# Scaled agile deliveries; do we still need estimates?
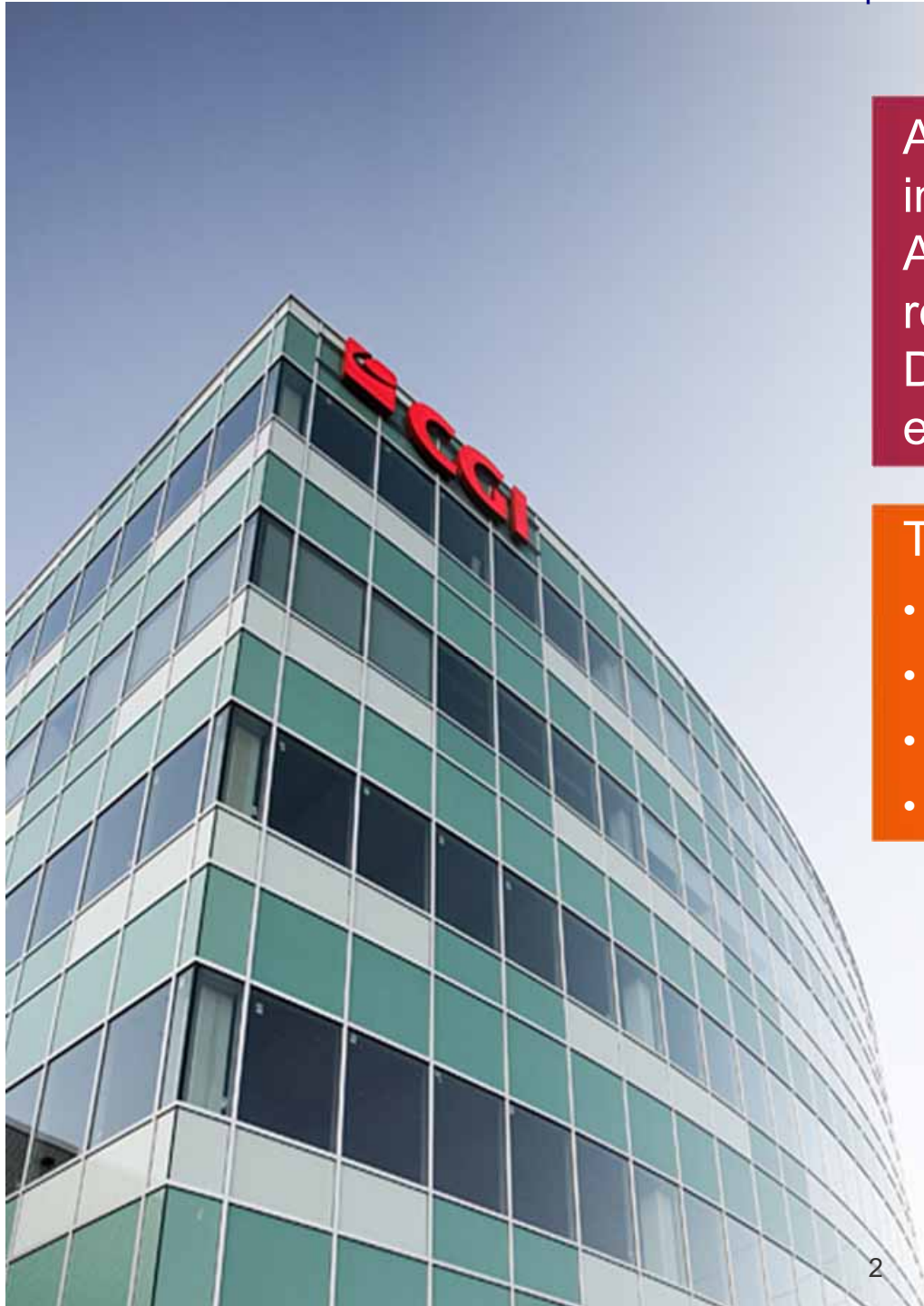ICEAA Workshop 2018

Eric van der Vliet
Director CGI - Estimation Centre

**ICEAA** International Cost Estimating and Analysis Association

**CGI**

Experience the commitment®

Agile becomes more and more important across the IT Industry. Agile is a different lifecycle that requires the right level of control. Does this mean that we still need estimates?

Topics for this session

- The need for estimation in Agile
- The need for control in Agile
- The relation with business value
- The contractual agreement

2

# The challenge with agile deliveries

- More contracts are demanded based on Agile
- Larger contracts require a Scaled Agile delivery model
- A popular Scaled Agile delivery models is the Scaled Agile Framework (SAFe® )
- Scaled Agile deliveries are requesting more for fixed price contracts

# How does that impact the estimation?

- Do we still need an estimate for Agile deliveries?
- Should we just start the development process?
- Are next steps just depending on the creativity of the Agile team?

**Portfolio Backlog**

CGI
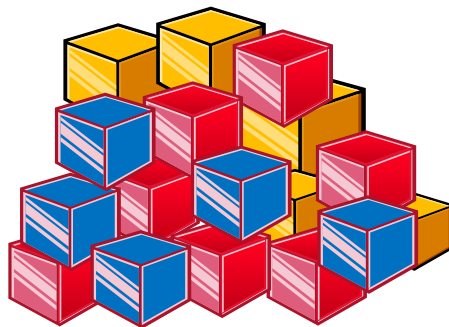
# Agile terminology

- Epic
  - An Epic describes the business need for a solution development
- Feature
  - A Feature is a service that fulfills a stakeholder need
- Story
  - Stories are short descriptions of a small piece of desired functionality



**CGI**

# Estimate the building of a stone house

- Objective is to build a stone house
- High level features are defined in a backlog
- Delivery approach is Agile

