Customer Orientation

Lean Six Sigma

Agile Processes

Project Estimations

Transfer Functions

Intuitionism and Computer Science

Why Computer Scientists do not Like the Axiom of Choice

Thomas M. Fehlmann, Zürich
Euro Project Office AG
E: info@e-p-o.com
H: www.e-p-o.com

16th Annual International Conference on Information Technology & Computer Science, 18-21 May 2020, Athens, Greece

QSM®

euro project office
Master your project.

Dr. Thomas Fehlmann

Customer Orientation

Lean Six Sigma

Agile Processes

Project Estimations

Transfer Functions

- 1981: Dr. Math. ETHZ
- 1991: Six Sigma for Software Black Belt
- 1999: Euro Project Office AG, Zürich
- 2001: Akao Price 2001 for original contributions to QFD
- 2003: SwissICT Expert for Software Metrics
- 2004: Member of the Board QFD Institute Deutschland – QFD Architect
- 2007: CMMI for Software – Level 4 & 5
- 2011: Net Promoter® Certified Associate
- 2013: Vice-President ISBSG
- 2016: Academic Member of the Athens Institute for Education and Research

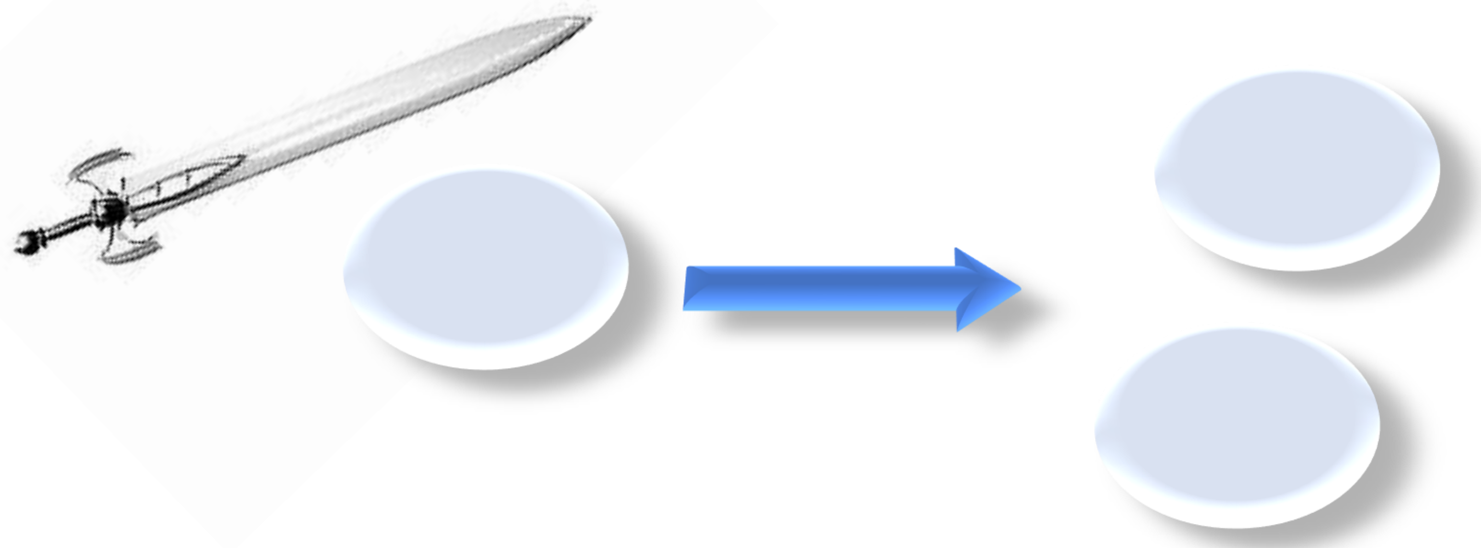
Eberhard Kranich

- Mathematics and Computer Science
- Emphasis on Mathematical Statistics
- Mathematical Optimization
- Theory of Polynomial Complexity of Algorithms
- Six Sigma Black Belt for Software Development
- Software Quality Assurance Manager

The Judgement of Salomon

Wallfahrtskirche Frauenberg, Steiermark

The Theorem of Banach-Tarski



- According the Theorem of Banach-Tarski, any solid sphere can be divided into pieces such that the pieces can be reassembled into two spheres, with each sphere the same size as the original sphere
- It can be proven using the **Axiom of Choice**
  - By constructing non-measurable sets that require an uncountable number of choices

Customer Orientation
Lean Six Sigma
Agile Processes
Project Estimations
Transfer Functions

