SAP Inside Track Hamburg 2014

Welcome

My name is Enno Wulff and I'd like to welcome you to this SAP Inside Track in Hamburg. I would like to thank Peter Langner who has invited me to come and give a paper of my free choice. I had to think some time about the opportunity that Peter had offered me, because at first I doubted if there was something useful I could tell you but nevertheless myself esteem deemed it necessary to be here and offer you some of my knowledge and ideas that hopefully will be of interest to you.

Most of you know me because of my website *Tricktresor.de* where I present SAP programming related problems and solutions to German speaking audiences. The Tricktresor is very popular and many people like the articles I share but is there something I could TALK about? All articles are about small programming problems and all I have to I do is to write them down, Including my solution to that problem. Is that something special?

My experience tells me: Yes it is! So let me talk about INFORMATION and DOCUMENTATION. About problems and solutions. Let me tell you something about how to spread information and how to preserve and remember it.

Vita

Let me tell you a short history on how I came to inventing the Tricktresor and why I think I can tell you something about information and documentation.

At the end of my education as an IT Business Engineer I was working in an SAP team with Karl. Karl was a very professional and experienced consultant. He had a very slow and thoughtful Austrian behaviour and he wrote down everything he did and experienced at the customer. If I said everything, I mean **everything**! In meetings, he would write memos on his laptop. After implementing a function, he made a note. After having changed a program or Dynpro he would then document it.

He would use cryptic acronyms: # (hash key) plus initials for persons he talked to. $ (dollar sign) plus date for time stamp and & (ampersand) plus other key words for function modules, forms, programs, transactions and so on. I smiled at this strange attitude and I thought: He is an old man... I have a young and flexible brain. I can **remember** these silly things that he **writes** down. I kept smiling for exactly two weeks. After two weeks, I had to go to Karl and asked him the name of a program I had to use. Karl took a look at his text file and searched for &RP and found the right program.

From this time on, I also started to writing down things I had done, reports I had used, tricks I had learned. Sometime later on as new things were getting less, I wrote down less. But when I wanted to write down something I had to start my black & white laptop and waited for windows to come up. This was annoying and I wanted to have the information on the internet.

I didn't mind others reading my tips and tricks. But being aware that others would read my information I wanted to present the information in the most useful way. That meant I wanted to describe the problem I had and write down the solution. If possible, I wanted to use a simple small program that exactly represented the problem and the solution and nothing else.

That was about 15 years ago. At the time, the only information available about SAP programming was SAP Help itself, a bunch of undocumented batch-input-programs and the website of Imre Kabai (kabai.com). The website still exists!

My intention and interest was to create a website that not only held information but also described everything that belonged to the problem. I realized though, that this was not that easy! Writing down the solution is easy, but describing the problem is very hard!

What is information?

Information is at least a fact. Words. Hand-tight data. But normally this is not enough. What is missing? Mostly the context. When searching for information the challenge is to do so within the right context.

Information is – let's call it – multi-dimensional. The exact point of view, the context, the surrounding are very important. If you change the context, then information will be misleading and perhaps useless.

Misleading Information

I would like to introduce to you an effect that is very hard to identify: The pseudo information. "Pseudo" originates from the Greek language and means "not true". I am no information scientist or anything like that so I don't know if there is another – a “real” – word for what I mean. A pseudo information is a fact that seems to be true but is wrong. Pseudo Information might result from unproved information, expectations or even incorrect information.

To explain what I have in mind, let me tell you a story. Last month I was jogging in the local forest – (okay, I was not, I just told you that to make it more authentic!) and I heard someone shouting in a deep loud voice: HEKTOOOOR! Come here! Immediately I had the impression of a tattooed muscle-bound rocker with his pit-bull. This is an example of pseudo information because neither did I see the person shouting nor did I see the pit-bull. What I **expected to be** a pit-bull does not mean that there **was** a pit-bull.

