Benchmarking agile software development

Harold van Heeringen
Drs. Harold van Heeringen

Graduated in Business economics at the University of Groningen in 1997
>20 years experience in IT, >15 years in software measurement and metrics
Married, 3 kids, living in Veendam (North of the Netherlands)

**Hobbies** – Chess, soccer and software metrics:

**METRI** – Consultant Estimation & Performance Measurement

**NESMA** – Board member International cooperation and partnerships

**ISBSG** – Immediate Past President (2011-2019)

**COSMIC** – Dutch representative in the International Advisory Council (IAC)

**ICEAA** – Trainer of CEBoK chapter 12: Software Cost Estimation

**sCEBoK** – initiator and module developer
Low industry Performance Measurement maturity
High Performance Teams – really?

-80%
-60%
-40%
-20%
0%
20%
40%
60%
80%
100%

Low Performing team
Industry Average
High Performing team
Agile Hypecycle

- Peak of Inflated Expectations
- Plateau of Productivity
- Trough of Disillusionment
- Technology Trigger

Visibility vs. Time

2019

België scoort slechter
Interne risicorapportages ING, score moet onder de 2,5 liggen

<table>
<thead>
<tr>
<th>METRI</th>
<th>IT STARTS WITH THE FACTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ING Nederland</strong></td>
<td><strong>ING België</strong></td>
</tr>
<tr>
<td>IT foundation</td>
<td></td>
</tr>
<tr>
<td>Q I</td>
<td>Q II Jul '18</td>
</tr>
<tr>
<td>2.4</td>
<td>2.1</td>
</tr>
<tr>
<td>Platform security</td>
<td></td>
</tr>
<tr>
<td>2.6</td>
<td>2.5</td>
</tr>
<tr>
<td>Security monitoring</td>
<td></td>
</tr>
<tr>
<td>2.6</td>
<td>2.5</td>
</tr>
<tr>
<td>Change management</td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>2.2</td>
</tr>
<tr>
<td>IT resilience</td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>2.2</td>
</tr>
<tr>
<td>Identity and access management</td>
<td></td>
</tr>
<tr>
<td>2.6</td>
<td>2.7</td>
</tr>
<tr>
<td>Cybercrime</td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>2.2</td>
</tr>
<tr>
<td>Score juli 2018</td>
<td>Score juli 2018</td>
</tr>
<tr>
<td>2.3</td>
<td>2.5</td>
</tr>
</tbody>
</table>
See the big picture

...AND THIS LITTLE RED THINGY IS OUR AGILE PROJECT

FINALLY WE’RE AGILE!
The industry practice: story points
Key metrics for Agile teams

- **Productivity**
  - Effort hours spent
  - Size of the delivered Software Product

- **Cost Efficiency**
  - Team cost
  - Size of the delivered Software Product

- **Velocity**
  - Duration (months)
  - Size of the delivered Software Product

- **Product Quality**
  - Defects Delivered
  - Size of the delivered Software Product

- **Code Quality Metrics**
  - Maintainability
  - Reliability
  - Performance
  - Security
  - Technical Debt
When Agile Teams Think about Function Points

They Imagine This.....
It's ugly, old, and evil
It may even eat children!

Something we did in the 80's, and even then it always failed!

But... we also used the meter, the liter, the kilo in the 80's
And still do...

Function Points measure functionality **regardless**
• The Technical implementation (e.g. programming language)
• The Implementation method (e.g. Agile)
• Other non-functional requirements (e.g. availability)
Why not?

• They see overdocumentation
Why not?

- They see over-waterfall
Why not?

- They see a management tool
Automated Function Points

- Implement functional sizing without bothering the teams!

