Output-based contracting in an Agile environment

Source: https://maritasoverpeinzingen.wordpress.com/tag/scrum/

by Nico Oosterom Product Owner at Glencore Agriculture &
Richard Sweer Project Manager at Finidy

version 8 October 2019 – v1.1

GLENCORE AGRICULTURE

IWSM MENSURA

The future of IT Cost Estimation
Trends for the new Decade

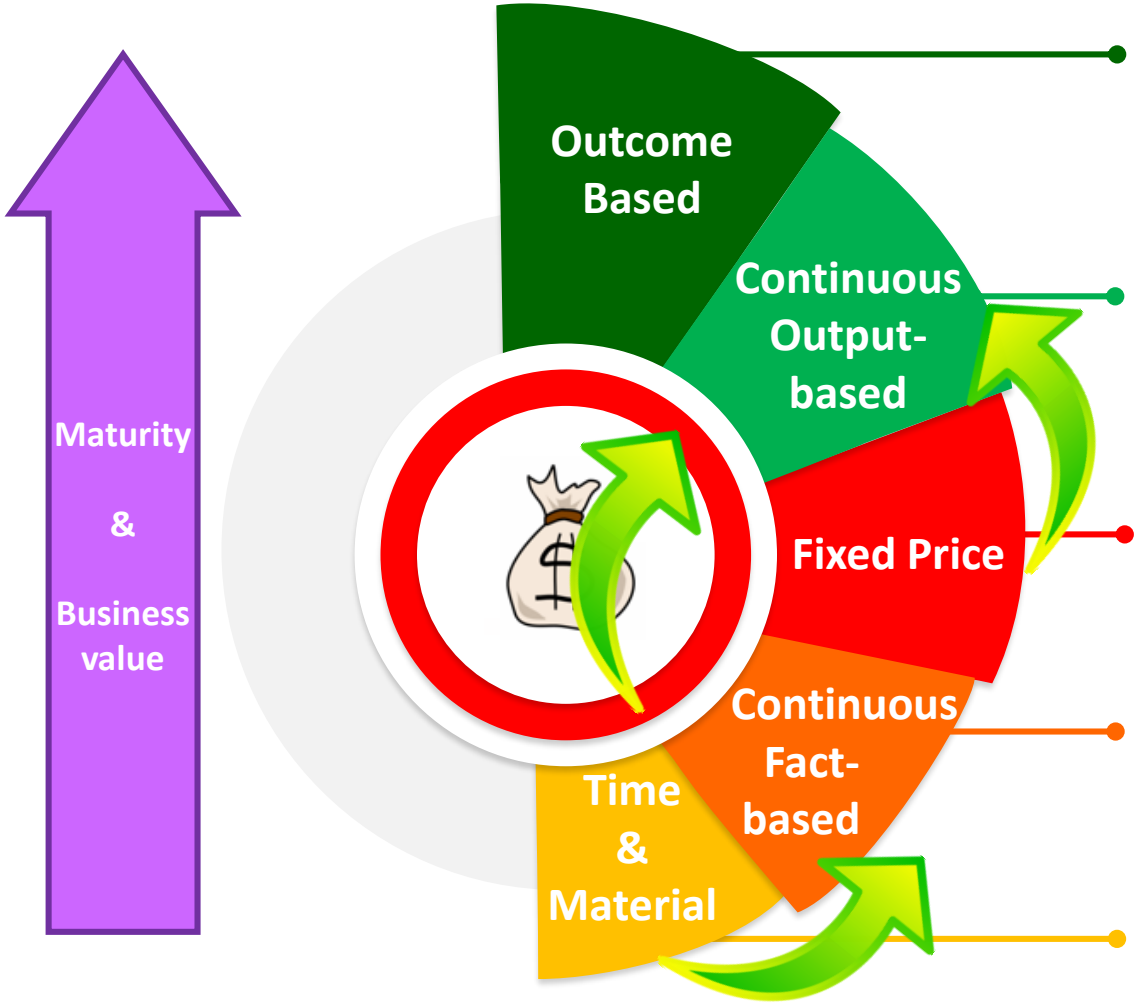
Finidy if software really matters

Agenda

- Agile and commercial delivery models
- Core metrics – even for every Agile project/value streams
- The model – 4 KPIs with at least 35 underlying metrics

- Glencore Agriculture and our challenge
- Global contour of output-based contract
- How to monitor and control
- Our experiences with an output-based contract

Intro - Agile needs different commercial delivery models!



- Payment is tied to % of business benefits (reward sharing)

- Supplier owns delivery and is paid per output delivered (e.g. Function Points)
- Supplier is not paid if the quality requirements are not met (Definition of Done)
- Higher productivity means more throughput and more profit (win-win scenario)
- Most of risk with supplier

- Fixed specification
- Fixed price and mostly fixed date
- Changes with a fee (mostly more than expected)
- Risk with supplier and often also with customer (changes)

- Grip and control on costs, duration and quality by using metrics
- Budget and duration risks with customer
- Lower productivity and quality means higher revenue for supplier

- Traditional hour tariff based
- Lower productivity and quality means higher revenue for supplier
- Many vendors ignore productivity and quality
- Most of the risk with customer, no control due to lack of metrics

Core metrics – Productivity, Cost effectiveness and Quality

- **Productivity rates (Faster)**

$$\frac{\text{Hours of effort}}{\text{functional size/scope (e.g. Function Points) of delivered software}}$$

- **Cost effectiveness (Cheaper)**

$$\frac{\text{Project Euro cost}}{\text{functional size/scope (e.g. Function Points) of delivered software}}$$

- **Product quality (Better)**

$$\frac{\text{Defects}}{\text{functional size/scope (e.g. Function Points) of delivered software}}$$

Core metrics – Size is a significant factor for effort and duration

1. **Project size/scope (Lines of code, Function Points, etc.)**
   - Effort for 1.000 FP project ~ 8 man-year[1)]
   - Effort for 10.000 FP project ~ 200 man-year [1)]

   Man-year increase of 25 times!

2. **Kind of software (factor 30-40)**
   - Nuclear power plant, air traffic system, bank system, etc.

