Customer Orientation

Lean Six Sigma

Agile Processes

Project Estimations

Transfer Functions

Functional Size Measurement in Agile

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

ICSEA Barcelona 5./6. October 2021
ESTAMOS: Empirical Studies and Measurement of Software

QSM®

euro project office
Master your project.

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
- 2015: Collaboration with QSM Associates Switzerland
- 2016: Academic Member of the Athens Institute for Education and Research
- 2017: Functional Sizing in Agile Product Development

Silvan Fehlmann

Six Sigma for Software

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

Goals of this Presentation

1) *Measure functional size of agile software development per sprint, not only per finished product*

2) *The amount of FUR implemented per sprint continually decreases while NFR increases*

3) *Comparing story points and functional size is useful*

4) *Choose IFPUG or COSMIC?*

Agenda

The Gedankenexperiment

The Findings

The Fractal Curve

Conclusion

Agenda

The Gedankenexperiment

The Findings

The Fractal Curve

Conclusion

Le
Six Sigma
Agile Processes
Project Estimations
Transfer Functions

From Projects to Sprints – Agile Development Cycle

fail

Independent Testing

pass

24h

Priority

10 day

Task in Jira

Definition of Done

Product Backlog
"*User Stories*"

Sprint Backlog
"*Work Items*"

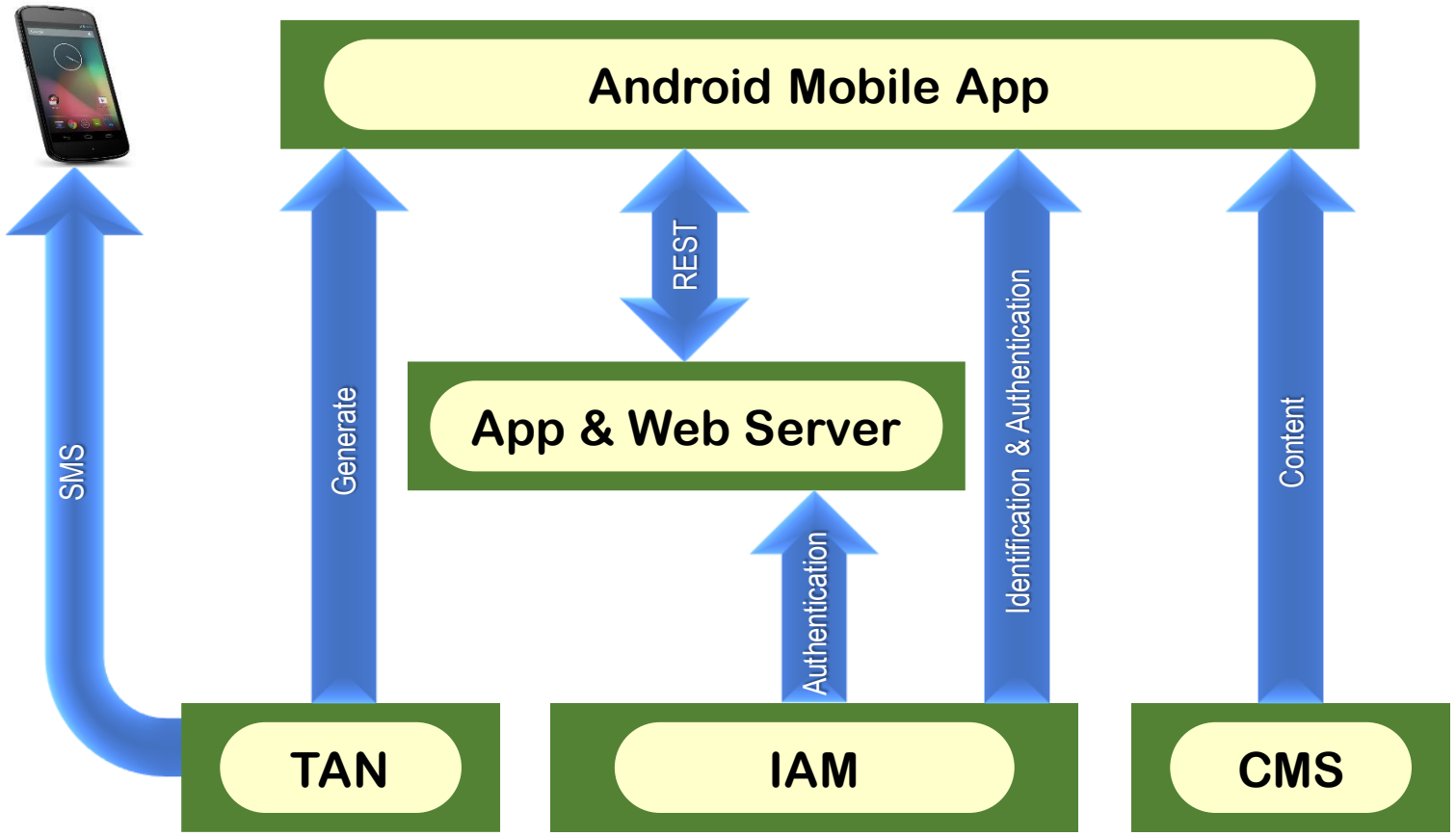
Sprint

Work Increment
of finished software

Eight User Stories – the Initial Backlog

Label	As a…	I want to…	Such that…	So that…
Login	App User	be sure to access my Giro Account	By using Fingerprint for identification and TAN for authentication	I can be confident for my privacy
Scan QR Code	App User	scan my bills	typing in IBAN and reference information is no longer necessary	paying bills is with one click
