the product is for sale to children, then the look and feel requirement is that it be colorful and look like it's intended for children.

You would also consider here the design of the package if this were to be a manufactured product. The package may have some requirements as to its size, style, and consistency with other packages put out by your organization, etc. Keep in mind the European laws on packaging. There is a requirement that the package not be significantly larger than the product it encloses.

The requirements that you record here will guide the designers to produce a product as envisioned by your client.

Motivation

Given the state of today's market and people's expectations, we cannot afford to build products that have an inadequate appearance. Once the functional requirements are satisfied, it is often the appearance of products that determines whether they are successful or not. Your task in this section is to determine precisely how the product shall appear to its intended consumer.

Considerations

The look and feel requirements specify the your client's vision of the product's appearance. The requirements may at first seem to be rather vague – "conservative and professional appearance" – but these will be quantified by their fit criterion. The fit criterion in this case gives you the opportunity to extract from your client precisely what is meant, and gives the designer precise instructions on what he is to accomplish.

11 Usability and Humanity Requirements

This section is concerned with requirements that are there because of the characteristics of the hands-on users.

11a. Ease of use.

Content

This section describes your client's aspirations for how easy it will be for the intended users of the product to operate it. The product's usability is derived from the abilities of the expected users of the product and the complexity of its functionality.

The usability requirements should cover such things as:

Efficiency of use – how quickly or accurately the user can use the product.

Ease of remembering – how much is the casual user expected to remember about using the product

Error rates – for some products it is crucial that the user commits very few, or no, errors.

Overall satisfaction in using the product – this is especially important for commercial, interactive products where there is a lot of competition. Web sites are good example of this.

Feedback – how much feedback does the user need in order to feel confident that the product is actually accurately doing what the user expects. The necessary degree of feedback will be higher for some products (eg: safety critical) than in others.

Motivation

To guide the product's designers into building a product that will meet the expectations of its eventual users.

Examples

"The product shall be easy for 11 year-old children to use."

"The product shall help the user to avoid making mistakes."

"The product shall make the users want to use it."

"The product shall be used by people with no training, and possibly no understanding of English."

Fit Criterion

These examples may seem simplistic, but they do express the intention of the client. To completely specify what is meant by the requirement it is necessary to add a measurement of acceptance. We call this a fit criterion. The fit criterion for the above examples would be:

[An agreed percentage, say 90%] of a test panel of 11 year olds shall be able to successfully complete [list of tasks] within [specified time]

One month's use of the product shall result in a total error rate of less than [an agreed percentage, say 2%]

An anonymous survey shall show that [an agreed percentage, say 75%] of the users are regularly using the product after [an agreed time] familiarization period.

Considerations

Refer back to Section 3, the Users of the System, to ensure that you have considered the usability requirements from the perspective of all the different types of users.

It may be necessary to have special consulting sessions with your users and your client to determine whether there are any special usability considerations that must be built into the product.

You could also consider consulting a usability laboratory that has experience with testing the usability of products that have constraints (sections 1-7 of this template) similar to yours.

11b. Personalization and internationalization requirements

Content

This section describes the way in which the product can be altered or configured to take into account the user's personal preferences or choice of language.

The personalization requirements should cover such things as:

Languages, spelling preferences, language idioms

Currencies including the symbols and decimal conventions

Personal configuration options – there are a myriad of these

Motivation

To ensure that the product's users do not have to struggle with, or meekly accept, the cultural conventions of the builder.

Examples

"The product shall retain the buyer's buying preferences."

"The product shall allow the user to select a chosen language."

Considerations

Consider the locations of the potential customers and users of your product. Any out of country users will welcome the opportunity to convert to their home spelling and expressions.

By allowing users to customize the way in which they use the product, you are giving them the opportunity to participate more closely with your organization, as well as give them their own personal user experience.

You might also consider the configurability of the product. This allows different users to have different functional variations of the product.

11c. Ease of learning.

Content

A statement of how easy it should be to learn to use the product. This will range from zero time for products intended for placement in the public domain (for example a parking meter or a web site) to a considerable time for complex, highly technical products. (We know of one product where it was necessary for graduate engineers to spend 18 months in training before being qualified to use the product.)

Motivation

To quantify the amount of time that your client feels is allowable before a user can successfully use the product. This requirement will guide designers in how users will learn the product. For example, the designers may build elaborate interactive help facilities into the product, or the product may be packaged with a tutorial. Alternatively the product may have to be constructed so that all of its functionality is apparent upon first encountering it.

Examples

"The product shall be easy for an engineer to learn."

"A clerk shall be able to be productive within a short time."

"The product shall be able to be used by members of the public who will receive no training before using it."

"The product shall be used by engineers who will attend 5 weeks of training before using the product."

Fit Criterion

Fit criterion for the above example requirements are:

An engineer shall produce a [specified result] within [specified time] of beginning to use the product, without needing to use the manual.

After receiving [number of hours] training a clerk shall be able to produce [quantity of specified outputs] per [unit of time].

[Agreed percentage] of a test panel shall successfully complete [specified task] within [specified time limit].

The engineers shall achieve [agreed percentage] pass rate from the final examination of the training.

Considerations

Refer back to Section 3, the Users of the Product, to ensure that you have considered the ease of learning requirements from the perspective of all the different types of users.

11d. Understandability and Politeness requirements.

This section is concerned with discovering requirements related to concepts and metaphors that are familiar to the intended end-users.

Content

This specifies the requirement for the product to be understood by its users. While usability refers to ease of use, efficiency etc., understanding determines whether the users instinctively know what the product will do for them. In other words, the product fits into their view of the world. You can think of this as the product being polite to its users and not expecting them to know or learn things that have nothing to do with their business problem.

Motivation

To avoid forcing the user to learn terms and concepts that are part of the product's internal construction and are not relevant to the users' world. To make the product more comprehensible and thus more likely to be adopted by its intended users.

Examples

"The product shall use symbols and words that are naturally understandable by the user community".

"The product shall hide the details of its construction from the user."

Considerations

Refer back to Section 3, the Users of the Product, and consider the world from the point of view of each of the different types of users.

11e. Accessibility requirements.

Content

The requirements for how easy it should be for people with common disabilities to access the product. These disabilities might be to do with sight, physical disablement, hearing, cognitive, or others.

Motivation

In many countries it is required that some products are made available to the disabled. In any event, it is self-defeating to exclude this sizable community of potential customers.

Examples

"The product shall be usable by partially-sighted users."

"The product shall conform to the Americans with Disabilities Act."

Considerations

There are users with disabilities other than the commonly-described ones. Similarly, there are partial disabilities that are fairly common. A simple, and not very consequential example, is that approximately 20% of males are red-green color blind.

12 Performance Requirements

12a. Speed and latency requirements

Content

Specifies the amount of time available to complete specified tasks. These often refer to response times. They can also refer to the product's ability to fit into the intended environment.

Motivation

Some products, usually real-time products, must be able to perform some of their functionality within a given time slot. Failure to do so may mean catastrophic failure (for example a ground-sensing radar in an airplane fails to detect an upcoming mountain) or the product will not cope with the required volume of use (an automated ticket selling machine).

Examples

"Any interface between a user and the automated system shall have a maximum response time of 2 seconds"

"The response shall be fast enough to avoid interrupting the user's flow of thought"

"The product shall poll the sensor every 10 seconds"

"The product shall download the new status parameters within 5 minutes of a change"

Fit Criterion

Unit of measurement

Required range of values

Considerations

There is a wide variation in the importance of different types of speed requirements. If you are working on a missile guidance system then speed is extremely important. On the other hand, an inventory control report that is run once every 6 months has very little need for split second speed.

Customize this section of the template to give examples of the speed requirements that are important within your environment.

12b. Safety critical requirements

Content

Quantification of perceived risk of possible damage to people, property and environment. Note that different countries have different standards so the fit criteria must specify precisely which standards the product must meet.

Motivation

To understand and highlight the potential damage that could occur when using the product within the expected operational environment.

Examples

"The product shall not emit noxious gases that damage people's health."

"The heat exchanger shall be shielded from human contact."

Fit Criterion

Description of the perceived risk

Factors that could cause the damage

Unit for measuring the factors that could cause the damage

"The product shall be certified to comply with the Health Department's standard E110-98. This is to be certified by qualified testing engineers."

"No member of a test panel of [specified size] shall be able to touch the heat exchanger. The heat exchanger must also comply with safety standard [specify which one]."

Considerations

The sample requirements given above apply to some, but not all, products. It is not possible to give examples of every variation of safety critical requirement. To make the template work in your environment, you should customize it by adding examples that are specific to your products.

If you are building safety critical systems then the relevant safety critical standards are already well specified. You will likely have safety experts on your staff. These safety experts are the best source of the relevant safety critical requirements for your type of product. The safety experts will almost certainly have copious information that you can use.

Consult your legal department. They will be aware of the kinds of lawsuits that have resulted from product safety failure. This is probably the best starting place for generating relevant safety requirements.

12c. Precision or accuracy requirements

Content

Quantification of the desired accuracy of the results produced by the product.

Motivation

To set the client and user expectations for the precision of the product.

Examples

"All monetary amounts shall be accurate to 2 decimal places."

"Accuracy of road temperature readings shall be within + or - 2 degrees centigrade."

Fit Criterion

Unit of measure plus degree of precision

Considerations

If you have done any detailed work on definitions, then some precision requirements might be adequately defined by definitions in section 5.

12d. Reliability and Availability requirements

Content

This section quantifies the necessary reliability of the product. This is usually expressed as the allowable time between failures, or the total allowable failure rate.

It also quantifies the expected availability of the product.

Motivation

It is critical for some products not to fail too often. This section allows you to explore the possibility of failure and to specify realistic levels of service. It also gives you the opportunity to set client and user expectations about the amount of time that the product will be available for use.

Examples

"The product shall be available for use 24 hours per day, 365 days per year."

"The product shall be available for use between the hours of 8:00am and 5:30pm."

"The escalator shall run from 6am until the last flight arrives at 10pm."

"The product shall achieve 99% up time."

Considerations

Consider carefully whether the real requirement for your product is that it is available for use, or that it does not fail at any time.

Consider also the cost of reliability and availability, and whether it is justified for your product.

12e. Robustness or fault tolerance requirements

Content

Robustness specifies the ability of the product to continue to function under abnormal circumstances.

Motivation

To ensure that the product is able to provide some or all of its services after or during some abnormal happening in its environment.

Examples

"The product shall continue to operate in local mode whenever it loses its link to the central server."

"The product shall provide 10 minutes of emergency operation should it become disconnected from the electricity source."

Considerations

Abnormal happenings can almost be considered normal. Our products are so large and complex that there is a good chance that at any given time, one component will not be functioning correctly. Robustness requirements are intended to prevent total failure of your product.

You could also consider disaster recovery in this section. This refers to the ability of the product to re-establish acceptable performance after faults or abnormal happenings.

12f. Capacity requirements

Content

This section specifies the volumes that the product must be able to deal with and the numbers of data stored by the product.

Motivation

To ensure that the product is capable of processing the expected volumes.

Examples

"The product shall cater for 300 simultaneous users within the period from 9:00am to 11:am. Maximum loading at other periods will be 150."

"During a launch period the product shall cater for up to 20 people to be in the inner chamber."

Fit Criterion

In this case, the requirement description is quantified, and thus can be tested.

12g. Scalability or extensibility requirements

Content

This specifies the expected increases in size that the product must be able to handle. As a business grows (or is expected to grow) our software products must increase their capacities to cope with the new volumes.

Motivation

To ensure that the designers allow for future capacities.

Examples

"The product shall be capable of processing the existing 100,000 customers. This number is expected to grow to 500,000 within three years."

"The product shall be able to process 50,000 transactions an hour within two years of its launch."

12h. Longevity requirements

Content

This specifies the expected lifetime of the product.

Motivation

To ensure that the product is built based on an understanding of expected return on investment.

Examples

"The product shall be expected to operate within the maximum maintenance budget for a minimum of 5 years".

13 Operational Requirements

13a. Expected physical environment

Content

This section specifies the physical environment in which the product will operate.

Motivation

To highlight conditions that might need special requirements, preparations or training. These requirements ensure that the product is fit to be used in its intended environment.

Examples

"The product shall be used by a worker, standing up, outside in cold, rainy conditions."

"The product shall be used in noisy conditions with a lot of dust."

"The product shall be able to fit in a pocket or purse."

"The product shall be usable in dim light."

"The product shall be not add to the noise in the environment."

Considerations

The work environment: Is the product to operate in some unusual environment? Does this lead to special requirements? Also see section 11 - Usability.

13b. Expected technological environment

Content

Specification of the hardware and other devices that make up the operating environment for the new system.

Motivation

To identify all the components of the new system so that the acquisition, installation and testing can be effectively managed.

Considerations

Describe the hardware and other devices that make up the operating environment for the new system. This may not be known at the time of the requirements process, as these devices may be decided at design time.

It may be that the operating environment is complex, and becomes a subject of requirements study itself.

Special considerations should also be given if the product is to be embedded in a device.

If the expected operating environment is the same or similar to the current one, then this might be adequately covered in section 4b - Implementation Environment of the Current System.

13c. Partner applications

Content

Description of other applications with which the product must interface.

Motivation

Requirements for interfacing to other applications often remain undiscovered until implementation time. Avoid a high degree of rework by discovering these requirements early.

Examples

“We must be able to interface with any html browser.”

“The new version of the spreadsheet must be able to access data from the previous 2 versions”

“Our product must interface with the applications that run on the remote weather stations”

Fit Criterion

For each inter-application interface specify:

The data content

The physical material content

The medium that carries the interface

The frequency

The volume

13d. Productization requirements

Content

Any requirements that are necessary to make the product into a distributable or saleable item. It is also appropriate to describe here the operations to be performed to have a software product successfully installed.

Motivation

To ensure that if work has to be done to get the product out the door, then it becomes part of the requirements.

Examples

“The product shall be distributed as a ZIP file. ”

“The product shall be able to be installed by an untrained user without recourse to separately-printed instructions.”

“The product shall be of a size that it can fit onto one CD.”

Considerations

Some products have special needs to turn them into a saleable, or usable product. You might consider that the product has to be protected such that only paid-up customers can access it. This might be implemented as a dongle, a daily keyword, a check that no other copy of the product is running on the network at the same time.

Most commercial products have some needs in this area.

14 Maintainability and Support Requirements

14a. Maintenance requirements

Content

A quantification of the time necessary to make specified changes to the product.

Motivation

To make everyone aware of the maintenance needs of the product.

Examples

“New MIS reports must be available within one working week of the date the requirements are agreed”

“A new weather station must be able to be added to the system overnight”

Considerations

There may be special requirements for maintainability, such as this product must be able to be maintained by its end-users, or developers who are not the original developers. This has an effect on the way that the product is developed, and there may be additional requirements for documentation or training.

You might also consider writing testability requirements in this section.

14b. Special conditions that apply to the maintenance of the product

Content

Specification of the intended release cycle for the product and the form that the release will take.

Motivation

To make everyone aware of how often it is intended to produce new releases of the product.

Examples

“The maintenance releases will be offered to end-users once a year.”

“Every registered user will have access to our help site via the Internet.”

Fit Criterion

Description of type of maintenance + amount of effort budgeted

Considerations

Do you have any existing contractual commitments or maintenance agreements that might be affected by the new product?

14c. Supportability requirements

Content

This specifies the level of support that the product requires. This is often done using a help desk. If there are to be people who provide support for the product, is that considered part of the product and are there any requirements for that support. You might also build support into the product itself, in which case this is the place to write those requirements.

Motivation

To ensure that the support aspect of the product is adequately specified.

Considerations

Consider the anticipated level of support, and what forms it might take. For example, there may be a constraint that there is to be no printed manual. Or you might consider that the product is to be entirely self-supporting.

14d. Adaptability requirements

Content

Description of other platforms or environments to which the product must be ported.

Motivation

To quantify client and user expectations about the platforms on which the product will be able to run.

Examples

“The product is expected to run under Windows XP and Linux”

“The product might eventually be sold to the Japanese market”

“The product is designed to run in offices, but we intend to have a version which will run in restaurant kitchens”

Fit Criterion

Specification of system software on which the product must operate.

Specification of future environments in which the product is expected to operate.

Time allowed to make the transition.

Considerations

Ask questions from your marketing department to discover unstated assumptions that have been made about the portability of the product.

14e. Installation requirements

Content

Description of the effort needed to install the product.

Motivation

To quantify client and user expectations about the amount of time, money and resources they will need to allocate in order to install the product.

Examples

“The product shall be able to be installed in the specified environment within 2 working days.”

Considerations

Ask questions from your marketing department to discover unstated assumptions that have been made about the specified environment and the customers' expectations of how long installation will take and how much it will cost.

15 Security Requirements

15a. Access requirements

Content

Specification of who has authorized access to the product (both functionality and data), and under what circumstances that access is granted, and to what parts of the product access is allowed.

Motivation

To understand the expectations for confidentiality aspects of the system.

Examples

"Only direct managers can see the personnel records of their staff."

"Only holders of current security clearance can enter the building."

Fit Criterion

System function name or system data name

User role/s and/or names of people who have clearance

Considerations

Is there any data that is sensitive to the management? Is there any data that low-level users do not want management to have access to? Are there any processes that might cause damage or might be used for personal gain? Are there any people who should not have access to the system?

Avoid solving how you will design a solution to the security requirements. For instance, don't design a password system. Your aim here is to identify what the security requirement is. The design will come from this description.

Consider asking for help. Computer security is a highly-specialized field, and one where improperly-qualified people have no business being. If your product has need of more than average security, we advise that you make use of a security consultant. They are not cheap, but the results of inadequate security can be even more expensive.

15b. Integrity requirements

Content

Specification of the required integrity of databases and other files, and of the product itself.

Motivation

To understand the expectations for the integrity of the product's data. To specify what the product will do to insure its integrity in the case of an unwanted happening such as attack from the outside of unintentional misuse by an authorized user.

Examples

"The product shall prevent its data from incorrect data being introduced."

"The product shall protect itself from intentional abuse."

Considerations

Organizations rely more and more on their stored data. If this data should be come corrupt or incorrect, or indeed disappear, then it could be fatal. For example, it is true that almost half of small businesses go bankrupt after a fire destroys their computer systems. Integrity requirements are aimed at preventing complete loss, as well as corruption, of data and processes.

15c. Privacy requirements

Content

Specification of what the product has to do to insure the privacy of individuals that it stores information about. The product must also ensure that all laws about privacy of an individual's data are observed.

Motivation

To ensure that the product complies with the law, and to protect the individual privacy of your customers. Few people today look kindly on organizations that do not observe their privacy.

Examples

"The product shall make its user aware of its information practices before collection data from them."

"The product shall notify customers of changes to its information policy."

"The product shall reveal private information only in compliance with the organization's information policy."

"The product shall protect private information in accordance with relevant privacy laws / the organization's information policy."

Considerations

Privacy may well have legal implications, and you are advised to consult with your organization’s legal department about the requirements to be written in this section.

Consider what notices are required to be issued to your customers before collecting personal information. This can go so far as to warn them if you intend to put a cookie in their computer. Also, do you have to do anything to keep the customer aware that you hold personal information?

The customer must always be in a position to give or withhold consent when private data is collected or stored. Similarly, the customer should be able to view any private data, and where appropriate, ask for correction of the data.

Also consider the integrity and security of private data. A common example of this is the storage of credit card information.

15d. Audit requirements

Content

Specification of what the product has to do (usually retain records) to permit the required audit checks.

Motivation

To build a system that complies with the appropriate audit rules.

Considerations

This section may have legal implications. You are advised to seek the approval of your organization’s auditors for what you write here.

You should also consider whether the product should retain information on who has used it. The intention is to provide security in the form that a user may not later deny having used the product, or participated in some form of transaction using the product.

15e. Immunity requirements

Content

The requirements for what the product has to do to protect itself from infection by unauthorized or undesirable software programs, such as viruses, worms, Trojan horses and others.

Motivation

To build a product that is as secure as possible from malicious interference.

Considerations

Each day brings more malevolence from the unknown, outside world. People buying software, or any other kind of product, expect that it can protect itself from outside interference,

16 Cultural and Political Requirements

16a. Cultural requirements

Content

This section contains requirements that are specific to the sociological factors that affect the acceptability of the product. If you are developing a product for foreign markets then these requirements are particularly relevant.

Motivation

To bring out in the open requirements that are difficult to discover because they are outside the cultural experience of the developers.

Examples

"The product shall not be offensive to religious or ethnic groups."

"The product shall be able to distinguish between French, Italian and British road numbering systems."

"The product shall keep a record of public holidays for all countries in the European Union and for all states in the United States."

Considerations

Question whether the product is intended for a culture other than the one with which you are familiar. Ask whether people in other countries or in other types of organizations will use the product. Do these people have different habits, holidays, superstitions, cultural norms that do not apply to your own culture? Are there colours, icons or words that have different meanings in another cultural environment?

16b. Political requirements

Content

This section contains requirements that are specific to the political factors that affect the acceptability of the product.

Motivation

To try to understand requirements that sometimes appear irrational.

Examples

"The product shall be installed using component X."

"The product shall make all functionality available to the managing director."

"The product shall be developed using XYZ standards."

