

# 1 Title

## 1.1

Human Requirements Engineering – The reason people are different and why we fail to tap their full potential

## 2 Abstract

How many problems in your project are caused by inappropriate interaction with the people involved? People are different – we think everybody can agree with this statement. Nevertheless, this insight is easily forgotten in our day-to-day work. Due to the importance of communication and agreement in Requirements Engineering, people and their individual characteristics play a very important role in it. Thus it is particularly crucial in the area Requirements Engineering that we focus on the people involved and their personalities.

This article investigates the reoccurring problems in different disciplines of Requirements Engineering and relates these problems to the people involved in the process. By approaching the issues in this way, the roots of many problems and possible solutions may be found without a need to establish new methods and processes.

The goal of this article is to shift our focus onto different personalities - and especially to the strengths and fears - of the people involved in Requirements Engineering in our everyday work.

### 2.1 *(Footnote) Note on Gender Neutrality*

For clarity's sake, the use of the male gender is used throughout the text, even though both genders are being addressed.

## 3 Main Part

Your development projects rise and fall with the people involved in them. People are the most important component of your project. This insight is not new [10]. Thus it is all the more astonishing to observe time and again how project participants are dealt with. It is also astonishing that in many companies no real efforts are being made to understand each individual person, to try to analyze their needs, wishes and fears, and to do everything possible to create a working environment in which each person can reach their highest potential. "In order to motivate [people] to change their behavior, you need to understand where they come from and why they are as they are" [8]. Successful projects place great value on the people involved in the project.

What is valid for a successful project in general is valid for successful Requirements Engineering in particular. The goal of Requirements Engineering is to comprehensively define the product to be developed. All people who contribute to this definition are the stakeholders of the product [6]. As a rule, the number of these stakeholders is very high, encompassing at least the customer, the management, the marketing, the end user and the development team. Furthermore, there are many other potential stakeholder groups. All these people have different professional backgrounds, different experiences, and different interests in the product. They also probably have different religious and cultural backgrounds, and are different in

gender and age. The list of differences between the people involved in Requirements Engineering is almost endless.

The task of Requirements Engineering is to bring all these different people together and to cast their perceptions, ideas and wishes into the form of a requirements specification. In order to achieve this, it is highly important that the people involved communicate and understand each other. The difficult task of bringing these people together falls to you as the requirements engineer.

It is clear that to achieve this task various methods such as use-case analyses, process models such as the Rational Unified Process®, quality models such as CMMI®, and tools such as DOORS® are only of limited assistance. All of these are of little use to you if you do not place the people and their characters at the forefront of your day-to-day work. The successful requirements engineer must understand his stakeholders and know how to deal with different personalities.

Take a look at the following scenario in which people with very different types of personality come together.

### Starting Point



We are currently at a project kickoff meeting. Among those present are a customer representative, a marketing expert, the project manager, a technical expert and you in your role as requirements engineer.

During the kickoff meeting, the most important requirements placed on the product to be created should be defined. In addition, decisions on the go/no-go of the project need to be made. During the course of the meeting, you notice that complications tend to arise at different times between the people involved:

- When it comes to the question of whether certain features are feasible or not, there is a difference of opinion between the marketing expert and the technical expert. While the marketing expert sees no problems and is of the opinion that everything would be absolutely feasible, the developer sees that the implementation of the desired features holds considerable risks. When questioning the developer about his opinion, however, he merely refers to his many years of experience or speaks of technical implications which are difficult to comprehend for the others.
- The customer continually insists on analyzing the requirements in a very detailed way and recording them in writing, while the technical expert would rather focus generally on the use cases. The expert's argument is that he

wishes to understand the entire picture and does not wish to get tangled up in unimportant details. The customer, by contrast, would like to have the opportunity to have a good system for monitoring the progress of the project by tracking which features have been implemented.

- The project manager continuously strives to find a compromise for all parties involved and to advance the meeting. Yet in trying to make everyone happy, he does not get anywhere, as he is attempting to avoid confrontation.

