



**ICB**

Institut für Informatik und  
Wirtschaftsinformatik

Brian Berenbach, Maya Daneva,  
Jörg Dörr, Samuel Fricker,  
Vincenzo Gervasi, Martin Glinz,  
Andrea Herrmann, Benedikt Krams,  
Nazim H. Madhavji, Barbara Paech,  
Sixten Schockert, Norbert Seyff (Eds.)



# 17th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2011)

**ICB-RESEARCH REPORT**

Proceedings of the REFSQ 2011 Workshops REEW,  
EPICAL and RePriCo, the REFSQ 2011 Empirical Track  
(Empirical Live Experiment and Empirical Research  
Fair), and the REFSQ 2011 Doctoral Symposium



Die Forschungsberichte des Instituts für Informatik und Wirtschaftsinformatik dienen der Darstellung vorläufiger Ergebnisse, die i. d. R. noch für spätere Veröffentlichungen überarbeitet werden. Die Autoren sind deshalb für kritische Hinweise dankbar.

The ICB Research Reports comprise preliminary results which will usually be revised for subsequent publications. Critical comments would be appreciated by the authors.

---

Alle Rechte vorbehalten. Insbesondere die der Übersetzung, des Nachdruckes, des Vortrags, der Entnahme von Abbildungen und Tabellen – auch bei nur auszugsweiser Verwertung.

All rights reserved. No part of this report may be reproduced by any means, or translated.

---

**Proceedings Edited By:**

Brian Berenbach  
Maya Daneva  
Jörg Dörr  
Samuel Fricker  
Vincenzo Gervasi  
Martin Glinz  
Andrea Herrmann  
Benedikt Krams  
Nazim H. Madhavji  
Barbara Paech  
Sixten Schockert  
Norbert Seyff

**ICB Research Reports**

**Edited by:**

Prof. Dr. Heimo Adelsberger  
Prof. Dr. Peter Chamoni  
Prof. Dr. Frank Dorloff  
Prof. Dr. Klaus Echtele  
Prof. Dr. Stefan Eicker  
Prof. Dr. Ulrich Frank  
Prof. Dr. Michael Goedicke  
Prof. Dr. Volker Gruhn  
Prof. Dr. Tobias Kollmann  
Prof. Dr. Bruno Müller-Clostermann  
Prof. Dr. Klaus Pohl  
Prof. Dr. Erwin P. Rathgeb  
Prof. Dr. Enrico Rukzio  
Prof. Dr. Albrecht Schmidt  
Prof. Dr. Rainer Unland  
Prof. Dr. Stephan Zelewski

---

**Contact:**

Institut für Informatik und  
Wirtschaftsinformatik (ICB)  
Universität Duisburg-Essen  
Universitätsstr. 9  
45141 Essen  
Tel.: 0201-183-4041  
Fax: 0201-183-4011  
Email: [icb@uni-duisburg-essen.de](mailto:icb@uni-duisburg-essen.de)

**ISSN 1860-2770 (Print)**  
**ISSN 1866-5101 (Online)**



## Managing Imperfection in Requirements: a Method and a Jigsaw Puzzle Metaphor

Maria Pinto-Albuquerque<sup>1,2</sup>, Awais Rashid<sup>1</sup>,

<sup>1</sup>School of Computing and Communications, Lancaster University, United Kingdom

<sup>2</sup>Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal

[maria.albuquerque@iscte.pt](mailto:maria.albuquerque@iscte.pt), [marash@comp.lancs.ac.uk](mailto:marash@comp.lancs.ac.uk)

**Abstract.** Effective requirements engineering in the presence of imperfection remains a major research problem and there is a lack of metaphors to aid communication during consultation with stakeholders. We propose a metaphor to promote and improve communication between stakeholders and requirements engineers when working together on requirements documentation, to detect and handle imperfection. The metaphor is based on jigsaw puzzles, where each puzzle piece represents a concern (i.e., a set of requirements). When the requirement's text contains imperfections potentially leading to conflicts with other concerns, the respective puzzle pieces have a matching that almost fits but not perfectly. We argue that using such a metaphor fosters team work and communication towards detecting and analyzing imperfections in stakeholder consultation meetings. It also makes stakeholders feel that imperfections in requirements documentation are their problem too. Having the jigsaw puzzle, a game widely known and very easy to learn and play, provides a good metaphor for group work among persons with heterogeneous background, and introduces fun in a work usually perceived as boring. Together with the jigsaw puzzle metaphor we present a method that receives a text document and through the integration of tools and heuristics supports the detection of imperfections, in particular conflicts. The requirements engineer can then select the most pertinent conflicts and require the appropriate tool to produce a jigsaw puzzle to be used in a consultation meeting with stakeholders. We have already conducted an experiment, which indicated interesting results, in particular an effective improvement of communication. We plan to perform two more experiments.