Considerations

Did you intend to develop the product on a Macintosh, when the office manager has laid down a edict that only Windows machines are permitted?

Is a director also on the board of a company that manufactures products similar to the one that you intend to build?

Whether you agree with these political requirements has little bearing on the outcome. The reality is that the system has to comply with political requirements even if you can find a better/more efficient/more economical solution. A few probing questions here may save some heartache later.

The political requirements might be purely concerned with the politics inside your organization. However there are situations when you need to consider the politics inside your customers' organizations or the national politics of the country.

17 Legal Requirements

17a. Compliance requirements

Content

A statement specifying the legal requirements for this system..

Motivation

To comply with the law so as to avoid later delays, law suits and legal fees.

Examples

"Personal information shall be implemented so as to comply with the data protection act."

Fit Criterion

Lawyers' opinion that the product does not break any laws.

Considerations

Consider consulting lawyers to help identify the legal requirements.

Are there any copyrights/intellectual property that must be protected? Alternatively, do any competitors have copyrights that you might be in danger of infringing?

Is it a requirement that developers have not seen competitors' code or even have worked for competitors?

Is there any pending legislation that might affect the development of this system?

Are there any aspects of criminal law you should consider?

Have you considered the tax laws that affect your product?

Are there any labour laws – eg: working hours – relevant to your product?

17b. Standards requirements

Content

A statement specifying applicable standards and referencing detailed standards descriptions. This does not refer to the law of the land, think of it as an internal "law" imposed by your company.

Motivation

To comply with standards so as to avoid later delays.

Example

"The product shall comply with MilSpec standards."

"The product shall comply with insurance industry standards".

"The product shall be developed according to SSADM standard development steps."

Fit Criterion

The appropriate standard-keeper certifies that the standard has been adhered to.

Considerations

It is not always apparent that there are applicable standards because their existence is often taken for granted. Consider the following:

Are there any industry bodies that have applicable standards?

Has the industry a code of practice, watchdog or ombudsman?

Are there any special development steps for this type of product?

18 Open Issues

Issues that have been raised and do not yet have a conclusion.

Content

A statement of factors that are uncertain and might make significant difference to the product.

Motivation

To bring uncertainty out in the open and provide objective input to risk analysis.

Examples

"Our investigation into whether or not the new version of the processor will be suitable for our application is not yet complete."

"The government are planning to change the rules about who is responsible for gritting the motor ways, but we do not know what the changes might be."

Considerations

Are there any issues that have come up from the requirements gathering that have not yet been resolved? Have you heard of any changes that might occur in the other organizations/systems on your context diagram? Are there any legislative changes that might affect your system? Any rumors about your hardware/software suppliers that might have an impact?

19 Off-the-Shelf Solutions

19a. Is there a ready-made product that could be bought?

Content

List of existing products that should be investigated as potential solutions. Reference any surveys that have been done on these products.

Motivation

To give consideration to whether or not a solution can be bought.

Considerations

Is it possible to buy something that already exists or is about to become available? It may not be possible at this stage to say with a lot of confidence, but any likely products should be listed here.

Also consider whether there are products that must not be used.

19b. Can ready-made components be used for this product?

Content

Description of the candidate components, either bought-in or built by your company, that could be used by this project. List libraries that could be a source of components.

Motivation

Reuse rather than reinvention.

19c. Is there something that we could copy?

Content

List of other similar products or parts of products that we can legally copy.

Motivation

Reuse rather than reinvention.

Examples

"Another electricity company has built a customer service system. Their hardware is different from ours but we could buy their specification and cut our analysis effort by approximately 60%."

Considerations

While a ready-made solution may not exist, there may well be something that, in its essence, is similar enough that you could copy, and possibly modify, to better effect that starting from scratch. This is dangerous because it relies on the base system being of good quality.

This question should always be answered. The act of answering will force you to look at other existing solutions to similar problems.

20 New Problems

20a. What problems could the new product cause in the current environment?

Content

A description of how the new product will affect the current implementation environment. This section should also cover things that the new product should *not* do.

Motivation

The intention is to discover early any potential conflicts that might otherwise not be realized until implementation time.

Examples

Any change to the scheduling system will affect the work of the engineers in the divisions and the truck drivers.

Considerations

Is it possible that the new system will damage some already existing system? Can people be displaced, or affected by the new system?

This requires a study of the current environment. A model highlighting the effects of the change is a good way to make this information widely understandable.

20b. Will the new development affect any of the installed system?

Content

Specification of the interfaces between new and existing systems.

Motivation

Very rarely is a new development intended to stand completely alone. Usually there is some existing system that the new one must coexist with. This question forces you to look carefully at the existing system and examine it for potential conflicts with the new development.

20c. Will any of our existing users be adversely affected by the new development?

Content

Details of any adverse reaction that might be suffered by existing users

Motivation

Sometimes existing users are using a product in such a way that they will suffer ill effects from the new system/feature. Identify any likely adverse user reaction, determine whether we care and what precautions we will take.

20d. What limitations exist in the anticipated implementation environment that may inhibit the new product?

Content

Statement of any potential problems with the new automated technology or new ways of structuring the organization.

Motivation

The intention is to make early discovery of any potential conflicts that might otherwise not be realized until implementation time.

Examples

The planned new server is not powerful enough to cope with our projected growth pattern.

The size/weight of the new product does not fit into the physical environment.

The power capabilities will not satisfy the new product's projected consumption.

Considerations

This requires a study of the intended implementation environment.

20e. Will the new product create other problems?

Content

Identification of situations that we might not be able to cope with.

Motivation

To guard against situations where the product might fail.

Considerations

Will we create a demand for our product that we are not able to service? Will the new system cause us to fall foul of laws that do not currently apply? Will the existing hardware cope?

There are potentially hundreds of unwanted effects. It pays to answer this question very carefully.

21 Tasks

21a. What steps have to be taken to deliver the system?

Content

Details of the life cycle and approach that will be used to deliver the product. A high level process diagram showing the tasks and interfaces between them is a good way to communicate this information.

Motivation

To specify the approach that will be taken to deliver the product so that everyone has the same expectations.

Considerations

Depending on the level of maturity of your process, the new product will be developed using your standard approach. However, there are some circumstances that are special to a particular product and will necessitate changes to your lifecycle. While these are not a product requirement, they are needed if the product is to be successfully developed.

If possible, attach an estimate of the time and resources need for each task based on the requirements that you have specified. Tag your estimates to the events/use cases/functions that you specified in sections 8 and 9.

Do not forget data conversion, user training and cutover. We have listed these because they are usually ignored when projects set implementation dates.

21b. Development phases

Content

Specification of each phase of development and the components in the operating environment.

Motivation

To identify the phases necessary to implement the operating environment for the new system so that the implementation can be managed.

Fit Criterion

Name of the phase

Required operational date

Operating environment components included

Functional requirements included

Non-functional requirements included

Considerations

Identify which hardware and other devices are necessary for each phase of the new system. This may not be known at the time of the requirements process, as these devices may be decided at design time.

22 Cutover

22a. What special requirements do we have to get the existing data, and procedures to work for the new system?

Content

A list of the Cutover activities. Timetable for implementation.

Motivation

To identify cutover tasks as input to the project planning process.

Considerations

Will you be using phased implementation to install the new system? If so, describe the requirements that will be implemented by each of the major phases.

What data conversion has to be done? Are there special programs to be written to transport data from an existing system to the new one? If so, the requirements for this program(s) are to be described here.

What manual backup is needed while the new system is installed?

When are each of the major components to be put in place, when are phases of the implementation to be released?

Is there a necessity to run the new product in parallel with existing product?

Will we need additional/different staff?

This section is the timetable for implementation of the new system.

22b. What data has to be modified/translated for the new system?

Content

List of data translation tasks.

Motivation

To discover missing tasks that will affect the size and boundaries of the project.

Fit Criterion

Description of the current technology that holds the data

Description of the new technology that will hold the data

Description of the data translation task/s

Foreseeable problems

Considerations

Every time you make an addition to your dictionary (see section 5) ask the question "What are all the places that this data is currently held and will the new system affect that implementation?".

23 Risks

All projects involve risk. By this we mean the risk that something will go wrong. Risk is not necessarily a bad thing, as no progress is made without taking some risk. However, there is a difference between unmanaged risk – say shooting dice at a craps table – and managed risk where the probabilities are well understood, and contingencies made. Risk is only a bad thing if the risks are ignored and they become problems. Risk management is assessing which risks are most likely to apply to the project, deciding a course of action if they become problems, and monitoring projects to give early warnings of risks becoming problems.

This section of your specification should contain a list of the most likely and the most serious risks for your project. Against each risk include the probability of that risk becoming a problem. Capers Jones' book *Assessment and Control of Software Risks*. Prentice-Hall, Englewood Cliffs, NJ. 1994 gives comprehensive lists of risks and their probabilities, you can use these as a starting point. For example, Jones cites the following risks as being the most serious:

- Inaccurate metrics
- Inadequate measurement
- Excessive schedule pressure
- Management malpractice
- Inaccurate cost estimating
- Silver bullet syndrome
- Creeping user requirements
- Low quality
- Low productivity
- Cancelled projects

Use your knowledge of the requirements as input to discovering risks that are most relevant to your project.

It is also useful input to project management if you include the impact on the schedule, or the cost, if the risk does become a problem.

24 Costs

For details on how to estimate requirements effort and costs, refer to our book: **Requirements-led Project Management: Discovering David's Slingshot**, Addison Wesley, 2005

The other cost of requirements is the amount of money or effort that you have to spend building them into a product. Once the requirements specification is complete, you can use one of the estimating methods to assess the cost, and express this in a monetary amount or time to build.

There is no best method to use when estimating. However your estimates should be based on some tangible, countable, artifact. If you are using this template then, as a result of doing the work of requirements specification, you are producing many measurable deliverables. For example:

Number of input and output flows on the work context

Number of business events

Number of product use cases

Number of functional requirements

Number of non-functional requirements

Number of requirements constraints

Number of function points

The more detailed work you do on your requirements the more accurate will be your deliverables. Your cost estimate is the amount of resources you estimate each type of deliverable will take to produce within your environment. You can do some very early cost estimates based on the work context. At that stage, your knowledge of the work will be general and you should reflect this by making the cost estimate a range rather than one figure.

As you get more knowledge about the requirements we suggest you try using function point counting – not because it is an inherently superior method - but because it is so commonly accepted. So much is known about it, that it is possible to make easy comparisons with other products, and other installations' productivity.

It is important that your client knows at this stage what the product is likely to cost. You usually express this as a total cost to complete the product, but you may also find it advantageous to be able to point out the cost of individual requirements.

Whatever you do, do not leave the costs in the lap of hysterical optimism. Make sure that this section includes meaningful numbers based on tangible deliverables.

25 User Documentation and Training

25a. The plan for building the user documentation.

Content

List of the user documentation that will be supplied as part of the system and to describe the training that will be available.

Motivation

To set expectations for the documentation and training and to identify who will be responsible for creating it.

Considerations

Which documents will you need to deliver to whom – bear in mind that this is dependent on your organizational procedures and roles.

For each document consider:

- The purpose of the document
- The people who will use the document
- Maintenance of the document

What level of documentation is expected? Will the users be involved in the production of the documentation? Who will be responsible for keeping the documentation up to date? What form will the documentation take? What training will be necessary? Who will design the training? Who will provide the training?

26 Waiting Room

Requirements that will not be part of the agreed product. These requirements might be included in future versions of the product.

Content

Any type of requirement.

Motivation

To allow requirements to be gathered, even though they cannot be part of the current development. To ensure that good ideas are not lost.

Considerations

The requirements gathering process often throws up requirements that are beyond the sophistication of, or time allowed for, the current release of the product. This section is a hold-all for requirements in waiting. The intention is to avoid stifling your users and clients by having a repository of future requirements. You are also managing expectations by making it clear that you take these requirements seriously but they will not be part of the agreed product.

Many people use the waiting room as a way of planning future versions of the product. Each requirement in the waiting room is tagged with its intended version number. As a requirement progresses closer to implementation then you spend more time on it and add details like the cost and benefit attached to the requirement.

27 Ideas for Solutions

When you are gathering requirements you are focusing on finding out what the real requirements are, you are trying to avoid coming up with solutions. However when creative people start to think about a problem they always have ideas. This section of the template is a place to put those ideas so that you do not forget them and so that you can separate them from the real business requirements.

Content

Any idea for a solution that you think is worth keeping for future consideration. This can take the form of rough notes, sketches, pointers to other documents, pointers to people, pointers to existing products….the aim is to capture, with the least amount of effort, an idea that you can come back to later.

Motivation

To make sure that good ideas are not lost and to help you separate requirements and solutions.

Considerations

Whilst you are gathering requirements you will have solution ideas, this is a way of capturing them. Bear in mind that this section will not necessarily be published as part of the specification that you publish.

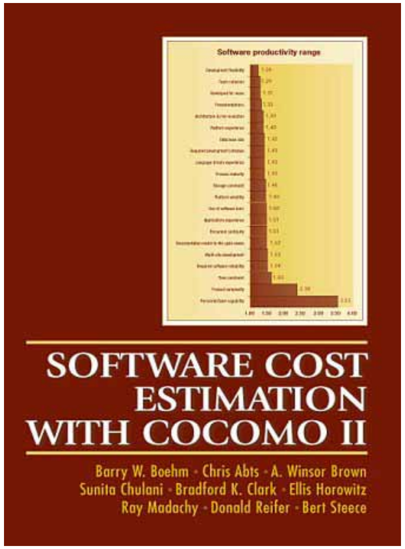
Aufwands- und Kostenschätzung
COCOMO II und CostXPert

Typische Entscheidungen im Projektmanagement

- Soll das Projekt gestartet/weitergeführt werden?
- Mit welchem Aufwand ist zu rechnen?
- Mit welcher Ressourcenbelastung ist im Life-Cycle zu rechnen?
- Wie wirkt sich die Unerfahrenheit mit dem Anwendungsbereich aus?
- …

COCOMO (COnstructive Cost Model)

- Empirisches Modell das auf Projekterfahrungen beruhrt
- 1981 erstmals von Barry W. Boehm veröffentlicht
- Die neue Version COCOMO II berücksichtigt aktuelle Entwicklungsparadigmen


Software productivity range
SOFTWARE COST ESTIMATION WITH COCOMO II
Barry W. Boehm · Chris Abts · A. Winsor Brown
Sunita Chulani · Bradford K. Clark · Ellis Horowitz
Ray Madachy · Donald Reifer · Bert Steece

Prentice-Hall, July 2000

http://sunset.usc.edu/research/COCOMOII/ 3

COCOMO II

- Grösse als grundlegender Eingangsparameter $PM_{nom} = A \times (Size)^B$
  - Gemessen in LOC, Function Points

- Exponent: Economies / diseconomies of scale $B = 0.91 + 0.01 \times \sum W_i$
  - *Precedentedness*

    Gibt es schon Erfahrung einer Organisation mit einem Projekttyp?
  - *Development Flexibility*

    Wie flexibel ist der Entwicklungsprozess?
  - *Architecture/Risk Resolution*

    Wie umfangreich wurde die Risikoanalyse durchgeführt?
  - *Team Cohesion*

    Wie gut kennt sich das Team? Wie hoch ist die Kooperationsbereitschaft?
  - *Process Maturity*

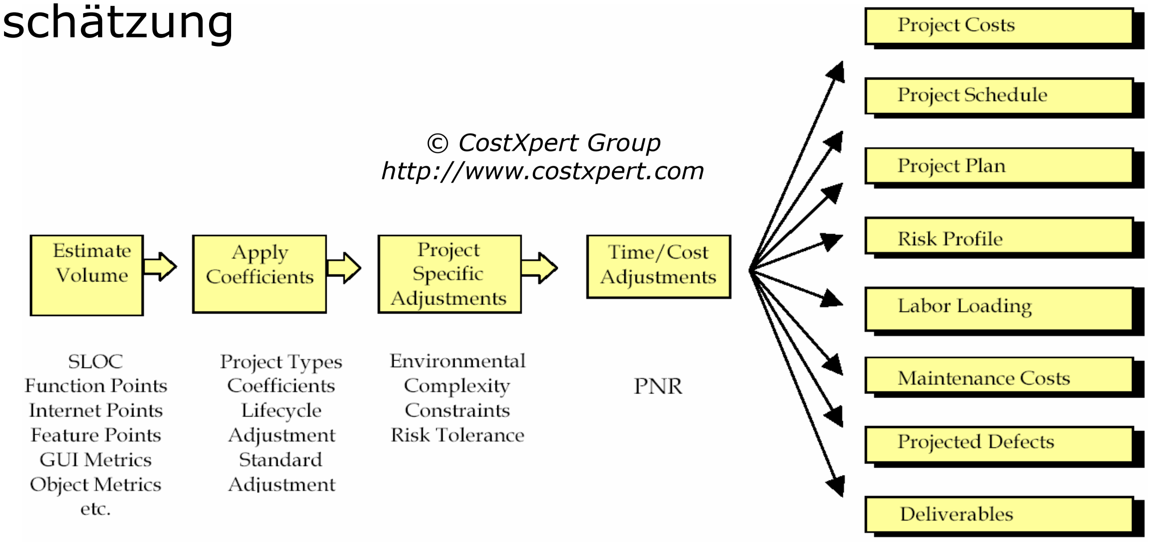
    Wie hoch ist der Reifegrad der Organisation?

COCOMO II Kostenfaktoren

- *Product attributes*
  - Required software reliability (RELY)
  - Database size (DATA)
  - Required Reusability (RUSE)
  - …
- *Platform attributes*
  - Execution time constraint (TIME)
  - Main storage constraint (STOR)
  - Platform volatility (PVOL)
- *Personnel attributes*
  - Analyst capabilities (ACAP)
  - Programmer capabilities (PCAP)
  - Platform experience (PEXP)
  - …

Werkzeugbeispiel: CostXpert

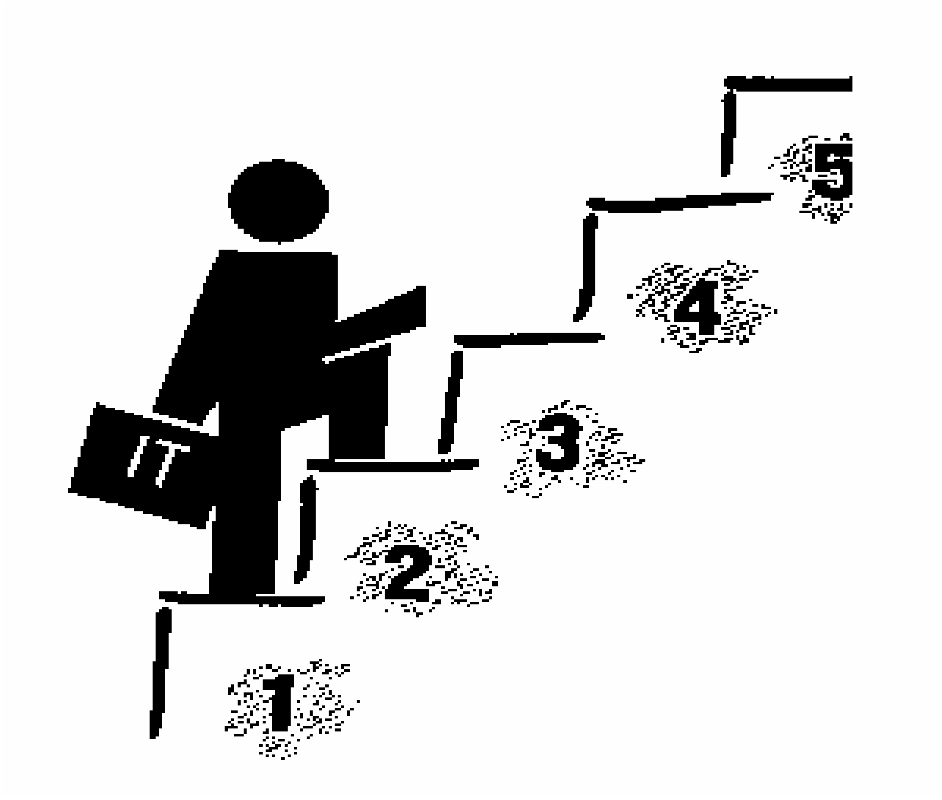
- Unterstützt COCOMO II als Methode
- Basiert auf den Daten von über 20,000 Software Projekten
- Verschiedene Methoden zur Größenschätzung



© CostXpert Group
http://www.costxpert.com

8

Software-Prozeßverbesserung am Beispiel ISO 15504 (SPICE)



Was ist ein Prozess?