Similar things happen with computer problems. If there is a short dump new_page_allocation_failed and you expect a basis problem it does not mean that it is a problem of your SAP administrator! In most cases the reason for excess memory consumption is located in wrong or inefficient programming.

Incorrect information

If someone tells you that the reason for a problem is the legacy system which cannot deliver certain information then you are in a similar “information trap”. You hear something and you cannot prove the validity of the information. Of course you cannot prove every information you get. But you could ask: “Did you find the error in the legacy system or do you suspect the error to be there because you didn’t find the error in the SAP-System?” Whether you believe this or not, depends on the status and authenticity of the person who gives you the information.

Confusion

Languages are always a good source for misunderstandings. Native speakers, idioms,” False friends,” special vocabulary and so on make it very hard for non-native speakers to communicate. Even more if business language in a German company is English. If you are not really really good in the foreign language than you will have to translate everything from German to English. Someone else will have to translate back to German. So if you want to create confusion, force your employees to communicate in English. Topics, which are already hard to understand in one’s own language, will be even harder in foreign languages.

In most cases you will know roughly what is meant although a wrong phrase has been used or if some words have been mixed up, but you don't know if every person involved thinks the same…

Right or wrong

All kinds of wrong information are critical because they don't show the core problem; they lead in the wrong direction. Of course this is quite clear but it isn't obvious if your information is right or wrong.

That means that you should not only search for solutions but firstly ensure that the information you have is very clear and accurate.

How often have you experienced something like this: "The saving of material takes a long time. Can you please check if there is something wrong!?" After having searched for hours in the material master application without any result you hear: "Oh sorry. I meant saving the materials in the sales order…"

Additional information

My wife always says: Talk to me, it will help! And she is right. Because she is my wife. If you give information, try to give a bit more of information than is exactly needed. If you give someone a bit more information you are doing two things: In first place you offer a little more help. Secondly you show the asking person that you are willing to help some more and that your colleague is welcome. Do not only serve the beer but also some peanuts.

The best example I ever heard of on this statement has been the doctor's receptionist who called a patient to postpone the appointment because the doctor extended his vacation. The woman was very angry because the date was very important to her and the doctor is now sitting a week longer in the sun while she is having sorrows. If the receptionist had told the woman that the doctor has been helping people in a crisis region with heavy diseases and he has now decided to stay a week longer, then the woman would have accepted this fact and she would have been glad to be treated by such a respectable doctor.

Outdated Information

Information is very fast nowadays. And information is outdated very quickly. Things that are valid for one Release must not be true with the next Release. If you set up rules upon a specific fact you must periodically check if the precondition still is true.

What is important for remembering and handling information?

There are some things that are important for learning, for presenting, documenting and remembering.

I will try to tell you some interesting facts.

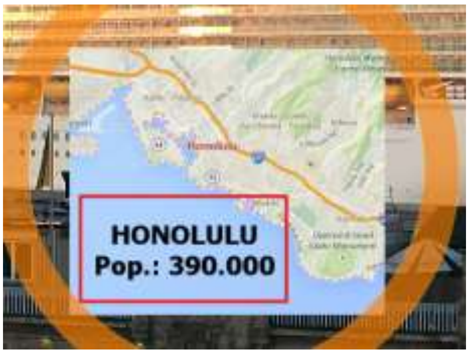
Emotions

Information is also about emotions. You remember better if you are feeling happy or excited at the moment. If you are stressed or unhappy or if you are concerned about something it will be hard for you to concentrate, to learn, to remember facts.