Keywords: Requirements, Communication, Team work, Stakeholders, Imperfection, Conflict Detection, Jigsaw puzzle metaphor, Empirical evaluation

### 1 Problem and its Relevance

The development of software is a complex task, requiring the acquisition and processing of a huge amount of information. Software engineering involves a large number of decisions to be taken at the right time. These decisions are based on

information and produce information that should be recorded and presented in an accessible manner for both the software engineers and the stakeholders.

Providing appropriate and usable mechanisms to represent information is difficult because information in software development is heterogeneous, inputted and used by a heterogeneous professional community and also inherently not perfect. This is particularly true in requirements engineering. Existing software development approaches aim to come to a perfect set of requirements. But they also acknowledge the difficulty of defining requirement specifications with desired quality. Due to this difficulty some approaches have started to include tools, like iteration and heuristics, to ensure the quality of the resulting software system. But for a non-trivial system, it is unrealistic to assume that the development is performed in an ideal case where all the imperfect information will become perfect, the requirements will not change and/or it will be possible to correct all the problems caused by wrong assumptions. Furthermore, it is unrealistic that it will be possible to perform the large number of iterations needed as these are restricted by time and financial constraints [13].

Requirements are pieces of information where imperfection, like incompleteness, ambiguity, and conflict are inherently present. It is not realistic to think that imperfection can be removed before or during systems engineering. Thus, decisions during development are almost always made in the presence of imperfection. The best we can aim is at making explicit the information on the imperfection facets of requirements and deal effectively with such imperfection, for instance, resolve it where possible or be mindful of it when making decisions [12].

By imperfection we mean things like incompleteness, misplacement, ambiguity, and conflict. In our research we focus on conflicting or ambiguous requirements expressions, and not conflict due to different interests of stakeholders and/or software engineers.

The ultimate reason to address imperfection in requirements is that if imperfection is not made explicit and appropriately handled, it will be supposed that requirements are perfect. Development will proceed and the system produced will contain errors or unwanted characteristics. Big disasters, costing even human lives, due to imperfection in requirements are well known in systems engineering history (e.g., Therac-25 [11], Mars Climate Orbiter [18], etc.). It is also known that the sooner the errors are corrected the less costly they are [5].

## **2 Proposed Solution**

### **2.1 Novel Aspects of the Work**

We propose a metaphor based on jigsaw puzzles in order to promote and improve communication amongst system stakeholders and between stakeholders and requirements engineers. This metaphor is to be used during consultation meetings with stakeholders, fostering group work to review requirements documentation in order to detect and handle imperfection. In these meetings, typically requirements engineers want to focus in a small number of concerns (say 4 to 6) [17] and facilitate

involvement of stakeholders. Our hypothesis is that this jigsaw puzzle with imperfect interlocking shapes between pieces (representing concerns with potential conflict) fosters team work and communication towards detecting and analyzing imperfections in requirements meetings. When the requirement's text contains imperfections potentially leading to conflicts with other concerns, the respective puzzle pieces have a matching that almost fits but not perfectly. We also foresee that this metaphor will improve stakeholder's awareness that imperfections in requirements are their problem too and thus increase their commitment to cooperate in solving those imperfections. We believe this will happen because both stakeholder's and engineers have to work with the same common "document", i.e. the puzzle, and not each with its own copy of requirements text.

We propose a method to support the management of imperfect information in requirements. This method receives requirements documents written in natural language, and integrates existing tools alongside with heuristics to detect, classify, and prioritize imperfections. For the most pertinent cases of requirements conflicting, the requirements analyst(s) has the option to produce, supported by a developed tool, a set of jigsaw puzzle pieces to be used in consultation meetings with stakeholders to review that part of the requirements documentation. The method prescribes that the remaining imperfections are handled through annotations. These annotations record information about the imperfection and describe how to proceed in order to manage it (e.g., what to ask, to whom). The method is to be iteratively repeated, as long as the requirements team considers it useful and according to time and budget constraints. This means that, at different moments during development, diverse parts of the requirements can be selected to be discussed in consultation meetings with stakeholders, or amongst analysts. For each set of requirements a different jigsaw puzzle set is produced.

