Customer Orientation

Lean Six Sigma

Agile Processes

Project Estimations

Transfer Functions

Monitoring Agile Development

Distinguishing Bugs, Changed Requirements, and Technical Debt Removal

Thomas M. Fehlmann, Zürich
Euro Project Office AG

E: info@e-p-o.com

H: www.e-p-o.com

ISBSG
IT Confidence 2023

euro project office
Master your project.

ISBSG
serving the IT industry since 1997

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
- 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
- 2020: Autonomous Real-time Testing for 4th Generation Products
- 2022: ASQF Specialty Group Leader Switzerland

Customer Orientation

Lean Six Sigma

Agile Processes

Project Estimations

Transfer Functions

Goals of this Presentation

1) *Agile Development benefits from continuous monitoring sprint by sprint with function points*

2) *Requirements change continuously affecting the scope of development*

3) *Removing Technical Debt adds value*

Agenda

Monitoring Sprints

Technical Debt

Conclusion

Agenda

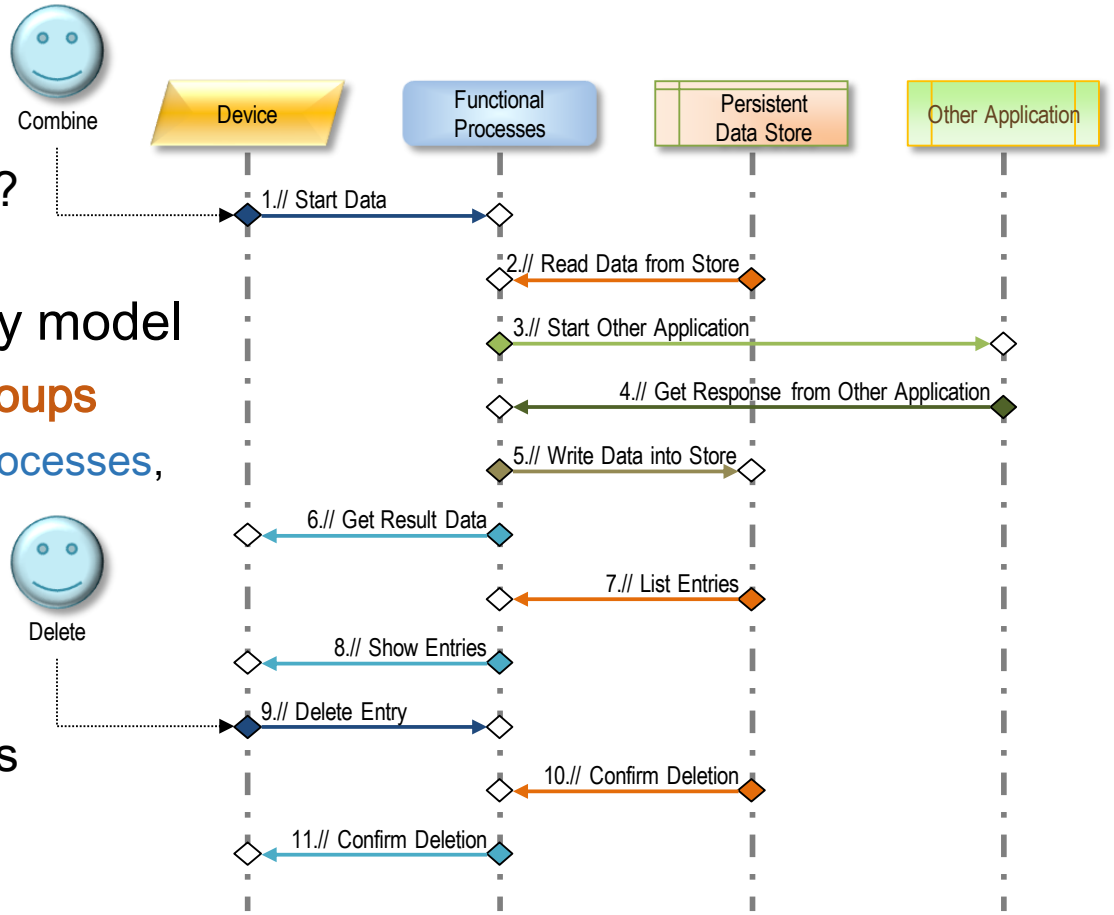
Monitoring Sprints

Technical Debt

Conclusion

Measuring Functionality with ISO 19761 COSMIC

- ISO/IEC 14143
  - What is a model for software functionality?
- ISO/IEC 19761 COSMIC is a functionality model
  - A Data Movement Map identifies Data Groups
    - Data moved between Devices, Functional Processes, Persistent Stores, and Other Applications
    - Triggered by external events
  - Useful for communications, embedded SW, IoT, cyber-physical systems such as medical instruments and autonomous cars



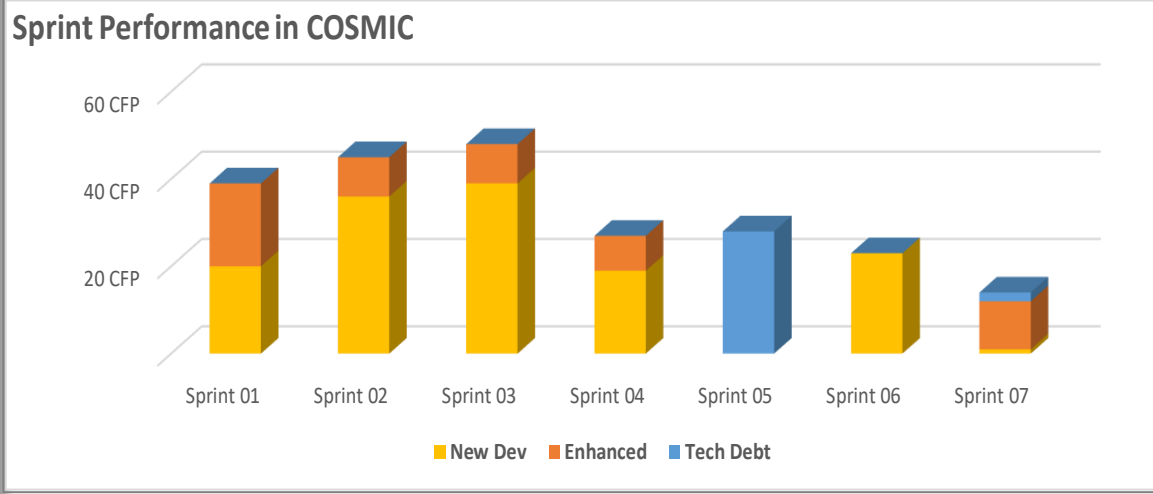
- Size = Number of Data Movements
  - When you can measure, you can use Six Sigma techniques to provide the right functionality, and provide it right