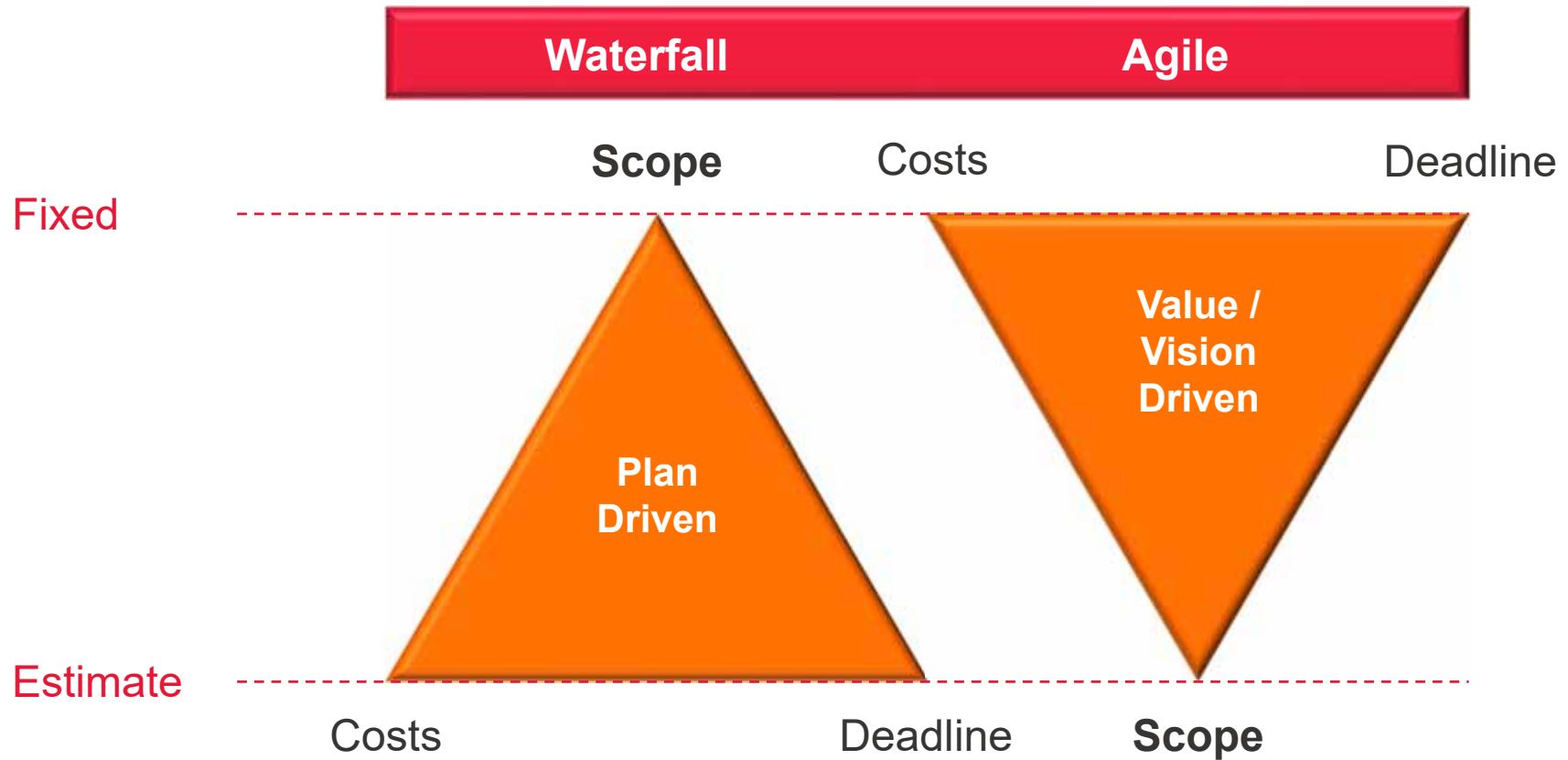- How do we estimate the building costs?



**CGI**

6

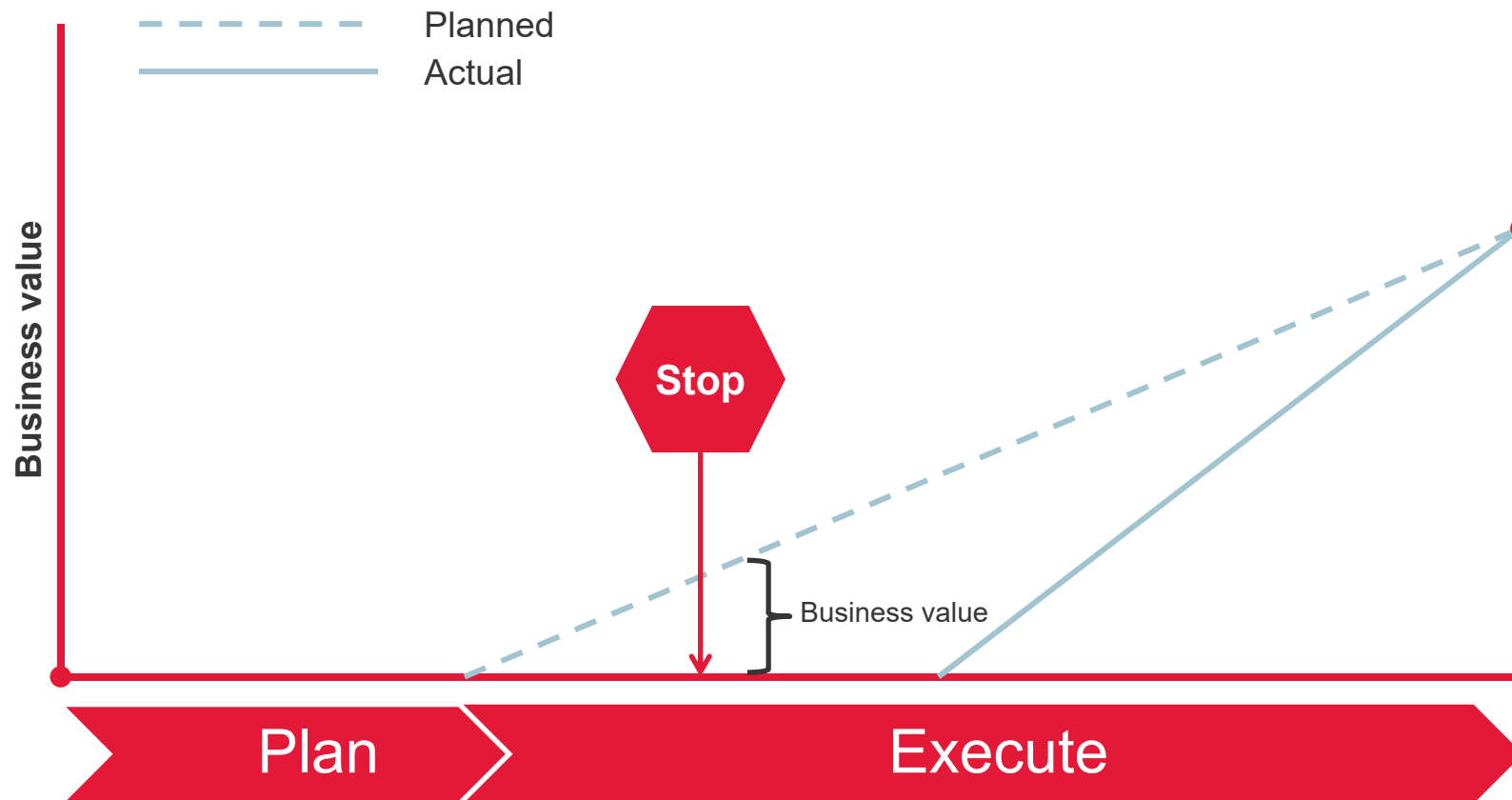# How to be more accurate with delivery?

- We have a backlog with high level features
- During the build features can change

- Is this a reasonable scenario?