Agenda

Intuitionism

Combinatory Logic

An Algebra of Tests

Conclusion

Agenda

Le
Six Sigma

Agile
Processes

Project
Estimations

Transfer
Functions

Intuitionism

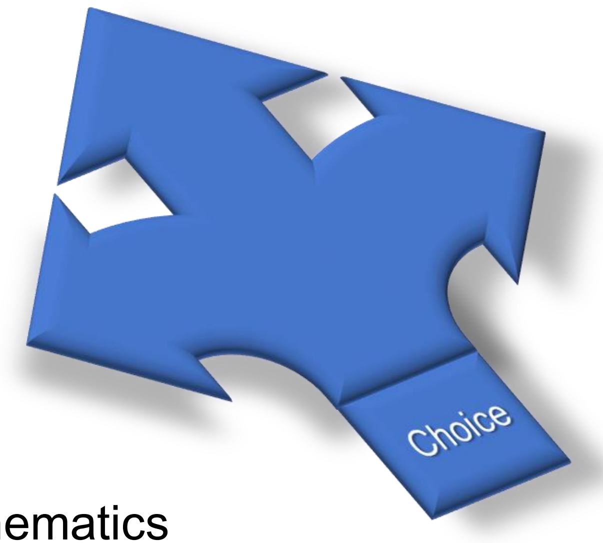
Combinatory Logic

An Algebra of Tests

Conclusion

The Axiom of Choice

- The **Axiom of Choice** (AC) says that given any family of non-empty sets S_i for $i \in I$, there exists a function f such that $f(i) \in S_i$ for all $i \in I$
  - f is called a **Choice Function**
  - A choice function simply selects one element from each set
- A Choice Function is a difficult notion for infinite sets
  - Which element to choose?
  - When are we finished with choosing?
  - When is our choice really representative?
- Assuming **AC**, the Banach-Tarski Paradox can be proven
  - Thus, difficult to believe the AC seems indispensable in mathematics



Leibniz Formula for π

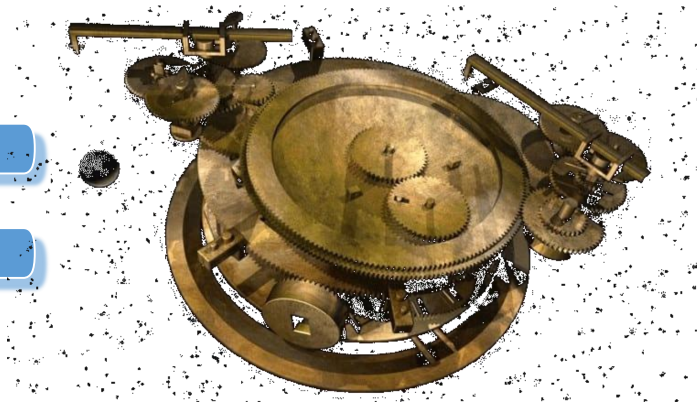
$$1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\cdots=\frac{\pi}{4}$$

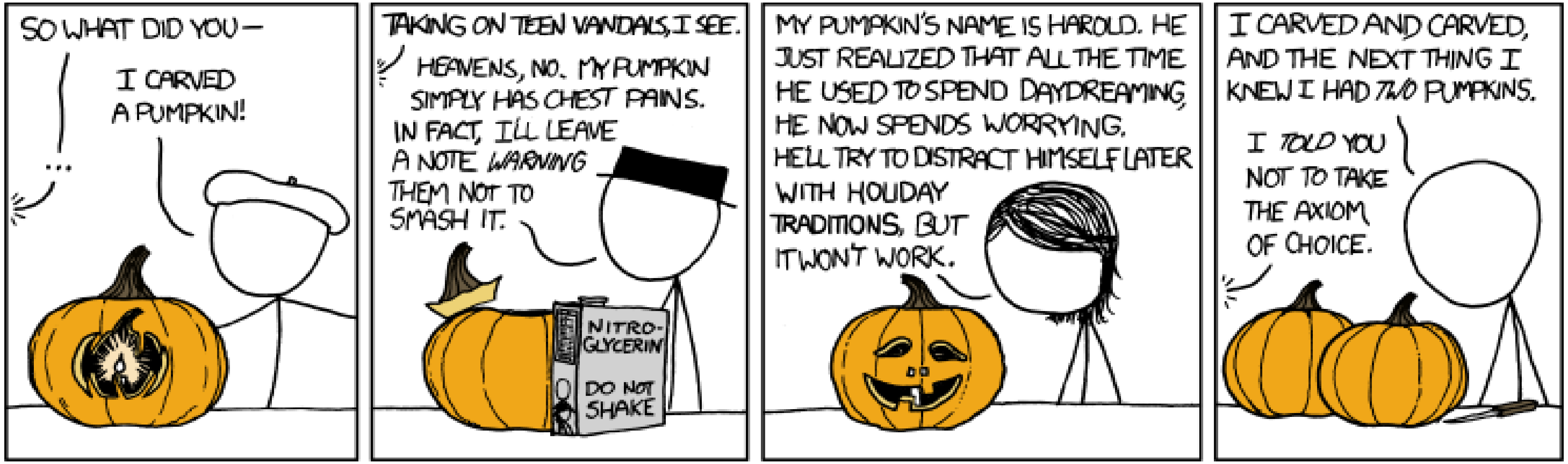
$$=\sum_{n=0}^{\infty}\left(\frac{1}{4n+1}-\frac{1}{4n+3}\right)$$

The Intuitionistic Variant of the Axiom of Choice

- "There Exists" means
  - There exists an algorithm that allows to select exactly one representative from each collection of sets
  - The choice function is computable