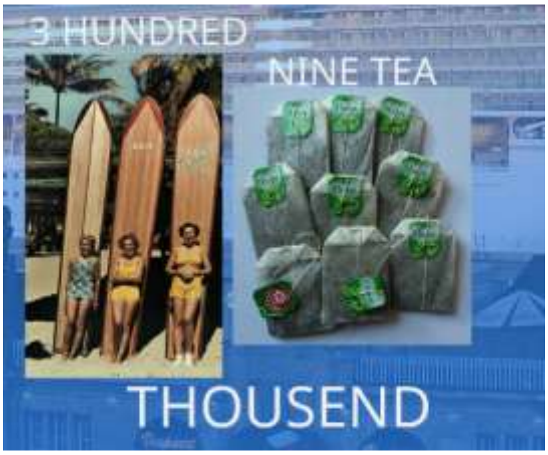
I liked the idea of the SAP Train Race: 2 Countries, 72 hours, 50 developers. Pure emotions. Highest concentration. Fun. Colleagueship. Unique experience.

Weirdness

You will remember best if the information itself or the context or how the information has been presented is somehow strange, weird or funny. If I presented to you the population of Honolulu like this, you might soon forget it.



If I presented it in another way you might not forget it for a long time or you might remember it forever...? Because of this picture you will not have to remember a simple fact. You can reproduce the fact by the pictures you have seen.

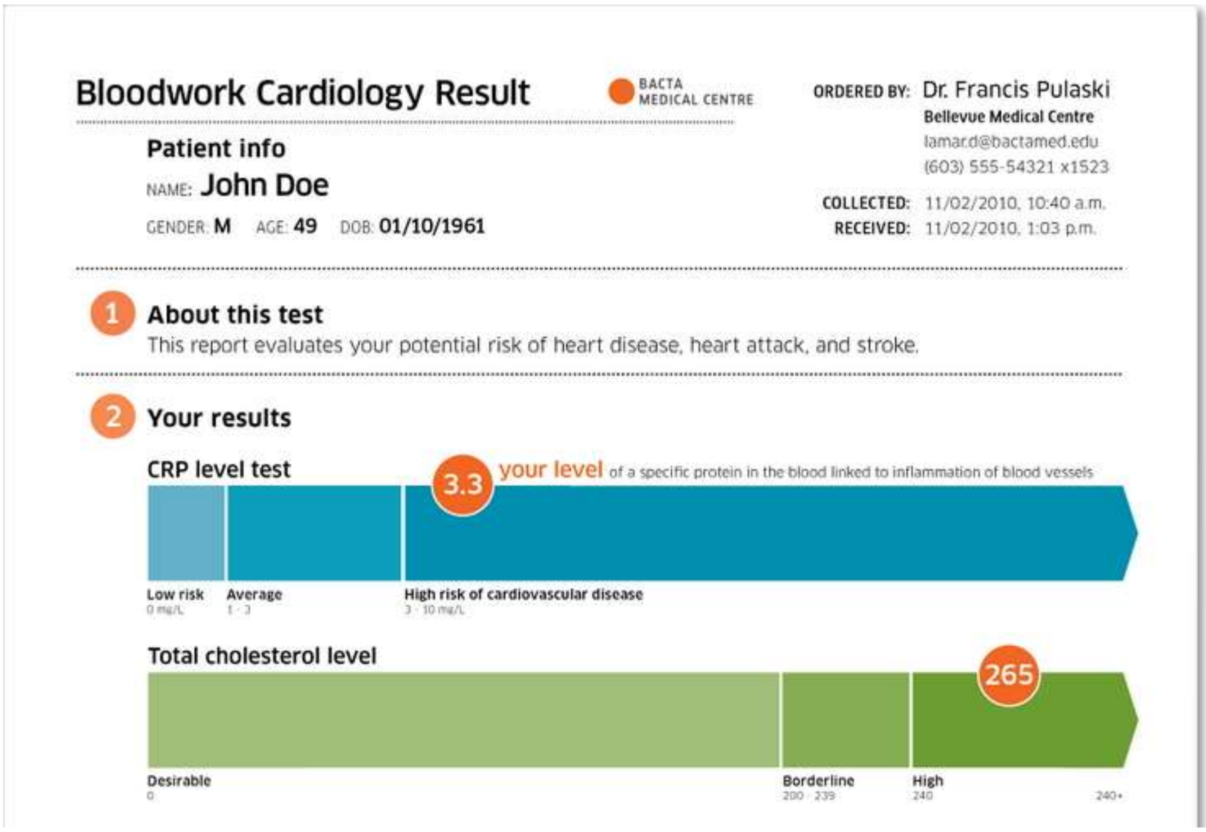


Of course it is not sensible to put every information into funny pictures or weird symbols. I only wanted to demonstrate that it makes a difference how information is presented.