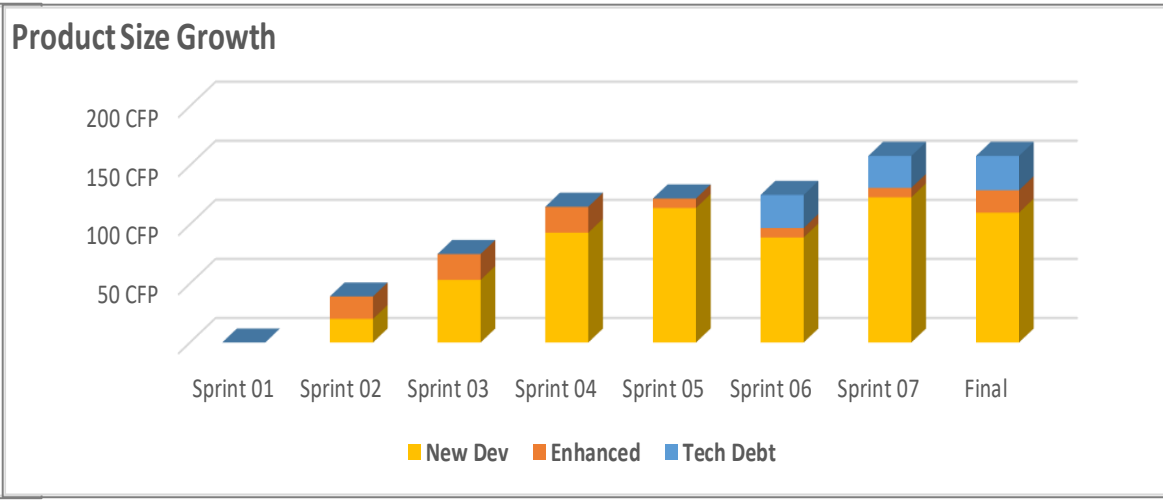
Seven Sprints

- Sprint 01 – Allegro
  - Two user stories from backlog
  - Two new user stories
- Sprint 02 – Andante
  - Three user stories from backlog
  - One new functional 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 non-functional user story
- Sprint 06 – Menuetto
  - Three new user stories
- Sprint 07 – Finale
  - One new user story
  - Final Tests

Sprint Performance & Product Growth

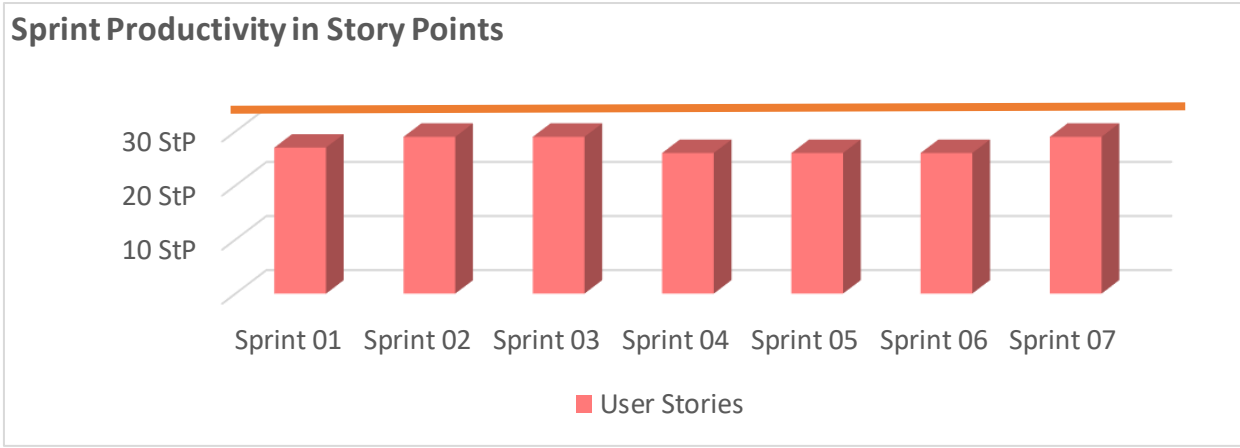
- Most new functionality implemented in the first three sprints
  - Sprint 04: Marche Funèbre – Fixes
  - Sprint 05: Intermezzo – TD Removal
- Product Size Growth
  - Not the sum of sprints!
  - Growth decreases
  - Fractal growth curve





Retrospective

- Seven (7) out of eight (8) user stories from initial 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