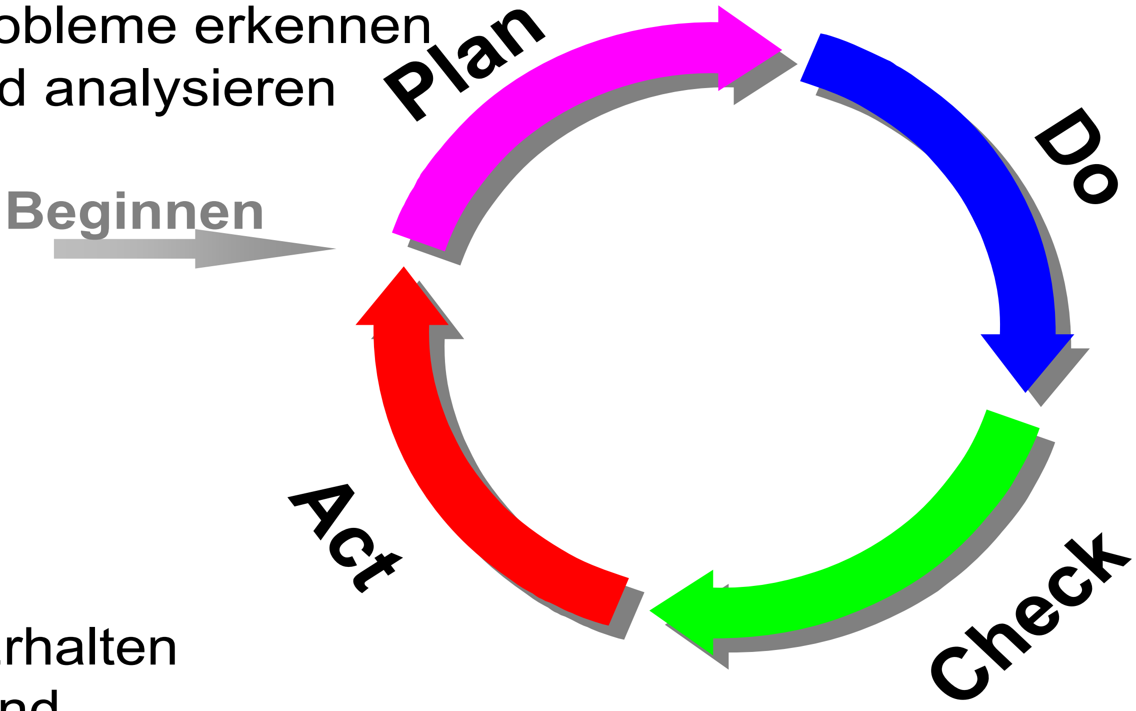
A set of interrelated activities, which transform inputs into outputs.

[ISO 15504-9]

The quality of a product or service is governed by the quality of the processes used to develop and evolve it.

Watts Humphrey

PDCA Cycle



Probleme erkennen
und analysieren

Beginnen

Plan

Do

Prozesse
einrichten,
messen,
verbessern

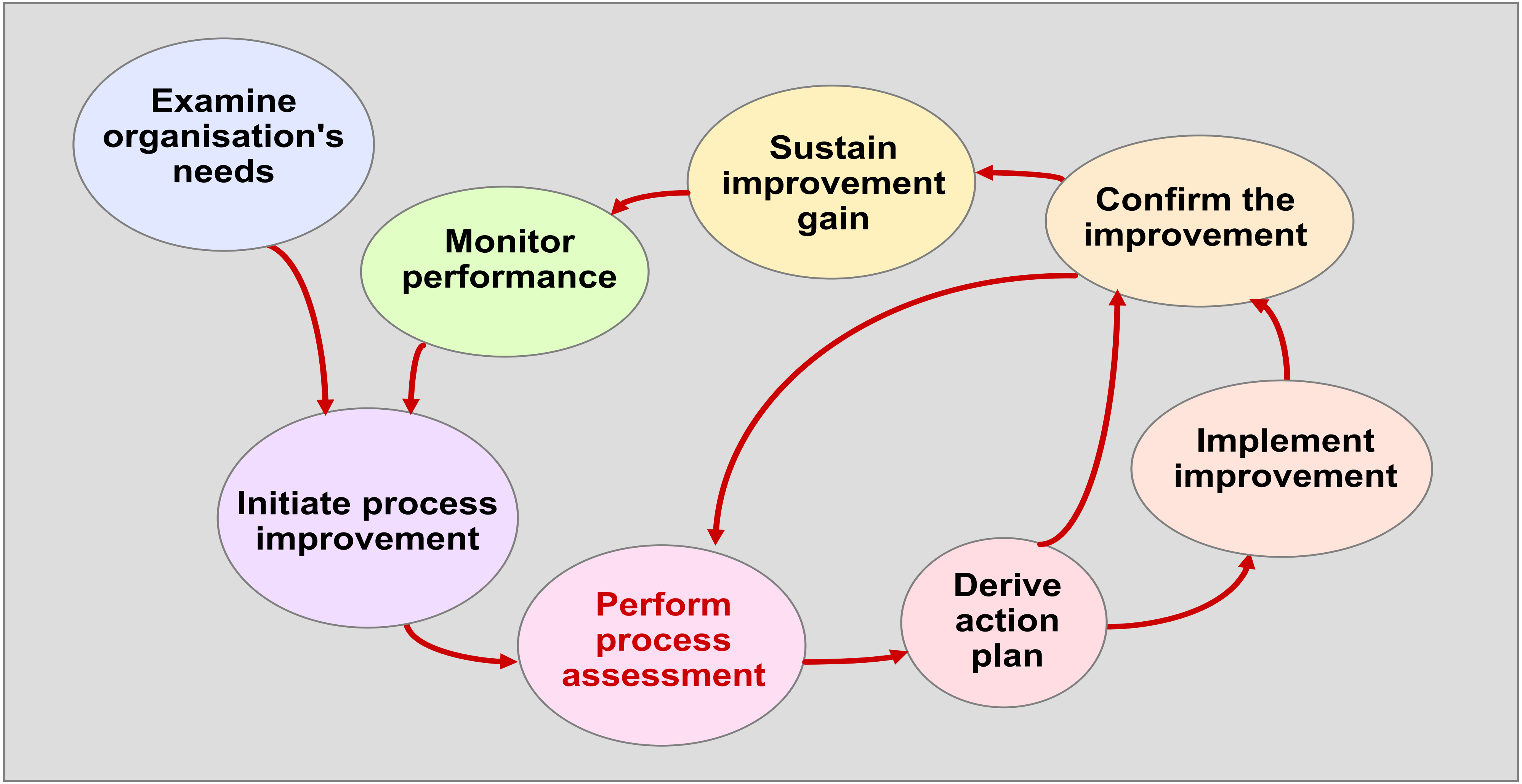
Check

Sind die
Ziele
erreicht?

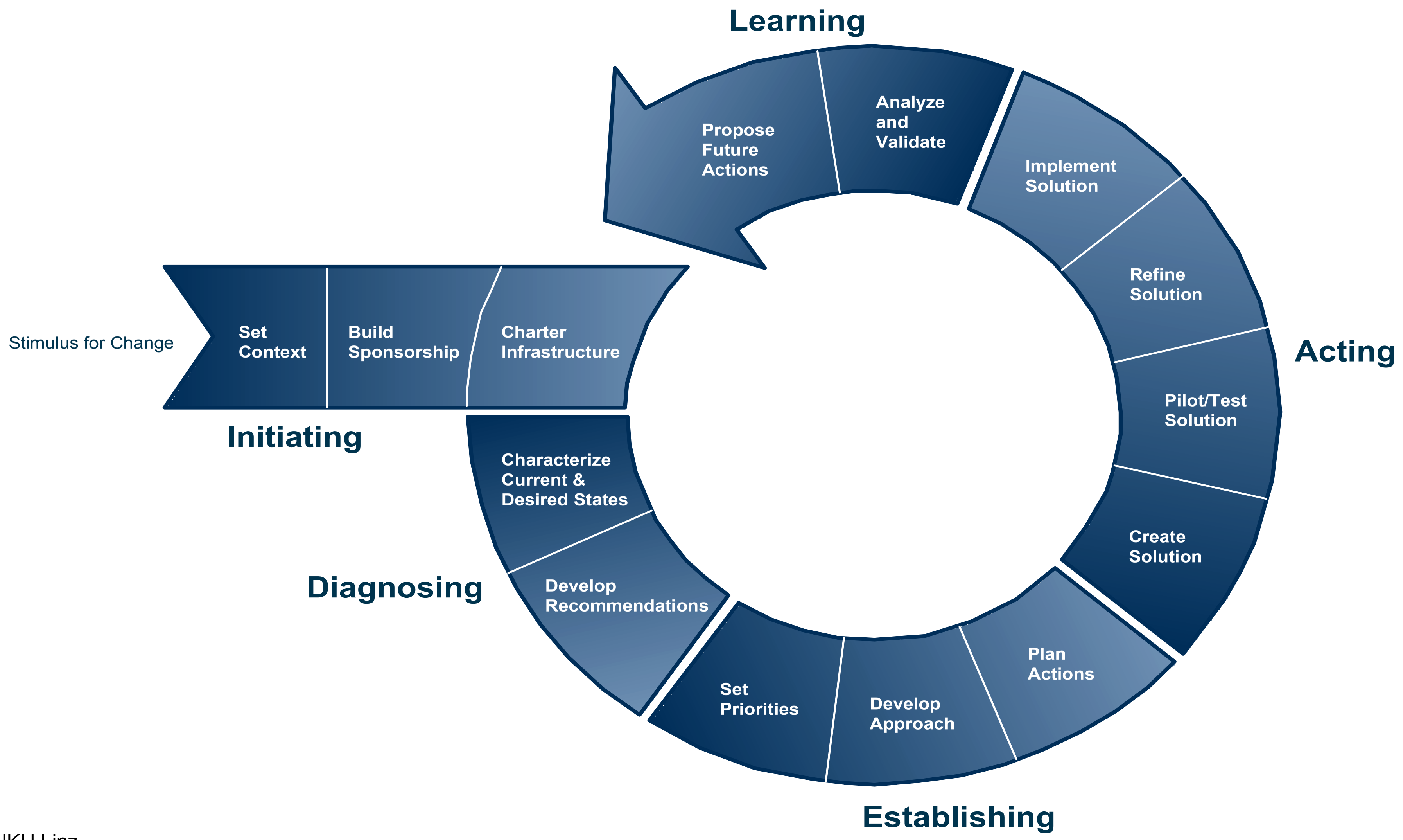
Act

Erhalten
und
ausweiten

Process Improvement Steps



SEI IDEAL Approach [McFeeley-96]



http://www.sei.cmu.edu/ideal/

The First Step ...

Determine your current position:

Before you start,

you need to know *where* you are.

If you don’t know where you are,

a map won’t help.

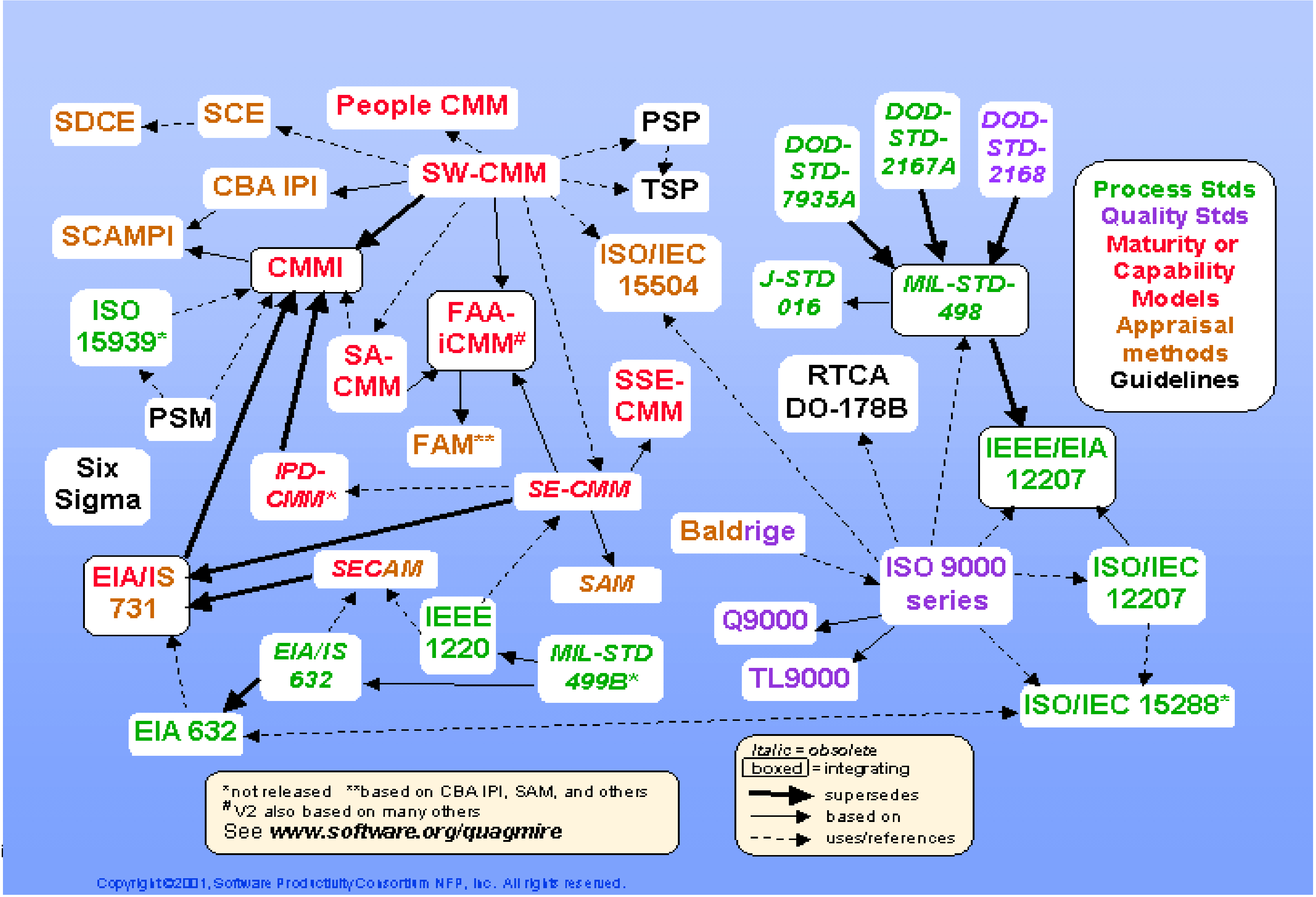
… The Next Step

Determine your path:

Once you know *where* you are, you also need to know *where to* go to and *how to* get there.

If you don’t know where you are going, any road will do.

Software Process Quagmire



SDCE

SCE

People CMM

PSP

TSP

SW-CMM

CBA IPI

SCAMPI

CMMI

ISO 15939*

PSM

Six Sigma

SA-CMM

FAA-iCMM#

FAM**

IPD-CMM*

ISO/IEC 15504

SSE-CMM

SE-CMM

EIA/IS 731

SECAM

SAM

IEEE 1220

EIA/IS 632

MIL-STD 499B*

EIA 632

DOD-STD-7935A

DOD-STD-2167A

DOD-STD-2168

J-STD 016

MIL-STD-498

RTCA DO-178B

IEEE/EIA 12207

Baldrige

ISO 9000 series

ISO/IEC 12207

Q9000

TL9000

ISO/IEC 15288*

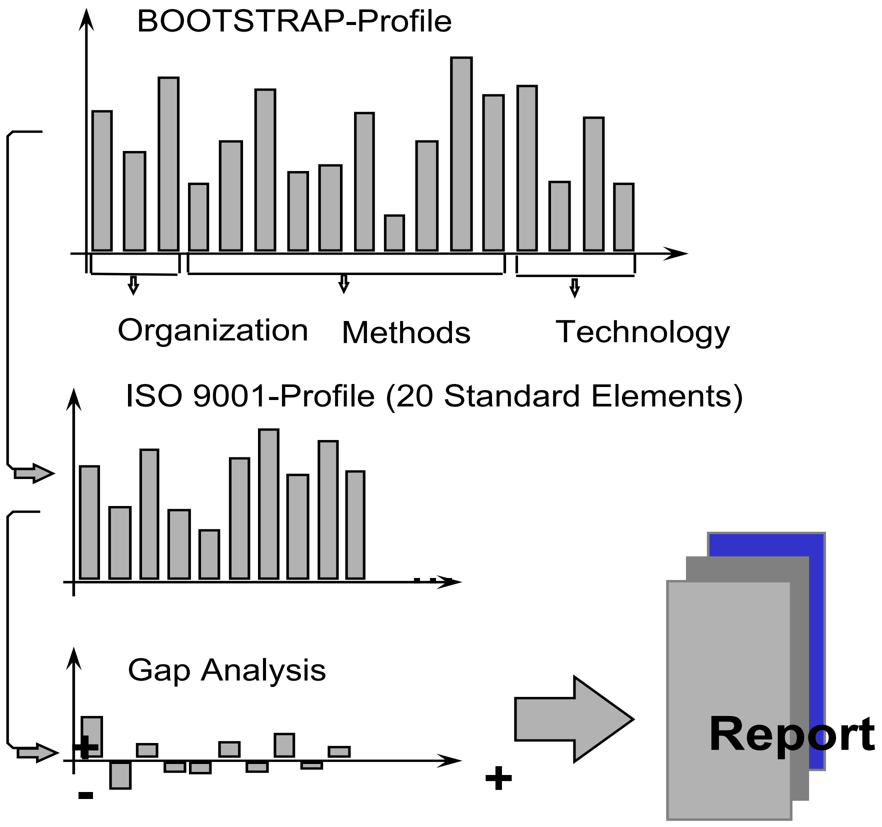
Process Stds
Quality Stds
Maturity or Capability Models
Appraisal methods
Guidelines

*not released **based on CBA IPI, SAM, and others
#V2 also based on many others
See *www.software.org/quagmire*

Italic = *obsolete*
boxed = integrating
supersedes
based on
uses/references

Software Prozesse messen?!

- Standards:
  - ISO 9000
  - ISO 15504 (SPICE) [Rout-95]
- Industrieverfahren:
  - CMM, CMMI
  - BOOTSTRAP [Kuvaja-94]
  - TRILLIUM
  - SynQuest [Steinmann-96]

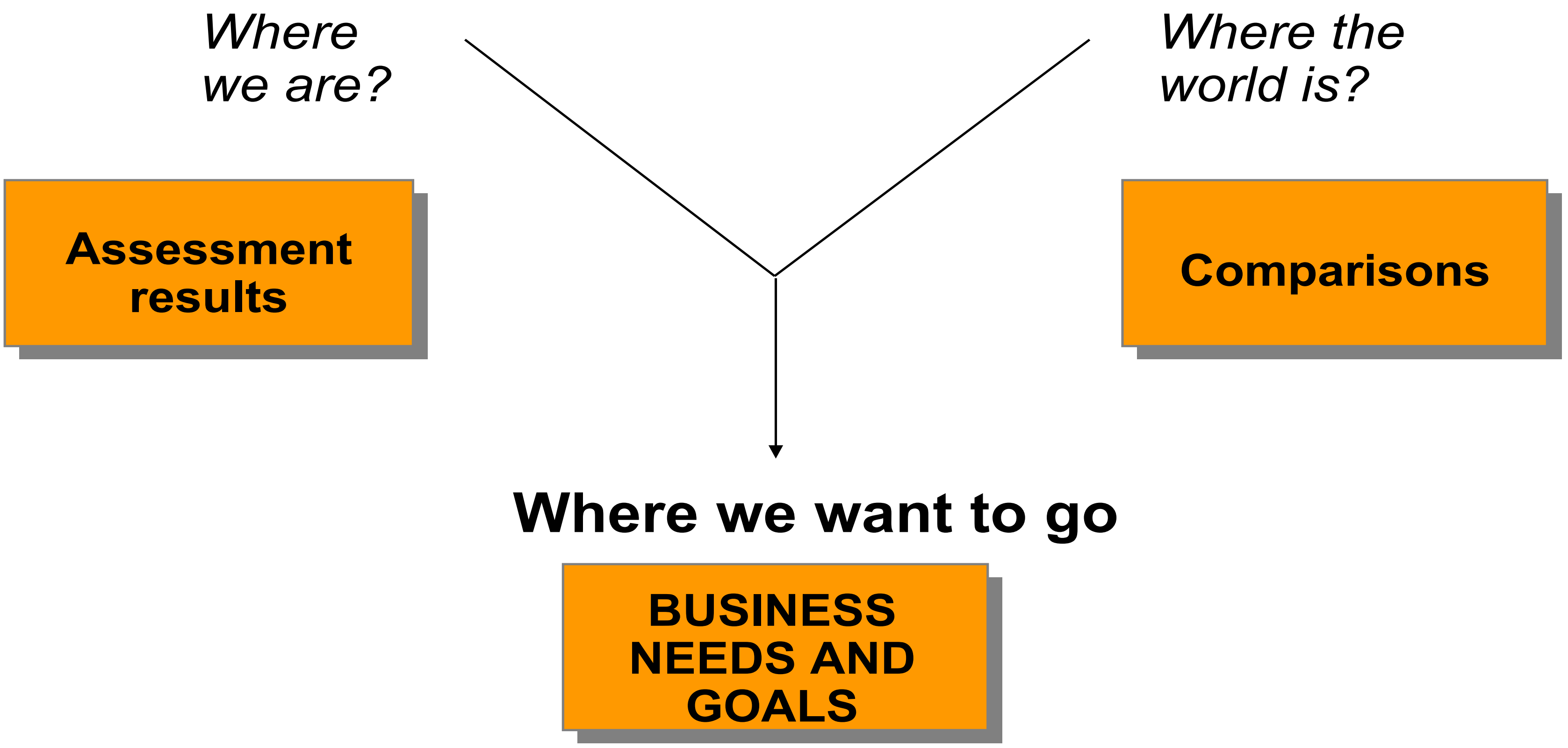


Prozeßbewertung und -zertifierung

- Zentrale Stellung des "SW-Prozesses"
- (gemessene) Qualität eines Prozesses ist wesentlicher Faktor
- “Prozeß-Gütesiegel” für außenstehende Vertragspartner

- ISO-9000: ja oder nein
- TICKIT: yes or no
- CMM: 1.0 bis 5.0
  - Pionier
  - amerikanisch
  - große Projekte
  - Militärstandard
- Bootstrap: <3.3 , **5.0**, 1.2, 2.2, ...>
  - Verfeinerer
  - europäisch
  - alle Projektgrößen
  - ISO 9000
- SPICE: <4.1 , **6.0**, 2.1, 2.1, ...>

ASSESSMENT OBJECTIVES - Process Improvement



Where we are?

