The Fixpoint Combinator in Combinatory Logic

A Step towards Autonomous Real-time Testing of Software?

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

17th Annual International Conference on Information Technology & Computer Science, 17-20 May 2021, Athens, Greece
Dr. Thomas Fehlmann

- 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
- 2017: Supporting Functional Sizing with Agile Product Development
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
Goals of this Presentation

1) Investigate the role of the Fixpoint Combinator for Software Testing

2) Understand how theoretical science contributes to advances in applied sciences and its economical impact

3) Have fun with Combinatory Logic
Combinatory Logic

The Graph Model

Software Testing

Conclusion
Foundation of Logic (Syllogistics)
- Categories
- Interpretation
- Prior Analytics
- Posterior Analytics
- Topics
- On Sophistical Refutations

What means
- For All? (∀x ... )
- There Exists? (∃x ... )

One Model:
- Euclidian Geometry
**S, K are terms of Combinatory Logic**

Whenever $P$ and $Q$ are terms of Combinatory Logic, then $P \cdot Q$ is also a term of Combinatory Logic

- Closed under *Combination*
- Replace variables by substitution

Create new Combinators

- Define the identity combinator
- Proof:
  - For all terms $P$ of Combinatory Logic

And many more…

- $K \cdot P \cdot Q = P$

- $S \cdot P \cdot Q \cdot R = P \cdot Q \cdot (P \cdot R)$

- $I := S \cdot K \cdot K$

- $I \cdot P = S \cdot K \cdot K \cdot P$
  - $= K \cdot P \cdot (K \cdot P)$
  - $= P$
The Lambda Theorem

- The **Abstraction Operator** $\lambda x.$ is defined recursively on sub-terms of term $M$

  $$\lambda x. x = I$$

  $$\lambda x. M = K \cdot M \text{ if } M \text{ different from } x$$

  $$\lambda x. M \cdot N = S \cdot \lambda x. M \cdot (\lambda x. N)$$

- Applying the Lambda combinator $\lambda x. M$ to any combinatory term $N$ replaces all occurrences of the variable $x$ in the term $M$ by $N$

  ➤ Written as

  $$\ (\lambda x. M) \cdot N$$
Given any combinatory term $Z$, the **Fixpoint Combinator** $Y$ generates a term

$$Y \cdot Z$$

called *Fixpoint* of $Z$ because it fulfills

$$Y \cdot Z = Z \cdot (Y \cdot Z)$$

The Fixpoint Combinator can be defined as

$$Y := \lambda f. \left( \lambda x. f \cdot (x \cdot x) \right) \cdot \left( \lambda x. f \cdot (x \cdot x) \right)$$

$$= S \cdot \left( S \cdot (S \cdot (K \cdot S) \cdot (S \cdot (K \cdot K) \cdot I)) \cdot \left( S \cdot (S \cdot (K \cdot S) \cdot (K \cdot I)) \cdot (K \cdot I) \right) \right) \cdot \left( S \cdot (S \cdot (K \cdot S) \cdot (S \cdot (K \cdot K) \cdot I)) \cdot \left( S \cdot (S \cdot (K \cdot S) \cdot (K \cdot I)) \cdot (K \cdot I) \right) \right)$$
Combinatory Logic

The Graph Model

Software Testing

Conclusion
The Graph Model of Combinatory Logic

\[ \{x_1, x_2, \ldots, x_n\} \rightarrow y \]

- Combinatory Logic
  - Connecting controls \( x_1, x_2, \ldots, x_n \) to responses \( y \)
  - Looks like a Neural Network
A directed graph, together with a firing law at all its nodes, constitutes the connective basis of the brain model \( \mathcal{A} \)

- Its elements are called **Neurons**

The model itself is built on this basis by identifying brain functions with parts of the firing history \( \{a_1, \ldots, a_m\} \rightarrow b \), where \( a_1, \ldots, a_m \) and \( b \) are neurons in \( \mathcal{A} \)

