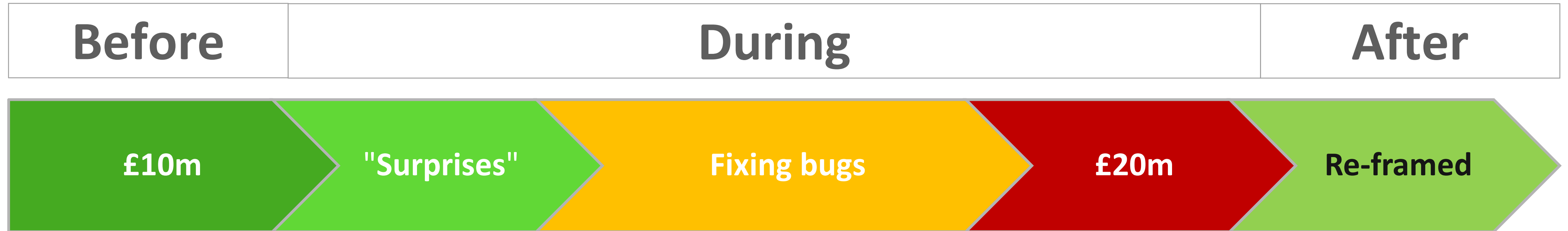


The Economics of Automated Software Sizing

Functional sizing of software has been proven as a reliable means of costing software projects. In this presentation I will describe, the nature, speed and precision of automated functional sizing from a written set of requirements for early cost estimation. I will then describe the substantial economic benefits of doing so.

Mental models more than hard numbers



Unknowns

Quality



Described by executives as a qualified
“success”

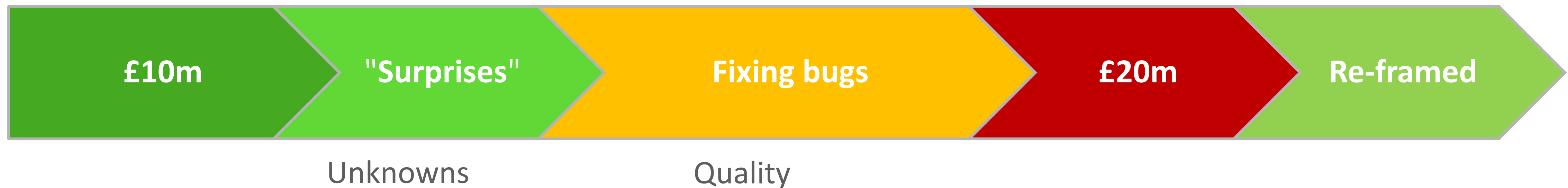


What is the total consequential impact of the wrong decision?



1. The increase cost
2. The lost benefits
3. The opportunity cost vs a different path

Budgeted: £10m on the basis of £10m bens p.a. for 5 years. (net ben £40m).



1 year delay, doubled cost:

Increase cost £10m,

+ one less year of benefits £10m

+ subsequent benefit is 25% lower p.a. £10m

+Opportunity cost £10m

Total consequential cost £40m

4x

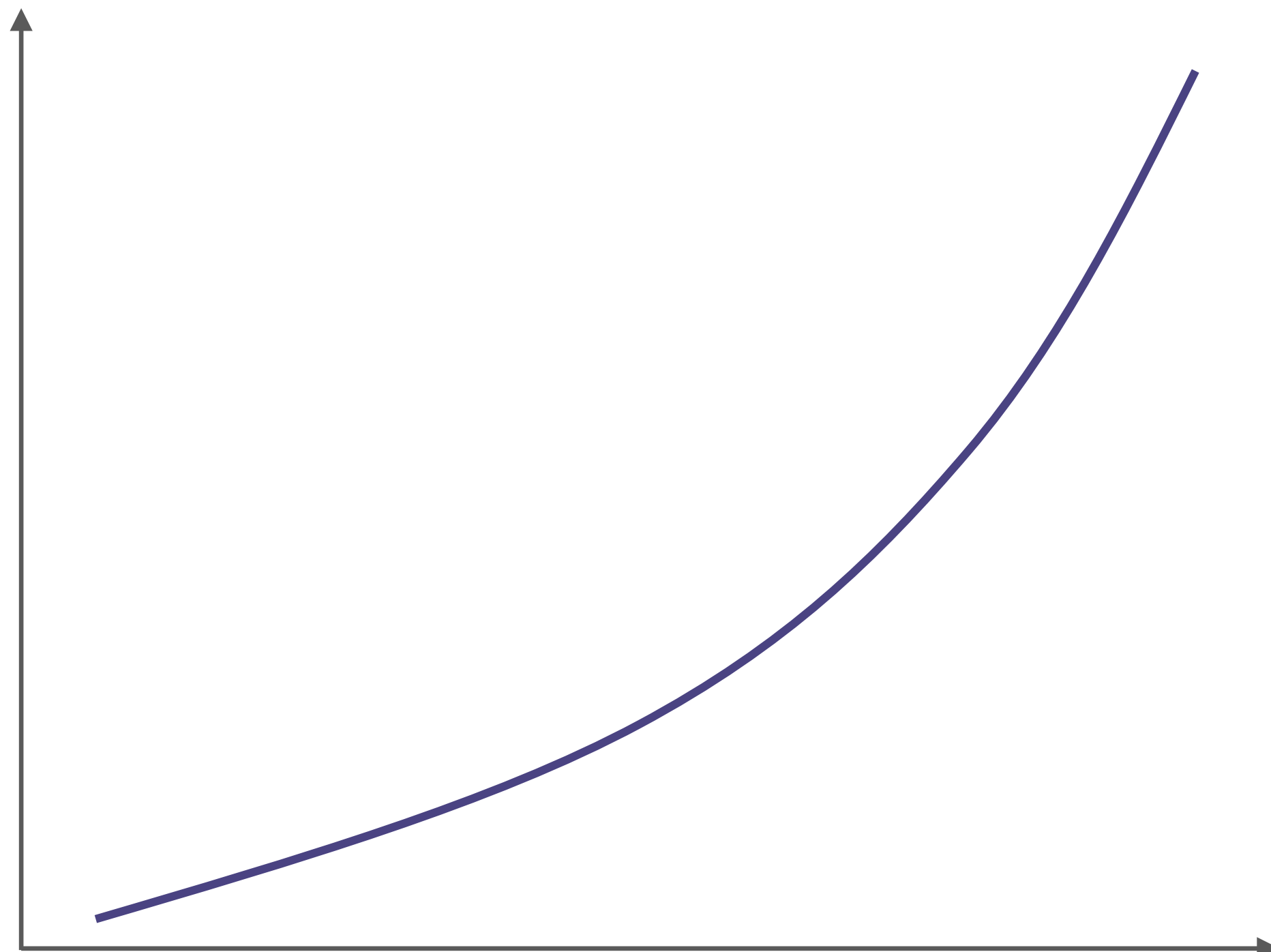
**Real Cost was 4
times the initial
estimate!**

Why do Software projects go over budget and schedule?



- Poor executive engagement
- Too ambitious (size)
- Poor requirements
- Poor project mgt (scope & risk)
- Poor attention to quality (esp early)

Cost & Risk of failure




Size




small is beautiful

What is small..?

Articulated functional requirements
“knowns”

M 

Unarticulated functional requirements
“unknowns”



Unknowables

Non Functionals

Other

We tend to:

Estimate this

And add a bit for this

Total Functional Size Estimate			
Sized requirements	58	259 CFP	← Functionality Found
Ambiguous requirements (ie. no functionality detected)	59	263 CFP <i>Estimated</i>	← <i>Inferred</i>
All functional requirements (sized + ambiguous)	117	522 CFP <i>Estimated</i>	
Potential missing requirements (from CRUD analysis)	130	423 CFP <i>Estimated</i>	← <i>Inferred</i>
Total Potential Size (sized + ambiguous + missing)	247	945 CFP <i>Estimated</i>	← <i>Inferred</i>

📌 CFP = COSMIC Function Points

Knowns

Knowable Unknowns

ScopeMaster provides traceable evidence for cost estimators and project managers of the likely final cost – BEFORE investment commitment.