3. **Effectiveness of individuals of team (factor 10-20)**

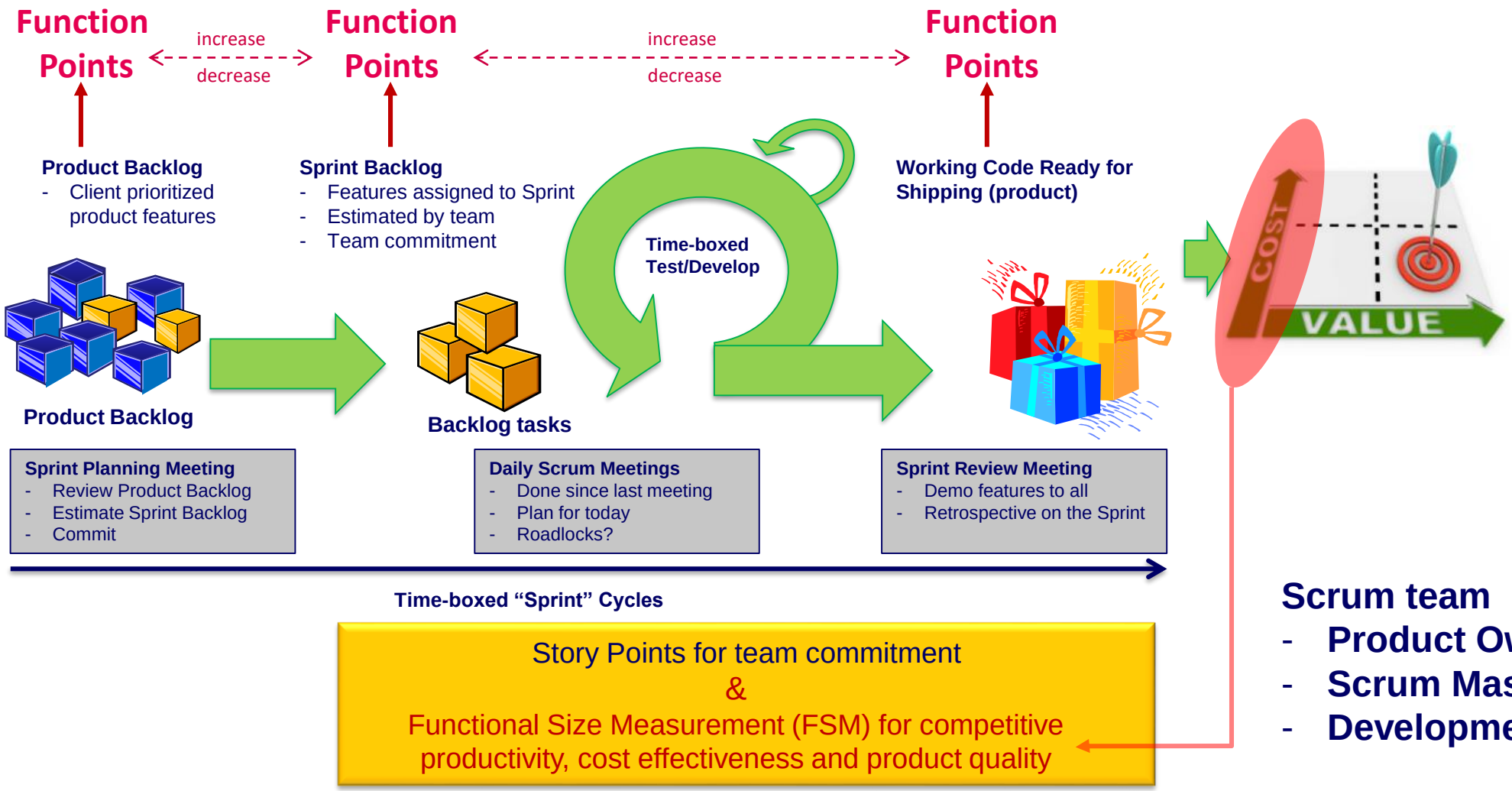
4. **Programming languages (factor 2-10)**
   - .NET, Java, Mendix, OutSystems, Thinkwise, Codeless, Angular, Polymer, Oracle, SQL Server, etc.

[1)] Prof. dr. C (Chris) Verhoef, Science of Computer Programming, volume 45, number 1, October 2002

Core metrics - Choose the right functional size metric

- **Functional Size Measurement (FSM)**
  - ISO/IEC 14143-1:2006 Functional Size Measurements
- **Functional Size (FS)**
  - A size of the software derived from quantifying the Functional User Requirements
- **Nesma: High level counting is the 'standard'**
- Examples
  - **ISO/IEC 24570:2018 NESMA (version 2.3)**
  - **NESMA for software enhancement (version 2.3)**
  - ISO/IEC 29881:2010 FiSMA (version 1.1)
  - ISO/IEC 20269:2009 IFPUG (version 4.3.1)
  - ISO/IEC 19761:2017 COSMIC FFP (version 4.0.2)
  - Interface points (Finidy)

Core metrics – Story- and Function Points can both be used



Story Points for team commitment
&
Functional Size Measurement (FSM) for competitive productivity, cost effectiveness and product quality

Scrum team
- **Product Owner**
- **Scrum Master**
- **Development team**

The model – 4 KPIs with at least 35 underlying metrics

- **USP: increase success rate of a software IT project**
  - Combination of at least 35 metrics which makes it possible to increase the predictability of costs, turnaround time and quality of IT projects (Agile/Scrum, Kanban or Waterfall)
  - These metrics are divided into four areas
    - Better – Quality
    - Faster – Time to market
    - Happier – Satisfaction
    - Cheaper – Productivity

APPROVED BY

"Cognizant is one of the world's leading professional services companies, transforming clients' business, operating and technology models for the digital era. Cognizant helps customer focus on their core and provides value by executing projects tied to output/outcome to clients' business needs. As an example, Cognizant is helping a customer in The Netherlands to modernize their landscape by executing the program in Output-based Agile/Iterative model linked to Function Points delivered (based on NESMA standards)"

Saket Gulati – Head of Markets, The Netherlands (Cognizant).

Cognizant

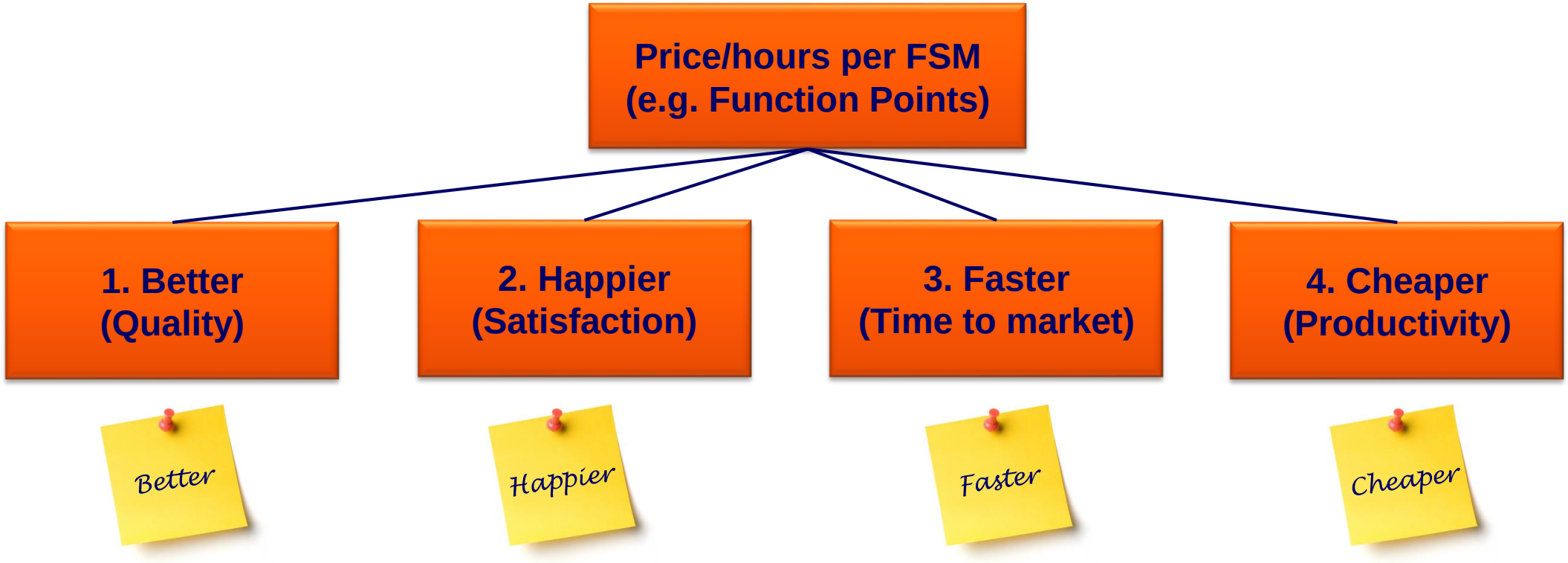
- **Supports two types of 'appearances'**
  - Continuous Fact-based (calibrate the values of the KPIs/metrics)
    - 4 KPIs with at least 35 underlying metrics - paying Time and Materials
  - Continuous Output-based (basis for Output-based contracting)
    - 4 KPIs with at least 35 underlying metrics - paying price per **FSM (e.g. Function Points)**