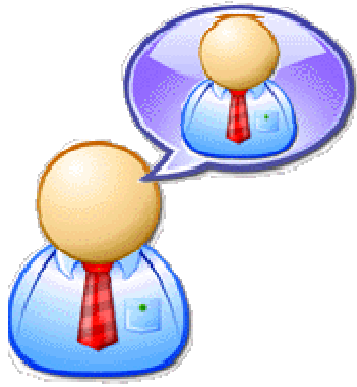
It soon becomes clear that you may only determine the necessary requirements placed on the product to be developed by understanding the different people and the motives behind their actions. One possible procedure to achieve this is the Temperament Sorter II according to David Keirse (see section 4).

The scenario described above shows in a striking way the interaction between completely different personalities during an example requirements process. The key to successful Requirements Engineering lies in understanding the people involved in the requirements process. To do this, you can ask yourself the following questions as the requirements engineer: What kind of person am I currently dealing with? Why is he reacting in the way that he does? How can I best handle this person in this situation so that I achieve my objectives as requirements engineer? And the following question in particular needs to be addressed by you: How should I not behave in order to avoid endangering the achievement of my goals?

Some typical problematic scenarios from different disciplines of Requirements Engineering are outlined in the following. The strengths and weaknesses of the participants in these scenarios are based on a small number of different personality types. As the behavior of people is often due to different factors, considerably more information would be necessary for a definite type allocation. In order to make the scenarios clear, the type which best suits the person described has been selected. By means of the personality type, it is possible for us to derive the background and reasons for the type of behavior, enabling possible solutions and "Don't Do's" to be revealed.

The personality of the requirements engineer will not be taken into account in this article. However, it is clear that your personality type in your role as requirements engineer influences the scenarios to the same extent as that of the person you are dealing with. And although you cannot change your personality, the art of being a good requirements engineer lies in the ability to adapt your behavior appropriately to different situations in order to achieve the best possible results with the people involved.

### **Scenario: Requirements Elicitation – Incomprehensible Requirements**



Take a look at the following scenario: When eliciting requirements, you would like to determine the requirements of a particular stakeholder, e.g. a representative of the end user. This person is known for his very detailed and in-depth knowledge and, from experience, he is also able to understand complex problems. You deploy a number of different techniques in order to draw the requirements out of him. These techniques include brainstorming, interviews and questionnaires, or the use of prototypes [6]. At the end of this process, it has not been possible to tease out any requirements from the stakeholder that may be generally comprehended by everyone. The requirements given are either too abstract or they falsely assume that previous knowledge exists. Often the requirements that are established contain very specific and difficult specialist terms. It is obvious that the stakeholder has a relatively good idea of how the product to be developed should behave, but he is not able to convey these requirements in a comprehensible manner.

If one looks at this scenario from the view of the types according to Keirse, an in-depth understanding of the system combined with the difficulty in communicating this knowledge indicate that the stakeholder is a "Field Marshal" (ENTJ, see box). According to the explanations in [3], the reason for such behavior lies in the fact that the stakeholder is not adept at understanding the needs of other people. Thus he does not easily recognize how the requirements should be described and explained so that those with a different way of thinking can understand them. He sees his ideas as the only correct ones and expects the others to implement them accordingly. In a sense, he lives in his own world and expects that his world-view will be assumed by the other people in his environment. If that does not happen, misunderstandings and problems result.

Now that we understand the stakeholder better it remains to be seen how to obtain clear, correct and comprehensible requirements with him.

- Probably the most important point is the use of a glossary in which the complex specialist terms are explained. On the one hand, this compels the stakeholder to think about his world-view and to convey this to his environment. On the other hand, it enables the users of the requirements to understand the terminology used by the stakeholder.
- Have a second person accompany the stakeholder. This second person should have the stakeholder's understanding of the specialist area and the associated terminology, but must also be able to translate this so that it is comprehensible to everyone.
- Use requirement templates. These allow the stakeholder less leeway for formulating incomprehensible sentences.

- Ask your stakeholder how a test for the requirement he has just formulated might look like. This question may help in getting the stakeholder to provide concrete criteria for his requirements.
- Ask closed questions as far as possible, i.e. questions that only allow the stakeholder to answer with “yes” or “no”. This prevents a burgeoning understanding of the requirement on your part being destroyed by new and incomprehensible descriptions.
- Allow your stakeholder to outline a use case. This serves to make the requirement more concrete.