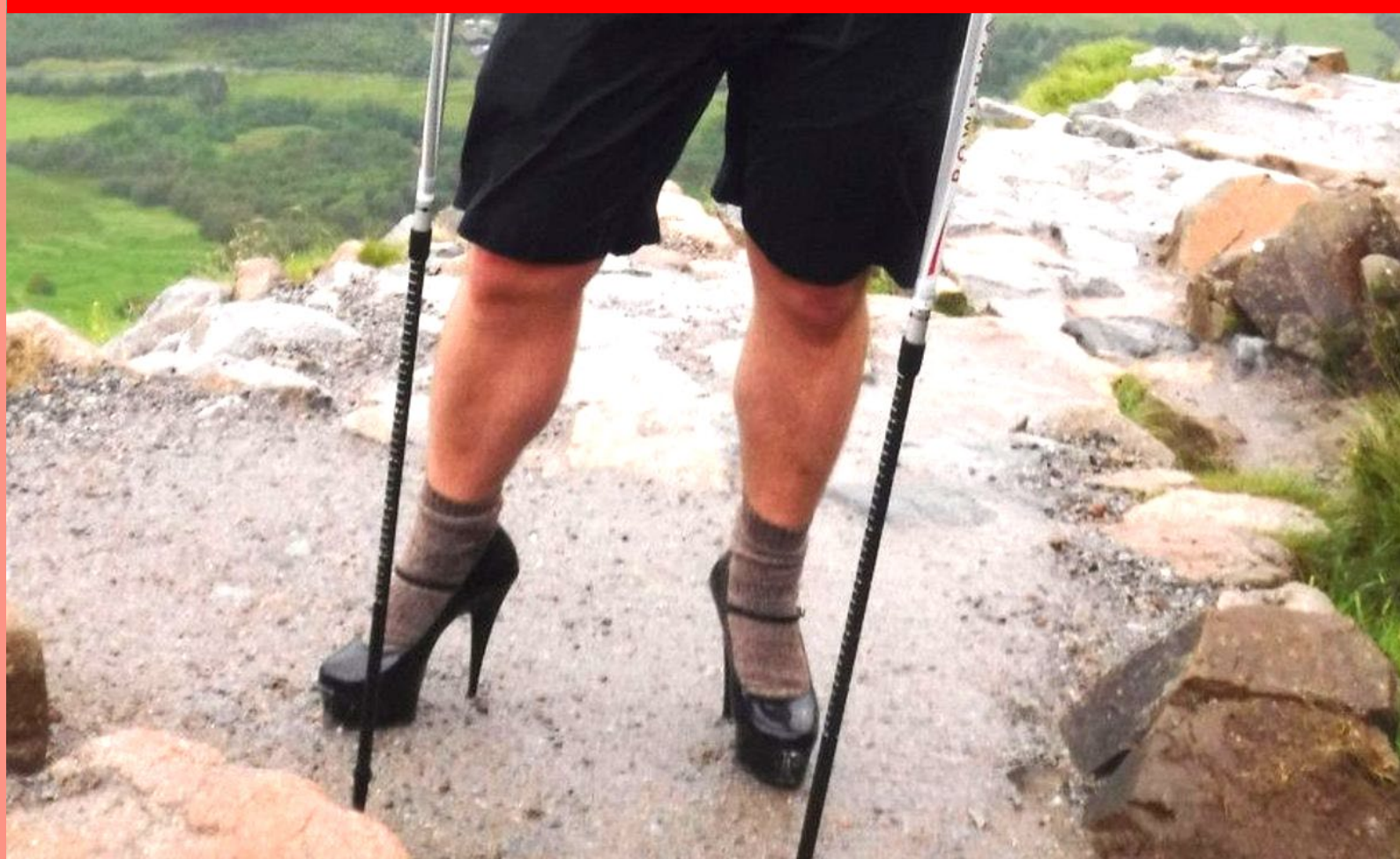
PBIs
Story Points
Flow metrics
Planning poker
Scaled Agile Framework (SAFe) ...
Scrum@Scale (SaS) ...
Large Scale Scrum (LeSS) ...
Nexus. ...
Disciplined Agile (DA) ...
= very little help to a cost estimator



Be rational, equip yourself for success...