Assessment results

Where the world is?

Comparisons

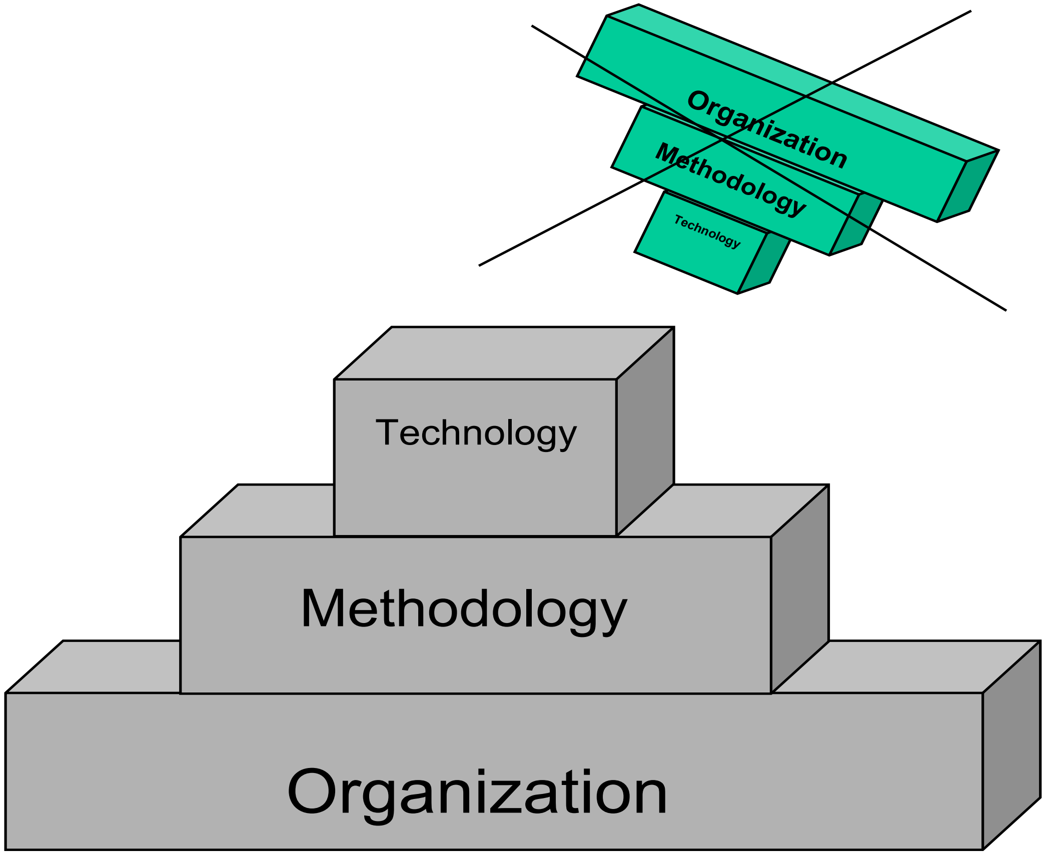
Where we want to go

BUSINESS NEEDS AND GOALS

Prozessverbesserung (Beispiele)

- kurzfristig
  - Einführung von Projektplanung und -überwachung
  - ....
- mittelfristig
  - Berichtswesen neu organisieren
  - z.B. objektorientierte Verfahren einführen
  - Kostenschätzungstools einführen
- langfristig
  - Neugliederung von Abteilungen (z.B. Kundentyp-orientiert)
  - Qualitätsstandards stetig anheben

Nicht mit der Technik beginnen!



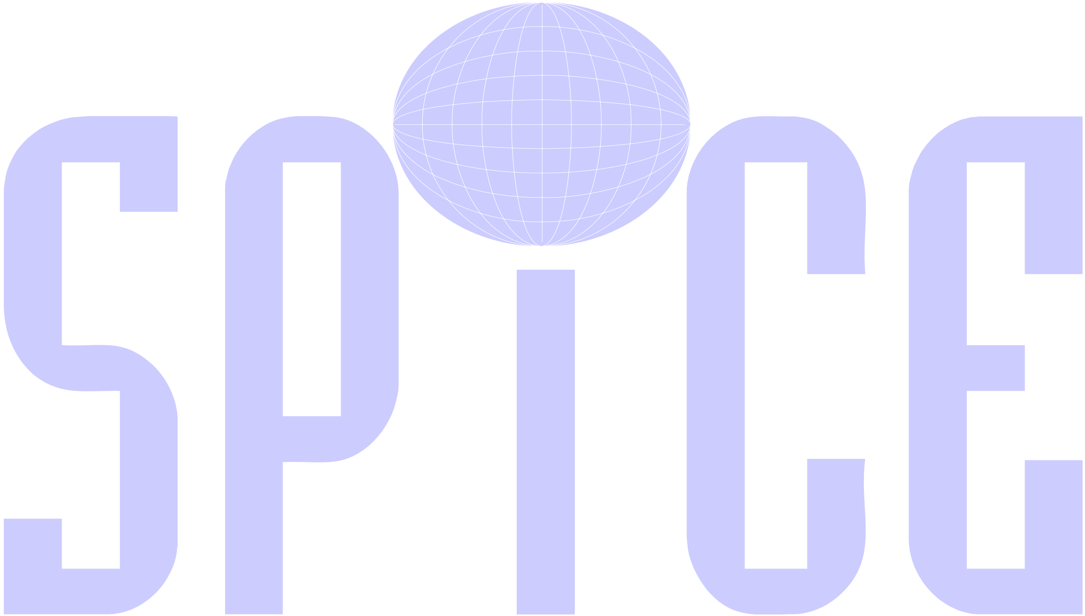
Process Assessment - a History

- 1987 Humphrey / Sweet Report from SEI
- 1989 SEI issues its first questionnaire; Watts Humphrey’s book published
- 1991 CMM V1.0 released
- 1991 ISO requests a study on process assessment
- 1993 ISO accepts new work item on PA
- 1993 CMM V1.1 released
- 1994 SPICE Standard released
- 1995 SE-CMM 1.1
- 1999 CMMI 0.2
- 2000 CMMI 1.0
- 2002 CMMI 1.1

Many organizations now have their own schemes

SPICE: ISO 15504

[Dorling-93]
[Rout-95]
[Simon-96]



Software Process Improvement and
Capability dEtermination

Was ist ISO/IEC TR 15504 (SPiCE)?

- Internationale Norm zur Bewertung der SW-Entwicklungsprozesse
  - Harmonisierung der bestehenden Ansätze (BOOTSTRAP, CMM, TRILLIUM)
  - Abgestimmt mit den gängigen Entwicklungsstandards (z.B. ISO 12207)
- Zielsetzungen
  - Grundlage für die kontinuierliche Prozessverbesserung
  - Bestimmung der Leistungsfähigkeit
- Der Rahmen
  - Umfassend
    - Nicht nur die eigentliche Entwicklung, sondern auch Beschaffung, Lieferung, Entwicklungsunterstützung, Betrieb und Wartung
  - Modular
    - Erlaubt die Fokussierung auf einige spezifischen Prozesse
    - Getrennte Bewertung jeder einzelner Prozess

dEtermination

SPiCE - Software Process Improvement & Capability ~~Evaluation~~

SPICE: Die Prozessarchitektur

Customer-Supplier

CUS.1 Acquisition
- CUS.1.1 Acquisition Preparation
- CUS.1.2 Supplier Selection
- CUS.1.3 Supplier Monitoring
- CUS.1.4 Customer Acceptance

CUS.2 Supply

CUS.3 Requirements Elicitation

CUS.4 Operation
- CUS.4.1 Operational Use
- CUS.4.2 Customer Support

Engineering

ENG.1 Development
- ENG.1.1 System Requirements Analysis & Design
- ENG.1.2 Software Requirements Analysis
- ENG.1.3 Software Design
- ENG.1.4 Software Construction
- ENG.1.5 Software Integration
- ENG.1.6 Software Testing
- ENG.1.7 System Integration & Testing

ENG.2 System & Software Maintenance

Support

SUP.1 Documentation
SUP.2 Configuration Management
SUP.3 Quality Assurance
SUP.4 Verification
SUP.5 Validation
SUP.6 Joint Reviews
SUP.7 Audit
SUP.8 Problem Resolution

Management

MAN.1 Management
MAN.2 Project Manangement
MAN.3 Quality Management
MAN.4 Risk Management

Organisation

ORG.1 Organisational Alignment

ORG.2 Improvement
- ORG.2.1 Process Establishment
- ORG.2.2 Process Assessment
- ORG.2.3 Process Improvement

ORG.3 Human Resource Management
ORG.4 Infrastructure
ORG.5 Measurement
ORG.6 Reuse

PROCESS CATEGORIES DESCRIPTIONS

- **Customer-Supplier**
  - Impact customer & supplier relationship/partnership, support product and service transition to customer including its correct operation & use
- **Engineering**
  - Specification, implementation and/or maintenance of system and software product and/or service (including documentation)
- **Support**
  - Provide support activities to other processes
- **Management**
  - Management of project or process in a software development life cycle
- **Organization**
  - Establishment of organizational business goals, resources (including process) and infrastructure

WHAT DOES EACH PROCESS INCLUDE?

Process Description & Identifier

Process Purpose Statement

Measurable

Supporting Notes

Perform configuration management (SUP.2)

The purpose of the *Perform configuration management* process is to establish and maintain the integrity of all the work products of a process or project.

As a result of successful implementation of the process

- all relevant items generated by the process/project will be identified, defined and baselined;

- modifications and releases of the items will be controlled;

- the status of the items and modification requests will be recorded and reported;

- the completeness and consistency of the items will be ensured;

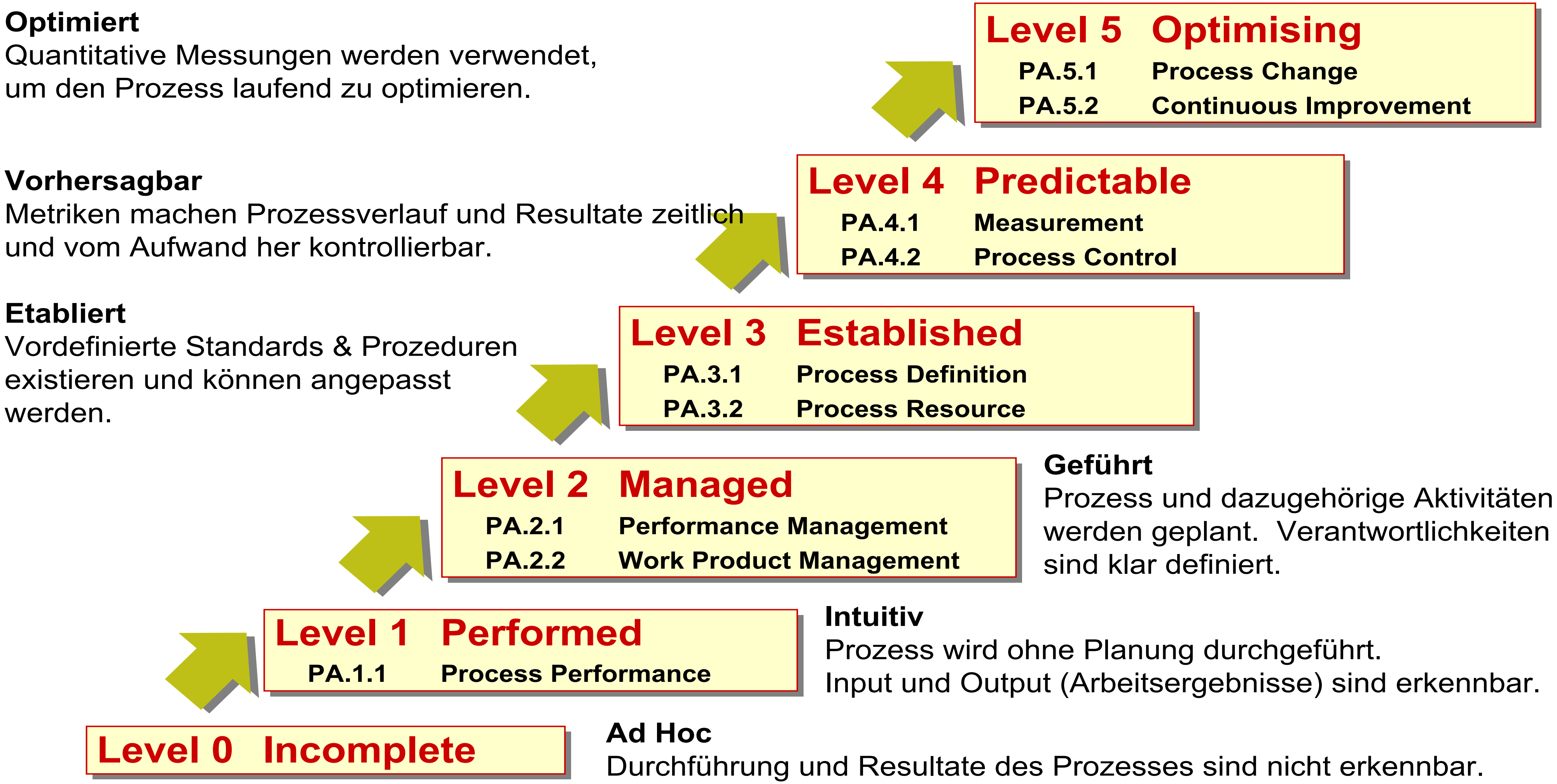
- storage, handling and delivery of the items will be controlled.

Notes:

21. This process is identical to the Configuration Management Process, one of the Support Life Cycle Processes in ISO 12207.

22. The process supports performance of the process attribute 2.2 in those instances where it is invoked.

Prozessattribute und -stufen



Level 5 Optimising

PA.5.1 Process Change

PA.5.2 Continuous Improvement

Optimiert
Quantitative Messungen werden verwendet, um den Prozess laufend zu optimieren.

Level 4 Predictable

PA.4.1 Measurement

PA.4.2 Process Control

Vorhersagbar
Metriken machen Prozessverlauf und Resultate zeitlich und vom Aufwand her kontrollierbar.

Level 3 Established

PA.3.1 Process Definition

PA.3.2 Process Resource

Etabliert
Vordefinierte Standards & Prozeduren existieren und können angepasst werden.

Level 2 Managed

PA.2.1 Performance Management

PA.2.2 Work Product Management

Geführt
Prozess und dazugehörige Aktivitäten werden geplant. Verantwortlichkeiten sind klar definiert.

Level 1 Performed

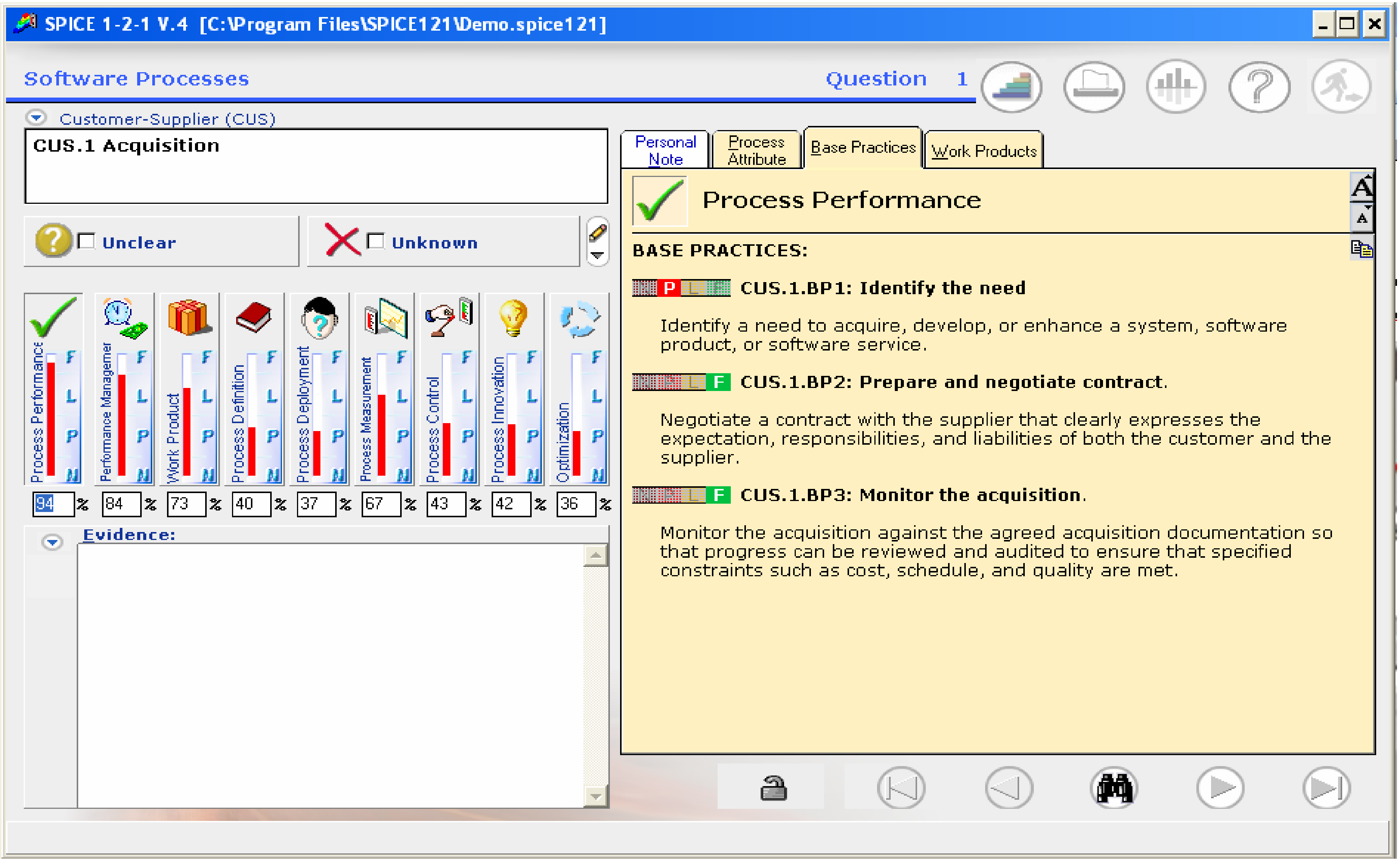
PA.1.1 Process Performance

Intuitiv
Prozess wird ohne Planung durchgeführt. Input und Output (Arbeitsergebnisse) sind erkennbar.

Level 0 Incomplete

Ad Hoc
Durchführung und Resultate des Prozesses sind nicht erkennbar.

Prozessbewertung -- www.spice121.com



CUS.1 Acquisition

□ Unclear

□ Unknown

Process Performance	Performance Managemer	Work Product	Process Definition	Process Deployment	Process Measurement	Process Control	Process Innovation	Optimization
94 %	84 %	73 %	40 %	37 %	67 %	43 %	42 %	36 %

Evidence:

Process Performance

BASE PRACTICES:

P L F **CUS.1.BP1: Identify the need**

Identify a need to acquire, develop, or enhance a system, software product, or software service.