Use Giro Account	App User	use my Giro Account	I can access banking services with my Smartphone	to pay bills
Create Transactions	App User	create transactions	it's simple	to pay bills
Edit Transactions	App User	view & edit transactions	I'm informed about what I'll pay	account status remains under control
Schedule Execution	App User	select the date of execution	I can plan for my account balance	account status remains under control
Account Status	App User	review account status	all pending transactions are considered	account status remains under control
Refill	App User	link to a savings account	I can refill my Giro Account	I'm able to pay my bills

The Architecture

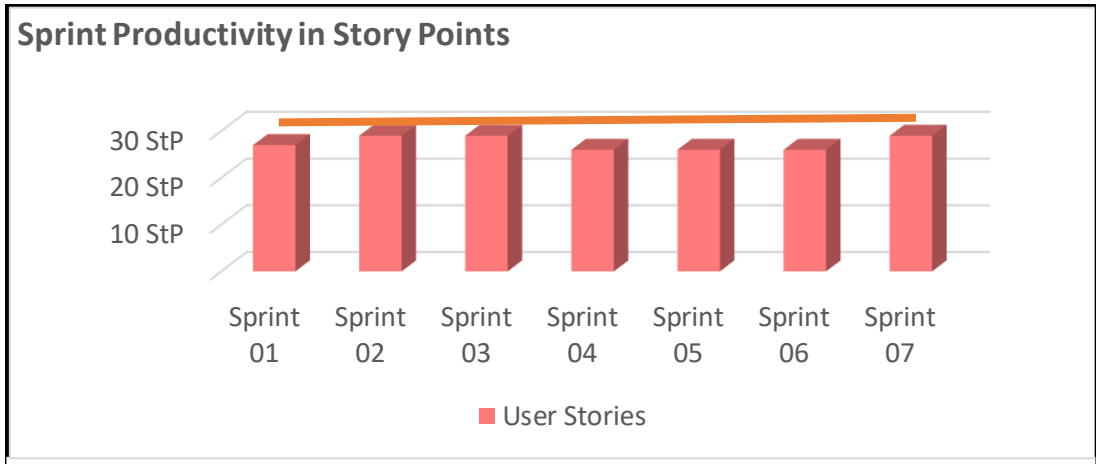


Seven Sprints

- Sprint 01 – **Allegro**
  - One user story from backlog
  - Two new user stories
- Sprint 02 – **Andante**
  - Three user stories from backlog
  - One new user story
- Sprint 03 – **Scherzo**
  - Two user stories from backlog
  - Three new user stories
- Sprint 04 – **Marche Funèbre**
  - One user story refactored
  - Two new user stories
- Sprint 05 – **Intermezzo**
  - Removing Technical Debt
  - One new user story
- Sprint 06 – **Menuetto**
  - Three new user stories
- Sprint 07 – **Finale**
  - One new user story
  - Final Tests

Retrospective

- Six (6) out of seven (7) user stories from backlog implemented
  - One converted in new user stories
- Thirteen (13) new user stories found during sprints
  - Not all fully functional
  - Means there is effort needed, expressed in story points, but no functionality added



Productivity in StP

Total StP	*#Sprints*	*Length*	*Hours/Day*	*Team Size*	*Delivery Rate*
192 StP	7	10 Days	7.2 h	6.8	**18 h/StP**

Agenda

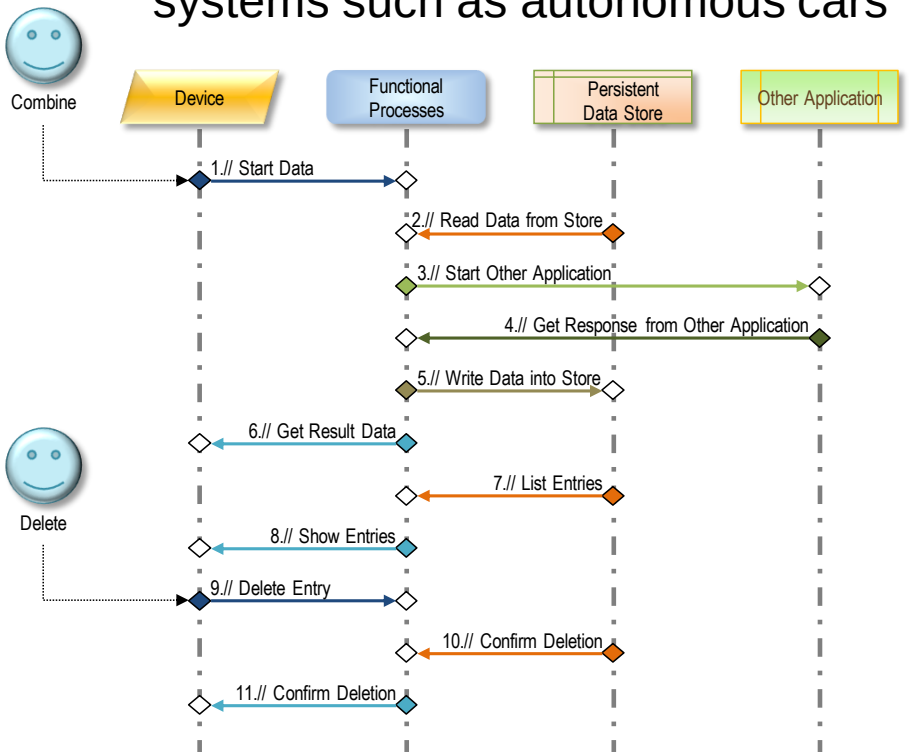
What is the Difference Between a Metric and a Count?