**CGI**

7

# Agile deliveries are value driven projects

| Waterfall | Agile |
|-----------|-------|

|  | **Scope** | Costs | Deadline |
|---|---|---|---|
| Fixed | | | |

**Plan Driven**

**Value / Vision Driven**

|  | Costs | Deadline | **Scope** |
|---|---|---|---|
| Estimate | | | |

CGI
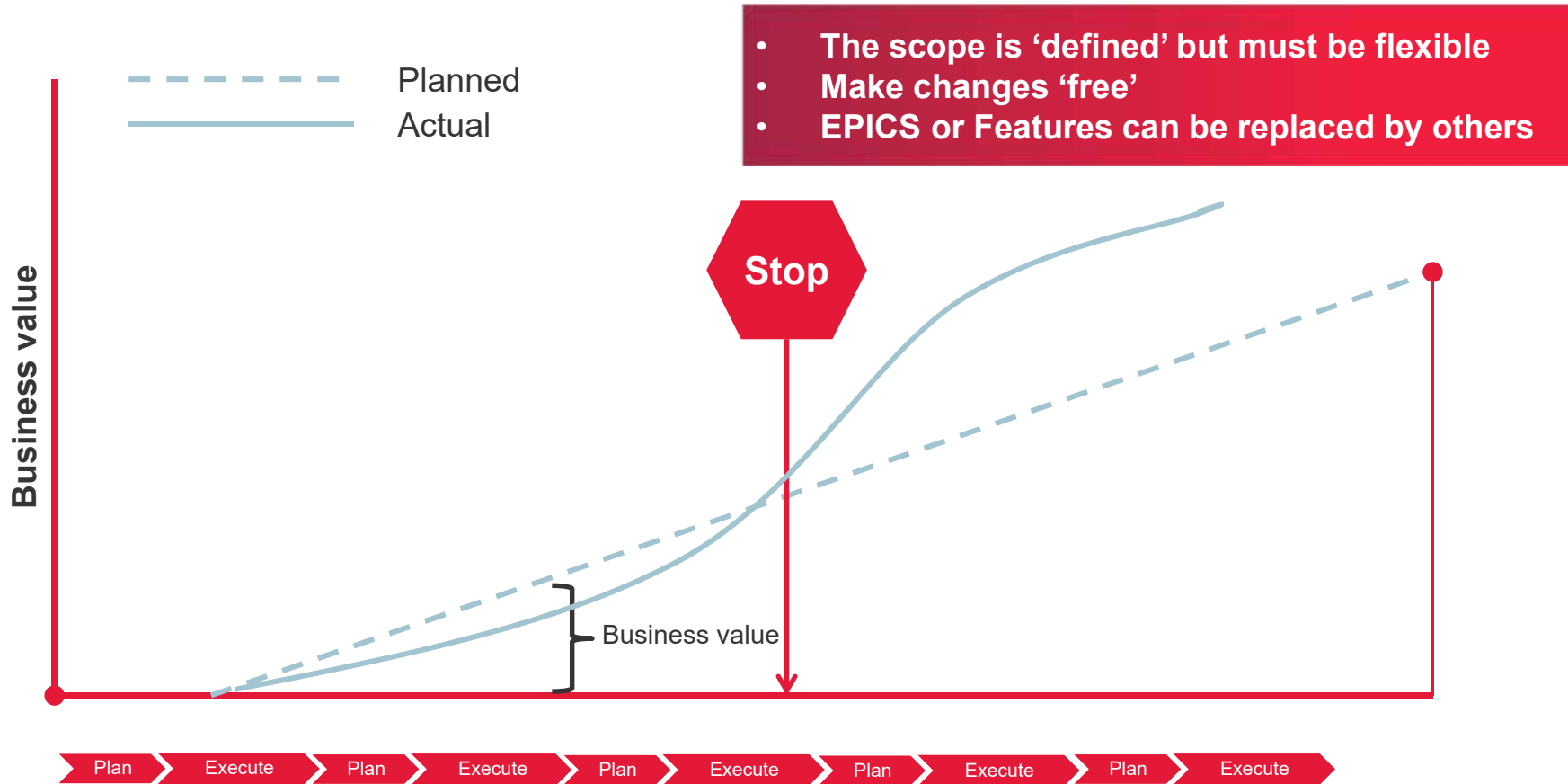
8

# Classic failure of the waterfall model

# Development in an Agile fixed-price context

# Development in an Agile fixed-price context



- - - - Planned
— Actual

- The scope is 'defined' but must be flexible
- Make changes 'free'
- EPICS or Features can be replaced by others