$\exists =$





Agenda

Intuitionism

Combinatory Logic

An Algebra of Tests

Conclusion

Agenda

Customer Orientation

Pr

Project Estimations

Transfer Functions

Intuitionism

Combinatory Logic

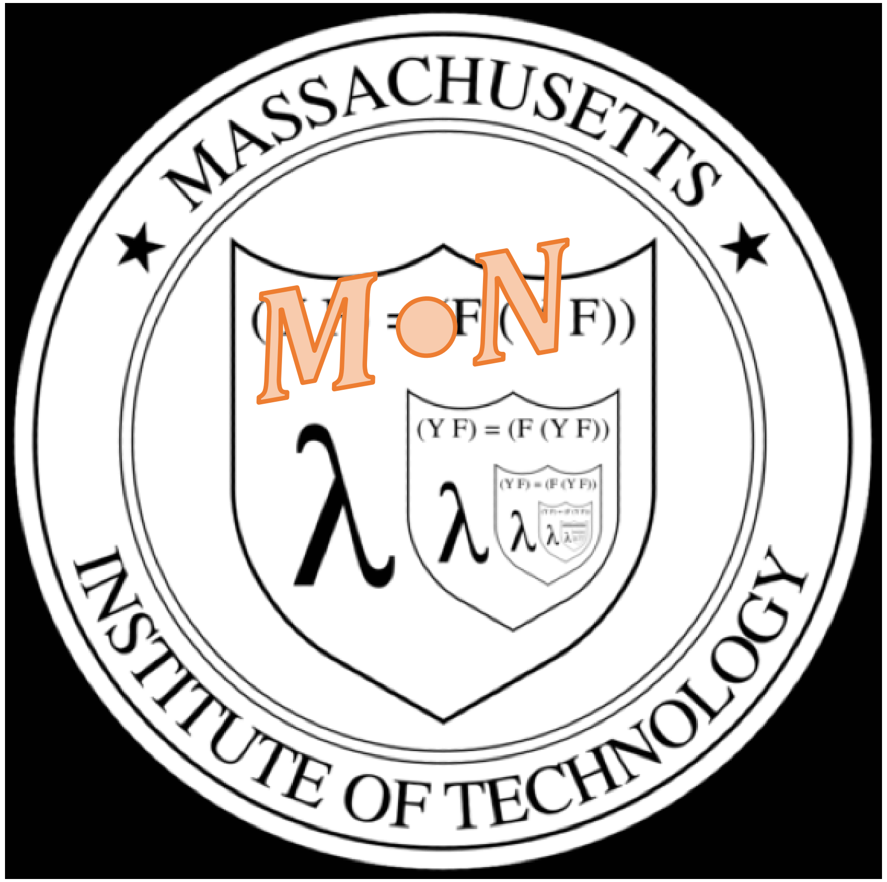
An Algebra of Tests

Conclusion

What is Combinatory Logic?

- A **Logic** is a formal structure
  - Of a set of terms
  - And a set of operators between terms

- **Combinatory Logic** consists of
  - An operation $M \bullet N$ defined for each pair of combinatory terms M, N in the combinatory algebra
  - $M \bullet N$ is again a combinatory term
  - Two distinguished **Combinators** called $\mathbf{S}$ and $\mathbf{K}$



The S-K Combinators

- The two Combinators $\mathbf{S}$ and $\mathbf{K}$ are characterized by the two equations

$$\mathbf{K} \bullet M \bullet N = M \text{ (Projection)}$$

$$\mathbf{S} \bullet M \bullet N \bullet L = M \bullet L \bullet (N \bullet L) \text{ (Substitution)}$$

With these two combinators any kind of program can be described. The $\mathbf{S}$-$\mathbf{K}$ calculus is **Turing-complete**, without variables and quantifiers!

- The identity combinator is defined as

$$\mathbf{I} := \mathbf{S} \bullet \mathbf{K} \bullet \mathbf{K}$$

$$I = S \bullet K \bullet K$$

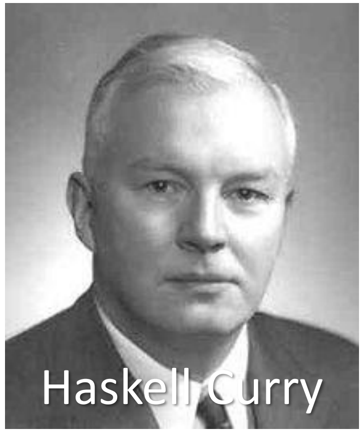
- Indeed, $\mathbf{I} \bullet M = \mathbf{S} \bullet \mathbf{K} \bullet \mathbf{K} \bullet M = \mathbf{K} \bullet M \bullet (\mathbf{K} \bullet M) = M$
- Combinatory algebras are the simplest language for describing formal languages

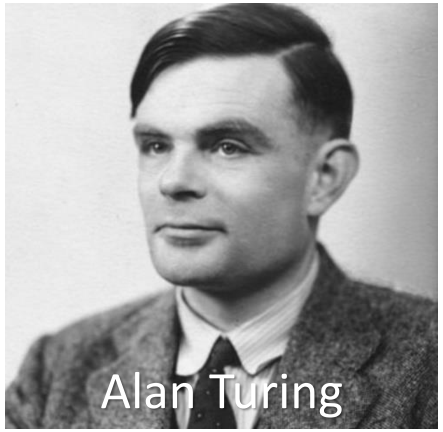
The Lambda Theorem

- The Lambda Theorem says that whenever there exists a Combinator $M(x)$ with some subterm x, there exists a combinator $\lambda x.Mx$ that fulfills

  $$(\lambda x.Mx) \bullet N = M(N)$$

  thus, by replacing x by N in M, this yields a Combinator too
- It means we can use the Lambda terms to construct schemata, combinators that take other combinatory terms as arguments, and all algorithms that run on a Turing machine
- **Foundation of Computer Science**





$\lambda x.Mx$

Models of Combinatory Logic

- A logic is only useful as there are **Models**
  - A model is an algebraic structure that fulfils the definitions given for the logic
  - Thus, it requires
    - A set providing the model elements, and
    - **S** and **K** Combinators
  - A method to combine two model elements
- Such a model is capable to describe the behavior of computers

