A Cost Model for Early Cost Calculation of Agile Deliveries

ICEAA Workshop 2017

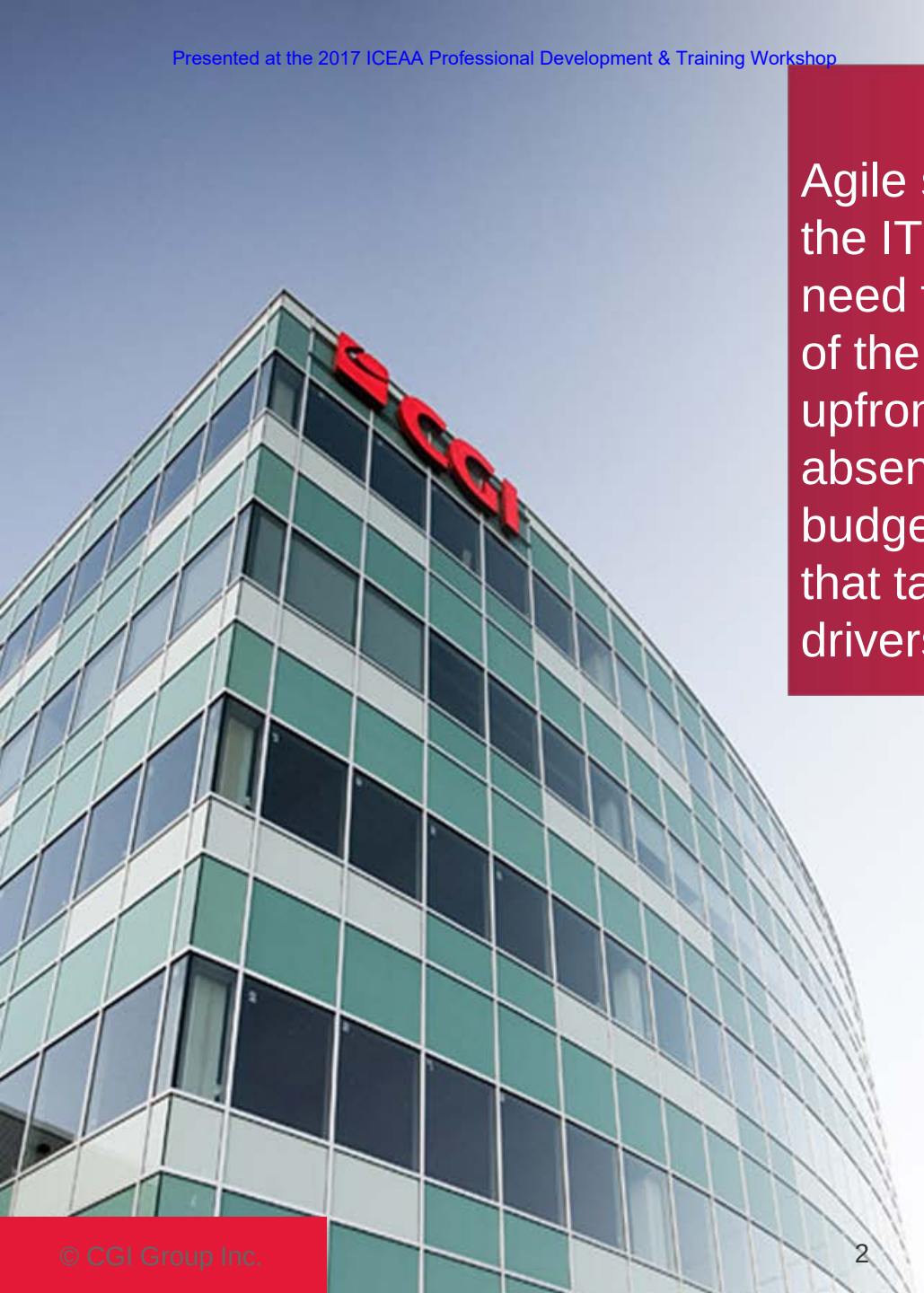
Eric van der Vliet

eric.van.der.vliet@cgi.com

ICEAA International Cost Estimating and Analysis Association

CGI

Experience the commitment®



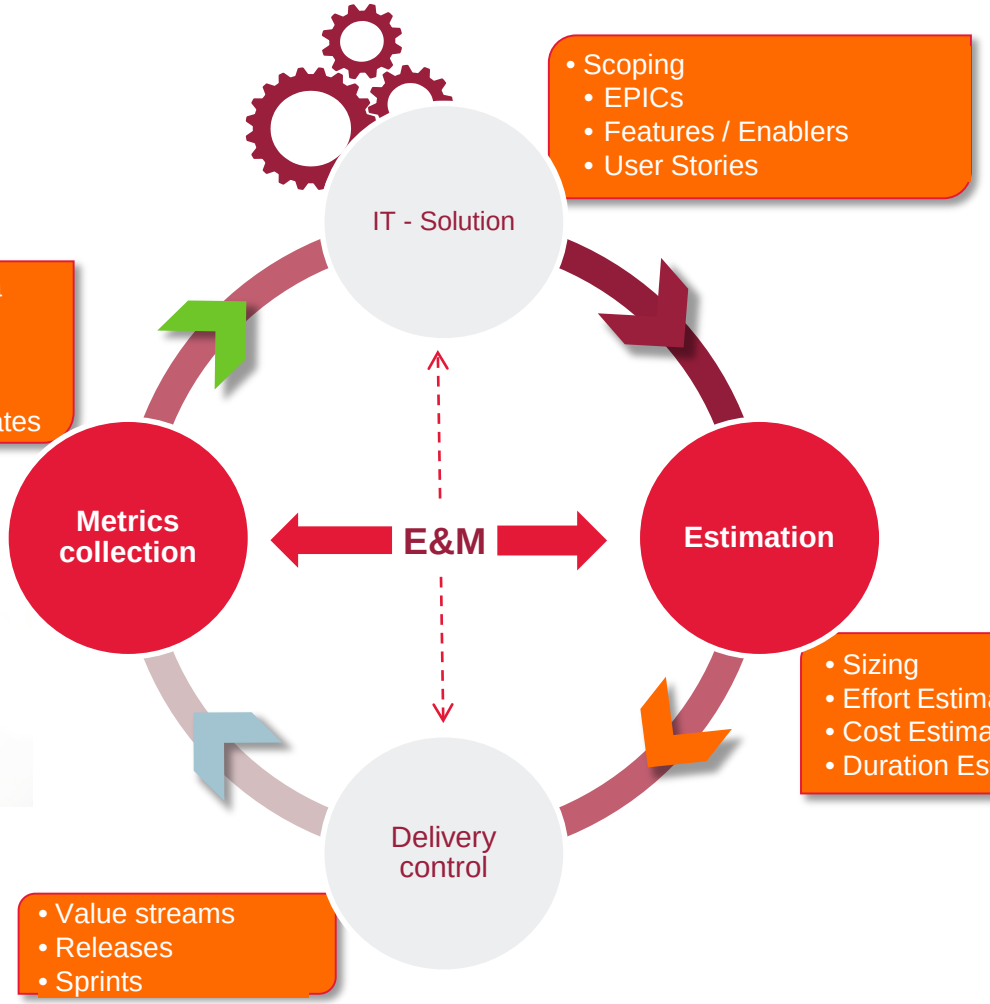
Problem statement

Agile software development provides the IT industry with the flexibility they need to keep up with the faster change of the business requirements. In agile upfront detailed specifications are absent, yet investment decisions need budget input. How to build a cost model that takes essential and additional cost drivers into account?

Introduction

- Agile deliveries were mainly small and controlled on sprint level
- How to manage Agile in combination with larger delivery contracts?
- Larger Agile contracts require an Agile cost model
- What cost drivers should be taken into account in such a cost model?