**Stop**

Business value

Business value

Plan ▸ Execute ▸ Plan ▸ Execute ▸ Plan ▸ Execute ▸ Plan ▸ Execute ▸ Plan ▸ Execute
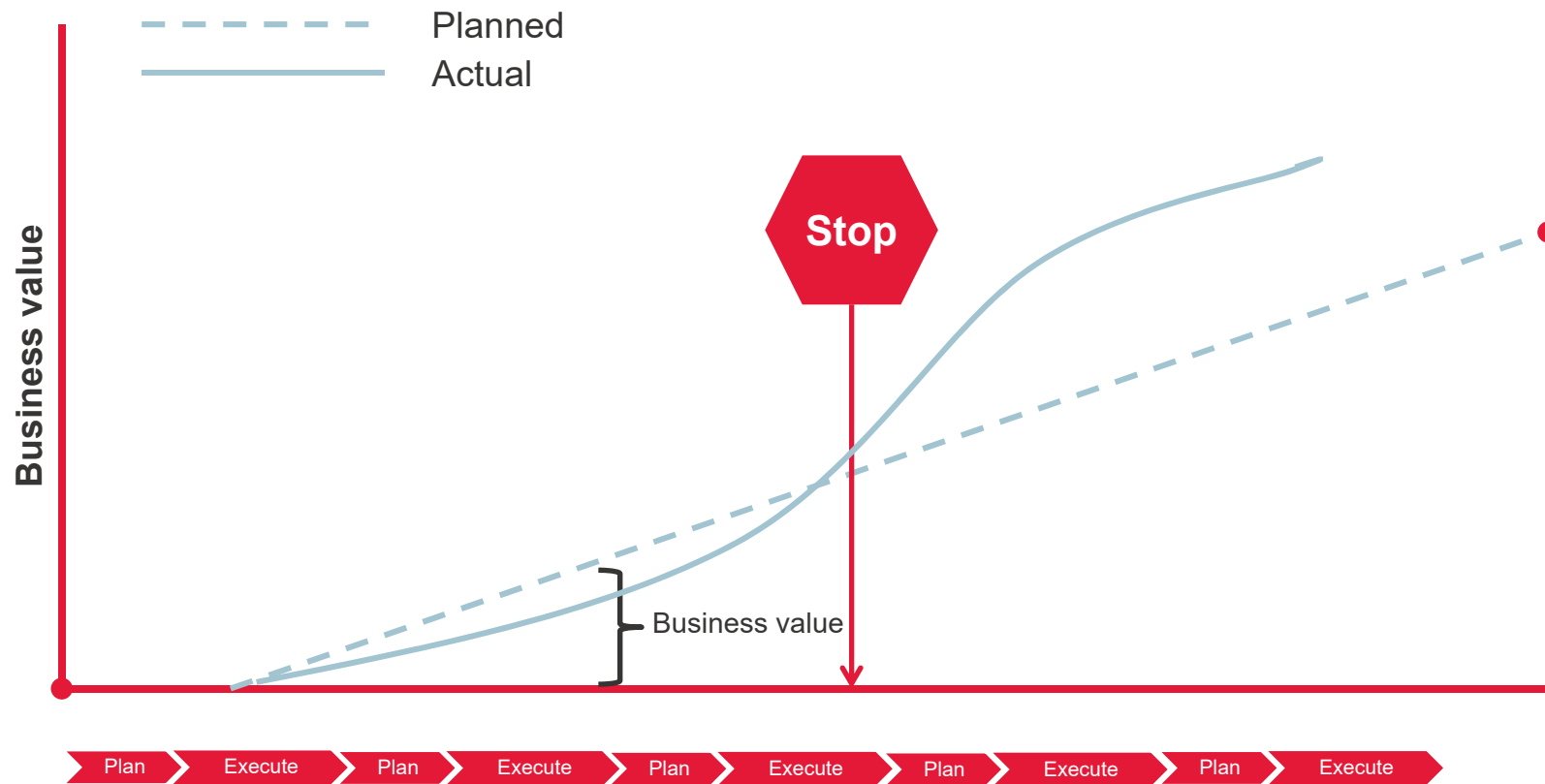
- Commitment for 60-80% of the 'defined' scope (must have)
- Make 20-40% of the scope flexible (would have)
- Define 20% of the scope as stretch objectives (nice to have)
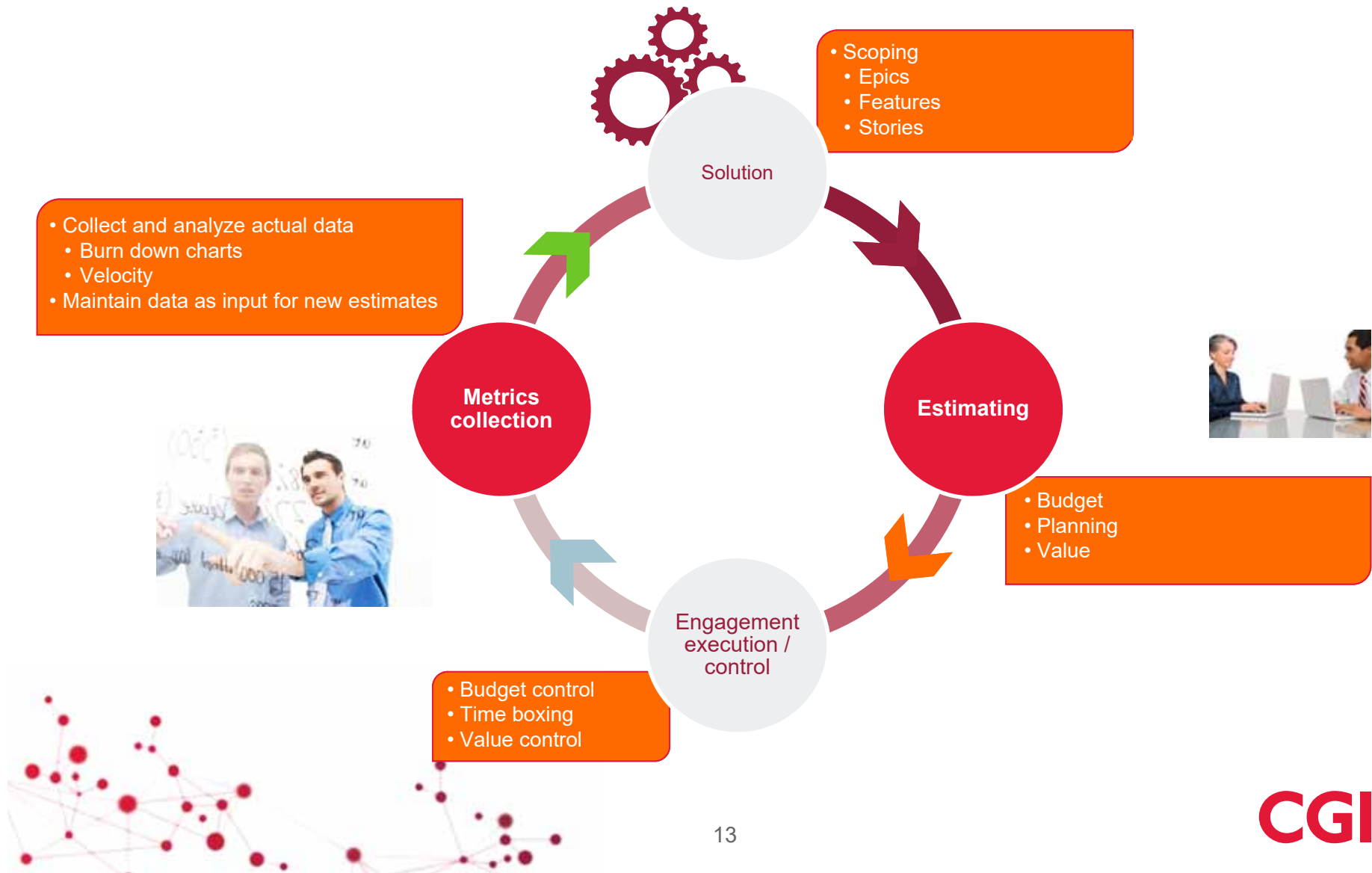
CGI

11

# A better way to manage the scope

- A program must start with an indication of what is desired
- A budget will be based on the known scope and identified risks
- The estimated budget will have a certain range (uncertainty)
- It's up to the program to <u>manage the scope</u>
- The team collaborates to deliver value that meets the desired objectives within the available budget

CGI

# Estimation in an Agile lifecycle

**Solution**
- Scoping
  - Epics
  - Features
  - Stories

**Estimating**
- Budget
- Planning
- Value

**Engagement execution / control**
- Budget control
- Time boxing
- Value control

**Metrics collection**
- Collect and analyze actual data
  - Burn down charts
  - Velocity
- Maintain data as input for new estimates