Above all, you should observe the following if you wish to successfully obtain requirements from this person:

- Deal with the contents of what is said, never how they are presented. If the stakeholder knew how to convey his ideas better, he would probably do it.
- Never reveal your lack of comprehension regarding the requirements that have been determined, even if they are difficult to understand. By doing so, you will only end up frustrating your stakeholder and may possibly ruin your chances of gaining good and important requirements.
- Do not simply reformulate the same questions again. The answers which you receive will rarely create more clarity. Rather, the new statements will only cause further confusion.

### **Scenario: Requirements Elicitation – Solution-Oriented Requirements**



It is your job to find the essential requirements of the product to be developed [6]. This is necessary in order to ensure that innovations are not quashed, and in order to generate a reusable specification.

Take a look at the following scenario: When eliciting requirements, you would like to determine the requirements of a particular stakeholder, e.g. a technical expert on the customer side. This person is known to be highly organized, reliable, dedicated and does not shy away from making necessary decisions. Instead of providing you with genuine requirements, however, this person merely gives technical solutions for implementing the unknown requirements behind them, e.g. “The user interface should have a dropdown menu from which the user can select a menu item” instead of “The product should offer the user the possibility of choosing between different options”. The stakeholder thus thinks and acts in a solution-oriented way.

The combination of an organized, reliable way of working and the solution-orientation in this scenario points to the “Supervisor” type (ESTJ, see section 4).

In this case, the reasons for the behavior of the expert lie in the fact that he already has a very clear idea of how the product should look at the end and how it should be created. What he does not see, however, is the significance of goals which lie beyond his level of thinking. With this clear image of the end product in his mind it is

thus difficult for him to take into account such aspects as reusability or transferability of requirements. He can clearly visualize the product. He is not aware, however, of the consequences of this solution-oriented way of thinking. This expert expects the product to be implemented precisely according to his ideas. He cannot understand that it is beneficial to keep the requirements free of solutions in order to get to the actual problem and to keep alternative solutions open.

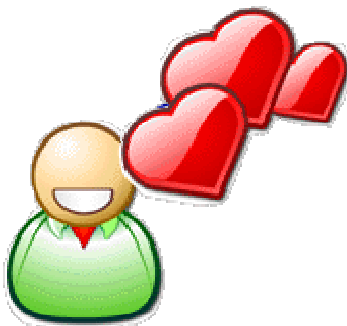
In order to obtain requirements from this person that meet your demands for problem-oriented and reusable requirements, you may undertake the following measures based on an understanding of this character:

- Ask the stakeholder to list the requirements of similar products, too. Although he will again describe a solution, this will allow you to extract the essence and thus the actual requirements from these different solutions.
- Ask the stakeholder for contingency plans should problems arise when implementing his proposed solutions. As he is the type to plan, he will gladly accept this request and will thereby reveal alternative technical solutions and possibilities which you would otherwise not have obtained from him. Here, too, you can extract the essential requirements from the various solutions.
- You can also deploy those people who tend to think in a solution-oriented way to produce a feasibility study, generate prototypes or provide evaluations on the efforts involved in cases involving a great technical risk. This is because solutions are in the foreground when there are great technical risks!

Try to avoid the following, however:

- Discussing the proposed solutions when trying to elicit requirements. This does not bring you nearer to your goal and can be very time-consuming.
- Continually demanding that the discussion be free of solutions. You will probably only meet with incomprehension and receive objections such as: “But it is important how it is going to be implemented!” or “If we are going to do it like that anyway, we might just as well take it up into the specification!”.
- Incorporating his requirements into your specification without checking them, even if working on them costs you time and effort.

### **Scenario: Change Management – Uncritical Acceptance of Change Proposals**



Let's take a look at the following typical scenario. In order to be able to implement the changes to the requirements desired by the customer and to incorporate them in a controlled manner into the project, you have established a Change Management system. Within this framework, the Change Control Board (CCB) meets regularly.

This is made up of a circle of people who are in a position to evaluate the changes desired by the customer, and who either reject or accept these.