Cost estimation in an (Agile) delivery lifecycle

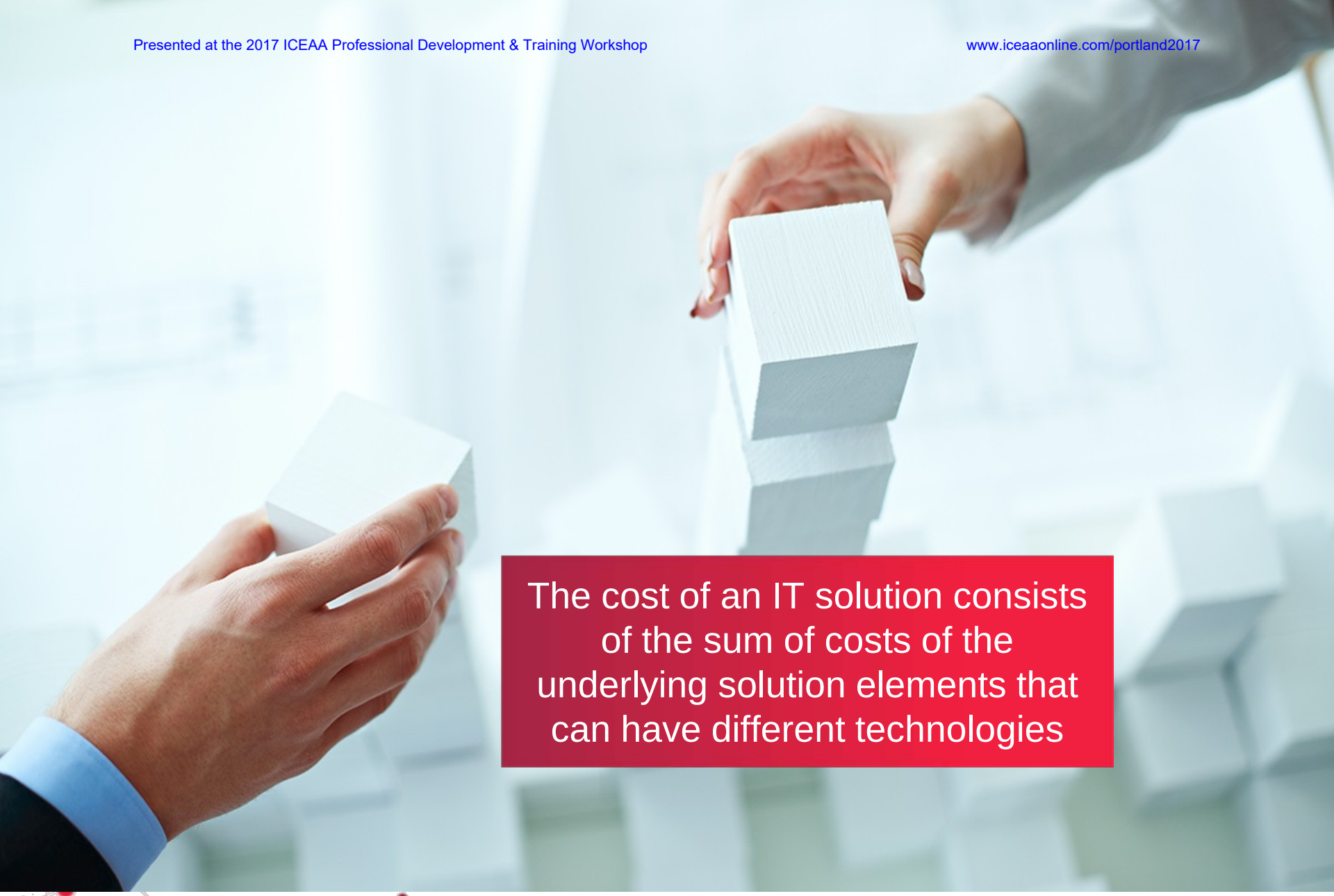


- Scoping
  - EPICs
  - Features / Enablers
  - User Stories

- Collection and analysis of actual data
  - Size
  - Velocity
  - Productivity
- Maintain data as input for new estimates

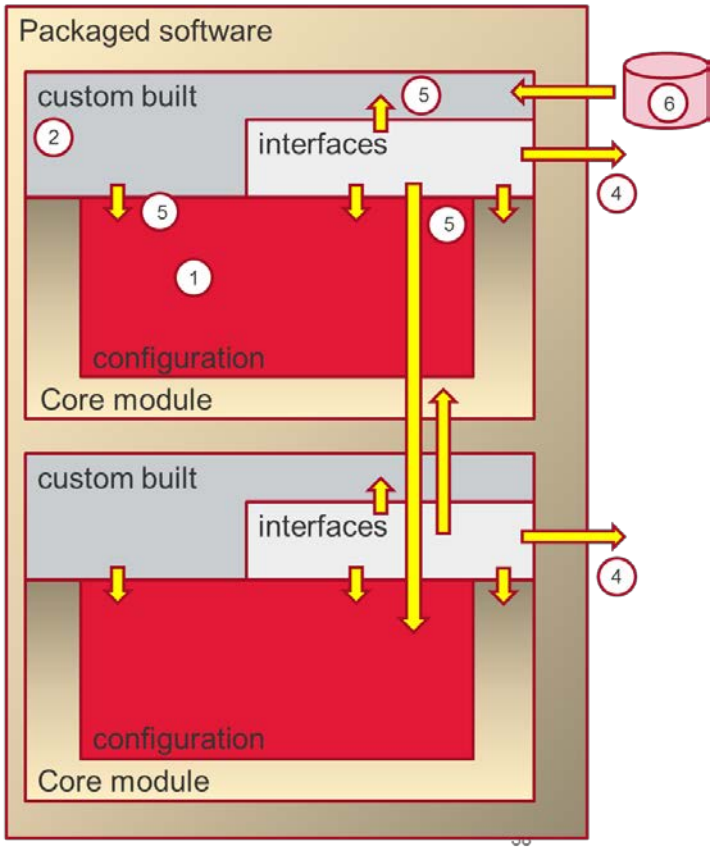
- Sizing
- Effort Estimation
- Cost Estimation
- Duration Estimation

- Value streams
- Releases
- Sprints



The cost of an IT solution consists of the sum of costs of the underlying solution elements that can have different technologies

Example - Package Implementation structure



1. Configuration
2. Custom built functionality
3. Core module(s) / standard functionality
4. External interfaces
5. Internal interfaces
6. Data