13

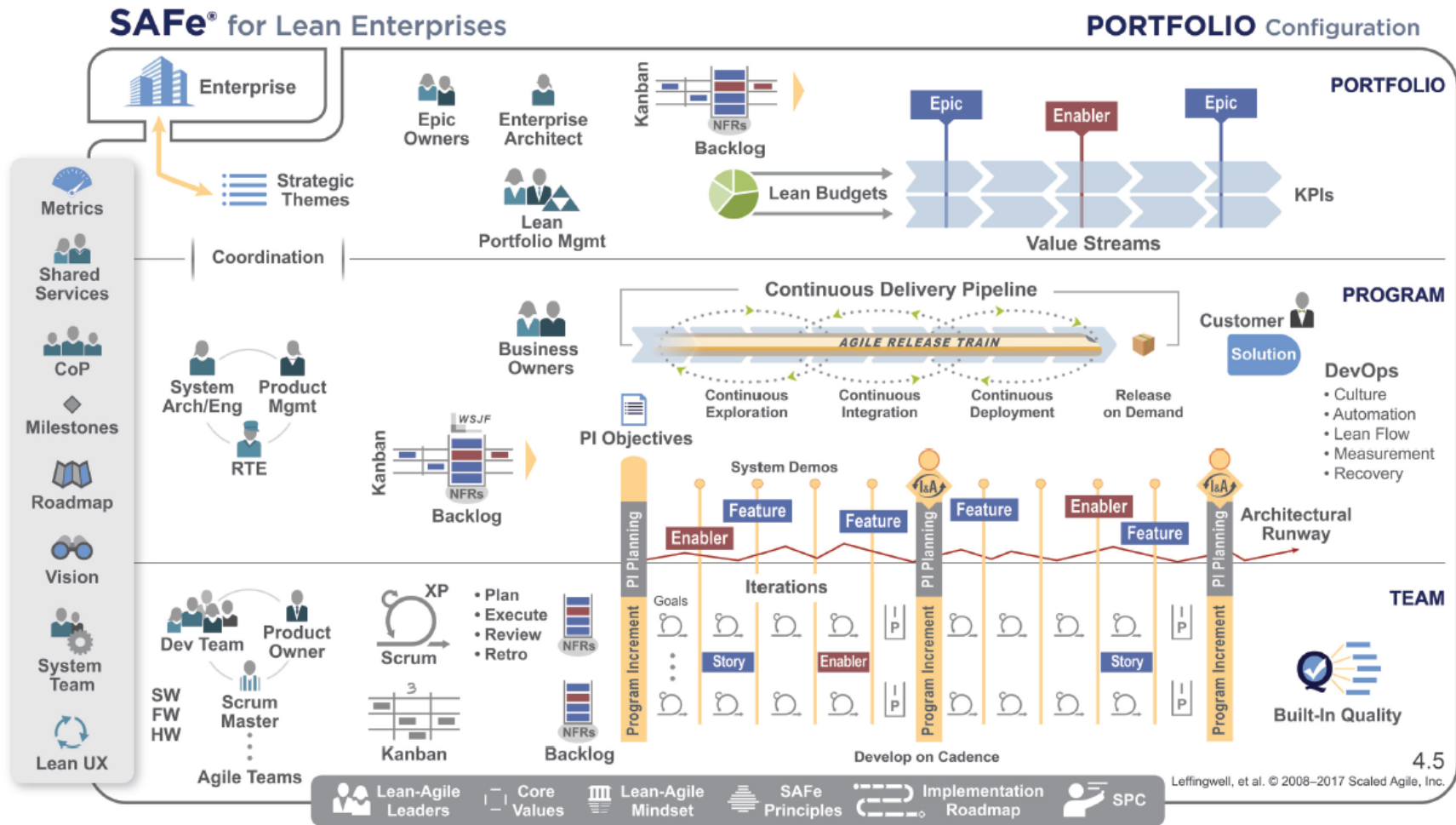**CGI**

# Agile scope break down per phase

- Budget / Roadmap:   -   Epic
  - Program Increment:  -   Feature
    - Iterations                    -   Story
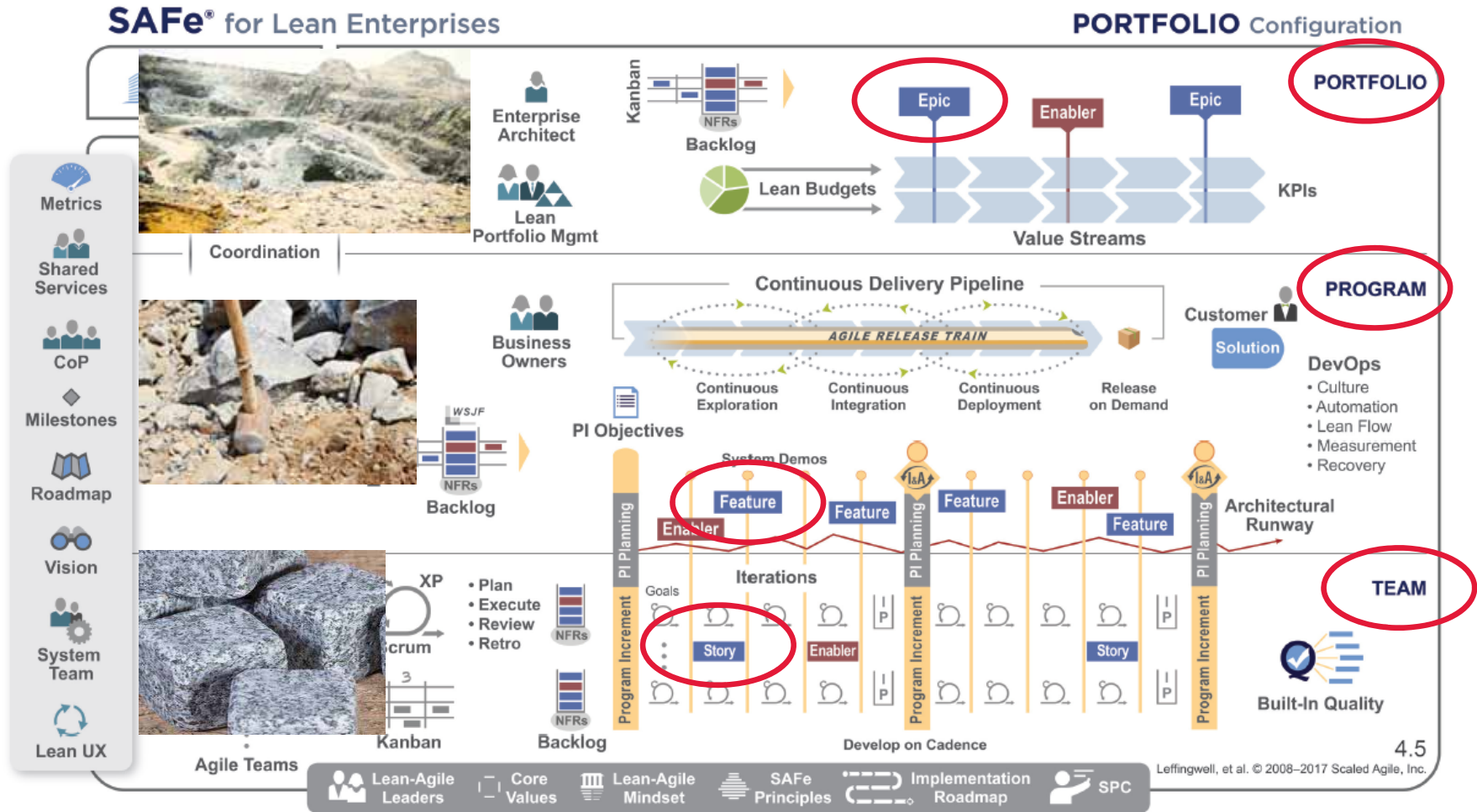      - Task/Subtask

CGI

# Scaled Agile Framework (SAFe®)

# Scaled Agile Framework (SAFe®)

# Can we easily change the scope?



Now let's try blue!
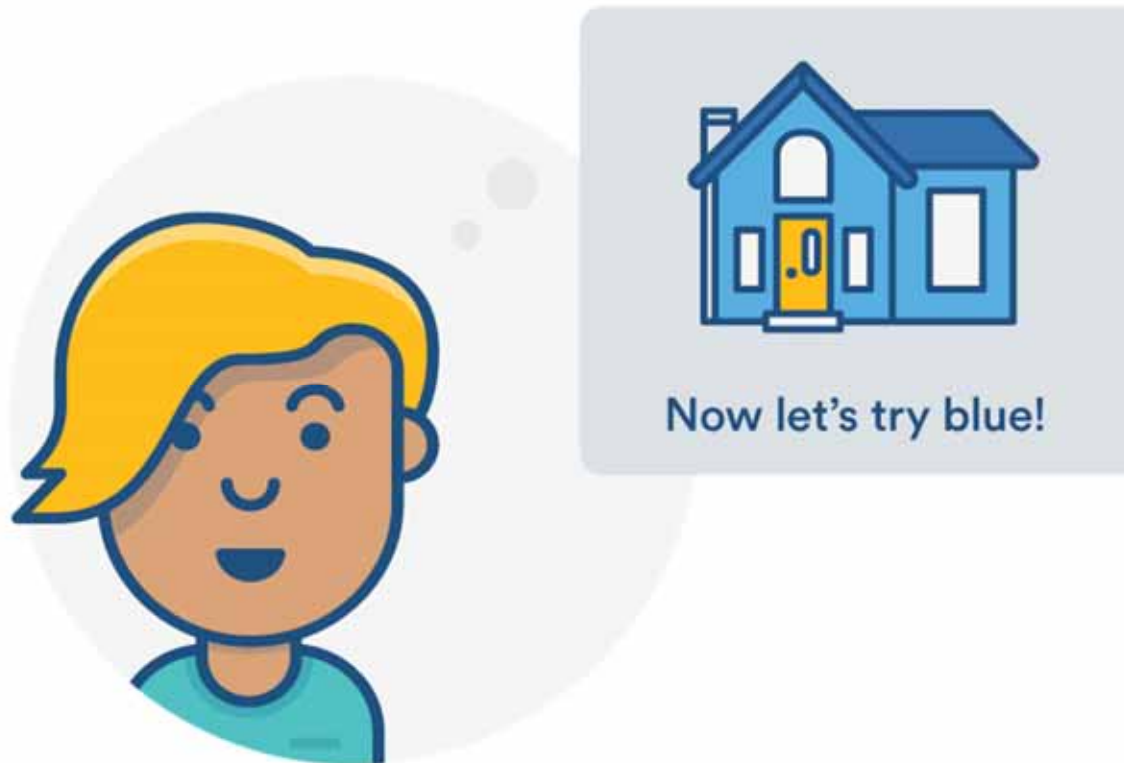
- Questions:
    - Is it reasonable to make this change?
    - What is the impact of making such a change?