Here you find out that one of the people involved in evaluating the desired changes (e.g. the developer involved) regularly deems such changes to be feasible and easy to implement when, in fact, they often turn out to be time-consuming and costly. This person does this particularly when the customer signals that the change is very important to them. Although this person is very popular with the customer due to his accommodating and well-meaning behavior, the project loses time and credibility due to this.

The inappropriately cooperative behavior of the person in question in this situation points to the Keirsey “Protector” type (ISFJ, see section 4).

A “Protector” puts the needs of other people above his own, which leads in this example to change requests being accepted in order to spare the person who made them any problems. Instead, this person would rather burden himself with the problems of implementing these changes. This type of person always tries to be friendly and considerate, which is advantageous in most cases. When complex or problematic changes desired by the customer need to be rejected, however, a more egoistic way of thinking would be more beneficial. “Protectors” feel very uncomfortable in conflict situations and always try to avoid them. Through uncritical acceptance, this person is able to circumvent confrontations with the person who has demanded the change. Although he is usually aware that this behavior may cause problems for him and the project in the long-run, this nevertheless seems to him to be the lesser of two evils.

In order to carry out the project according to plan and in a controlled manner, you need realistic evaluations with respect to the feasibility of demands and changes and to the efforts involved in implementing them. Here, you are usually dependent on the input of certain people. In order to arrive at realistic and honest evaluations with these people in the CCB when it comes to changes to requirements desired by the customer, you can pursue the following strategy:

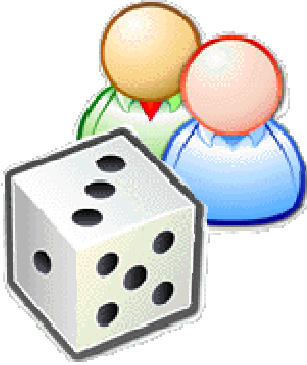
- Make clear to them that a correct evaluation will be esteemed considerably more by the project management and the customer in the long-run than a good-natured one. Even saying “no” to a request for change will be seen as an act of sincerity.
- If an honest evaluation nevertheless leads to conflicts with the customer, make clear that it is you, in your role as requirements engineer, who will carry the full responsibility and who will deal with the conflict in his place.
- Make the enquiry to this person via email and go into the CCB yourself with his input. Without time and personal pressure, the evaluations of a “Protector” are usually better.
- Sometimes give careful negative feedback in cases where evaluations are rather too optimistic.

If you wish to arrive at good results with this person in future, you should avoid the following behavior:

- Putting this person under pressure. He will bend to this pressure and give you the answer he thinks you want to hear.
- Reacting in a negative way to pessimistic evaluations. If you do, you will never again obtain an honest evaluation in future.

- Support this person in situations in which he is subjected to great pressure, e.g. during customer visits.

### **Scenario: Risk Management – Not Accepting Risks**



A Risk Management system is established in your project in order to continually recognize new risks, to keep an eye on known risks, and to deal with risks that have become problems [7]. Good Risk Management involves creating a project culture which, on the one hand, encourages the project participants to name risks of their own accord, and which allows you to ask the project participants to name risks when planning work packages, such as the implementation of requirements.

Let's take a look at the following, frequently occurring scenario: You ask a project member for his opinion on possible risks when implementing some requirements and you receive the answer that there are no risks and that it is certain that no problems will arise. Upon asking again, you are told not to worry so much and that "we will manage!" During implementation, it then becomes apparent that greater efforts are involved than originally planned. The project member who stated there were no risks had to work overtime and did his best to observe the deadline. The increased effort is obviously the direct result of an all too optimistic evaluation at the outset.

The Keirse type which best fits the hands-on, confident project member who likes to ignore risks is the "Promoter" (ESTP, see section 4).

"Promoters" are downright attracted by challenges and risks. If they come up against limitations, they want to transcend these. While the statement "You won't manage that!" discourages other types, this character will be spurred on even more to achieve it. If such a project member is faced with a task, he will strive to implement it and is seemingly not prepared or able to plan carefully or to follow the plans of others. "Promoters" tend to overestimate themselves and their capabilities as well as those of the team in which they are acting, and to ignore potential problems. This may lead at some point to reality catching up with them, forcing them to recognize that the implementation of requirements actually demands greater efforts, takes longer and costs more than planned. And it is the project which suffers as a result.