- A size count is not necessarily a size metric
  - A counting method counts objects of interest
  - A metric measures objects of interest
- Metrics comply to the VIM and the GUM:
  - The VIM: ISO/IEC Guide 99:2007, 2007. International Vocabulary of Metrology – Basic and general concepts and associated terms (Vocabulaire International de Métrologie – VIM)
  - The GUM: ISO/IEC CD Guide 98-3, 2015. Evaluation of measurement data - Part 3: Guide to Uncertainty in Measurement (GUM)

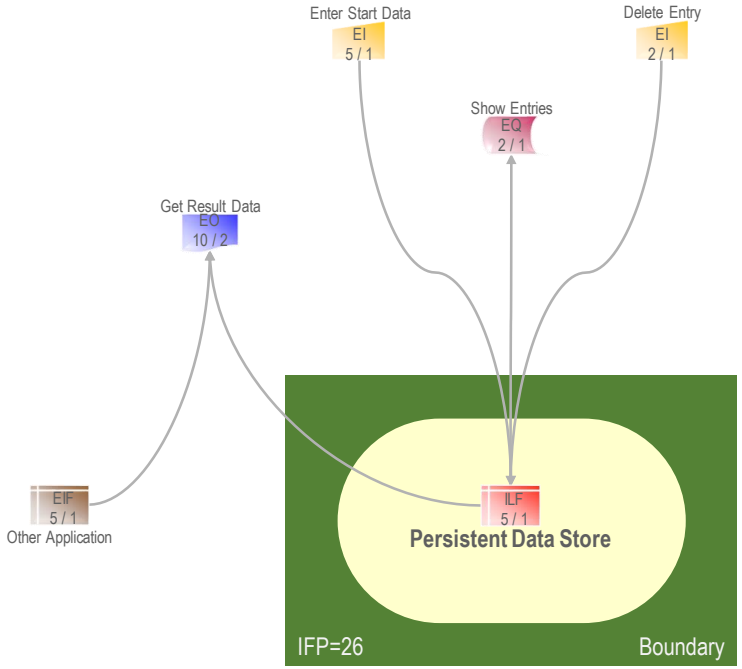
- In simple words: **You can add & subtract metrics**

IFPUG and COSMIC

- ISO/IEC 19761 COSMIC
  - Data Movement Map as SW model
  - Useful for communications, embedded SW, IoT, cyber-physical systems such as autonomous cars

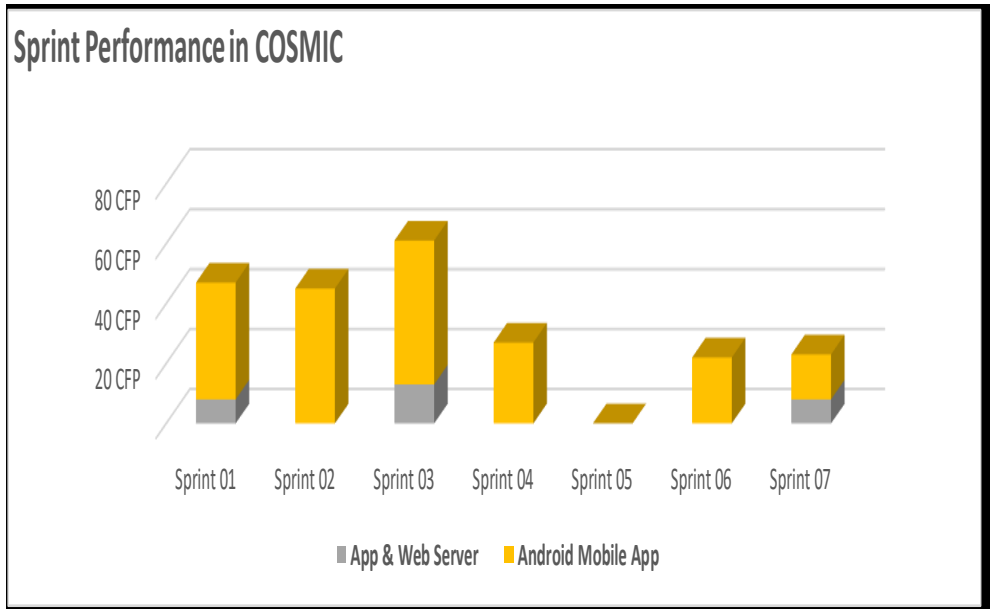


- ISO/IEC 20926 IFPUG
  - Transaction Map as SW model
  - Useful for transactional systems mostly for commercial applications
  - Suits our Android Mobile App

