

**SEACON 2015**  
**Wege und Muster erfahren**  
**Einsteigen, umsteigen, ankommen - mit den richtigen Tracks für Ihre IT!**

Ein paar Fakten vorweg...

2015 kamen rund 200 Profis zusammen (Teilnehmerinnen und Teilnehmer, Sprecherinnen und Sprecher, Studis, Orgaleute).

Besonders die Teilnahme der Studis hat mich gefreut, dieses Jahr gab es diese Teilnehmergruppe zum ersten Mal. Dafür ist SIGS DATACOM Kooperationen mit Hochschulen eingegangen, und natürlich war die Teilnahme für die Studis gebührenfrei. Schließlich sollen sie später als Chefs wieder kommen und ihre Leute auch hinschicken...

Es gab wieder drei Tracks:

- Softwarearchitektur / Technologie
- Prozesse / Anforderungen
- Management / Führung

Und: Bitte, bitte! Reicht für die SEACON 2016 was zum Track „Prozesse / Anforderungen“ ein, da gibt es jede Menge Bedarf für gute Beiträge! Und Publikum gibt es auch genug!

Jetzt zu den einzelnen Sessions...

Meeno Schrader<sup>1</sup>, Wetterprofi und Regattasegler

### **Routing: Der kürzeste Weg ist nicht immer der schnellste**

Keynote

Hier ging es nicht um das Problem des Handlungsreisenden aus Informatiksicht<sup>2</sup> (nicht mal aus lustiger Informatiksicht...). Trotzdem kam uns im Publikum Vieles sehr vertraut vor...

Routing, definiert aus Seglersicht: „Optimaler Weg auf ein Ziel hin unter Berücksichtigung der äußeren Umstände“

Aufgaben und Rolle des (segelnden) Routers

- optimale Wege suchen
- Neues erkunden
- Ankommen sicherstellen
- Blickwechsel als Kernkompetenz
  - in der Situation
  - je nach Details
  - Kontext
  - das große Ganze (wo ist der nächste Kontinent)

Geradezu auf den Wind zu kann man bekanntlich nicht segeln. Also kreuzt man. Früher ging das nur bis 70° gegen den Wind, heute sind die Segelboote immerhin schon fähig, bis zu 20° nah ran zu kommen. Dass es mal 0° werden, das wäre bahnbrechend (bitte gern her mit Euren Ideen).

Wenn man also Gegenwind hat, braucht man nicht zu fluchen und zu heulen, sondern man kann auch den Nutzen sehen: Eine Herausforderung, die die Stärken zum Vorschein bringt. Als gewiefter Profi kann man so der Konkurrenz davon fahren. Gegenwind heißt, man muss neue Wege gehen. Aber auch: Man kann neue Wege finden!

<sup>1</sup> <http://www.sea-con.de/seacon2015/konferenz/konferenzprogramm/sprecher-detail/meeno-schrader.html>  
[http://de.wikipedia.org/wiki/Meeno\\_Schrader](http://de.wikipedia.org/wiki/Meeno_Schrader)

<sup>2</sup> [http://www.mathepedia.de/Problem\\_des\\_Handlungsreisenden.aspx](http://www.mathepedia.de/Problem_des_Handlungsreisenden.aspx)  
[http://www.explainxkcd.com/wiki/index.php/399:\\_Travelling\\_Salesman\\_Problem](http://www.explainxkcd.com/wiki/index.php/399:_Travelling_Salesman_Problem)

Beim Segeln wie in der Arbeitswelt sollte Vertrauen eines der wichtigsten Güter sein, denn gemeinsam ist man stärker – das ist nicht neu. Und im Team kann man die Kräfte bündeln – das ist ebenfalls nicht neu.

Vertrauensbildend ist übrigens auch, auf ein Scheitern gefasst zu sein und nicht überrascht da zu stehen, dass etwas anders kommt als erwartet. Damit offen umzugehen und so die Angst auf ein sachliches Maß zu begrenzen, erhöht die Qualität des Ergebnisses und verbessert Atmosphäre in der Zusammenarbeit.