We propose a solution based on a visual metaphor because we believe metaphors may provide a good solution to address the needs of making imperfections in requirements explicit via visualization. A visual metaphor is an analogy that underlies a graphical representation of an abstract entity or concept with the goal of transferring properties from the domain of the graphical representation to that of the abstract entity or concept [4, 9]. The more complex visual metaphors (more complex than two- or three-dimension geometric ones) have been applied in its vast majority to artifacts and software that already exist (to show metrics, for program comprehension, for reverse engineering) [4]. Using visualization to support creation and decision in software development requires a different mindset: to provide visualizations for artifacts and organization that are not known, they are being built. We believe a well-assembled metaphor, making an analogy with a building process where in the beginning we do not know the organization but it starts to appear, can provide interesting tools for software development, focused on its "being built" nature. The jigsaw puzzle game provides a good metaphor for building something from different pieces that have to be correctly assembled to yield a final common product.

In order to accommodate usage by different professional profiles that cooperate in software development we should provide a metaphor that builds on a well-known concept to be easily used by the broadest professional profiles. The jigsaw puzzle, a

game widely known and very easy to learn and play, potentially provides a good metaphor for group work among persons with heterogeneous background. Last but not the least we should not forget that these consultation meetings with stakeholder's are usually boring and being the jigsaw puzzle a game introduces fun in work!

## 2.2 Research Methodology

We planned to perform three experiments. These experiments are concerned with understanding how requirements engineers and system stakeholders communicate and work together to detect, and analyze imperfections present in requirements documentation for a system. In particular, we want to study how requirement engineers and stakeholders communicate and handle imperfection with the jigsaw puzzle metaphor we have developed. Our hypothesis is that this jigsaw puzzle metaphor with imperfect interlocking shapes between pieces (representing concerns with potential conflict) fosters team work and communication towards detecting and analyzing imperfections in requirements and improves commitment of stakeholders in resolution of imperfection.

In the experiments we are going to emulate the "real-life context" of a consultation meeting between requirements engineers and stakeholders. In these meetings with stakeholders the investigators will perform the role of requirements engineers and facilitators, while participants will play the role of stakeholders.

These experiments have a mixed philosophical stance: positivist and constructivist. Thus the experiments will be confirmatory once they will be used to test the hypothesis described above. The experiments will be exploratory once they will be used to understand the capabilities and problems of the proposed metaphor, eventually leading to new hypothesis. When comparing the empirical results with the proposed theory, the conclusions may indicate that the results confirm the theory (as we expect), i.e. the proposed jigsaw puzzle metaphor facilitates and improves communication and work in order to handle imperfection in meetings with stakeholders. It may happen that the results contradict the theory. In this case we would have to reformulate our propositions. As our experiments have also an exploratory nature, the development of a rich case description will be very useful once it will enable to understand the capabilities and problems of the proposed metaphor, which can lead to an improvement of our theory.

We already performed an experiment with results indicating an effective improvement of communication during consultation with stakeholders. We presented a jigsaw puzzle set composed of 4 cardboard pieces with size 10 cm x 10 cm, and representing 4 concerns of the Crisis Management System [7]. Each piece has bumps and dips in one or more edges, which are a cue to how they should be assembled to build the puzzle. The participants were asked to try to build the puzzle. When doing this they discovered that there is a way the pieces fit but not perfectly. At this moment they were asked to read the text in the pieces, and scan what could be the possible sources of imperfection. The text written in a piece representing a concern is the same as in the requirements documentation. To improve readability we cut some not necessary text, made some abbreviations, displayed the text in list mode, and used



upper case letters to stress the “topic” of each requirement. Figure 1 shows a picture of the cardboard puzzle after being used in the experiment and Figure 2 shows the text for Availability and Reliability as it was published [7]. In this experiment participants discovered all the imperfections we were aware and some others we had not thought of a priori. Just to give an example, when looking at Availability and Reliability texts, triggered by the non-fitting interlocking shapes, participants could guess that requiring “max. failure rate: 0,001%” as in Reliability would not be compatible with allowing “downtime for maintenance: 2 h each 30 d” as Availability allows. In fact, if one discounts 24h to the hours a year contains, it is achieved a downtime of 0,274%, higher than 0,001%. However, if we observe from a different angle we can see that the insufficiency may instead inhabit in knowing if “downtime for maintenance” (term used in the Availability requirement) is to be interpreted or not in the same way as “failure” (term used in the Reliability requirement). The participants in the experiment also pointed out this other possible imperfection.