The model - Metrics must be transparent and create a win-win

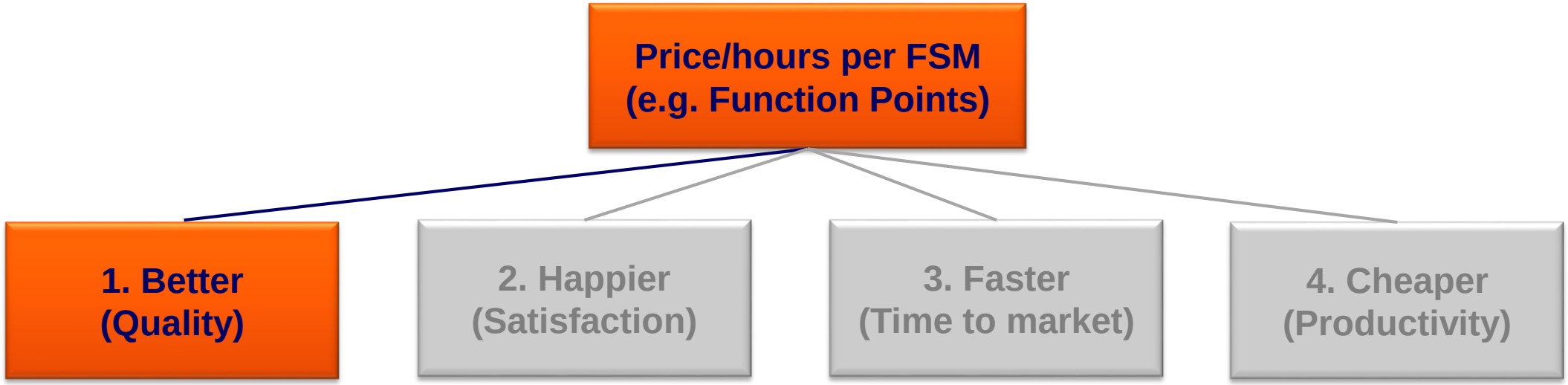
&

Calculation rules and guidelines of all used metrics/KPIs must be transparent (and published)

The model - Price per Function Point based on ± 35 metrics

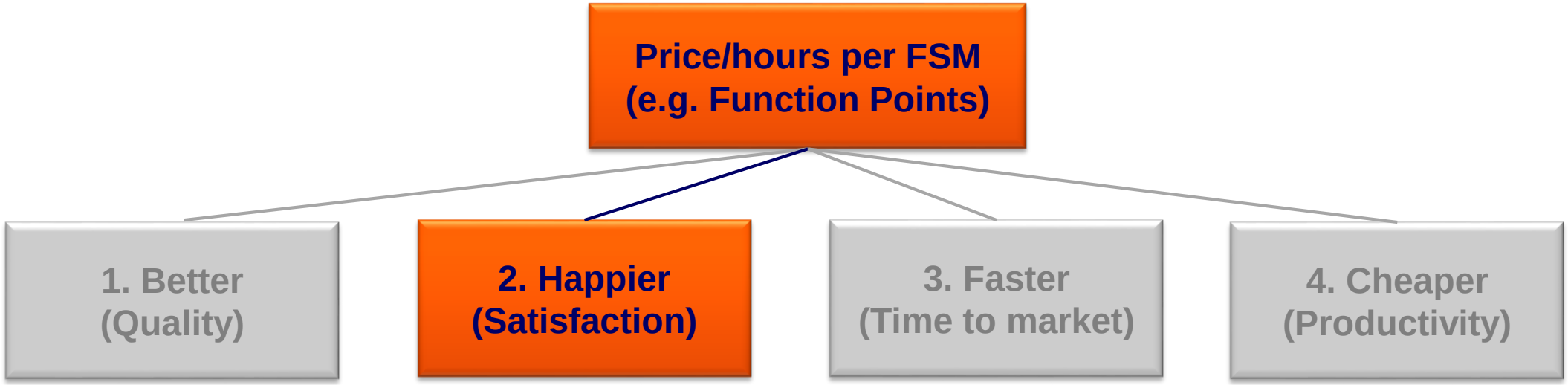


The model - Some examples of metrics



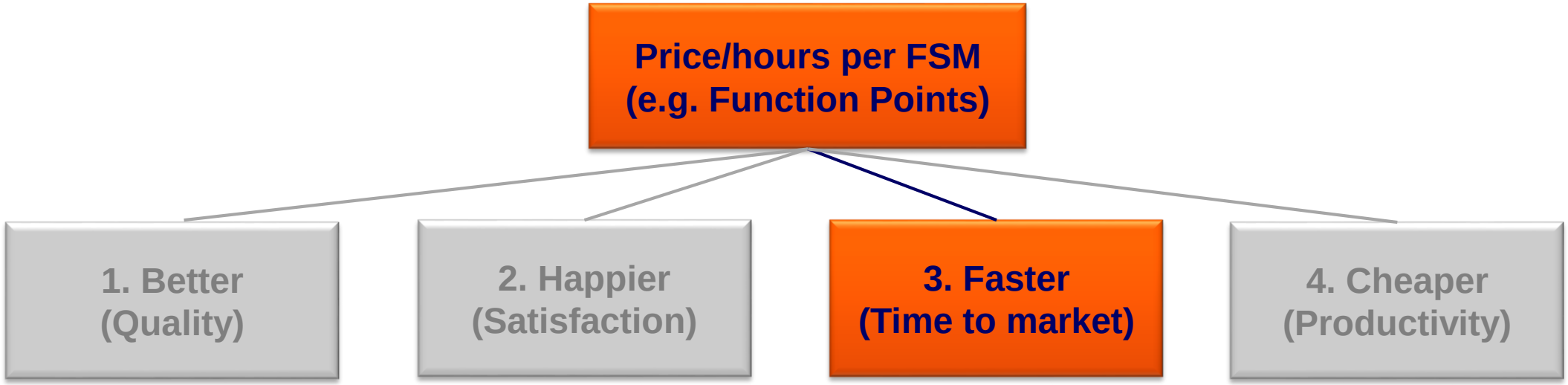
- Static source code quality (>300 sub-metrics by using tools)
- Software architecture quality/technical debt (by manual review)
- Functional code and decision coverage (measured by a tool) or per Function Point
- Defect removal efficiency metrics (measured per Function Point)
- Maximum amounts of ‘open’ defects per severity code in production per Function Point
- Etc.