“No estimates”



**Functional Sizing
(manual)**



Story Points or Story Counts



**Automated Functional analysis
and sizing**



“We learn as we go”

“We need to know the likely cost and duration”

They Say	We say
We learn as we go	Most unknowns are knowable*
It's wasteful to estimate prematurely	Almost effortless*
Estimation is BRUF	Sizing can be done at high granularity
Estimates are abused by managers	Story points & counts, yes, but not CFP
Estimation is a distraction	Estimation is automated
Focus on flow and flowmetrics	Addresses progress not size
Build and focus on value	But, it might not be worth starting
Don't know when it will be ready	We can
We can't tell you the cost	We can

*If using automated sizing & analysis

Automated Software Backlog Analysis

Functional Software Requirements

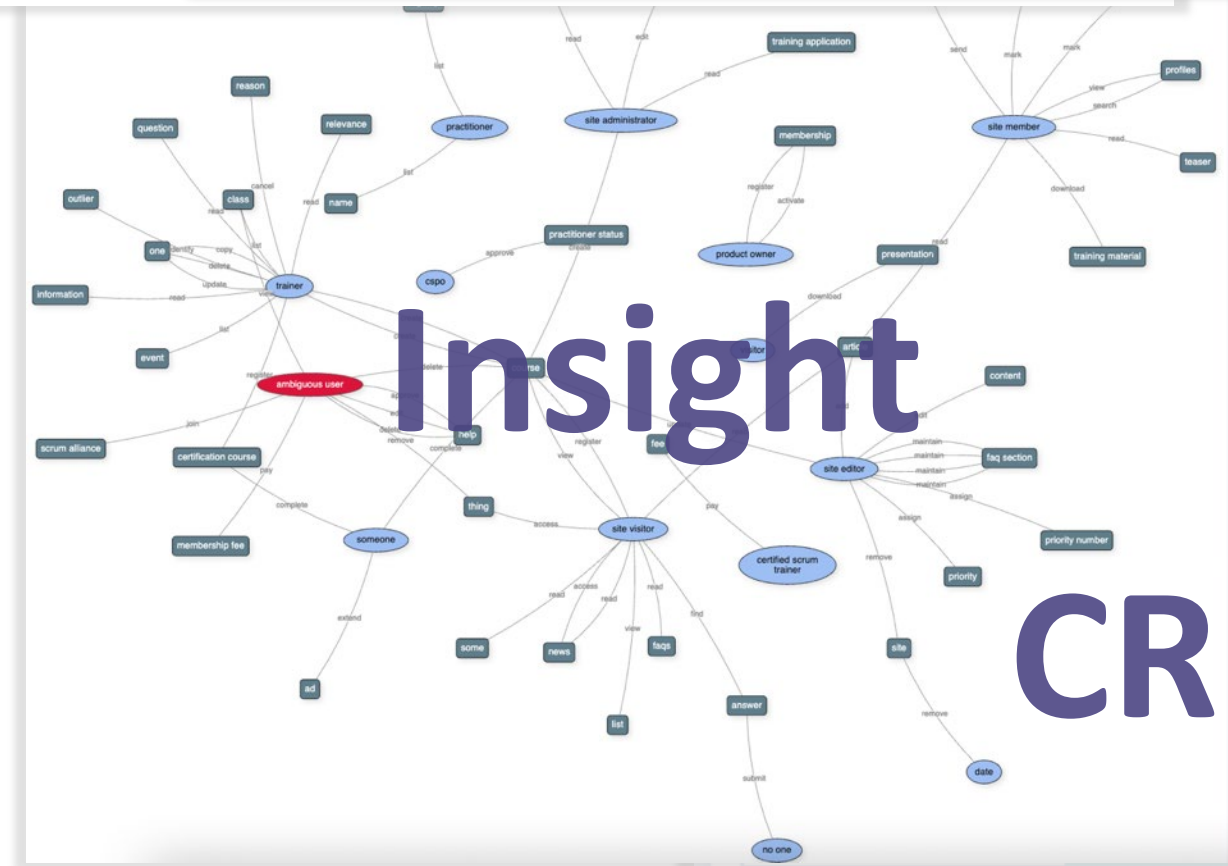
Direct or CSV or API

Analyse 2399 stories

Total Functional Size Estimate

Sized requirements	47	209 CFP
Ambiguous requirements (ie. no functionality described)	51	227 CFP <i>Estimated</i>
All functional requirements (sized + ambiguous)	98	436 CFP <i>Estimated</i>
Potential missing requirements (from CRUD analysis)	110	358 CFP
Total Potential Size (sized + ambiguous + missing)		