Agenda

Intuitionism

Combinatory Logic

An Algebra of Tests

Conclusion

Agenda

Customer Orientation

Lean Six Sigma

Transfer Functions

Intuitionism

Combinatory Logic

An Algebra of Tests

Conclusion

Arrow Terms are a Combinatory Algebra for Software Testing

$$\{x_1, x_2, \ldots, x_n\} \rightarrow y$$

Test Data — Test Data — Test Data — Test Result

- The powerset of all Arrow Terms is a Combinatory Algebra
  - Powerset means including tests of tests
  - Algebraic combination of tests is a powerful method for enlarging test coverage

- Test Data refer to specific Data Groups
  - Data Groups are needed for test automation

- Test Stories are represented by finite sets of Arrow Terms with coherent scope

What Happens with Infinitely many Test Data as Controls?

$$\{x_1, x_2, \dots, x_n\} \rightarrow y$$

- In set-theoretical notation, the formal definition is

$$\mathcal{G}_0(\mathcal{L}) = \mathcal{L}$$

$$\mathcal{G}_{n+1}(\mathcal{L}) = \mathcal{G}_n(\mathcal{L}) \cup \{\{a_1, \dots, a_m\} \rightarrow b | a_1, \dots a_m, b \in \mathcal{G}_n(\mathcal{L})\}, m = 0,1,2,3 \dots$$

- $\mathcal{G}(\mathcal{L})$ is the set of all (finite and infinite) subsets of the union of all $\mathcal{G}_n(\mathcal{L})$:

$$\mathcal{G}(\mathcal{L}) = \bigcup_{n \in \mathbb{N}} \mathcal{G}_n(\mathcal{L})$$

- Write an explicit choice function as an index:
  - Denote a Test Case by $x_i \rightarrow y = \{x_1, x_2, \dots, x_n\} \rightarrow y$
  - Test Stories are finite sets of test cases $(x_i \rightarrow y)_j$
  - Rule Sets are any set of test stories $(x_i \rightarrow y)$
  - Every arrow term has a recursively defined size

Short:

Arrow Term:	$x_i \rightarrow y$
Test Story:	$(x_i \rightarrow y)_j$
Test Size:	$\|x_i \rightarrow y\|$

Combining Tests

- Let M, N be two test stories, consisting of test cases. N is a set of test cases consisting of arrow terms of the form $b_i \subset \mathcal{G}(\mathcal{L})$. Then application of M to N is defined by

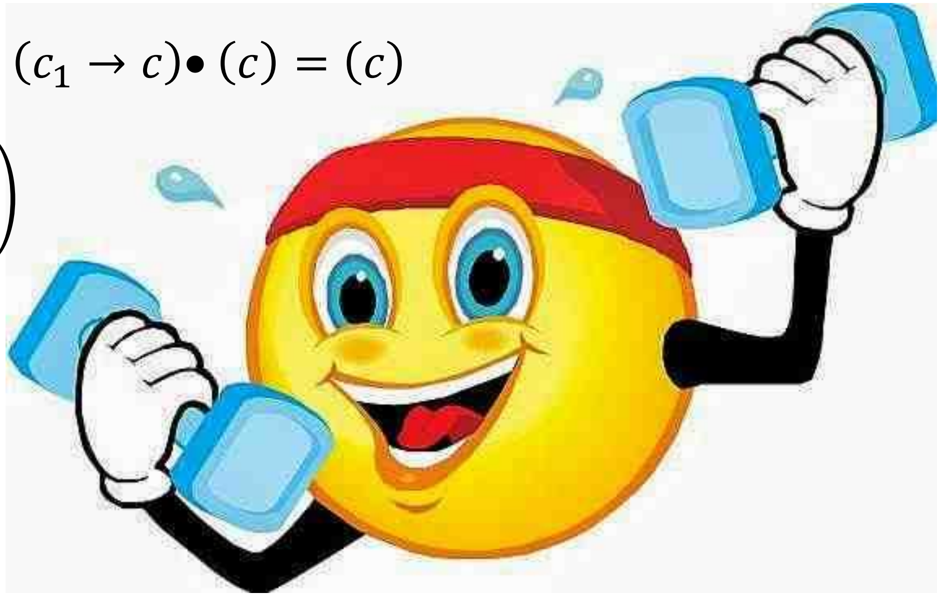
$$M \bullet N = \left((b_i \to a) \bullet (b_i)\right) = \{a \mid \exists b_i \to a \in M, b_i \subset N\}$$

$M \bullet N$

- If N is a test story, i.e., a finite, coherent set of arrow terms, the application $M \bullet N$ represents the selection operation that chooses those rules $(b_i \to a)$ from rule set M that are applicable to the test story N.
- Combining tests is a strong means to extend test stories as needed
  - The combination remembers which $b_i \to a \in M$ to select, because this was a constructive selection operation executed by a select function
  - This allows re-executing the selected tests in M
- Other combinators exist as well, such as projection, and many more

Arrow Terms – A Model of Combinatory Logic

- The following definitions demonstrate how arrow terms implement the combinators S and K
  - $\mathbf{I} = (a_1 \rightarrow a)$ is the Identification; i.e. $(a_1 \rightarrow a) \bullet (b) = (b)$
  - $\mathbf{K} = (a_1 \rightarrow \emptyset \rightarrow a)$ selects the 1st Projection:
    $$\mathbf{K} \bullet (b) \bullet (c) = \big((b_1 \rightarrow \emptyset \rightarrow b) \bullet (b)\big) \bullet (c) = (\emptyset \rightarrow b) \bullet (c) = (b)$$
  - $\mathbf{KI} = (\emptyset \rightarrow a_1 \rightarrow a)$ selects the 2nd Projection:
    $$\mathbf{KI} \bullet (b) \bullet (c) = \big((\emptyset \rightarrow c_1 \rightarrow c) \bullet (b)\big) \bullet (c) = (c_1 \rightarrow c) \bullet (c) = (c)$$
  - $$\mathbf{S} = \big(a_i \rightarrow (b_j \rightarrow c)\big)_1 \rightarrow \Big((d_k \rightarrow b)_i \rightarrow \big((b_j)_i \rightarrow c\big)\Big)$$
- Therefore, the algebra of arrow terms is a model of combinatory logic