Some of the behaviors and emotions we perceived are now described. When trying to build the puzzle, the participants started to propose to others what tactic to use, like: “make first the corners” and we could perceive they were exploring how to work in group. After this initial phase, participants collaborated as a group, not having problems in posing their questions or making comments, and saying things like: “wait a minute...” and then explaining their reasoning and offering their comments. The participants were handling the pieces, even taking them up of the table, and showing them to others. They used the direction of the text written on the pieces as another visual cue (in addition to the interlocking shapes) that helped to understand how the pieces should go together. During the phase of scanning for imperfections, the participants kept working as a group, and when faced with a possible imperfection, participants discussed among them. After achieving a consensus, one participant hand-wrote the group conclusion on the piece using numbers and/or letters to refer to the different pieces and phrases inside a piece. In some imperfection cases the participants proposed a common remedy for the imperfection. Some of the suggestions for improvements that come out from this experiment are: to have the possibility to identify, univocally, each requirement (each phrase in a piece) so that it can be referred to in comments anywhere in the puzzle; make the pieces bigger so that there is more free space for hand-written comments; and reinforce the jigsaw puzzle metaphor common cues such as the use of color on the surface of the puzzle pieces. We also discussed with participants the question: if it would be preferable to have virtual puzzle pieces instead of the presented cardboard pieces. Participants unanimously preferred the physical cardboard pieces, because of the possibility to handle the pieces, picking them up and showing to others while discussing, and also the possibility of writing on the pieces. But the idea of having (also) the virtual pieces supported, for instance by a digital imaging table is an appealing one. These digital supported pieces could provide instant visual feedback on decisions participants make when facing an imperfection (like changing a requirement text, or adding/deleting a requirement or concern). The exploration of the digitally supported puzzle pieces provides interesting questions for future work. Some of these questions are: how will the virtual puzzle adjust according to decisions participants make, like delete a

concern and thus a puzzle piece? What are the kinds of decisions allowed? How can the aspects participants appreciated in the cardboard pieces be maintained and adapted to the digital media?



**Fig.1.** The cardboard jigsaw puzzle after being used in the first experiment.

- **Availability**
  - The system shall be in operation 24 hours a day, everyday, without break, throughout the year except for a maximum downtime of 2 hours every 30 days for maintenance.
  - The system shall recover in a maximum of 30 seconds upon failure.
  - Maintenance shall be postponed or interrupted if a crisis is imminent without affecting the systems capabilities.
- **Reliability**
  - The system shall not exceed a maximum failure rate of 0.001%.
  - The mobile units shall be able to communicate with other units on the crisis site and the control centre regardless of location, terrain and weather conditions.

**Fig.2.** The text for Availability and Reliability as it is published [7].

We are now preparing a second experiment to explore pieces surface and eventually take profit from the jigsaw puzzle common cue of having an image in background guiding how the pieces should be assembled. The pieces size will be bigger: 12 cm x 12 cm, and the requirements will be labeled for reference in discussion and report. We will perform 3 sessions with 3 different groups using 3 systems, and each group will work as control group for a group in another session. For instance, in the 1<sup>st</sup> session the group will work with system A with jigsaw puzzle metaphor, and with system B with plain text. In the 2<sup>nd</sup> session, the group will work with system B with jigsaw puzzle metaphor, and system C with plain text. The 3<sup>rd</sup> session will work with system C with jigsaw puzzle metaphor, and system A with plain text. With this second experiment we want to confirm our hypothesis further and

explore the usage of piece surface that come out as a suggestion from the first experiment. We are, also developing a software tool to generate the jigsaw puzzle pieces.

### 3 Related Work and Conclusion

There are a number of research areas that have focused on particular aspects of imperfection in information such as conflict management or evolution.