Functional Sizing



Insight

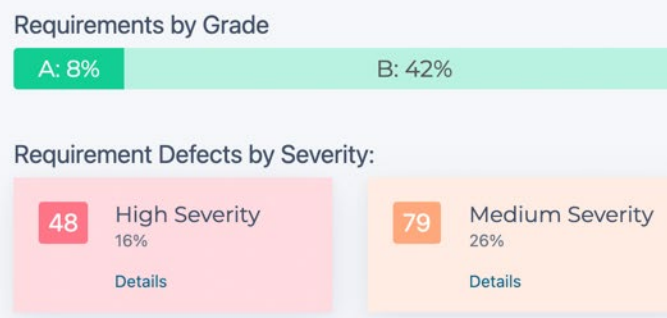
CRUD Analysis

Test Scenarios

```
***** POSITIVE TESTS *****/  
  
/** Positive test for edit site member profile **/  
  
INPUT edit(use VALID site member profile ID, VALID site member profile attributes){  
  assert that site member profile data is accepted;  
  assert that site member profile data is within acceptable boundaries;  
  assert that site member profile data contains no security risks;  
  assert that site member profile is created successfully;  
  assert confirmation message;  
  assert navigation is correct;  
}  
  
***** NEGATIVE TESTS *****/  
  
/** Negative test for edit site member profile using invalid site member profile ID **/  
  
INPUT edit(use INVALID site member profile ID, VALID site member profile attributes){  
  assert data is rejected;  
  assert site member profile not stored ;  
  assert error message;  
  assert error was logged ;  
  assert navigation is correct;  
}  
}
```

Design Inference

Requirements Quality Report



Q.A.

102 Functional requirements. 2.94 Defects per requirement: (300/102) exc advisories. 1.2 Defects per CFP: (300/249) exc advisories.