- \( \{a_1, \ldots, a_m\} \rightarrow b \) are called **Cascades**

Denote by \( \mathcal{G}(\mathcal{A}) \) the power set containing all cascades

- \( \mathcal{G}_0(\mathcal{A}) = \mathcal{A} \)
- \( \mathcal{G}_{n+1}(\mathcal{A}) = \mathcal{G}_n(\mathcal{A}) \cup \{ \{a_1, \ldots, a_m\} \rightarrow b | a_1, \ldots, a_m, b \in \mathcal{G}_n(\mathcal{A}), m \in \mathbb{N} \} \)

A **Thought** is a finite set of cascades

- It can contain thoughts about thoughts
Let $M$, $N$ be two thoughts, each consisting of cascades

- Let $N$ consist of cascades of the form $b_i$
- The subscript $i$ denotes a finite element selection from $N$

Then application of $M$ to $N$ is defined by

$$M \cdot N = (b_i \rightarrow a) \cdot b_i = \{a | \exists b_i \rightarrow a \in M, b_i \subset N\}$$

If $M$, $N$ are thoughts, i.e., a finite, coherent set of cascades

- The application $M \cdot N$ represents the selection operation
- Choosing those cascades $b_i \rightarrow a$ from thought $M$ that are applicable to the thought $N$
Focusing using Attractors

- Let $X$ be a Control Problem
  - $X$ is an expandable, unorganized set of thoughts
  - Apply the **Controlling Operator** $\mathcal{C}$ to $X$ with the aim to accomplish control
    
    $$\mathcal{C} \cdot X = X$$

- Solutions are obtained by **Control Sequences**, also known as **Focusing**
  
  $$X_{i+1} = \mathcal{C} \cdot X_i, \ i \in \mathbb{N}$$

- Starting with an initial thought $X_0$
  - Based on the control sequence $X_0 \subseteq X_1 \subseteq X_2 \subseteq \ldots$
  - Towards an optimum solution containing all elements of the control sequence
  - This optimum solution is called an **Attractor**
Attractors are control sequences
- Approximating a fixpoint solution
- With finitely many cascades
  - Or arrow terms
- Increasing in size
- Increasing in precision
  - Think of an unlimited Tower of Hanoi
  - Approaching its top
Combinatory Logic
The Graph Model
Software Testing
Conclusion
The **Data Movement Map** is a model for the software under test:

- Simplified UML
- Objects identified in code or service

Developers and Testers can “walk” the data movements when planning or executing tests:

- Functionality and Defects become visible and measurable to the development team

Same Metric:

- ISO/IEC 19761 COSMIC
The **Data Movement Map** is a model for the software under test

- Simplified UML
- Objects identified in code or service

Developers and Testers can "walk" the data movements when planning or executing tests

- Functionality and Defects become visible and measurable

Same Metric:

- **ISO/IEC 19761 COSMIC**

### Functional Size

- Number of Data Movements needed to implement all functional User Stories (FUR)

### Test Size

- Number of Data Movements executed in Test Cases

### Test Intensity

- Total Test Size divided by Total Functional Size

### Test Coverage

- Percentage of Data Movements covered by at least one Test Case

### Defect Count

- Number of Data Movements affected by some defect detected in a Test Story
Combinatory Algebra for Continuous Software Testing

\[ \{x_1, x_2, \ldots, x_n\} \rightarrow y \]

- Tests represented as **Arrow Terms** constitute a **Combinatory Algebra**
  - Combination of tests is a powerful method for enlarging test coverage
  - Combination means you can apply one set of tests to another

- **Test Data** refer to specific **Data Groups** with specific characteristics
  - Data Groups moved by Data Movements are needed for test automation

- Tests can **pass** or **fail**
  - We combine only tests that pass
Two Ways to Combine Test Cases

- **Combine Unit Tests**
  - Combining sets of terms in a Combinatory Algebra is unlimited
  - However, their associated data movement maps must overlap
    - Otherwise, the test cannot be executed
  - Usually within a single Test Story
    - To cover variations in data groups

- **Combine Test Stories**
  - Combine test cases whose data movement map overlaps
  - Often among different Test Stories
    - Combine two Test Stories from different components within a complex system
    - For instance, door controls with driving controls