17

**CGI**

# It depends on the Execution Horizon



- If we are determining the budget it's easier to make a change
- The change (e.g. color of the house) will be cheaper



- Closer to the end point, making a change has more impact
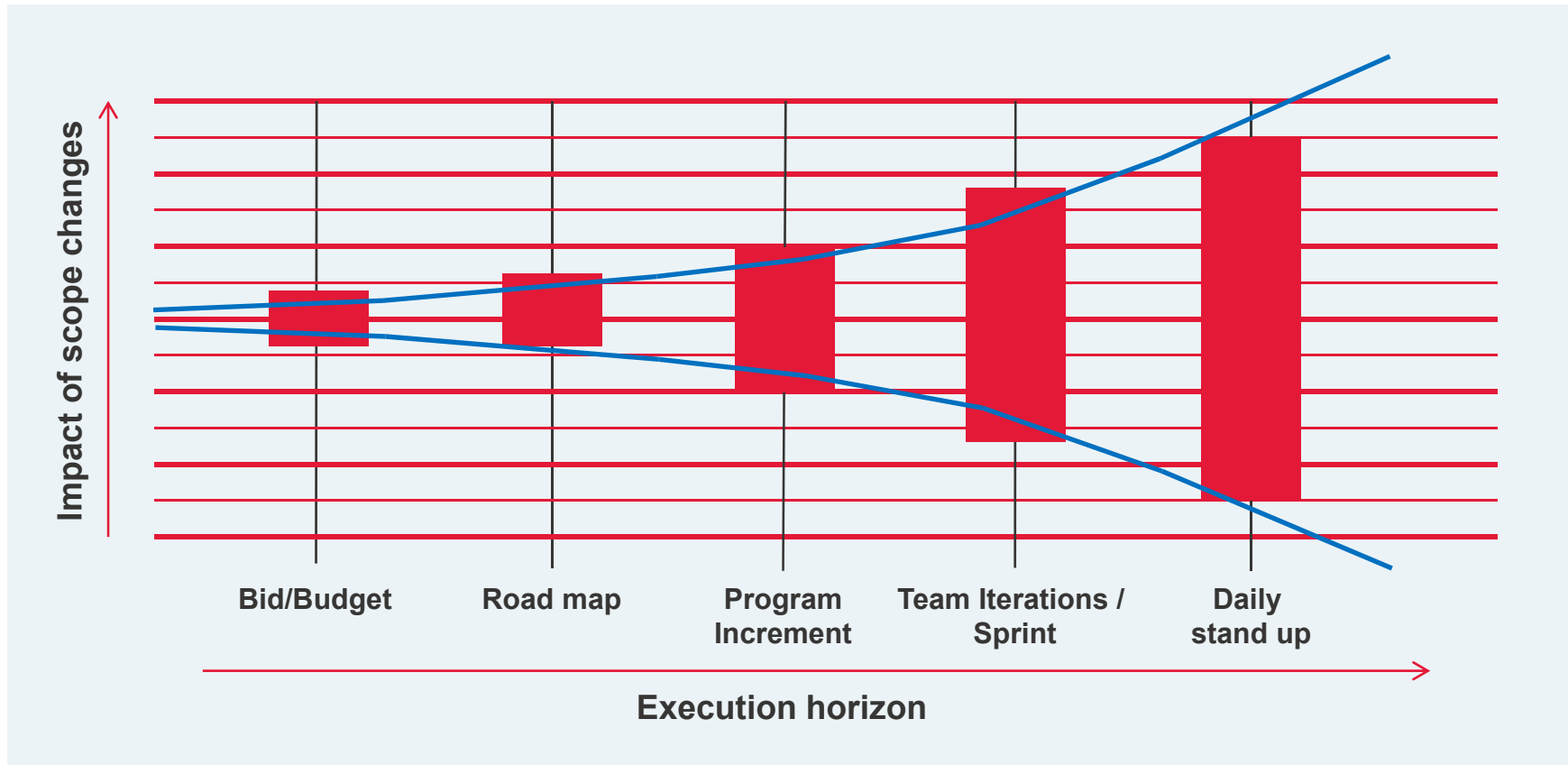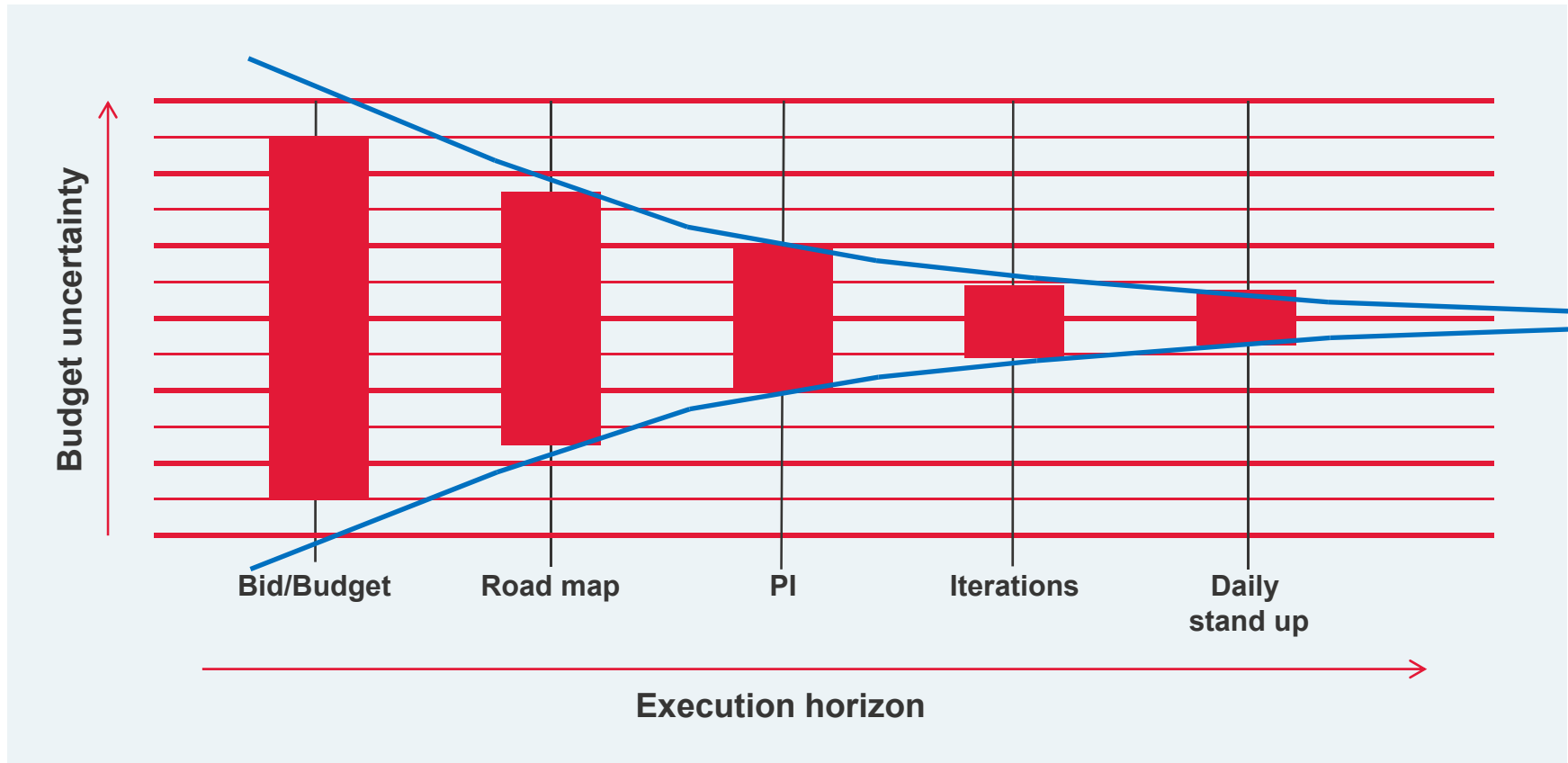- For example repainting the house will be very expensive