Now we Have a Fountain to Generate Test Cases

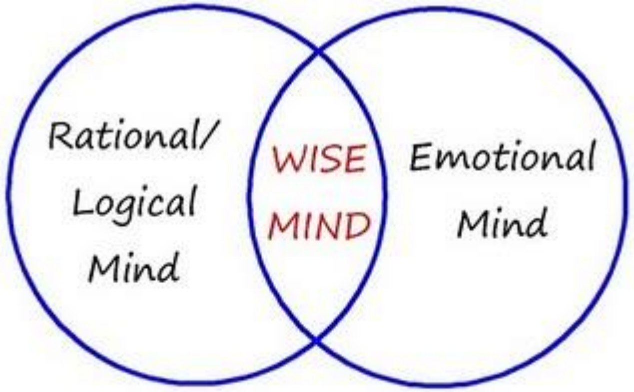
- A test case generator
- By combining other test cases
- By extracting new test cases from existing test cases
- All happens within a given test story

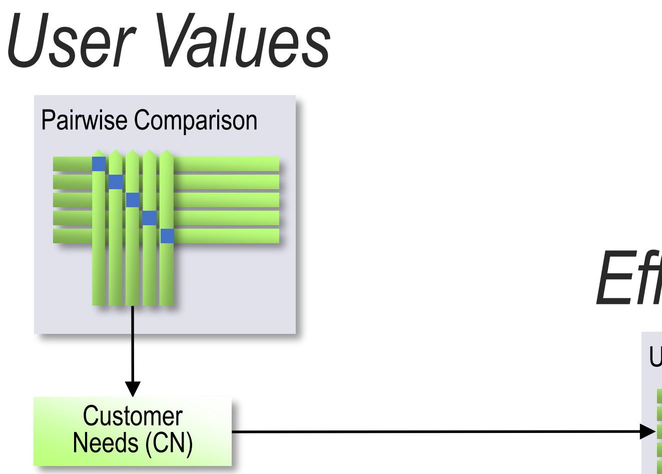
The Main Problem

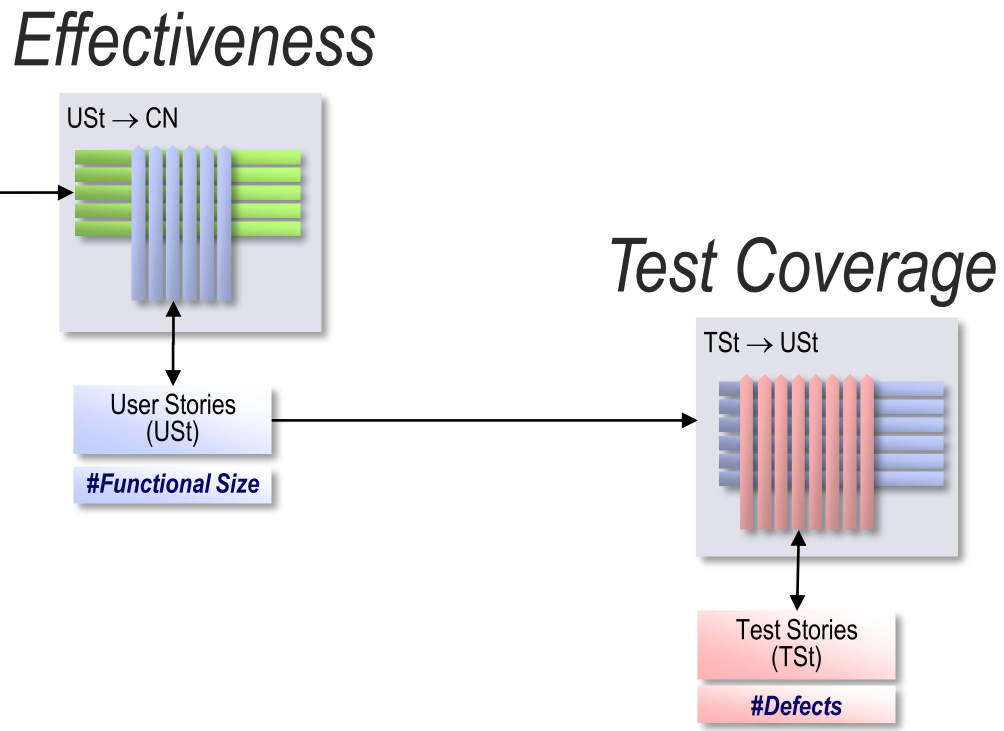
- How to select those test cases that are relevant to the purpose of the tests?
  - We cannot execute all possible tests
  - That would possibly exceed the lifetime of the universe
- We need **Transfer Functions**!

$$y = Ax$$

Goal Profile for User Values – Using Transfer Functions







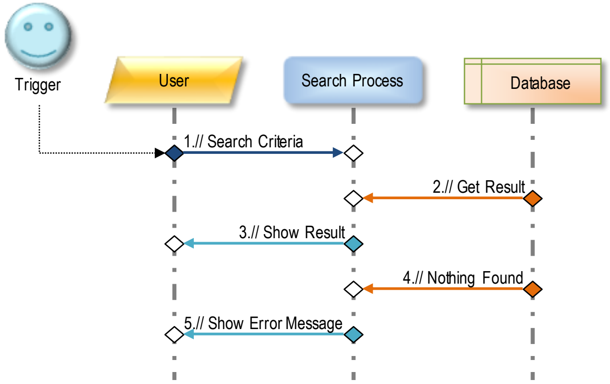
- How to get a goal profile for User Values?
  - Privacy Needs
  - Safety Needs
  - Emotional Needs
  - Business Needs

How do We Know What is Relevant?