---

**You can combine anything in a Combinatory Algebra**
Goal Profile for User’s Values – Using Transfer Functions

- How to get a goal profile for User’s Values?
  - Privacy Needs
  - Safety Needs
  - Emotional Needs
  - Business Needs
### Customer's Needs

<table>
<thead>
<tr>
<th>Customer's Needs Topics</th>
<th>Attributes</th>
<th>AHP Priorities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y.a Drive Fast</td>
<td>Y1 Agile Driving</td>
<td>16% 0.34</td>
</tr>
<tr>
<td></td>
<td>Y2 Smooth Driving</td>
<td>14% 0.30</td>
</tr>
<tr>
<td></td>
<td>Y3 Arrive in Time</td>
<td>23% 0.50</td>
</tr>
<tr>
<td>Y.b Drive Safe</td>
<td>Y4 Avoid Incidences</td>
<td>27% 0.59</td>
</tr>
<tr>
<td></td>
<td>Y5 No Surprises</td>
<td>21% 0.45</td>
</tr>
</tbody>
</table>

#### Customer's Needs Attributes
- Drive fast
  - Arrive safe
  - Do not block other traffic
  - Have fun
- Smooth driving
  - Drive predictably
  - Do not break unnecessarily
- Arrive in time
  - Arrive predictably
  - Avoid obstacles
- Avoid incidences
  - Drive foreseeful
  - Know what's ahead
  - Know my way
- No surprises
  - Communicate
  - Never surprise anybody
  - Give signs

#### AHP Priorities

<table>
<thead>
<tr>
<th>Y.a Drive Fast</th>
<th>Y1 Agile Driving</th>
<th>0.34</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y2 Smooth Driving</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td>Y3 Arrive in Time</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>Y4 Avoid Incidences</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>Y5 No Surprises</td>
<td>0.45</td>
</tr>
</tbody>
</table>

#### User Stories

<table>
<thead>
<tr>
<th>User Stories</th>
<th>Goal Profile</th>
<th>Q001</th>
<th>Q002</th>
<th>Q003</th>
<th>Q004</th>
<th>Q005</th>
<th>Q006</th>
<th>Achieved Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1 Agile Driving</td>
<td>0.34</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td>y2 Smooth Driving</td>
<td>0.30</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0.32</td>
<td></td>
</tr>
<tr>
<td>y3 Arrive in Time</td>
<td>0.50</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>0.49</td>
<td></td>
</tr>
<tr>
<td>y4 Avoid Incidences</td>
<td>0.59</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>0.58</td>
<td></td>
</tr>
<tr>
<td>y5 No Surprises</td>
<td>0.45</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>9</td>
<td>2</td>
<td>0.42</td>
<td></td>
</tr>
</tbody>
</table>

#### Solution Profile for User Stories

<table>
<thead>
<tr>
<th>y1 Agile Driving</th>
<th>0.42</th>
<th>0.35</th>
<th>0.28</th>
<th>0.54</th>
<th>0.21</th>
<th>0.53</th>
<th>0.05</th>
</tr>
</thead>
</table>

#### Total Effort Points

- 92
- Convergence Range: 0.10
- Convergence Limit: 0.20

- Count number of Data Movements in User Stories that support some specific Users’ Value

---

**Notes:**
- Y.a: Customer’s Needs
  - Y1: Agile Driving
  - Y2: Smooth Driving
  - Y3: Arrive in Time
- Y.b: Drive Safe
  - Y4: Avoid Incidences
  - Y5: No Surprises

---

**Diagram:**

- Customer's Needs: Y.a Drive Fast, Y.b Drive Safe
- Attributes: Arrive safe, Do not block other traffic, Have fun, Drive predictably, Do not break unnecessarily, Arrive predictably, Avoid obstacles, Drive foreseeful, Know what's ahead, Know my way, Communicate, Never surprise anybody, Give signs
- AHP Priorities: 16% 0.34, 14% 0.30, 23% 0.50, 27% 0.59, 21% 0.45

---

**User Stories:**

- y1 Agile Driving: 0.34, 4, 3, 1, 2, 5, 0.38
- y2 Smooth Driving: 0.30, 4, 3, 3, 2, 2, 0.32
- y3 Arrive in Time: 0.50, 5, 1, 3, 7, 3, 0.49
- y4 Avoid Incidences: 0.59, 4, 4, 3, 4, 8, 0.58
- y5 No Surprises: 0.45, 3, 1, 6, 9, 2, 0.42