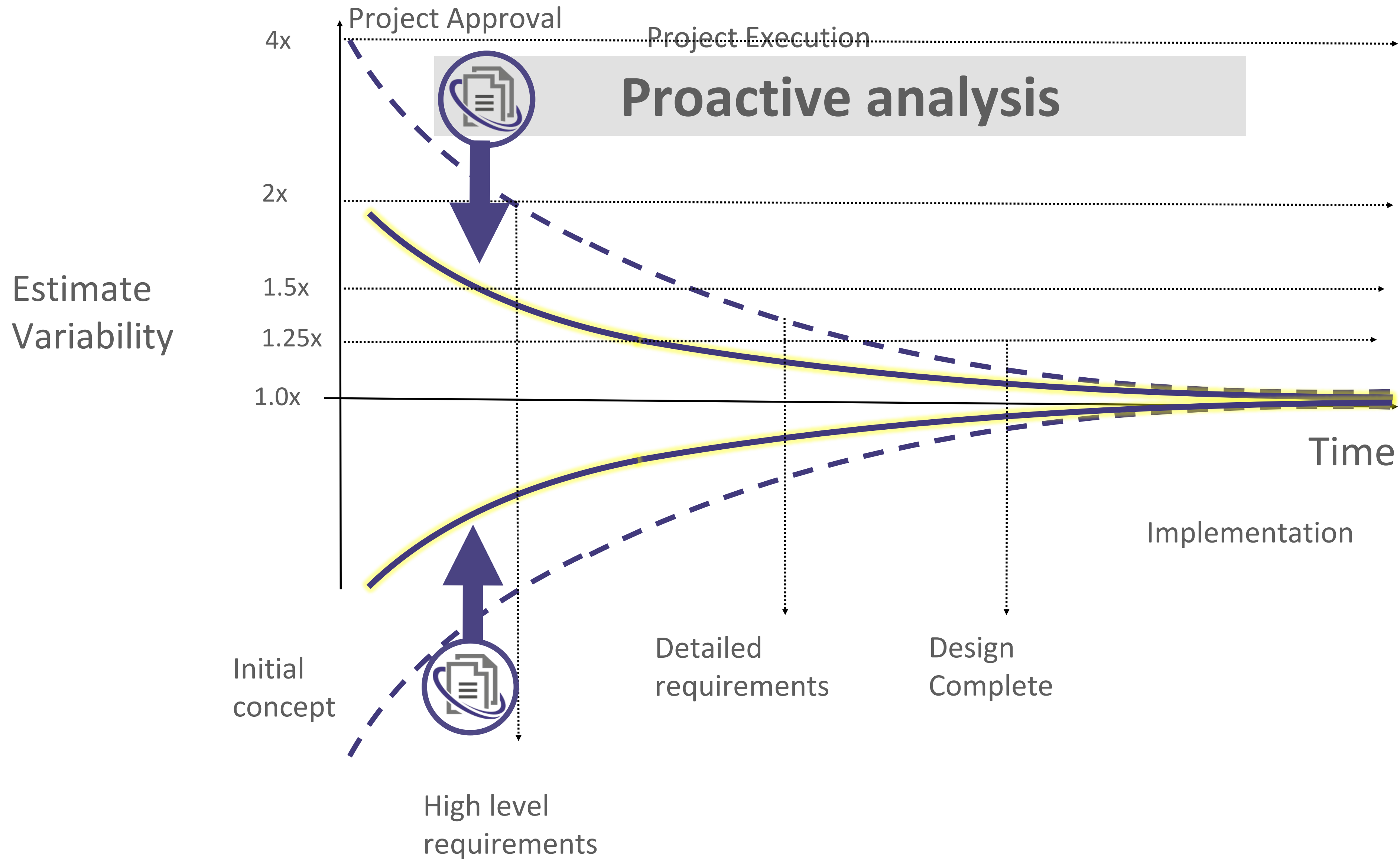
By Requirement

AWS-102 add ad Sprint:1 Assoc B

As a scrum practitioner I want to add ad

- CONCISENESS Concise, ratio of words per CFP is 2 (<10 is good). Total word count is 9 (<50 is good).
- FUNCTIONAL SIZE Suitable size 4 CFP
- POTENTIAL MISSING Potentially missing from this set of requirements: Read ad, Delete ad,
- BENEFITS No stated benefits

Only proactive analysis improves early estimate accuracy





1 .Size matters

**2 .Equip yourself with
the best metrics for size**

**3 .Let automation do the heavy
lifting and provide objective
assessment.**

And now let us consider requirements quality....

Business analyst

Dev

Stakeholder

Architect

UI

Tester



A misinterpreted user story is the birth of a bug

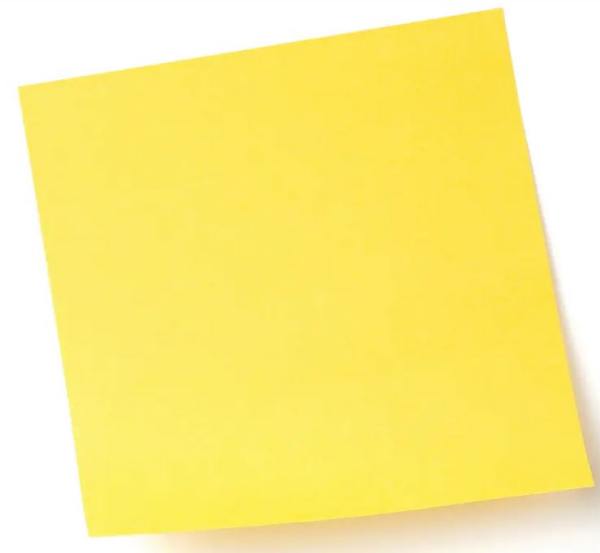
Re-coding a user story because it was misinterpreted, causes it to cost **2.5x**

So, requirements **quality** is a direct and major factor in **cost**.