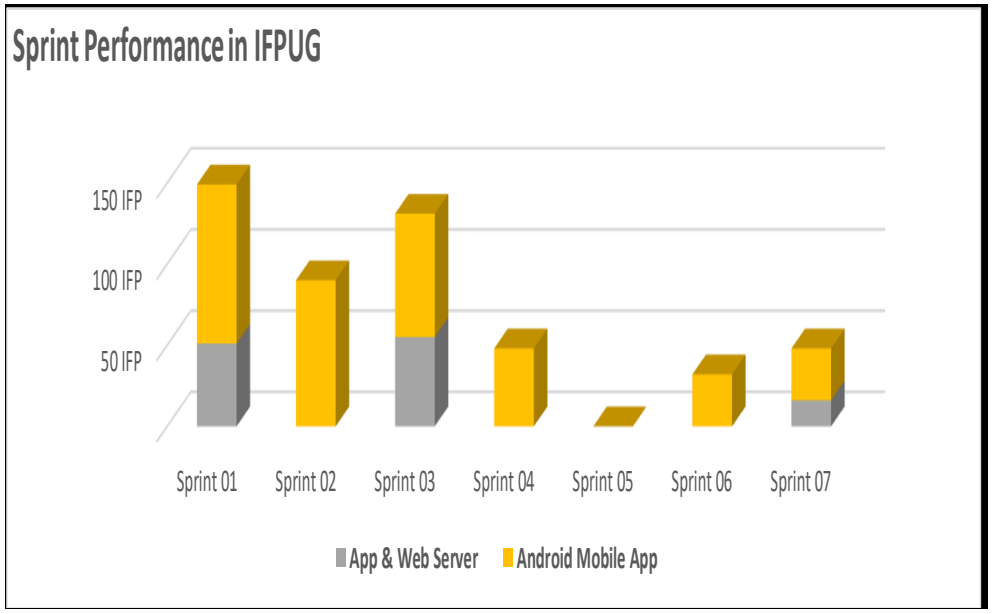


Sprint Performance

- Most new functionality in the first four sprints
  - ➡ Marche Funèbre - refactoring only

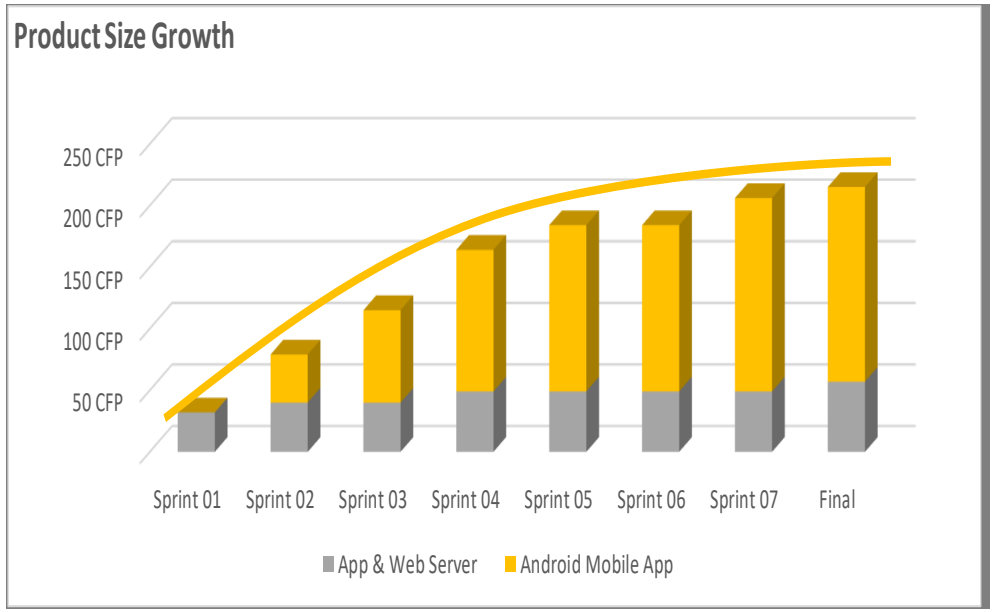


- Most new functionality in the first four sprints; more bias
  - ➡ Similar pattern