After analyzing this person's character, it is possible for you to incorporate him in a profitable way into your Requirements Engineering process. To do this, ensure that you observe the following:

- When enquiring about risks, demand concrete and verifiable reasons for his optimistic evaluation. This procedure will allow you to better assess the reasons for his evaluation.

- Always calculate a potential risk into the evaluations given by this person. In so doing, you should also gather the input of more reliable team members.
- Continue to point out that the project goals may only be reached in the timeframe and budget if all risks are known.
- Give this person some freedom, as such types of people tend to function as catalysts and motivators within a group. Particularly in seemingly hopeless situations, these people are often able to guide the project to a successful end.
- Give such people extremely time-critical tasks where a systematic and methodical procedure is not possible.

If you wish to receive honest statements on potential risks from people of this type, then:

- Do not present problems as challenges. This leads to the person seeing only challenges, but no longer any risks.
- Do not point to the fact that certain requirements need to be implemented quickly. This will only lead to the person believing that the requirement really doesn't hold any risks.

### **Scenario: Quality Management – Refusal to be Monitored**



Within the framework of the general Quality Management system, you have organized for reviews to be carried out at regular intervals for various documents within the project. These include reviews of the requirements specification and the working results derived from this, as well as the design specification, the source code and the test cases [9].

Does the following scenario sound familiar to you? Due to deadline pressure you would like to monitor the progress of the project during the design specification stage more stringently. As collaboration between the various departments is crucial, you are forced to bring forward the deadline for the planned reviews. You are having difficulties, however, in getting the author of the specifications to provide reviews at the intermediate stages. He tries to wriggle his way out of the task by arguing that the document is not yet in a perusable state, or by bluntly stating that he does not see the sense in the continual reviews. If you nevertheless succeed in carrying out a review, the author is often very uncooperative by stating that it is not yet finished. He also feels personally attacked by criticism. This behavior is astounding to you, as the author is usually very cooperative. He is also known for delivering high-quality results.

The combination of the basically cooperative nature of the author with a clear rejection of criticism and monitoring of his way of working points to the “Champion” type (ENFP, see section 4).

The author would like to carry out his work independently and to be answerable only to himself. Any monitoring or criticism of his way of working by other project participants leads to him refusing to talk about his work. He tends to push the responsibility for any problems that arise onto other people by using seemingly logical arguments. If he feels put under pressure by continually being asked for a review, there is a risk that he will build up a prejudice against you. This may lead to your arguments being rejected, irrespective of whether they are right, or not. If the “Champion” is placed under even greater pressure, this may even lead to destructive behavior, e.g. review comments may not be appropriately implemented.

The following behavior on your part will allow you to carry out successful reviews with this person under the conditions described:

- Present the review as a platform for passing on knowledge to others. This statement is initially correct and it flatters the author of the document to be checked, as it puts him in the position of an expert who is able to impart his knowledge to others.
- Ensure that the author is informed of the review deadlines in good time. If necessary, allow him to make a suggestion for the deadline himself. This will give him the opportunity to complete topics that he has started.
- Convey to him the reasons why this review should be carried out! If he understands the reasons behind the review, he will be more cooperative.
- Brief the other review participants prior to the actual review and in the absence of the author by telling them to focus only on the main points and to make only objective remarks.
- If somebody nevertheless strikes the wrong note during the review, reprimand this person. Act as a protective shield for the author, warding off any inappropriate comments.
- Following the review, ask the author in confidence to appropriately annotate those comments that should not be implemented in the reviewed document with good reason, and to incorporate the other comments into the document.

If you wish to carry out fruitful reviews in future with a person revealing the character traits outlined, be sure to observe the following:

- Do not exert too much pressure on this person by asking for a review deadline too often or by continually enquiring about the work status.
- Even if you do not completely trust this person to incorporate all the review results into the document, only check the document and the annotated review protocol after a previously established time, and not the working method of the author.
- Never personally criticize the person or make biased criticism of the reviewed document.