The model - Some examples of metrics



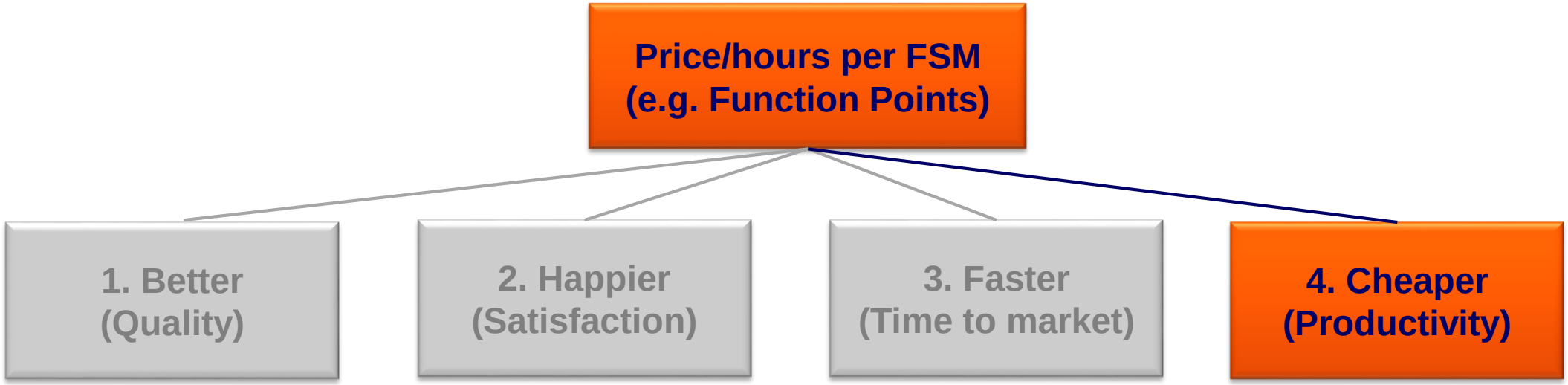
- (Key)-user satisfaction (demo and after/every x period)
- Development team satisfaction
- Product Owner satisfaction (demo and after/every x period)
- Stakeholders satisfaction after/every x period
- Etc.

The model - Some examples of metrics



- Response and resolution time of defects per severity code in acceptance and production environment (MTTR and MTBF)
- Amounts of function points to deliver per sprint with one team
- Amounts of function points to deliver per sprint with <n> teams
- Etc.

The model - Some examples of metrics



- Price per function point with one team
- Price per function point with <n> teams
- Price per function points after implementing all the generators/generic components
- Price per function point for module x
- Etc.

The model – Start simple with a Excel sheet

Dashboard value stream/project xyz

Better - code quality	Target	Where	Average	Sprint 1	Sprint 2	Sprint 3	Sprint 4
TIOBE - Code Coverage	B	DoD	C	A	B	C	D
TIOBE - Abstract interpretations	C		B	A	D	B	A
TIOBE - Cyclomatic complexity	E		B	A	B	F	F
SonarQube - Bugs - reliability	B		A	C	C	B	A
SonarQube - Vulnerabilities - security	C		A	A	B	D	D
Better - Defect metrics	Target		Average	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Maximum defect with severity 1 in first iteration of UAT (defects/FP)	1	UAT	1,667	1	3	3	1
Maximum defects with severity 1 and 2 in Production (defects/FP)	2	Production	1,833	2	1	2	3
Defect with severity 1 and 2 in code before production move	0	DoD	0,5	0	1	1	1
Happier - Satisfaction metrics (value between 1 and 10 where 10 is most satisfied)	Target		Average	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Development team happiness (per sprint)	7	Sprint review	6,75	7,5	7,0	6,0	7,0
Development team collaboration (per sprint)	7	Sprint review	6,75	8,5	8,0	6,0	5,0
Product Owner satisfaction (per sprint)	7	Sprint review	7,92	7,5	7,0	8,0	8,0
Faster - through-put metrics	Target		Average	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Amount of FP per sprint	50	DoD	48	20	30	55	60
Minimum amount of Story Points/FP in stock	60	DoD	52,5	20	40	55	60
Faster - response and resolution	Target		Average	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Response & resolution time for defects severity 1 (during UAT in days)	2	UAT	2,0	2,0	2,0	2,0	2,0
Response & resolution time for defects severity 1 (during production in days)	1	Production	1,43	2,0	0,6	2,0	1,0
Response & resolution time for defects severity 2 (during production in days)	2	Production	1,5	2,0	3,0	1,0	1,0
Cheaper - productivity metrics (per sprint in hours/FP)	Target		Average	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Business analyst	4	DoD	4,7	6,0	5,0	5,0	3,5
Build (included unit tests)	10	DoD	8,3	7,0	8,0	10,0	8,0
Testing (view, API, E2E)	6	DoD	6,8	7,0	7,0	8,0	7,0
Total	20	DoD	19,8	20,0	20,0	23,0	18,5

GLENCORE
AGRICULTURE

Glencore Agriculture

Nico Oosterom

Agenda

- Introduction – what do we see in the market/problem statement
- Core metrics for every (agile) project and the role of Functional Size Measurement
- The model – 4 KPIs and 35-40 underlying metrics

- Glencore Agriculture and our challenge
- Global structure & content of our output-based contract
- How we monitor and control
- Our experiences with an output-based contract

Glencore Agriculture - who are we

Glencore Agriculture is a market leader in originating, handling, processing and marketing agricultural commodities, including grain, oilseeds, pulses, sugar, rice, cotton, vegetable oils, protein meals and biodiesel.

270+
Storage & handling facilities

17 countries

35+
Processing & Refining facilities

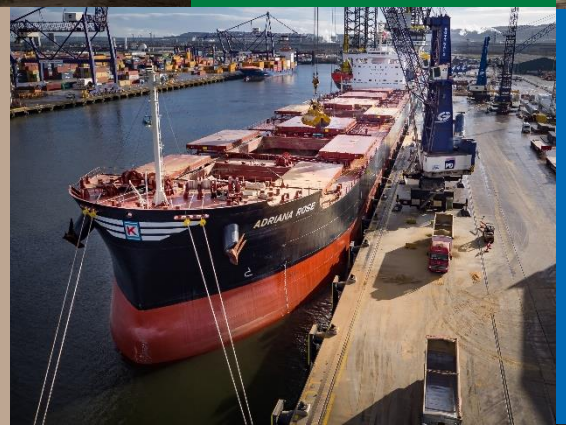
11 countries

13,000+ employees worldwide

23 port terminals

8 countries

33 marketing offices



180+
ocean-going Vessels

1,900+
owned or leased rail wagons

GLENCORE AGRICULTURE

IWSM MENSURA
The future of IT Cost Estimation
Trends for the new Decade

Finidy
if software really matters

Our challenge is to modernize 8.000 Function Points

- Glencore Agriculture has written its existing Trading and Traffic System (ERP system) in Powerbuilder (Client) and PL-SQL/Oracle (Backend).
  - Powerbuilder is set to be replaced by a different programming environment
- The existing system is complex and comprehensive.
  - Used by over 900 users in more than 16 countries, covering the process from contract entry to invoicing
  - Scope of the system determined by conducting a NESMA 2.3 Function Point Analysis.
    Online scope (excluding batches and interfaces) is approximately 13,000 Function Points.
- After a proper preliminary study, we have chosen for the combination of Oracle (DBMS), Microsoft .NET and TypeScript/Angular as the new default platform.
- Rebuild while the shop is still open:
  - Domain driven / Modular rebuild
  - Keep using single Oracle database

Determine what's in the price per FP – Deliverables & activities

Key success factors

- Describe all the 'system' deliverables (the output) – based on PRINCE2 template (step 1)
- Determine what's in the price per FP and what's not in the price per FP (step 2 & step 3)
  - For each product and for each activity/Scrum ceremony
- Agree on the 4 main KPI's (Better, faster etc.) and at least 35 underlying metrics

Product descriptions – 'system' deliverables
- ARC-01: Project Start Architecture
- BA-01: User Stories
- BA-02: Non-functional requirements
- BA-03: UX-design
- BUILD-01: Technical design
- BUILD-02: Unit test scripts
- **BUILD-03: Source code**
- BUILD-04: Release notes
- BUILD-05: Deployment packages
- TEST-01: Product risk analysis
- TEST-02: Component (API/VIEW)
- TEST-03: End-to-End
- TEST-04: Load & performance
- TEST-05: Release advice
- ….