---

**Solution Profile:**

- 0.42, 0.35, 0.28, 0.54, 0.21, 0.53

---

**Total Effort Points:**

- 92

---

**Convergence**

- Range: 0.10
- Limit: 0.20
## Intelligent Selection of Test Cases

### Test Coverage

<table>
<thead>
<tr>
<th>Test Stories</th>
<th>Goal Test Coverage</th>
<th>Achieved Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>People around</td>
<td>1) CT-A.1</td>
<td></td>
</tr>
<tr>
<td>Obstacle ahead</td>
<td>2) CT-B.1</td>
<td></td>
</tr>
<tr>
<td>Get route</td>
<td>3) CT-C.1</td>
<td></td>
</tr>
<tr>
<td>Change route</td>
<td>4) CT-C.2</td>
<td></td>
</tr>
<tr>
<td>Update position</td>
<td>5) CT-C.3</td>
<td></td>
</tr>
<tr>
<td>Approval</td>
<td>6) CT-D.1</td>
<td></td>
</tr>
<tr>
<td>Arrival time</td>
<td>7) CT-E.1</td>
<td></td>
</tr>
<tr>
<td>Leannings</td>
<td>8) CT-E.2</td>
<td></td>
</tr>
<tr>
<td>Keep under control</td>
<td>9) CT-F.1</td>
<td></td>
</tr>
<tr>
<td>Brake action</td>
<td>10) CT-F.2</td>
<td></td>
</tr>
<tr>
<td>Avoid stops</td>
<td>11) CT-F.3</td>
<td></td>
</tr>
</tbody>
</table>

### Deployment Combinator

#### User Stories

<table>
<thead>
<tr>
<th>User Stories</th>
<th>Goal Coverage</th>
<th>Achieved Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q001 Populated Area</td>
<td>0.42</td>
<td>0.44</td>
</tr>
<tr>
<td>Q002 Obstacle</td>
<td>0.35</td>
<td>0.40</td>
</tr>
<tr>
<td>Q003 Know my Way</td>
<td>0.28</td>
<td>0.26</td>
</tr>
<tr>
<td>Q004 Amend my Way</td>
<td>0.54</td>
<td>0.56</td>
</tr>
<tr>
<td>Q005 Check my Way</td>
<td>0.21</td>
<td>0.24</td>
</tr>
<tr>
<td>Q006 Able to Stop</td>
<td>0.53</td>
<td>0.46</td>
</tr>
</tbody>
</table>

### Ideal Profile for Test Stories:

| Test Cases Coverage | 0.46 | 0.42 | 0.22 | 0.17 | 0.33 | 0.21 | 0.30 | 0.21 | 0.27 | 0.22 | 0.36 |

### Test Size

- Total Test Size: 670
- Convergence Range: 0.10
- Convergence Limit: 0.20

### Convergence Gap

- 0.42
- 0.35
- 0.28
- 0.54
- 0.21

### Convergence Gap

- 0.42
- 0.35
- 0.28
- 0.54
- 0.21
Now let’s turn on the Test Generator…

- Stay with Test Stories and User Stories
- The Test Generator produces Test Cases that he can prove to yield correct results
  - In terms of privacy protection
  - And in terms of safety in driving

<table>
<thead>
<tr>
<th>User Stories</th>
<th>Goal Test Coverage</th>
<th>Test Stories</th>
<th>Achieved Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q001 Populated Area</td>
<td>0.42</td>
<td>39 27 5 7 9 6 10 8 12 14</td>
<td>0.49</td>
</tr>
<tr>
<td>Q002 Obstacle</td>
<td>0.35</td>
<td>16 18 10 5 14 5 11 9 13 16 10</td>
<td>0.36</td>
</tr>
<tr>
<td>Q003 Know my Way</td>
<td>0.28</td>
<td>4 5 13 5 13 9 9 6 7 9 8</td>
<td>0.20</td>
</tr>
<tr>
<td>Q004 Amend my Way</td>
<td>0.54</td>
<td>38 24 11 11 18 7 17 8 13 13 17</td>
<td>0.56</td>
</tr>
<tr>
<td>Q005 Check my Way</td>
<td>0.21</td>
<td>2 6 5 7 23 12 8</td>
<td>0.16</td>
</tr>
<tr>
<td>Q006 Able to Stop</td>
<td>0.53</td>
<td>40 31 5 2 10 4 6 8 10 8 13</td>
<td>0.51</td>
</tr>
</tbody>
</table>