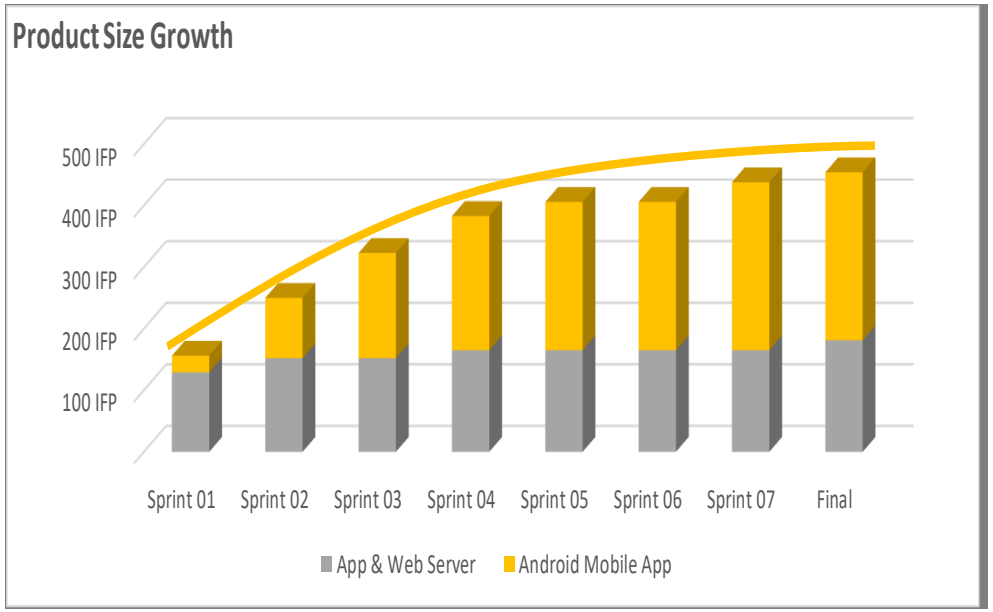


Product Growth

- COSMIC
  - Growth decreases
  - Exact measurement



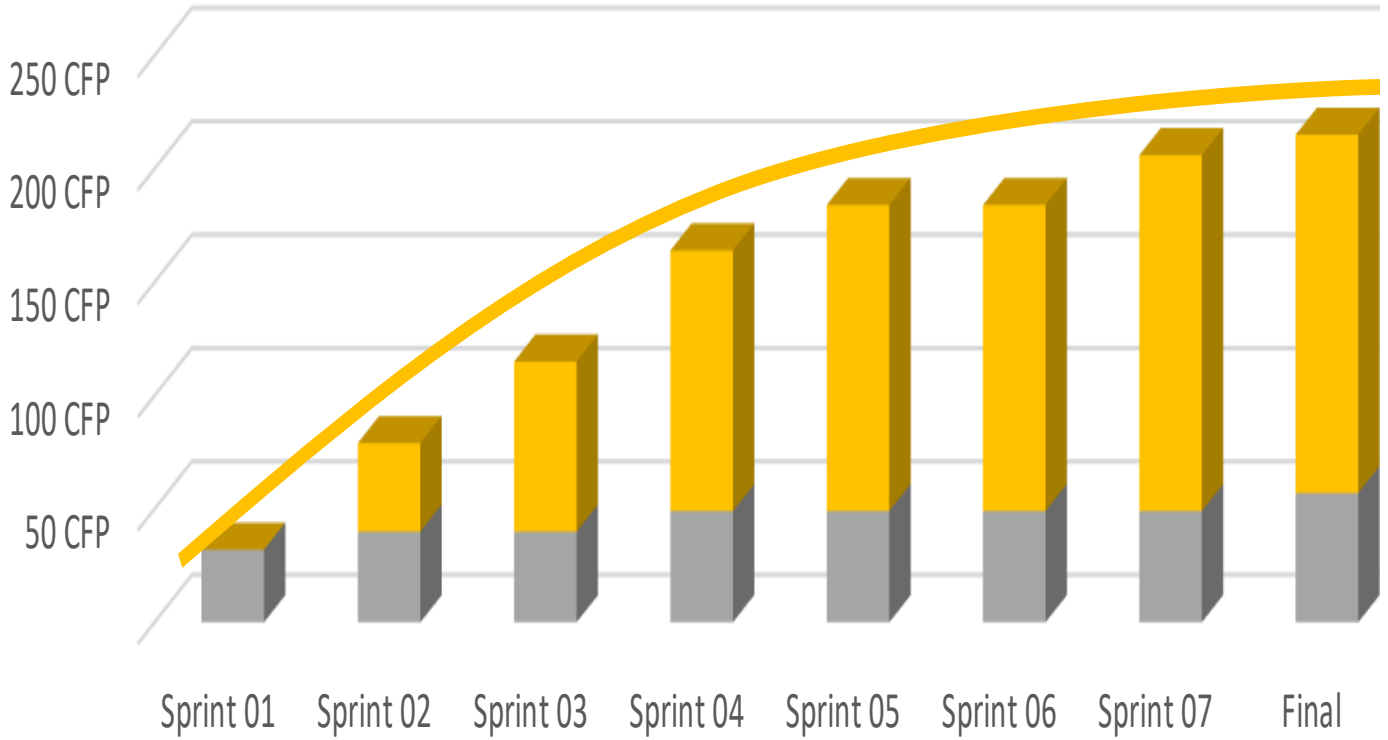
- IFPUG
  - More bias, more Function Points
  - Same curve