Step 1

BUILD-03: Source code	
Objective	Collection of files containing codes written in a specific language (e.g. Typescript, C#, PL/SQL) that is 'translated' into a set of instructions that can executed by an execution environment (e.g. JavaScript Engine, .NET , Oracle).
Composition	Source code is distinguished based on technology • Frond end (typescript / HTML 5 / SCSS) • Back end (C#) • Database (Oracle)
Appearance and format	IDE (Visual Studio, PL/SQL Developer). Source code is stored in T repository.
Acceptance/ quality criteria	Meet the Definition of Done Tiobe TICS ratings: • Abstract Interpretation A • Cyclomatic complexity B • Compiler warnings C no level 1 • Code standards A • Code duplication C • Dead Code A
….	……

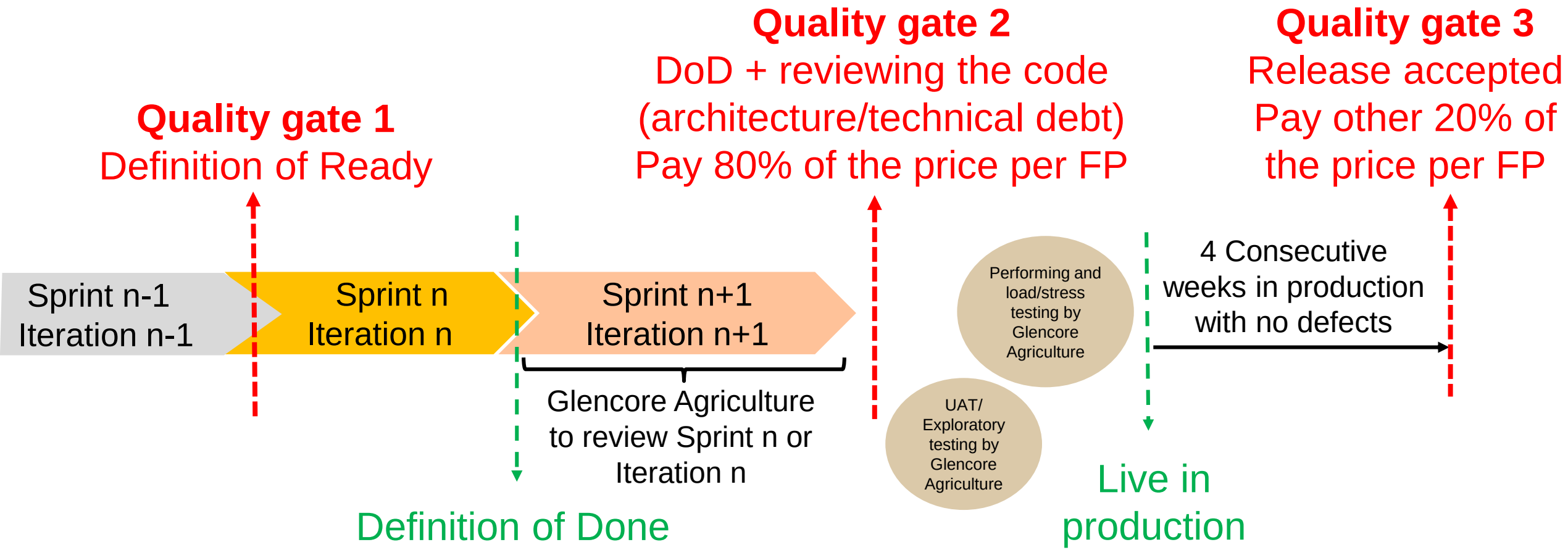
Step 2

'system' deliverables	Included in price/FP
ARC-01: Project Start Architecture	YES
ARC-02: Project Architecture viations	YES
A-01: Non-functional requirements	YES
BA-02: User Stories	YES
BA-03: UX design	NO
…	…