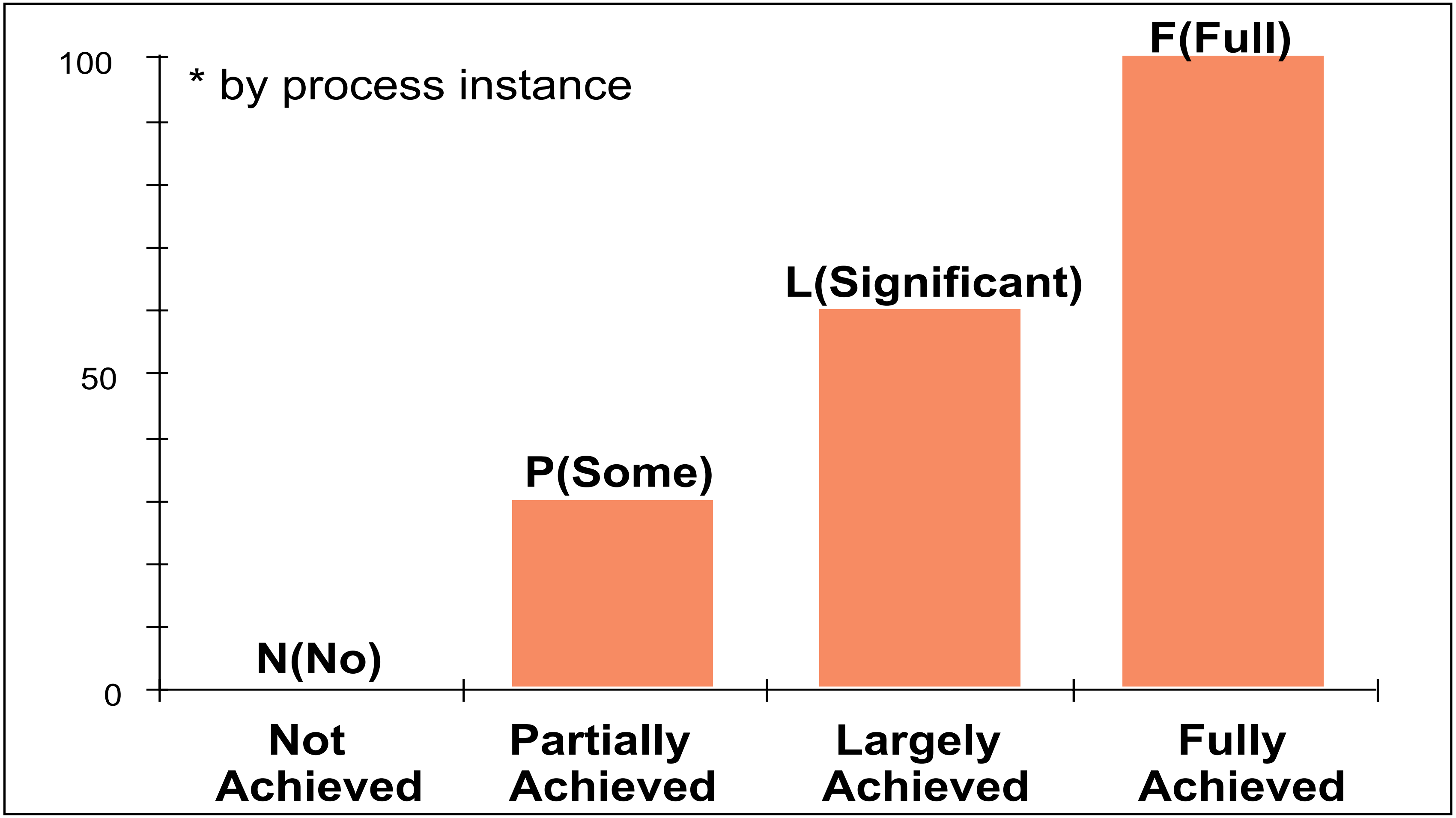
L F **CUS.1.BP2: Prepare and negotiate contract.**

Negotiate a contract with the supplier that clearly expresses the expectation, responsibilities, and liabilities of both the customer and the supplier.

L F **CUS.1.BP3: Monitor the acquisition.**

Monitor the acquisition against the agreed acquisition documentation so that progress can be reviewed and audited to ensure that specified constraints such as cost, schedule, and quality are met.

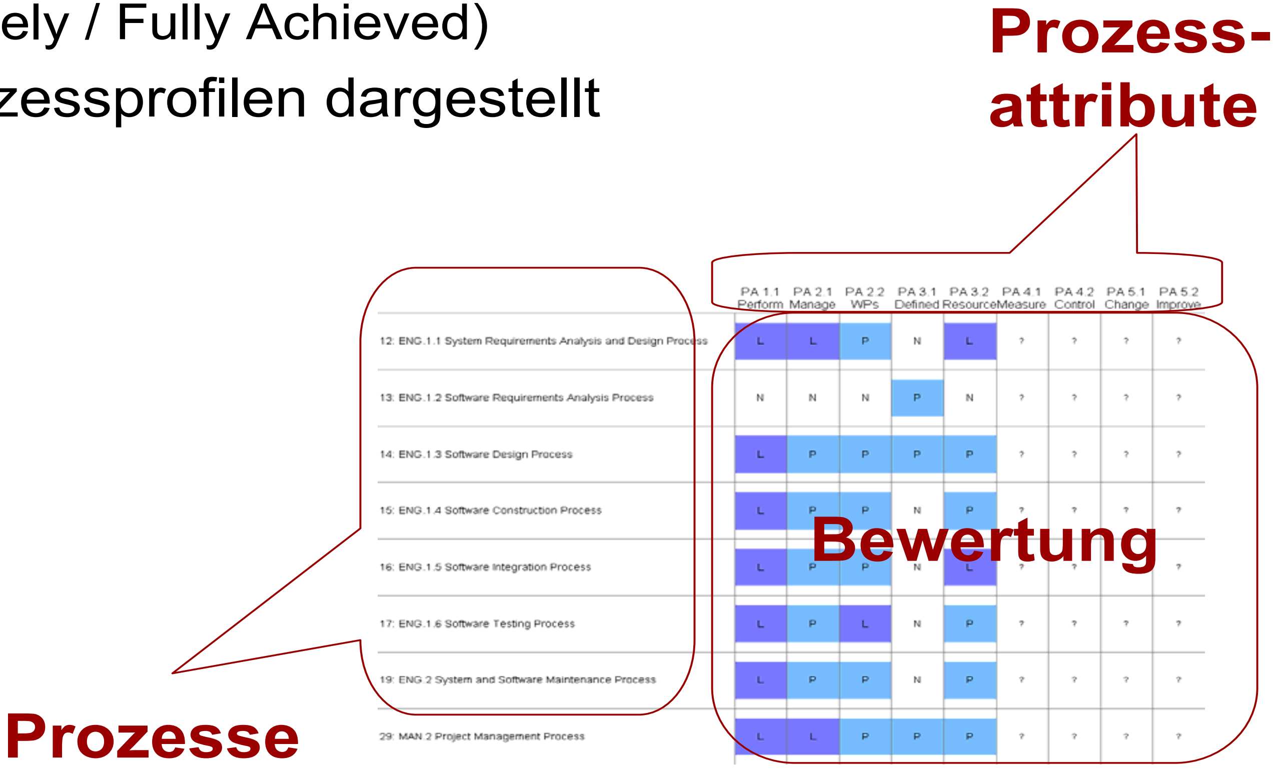
PROCESS ATTRIBUTE RATING



Prozessprofile

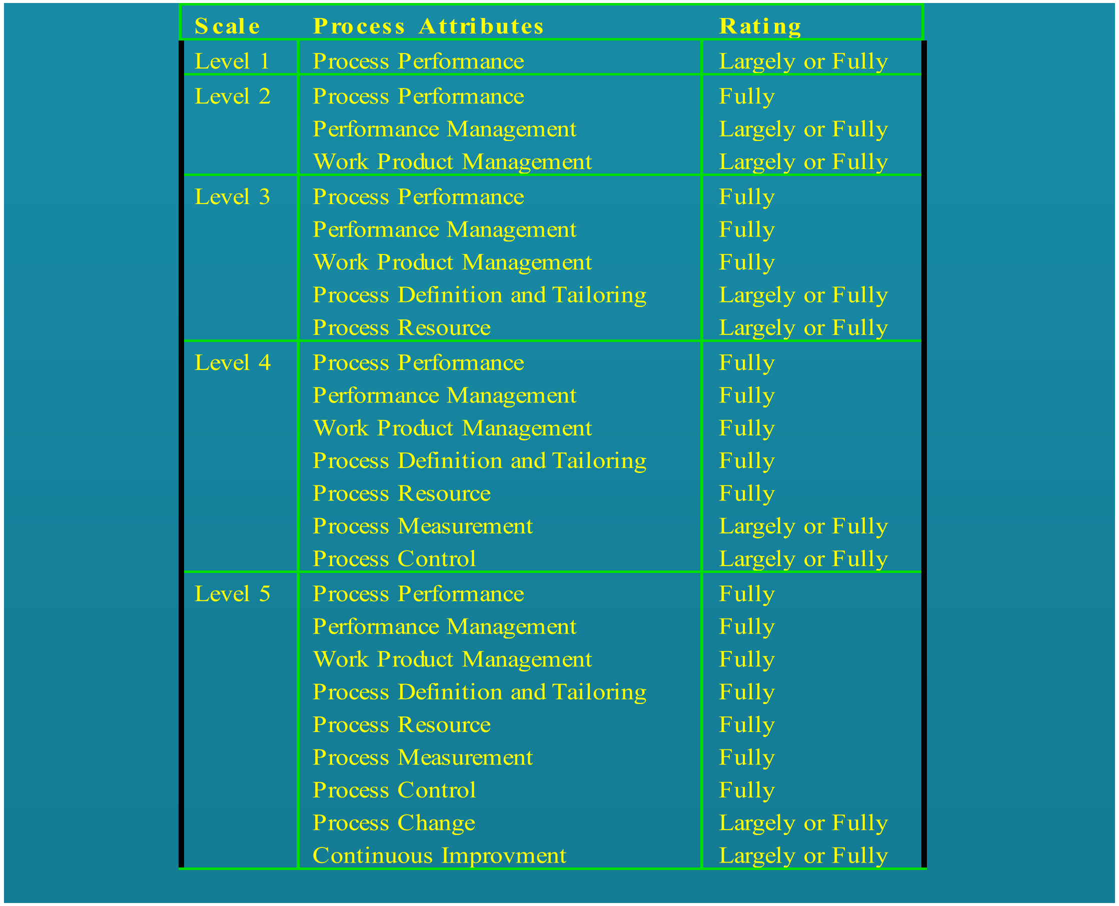
- Jeder Prozess wird einzelnen bewertet
  - 1 Prozessattribut (PA) für Stufe 1,
    2 PAs für die Stufen 2 bis 5
  - Bewertungsskala
    (Not / Partially / Largely / Fully Achieved)
- Resultate sind in Prozessprofilen dargestellt
  - Prozesse vs. PAs
  - Farbkodiert

N	...Not achieved
P	...Partially achieved
L	...Largely achieved
F	...Fully achieved


Prozess-
attribute
Bewertung
Prozesse

	PA 1.1 Perform	PA 2.1 Manage	PA 2.2 WPs	PA 3.1 Defined	PA 3.2 Resource	PA 4.1 Measure	PA 4.2 Control	PA 5.1 Change	PA 5.2 Improve
12: ENG.1.1 System Requirements Analysis and Design Process	L	L	P	N	L	?	?	?	?
13: ENG.1.2 Software Requirements Analysis Process	N	N	N	P	N	?	?	?	?
14: ENG.1.3 Software Design Process	L	P	P	P	P	?	?	?	?
15: ENG.1.4 Software Construction Process	L	P	P	N	P	?	?	?	?
16: ENG.1.5 Software Integration Process	L	P	P	N	L	?	?	?	?
17: ENG.1.6 Software Testing Process	L	P	L	N	P	?	?	?	?
19: ENG.2 System and Software Maintenance Process	L	P	P	N	P	?	?	?	?
29: MAN.2 Project Management Process	L	L	P	P	P	?	?	?	?

CAPABILITY LEVEL DEFINITION RULES



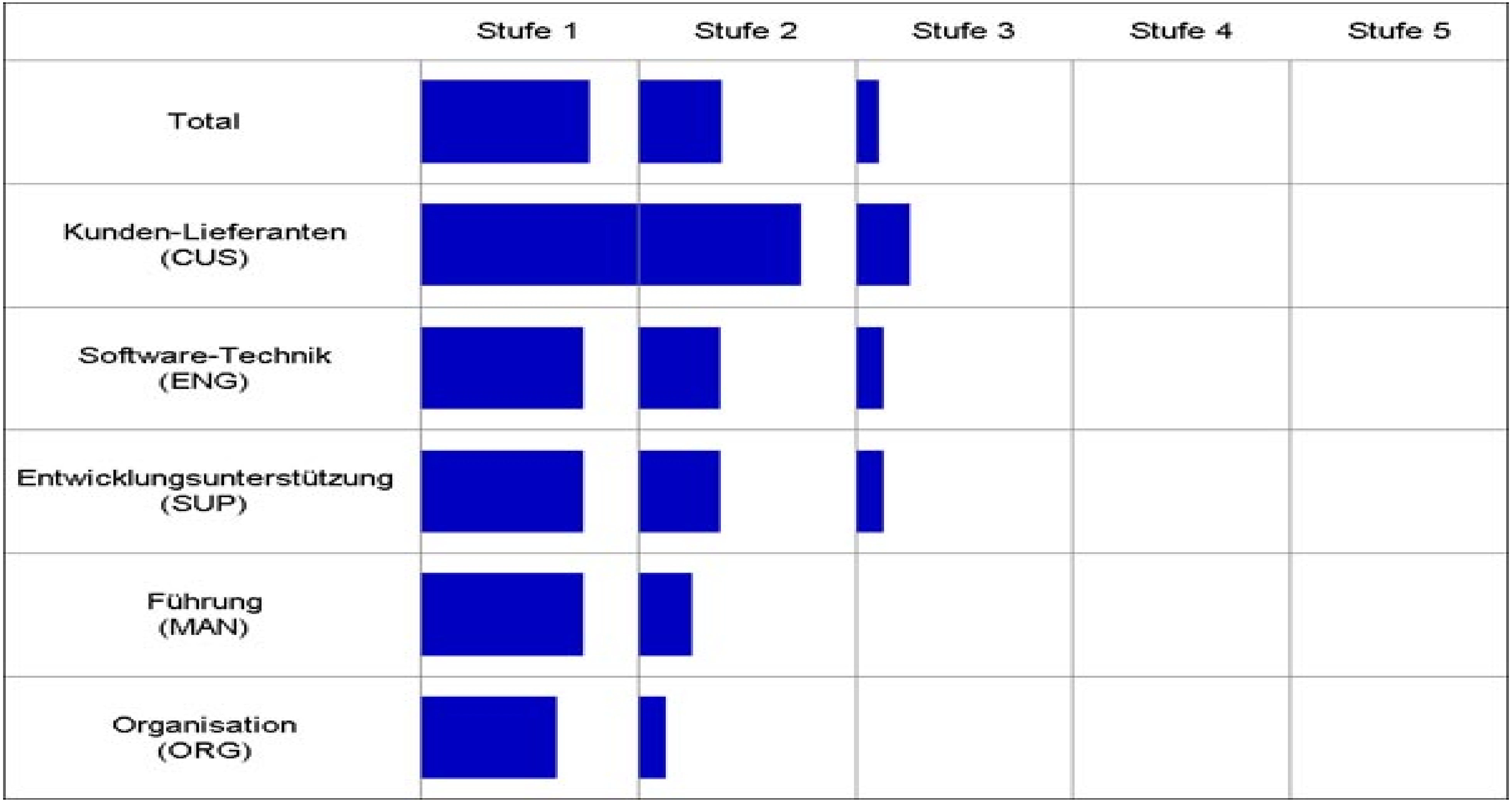
Scale	Process Attributes	Rating
Level 1	Process Performance	Largely or Fully
Level 2	Process Performance	Fully
	Performance Management	Largely or Fully
	Work Product Management	Largely or Fully
Level 3	Process Performance	Fully
	Performance Management	Fully
	Work Product Management	Fully
	Process Definition and Tailoring	Largely or Fully
	Process Resource	Largely or Fully
Level 4	Process Performance	Fully
	Performance Management	Fully
	Work Product Management	Fully
	Process Definition and Tailoring	Fully
	Process Resource	Fully
	Process Measurement	Largely or Fully
	Process Control	Largely or Fully
Level 5	Process Performance	Fully
	Performance Management	Fully
	Work Product Management	Fully
	Process Definition and Tailoring	Fully
	Process Resource	Fully
	Process Measurement	Fully
	Process Control	Fully
	Process Change	Largely or Fully
	Continuous Improvment	Largely or Fully

ASSESSMENT RESULT PRESENTATION: PROCESS PROFILE

CL5	P	N	N	N	N	N	N	N	N	N	N	F	F
	L	N	P	N	N	N	N	N	N	N	N	F	F
CL4	L	P	L	P	P	P	P	P	P	P	P	F	F
	L	L	L	P	P	P	P	P	P	P	P	F	F
CL3	F	L	F	L	L	L	P	P	F	F	F	F	F
	F	L	F	L	F	L	P	P	L	F	F	F	F
CL2	F	F	F	L	F	L	P	L	F	F	F	F	F
	F	F	F	F	F	F	P	F	F	F	F	F	F
CL1	F	F	F	F	F	F	L	F	F	F	F	F	F
	P1	P2	..	..	..	..	Pn						

Processes →

Aggregation der Daten



PROCESS CATEGORY EXAMPLE - MANAGEMENT (MAN)

MAN Processes

Manage the project (MAN.1)

Manage quality

Manage risks

Manage subcontractors

MAN.1 (Manage the project)

Process Purpose:

Define the processes necessary to establish, coordinate and manage
a project and the resources necessary to produce a product or service.

Successful Implementation =

- **define scope of work**
- **size, estimate, plan, track and measure the tasks and resources necessary to complete the work**
- **identify & manage project interfaces**
- **perform corrective action when**

CAPABILITY LEVEL 1

Capability Level: **Performed Process (Level 1)**

Description

The implemented process achieves its defined purpose.

Attribute(s)

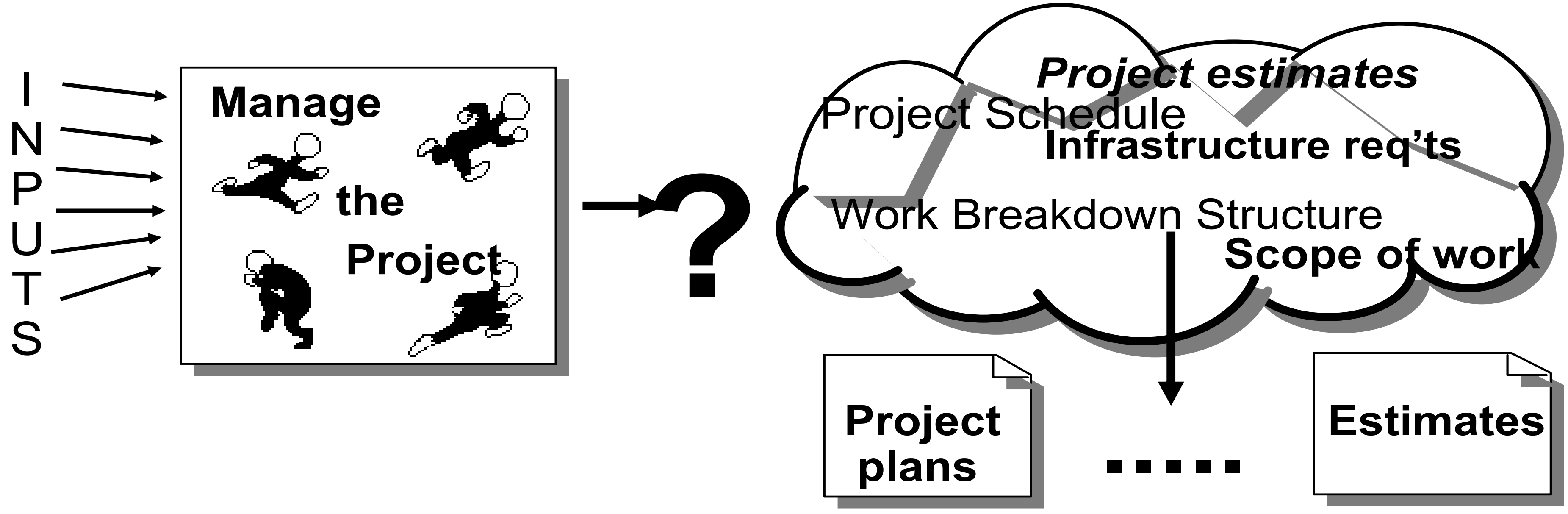
Process Performance (A 1.1)

- **utilization of a set of practices that are initiated and followed**
- **identified input work product**
- **identifiable output work products**
- **satisfies the purpose of the process**

EXAMPLE: Process Category MANAGE

MAN.1 Purpose:
Define the processes necessary to establish, coordinate and manage a project and the resources necessary to produce a product

Level 0 Incomplete **Level 1 Performed**



CAPABILITY LEVEL 2

Capability Level: Managed Process (Level 2)

Description

The Performed process delivers work products of acceptable quality within defined timescales and resource needs.

Attribute(s)

Performance management (A 2.1)

- management of the execution of the process to produce work products within stated time and resource requirements

Work product management (A 2.2)

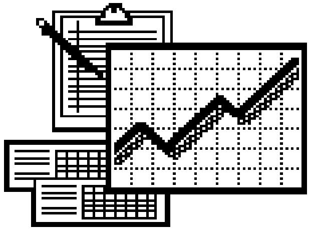
- work products - are documented and controlled
- meet their functional and non-functional
requirements - are in line with the product quality goals

EXAMPLE: Process Category MANAGE

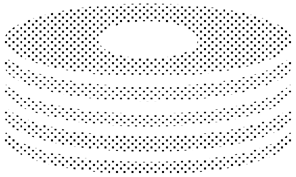
Level 2 Managed

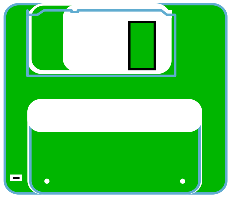
- Estimate resources
- Develop a plan(for the Project plan)
- Allocate resources
- Implement the plan
- Define the quality of the project plan

- Track compliance to the plan
- Make corrections to the plan
- Audit the contents of the project plan with applicable standards
- Work products reviews to verify quality



- Place work products under configuration management
- Manage the work products





CAPABILITY LEVEL 3

Capability Level: Established Process (Level 3)

Description

The Managed process performs using a defined process based upon good software engineering principles.