**CGI**

# Impact of scope changes vs execution horizon

# Budget uncertainty vs execution horizon

# Estimation Accuracy

| Estimate class | Primary characteristics | Secondary characteristics | | |
|---|---|---|---|---|
| | *Maturity level of solution definition (% of completion)* | *End Usage*<br>*Typical purpose of estimate* | *Methodology*<br>*Typical estimating method* | *Expected accuracy range*<br>*Typical variation in low and high range* |
| *Class 5* | 0 to 2% | Initial budget during bid start | ROM | L: -50% to +100% |
| *Class 4* | 1 to 15% | Solution selection and roadmap | EPIC; Unit cost; Parameters (high level) | L: -30% to +80% |
| *Class 3* | 10 to 40% | Start of a Program Increment based on backlog | Features; Unit cost; Parameters (assumptions) | L: -20% to +50% |
| *Class 2* | 30 to 75% | Team iterations, Detailed input | Stories; Unit cost; Parameters (detailed level) | L: -10% to +30% |
| *Class 1* | 65 to 100% | Daily stand-up | Stories; Task level | L: -10% to +20% |

Based on: AACE International Recommended Practice No. 56R-08; Rev. December 5, 2012; TCM Framework: 7.3 – Cost Estimating and Budgeting

21

**CGI**

## 5 Reasons for estimation in an Agile delivery

**1. Investment decisions**

- A proper cost estimate is required to determine the possible return on investment

**2. Compare alternatives**

- Multiple design alternatives will be created and compared based on costs

**3. Challenge vendor and contractor estimates**

- Cost estimates to challenge quotes and validate suppliers estimates

**4. Cost control**

- Estimates are required to decide if teams need to be scaled up or down to achieve objectives as well as the impact of scope changes

**5. Evaluation and benchmarking**

- To evaluate the performance of engagements against internal or external benchmarks

22

CGI
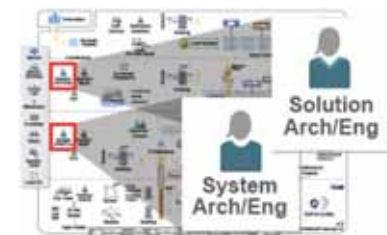
# Estimation techniques

- The following estimation techniques can be applied to agile deliveries:
  - Planning Poker / Story points
  - T-shirt sizing
  - Solution Based Estimation
  - Analogy Based Estimation
  - Function Point Analysis

# What are Story Points (SPs)?

- Story points are used to size the amount of work in a product backlog (stories)
- The story point size is used to
  - Estimate the implementation effort required
  - Determine the amount of backlog items that can be delivered in a time frame
- A point value is assigned to each story
- Raw point values are unimportant; what matters are the relative values
  - E.g.: 3 story points should be three times the work of 1 story point

- Team's should include what affects the effort in the estimate
  - Amount of work (activities)
  - Complexity of the work
  - Uncertainty in doing the work

24

**CGI**

# Planning Poker / Story Points

- Each member of a team is actively involved
- Product backlog user stories are presented by the product owner
- The story is discussed with the team
- Team members select a card that are revealed at the same time
- Team members with the highest and lowest card present their arguments
- The process is repeated till there is consensus

- All members of the Scrum team participate
- Planning Poker is time-boxed
- At the beginning, a reference story is selected and estimated with a reference value (e.g. start with a small story with the value 1)
- The following estimates are relative to this reference story



25

**CGI**

# 3.8 Planning Poker – Story Points - 2

- Planning poker serves two goals
  - Team commitment
  - Team consistent delivery
- Within a team story points must be comparable
- Story points between teams are not comparable

- Are Story Points (SPs) comparable with a standard like Function Points (FPA)?
  - No, Story Points is a relative size measure where FPA is an absolute size measure based on an ISO standard
- Is it possible to convert Story Points in Function Points?
  - No, because of the relativity of SPs