SAP

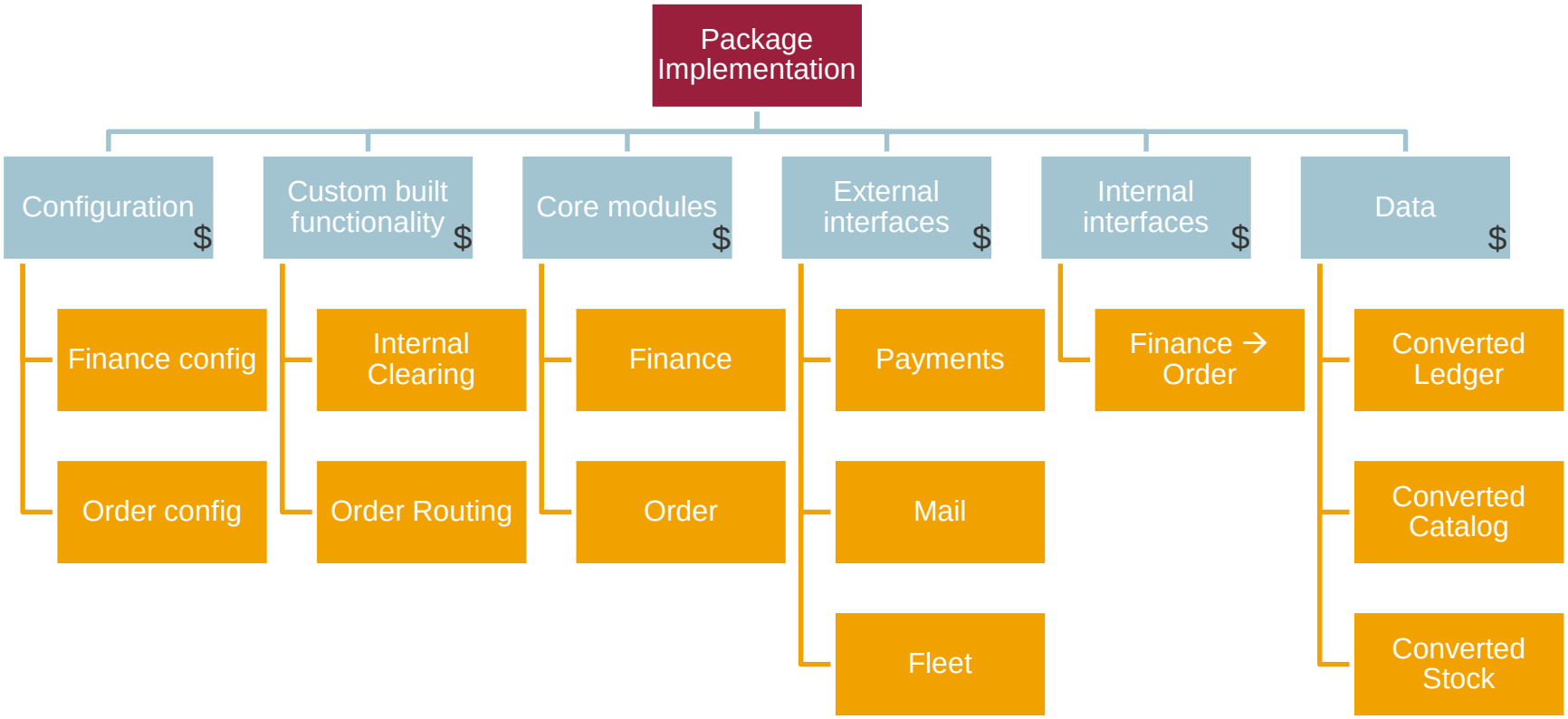
ORACLE E-BUSINESS SUITE

Microsoft Dynamics NAV

SIEBEL eBusiness

CGI

Cost elements package implementation



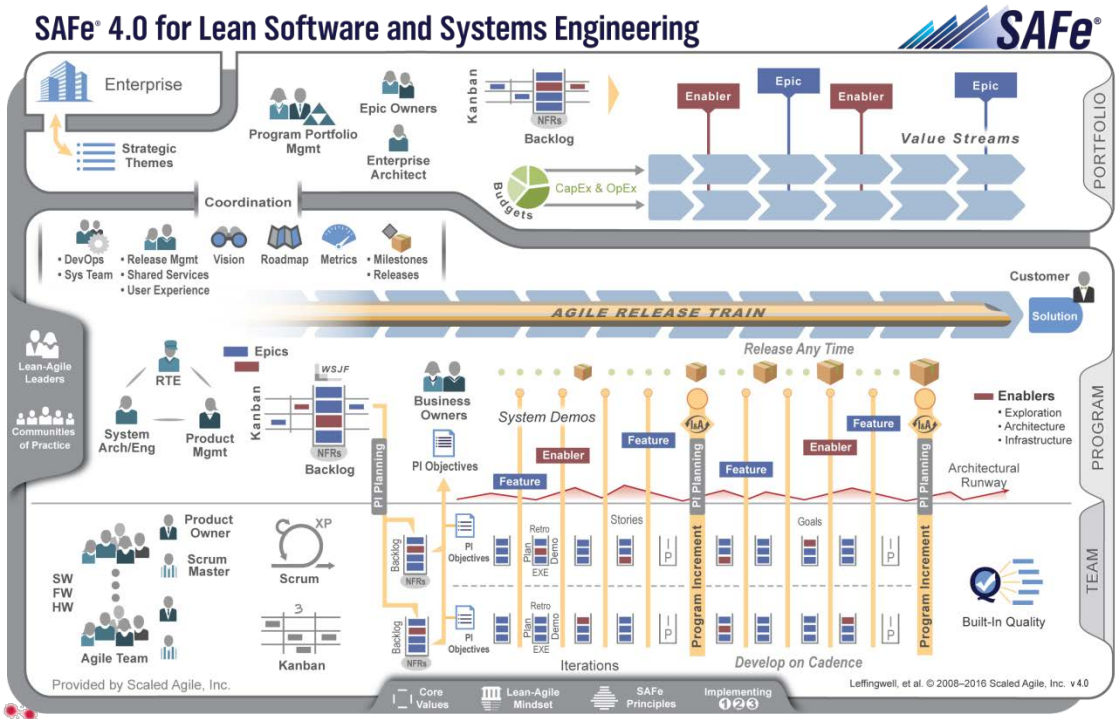
- Elements will have a different size
- Elements can have a different productivity
- Determine effort / costs (preferably) on element level

Large IT programs with a complex solution in combination with an Agile delivery, require a agile management framework like the Scaled Agile Framework (SAFe)

CGI

What is the Scaled Agile Framework (SAFe®)

- SAFe® is a freely revealed knowledge base of integrated, proven patterns for enterprise Lean-Agile development
- Synchronizes alignment, collaboration, and delivery for large numbers of teams



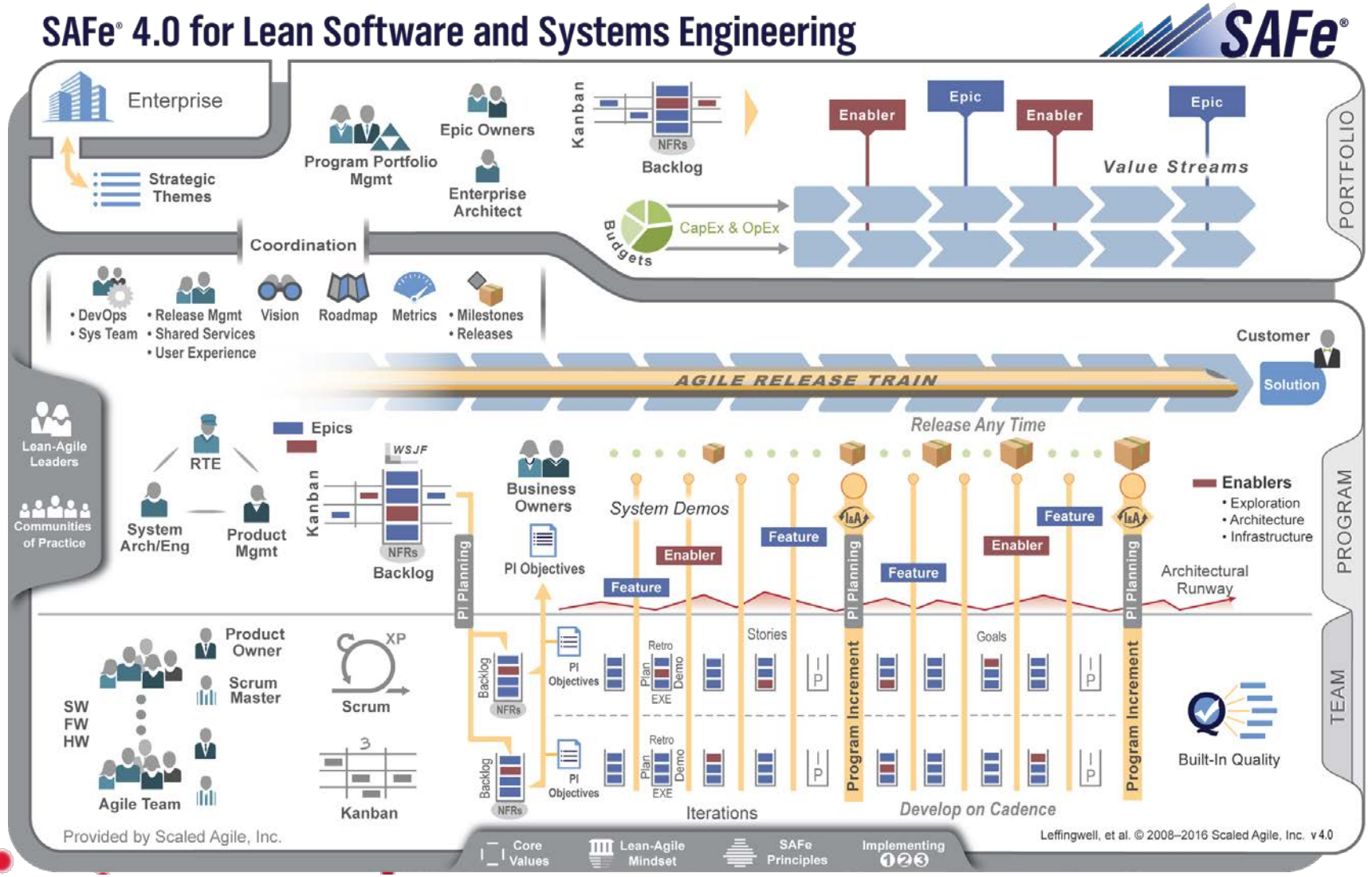
Core values:

1. **Built-In Quality**
2. **Program execution**
3. **Alignment**
4. **Transparency**

www.scaledagileframework.com

CGI

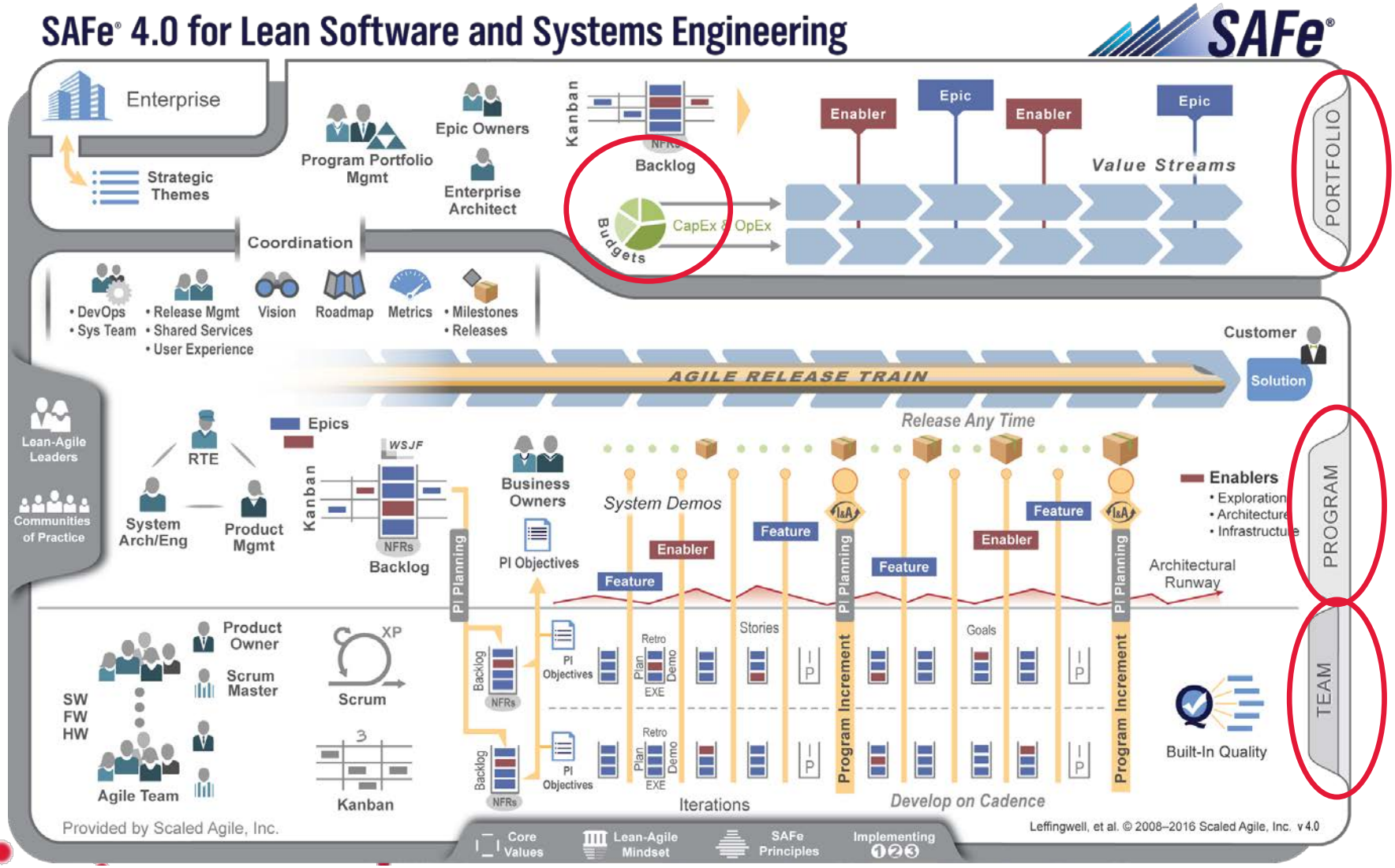
Scaled Agile Framework (SAFe®)



www.scaledagileframework.com

CGI

Scaled Agile Framework (SAFe®)



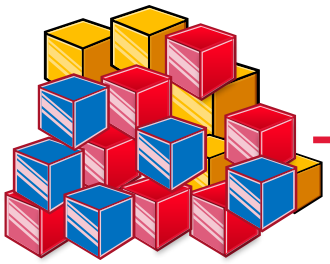
www.scaledagileframework.com

CGI

Relative sizing (story points) is mostly applied on team level. To determine the costs on portfolio level a more objective size estimation method is required

Cost estimation of a Scaled Agile delivery

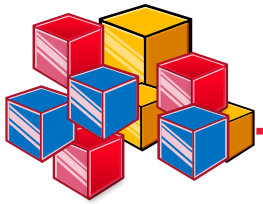
according to the Scaled Agile Framework (SAFe®)



Portfolio EPICS are assigned to programs

Portfolio Backlog

Portfolio
- EPICS with business needs
- Sizing based on size from lower levels
- Consolidation of the size
- Normalized Story Points required



Program Features are assigned to teams

Program Backlog

Program
- Features based on EPICS
- Size by means of Story Points
- Determine velocity of teams
- Assign features to teams

Team Backlog

Team
- Estimation based on stories
- Backlog based on velocity
- Estimation of tasks (optional)