Agenda

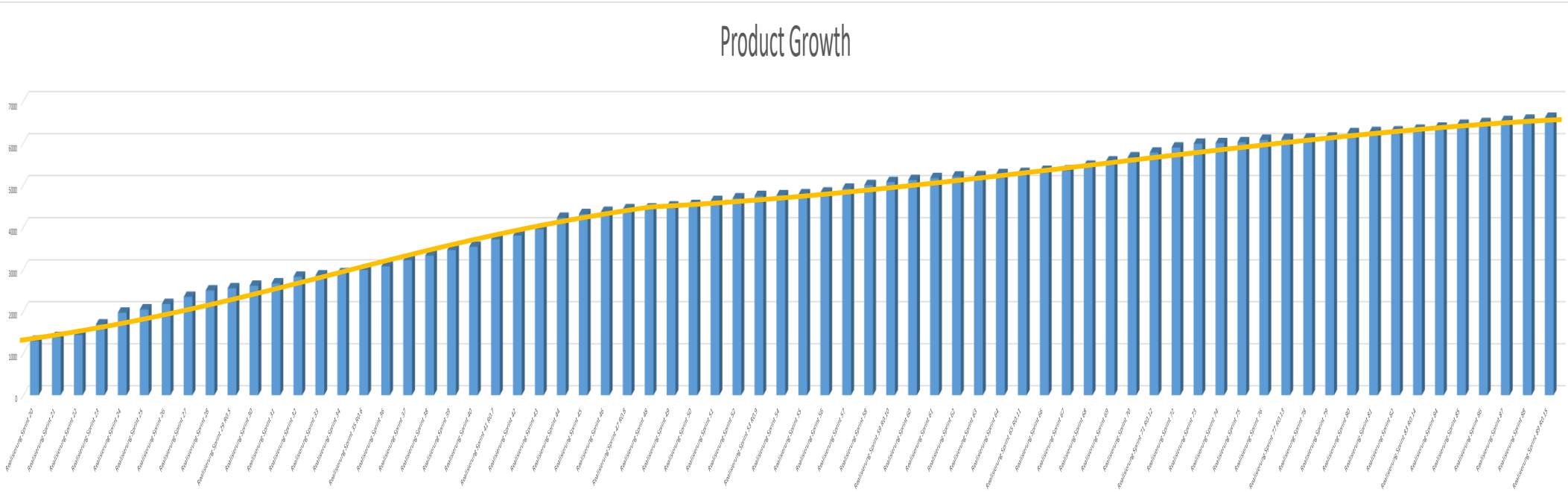
The Fractal Growth Curve for Functionality

- It looks like a fractal growth curve
  - Adding functionality incrementally
  - Increments get smaller and smaller



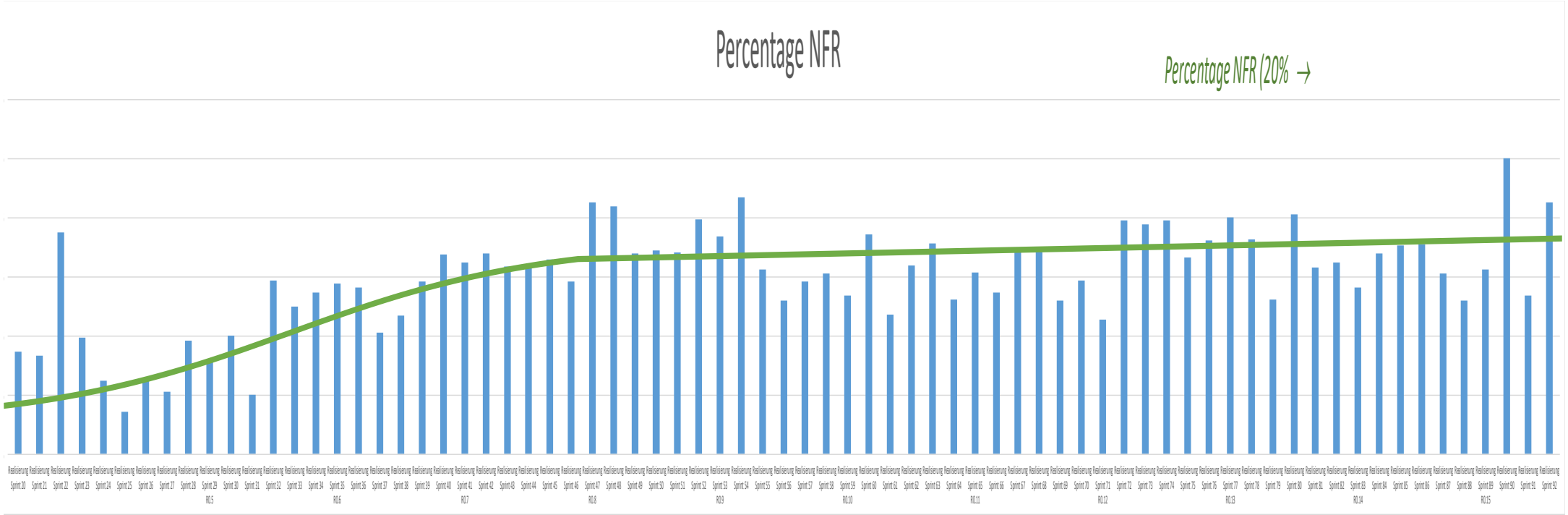
A Real Development Project with 90 Sprints

- 90 Sprints over two years exhibit a similar pattern
  - ➡ Logarithmic curve seems a good approximation for fractal growth curve



A Real Development Project with 90 Sprints

- The number of NFR User Stories increases compared to functional User Stories
  - However, the percentage stabilizes around ~ 70% NFR
  - Less predictable with more variations and some outliers



What causes Fractals?