## **Summary**

Did you recognize yourself in any of the scenarios described? Are the suggested solutions plausible? Are you looking for solutions to similar problems? Unfortunately, there are no standard answers to dealing with people in the correct way – we have to disappoint you there. Such standard answers are mostly so general that they are of little use, as they do not address concrete situations and the people involved.

By looking at people in the way we have described, it is nevertheless possible to recognize and understand the motives for a person’s behavior that are rooted in their

personality. On the one hand, this knowledge makes it easier for us to recognize and choose the correct way of acting. On the other, it prevents us from intensifying the problem in dealing with a person by approaching him in the wrong way.

You should take care, however, not to make too quick a judgment about the type of person you are dealing with, and should not carve this judgment in stone. Frequently, similar reactions on the part of different types can have completely different reasons. It is thus important to develop a keen eye in order to find the true motives behind people's behavior. These considerations are particularly important in Requirements Engineering due to the crucial role that communication and understanding play in it.

## 4 Personality Types and Kersey's Temperament Sorter II

The idea behind categorizing people into different types is that while people do not all act out of exactly the same motives, they do not act out of completely different ones either. Classes of people may be found in which those who think and act in a similar way may be grouped together. This classification reflects behavioral tendencies, which helps in understanding the goals, motives and backgrounds of the particular actions in each case.

In the 20th Century, C.G. Jung proposed the idea of personality types. This idea was taken up in the middle of the last century by Isabel Myers and Kathryn Briggs, who developed a questionnaire named "The Myer-Briggs Type Indicator" (MBTI) for classifying types. In [1], David Keirsey presents his Temperament Sorter II for these types. The German version of this may be obtained free-of-charge under [2]. For more detailed information, we would like to refer you to the referenced literature. In [4], you will find more in-depth considerations regarding the differences between people and the effects these have on people's lives and careers.

According to the methods named above, people may be classified using four opposing pairs of characteristic traits:

- **Extraverted** (expressive) vs. **Introverted** (reserved) (related to social behavior)
- **Sensory** (observant) vs. **INtuitive** (introspective) (related to the perception of the world)
- **Thinking** (tough-minded) vs. **Feeling** (friendly) (related to decision making)
- **Judging** (scheduling) vs. **Perceiving** (probing) (related to way of life)

Thus there are 16 different types, which are each made up of 4 traits. The scenarios outlined above are based on these types.

From these individual types, Keirsey forms groups which each have relatively many traits in common. In order to obtain a fundamental understanding of the differences between the types, we have provided a brief overview of these groups in the following.

### **Artisans, SensoryPerceiving (approx. 3/8 of people)**

Artisans are people who love creating and achieving things. They live in the here and now. They see chances and seize them immediately. They do not like to be held in check by rules. They act very pragmatically and in a solution-oriented way. Problems and limitations represent challenges for them and, if possible, they are able to act quickly and decisively. Artisans are very service-oriented and are well-suited to positions where something needs to be generated, achieved or presented. An example of an Artisan is the marketing expert described in the first scenario. Another

breed of the Artisan is the **Promoter** (ESFP) described in the Risk Management scenario.

### **Guardians, SensoryJudging (approx. 3/8 of people)**

Guardians are responsible for bringing order into the world. Anything that involves analyzing and monitoring tasks, or the upkeep of what is already in place, lies in their interest. From a career and social perspective, they are interested in continually moving up, and are thus very conservative and hierarchical in their thinking. Guardians are usually reliable, punctual and exact. It is thus not really astounding that Guardians make up the greatest proportion of managers [5]. Otherwise, Guardians are well-suited to fields which promise security (e.g. teachers or officials). The need to plan and to monitor is seen in the behavior of the customer in the scenario described at the start of the article. Sub-types of the Guardian include the **Protector** (ISFJ) presented in the Change Management scenario and the **Supervisor** (ESTJ) described in the second requirements elicitation scenario.

### **Rationals, iNtuitiveThinking (approx. 1/8 of people)**

Rationals are the (master-) minds of humankind. They know how to analyze and see through systems. Their perspective is directed at the big picture and tends to be long-term in nature. It is guided by the urge to gain knowledge and understanding. Routine tasks and details are often seen as onerous. In order for their abilities to really unfold, they need adequate freedom and leeway for their actions. Rationals feel at home mainly in those areas where knowledge and research are in the foreground, such as at universities. How these traits express themselves in concrete project situations is shown by the example of the developer in the first scenario. A representative of the Rational type is also presented in the form of the **Field Marshal** (ENTJ) in the first requirement elicitation scenario.