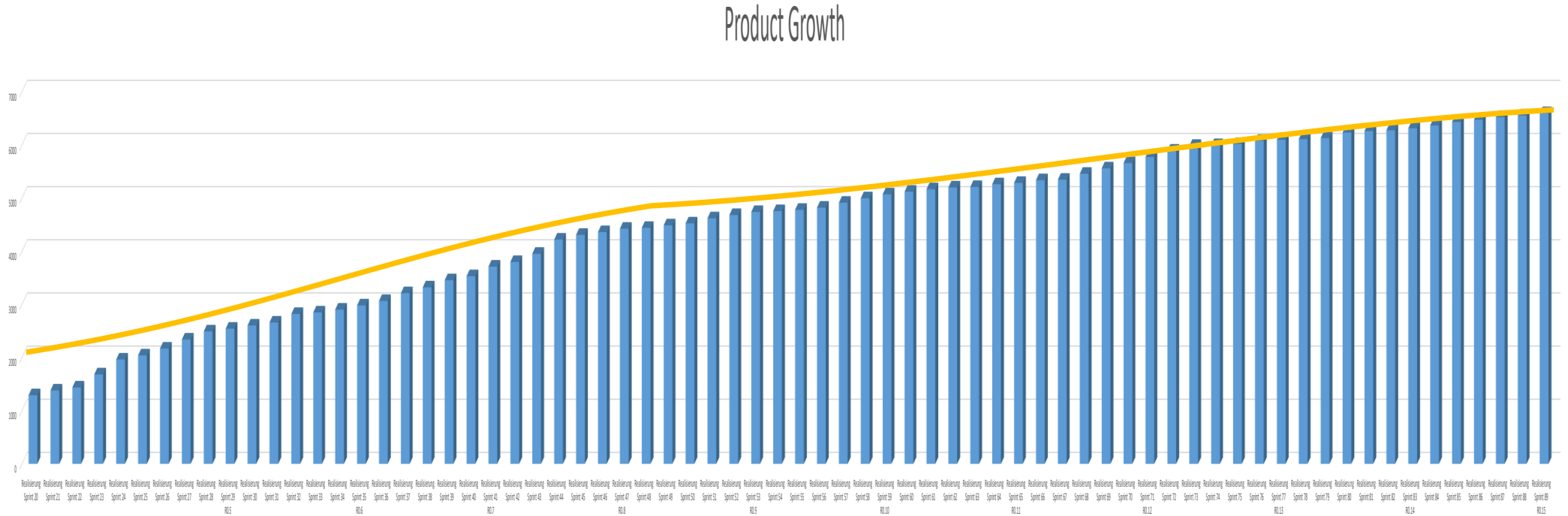
Total	*#Sprints*	*Length*	*Hours/Day*	*Team Size*	*Velocity*
192 StP	7	7 Days	7.0 h	7	**27 StP/Sprint**

Productivity in COSMIC FP

Total	*#Sprints*	*Length*	*Hours/Day*	*Team Size*	*Functional Velocity*
224 CFP	7	7 Days	7.0 h	7	**32 CFP/Sprint**

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

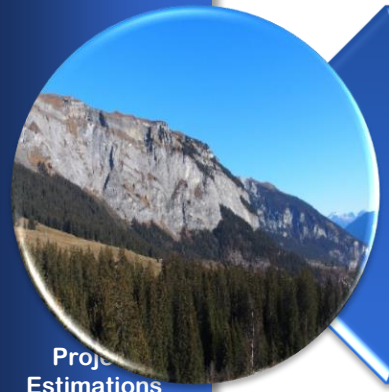


What Causes the "Decrease of Functional Productivity"?

- Why managers get software development wrong
  - 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, changing some, but not adding 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
    - But it adds value for the customer!
- The development team wants to deliver a viable product fast
  - Especially true for DevOps
  - Customer should agree to pay for removing Technical Debt

Agenda

Monitoring Sprints



Technical Debt

Conclusion

The Definition of Done Allows no Bugs in Code (Unit Testing)

- User Story:
  - As a certain user
  - I want something being done
  - Such that I get a sound response
  - Because I want safety & security