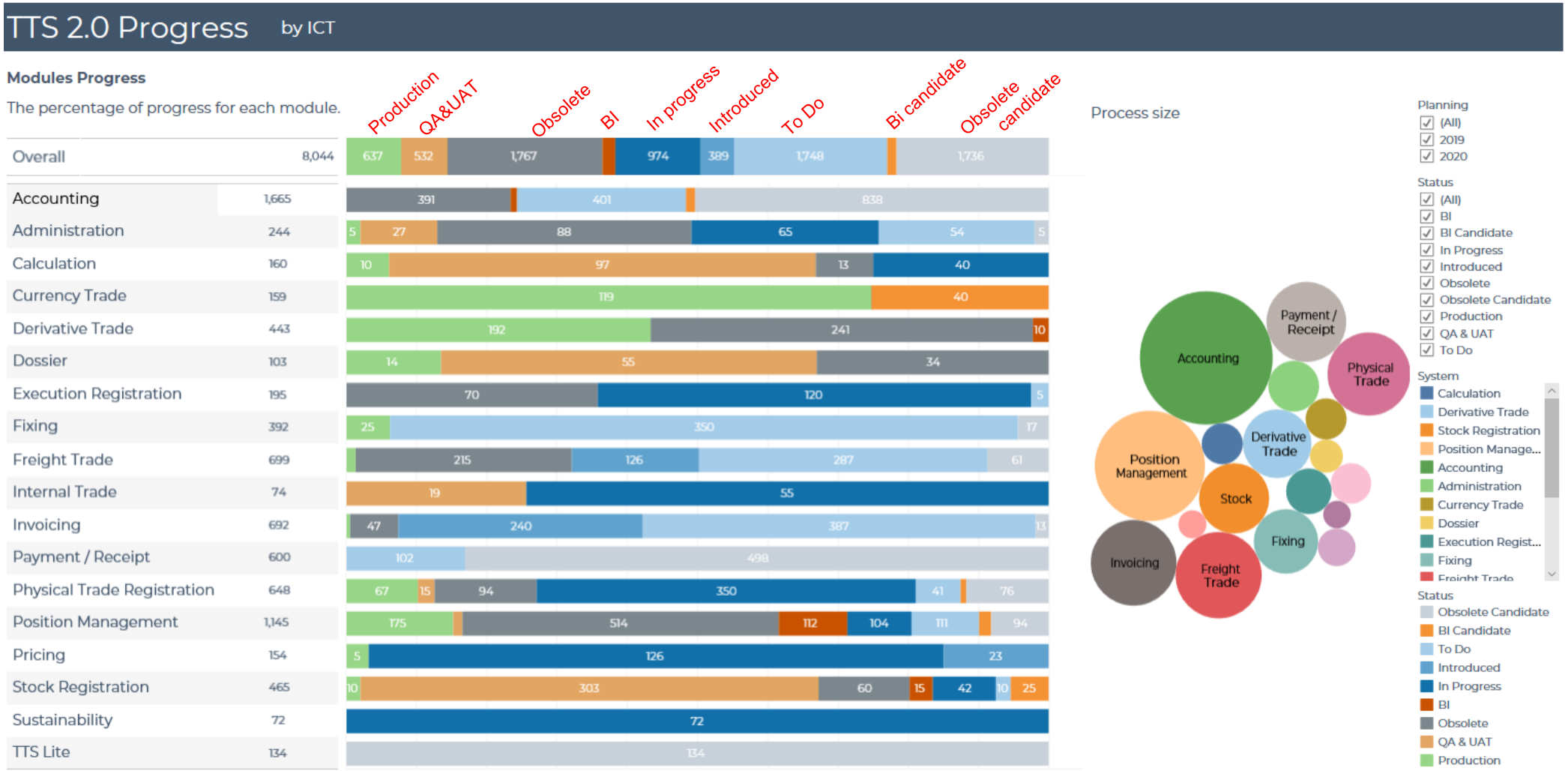
Step 3

Activities included in price/FP	Activities excluded from price/FP
Generic components and accelerators	Coding and testing of PL/SQL programs
Deployment (up to Acceptance)	Design, implementation or support of infrastructure
Handover to line organization	Load, Stress and Performance tests
Scrum Master and Project Management	Production Deployment
Support for solving defects	…
….	

Payments (price per FP) associated to clear output moments



Fact-based modernization with the use of Function Points



GLENCORE AGRICULTURE

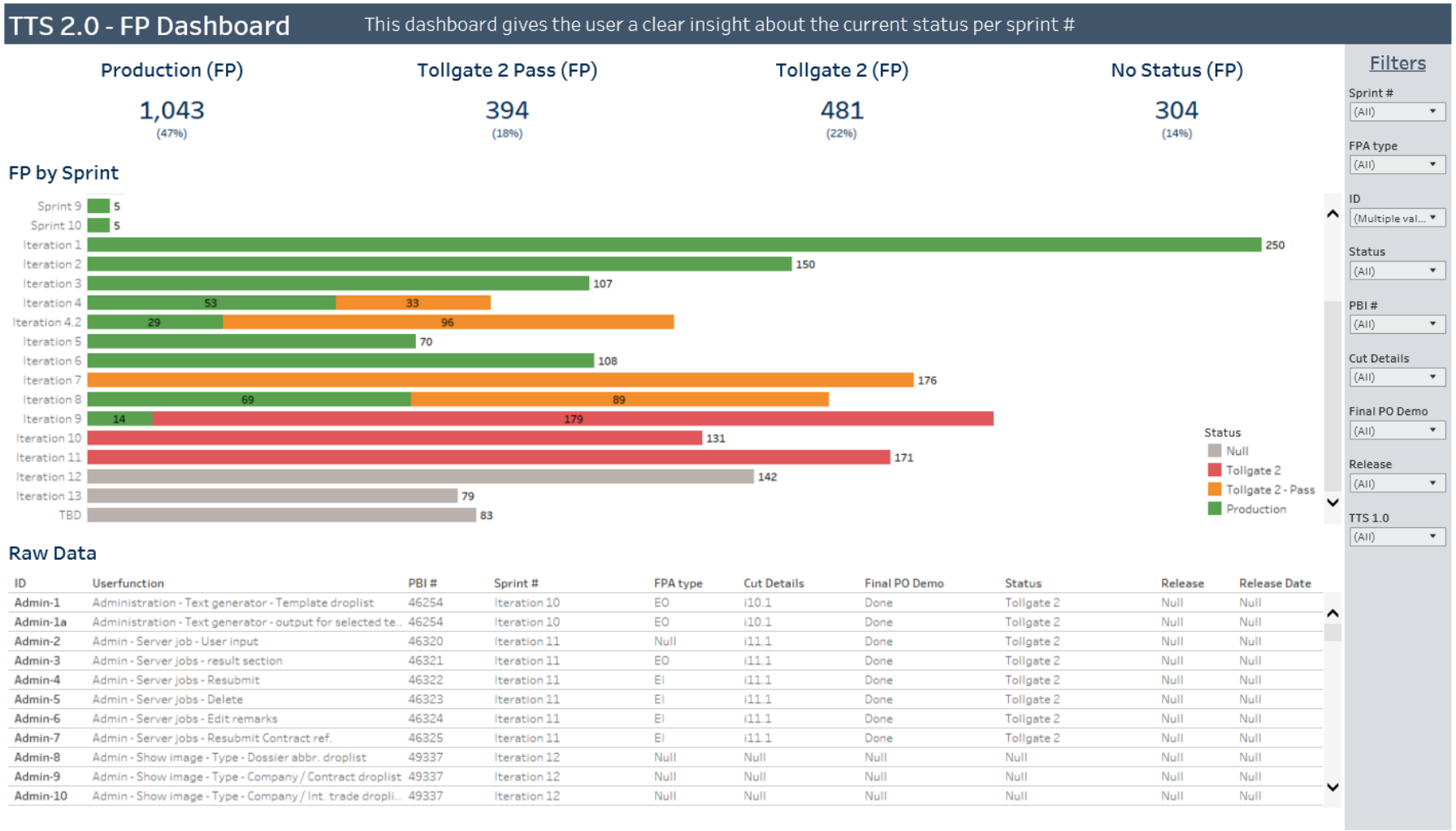
IWSM MENSURA

The future of IT Cost Estimation

Trends for the new Decade

Finidy if software really matters

Fact-based progress with the use of Function Points



GLENCORE AGRICULTURE

IWSM MENSURA

The future of IT Cost Estimation

Trends for the new Decade

Finidy if software really matters

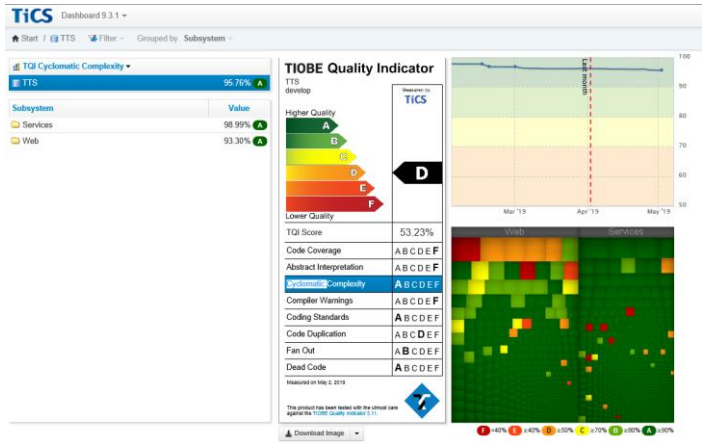
Work excluded from price per FP is calculated as ‘virtual’ FP’s

- We have a bonus/Malus on throughput (# Function Points per period)
- Throughput is based on ‘real’ Function Points and ‘virtual’ Function Points
  - Not everything can be counted as Function Points – framework upgrades, downtime etc.
  - ‘Virtual’ Function Points are RFC hours divided by agreed productivity
  - In example below (using 20 hours per FP) the throughput is 250+10=260 Function Points