Attribute(s)

Process definition (A 3.1)

- **process uses a process definition based upon the standard process in order to contribute to the defined organizational business goals**

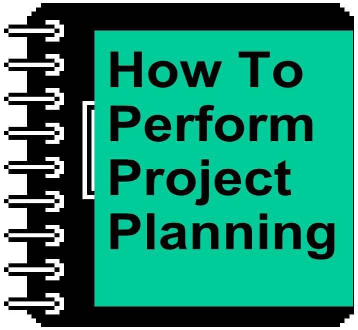
Process resource (A 3.2)

- **process effectively utilizes skilled human resources and infrastructure to contribute to the defined organizational business goals**

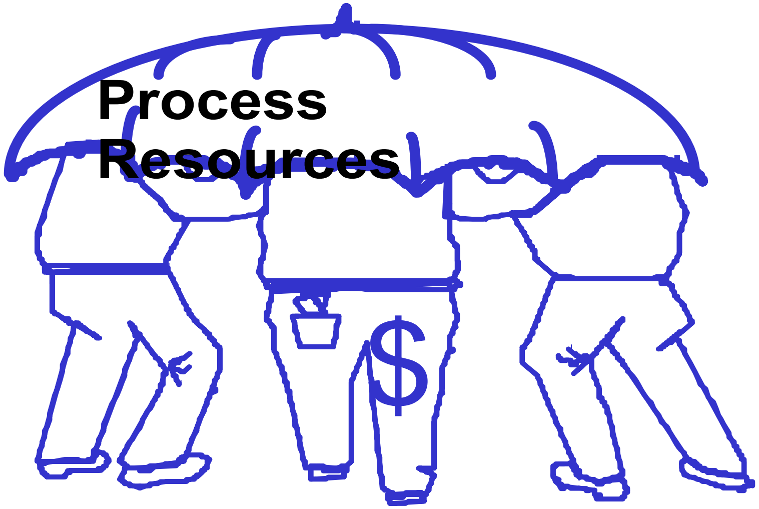
EXAMPLE: Process Category MANAGE

Level 3 Established

- Standardize the process for managing projects
- Document tailoring guidelines/rules
- Maintain the “How to” document



- Identify and allocate qualified resources to define the “How to” process
- Provide process definition



CAPABILITY LEVEL 4

Capability Level: **Predictable Process (Level 4)**

Description

The Established process is performed consistently within defined control limits to achieve its goals.

Attribute(s)

Process measurement (A 4.1)

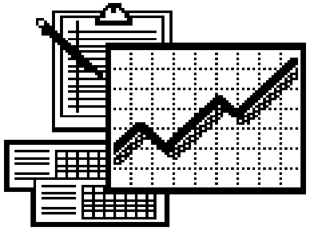
- process is supported by defined goals and measures which are used to ensure the achievement of overall process goals

Process control (A 4.2)

- process is controlled through the collection and analysis of measures to control and correct the performance of the process to reliably achieve the defined process goals.

EXAMPLE: Process Category MANAGE

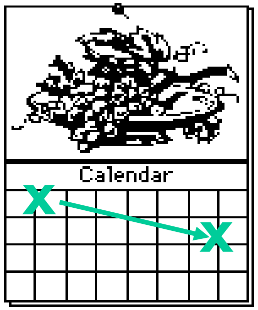
Level 4 Predictable



- Define process & product metrics/goals

- Collect defined measures ie. “How good is the planning process.”

- Analyze and use measures eg. replanning vs. initial planning



CAPABILITY LEVEL 5

Capability Level: **Optimizing Process (Level 5)**

Description

The Predictable process optimizes its performance to meet current and future needs and achieves repeatability in meeting defined business goals.

Attribute(s)

Process change (A 5.1)

- changes to the definition, management and performance of the process are controlled to achieve the business goals

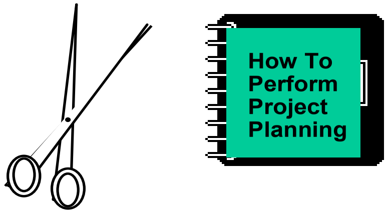
Continuous improvement (A 5.2)

- changes to the process are identified and implemented to ensure continuous improvement in fulfillment of the business goals

EXAMPLE: Process Category MANAGE

Level 5 Optimizing

- Identify improvements to the process ie. causal analysis.
- Improve the standard process ie. eliminate defects in project estimation
- Verify effectiveness of the changes
- Provide best practices and benchmark



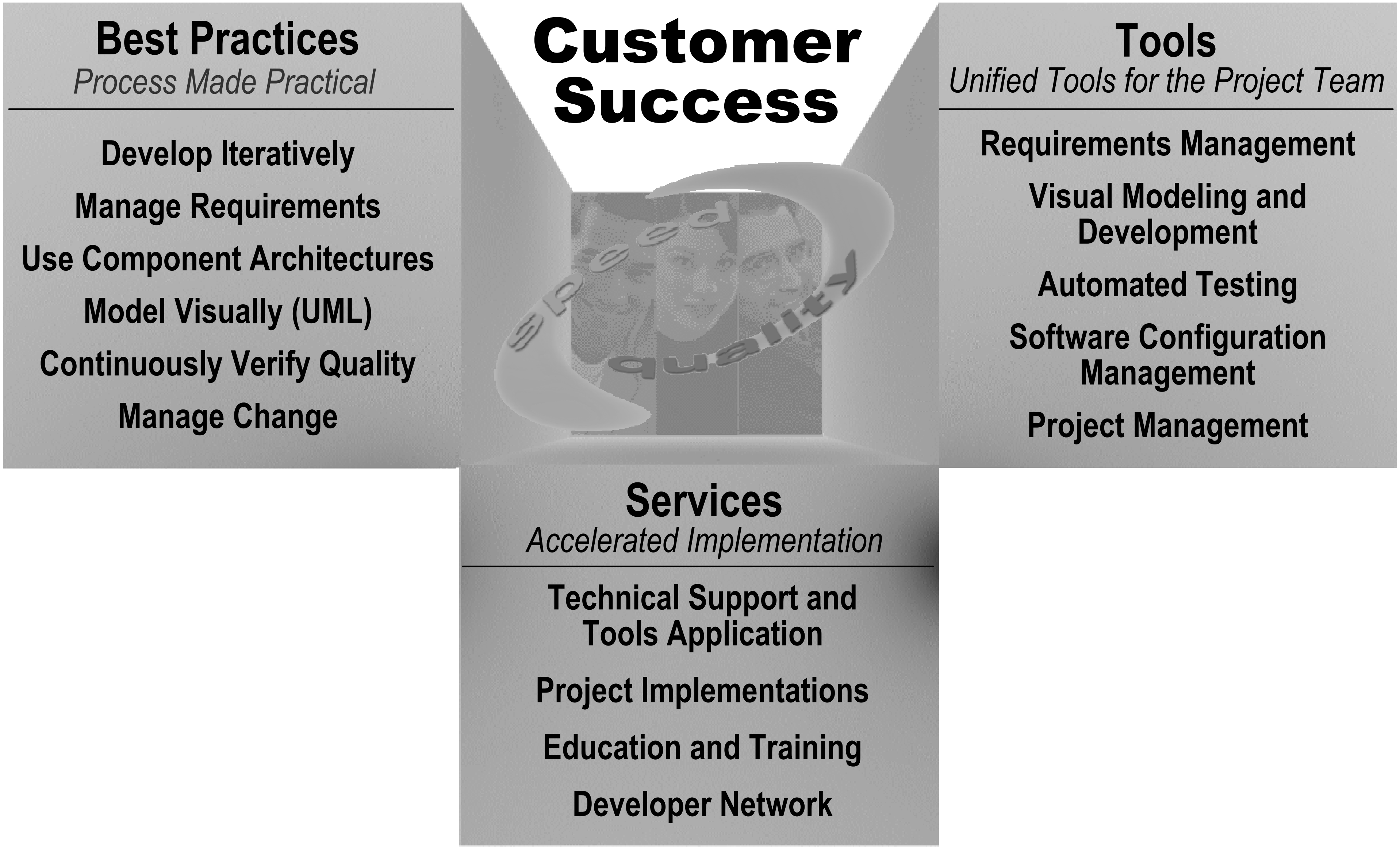
speed quality

Rational®

the software development company

Success Starts with Requirements Management

Rational: *The Platform for Software Development*



Customer Success

Best Practices
Process Made Practical

Develop Iteratively

Manage Requirements

Use Component Architectures

Model Visually (UML)

Continuously Verify Quality

Manage Change

Tools
Unified Tools for the Project Team

Requirements Management

Visual Modeling and Development

Automated Testing

Software Configuration Management

Project Management

Services
Accelerated Implementation

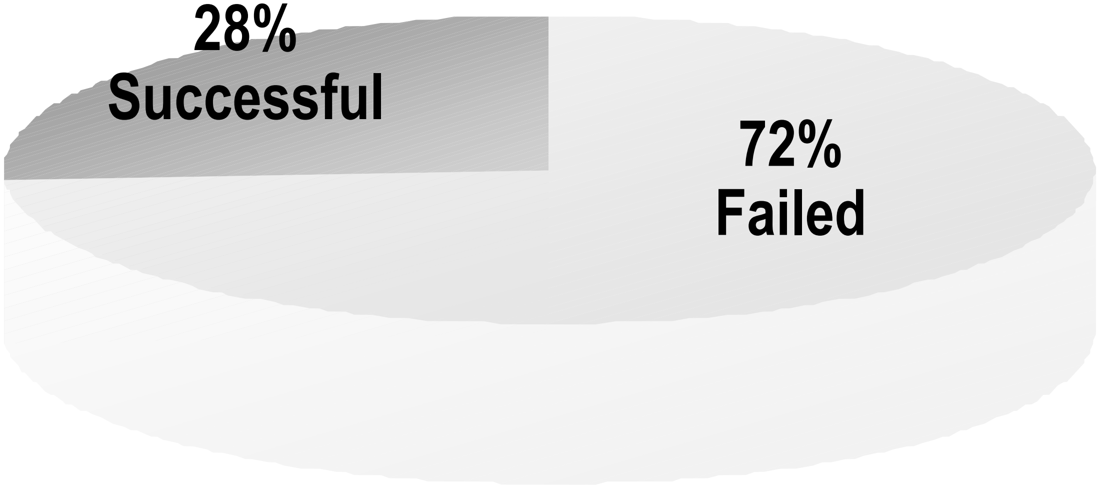
Technical Support and Tools Application

Project Implementations

Education and Training

Developer Network

The State of the Software Industry



- Over budget
  - Average cost overrun: 189%
- Delivered late
  - Average schedule overrun: 222%
- Failed to meet expectations
  - Average coverage: 61%

Standish Group, *CHAOS: A Recipe For Success, 2001*

Project Failure Factors

- ☑ Lack of User Input
- ☑ Unclear Objectives
- ☑ Incomplete Requirements and Specifications
- ☑ Changing Requirements and Specifications
- ☑ Lack of Planning

Standish Group, '00

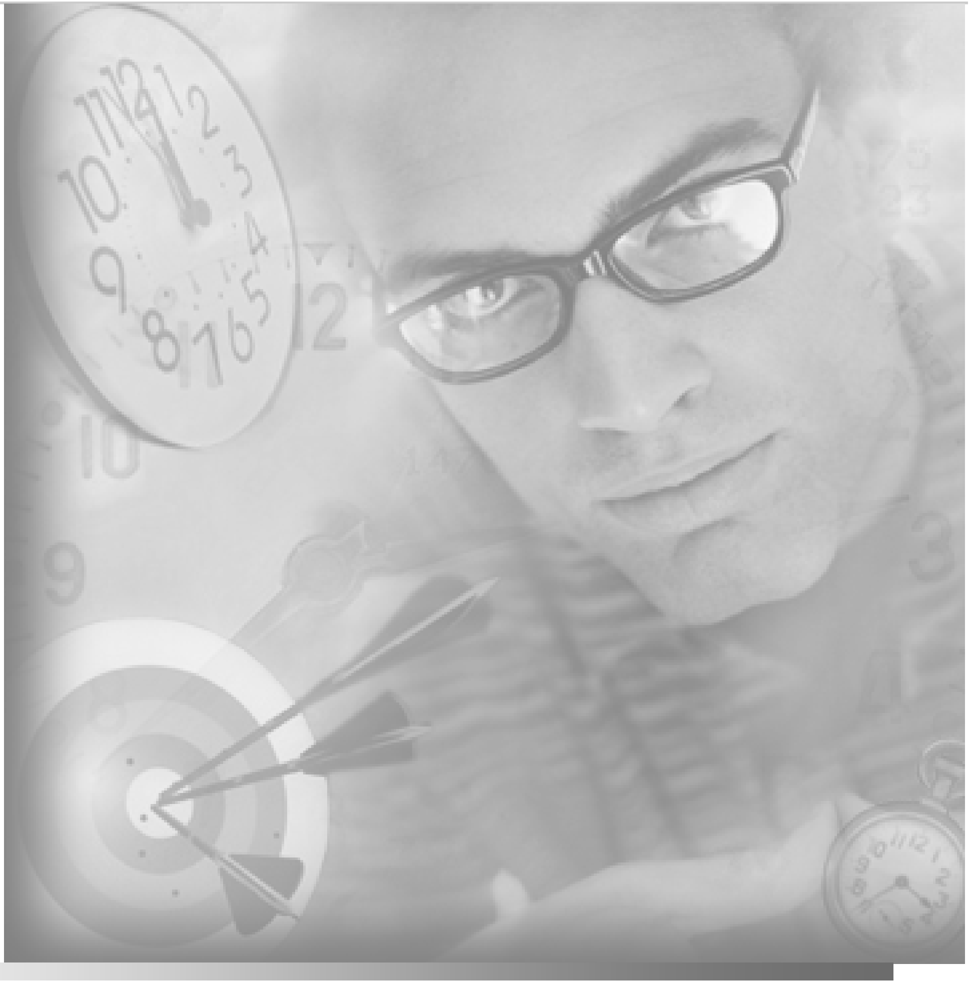
What is Requirements Management?

A systematic approach to eliciting, documenting, organizing, and tracking changing requirements

Ensuring your team identifies, builds, tests and documents the right system for your customer

Defining Software Success

- Satisfied customers
- Customers are satisfied when you
  - Meet expectations
  - Deliver on time
  - Deliver on budget



Success starts with requirements management

Rational RequisitePro

- Get control of your requirements
- Manage change
- Effectively communicate requirements

**Success starts with
requirements management**

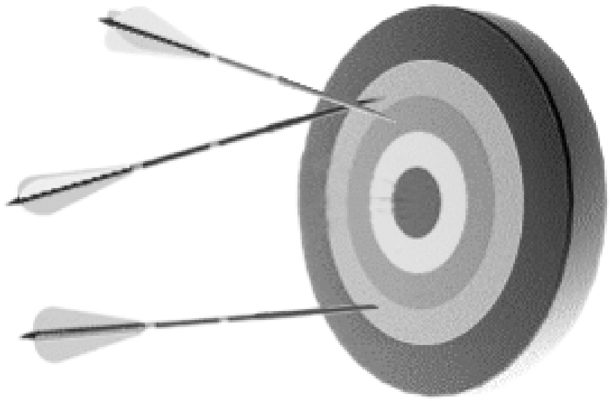
Rational RequisitePro

- Get control of your requirements
- Manage change
- Effectively communicate requirements

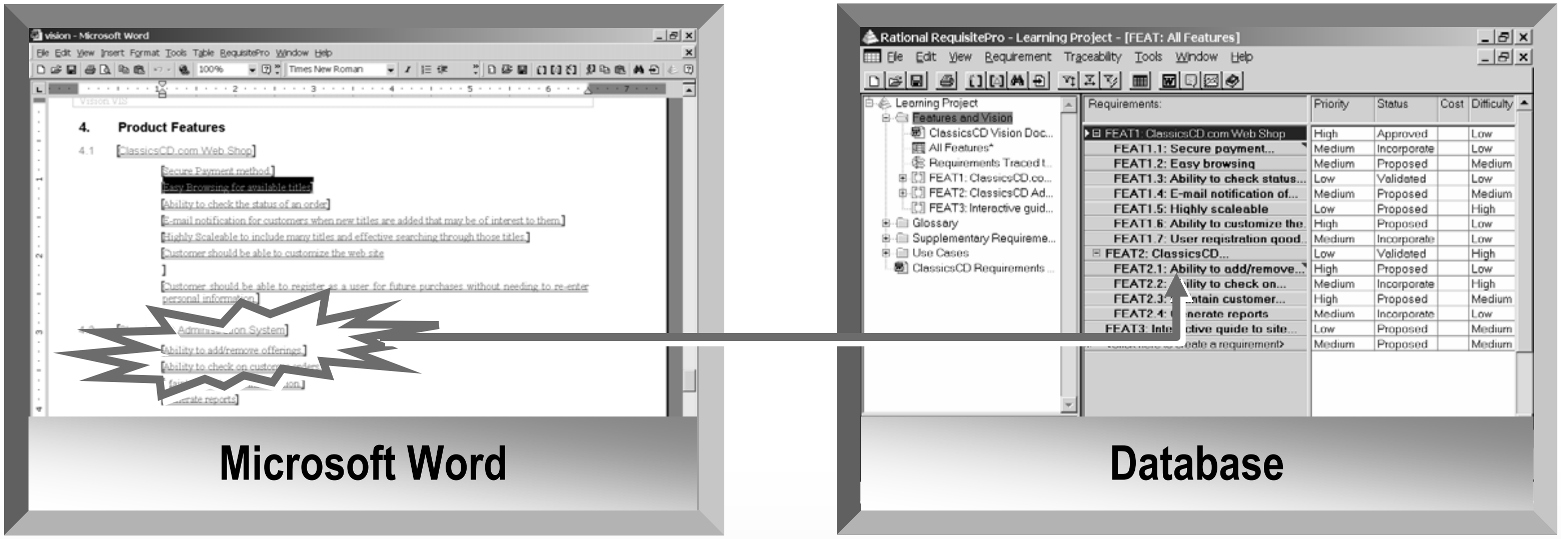
Success starts with requirements management

Uncontrolled Requirements

- Most organizations manage requirements with documents alone
  - Documents get e-mailed around and no one knows which is the latest version
  - Documents are not effective for analyzing, prioritizing, and tracking of information
  - It is difficult to extract useful information about project status from requirements documents



Managing Requirements With RequisitePro


vision - Microsoft Word
File Edit View Insert Format Tools Table RequisitePro Window Help
100%
Times New Roman
Rational RequisitePro - Learning Project - [FEAT: All Features]
File Edit View Requirement Traceability Tools Window Help
Learning Project
Features and Vision
ClassicsCD Vision Doc...
All Features*
Requirements Traced t...
FEAT1: ClassicsCD.co...
FEAT2: ClassicsCD Ad...
FEAT3: Interactive guid...
Glossary
Supplementary Requireme...
Use Cases
ClassicsCD Requirements ...