26

**CGI**

# T-Shirt sizing – Story Points

- Teams sometimes overanalyze when they estimate story points
- A non-numerical system like T-shirt sizing can be more effective
- A team estimates in what size category a story falls:
  - extra-small; small; medium; large, extra-large; double extra-large
- Each size category is converted in story points
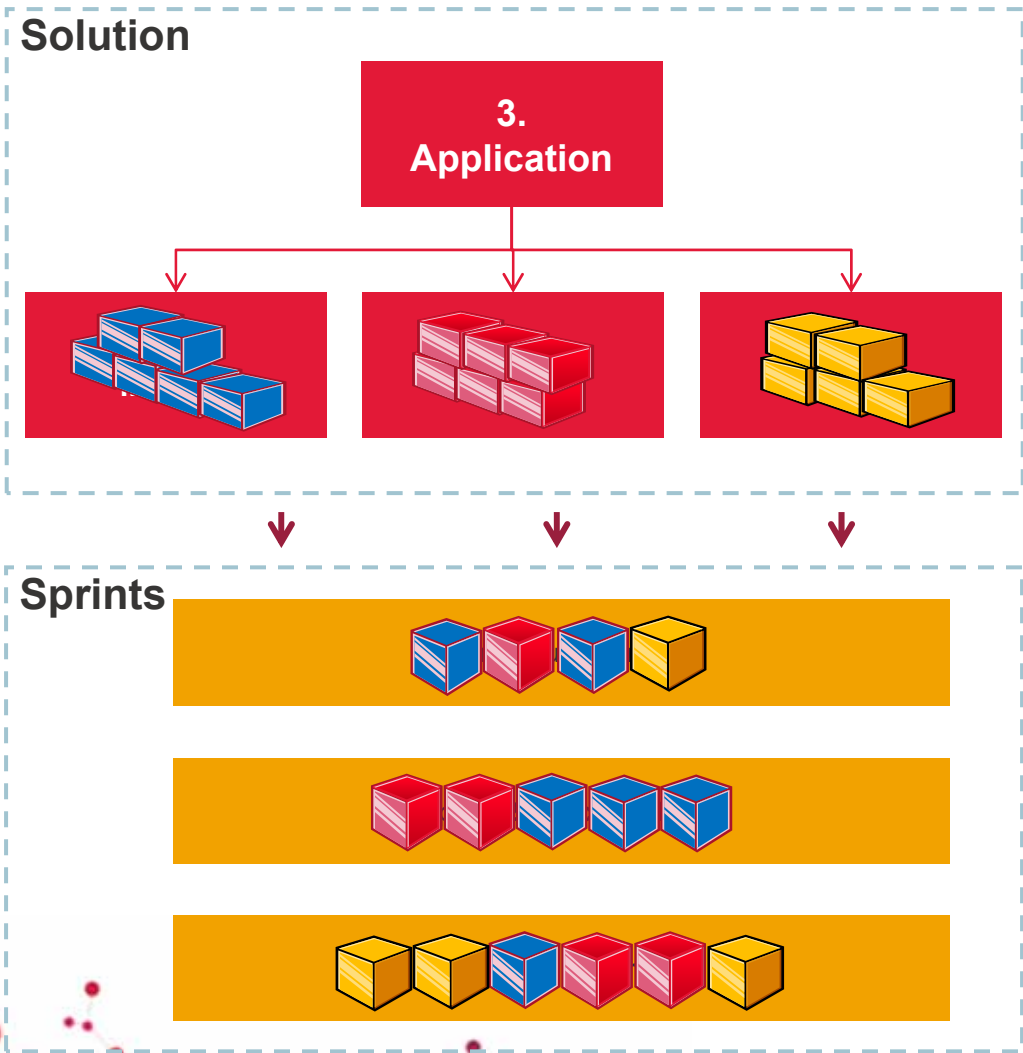
| T-Shirt Size | From | To |
| --- | --- | --- |
| Extra-small | 1 | 2 |
| Small | 2 | 4 |
| Medium | 4 | 8 |
| Large | 8 | 16 |
| Extra-large | 16 | 32 |

- Teams choose what amount of Story Points is assigned to a T-shirt size
- A practical issue is that non-numerical scales are generally less granular

CGI

# Solution Based Estimation

**Solution**

**3. Application**

**Sprints**

**Application Release**

The estimated costs for a solution is the sum of the estimated costs of the underlying solution elements

The application solution element is estimated based on the features assigned to that solution elements

The features will be realized in multiple sprints resulting in lower level solution elements

Actual costs of a solution elements and features can be collected on release level if related sprints are completed

28

**CGI**

# Analogy Based Estimation
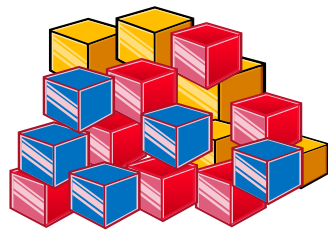
**Portfolio Backlog**

**Solution Breakdown Structure**

In case a same type of solution has been build before, the historical data could be used to perform an analogy based estimation

**Historical data**

**CGI**

# Function Point Analysis (FPA)

- Function Point Analysis is used to size applications against an ISO standard
- The sizing is based on Transactions and Logical Files
- Function Points (FPs) is about size, not about effort, cost or duration
- The FP size is input for the estimation of effort, duration and costs
- The mostly used standards for FPA are
  - IFPUG (www.ifpug.org)
  - Nesma (www.nesma.org)



**CGI**
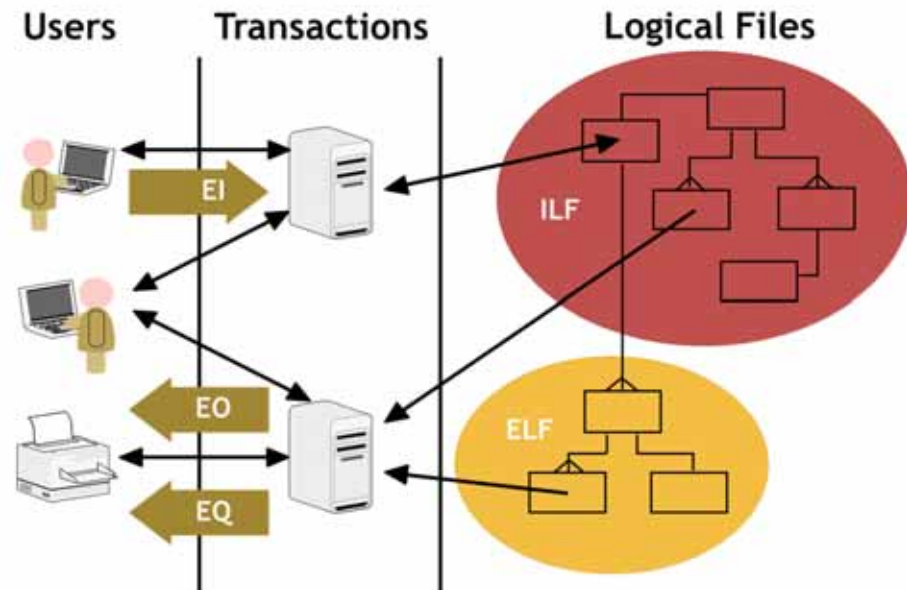
# Structure of a Scaled Agile delivery

**Portfolio Backlog**

Portfolio epics are
assigned to programs

**Portfolio**
- EPICS with business needs
- Sizing based on EPICS
- Size by means of story points
- Sizing in FPA (market conform)

**Program  Backlog**

Program features and enablers
are accepted by teams

**Program**
- Features / Enablers
- Size by means of Story Points
- Sizing in FPA (market conform)
- Use of historical data

**Team  Backlog**

**Team**
- Estimation based on stories
- Estimation of tasks (optional)
- Maintenance
- Technical depth

31

CGI

# Functional Size for scaled Agile deliveries

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

**Portfolio Backlog**     **Program Backlog**     **Team Backlog**     Scrum     Built-In Quality

| **Portfolio** | **Program** | **Team** | **Iteration plan** |
|---|---|---|---|
| - EPICS with business needs | - Features / Enablers | - Estimation based on stories | - PDCA |
| - Sizing based on EPICS | - Size by means of Story Points | - Estimation of tasks (optional) | |
| - Size by means of story points | - Sizing in FPA (market conform) | - Maintenance | Develop on cadence |
| - Sizing in FPA (market conform) | - Use of historical data | - Technical depth | Release any time |

**Actions:**
- Define proxies (Functional sizing)
- Define costs / FP and criteria

**Actions:**
- Define tooling for control and actual data collection
- Define how to measure the actual velocity

**Actions:**
- Gather metrics (release level)
- Provide metrics for new estimates

32

**CGI**

# Conclusions

- Scaled Agile deliveries require a different management approach
- Scaled Agile deliveries require a different estimation approach
- A combination of estimation approaches can be applied
- Metrics are essential for historical data but also to manage the delivery
- Cost estimation doesn't reduce the need to focus on value
- The use of Agile deliveries doesn't reduce the need to estimate the costs

*Thank you!*

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

**CGI**

Experience the commitment®

*Questions?*

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

*Backup slides*

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

**CGI**

Experience the commitment®