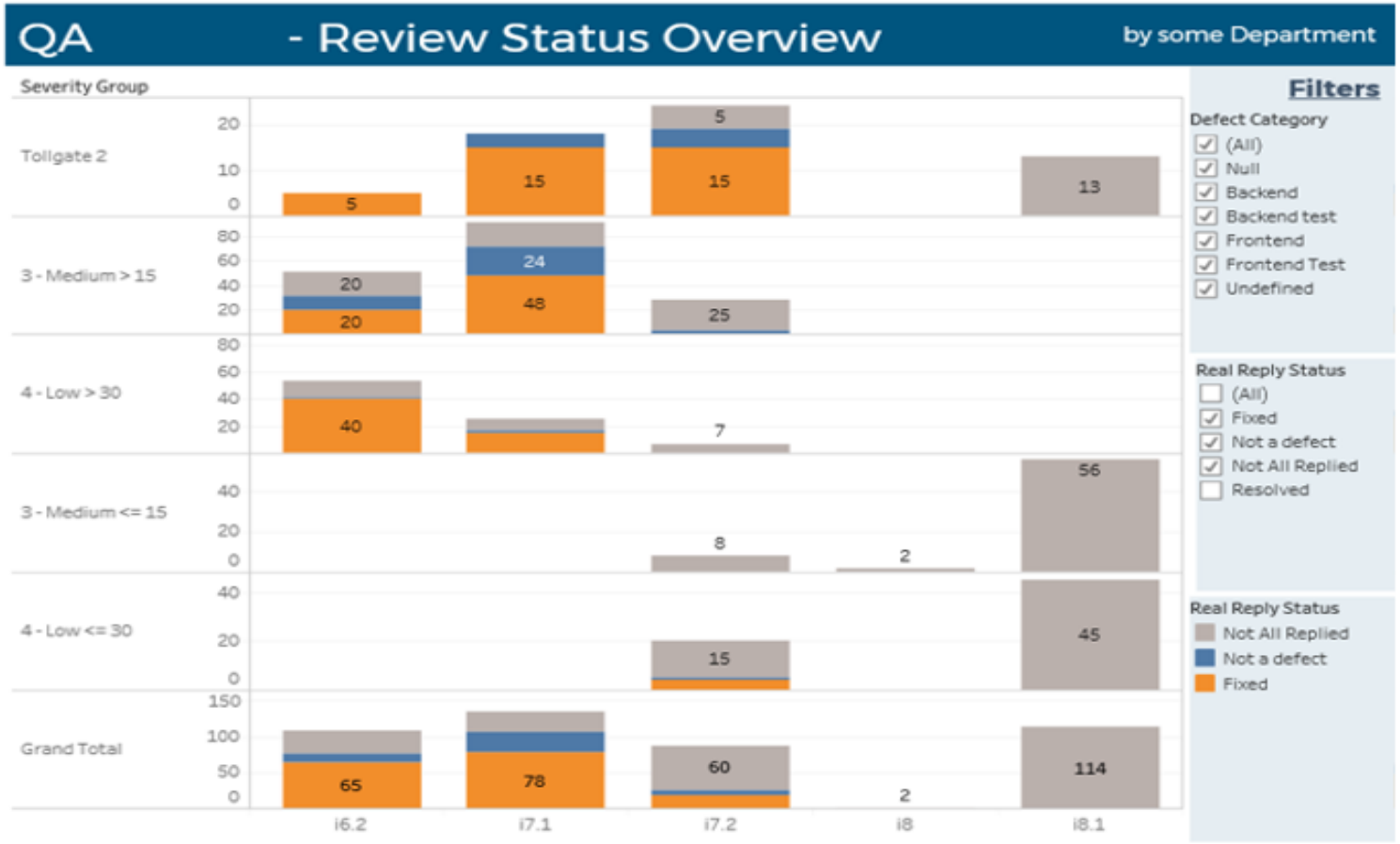
Iteration	Start date	End date	‘real’ Function Points
Iteration 6	Mon 3 Dec 2018	Fri 11 Jan 2019	105 FP
Iteration 7	Mon 14 Jan 2019	Fri 1 Mar 2019	145 FP
			250 FP

RFC	Description	Hours	‘Virtual’ Function Points
RFC-2019-01	Iteration 6 - downtime	80	4 FP
RFC-2019-02	Upgrade AG-grid	120	6 FP
			10 FP