Im Bild zu sehen: „Achtung, es lauert ein Gespräch!“ Die Kommunikation zwischen Rekordweltumseglerin (rechts) und Router (Wetterexperte und Kurs-Empfeher, links) muss kurz und präzise sein, denn das Telefon verschluckt eine Menge Infos und die Verbindung ist u. U. sehr schlecht. Auch hier hilft wieder.... das Vertrauensverhältnis.

Gemeinsam lassen sich dann Hürden umschiffen, und zwar mit Hilfe von...

- Szenarien
- Umwegen
- Weitblick

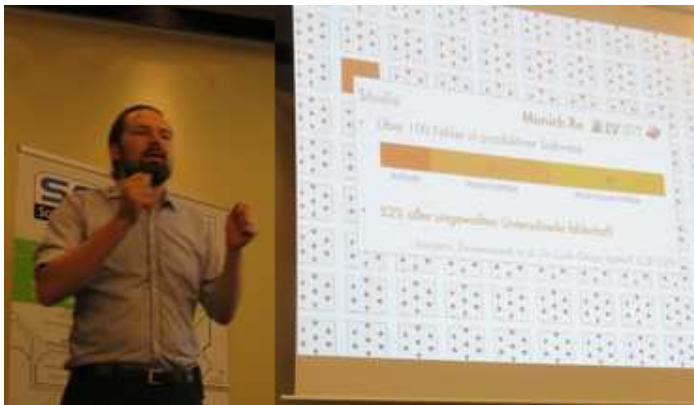
Dr. Elmar Jürgens, Mitgründer der CQSE GmbH

## Vom Wiegen wird die Sau nicht fett Durch Qualitätsanalyse zur Qualitätsverbesserung

Dann gibt es da dieses Tool, das zeigt, wo im Code überall copy & paste drin ist. Da kann man sehen, wo der Code so oft vorkommt, dass ich mal lieber eine Bibliothek draus mache... oder wo ich beim Bugfixen mehrfach ran muss. Außerdem hilft es der Teamleiterin, dem Chef zu zeigen, dass Zeit für's Refactoring sinnvoll ist und warum. Denn man sieht, wo riskanter Wildwuchs aufläuft. Neben Code-Klonen prüft das Tool im Code außerdem Verschachtelungstiefen (bei Tiefe 29 kann man sich schon denken, dass da vielleicht irgendwo eine Fallunterscheidung doppelt sein könnte... Code ist bekanntlich etwas, das organisch wächst, wenn man nicht ganz doll aufpasst. Und dann gibt es noch den Abgleich von festgelegter Architektur und Umsetzung in Code. Die Mittelschicht umschiffen mag zwar verboten sein, aber ist doch so schön schnell hingecodet.

Dr. Elmar Jürgens warnte klugerweise auch gleich davor, dem Tool blind zu vertrauen. Blindes Vertrauen in ein Tool... könnte Dir nicht passieren, oder...? Die Mischung von Werkzeugen und der Einsatz gesunden Menschenverstands (!) macht den Erfolg aus. Dementsprechend lautet seine Empfehlung: Fangt mal mit dem neusten Code an, das Tool stellt Euch nach dem Commit eine Rückmeldung zur Verfügung. Umfassendes Refactoring ist teuer und lohnt sich nur bei hohem erwarteten Benefit oder in geschäfts- oder systemkritischen Fällen.

Zurück zum geklonten Code:



Bugfixing ist wie Memory für IT-ler... ich finde ein Codefragment, das gefixt werden muss, und ich weiß, es gibt noch mehr (Stichwort copy & paste)... hier gucken... Nein, das is es nicht. Da gucken... nein, auch nicht. Dort gucken... ah ja! Und es waren aber nicht nur zwei Stellen, oder? Hm....

Übrigens (kann man schlecht lesen) steht in der Box vor den Memorykarten das Ergebnis einer Studie, und zwar: Dass 52 % aller ungewollten Unterschiede von Code-Klons fehlerhaft sind.