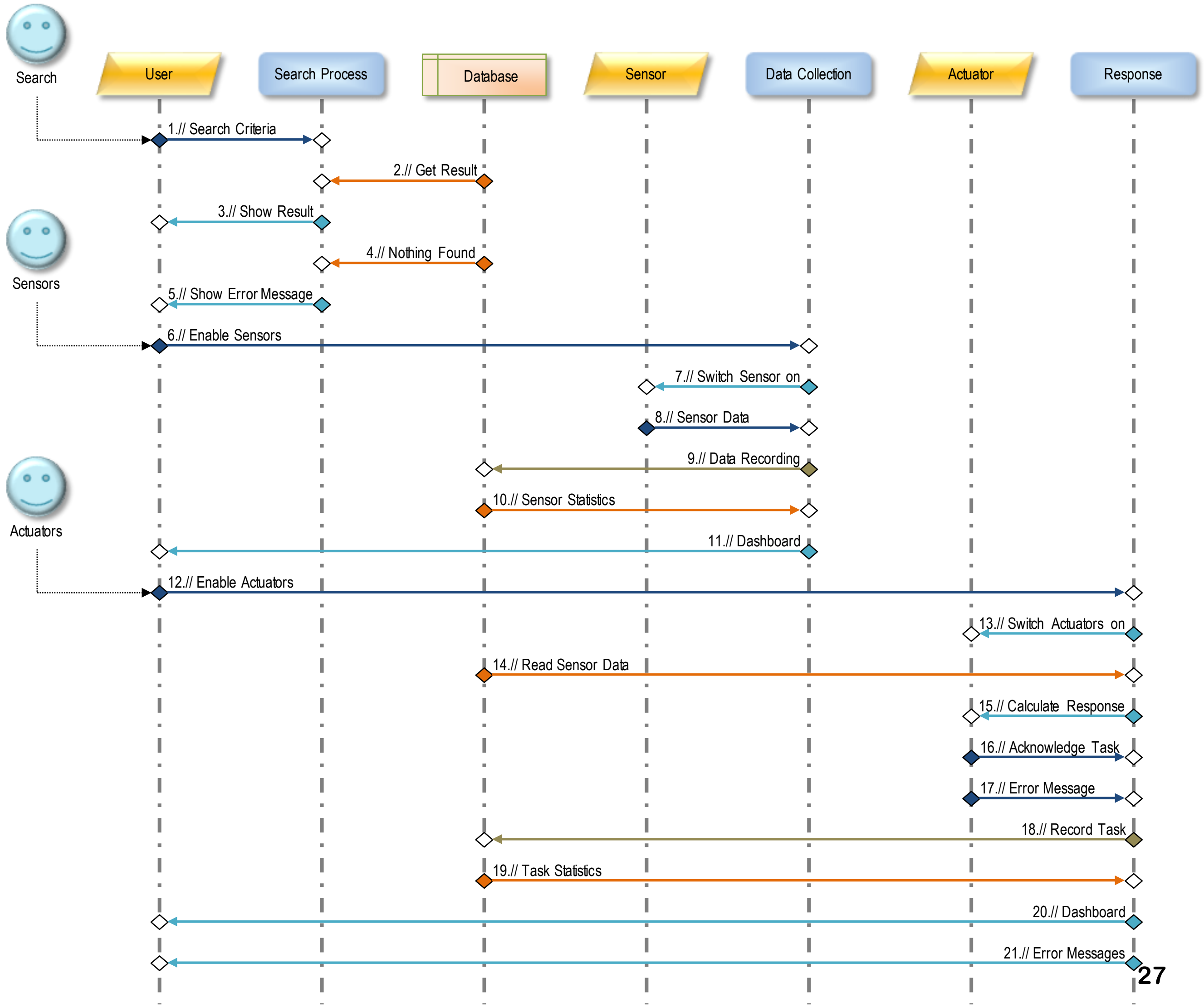
$$\text{Convergence Gap} = \|y - Ax\|$$

- We need a Goal Profile
  - Expressing the relative importance of User Stories
  - Agile Teams have such a profile
    - They need it for setting priorities in the Sprints!
- Compare what's being tested by the Test Stories with the User Story Profile
  - By counting frequency of data movements being executed
- If the Convergence Gap is close to zero
  - Test Stories test the responses that users expect
- If the Convergence Gap opens
  - Some Test Cases that matter for the users are missing, or superfluous

From Simple Search to an IoT Concert



Simple Search
IoT Concert



- Start with a simple search app
- Add sensors and actuators
  - Become an IoT Concert
  - Enhance search results by data observed by Sensors
  - Use Actuators to move things

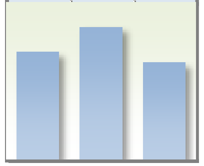
IoT Needs remain; Functional Effectiveness evolves