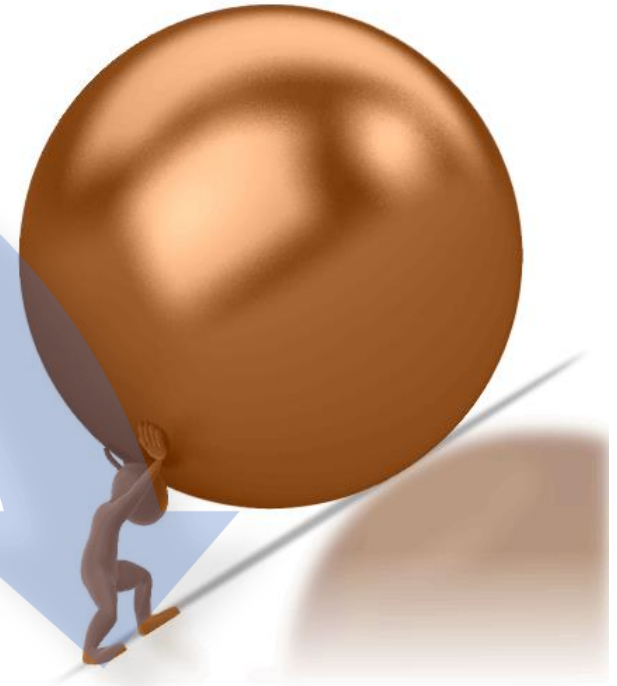
Refinement Meeting

- Acceptance Criteria:
  - Test Case 1: It's done indeed
  - Test Case 2: It's sound response
  - …
  - Considering future integration into some system

No Bugs left in released Code!

Technical Debt and its Fellow Buddies

- **Technical Debt**
  - Happened-upon technical debt – the development team was unaware it existed
  - Known technical debt – has been made visible using one of many approaches
  - Targeted technical debt – known debt on purpose servicing by the development team



- **Code Smell**
  - Any characteristic in the source code of a program that possibly indicates a deeper problem
  - Measurable by code scanners
    - Cit. Kent Beck

- **Big Ball of Mud**
  - A software system that lacks a perceivable architecture
  - A haphazardly structured, sprawling, sloppy, spaghetti-code, Goofy-tape-and-baling-wire
    - Cit. Brian Foote & Joseph Yoder

Technical Debt Reporting

- **Technical Debt**
  - Happened-upon technical debt – the development team was unaware it existed, but it became apparent during development
  - Known technical debt – has been made visible using one of many approaches
  - Targeted technical debt – purposefully accumulated by the development team

- Development teams are aware of technical debt, and they can report it
  - Removal of technical debt creates value for the user
  - Technical Debt should be measured, and removal should be a billable effort
    - In case a customer doesn't agree to pay for removal, it's his decision
  - In a contract you can agree on Code Smell metrics that must be met and a maximum amount of related technical debt
    - Requires a customer with technical expertise

New Requirements and Technical Debt Reporting

Data Movements

	Name	Label	Data Movement Sub-Process Description	Trigger		Action	Development Type
32)	E032	Start Scan	Start scanning	T06	Scan Cod	Add	New Development
33)	X027	Request Scanner Page	From CMS			Add	Enhancement
34)	E030	Provide Scanner Page	For displaying Scanner details			Add	Enhancement
35)	W007	Save Settings	Temporarily save camera settings			Add	Enhancement
36)	E031	Stop Posting	Stop the QR Scanner posting images			Add	Enhancement
37)	E011	Scan QR	Use Photo to scan QR code			Change	Enhancement
38)	R023	Get Settings	Get posting settings back but keep data for use outside the Moble			Add	New Development
39)	X029	Restore Settings	Restore previous settings for image posting			Add	New Development
40)	R005	Valid Giro Account	Get validated Giro Account			Add	New Development
41)	X012	Warning: Transaction not covere	Transaction pending; waiting for coverage			Add	New Development
42)	W002	Prepare Transaction	Prepare transaction for future execution			Add	Re-Development
43)	X011	Confirm Transaction	Confirm transaction scheduled for execution			Add	Re-Development
44)	E035	Transaction Confirmed	Confirmed transaction as scan result			Add	Re-Development
45)	W013	Record Transaction	Record everything done in this transaction such that it can be reve			Add	Re-Development