Wer das Tool benutzen möchte, kann sich unter [www.cqse.eu/en/products](http://www.cqse.eu/en/products) oder [www.cqse.eu/en/blog](http://www.cqse.eu/en/blog) schlau machen.

## Von Stoppuhren und Messprotokollen

Noch so ein Ding, von dem man immer mal wieder eingeholt wird: Im Nachhinein Performanceproblemen erklären Dieser Vortrag erklärte, wo und wie es sinnvoll sein kann, sich Stoppuhren einzubauen. Also sozusagen das Licht anzuschalten, damit man sieht, wo der Stau ist. Was muss man tun?

- Grenzwerte festlegen, ab denen es interessant wird
- Häufigkeiten berücksichtigen (z. B. bis zu 5 % Ausnahmen erlauben, bei denen es etwas länger dauern darf – wie lang dann?)
- Ansatzpunkte für die Messung bestimmen (z. B. im Nachrichtensystem: welche Nachrichtentypen sehen wir uns an?)

Im Fallbeispiel konnte man schließlich anhand der Kurven, die die Timestamps im Laufe der Zeit zeigten, einiges sehen bzw. aufgrund der Kurven durch Recherchen herausfinden, zum Beispiel:

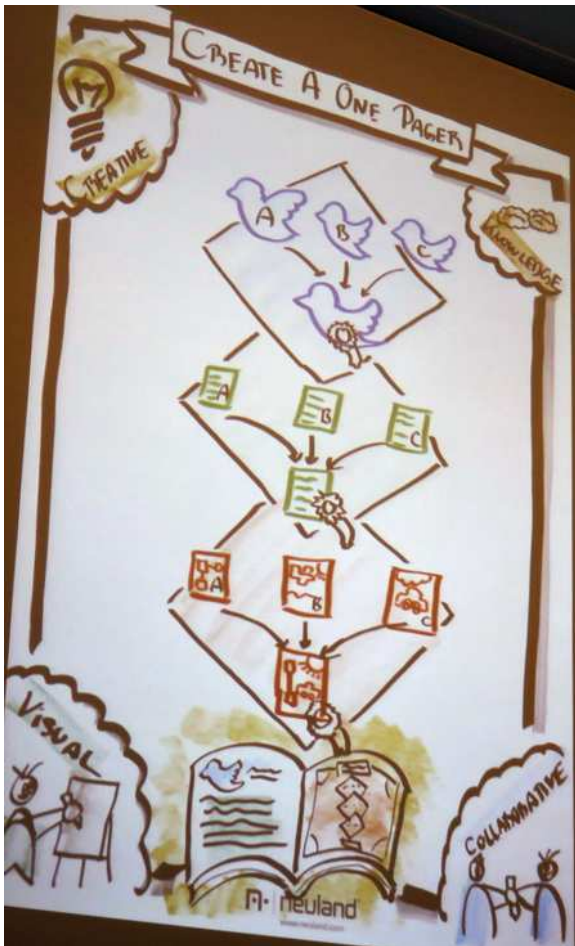
- DELETes machten die Datenbank immer langsamer – das Aufräumen braucht länger und länger
- Durchschnittszeiten wuchsen ganz langsam, aber stetig an, d. h. große Probleme in der Zukunft
- Der Garbage Collector schaffte es nicht mehr, ordentlich aufzuräumen, „Müllberge“ wachsen und bremsen das System aus

Fabian Schiller

## #PairSensing: Wissen teilen – kompakt und unterhaltsam

Fabian Schiller stellte in diesem Hands-on-Workshop eine Methode vor, mit der man auf unterhaltsame und kompakte Weise Wissen dokumentieren kann. Grundlage für die Dokumentation ist ein Querformat, das links aus Text und rechts aus einer Grafik besteht. (Meinetwegen auch anders herum). Wie das aussehen kann, könnt Ihr in der Publikation „Agile Planet“ sehen: <https://leanpub.com/agileplanet>

Kurz visualisiert, sieht die Methode so aus:



Zuerst schreibt jeder einen Tweet (rund 140 Zeichen), die das Thema beschreiben.