	User Values Topics	Attributes			Weight	Profile
FUR	y1 Extensible	*Easy to extend IoT*	*Device independent*	*Flexible*	*29%*	*0.54*
	y2 Open	*Open Source*	*Open Interfaces*		*24%*	*0.45*
NFR	y3 Reliable	*Always correct*	*Always secure*	*Safe*	*37%*	*0.68*
	y4 Fast	*No waiting*			*11%*	*0.20*

Simple Search

User Values
Deployment Combinator

User Values		*Goal Profile*	Q001 Search Data	Q002 Answer Questions	Q003 Keep Data Safe	*Achieved Profile*
y1	Extensible	0.54	3	1	1	0.50
y2	Open	0.45		4		0.48
y3	Reliable	0.68	2	2	3	0.70
y4	Fast	0.20	1		1	0.18
Solution Profile for User Stories:			0.55	0.68	0.49	*Convergence Gap*
						0.06



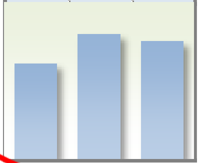
18 Total Effort Points
0.10 Convergence Range
0.20 Convergence Limit

IoT Concert

User Values
Deployment Combinator

User Values		*Goal Profile*	Q001 Search Data	Q002 Answer Questions	Q003 Keep Data Safe	*Achieved Profile*
y1	Extensible	0.54	8	11	9	0.52
y2	Open	0.45	7	10	4	0.39
y3	[illegible]	0.68	10	13	16	0.72
y4	Fast	0.20	5	4	4	0.24
Solution Profile for User Stories:			0.49	0.64	0.60	*Convergence Gap*
						0.08

Changed!



101 Total Effort Points
0.10 Convergence Range
0.20 Convergence Limit

Test Cases

Test Story	Case 1	Test Data	Expected Response	Case 2	Test Data	Expected Response
CT-A.1 Reliable Responses	CT-A.1.1	{Enter valid Search String}	Retrieved in Database	CT-A.1.2	{Enter invalid Search String}	Invalid Search String
CT-A.2 Detect Missing Data	CT-A.2.1	{Enter valid Search String for No Data}	No Data Available	CT-A.2.2	{Enter invalid Search String}	Invalid Search String
CT-A.3 Data Stays Untouched	CT-A.3.1	{Enter valid Search String}	Return identical Answer	CT-A.3.2	{Enter invalid Search String}	Invalid Search String

Simple Search

IoT Concert

Test Story	Case 3	Test Data	Expected Response	Case 4	Test Data	Expected Response	Case 5	Test Data
CT-A.1 Reliable Responses	CT-A.1.3	{Sensor Readings}	Retrieved in Database	CT-A.1.4	{Transmission Error}	No Data available	CT-A.1.5	{Actuator Enable
CT-A.2 Detect Missing Data	CT-A.2.3	{Sensor Off}	No Data available	CT-A.2.4	{Sensor Off}	Dashboard Indication	CT-A.2.5	{Actuator Off}
CT-A.3 Data Stays Untouched	CT-A.3.3	{Same Readings Again}	Return identical Answer	CT-A.3.4	{Transmission Interference}	Data Rejected	CT-A.3.5	{Actuator Set}

$$\left((b_i \to c)_j \to a\right)$$

where $(b_i \to c)_j$ is in the sensor's unit tests

Test Coverage for the Full IoT Concert

Simple Search

Test Coverage
Deployment Combinator

User Stories	Goal Test Coverage	1) CT-A.1 Reliable Responses	2) CT-A.2 Detect Missing Data	3) CT-A.3 Data Stays Untouched	Achieved Coverage
Q001 Search Data	0.55	4	5	7	0.59
Q002 Answer Questions	0.68	3	7	5	0.57
Q003 Keep Data Safe	0.49	3	7	5	0.57
Ideal Profile for Test Stories:		0.37	0.69	0.62	Convergence Gap **0.14**

46 Total Test Size
0.15 Convergence Range
0.20 Convergence Limit

IoT Concert

Test Coverage
Deployment Combinator

User Stories	Goal Test Coverage	1) CT-A.1 Reliable Responses	2) CT-A.2 Detect Missing Data	3) CT-A.3 Data Stays Untouched	Achieved Coverage
Q001 Search Data	0.49	33	19	23	0.54
Q002 Answer Questions	0.64	42	18	26	0.64
Q003 Keep Data Safe	0.60	35	19	20	0.54
Ideal Profile for Test Stories:		0.78	0.39	0.49	Convergence Gap **0.08**

235 Total Test Size
0.10 Convergence Range
0.20 Convergence Limit

- Automatic selection of additional test cases based on
  - Same Test Stories
  - Analogous Sensor Entries and Responses
  - Keep Convergence Gap → 0 as the selection criterion

Agenda

Intuitionism

Combinatory Logic

An Algebra of Tests

Conclusion

Agenda

Customer Orientation

Lean Six Sigma

Agile Processes

Project Estimations

Intuitionism

Combinatory Logic

An Algebra of Tests

Conclusion

Open Questions

- Will testing tools benefit from using Combinatory Algebra?
- Can we define combinators that target at generating meaningful additional tests?
- Is there a connection between matrix sensitivity analysis and such combinators?
  - ➡ The two last questions may not only lead to practical solutions, but also provide highly interesting theoretical insights towards the axiom of choice

Conclusions

Computer science uses choice functions only in a constructive way; existence of a choice always means existence of an algorithm that does the choice

- This is counter-intuitive to human perception of the world and might partly explain the difficulties encountered with digitalization and Artificial Intelligence (AI)
- The Axiom of Choice is connected to the Russel Paradox: No rule-based system will ever be free from contradictions

Arrow terms are an extremely rich structure for representing models for quite different topics such as

- The way how the brain thinks
- Product improvement with focus on customer needs by Quality Function Deployment (QFD)
- Testing of complex, software-intense systems with thousands of Embedded Control Units (ECU)

Questions?