Ideal Profile for Test Stories: 0.63 0.47 0.17 0.13 0.26 0.15 0.23 0.17 0.23 0.18 0.29

Convergence Gap: 0.13

Test Status Summary:
- Total CFP: 30
- Test Size in CFP: 745
- Test Intensity: 24.8
- Defect Density: 0.0%
- Defects Found in Total: 0
- Defects Pending for Removal: 0
- Data Movements Covered: 100%

745 Total Test Size
0.10 Convergence Range
0.20 Convergence Limit
After a few Test Cases more the Convergence Gap opens...

Adding Test Cases improves Test Density

- Here: Test Cases for image recognition in rain, or at night
- This might lead to lower the focus on Users’ Values
- The Convergence Gap opens

Test Coverage

Deployment Combinator

<table>
<thead>
<tr>
<th>User Stories</th>
<th>Goal Test Coverage</th>
<th>Test Stories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q001 Populated Area</td>
<td>0.42</td>
<td>39 27 9 11 9 6 10 8 12 16</td>
</tr>
<tr>
<td>Q002 Obstacle</td>
<td>0.35</td>
<td>16 18 14 9 14 5 11 9 13 20 11</td>
</tr>
<tr>
<td>Q003 Know my Way</td>
<td>0.28</td>
<td>4 5 18 9 13 9 9 6 7 13 10</td>
</tr>
<tr>
<td>Q004 Amend my Way</td>
<td>0.54</td>
<td>38 24 19 19 18 7 17 8 13 21 18</td>
</tr>
<tr>
<td>Q005 Check my Way</td>
<td>0.21</td>
<td>2 11 5 7 23 12 8 24</td>
</tr>
<tr>
<td>Q006 Able to Stop</td>
<td>0.53</td>
<td>40 31 5 2 10 4 6 8 10 10 14</td>
</tr>
</tbody>
</table>

Ideal Profile for Test Stories: 0.58 0.43 0.26 0.21 0.15 0.23 0.16 0.21 0.24 0.31

Test Status Summary

- Total CFP: 30
- Test Intensity: 27.5
- Total Test Size in CFP: 824
- Defects Found in Total: 0
- Defect Density: 0.0%
- Defects Pending for Removal: 0
- Data Movements Covered: 100%
The Double-Decker Tilting Long-Distance Multiple Unit Trainset (D2TLDMUTS) serves as an example to explain the new concepts. D2TLDMUTS is pronounced “Double-Tiddlemutzz”, with a sharp “zz” hiss at the end. It refers to a large Intercity multiple unit trainset, able to run on international railway traffic as a double-decker with restaurant, with children’s corner, offering space for people with disabilities, featuring roll compensation for faster driving around a curve, comfortable enough for three to six hours of daytime train riding.

It has been ordered by a European railway operator, originally targeted for 2013 but entering regular service in late 2020 only.

- Without restaurant because of COVID
- Commissioning started in February 2019 and lasted well into 2021
The method of choice to find priorities is the Analytic Hierarchy Process (AHP).