Aufgabe: „Formuliere den Tweet, der das Thema für Dich am besten ausdrückt.“

Das ist bewusst so offen formuliert, denn die unterschiedlichen Erfahrungswelten und das unterschiedliche Verständnis der Beteiligten ist das, was für die kreative Lösung am Ende wichtig ist.

Dann liest jeder den Tweet des andern und die Beiden einigen sich auf eine gemeinsame Tweetversion.

Als nächstes geht es wieder in die Einzelarbeit: Jeder schreibt in rund 200 Wörtern auf, was das Thema für sie oder ihn am besten definiert. 6 bis 8 Stichpunkte sind dafür ein guter Anhaltspunkt. Dann liest man wieder gegenseitig die Texte und einigt sich auf einen gemeinsamen Text.

Aus den Infos aus Tweet und Text entsteht jetzt wieder erst in Einzelarbeit und dann als gemeinsames Ergebnis eine Skizze bzw. grafische Elemente, die zum Thema und zu den Texten passen.

Das alles unter Zeitdruck, denn der macht kreativer.

Auch ich hab ein frei gewähltes Thema mit einem anderen Konferenzteilnehmer zusammen bearbeitet (Bild rechts), knapp definiert und visualisiert.

So sieht das aus (Bild rechts), wenn es GAAANZ schnell gehen muss. 30 Min für alles!  
In unserm Fall zum Thema „Standardsoftware oder Individualsoftware?“

Das Ergebnis des Workshops war jeweils eine kurze Dokumentation des gewählten Themas sowie das Verständnis der Methode. Ich glaube, ich werde die Methode bei Gelegenheit mal einsetzen. Man braucht dafür nicht mehr als 45 bis 75 Min Zeit.



## Pecha Kucha<sup>3</sup>

*Out of Context: Jetzt weiß ich, wie man sich die korrekte Aussprache merkt: Denk einfach das U wech und dann in einem Rutsch durch, also „Petschak-tscha“, Betonung auf der 2. Silbe.*

Im ersten Kurzvortrag bin ich wieder mal dran erinnert worden: Star Wars in ASCIIart<sup>4</sup> ... wird nur nach 1 Stunde ohne Ton vielleicht etwas anstrengend ; -)

Im zweiten Kurzvortrag höre ich den Namen Margret Hamilton, die als IT-Pionierin den Begriff „Software engineering“ geprägt hat. Als Direktorin war sie beim Apollo-Raumfahrtprogramm beteiligt. Mehr dazu gibt's natürlich im Web: [http://en.wikipedia.org/wiki/Margaret\\_Hamilton\\_%28scientist%29](http://en.wikipedia.org/wiki/Margaret_Hamilton_%28scientist%29)

Auch das Toyota-Produktionssystem kam mal wieder vor, auf dem die Idee der Agilität auch fußt: [http://www.artoflean.com/files/Basic\\_TPS\\_Handbook\\_v1.pdf](http://www.artoflean.com/files/Basic_TPS_Handbook_v1.pdf)

Kernpunkte hier

- kontinuierliche Verbesserung
- Respekt vor Menschen

... mit dem Ziel: Waste (Verschwendung, Müll) vermeiden

Schließlich noch was „Schönes“... ein Kurzvortrag darüber, dass empfundene Schönheit einer Software (also wie ansprechend das Design ist) Usabilityprinzipien beeinflussen kann. Als ansprechend empfundene Oberflächen können (kleinere) Usabilityfallen weniger schmerzhaft erscheinen lassen. Studienergebnis: Schönheit kann Usability überstrahlen

Ursache: Emotionen, unbewusst und unheimlich mächtig, lassen Menschen eine Aufgabe leichter und kreativer lösen, z. B. wenn sie glücklich sind

---

<sup>3</sup> [http://de.wikipedia.org/wiki/Pecha\\_Kucha](http://de.wikipedia.org/wiki/Pecha_Kucha)

<sup>4</sup> <http://www.asciimation.co.nz/>