Customer Orientation

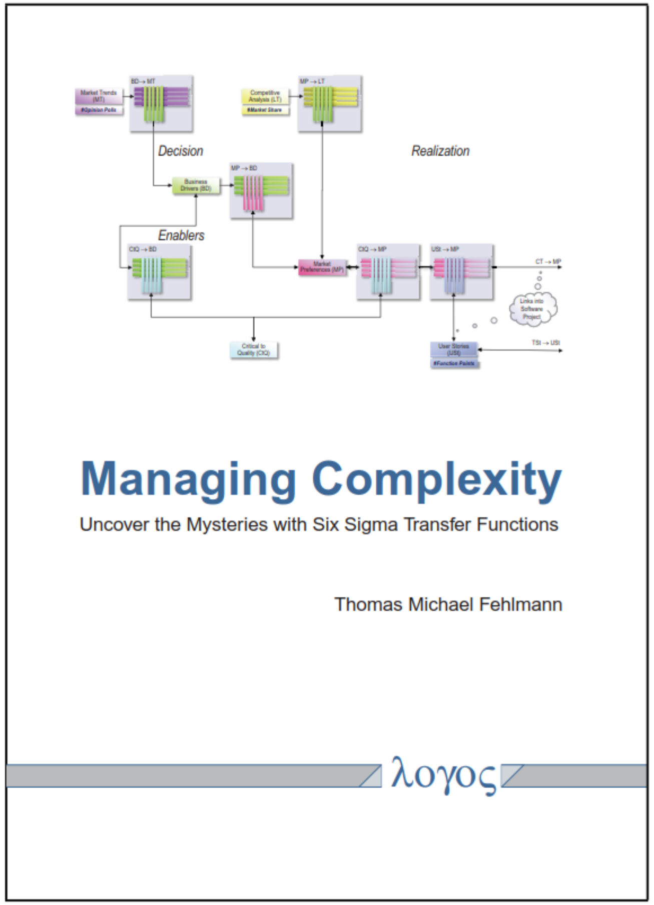
Lean Six Sigma

Agile Processes

Project Estimations

Transfer Functions

Managing Complexity with Six Sigma Transfer Functions (2016)



Managing Complexity

Uncover the Mysteries with Six Sigma Transfer Functions

Thomas Michael Fehlmann

2017, 394 pages
ISBN 978-3-8325-4406-5
Price: 49.00 €

www.logos-verlag.de
Search for "Managing"

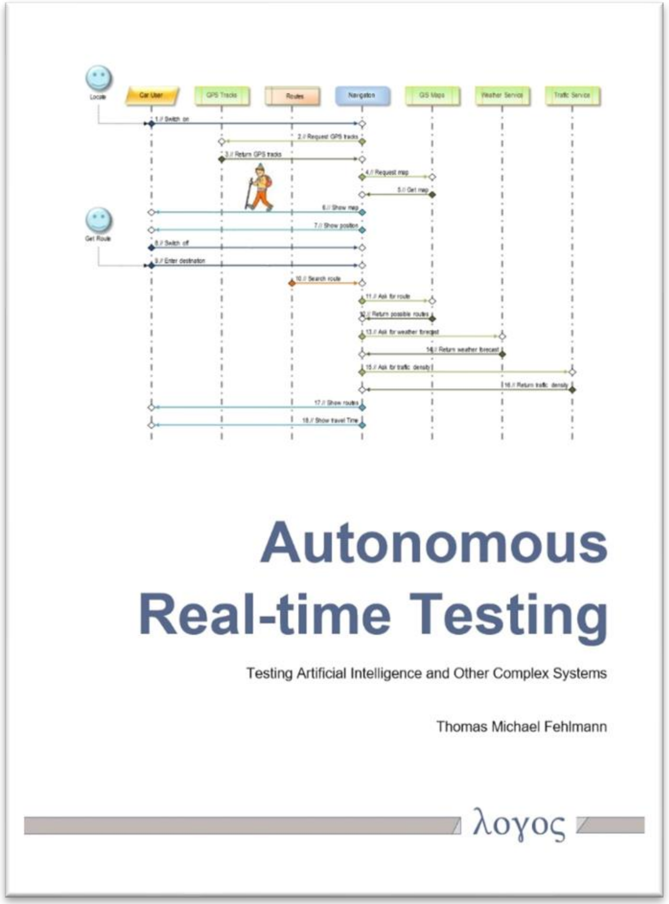
To purchase, please contact the bookselling trade or order online from Logos Press

Logos Verlag Berlin GmbH· Comeniushof – Gubener Str. 47 · D-10243 Berlin

λογος

Tel.: +49 (30) 42 85 10 90 · Fax: +49 (30) 42 85 10 92 · Internet: http://www.logos-verlag.com

Testing Artificial Intelligence & Other Complex Systems (2020)



Autonomous Real-time Testing

Testing Artificial Intelligence and other Complex Systems

Available since January 2020

www.logos-verlag.de
Search for "Testing"

To purchase, please contact the bookselling trade or order online from Logos Press

Logos Verlag Berlin GmbH· Comeniushof – Gubener Str. 47 · D-10243 Berlin

λογος

Tel.: +49 (30) 42 85 10 90 · Fax: +49 (30) 42 85 10 92 · Internet: http://www.logos-verlag.com