According ISO/IEC 16355

Surprise:

The need for unambiguous communication

**G04: Consistency** wins overall
# Consolidation among All AHP Software Components

## User Stories

<table>
<thead>
<tr>
<th>Test Stories</th>
<th>Test Stories</th>
<th>Test Stories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal Test Coverage</td>
<td>Goal Test Coverage</td>
<td>Goal Test Coverage</td>
</tr>
<tr>
<td>01) A.1</td>
<td>02) A.2</td>
<td>03) A.3</td>
</tr>
<tr>
<td>04) B.1</td>
<td>05) B.2</td>
<td>06) B.3</td>
</tr>
<tr>
<td>07) C.1</td>
<td>08) C.2</td>
<td>09) C.3</td>
</tr>
<tr>
<td>10) D.1</td>
<td>11) D.2</td>
<td>12) D.3</td>
</tr>
<tr>
<td>13) E.1</td>
<td>14) E.2</td>
<td>15) E.3</td>
</tr>
</tbody>
</table>

## Results

<table>
<thead>
<tr>
<th>Test Stories</th>
<th>Test Stories</th>
<th>Test Stories</th>
</tr>
</thead>
<tbody>
<tr>
<td>01) A.1</td>
<td>02) A.2</td>
<td>03) A.3</td>
</tr>
<tr>
<td>04) B.1</td>
<td>05) B.2</td>
<td>06) B.3</td>
</tr>
<tr>
<td>07) C.1</td>
<td>08) C.2</td>
<td>09) C.3</td>
</tr>
<tr>
<td>10) D.1</td>
<td>11) D.2</td>
<td>12) D.3</td>
</tr>
<tr>
<td>13) E.1</td>
<td>14) E.2</td>
<td>15) E.3</td>
</tr>
</tbody>
</table>

### Ideal Profile for Test Stories:

- **Attractor** $X_0$

---

### Graphs

- **Goal Test Coverage:**
  - Q001 Traction: 0.50
  - Q002 Comfort: 0.58
  - Q003 Stop: 0.37
  - Q004 Monitor: 0.54

- **Test Stories:**
  - Q001 Audio: 0.37
  - Q002 Information: 0.43
  - Q003 Entertainment: 0.48
  - Q004 Train Status: 0.47
  - Q005 Terminology: 0.29
  - Q006 Training: 0.38

- **Test Stories:**
  - Q001 Traction: 22 8 14 9 16 27 18 16 22 15 10
  - Q002 Comfort: 11 9 7 21 6 26 17 20 29 36 4
  - Q003 Stop: 28 14 19 16 25 15 5 8 3 6
  - Q004 Monitor: 15 10 7 17 8 20 5 2 2 38 57

- **Q006 Training:**
  - 01) A.1 | 02) A.2 | 03) A.3 |
  - 04) B.1 | 05) B.2 | 06) B.3 |
  - 07) C.1 | 08) C.2 | 09) C.3 |
  - 10) D.1 | 11) D.2 | 12) D.3 |
  - 13) E.1 | 14) E.2 | 15) E.3 |

- **Q001 Audio:**
  - 15 10 7 17 8 20 5 2 2 38 57

- **Q002 Information:**
  - 10 15 6 4 9 11 10 9 14 14 2 7

- **Q003 Entertainment:**
  - 14 9 16 15 19 11 10 16 11 6 12 8

- **Q004 Train Status:**
  - 8 10 14 7 17 8 10 14 11 6 12 6

- **Q005 Terminology:**
  - 4 6 10 2 2 16 14 13 15

- **Q006 Training:**
  - 4 4 8 4 8 12 4 6 14 6 25 17

### User Stories (Ideal Profile):

- **Attractor** $X_0$: 0.48 0.53 0.37 0.59 0.59 0.36 0.43 0.52 0.44 0.29 0.38 0.63 0.54 0.31 0.46
Extended Consolidation among All AHP Software Components

User Stories

<table>
<thead>
<tr>
<th>Q001 Traction</th>
<th>Q002 Comfort</th>
<th>Q003 Stop</th>
<th>Q004 Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.55</td>
<td>0.58</td>
<td>0.37</td>
<td>0.54</td>
</tr>
<tr>
<td>22 8 14 9 16 27 18 16 22 15 10</td>
<td>11 8 9</td>
<td>28 14 19 16 25 15 5 8 3 6</td>
<td>15 10 7 17 8 20 5 2 2 38 57</td>
</tr>
</tbody>
</table>

Ideal Profile for Test Stories: 0.33 0.18 0.23 0.20 0.20 0.24 0.20 0.28 0.46 0.39 0.31 0.28 0.27 0.22 0.33 0.22 0.26 0.30 0.25 0.31 0.30 0.30 0.23 0.23 0.66 0.52 0.20 0.21 0.23 0.22