4. Product Features

4.1 [ClassicsCD.com Web Shop]

[Secure Payment method]

[Easy Browsing for available titles]

[Ability to check the status of an order]

[E-mail notification for customers when new titles are added that may be of interest to them]

[Highly Scaleable to include many titles and effective searching through those titles]

[Customer should be able to customize the web site

]

[Customer should be able to register as a user for future purchases without needing to re-enter personal information]

4.2 [[illegible] Administration System]

[Ability to add/remove offerings]

[Ability to check on custom[illegible]]

[[illegible]on]

[[illegible]erate reports]

Requirements:	Priority	Status	Cost	Difficulty
FEAT1: ClassicsCD.com Web Shop	High	Approved		Low
FEAT1.1: Secure payment...	Medium	Incorporate		Low
FEAT1.2: Easy browsing	Medium	Proposed		Medium
FEAT1.3: Ability to check status...	Low	Validated		Low
FEAT1.4: E-mail notification of...	Medium	Proposed		Medium
FEAT1.5: Highly scaleable	Low	Proposed		High
FEAT1.6: Ability to customize the...	High	Proposed		Low
FEAT1.7: User registration good...	Medium	Incorporate		Low
FEAT2: ClassicsCD...	Low	Validated		High
FEAT2.1: Ability to add/remove...	High	Proposed		Low
FEAT2.2: [illegible]ility to check on...	Medium	Incorporate		High
FEAT2.3: [illegible]ntain customer...	High	Proposed		Medium
FEAT2.4: [illegible]nerate reports	Medium	Incorporate		Low
FEAT3: Inte[illegible]ctive guide to site...	Low	Proposed		Medium
[illegible] create a requirement>	Medium	Proposed		Medium

Microsoft Word

Database

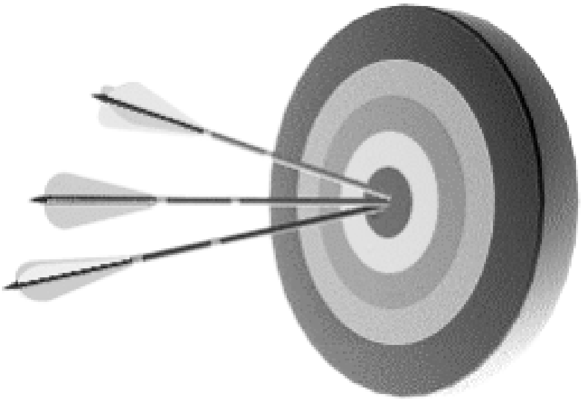
- ✓ A familiar way to work
- ✓ Contextual information
- ✓ Extensive formatting

- ✓ Security
- ✓ Organization
- ✓ Reporting and metrics

The Power of a Database and the Freedom of Word

Get Control of Your Requirements

- ✓ Access the latest requirements
  - Secure, centralized repository
- ✓ Organize and prioritize requirements
  - Dynamic linking to a database
- ✓ Know the status of your project
  - Filtering, sorting, and metrics



Leverage the Power of a Database and the Freedom of Word

Rational RequisitePro

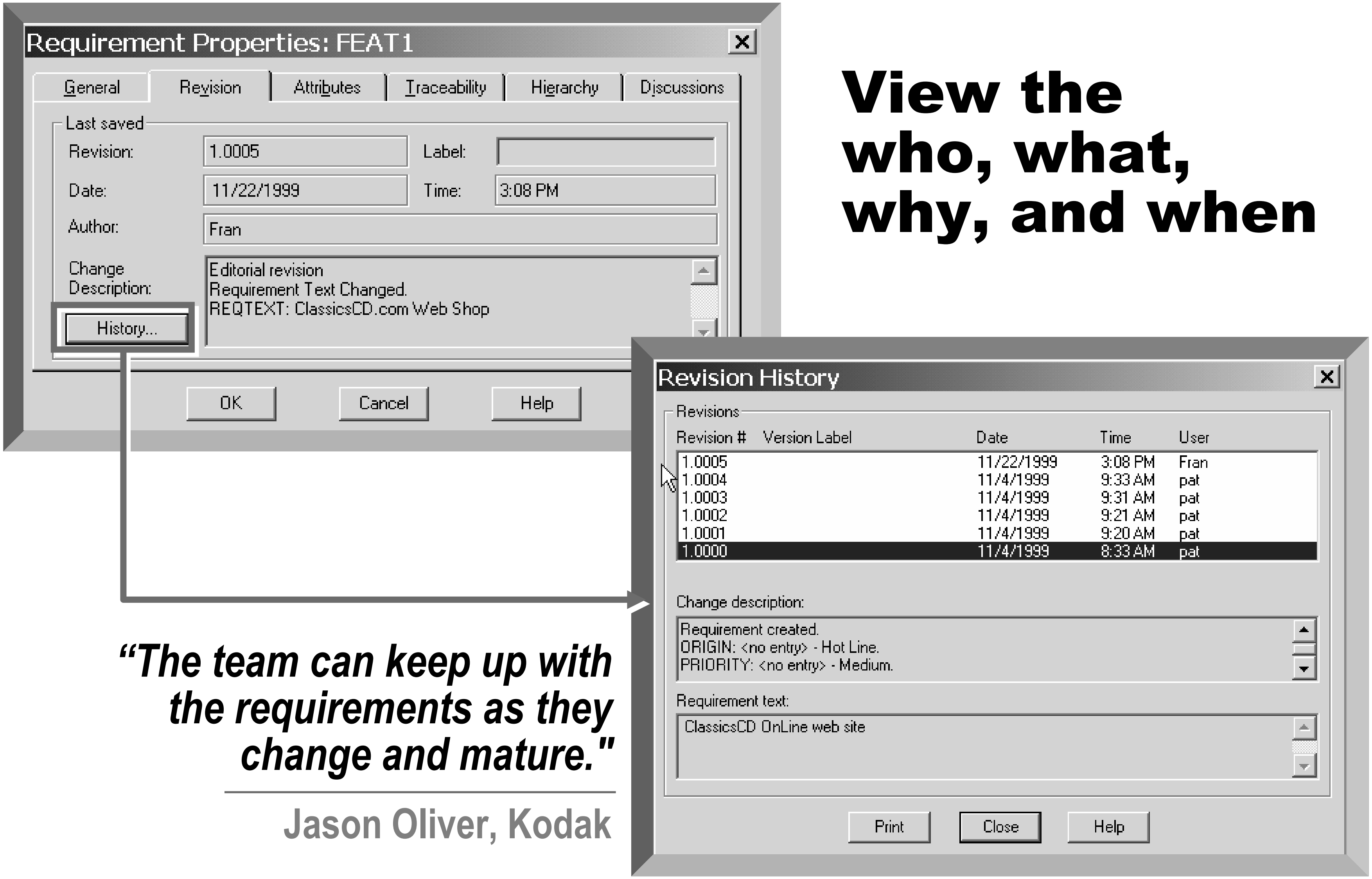
- Get control of your requirements
- Manage change
- Effectively communicate requirements

Success starts with
requirements management

Unmanaged Change

- Team members are often not aware of the changes
- Changes are accepted without considering the impact on the rest of the system

View the who, what, why, and when



Requirement Properties: FEAT1

Last saved

Revision: 1.0005

Label:

Date: 11/22/1999

Time: 3:08 PM

Author: Fran

Change Description:
Editorial revision
Requirement Text Changed.
REQTEXT: ClassicsCD.com Web Shop

Revision History

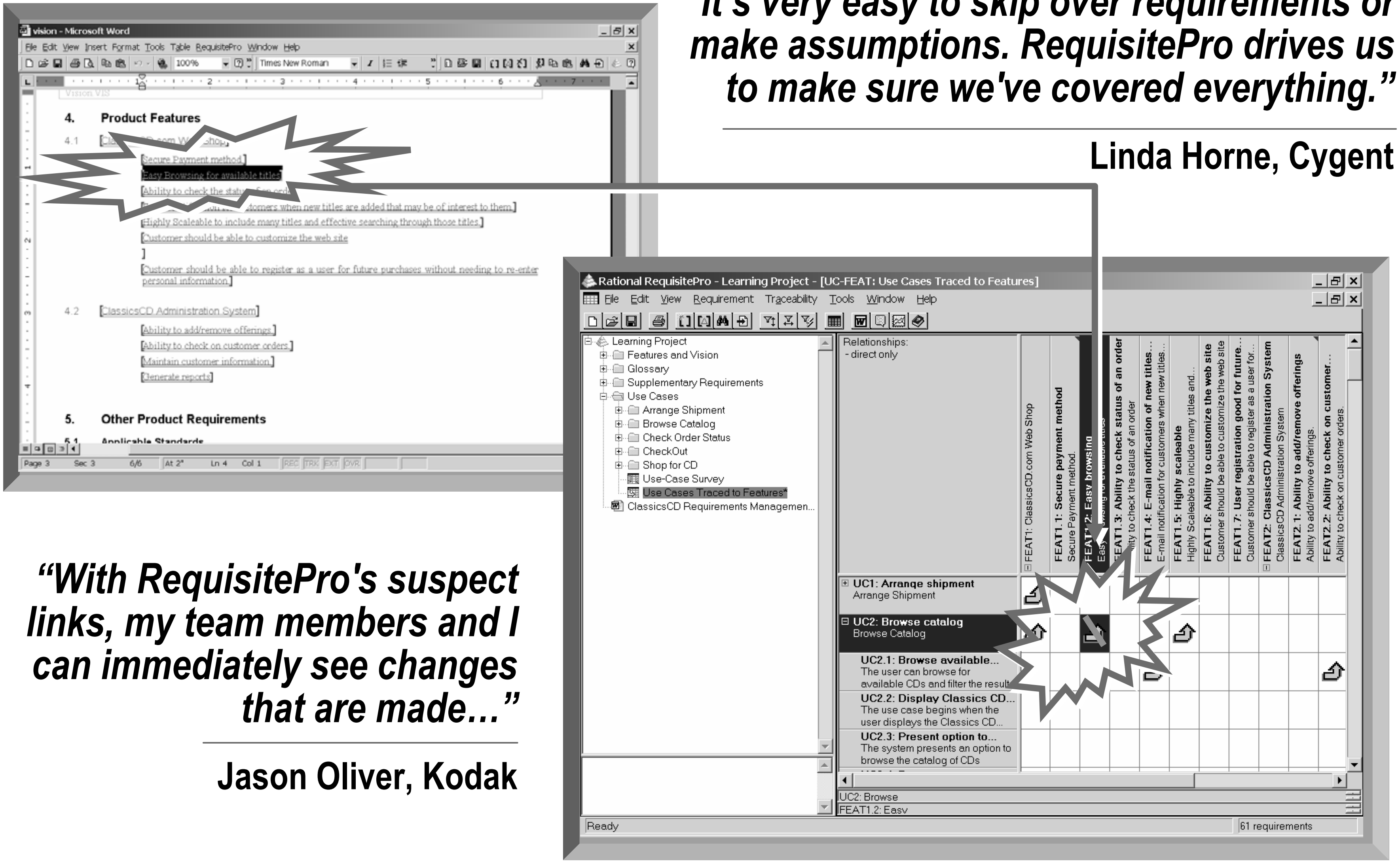
Revisions

Revision #	Version Label	Date	Time	User
1.0005		11/22/1999	3:08 PM	Fran
1.0004		11/4/1999	9:33 AM	pat
1.0003		11/4/1999	9:31 AM	pat
1.0002		11/4/1999	9:21 AM	pat
1.0001		11/4/1999	9:20 AM	pat
1.0000		11/4/1999	8:33 AM	pat

Change description:
Requirement created.
ORIGIN: <no entry> - Hot Line.
PRIORITY: <no entry> - Medium.

Requirement text:
ClassicsCD OnLine web site

> ***"The team can keep up with the requirements as they change and mature."***
>
> **Jason Oliver, Kodak**



vision - Microsoft Word

Vision VIS

4. Product Features

4.1 [ClassicsCD.com Web Shop]

[Secure Payment method]

[Easy Browsing for available titles]

[Ability to check the status of an order]

[... customers when new titles are added that may be of interest to them]

[Highly Scaleable to include many titles and effective searching through those titles]

[Customer should be able to customize the web site
]

[Customer should be able to register as a user for future purchases without needing to re-enter personal information]

4.2 [ClassicsCD Administration System]

[Ability to add/remove offerings]

[Ability to check on customer orders]

[Maintain customer information]

[Generate reports]

5. Other Product Requirements

5.1 Applicable Standards

Rational RequisitePro - Learning Project - [UC-FEAT: Use Cases Traced to Features]

Learning Project
- Features and Vision
- Glossary
- Supplementary Requirements
- Use Cases
  - Arrange Shipment
  - Browse Catalog
  - Check Order Status
  - CheckOut
  - Shop for CD
  - Use-Case Survey
  - Use Cases Traced to Features*
- ClassicsCD Requirements Managemen...

Relationships:
- direct only

FEAT1: ClassicsCD.com Web Shop

FEAT1.1: Secure payment method
Secure Payment method.

FEAT1.2: Easy browsing
Easy browsing for available titles

FEAT1.3: Ability to check status of an order
Ability to check the status of an order

FEAT1.4: E-mail notification of new titles...
E-mail notification for customers when new titles...

FEAT1.5: Highly scaleable
Highly Scaleable to include many titles and...

FEAT1.6: Ability to customize the web site
Customer should be able to customize the web site

FEAT1.7: User registration good for future...
Customer should be able to register as a user for...

FEAT2: ClassicsCD Administration System
ClassicsCD Administration System

FEAT2.1: Ability to add/remove offerings
Ability to add/remove offerings.

FEAT2.2: Ability to check on customer...
Ability to check on customer orders.

UC1: Arrange shipment
Arrange Shipment

UC2: Browse catalog
Browse Catalog

UC2.1: Browse available...
The user can browse for available CDs and filter the result

UC2.2: Display Classics CD...
The use case begins when the user displays the Classics CD...

UC2.3: Present option to...
The system presents an option to browse the catalog of CDs

UC2: Browse
FEAT1.2: Easy

Ready

61 requirements

“It’s very easy to skip over requirements or make assumptions. RequisitePro drives us to make sure we've covered everything.”

Linda Horne, Cygent

“With RequisitePro's suspect links, my team members and I can immediately see changes that are made…”

Jason Oliver, Kodak

Manage Change

- ✓ Make sure everyone is informed of changes
  - Requirement audit trail
- ✓ Understand the impact of accepting change
  - Traceability and suspect links

Understand the impact of change

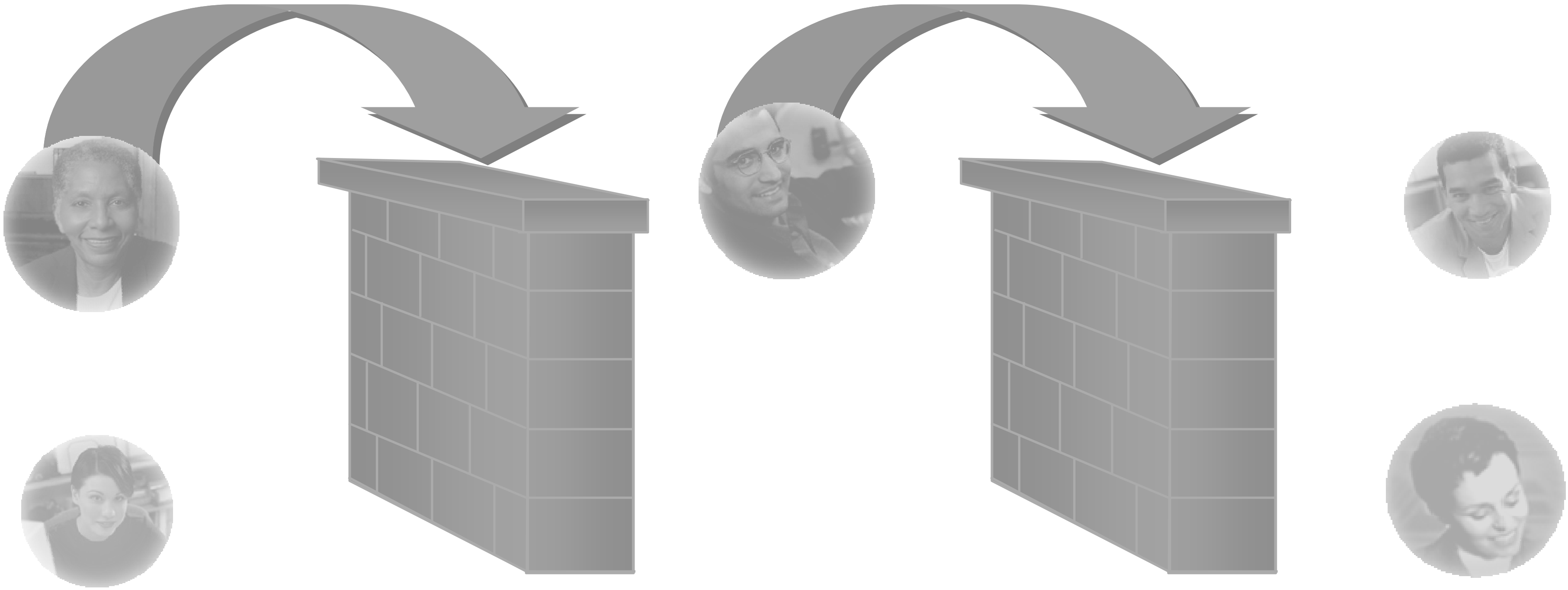
Rational RequisitePro

- Get control of your requirements documents
- Manage change
- Effectively communicate requirements

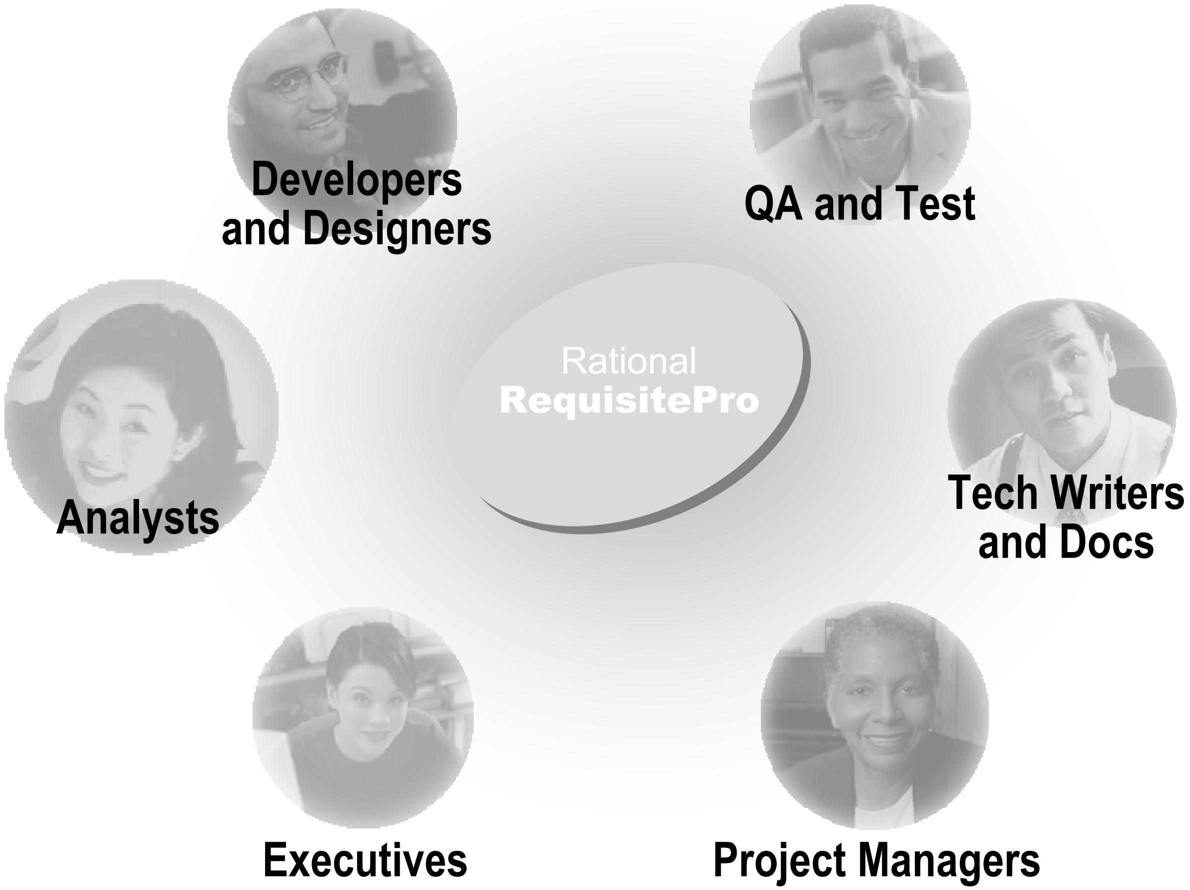
Success starts with
requirements management

Ineffective Communication

- Information is tossed over the wall between groups
- Tools don’t work together or share information
- Too much translation, redundancy and guesswork



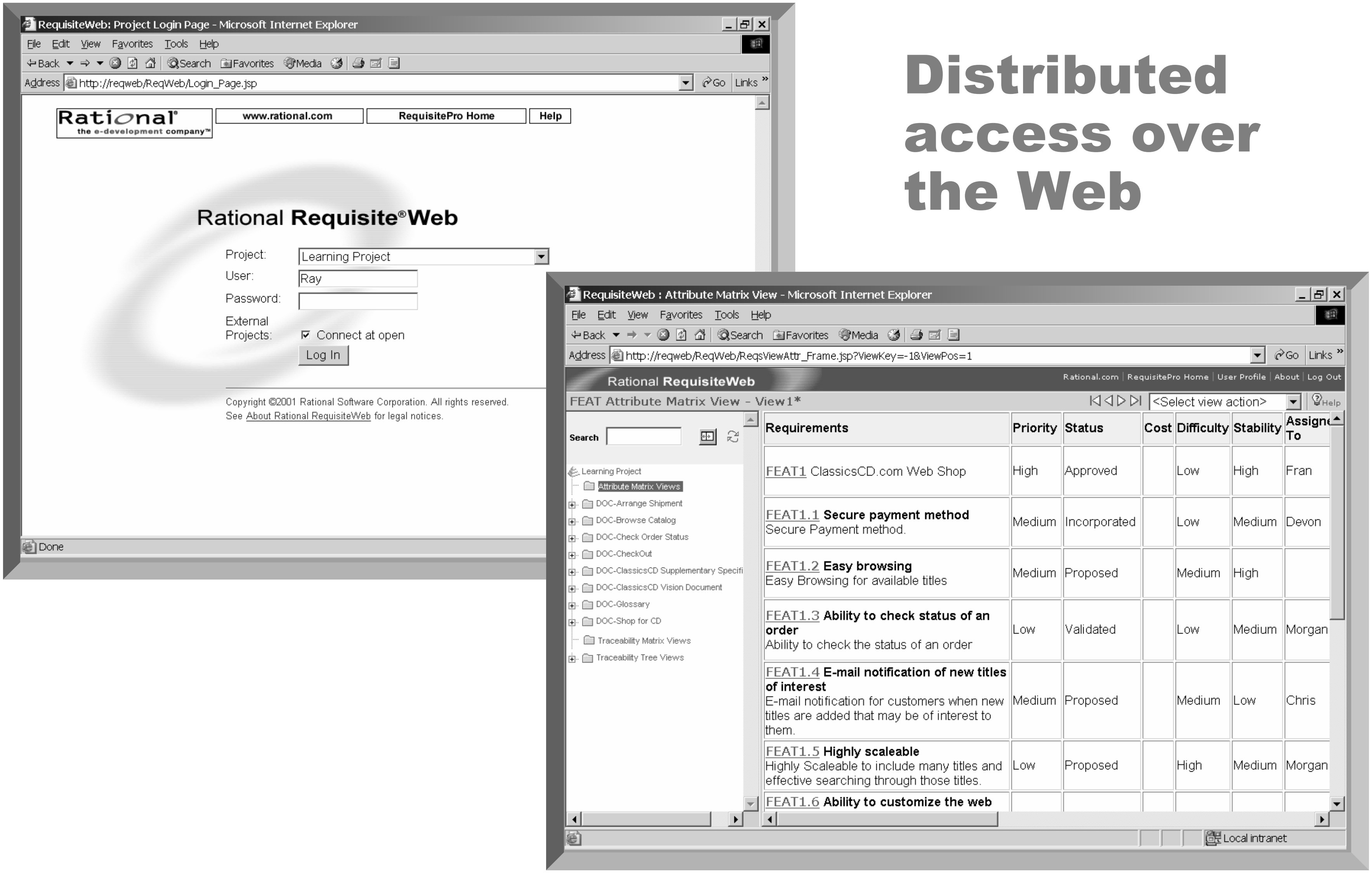
Everybody Needs Access to Requirements



"Rational RequisitePro has pulled everyone together on our projects."