- similar to IFPUG and Nesma FP
- ISO standard - ISO 19515:2019
- OMG/CISQ Standard

- Implement in the CI/CD pipeline of Agile teams
- No waste for the teams, while delivering value for management
Senior management is responsible and accountable

<table>
<thead>
<tr>
<th>Type of Decision</th>
<th>Measurement</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team size estimation</td>
<td>Function Points</td>
<td>Management</td>
</tr>
<tr>
<td>Performance measurement</td>
<td>Function Points</td>
<td>Management</td>
</tr>
<tr>
<td>Long term estimation</td>
<td>Function Points</td>
<td>Management</td>
</tr>
<tr>
<td>Benchmarking</td>
<td>Function Points</td>
<td>Management</td>
</tr>
<tr>
<td>Budgetting</td>
<td>Function Points</td>
<td>Management</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of Decision</th>
<th>Measurement</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine backlog priority</td>
<td>Story Points</td>
<td>Product owner</td>
</tr>
<tr>
<td>Sprint backlog items</td>
<td>Story Points</td>
<td>Team / product owner</td>
</tr>
<tr>
<td>Check progress SBI's</td>
<td>Story Points</td>
<td>Scrum master</td>
</tr>
</tbody>
</table>
• Agile Team Performance Measurement, based on standards
• Trends through time
• High performance teams vs. Low performance teams – learn and improve!
• Benchmark! METRI Data engine or ISBSG D&E data repository

• But what about the quality of the product?
## Product Metrics – static code analysis

<table>
<thead>
<tr>
<th>Description</th>
<th>Business value</th>
</tr>
</thead>
</table>
| **Transferability** |  - Avoid to be tied to a internal resource / team or outsourcer  
  - Improve team productivity  
  - Ease transfer between contractors, internal teams and outsourcer |
| **Changeability** |  - Improved maintenance ease and delays  
  - Improved predictability of application releases  
  - Improve time to market |
| **Robustness** |  - Reduce defects and bugs in production  
  - Lower the application downtimes  
  - Improve User Experience |
| **Efficiency** |  - Improve response time of the application  
  - Lower resources needs of the application  
  - Improve scalability |
| **Security** |  - Improve security of both the application and the critical business data used |
| **Maintainability (TQI)** |  - Lower general maintenance costs of the applications |
The total picture

**RISK**

- **Robustness**
  - Risk of critical failures in production
  - Score: 2.47

- **Efficiency**
  - Risk of performance / scaling issues
  - Score: 1.75

- **Security**
  - Risk of security breaches
  - Score: 1.83

- **Changeability**
  - Ease and speed of modifying
  - Score: 2.59

- **Transferability**
  - Ease and speed of learning
  - Score: 2.93

**MAINTAINABILITY**

- **Overall application functional size**
  - Score: 6,998

- **Performance and productivity**
  - EFP
  - Added: 64
  - Modified: 34
  - Deleted: 6
  - Total: 104

**ISO Standards for automated function points**

**FUNCTIONAL SIZE**

- 6,998 CISQ Compliant Automated FP4

**PRODUCTIVITY**

**BENCHMARK**
Practical case

Productivity Index

Cost Index

Quality Index

Velocity Index

Productivity Index

Cost Index

Quality Index

Velocity Index

2018Q2

2019Q1

2019Q2

2019Q3

Market average
Product Quality Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>2018Q2</th>
<th>2019Q1</th>
<th>2019Q2</th>
<th>2019Q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Quality Index (TQI)</td>
<td>2.67</td>
<td>2.82</td>
<td>2.97</td>
<td>3.01</td>
</tr>
<tr>
<td>Robustness</td>
<td>2.84</td>
<td>2.85</td>
<td>2.88</td>
<td>2.91</td>
</tr>
<tr>
<td>Efficiency</td>
<td>3.15</td>
<td>3.17</td>
<td>3.17</td>
<td>3.18</td>
</tr>
<tr>
<td>Security</td>
<td>2.15</td>
<td>2.67</td>
<td>2.88</td>
<td>3.12</td>
</tr>
<tr>
<td>Changeability</td>
<td>2.86</td>
<td>2.98</td>
<td>3.01</td>
<td>3.12</td>
</tr>
<tr>
<td>Transferability</td>
<td>3.15</td>
<td>3.15</td>
<td>3.10</td>
<td>3.09</td>
</tr>
</tbody>
</table>
Conclusions

Understand the difference in metrics and the use of metrics
• Team metrics vs Management metrics
• Story Point metrics vs. Function Point metrics

Don’t use Story point metrics for management decision making.
Don’t use Function Point metrics in the agile team, unless the team sees the value and wants to use them.
Implement manual or automatic functional size measurement **without bothering the teams**
Quality is part of the productivity!

**The goal is not to punish, but always to improve!**
A bright future!

2019: Story Points

Standardized performance metrics based on FP
Standardized product metrics