New Requirements pop up during Sprints that affect already implemented functionality

Bug Fixes are Not Counted

Technical Debt gets removed by Refactoring (Add, Change or Delete)

Software Development Contracting and Billing

- Assume a contract fixing the following performance goals

	New Development	Enhancement	Re-Development
Add:	11.0 PDR	9.1 PDR	24.4 PDR
Change:	22.0 PDR	11.3 PDR	36.0 PDR
Delete:	17.2 PDR	4.5 PDR	12.5 PDR

- Assuming that one hour of development time is remunerated with 50 EUR
  - New Development: Adding 2 CFP requires 2*11.0 = 22.0 Hours
  - Enhancement: Adding 5 and changing 6 CFP requires 5*9.1+6*11.3 = 113.3 Hours
  - Re-Development: Changing 2 CFP requires 3*36.0 = 72.0 Hours
- In total you can bill for Sprint 07: 50 * (22+113.3+72) = € 10'365.00
  - No need to collect time sheets

PDR = Product Delivery Rate (h/CFP)

COSMIC Function Point Count – Sprint 07 as Example

COSMIC Function Points Count

euro project office
Master your project.

Functional Process		Action	E	X	R	W	CFP Count		New Development		Enhancement		Re-Development		Total Hours
								Add:	11.0 PDR		9.1 PDR		24.4 PDR		
								Change:	22.0 PDR		11.3 PDR		36.0 PDR		Performance
								Delete:	17.2 PDR		4.5 PDR		12.5 PDR		€
F004	Identify & Authenticate	Add						0%		0%		0%		1	
F004		Change						0%		0%		0%		2	
F004		Delete						0%		0%		0%		3	
F006	Graphics	Add	2	2	3		7 CFP	29%	2 CFP	71%	5 CFP	0%		1	67.5 Hours
F006		Change	1	2	5	1	9 CFP	0%		67%	6 CFP	33%	3 CFP	2	175.8 Hours
F006		Delete						0%		0%		0%		3	
Total			3	4	8	1	**16 CFP**	13%	2 CFP	69%	11 CFP	19%	3 CFP		**243.3 Hours**

- The PDR for Adding, Changing, or Deleting Actions are different
  - ➡ Different for New Development, Enhancement, and Re-Development
  - ➡ But well known thanks to ISBSG's Repository for a suitable sample sprint (or project)

- They fit well into today's agile development culture
  - ➡ These rates can be used in contracts for remunerating Technical Debt Removal
  - ➡ Technical Debt Removal is reported as Re-Development

COSMIC Function Point Count – Sprint 07 as Example

COSMIC Function Points Count

euro project office
Master your project.

Functional Process		Action	E	X	R	W	CFP Count		New Development		Enhancement		Re-Development		Total Hours
								Add:	€ 550.00 / CFP		€ 680.00 / CFP		€ 350.00 / CFP		
								Change:			€ 800.00 / CFP		€ 350.00 / CFP		Currency
								Delete:			€ 80.00 / CFP		€ 50.00 / CFP		€
F004	Identify & Authenticate	Add						0%		0%		0%		1	
F004		Change						0%		0%		0%		2	
F004		Delete						0%		0%		0%		3	
F006	Graphics	Add	2	2	3		7 CFP	29%	2 CFP	71%	5 CFP	0%		1	€ 4'500.00
F006		Change	1	2	5	1	9 CFP	0%		67%	6 CFP	33%	3 CFP	2	€ 5'850.00
F006		Delete						0%		0%		0%		3	
Total			3	4	8	1	**16 CFP**	13%	2 CFP	69%	11 CFP	19%	3 CFP		**€ 10'350.00**

- If customers prefer to pay a fixed amount per function point
  - Easier for purchasing departments that do not know the ISBSG repository
  - You can select the currency as you want
  - it's a toggle switching between PDR and Currency
- Technical Debt Removal thus becomes explicit in a contract

ISBSG
serving the IT industry since 1997

euro project office
Master your project.