Visualization

It's no secret that pure facts or figures are hard to comprehend. Diagrams and graphics make it easier to understand.

In the following case of visualized data of a blood test specific values are not only made visible, it also shows the range of "accepted" data.



Bloodwork Cardiology Result

BACTA MEDICAL CENTRE

ORDERED BY: Dr. Francis Pulaski
Bellevue Medical Centre
lamar.d@bactamed.edu
(603) 555-54321 x1523

COLLECTED: 11/02/2010, 10:40 a.m.
RECEIVED: 11/02/2010, 1:03 p.m.

Patient info

NAME: John Doe

GENDER: M AGE: 49 DOB: 01/10/1961

1 About this test

This report evaluates your potential risk of heart disease, heart attack, and stroke.

2 Your results

CRP level test
3.3
your level of a specific protein in the blood linked to inflammation of blood vessels
Low risk
0 mg/L
Average
1 - 3
High risk of cardiovascular disease
3 - 10 mg/L
Total cholesterol level
265
Desirable
0
Borderline
200 - 239
High
240
240+

Source: http://www.informationisbeautiful.net/

Relation

It is always a good idea to set data and figures in relation to other data. It helps to set priorities… Mostly it does not help at programming or documentation issues… A short dump is a short dump. No matter how many processes are error-free. Relational presenting helps with quantized data. In Programming we mostly have data of high quality.

Authenticity

It is very important, where information comes from and whether the source is authentic. If you read something in a specialist's book, you would not doubt that this information is true. If a colleague, you perhaps don't like, told you something then you might not be willing to believe this fact.

Conclusion

I am sure, I haven't told you something new. These facts are not so spectacular, but using these facts for a better information retrieval and documentation is indeed a very hard job!

You should learn to document your efforts, problems and solutions! Otherwise you are like the lumberjack who states: I cannot sharpen my axe. I have to cut down trees…

Context

Information without context is nothing. Context is the challenge of one of the most current problems nowadays: Big data. Meanwhile, in big data the quest is to **find** the context. In **Small Data** the quest is to document the context in a way that most people will understand it. Everyone has a different level of knowledge. It is hard to find an adequate way of documenting in a quite simple and easy way to understand and also provide as much information as possible and needed for the different levels.

If you are writing a documentation or preparing for a lesson you will have to ask yourself: Who is the recipient? When preparing a lesson it's quite easy because at first you can determine the level of previous knowledge and then you do the preparation for learning. When documenting you do it because it is part of the quest. But mostly you do not have an addressee and you do not have an exact "target" of what you want the reader to learn.

Technical programming documentation consists only of a list of created objects. Everything else will be part of a process documentation or end user documentation. So how much of the process documentation should be added to the technical documentation? It's hard to say and depends on a lot of other parameters. What is important is that you have someone to check this and decide upon it!

When using design patterns it is very important to document the used patterns as they are often hard to recognize!

Implicit context

The statement "that will not work" is very common. You will need to have really deep knowledge on something to be 100% sure of that. Most of the statements might be true but some of them are only true because of the fact that we have an implicit agreement about the context and the surroundings. On one hand it is very useful that you do not need to explain the whole context every time but on the other hand it might restrict our thinking on a clever solution.

It is very important to be aware of this context! It is important because you need to ask yourself if the person you are talking to knows about this context or if you should explain the surroundings.