Measure code quality by SonarQube/Tiobe is not enough …



- You will need manual review too



GLENCORE AGRICULTURE

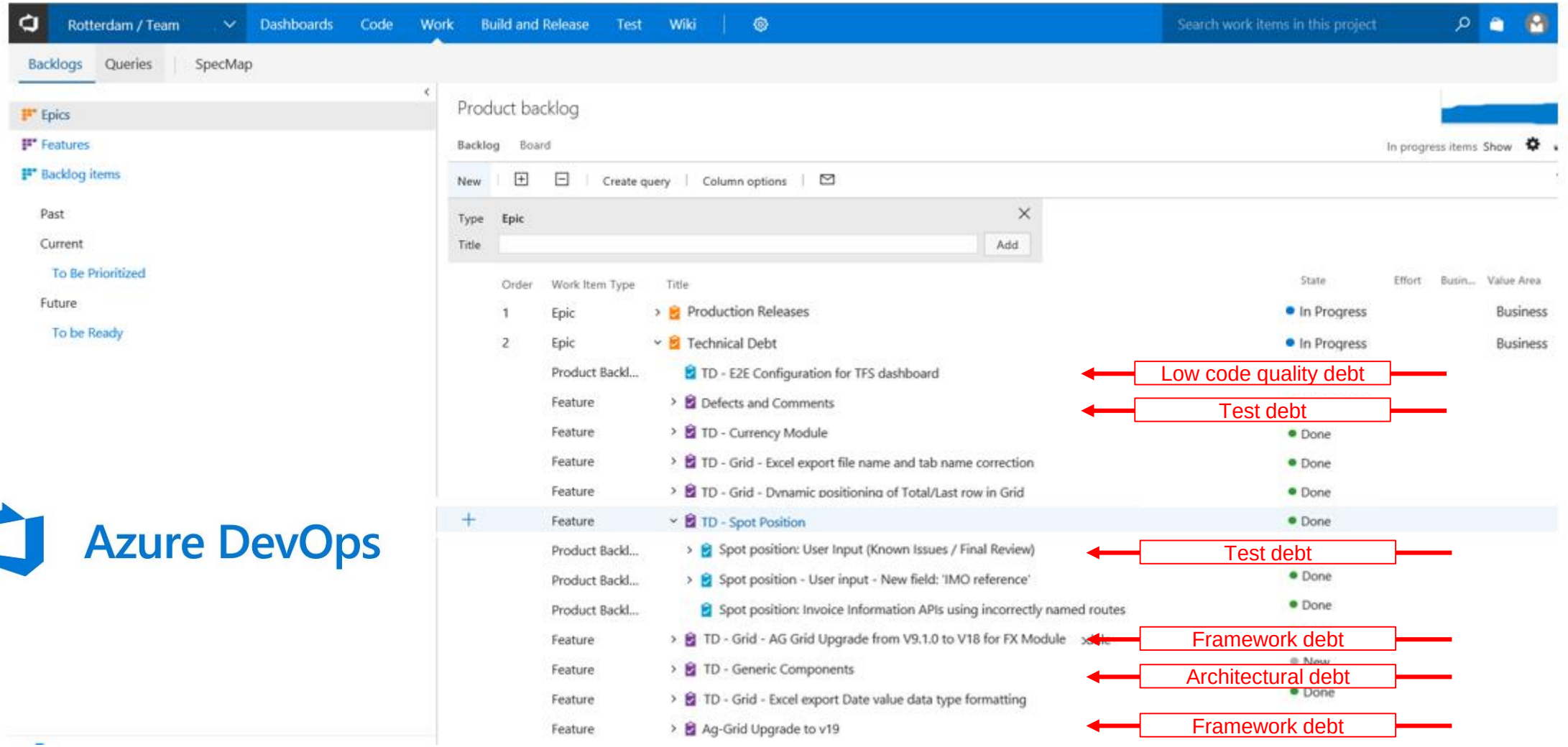
IWSM MENSURA

The future of IT Cost Estimation

Trends for the new Decade

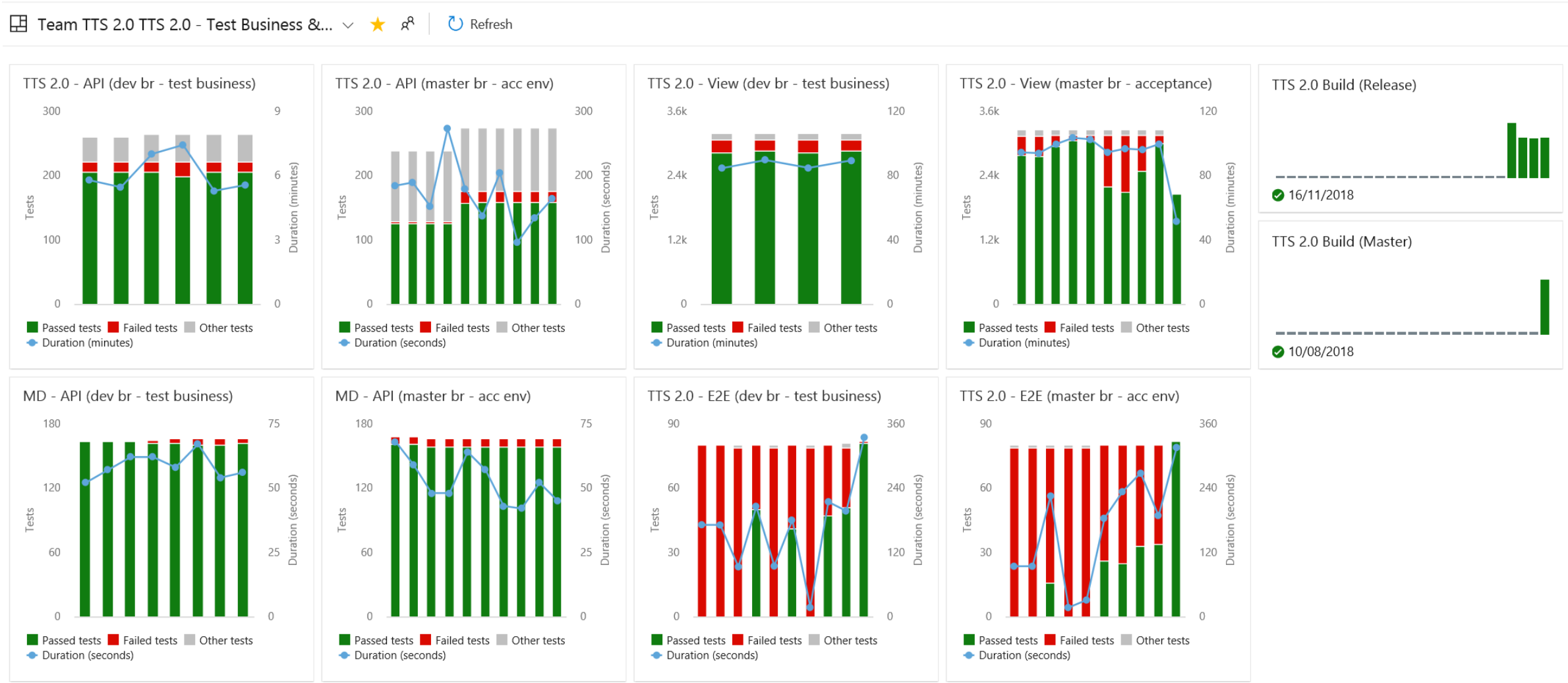
Finidy if software really matters

Make sure Technical Debt is captured, visible and planned!



Azure DevOps

QA metrics have to be fully integrated in the build pipelines



Aim for ‘zero tolerance’: monitor # of defects in UAT&PRD

Input for calculation of metrics	Release 3	Release 4	Release 5	Release 6	Release 7
Amount of FP	50	10	304	71	63
Start date UAT - first iteration	15-Feb-2019	15-Mar-2019	29-Apr-2019	24-May-2019	29-Jul-2019
End date UAT - first iteration	08-Mar-2019	29-Mar-2019	14-May-2019	17-Jun-2019	12-Aug-2019
Number of defects in first iteration with severity 1	0	0	1	0	0
End date UAT	08-Mar-2019	29-Mar-2019	14-May-2019	17-Jun-2019	12-Aug-2019
Total numbers of defects with severity 1 and 2 in UAT (defects/FP)	1	0	2	1	0
Start date production	24-Mar-2019	14-Apr-2019	24-May-2019	23-Jun-2019	25-Aug-2019
End date 4 weeks in production	21-Apr-2019	12-May-2019	21-Jun-2019	19-Jul-2019	20-Sep-2019
Total defects with severity 1 and 2 in production for 4 consecutive weeks	1	1	4	1	
Amount of defects with severity 3 and 4 in production for 4 consecutive weeks	0	0	3	0	

	Metric	Target	Release 3	Release 4	Release 5	Release 6	Release 7
9	Maximum defect with severity 1 in first iteration of UAT (defects/FP)		0,00	0,00	0,00	0,00	0,00
10	Maximum defects with severity 1 and 2 in UAT (defects/FP)		0,02	0,00	0,01	0,01	0,00
15	Maximum defects with severity 1 and 2 in production for 4 consecutive weeks		0,02	0,10	0,01	0,01	0,00
16	Maximum defects with severity 3 and 4 in production for 4 consecutive weeks (defects/FP)		0,00	0,00	0,01	0,00	0,00

Our experience with an output-based contract

- Explicit demand for price per Function Point brought different players to the table, some vendors stepped back
- Importance of very detailed product descriptions (system deliverables)
  Crucial to have very clear what's in the price per FP, saves a lot of discussion!
- Vendor should make thorough assessment to understand potential complexity.
- Don't bother dev team(s) with the commercial side, quality first
- Measured values of the KPIs/metrics reflects the reality
  - Based on the quality metrics, we notice when new employees are added to the project
  - Process 'bottlenecks' become visible (for example # FP's in stock)
  - Enable tooling to closely monitor where possible
- Helps to get focus for any improvement in all area's (automation etc.)

Questions?

Nico Oosterom
Nico.Oosterom@glencore.com
www.glencoreagriculture.com

Richard Sweer
Richard@finidy.nl
www.finidy.nl

Download here the whitepaper
More grip less guesswork in agile IT projects