Agenda

Customer Orientation

Lean Six Sigma

Agile Processes

Project Estimations

Monitoring Sprints

Technical Debt

Conclusion

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 the age of the user stories selected
    - If only recent user stories get selected, new product development is entering DevOps mode
- 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
- Technical Debt Removal should become explicit in a contract
  - If the customer agrees to pay for higher quality
  - Thanks to the FSM methods, measuring Technical Debt in Agile is simple
    - May not be suitable for traditional fixed-price contracts

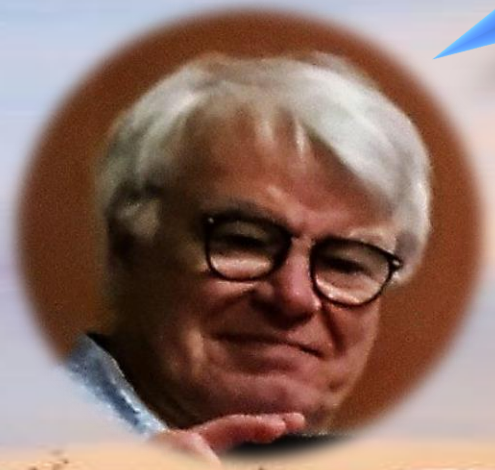
ISBSG
serving the IT industry since 1997

euro project office
Master your project.

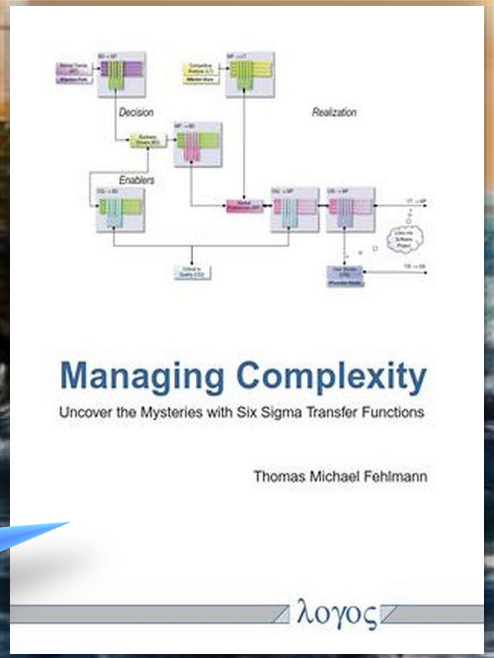
Questions?

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

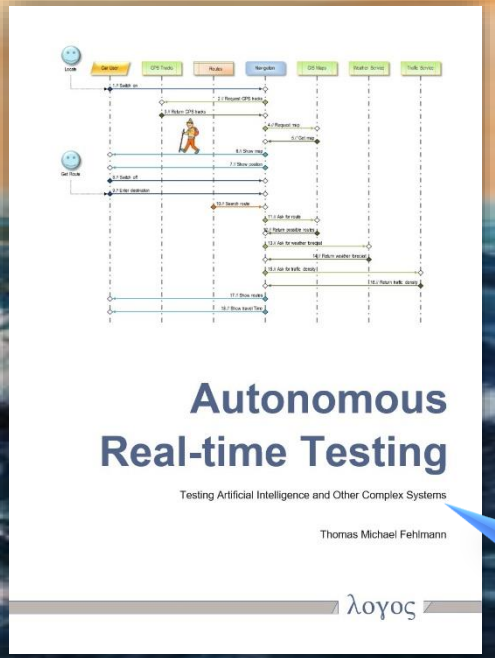
- Customer Orientation
- Lean Six Sigma
- Agile Processes
- Project Estimations
- Transfer Functions



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



Logos Press Berlin 2016



Logos Press Berlin 2020