I have a friend and colleague who often surprises me with leaving the implicit environment. It is very refreshing, because for him it is not only a question of "can this be done the normal way". If there is **any** technical possibility of doing something, he will think about it. Even if it is clear that this solution is out of the question because the effort to do this will be much too high. Having a solution – even if the solution is a bit unusual – is much better than having absolutely no idea!

Problem vs. solution

Due to the fact that it is often a long way between *having a problem* and *finding a solution* you have to think about this way afterwards and write it down. To recall all the right facts and all the wrong ways after a long time is a hard job, because it appears to be needless as you have reached your target: You have found the solution!

After having done this, you will need to describe all relevant issues to make others understand the context. In a programming environment it is easier than describing all necessary business environments in a process.

To understand the solution you have found it might be also useful to talk about other solutions or variants of the solution and describe why these are not ideal or why the presented solution is the best one.

Describing the problem and the solution is also very important when correcting errors. “It does not work” is not a valid excuse by the way! ;)

Problem solving

Information is not only about facts. It is also about experience and “doing”. The more often you use information the better you will remember it. This is the pitfall of the management of new information. You have dealt for a long time, with the problem and the solution is easy and logical. Everything fits together. But this “state of wisdom” will disappear very fast as the solution is mostly very small and your brain has not enough time and no motivation to save this information. For example, if you try to find out what is the pricing condition used for a specific process. You look through your notes, you ask your colleagues near you and you search the process documentation. You are searching for quite a long time. At least you have the right person on the phone. You type in the pricing condition and continue of what you wanted to do. Guess for how long you will remember this pricing condition... It is very unlikely that you will remember it a few weeks later. Therefore, you have to write down the information. How do you do this? Do you write ZABC in the right upper corner of your scribble book? If you do so, you have linked a lot more to the problem than only the information! You started an **activity** by writing. The **place** where you stored the information is not anonymous: It is in your notebook in the corner. You have written it in extremely heavy ink so that you will never again forget the pricing condition. Maybe you have put a decorated frame or scribble around the characters? All these additional information will help YOU to remember the information. If you want to **share** the information you will have to do lot more…

Tips for Problem solving

Extract the problem

- Try to reproduce the problem

- Write a small abstract (!) test program (it will help others to easily understand and reproduce the problem)
- Remove EVERYTHING that has nothing to do with the problem (easier debugging, easier to understand for others)

Describe the problem

- Describe the context (as much as it is needed for understanding the problem; the serial number of your hard disk might not be very useful in a programming problem…
- Write down the problem (it helps you to organize the facts)
- Be exact! (If you need someone's help it's not useful if the person has to guess)
- Start from the beginning (it helps others to follow your thoughts)

Ask your colleagues

- Ask a specific person to help you (if you ask many via Mail everyone might wait until someone reacts. If you ask someone directly not many people would refuse to help)
- Describe the problem (see above)
- Say what you have done so far (otherwise others will think about your problem, present a solution and you will answer: "I tried; it doesn't work." That is not very motivating. Presenting some efforts for problem solving also shows that you have already thought about the problem and the solutions by yourself. This will make others more willing to help.)

Knowledge in business

Mostly, there are specific people who have all the knowledge. In a company there is always one person who knows everything about a specific topic. If this person leaves the company, it then becomes very time-consuming to re-achieve the information.

In my opinion, every company should have its own knowledge base. It is grossly negligent to let employees save business relevant information in their own notebooks. Information is useless if it cannot be found.

So there are two things to do:

Build up a knowledge base

Use the tools you have. If you don't have a tool, buy one. Don't bother, the perfect tool does not exist yet! Use a wiki, install WordPress, buy SharePoint, SAP Jam, SAP Streamwork, Jive or whatever. At least, do something! If you don't, you might miss a big chance.

Try to find a tool that is easy to use and works with your common IT infrastructure such as Word, Excel and so on. This is very important for the recognition of the people who use this software.

There are a lot of enterprise collaboration tools on the market. Either for free or to buy.