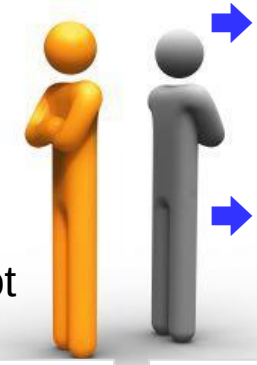
- Where fractality eventually originates
  - ➡ In later sprints, refactoring becomes more frequent adding no or few new functionality to the product
    - Typically, the user stories (requirements) are refined as well, add some, but not much new functionality to the product
  - ➡ Enhancing functionality does not let the product grow
    - However, refined, or new, user stories sometimes add new functionality, such as adding a transaction log
  - ➡ Removing Technical Debt does not add any functionality
    - Neither does fixing bugs – except the bug was in the user story
- The development team wants to deliver a viable product fast, at the beginning
  - ➡ Especially true for DevOps

Are there Counterexamples?

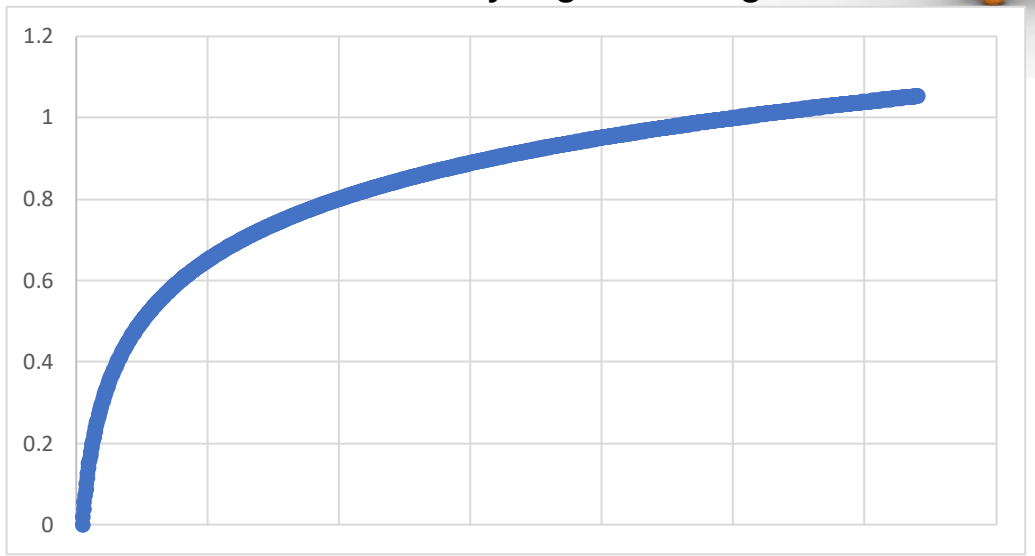
- It is not difficult to find counterexamples
  - ➡ Development sometimes focuses on new technology first before rolling all out
    - Then, the functional size of the product initially is kept small before the big bang
  - ➡ Products can shrink over time
    - Common observation in DevOps
    - Development teams initially had delivered functionality that proved to be useless later, or became obsolete
  - ➡ Solution domain can radically change during development
    - Functionality implemented can become completely obsolete
- While the fractal growth curve looks interesting for most agile development, it is certainly not some sort of natural law
  - ➡ What are the conditions that make agile development fractal?
  - ➡ Monitoring sprints is the only way to find out

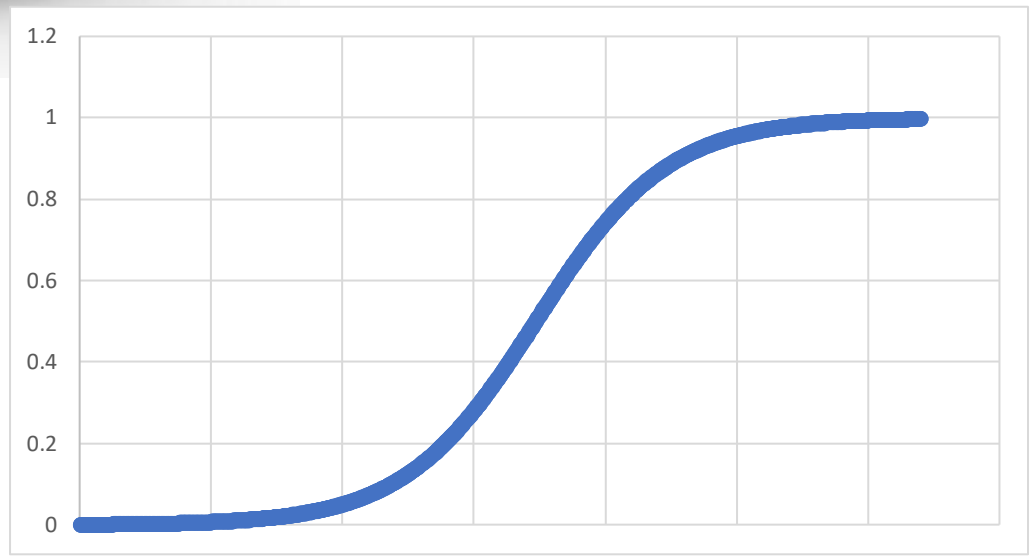
Two Hypothesis how to Model the Fractal Growth Curve