Cost if right
first time
£n

Cost if
reworked
**£2.5n +
delay**

- 5.5 CFP (@ £500 - £2000)
- Typically has 3-5 requirement defects.



Each Story

**Clear
Functional
Intent:**

Reduces misunderstanding,
avoidable discussions and rework.

Quality:

Ambiguity
User-oriented
Completeness
Sizeability
Testability

Sizing:

Consistent
< 15% of a manual sizing – (verified)

Speed:

Consistent
< 15% of a manual sizing

Insight:

Instant feedback = Agile



Cross references all functionalities
12 insightful reports and diagrams
Consistency
Completeness

Estimates omissions & Ambiguities

Backlog of 100 in 2-3 minutes

Insight – improvements – better overall estimates



Each Story

Search (Bad story example) US10 Ambiguous functional size: ?

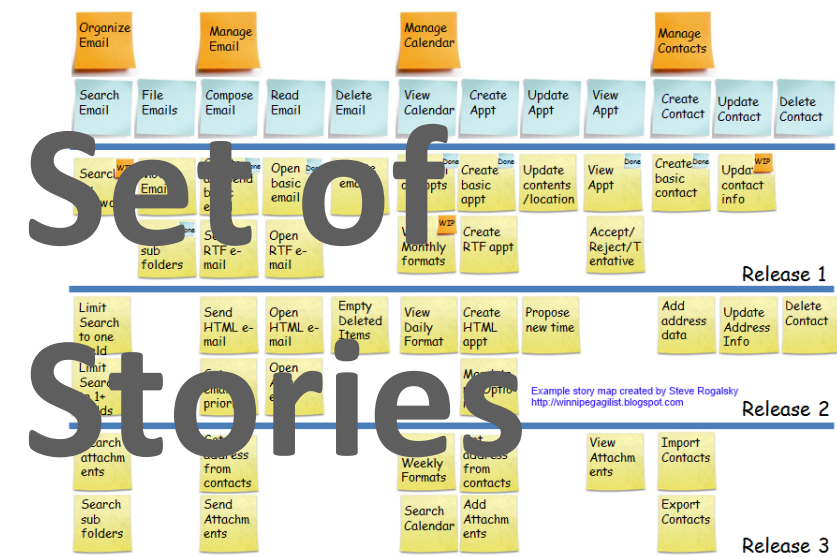
Whenever I am clicking on the cart tab, I go to a page that that has a title, shows a search bar and has a list of things I might want to buy. It's really fast and shows how many things that it found and the speed. I only want this developed in Javascript so that it seems really fast. Products that are not in inventory still show up. This needs to be in sprint 2 so the customer can see what it might look like early on. test that the prices are correct.

Edit Analysis **Quality E** Comments More

Quality E [About the quality scores](#) 571 tests run

High severity

- H FUNCTIONAL INTENT** No clear functional intent detected. Use clear functional verbs.
- M CONCISENESS** Try and reduce the number of words (93)
- M OBJECTS COUNT** Contains more than 7 potential objects, perhaps too large or complex
- M SO THAT** Functional requirement contains *so that* statement, recommend separating functionality from benefits
- M VERB CONCERN** Contains a verb that can be easily misinterpreted **see**, instead use the recommended ones.
- M VERB COUNT** Contains many verbs (25), consider reducing: am, clicking, go, has, shows, has, might, want, buy, 's, shows, found, want, developed, seems, are, show, needs, be, can, see, might, look, test, are
- L BENEFITS** No stated benefits
- L CODING ADVICE** Avoid describing how the functionality will be built **java**
- L CODING ADVICE** Avoid describing how the functionality will be built **javascript**
- Q ATTRIBUTES FOUND** Is **bar** an attribute or property of *search*? Try to avoid mentioning attributes in user stories.
- A AMBIGUOUS ADJECTIVE** Ambiguous terms can lead to misinterpretation: **fast**
- A AMBIGUOUS ADJECTIVE** Ambiguous terms can lead to misinterpretation: **many**
- A AMBIGUOUS AMBIGUITY OF REFERENCE** Ambiguous terms can lead to misinterpretation: **it**
- A AMBIGUOUS DANGLING ELSE** Ambiguous terms can lead to misinterpretation: **might**
- A AMBIGUOUS IMPLICIT AMBIGUITY** Ambiguous terms can lead to misinterpretation: **still**
- A AMBIGUOUS VERB** Ambiguous terms can lead to misinterpretation: **might**
- A GENERAL AMBIGUITIES** Avoid ambiguous terms **like**
- A NAVIGATION** Avoid navigation descriptions within the functional requirement **click**
- A OBJECTS CONFIRMED NONE** Contains no confirmed objects