It is also very important that you train all users as to how they should use the software! It is not only important how to use all the new functions but it is also important to explain the benefits.

Nominate a *knowledge base officer*, someone who encourages people in sharing information. He should be responsible for the tool and provide help and guidance in using it in an optimal way.

Try to use the gamification concept! Create funny user levels and make badges to make users want them. Take this topic but don't take it too seriously!

Reuse-Library

In SAP you have a complete reuse library you can use (transaction SE83). With transaction SLIB you can build your own library. Put general and important specific function modules in it. Write demonstration reports for using them. Document them. Document the context. You can put all of this information in your reuse library. It's in the standard SAP system! Use it! Refer to this transaction in your development guides. Encourage your team members to insert important functions if they are useful for other developers.

Plugins

Think in Plugins! Discuss different ways of developing reusable objects. You can save a lot of time when programming *objects* and general solutions. It takes a little time longer at the beginning, higher dependencies and it requires a little more preparation but afterwards you will greatly have a lot of advantages:

- Easier maintenance

- Analysing and correcting errors
- Easier improvements
- Reusability

Something about programmers

Programmers are very special people. Programmers don't like rules and programmers are lazy. I know that because I am a programmer. Programmers need freedom and the newest equipment. Give it to them – an adequate salary is obligatory – and you will have a loyal and hardworking team. Okay this is a theory and it is very stereotype. But I think you know what I mean.

Don't make the mistake in forcing programmers to follow useless guidelines! Guidelines are needed, There is no doubt about this and shouldn't be argued; But

a) only set up rules you can – and will! – control.

b) check guidelines for necessity.

A programmer will use a lot of effort and strength to bypass an existing stupid rule. If you have a rule that functions modules must not be called directly but need to be wrapped in a method you can be sure that you will have method names such as **z_mickey_mouse** or **z_may_the_force_b_with_u**! Since you have guidelines that forbid the use of comic characters or sexual content in method names I am quite sure that there is something wrong… ;)

Programmers like to play and experiment. The best way to get a programmer to do something is to tell him "they said it is not possible to do that." Developers need challenges and they need freedom to fulfil these challenges.

Why not do a monthly quest with a simple but challenging task? Find small programming challenges and let every programmer work out ideas and build prototypes. The result can be discussed the following month. Good ideas might be built in new developments and be stored in the reuse library.

Tips for spreading information

In the company where I work in, Inwerken AG, we periodically arrange workshops. We collect topics that we are interested in and urge someone to work on this topic and present it during the next workshop. It works because there is a colleague who is responsible for these workshops being held.

The way information is presented is very important for recalling. Of course, it differs between people. Some people are able to remember everything they have read. Other people need to really work on this. Most people, though, can remember best if information is being told by someone. Even if you do not remember the information itself you will remember the person who gave the information.

A lot of time is being spent on problem solving. If a solution is found it takes a lot of time to do the programming. If you periodically hold meetings on things you are currently working on, there will be many people who can give you a hint or who might say: "great! We need the same thing in our process."

I can tell you this will be hard work! You will need a lot of persuasive power. You will need a responsible person who takes care of the meetings. You will need someone to collect topics. The person should be a kind of connector between teams or departments. He needs to be aware of the problems within each team and can find topics that are interesting to every participant. Try to find small bonuses for everyone who gives a paper. But please: Do find something better than a plastic company pen!

If you carry out these types of meetings, I am sure that there will be less time wasted due to double (and triple) developments in your company.

Contaxt

At least I would like to present you a little tool that is part of Xing and is called CONTAXT. You can install it on your mobiles or tablets and you can easily get in contact with people in specific groups. I created the group SITHH2014 and you may join us with the code **RFC-BDT**. I would kindly like to ask only the participants in this room to join the group! Otherwise the meaning of this group is not the one we intended.

I hope I have managed to spark your interest, and I hope you have enjoyed and you will not forget my presentation.

Thank you for your attention