- The Fractal Growth Curve is best modelled by logarithm
  - ➡ Pro:
    - Limits fractal growth
    - Growth never terminates
  - ➡ Contra:
    - Product support ends and this cannot be modelled by logarithmic growth



- The Fractal Growth Curve is best modelled as a logistic function
  - ➡ Pro:
    - You cannot add functionality forever
    - products can lose functionality over time
  - ➡ Contra:
    - Under some conditions, the function exhibits chaotic behavior after some time





Agenda

The Gedankenexperiment

The Findings

The Fractal Curve

Conclusion

Findings from the Gedankenexperiment

- Management should monitor the characteristics of the fractal growth curve
- We know what to measure when, and how. We need both functional size and story points
- We prefer the subjective team measure captured by story points over effort measurements by counting hours
- Measuring agile development must address each sprint, not just the final product, or an initial vision
- Comparing the size of the product with the effort spent in sprints indicates how much work was spent in getting the requirements right
- While the product can be compared with the vision in terms of size, the implemented features might differ
- The question whether IFPUG or COSMIC shall be used for measuring agile depends on the product domain
  - ➡ Both methods work for managing development despite the lack of VIM/GUM compliance of IFPUG.

Conclusions

- Agile development must be monitored sprint by sprint
  - The aim of the sprints – here expressed by classical musical terms – is of essence
  - Understand team efforts
  - Monitor product behavior
- Agile metrics must include functional size
  - Functional size complements effort and code quality metrics
  - Testing cyber-physical systems is difficult without functional size metrics
    - Consider **Autonomous Real-time Testing**
- A final conjecture
  - The difference between the functional size of the product and total size of all sprints, plus the amount of enhancement works, might reflect the effort needed to find the correct requirements by the agile team

IARIA

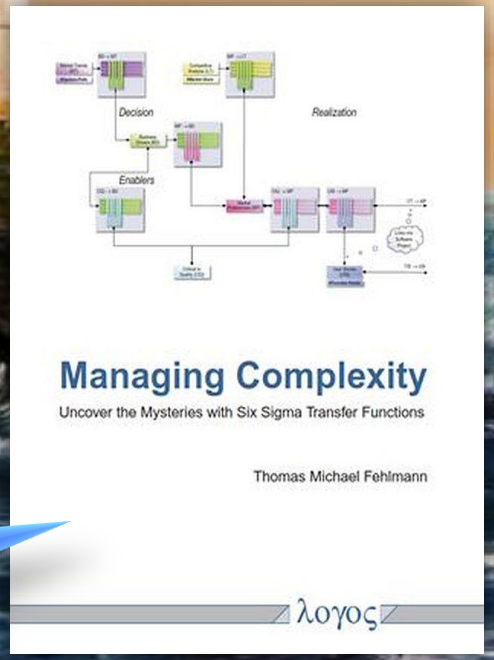
euro project office
Master your project.

Write e-Mail to
thomas.fehlmann@e-p-o.com
or connect to www.linkedin.com/in/thfehlmann

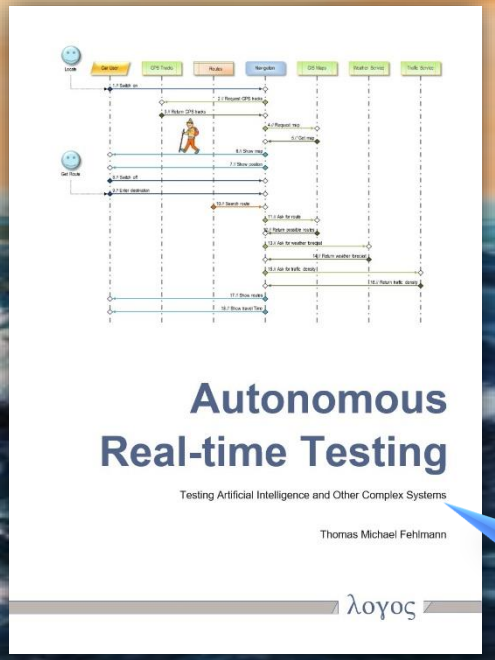
Questions?

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

www.logos-verlag.com
Search for "Managing" and "Testing"



Logos Press
Berlin 2016



Logos Press
Berlin 2020