### **Idealists, iNtuitiveFeeling (1/8 of people)**

Idealists are constantly searching for the meaning behind things. They are diplomatic, have empathy, are loyal, and they attach great value to personal relationships. Their strengths thus lie in dealing with people. Idealists think in an abstract way and are future-oriented. In contrast to Rationals, they look for solutions that are rather more on the creative and fanciful side than for optimum ones. Idealists are well suited to those fields in which interpersonal relationships play an important role. Typical examples include teachers, psychologists or consultants. The actions of the project manager in the first scenario correspond to the behavior of this type, for example. The **Champion** (ENFP) presented in the Quality Management scenario also belongs to idealists.

## **5 Literature**

- [1] *D.Keirse*: Please Understand me II, Prometheus Nemesis, 1998
- [2] [www.keirse.com/german.html](http://www.keirse.com/german.html)
- [3] [www.personalitypage.com](http://www.personalitypage.com)
- [4] *G.Dueck*: Wild Duck, Springer, 2004
- [5] *O.Kroeger+*: Type Talk at Work, Delta, 2002

- [6] *S.Robertson, J.Robertson: Mastering the Requirements Process*, Addison-Wesley, 1999
- [7] *T.de Marco, T. Lister: Walzing with Bears*, B&T, 2003
- [8] *T.de Marco: Deadline*, B&T, 1997
- [9] *A.Spillner, T.Linz: Software Testing Foundations*, Rocky Nook, 2006
- [10] *B.Hindl, J.Schmied, K.Hörmann, M.Müller: Basiswissen Software Projektmanagement*, 2. Auflage dpunkt.verlag, 2006

## 6 Curriculum Vitae

### 6.1 Paul-Roux Wentzel



Paul-Roux Wentzel studied computer science with philosophy as a minor subject at the University of Erlangen-Nuremberg. He also worked there as a trainer in the area of theoretical computer science and as an honorary tutor.

He has been working at method park Software AG since 2002 and has gained a great deal of experience in development and testing from several projects for customers of the method park Software AG in the embedded environment. He has been a consultant since 2005, focusing in particular on Requirements Engineering.

The project sizes have varied between 5 and 100 people; the application areas have spanned everything from simple microchips to complex control devices and infotainment systems in the automotive field.

Address: method park Software AG, Wetterkreuz 19a, 91058 Erlangen, Tel: +49 (0)9131 97206 0, Fax +49 (0)9131 97206 200, [www.methodpark.de](http://www.methodpark.de)

### 6.2 Otmar Seckinger



Otmar Seckinger studied media and computing at the University of Applied Sciences in Furtwangen, and Information Technology at the Griffith University Brisbane, Australia. Privately, he is active on the Board of the MinD Hochschul Netzwerk (Mensa in Germany college network).

He has been working at method park Software AG since 2004. Since 2005, he has been a consultant in the area of Requirements Engineering, Architecture and UML. The focus of his consulting lies in personal coaching. The customers he deals with come mainly from the embedded environment, particularly from the automotive area.

Address: see above.

### 6.3 *Bernd Hindel*



Prof. Hindel studied computer science in Erlangen and Green Bay (USA). After gaining his doctorate at the Chair for Programming Languages at the University of Erlangen (1991), he worked for the Central Research and Development Department at Siemens AG in Erlangen. From 1995 to 2001, Prof. Hindel was a member of the Executive Board at a medium-sized software company. In 2001 he founded the method park Software AG.

Prof. Hindel is a lecturer for software engineering at the University of Erlangen-Nuremberg and at the Volkswagen Car University (Volkswagen-Autouniversität). At the start of 2004, he was made scientific director of the International Software Quality Institute "iSQI", located in Erlangen.

As CEO of the method park Software AG, he received the TOP JOB award for being the top employer among medium-sized companies. In July 2005, the IHK (Chamber of Industry and Commerce) founder prize followed for innovative personnel management and product ideas, as well as international commitment.

Address: see above.