There are also, some methods that provide decision support for specific software activities that are hampered by imperfect information. They extend the expressive capabilities of the development process, through adding models, describing important properties of imperfect information, for instance by means of probability theory or fuzzy logic [21, 10, 15, 13]. There has been an effort to support imperfect information in software development tools, across the life cycle. This is especially true in the work of Noppen [13]. This work focuses on setting up the foundations for the support of imperfect information in software development. Communication and user interaction issues are not the focus of this work, recognizing that the user interaction with the imperfection models (which are mathematical models) can be difficult, in particular since the intended users do not necessarily know them. In fact, it is necessary to study what kinds of imperfect information exist and which ones are relevant for effective decision support and how. Skeels et al work gives a step in this direction in what concerns the uncertainty type of imperfection [16]. Our work aims to explore the potential synergy between Information Visualization and SE Visualization (as envisioned by Gotel et al [6]) to provide good metaphors for the integration and support of imperfect information in software development tools. In fact, visual metaphors have, from a long time, been used to represent information in software engineering. The diagrams are geometric-based metaphors and the tree (even the mathematical concept) is a metaphor. One example of more sophisticated visual metaphors is the city metaphor, which was used to visualize software code [3, 8, 14, 19, 20]. In the city metaphor a building or a district represents an object-oriented class and the visual characteristics are used to depict software characteristics and metrics. Another interesting metaphor is the landscape metaphor where landscapes are used to represent software systems [1]. Boccuzzo [2] propose the usage of the concept of well-shaped graphical visualization to represent that the corresponding artifact is well designed. This shows that the metaphor “language” can be used to express information about the artifacts that are being represented.

We believe the presented jigsaw puzzle metaphor promotes and improves, in a novel way, communication between stakeholders and requirements engineers when working together on requirements, to detect and handle imperfection.

## References

1. Balzer, M., Noack, A., Deussen, O., Lewerentz, C.: Software landscapes: Visualizing the Structure of Large Software Systems. In: VisSym 2004, Symposium on Visualization, pp. 261—266. Eurographics Association (2004)
2. Boccuzzo, S., Gall, H.C.: Cocoviz: Towards cognitive software visualization. In: IEEE Int. Workshop on Visualizing Software for Understanding and Analysis. IEEE CS(2007)
3. Charters, S.M., Knight, C., Thomas, N., Munro, M.: In: Visualisation for informed decision making; from code to components. In: International Conference on Software Engineering and Knowledge Engineering (SEKE '02), pp. 765—772. ACM Press (2002)
4. Diehl, S.: Software Visualization. Springer (2007)
5. Fagan, M.E.: Design and code inspections and process control in the development of programs. IBM Systems Journal 15(3), 182—211 (1976)
6. Gotel, O., Marchese, F., Morris, S.: The Potential Synergy between Inf. Visualization and SE Visualization. In: 12th Int. Conf. Inf. Visualization, pp. 547—552. IEEE CS (2008)
7. Kienzle, J., Guelfi, N., Mustafiz, S.: Crisis Management Systems: A Case Study for Aspect-Oriented Modeling, version 1.0.1, [http://www.cs.mcgill.ca/~joerg/taosd/TAOSD/TAOSD\\_files/AOM\\_Case\\_Study.pdf](http://www.cs.mcgill.ca/~joerg/taosd/TAOSD/TAOSD_files/AOM_Case_Study.pdf). May (2009)
8. Knight, C., Munro, M.C.: Virtual but visible software. In: International Conference on Information Visualisation, pp. 198—205. IEEE Computer Society (2000)
9. Lakoff, G., Johnson, M.: Metaphors We Live By. The University of Chicago Press (1980)
10. Lee, J., Kuo, J., Hsueh, N., Fanjiang, Y.: Trade-off requirement engineering, In: Lee, J.(ed.) Software Engineering with Computational Intelligence, pp. 51—72. Springer (2003)
11. Leveson, N.: Medical Devices: The Therac-25. In: Safeware: System Safety and Computers. Addison-Wesley (1995)
12. National Science Foundation: NSF Workshop on the Science of Design: Software and Software Intensive Systems. In: Sullivan K. (ed.), <http://www.cs.virginia.edu/~sullivan/sdsis/SDSIS%20Preliminary%20Report%20020210.pdf>. (2003)
13. Noppen, J.: Imperfect Information in Software Design Processes. Ph.D. Thesis, Enschede, The Netherlands (2007)
14. Panas, T., Berrigan, R., Grundy, J.: A 3d metaphor for software production visualization. In: International Conference on Information Visualization, p. 314. IEEE CS (2003)
15. Shaw, M.: Truth vs Knowledge: The Difference Between What a Component Does and What We Know It Does. In: 8th Int. Work. on Software Specification and Design (1996)
16. Skeels, M., Lee, B., Smith, G., Robertson, G.: Revealing Uncertainty for Information Visualization. In: AVI, Int. Work. Conf. on Adv. Visual Interfaces. ACM Press (2008)
17. Sommerville, I, Sawyer, P.: Requirements Engineering: A Good Practice Guide. Wiley (1997)
18. Stephenson, A. G., LaPiana, L. S., et al.: Mars Climate Orbiter Mishap Investigation Board Phase I Report, NASA (1999)
19. Wettel, R., Lanza, M.: Program Comprehension Through Software Habitability. In: 15th International Conference on Program Comprehension, pp. 231—240. IEEE CS (2007)
20. Wettel, R., Lanza, M.: Visualizing Software Systems as Cities. In: 4th IEEE Int. Work. on Visualizing Soft. For Understanding and Analysis, pp. 92--99. IEEE CS(2007)
21. Yen, J., Lee, J.: Fuzzy Logic as a Basis for Specifying Imprecise Requirements. In: 2nd IEEE Int. Conference on Fuzzy Systems (FUZZ-IEEE'93), pp. 745—749. IEEE CS (1993)