Absolute sizing vs Relative sizing

Let's use Fruit points; relative sizing is easy on a small scale

But becomes more difficult on a large scale

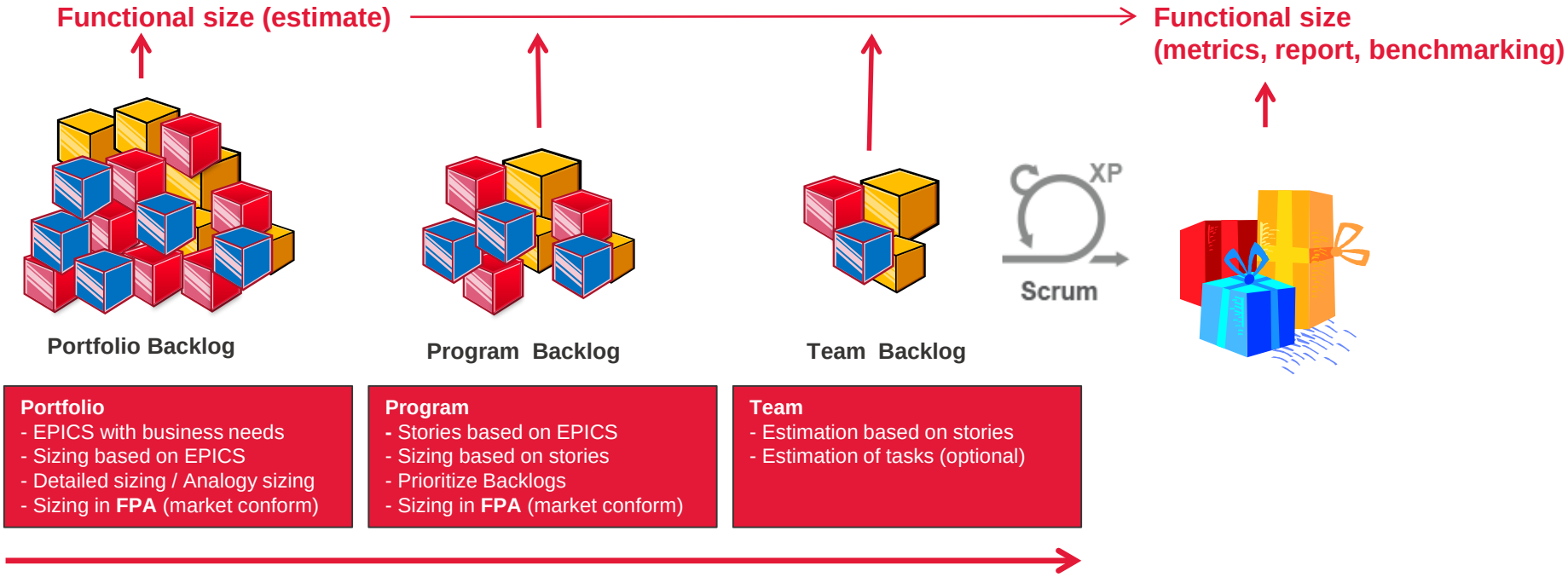
Challenges of using Story Points for cost estimation

- Story Points are relative and differ per team
- Use Story Points on an overall level requires normalization
- Normalization possibilities are limited (e.g. one SP is 8 hrs of effort)
- Teams can use different Story Point definitions as a starting point
- The difference in starting point will determine the difference in velocity

Team	Size (SP)	Velocity (SP / Sprint	Sprints
1	120	60	2
2	60	20	3

- For tracking the progress this is no issue
- For estimating the required budget this will not be useful

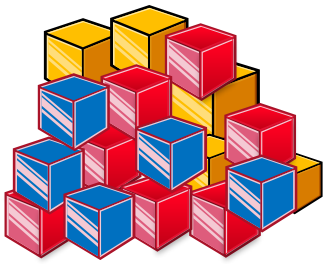
The use of functional sizing is recommended



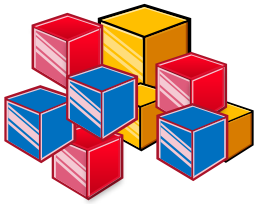
- Use of functional sizing according to an ISO standard:
  - COSMIC - Cosmic Function Points (CFP)
  - IFPUG - Function Points (FP)
  - Nesma - Function Points (FP)

This fits well in SAFe®

Functional size (estimate) → **Functional size (metrics, report, benchmarking)**



Portfolio Backlog



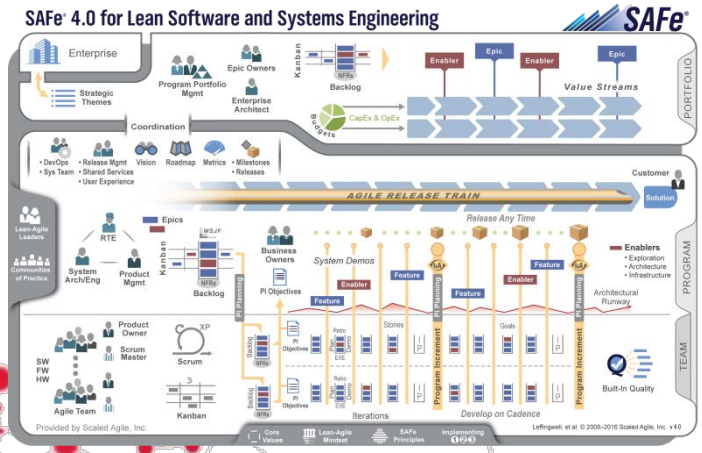
Program Backlog

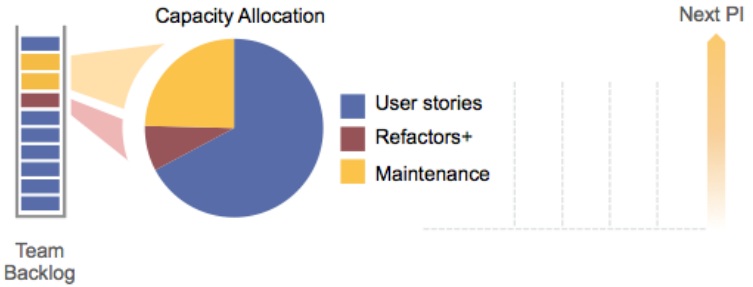
Team Backlog

Portfolio
- EPICS with business needs
- Sizing based on EPICS
- Detailed sizing / Analogy sizing
- Sizing in **FPA** (market conform)

Program
- Stories based on EPICS
- Sizing based on stories
- Prioritize Backlogs
- Sizing in **FPA** (market conform)

Team
- Estimation based on stories
- Estimation of tasks (optional)





www.scaledagileframework.com

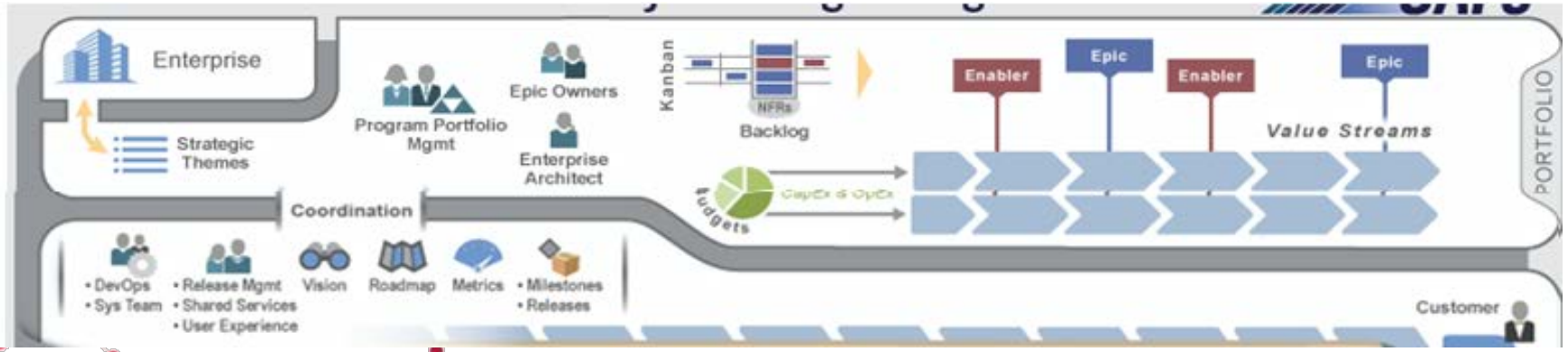
CGI

Sizing on portfolio level

- Analogy based sizing of Epics (e.g. Planning Poker)
- Estimation by comparison of Epics with statistical support (Historical data)
- T-Shirt sizing – Relative sizing of EPICs