Requirements Quality Report

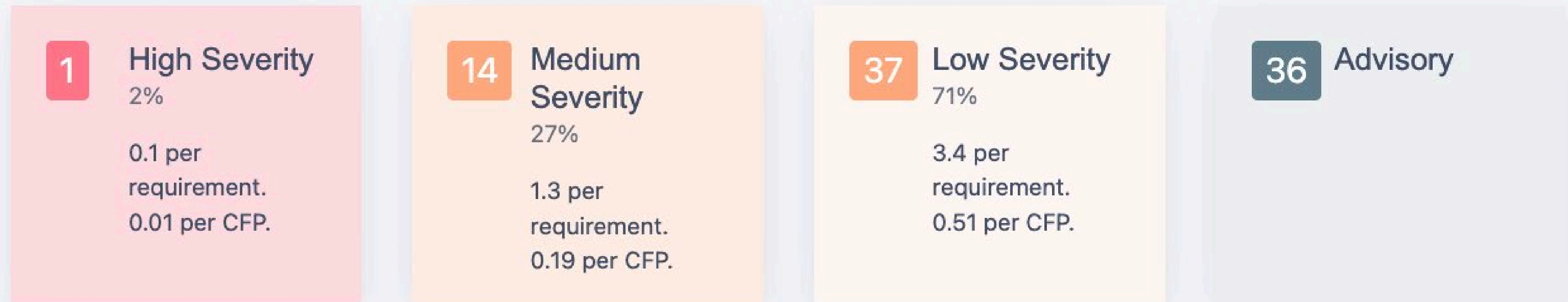
52 Defects. 4.73 Defects per requirement: (52/11) 0.71 Defects per CFP: (52/73) from 11 Functional requirements.

Overall quality: **E**

Requirements Quality



Defects by Severity: 52 + advisories



- **Utility project** reduced rework by over 50%.
- **Govt project** Sized and exposed 3000+ defects in 1 day.
- **Banking project** saved 50% vendor negotiations.
- **Defence project** circa 50% direct savings in 7 days.
- **Finance project** exposed 300% overcharging by vendor.

Sometimes unwelcome...

Automated Backlog Analysis

Functional
Sizing

Requirements
Insight



Instantly expose quality issues that affect cost and schedule

Objective, defensible, estimates, requiring negligible tech input



Benefits

The estimated benefits of using ScopeMaster, both direct and consequential.

- 2399 requirements
- Estimated cost: \$4,862,800 - \$14,588,400
- Least benefit: \$975,262

	Impact	Lowest Expected Benefits
1. Effort Saved	499 - 1792 days	\$244,598
- finding & fixing defects (5484)	78 - 1371 days	548 hours x 70 = \$38,360
- sizing (12157 CFP)	87 days	607.85 hours x 70 = \$42,550
- generating tests (5846)	334 days	2338.4 hours x 70 = \$163,688
2. Rework Avoided	7 - 11%	7% x 12157 CFP x 400 = \$340,396
3. Resource/Vendor Mgt	4 - 15%	4% x 12157 CFP x 400 = \$194,512
4. Faster to Value	3 - 6%	3% x 12157 CFP x 400 = \$145,884
5. Avoided Late Discovery	1-2%	3117 defects x 1% x 4 x \$400 = \$49,872
Total Benefit		\$975,262
Benefit per requirement (2399)		\$407

Direct benefits

Effort saved

+Better early decisions

Consequential benefits

Rework avoided

Faster to value

Avoid late defects

The least we have ever observed
just from knowing the size.

10 - 400%
of initial estimate

The typical direct and
consequential benefit of better
early estimates achievable with
automated analysis.