Attractor X₁

Customer Orientation

Lean Six Sigma

Agile Processes

Project Estimations

Transfer Functions
## Extended Consolidation among All AHP Software Components

### Test Stories

<table>
<thead>
<tr>
<th>User Stories</th>
<th>Test Stories</th>
<th>Test Stories</th>
<th>Test Stories</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Goal Test Coverage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q001 Traction</td>
<td>0.59</td>
<td>22 8 14 9 16 27 18 16 22 15 10</td>
<td>8</td>
</tr>
<tr>
<td>Q002 Comfort</td>
<td>0.58</td>
<td>11 9 7 21 6 26 17 20 29 36 4</td>
<td>7</td>
</tr>
<tr>
<td>Q003 Stop</td>
<td>0.37</td>
<td>28 14 19 16 25 15 5 8 3 6 8</td>
<td>1</td>
</tr>
<tr>
<td>Q004 Monitor</td>
<td>0.54</td>
<td>15 10 7 17 8 20 5 2 2 38 57</td>
<td></td>
</tr>
<tr>
<td>Q001 Audio</td>
<td>0.37</td>
<td>11 9 16 10 9 14 14 2 7 6 18 3</td>
<td></td>
</tr>
<tr>
<td>Q002 Information</td>
<td>0.43</td>
<td>8 11 10 10 9 14 14 2 7 6 18 3</td>
<td></td>
</tr>
<tr>
<td>Q003 Entertainment</td>
<td>0.48</td>
<td>14 9 16 15 19 11 10 16 11 6 12</td>
<td>8</td>
</tr>
<tr>
<td>Q004 Train Status</td>
<td>0.47</td>
<td>8 10 14 7 17 8 10 14 11 6 12 6 11</td>
<td>8</td>
</tr>
<tr>
<td>Q005 Terminology</td>
<td>0.29</td>
<td>9 13</td>
<td>10 10 2 2 16 14 13 15</td>
</tr>
<tr>
<td>Q006 Training</td>
<td>0.38</td>
<td>7</td>
<td>4 4 8 4 8 12 4 6 14 6 25</td>
</tr>
<tr>
<td>Q001 Stop</td>
<td>0.58</td>
<td>11</td>
<td>10 27 19 7 8 4 4</td>
</tr>
<tr>
<td>Q002 Start</td>
<td>0.53</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>Q003 Safety</td>
<td>0.35</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Q004 Pressure</td>
<td>0.50</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

### Ideal Profile for Test Stories

- Q001 Traction: 0.59
- Q002 Comfort: 0.58
- Q003 Stop: 0.37
- Q004 Monitor: 0.54
- Q001 Audio: 0.37
- Q002 Information: 0.43
- Q003 Entertainment: 0.48
- Q004 Train Status: 0.47
- Q005 Terminology: 0.29
- Q006 Training: 0.38
- Q001 Stop: 0.58
- Q002 Start: 0.53
- Q003 Safety: 0.35
- Q004 Pressure: 0.50

### Additional Data

- Ideal Profile for Test Stories
- User Stories
- AHP Software Components
- Test Stories
- Goal Test Coverage
- Acceleration
- Consumption
- Recuperation
- Air Condition
- Smooth Run
- Predictability
- Urgency
- Detection
- Alerts
- Climate
- Coverage
- Direct Messages
- Composed Messages
- Switch Sources
- Infotainment
- Departure Table
- Priority Messages
- Learn about Priority
- Train Location
- Next Arrival
- Standard Terms
- Translation
- Learn Terms
- Use Learnings
- Open Door
- Pressure
- Close Doors
- Door occupied
- Door blocks
- All doors
- Need Repair
- Door unusable

### Customer Orientation

- Lean Six Sigma
- Agile Processes
- Project Estimations
- Transfer Functions

---

![Attractor X2](image-url)
The Need for a Theory

- Why are we doing theoretical stuff like logic and other basic sciences?
- This is the way to invent new business models
  - Probably the only sure way
  - Otherwise, you get lost in the jungle
Questions?

www.logos-verlag.com
Search for “Managing” and “Testing”