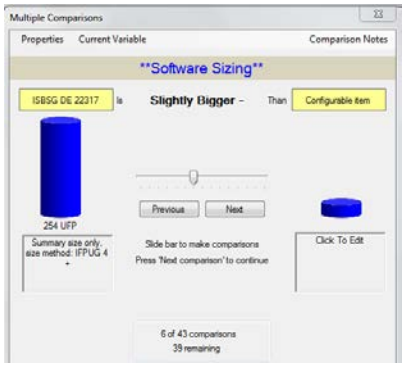
- The size estimate will result in a functional sizing (Function Points)
- Determine the uncertainty of the functional size

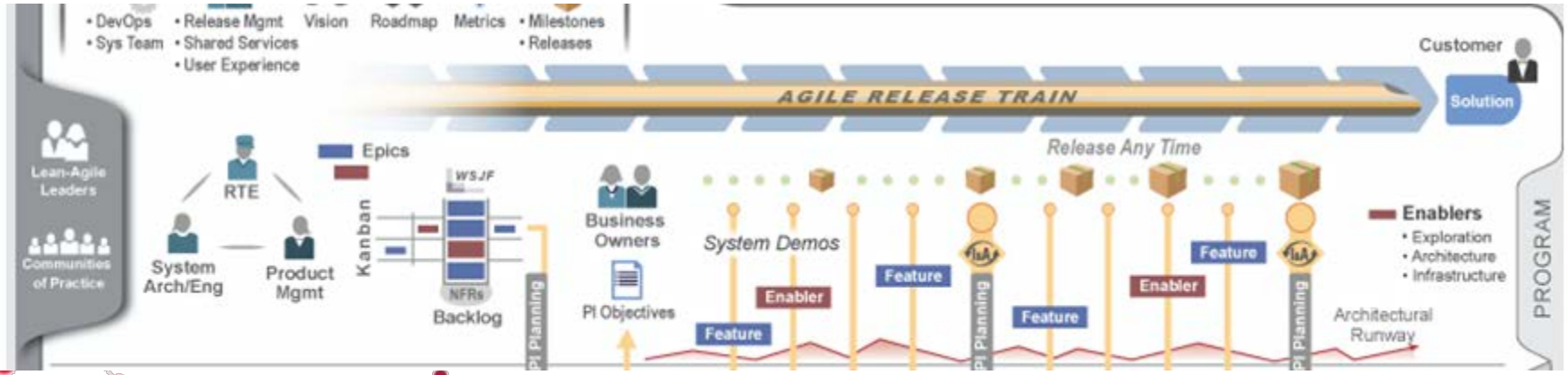
- Validate the sizing with the actual size (manual, automated)



Sizing on program level

- Analogy based sizing of Features (e.g. Planning Poker)
- Estimation by comparison of Features with statistical support (Historical data)
- T-Shirt sizing – Relative sizing of Features
- Manual sizing of FPs if enough details are available (FPA, QFP)
  - FPA = Function Point Analysis
  - QFP = Quick Function Points / Proxy Based Sizing
- Validate the sizing with the actual size (manual, automated)





Estimation on team level

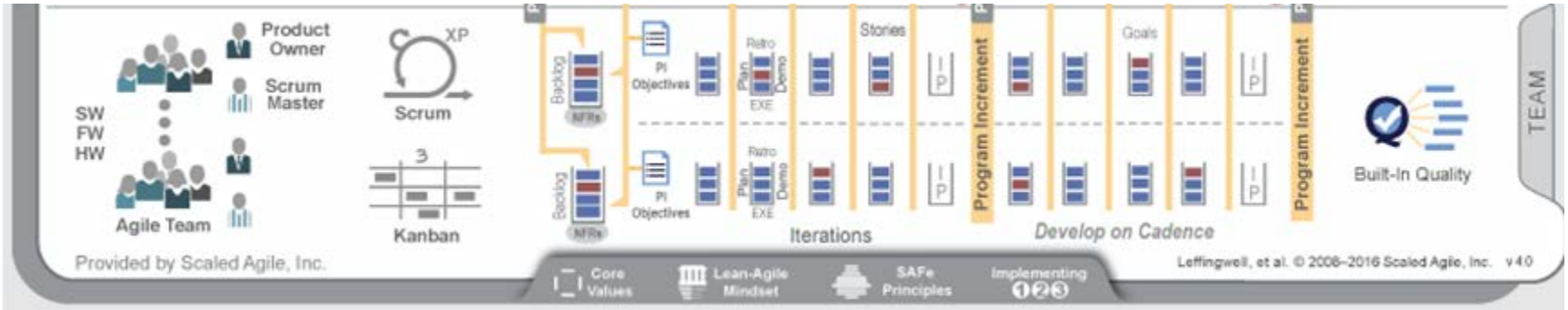
- Planning Poker (Story Points) will be mainly on team level
- Functional sizing (FPA) can mostly not be applied on team level (size is to small)

- Team level characteristics
  - User stories are defined
  - Teams size is defined
  - Duration of the sprint is fixed
  - Budget is fixed based on the team effort
  - Functionality is flexible

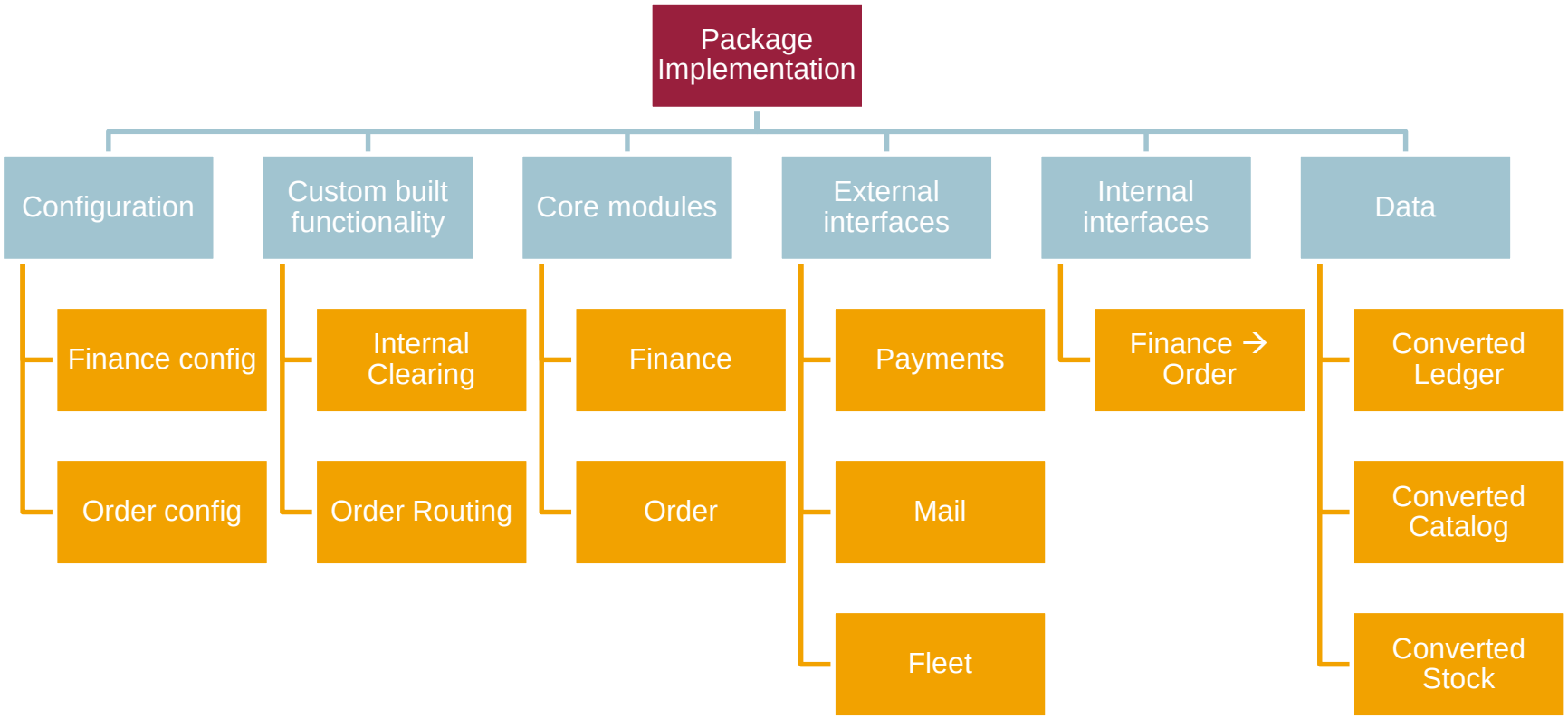
  - **Can we use a ratio between FPA and Story Points?**