Research Group	Core Research Topics
<b>Prof. Dr. H. H. Adelsberger</b> Information Systems for Production and Operations Management	E-Learning, Knowledge Management, Skill-Management, Simulation, Artificial Intelligence
<b>Prof. Dr. P. Chamoni</b> MIS and Management Science / Operations Research	Information Systems and Operations Research, Business Intelligence, Data Warehousing
<b>Prof. Dr. F.-D. Dorloff</b> Procurement, Logistics and Information Management	E-Business, E-Procurement, E-Government
<b>Prof. Dr. K. Echtele</b> Dependability of Computing Systems	Dependability of Computing Systems
<b>Prof. Dr. S. Eicker</b> Information Systems and Software Engineering	Process Models, Software-Architectures
<b>Prof. Dr. U. Frank</b> Information Systems and Enterprise Modelling	Enterprise Modelling, Enterprise Application Integration, IT Management, Knowledge Management
<b>Prof. Dr. M. Goedicke</b> Specification of Software Systems	Distributed Systems, Software Components, CSCW
<b>Prof. Dr. V. Gruhn</b> Software Engineering	Design of Software Processes, Software Architecture, Usability, Mobile Applications, Component-based and Generative Software Development
<b>Prof. Dr. T. Kollmann</b> E-Business and E-Entrepreneurship	E-Business and Information Management, E-Entrepreneurship/E-Venture, Virtual Marketplaces and Mobile Commerce, Online-Marketing
<b>Prof. Dr. B. Müller-Clostermann</b> Systems Modelling	Performance Evaluation of Computer and Communication Systems, Modelling and Simulation
<b>Prof. Dr. K. Pohl</b> Software Systems Engineering	Requirements Engineering, Software Quality Assurance, Software-Architectures, Evaluation of COTS/Open Source-Components
<b>Prof. Dr.-Ing. E. Rathgeb</b> Computer Networking Technology	Computer Networking Technology
<b>Prof. Dr. E. Rukzio</b> Mobile Mensch Computer Interaktion mit Software Services	Novel Interaction Technologies, Personal Projectors, Pervasive User Interfaces, Ubiquitous Computing
<b>Prof. Dr. A. Schmidt</b> Pervasive Computing	Pervasive Computing, Uniquitous Computing, Automotive User Interfaces, Novel Interaction Technologies, Context-Aware Computing
<b>Prof. Dr. R. Unland</b> Data Management Systems and Knowledge Representation	Data Management, Artificial Intelligence, Software Engineering, Internet Based Teaching
<b>Prof. Dr. S. Zelewski</b> Institute of Production and Industrial Information Management	Industrial Business Processes, Innovation Management, Information Management, Economic Analyses