Mireille Mallouh, MCI WorldCom

Access Requirements From Any Location


RequisiteWeb: Project Login Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://reqweb/ReqWeb/Login_Page.jsp
Go Links
Rational the e-development company
www.rational.com
RequisitePro Home
Help
Done
RequisiteWeb : Attribute Matrix View - Microsoft Internet Explorer
Address http://reqweb/ReqWeb/ReqsViewAttr_Frame.jsp?ViewKey=-1&ViewPos=1
Rational RequisiteWeb
Rational.com | RequisitePro Home | User Profile | About | Log Out
Search
Learning Project
Attribute Matrix Views
DOC-Arrange Shipment
DOC-Browse Catalog
DOC-Check Order Status
DOC-CheckOut
DOC-ClassicsCD Supplementary Specifi
DOC-ClassicsCD Vision Document
DOC-Glossary
DOC-Shop for CD
Traceability Matrix Views
Traceability Tree Views
Local intranet

Rational Requisite®Web

Project: Learning Project

User: Ray

Password:

External Projects: ☑ Connect at open

Log In

Copyright ©2001 Rational Software Corporation. All rights reserved.
See About Rational RequisiteWeb for legal notices.

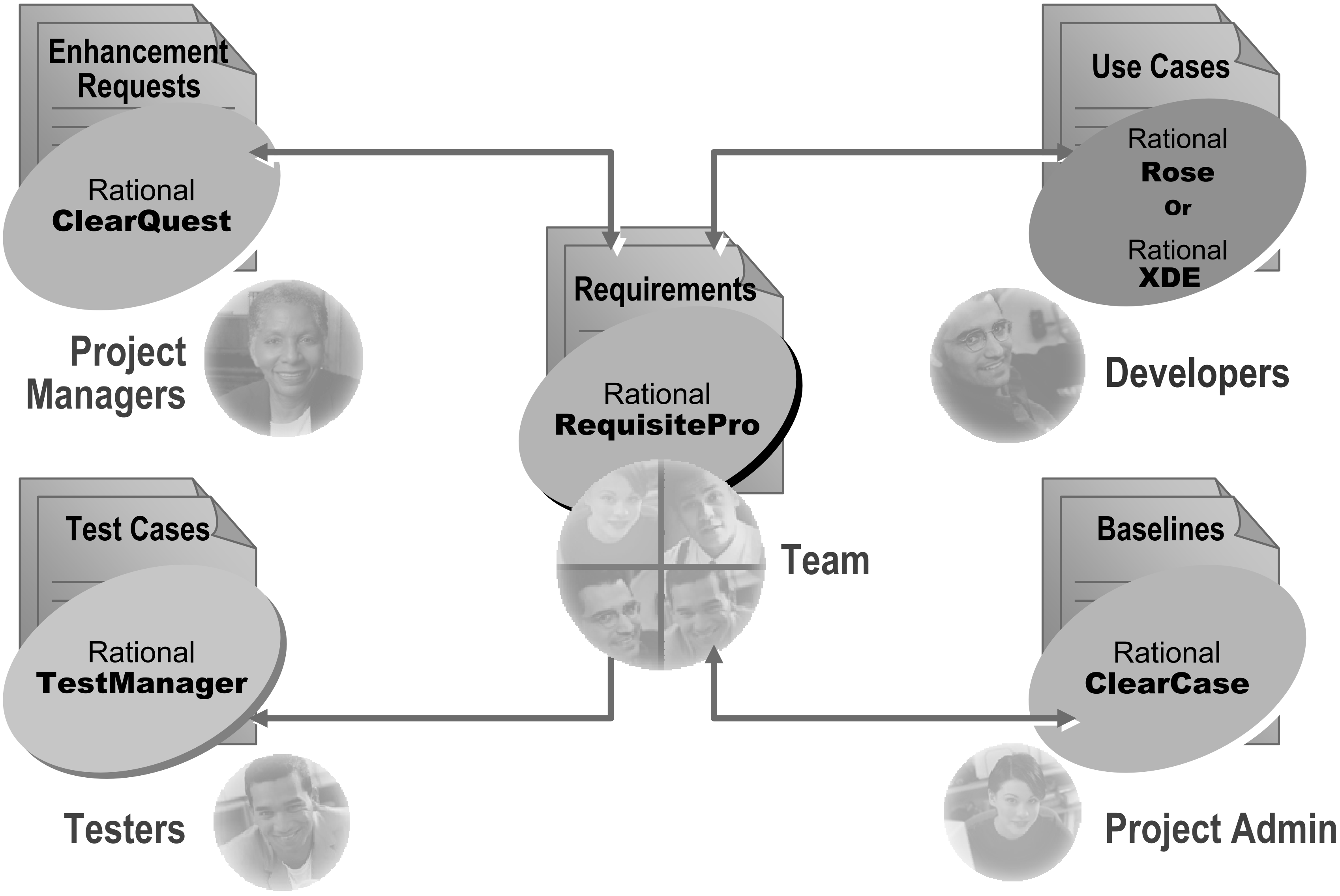
FEAT Attribute Matrix View - View1*

<Select view action>

Requirements	Priority	Status	Cost	Difficulty	Stability	Assigne To
FEAT1 ClassicsCD.com Web Shop	High	Approved		Low	High	Fran
FEAT1.1 **Secure payment method** Secure Payment method.	Medium	Incorporated		Low	Medium	Devon
FEAT1.2 **Easy browsing** Easy Browsing for available titles	Medium	Proposed		Medium	High	
FEAT1.3 **Ability to check status of an order** Ability to check the status of an order	Low	Validated		Low	Medium	Morgan
FEAT1.4 **E-mail notification of new titles of interest** E-mail notification for customers when new titles are added that may be of interest to them.	Medium	Proposed		Medium	Low	Chris
FEAT1.5 **Highly scaleable** Highly Scaleable to include many titles and effective searching through those titles.	Low	Proposed		High	Medium	Morgan
FEAT1.6 **Ability to customize the web**						

Distributed access over the Web

Integrate Requirements Across Tools and Teams



Enhancement Requests

Rational **ClearQuest**

Project Managers

Requirements

Rational **RequisitePro**

Team

Use Cases

Rational **Rose** or Rational **XDE**

Developers

Test Cases

Rational **TestManager**

Testers

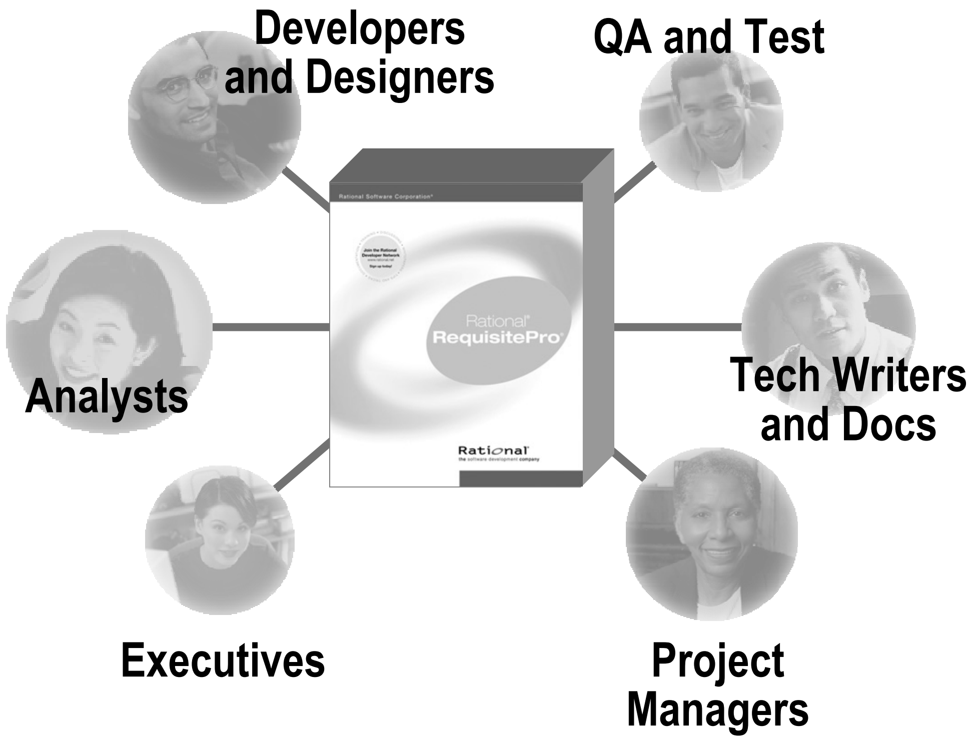
Baselines

Rational **ClearCase**

Project Admin

Effectively Communicate Requirements

- ✓ Easy access for all team members
  - RequisitePro
- ✓ Web access to requirements
  - RequisiteWeb
- ✓ Direct access to requirements
  - Guaranteed integrations



Integrate requirements across tools and teams

Succeed With Rational RequisitePro

- ✓ Meet your customers’ expectations
  - Get control over your requirements
  - Communicate your requirements effectively
- ✓ Deliver on time and on budget
  - Manage change

Success starts with requirements management

Rational Services

- Rational Developer Network
  - Rational RequisitePro Knowledge Center
  - Requirements Management Concepts Discussion Forum
  - Tech Notes
  - Documentation
  - General Availability
- Rational University
  - Business Modeling with UML (Methodology)
  - Requirements Management with Use Cases (Methodology)
  - Fundamentals of Rational RequisitePro – WBT
  - Fundamentals of Rational RequisitePro (v2001A and v2002)
  - Rational Suite AnalystStudio Workshop (analyst level course)

Rational Services

- Professional Services
  - Rational RequisitePro QuickStart
  - Rational Metrics Assessment
- Technical Support
  - Silver & Gold Maintenance Support
  - Scheduled Weekend Service

About Rational Software

One of the world’s largest software companies

- Founded in 1981
- 3,500+ employees
- $815 M revenues FY ended 3/01
- 65+ worldwide locations
- $160 million R&D spending
- Over 500,000 users

96 of the *Fortune 100*
rely on the Rational solution

Rational's Mission

"To ensure the success of customers who depend on developing or deploying software"

Thank You!

Software Development: *Many Stakeholders, Different Goals*

Project Leader

- I can quickly react to changing requirements
- Empowers me to make informed decisions
- Helps my team meet the project deadline

Liberates my team from unproductive activities

Analyst

- Makes requirements manageable
- Project priorities are clearly communicated
- Gives my team quick access to up to date requirements

Empowers my team to deliver the right solution

freedom TO CREATE

Rational User Conference 2002

August 18-22

Lake Buena Vista, Florida

Walt Disney World Swan & Dolphin

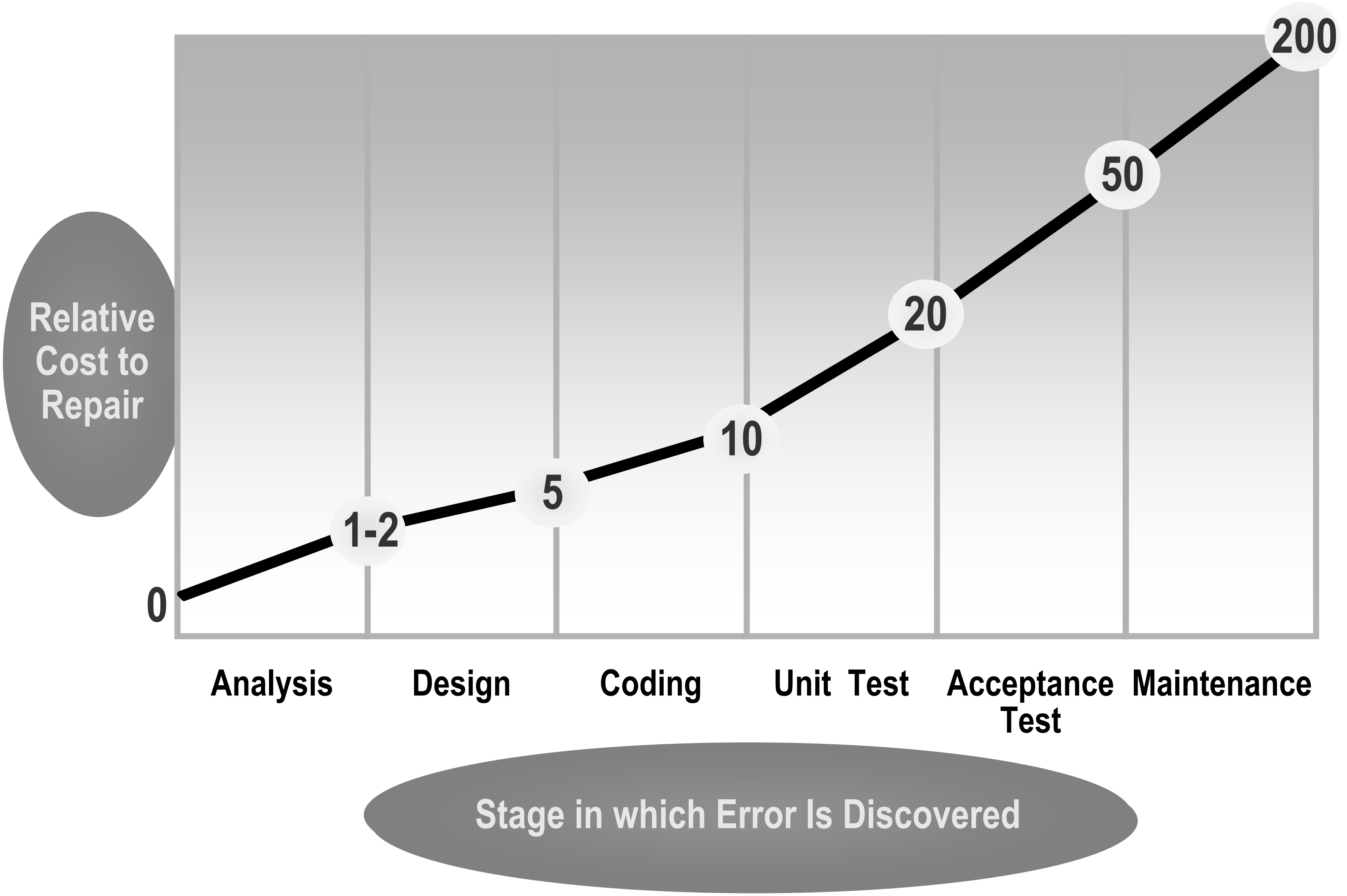
For more information visit
www.rational.com/ruc

- Over 2,000 customers and partners
- Over 200 sessions organized within 10 tracks
- Pre & Post Conference Training Session
- Rational Suite Hands-on Workshop
- Keynotes with industry leading experts
- Exhibit Hall showcasing complimentary products and services
- Unlimited Network Opportunities - Solutions Center, Birds of a Feather, Luncheon discussion tables, evening receptions

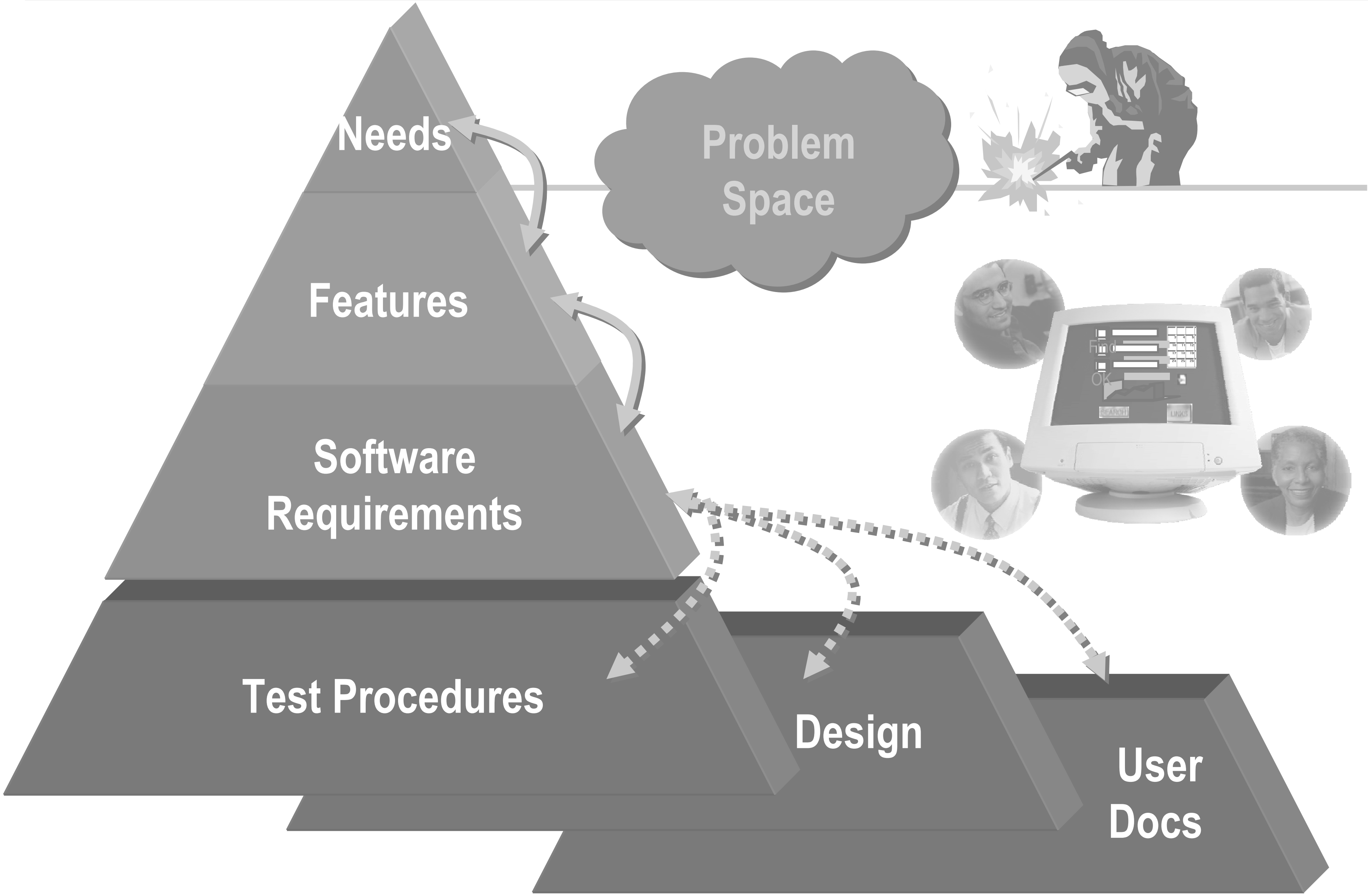
Rational®
the software development company

The Following Slides are Optional Grabber Slides

High Cost of Requirements Errors



Requirements Drive All Software Activities



Integrate Requirements Across Tools and Teams

Rational RequisitePro Integrations

Tool	Benefit
Rational Rose or XDE	Jumpstart design with requirements in use cases
Rational ClearQuest	Trace requirements to original stakeholder needs
Rational ClearCase	Baseline requirements along with code and models
Rational Unified Process	Leverage proven best-practices in your requirements management process
Rational TestManager	Drive the development of test cases from live requirements
Rational SoDA	Generate comprehensive, up-to-date project documentation

Tell Us What You Think

Are these slides working for you?
What would you like to see different?
Please send us your feedback!
http://field.rational.com/feedback/form.htm