→ **Functional size (metrics, report, benchmarking)**

↑

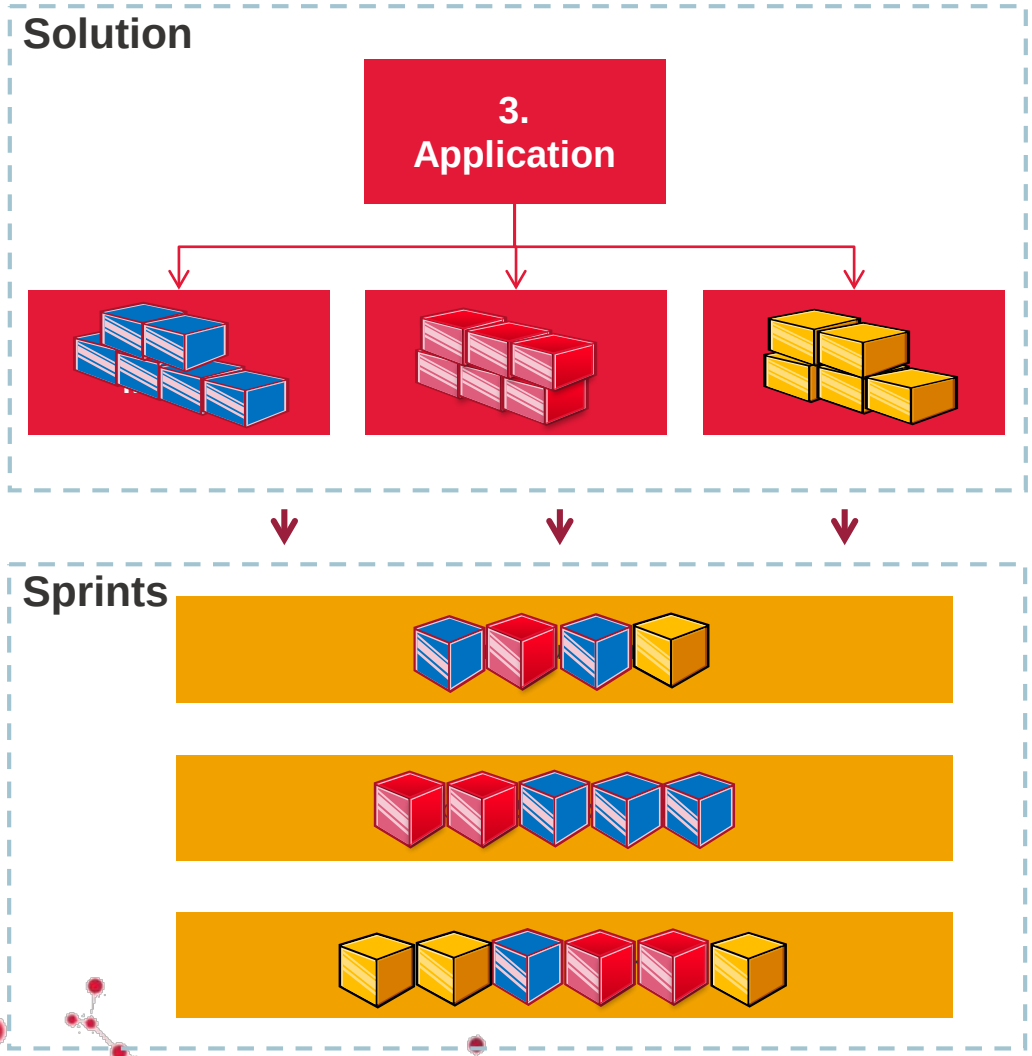


Determine the size of main solution elements



- Elements will have a different size
- Elements can have a different productivity
- Determine effort / costs (preferably) on element level

Solution elements are mapped on sprints

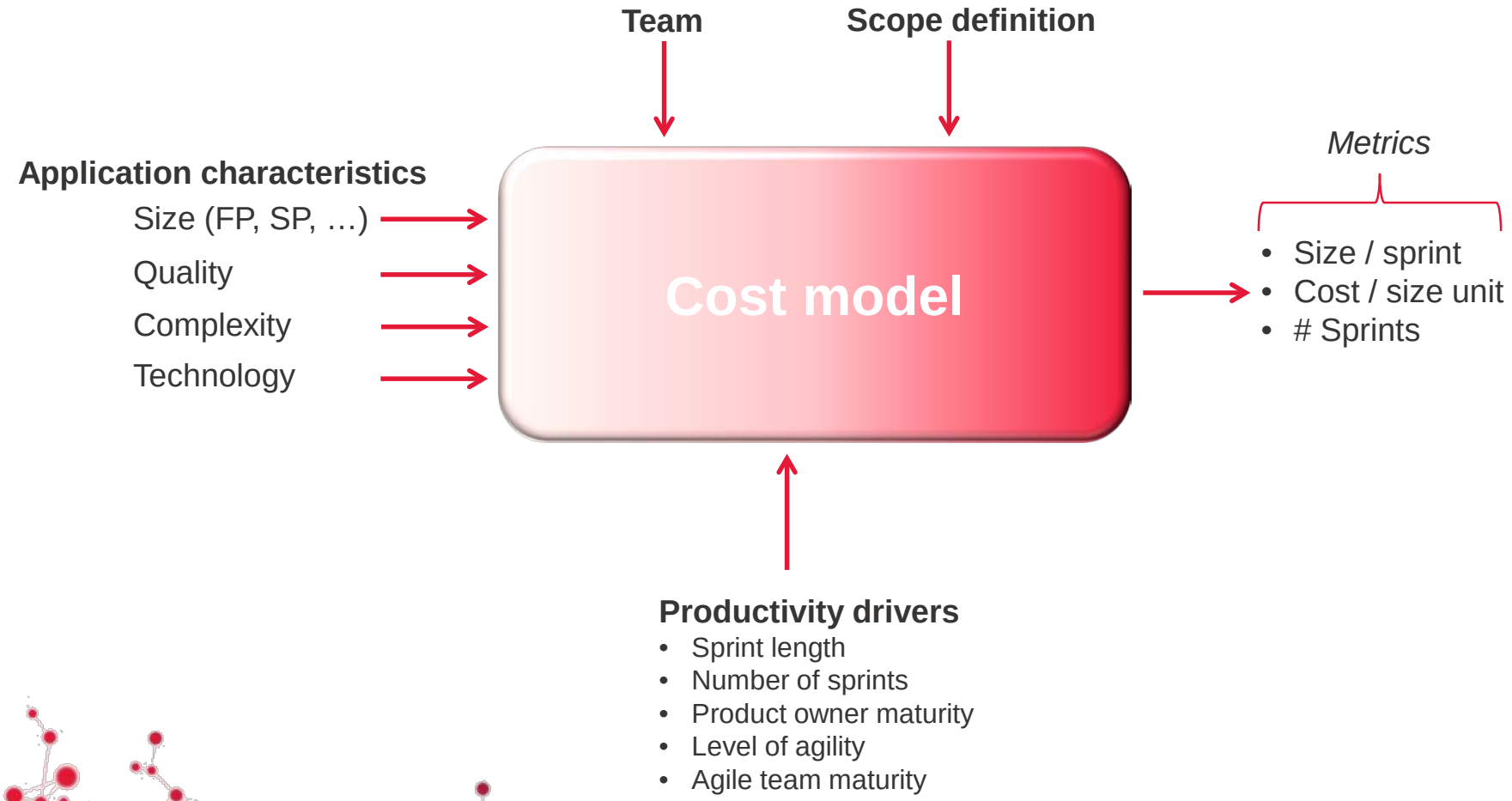


On sprint level it's difficult to compare sizing with the initial sizing on Portfolio or Program level. The size of a release can be compared with the initial sizing.

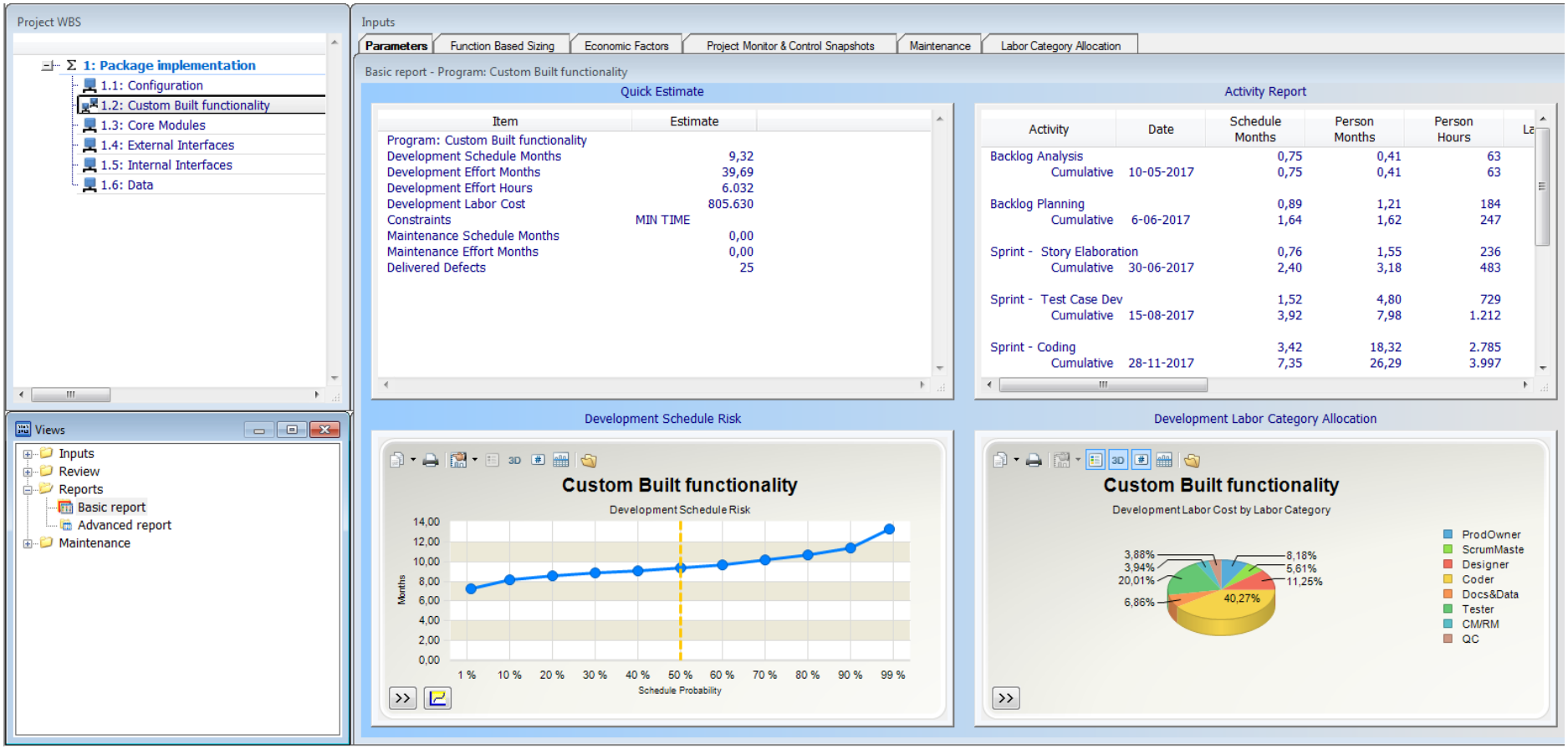
Based on the release the productivity for main solution elements can be determined and the team velocity.

Agile cost drivers – Input for the cost model

- Use of parametric tooling (e.g. Galorath, Price, QSM)

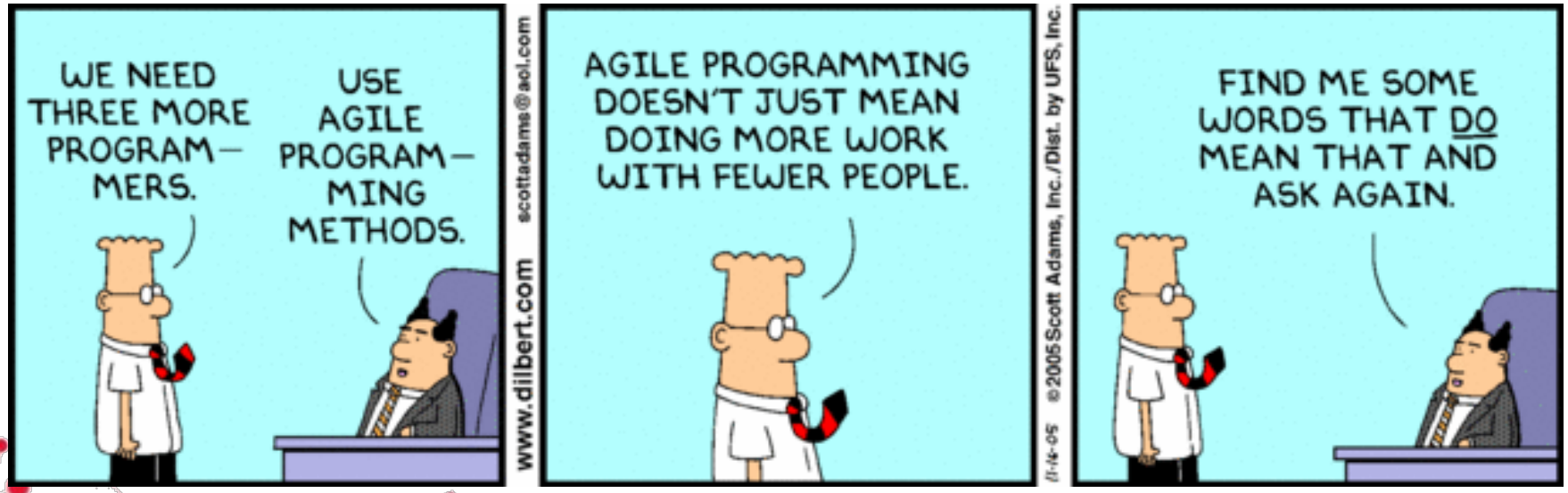


Example parametric estimation



Summary

- Cost estimation for a large Agile delivery requires a scaled approach
- Scaled Agile cost estimation requires solution based cost estimation
- Functional sizing and parametric estimation is recommended
- Costs can be determined based on size and using Agile cost drivers



Questions?

Mail: eric.van.der.vliet@cgi.com
Twitter: @ericvdvliet

Experience the commitment®

Thank you!

Mail: eric.van.der.vliet@cgi.com
Twitter: @ericvdvliet

Experience the commitment®