ScopeMaster

The Symbiotic Relationship Between Software Sizing and Requirements Quality

Colin Hammond
35+ years in IT, Founder of ScopeMaster
Colin.Hammond@scopemaster.com

KPMG, Royal Mail, Norwich Union, British American Tobacco, Waitrose
pwc, Pinpointers, Equitable Life, Sainsbury's
Ashridge, The Open University, Symantec.
Homebase, beazley, + more

Symbiotic relationship

Symbiotic Relationship is any type of a close and long-term biological interaction between two different biological organisms that is mutually beneficial.

Buffalo and Oxpecker
Infinite supply of bugs and parasites.
Hisses when predators are near.

ScopeMaster

Early Estimates

Why do we estimate cost beforehand?

Improve certainty, reduce risk, improve planning

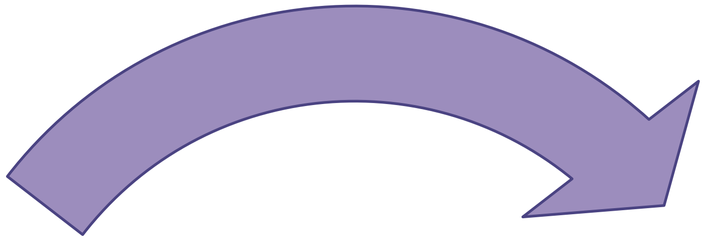
$1m

9 months

~~$10m~~

~~29 months~~

Mutual Benefit



Better requirements

Better Estimates

Activity of Functional Sizing

Embrace both activities

Cost Estimator

I started by trying to automate Functional Sizing, then pivoted

ScopeMaster

Why Functional Sizing

Functional sizing is all about:
data movements, data storage and retrieval

COSMIC

CFP

IFPUG
International Function Point
USERS GROUP

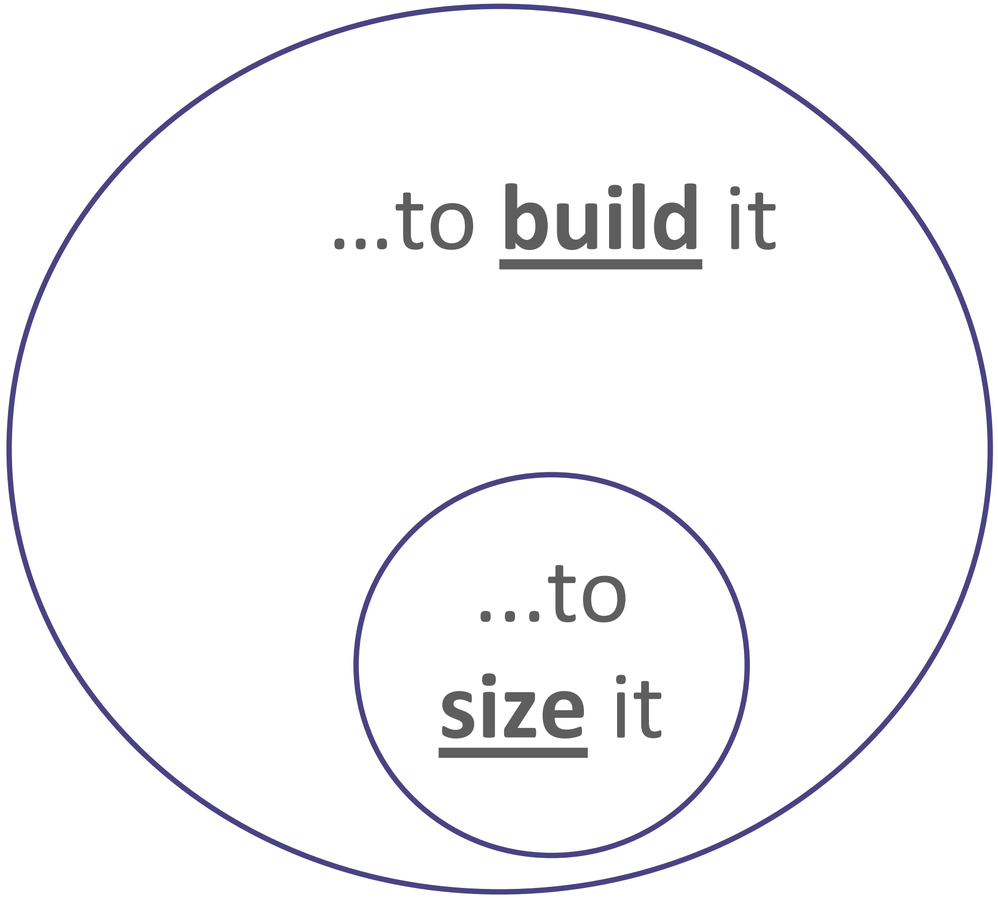
FP

If you know the Functional Size you can answer the questions of:
Resources, Schedule, Effort, Quality and more.

https://ronjeffries.com/articles/-z022/0222ff/est-cosmic=other/

ScopeMaster

What you need to know…



COSMIC

https://cosmic-sizing.org

IFPUG International Function Point USERS GROUP

https://ifpug.org

ScopeMaster

We want the best estimates we can achieve

If I wanted <u>bad</u> estimates what would I do?

I'd use poor requirements as a basis for my estimate

If I wanted to create <u>poor requirements</u> what would I do?

I'd skip functional sizing

ScopeMaster

Poor requirements are a form of technical debt.

Poor requirements will lead to a higher cost per function point.

1 FP delivered wrongly is likely to be 2.5x more costly than doing it right first time.

ScopeMaster

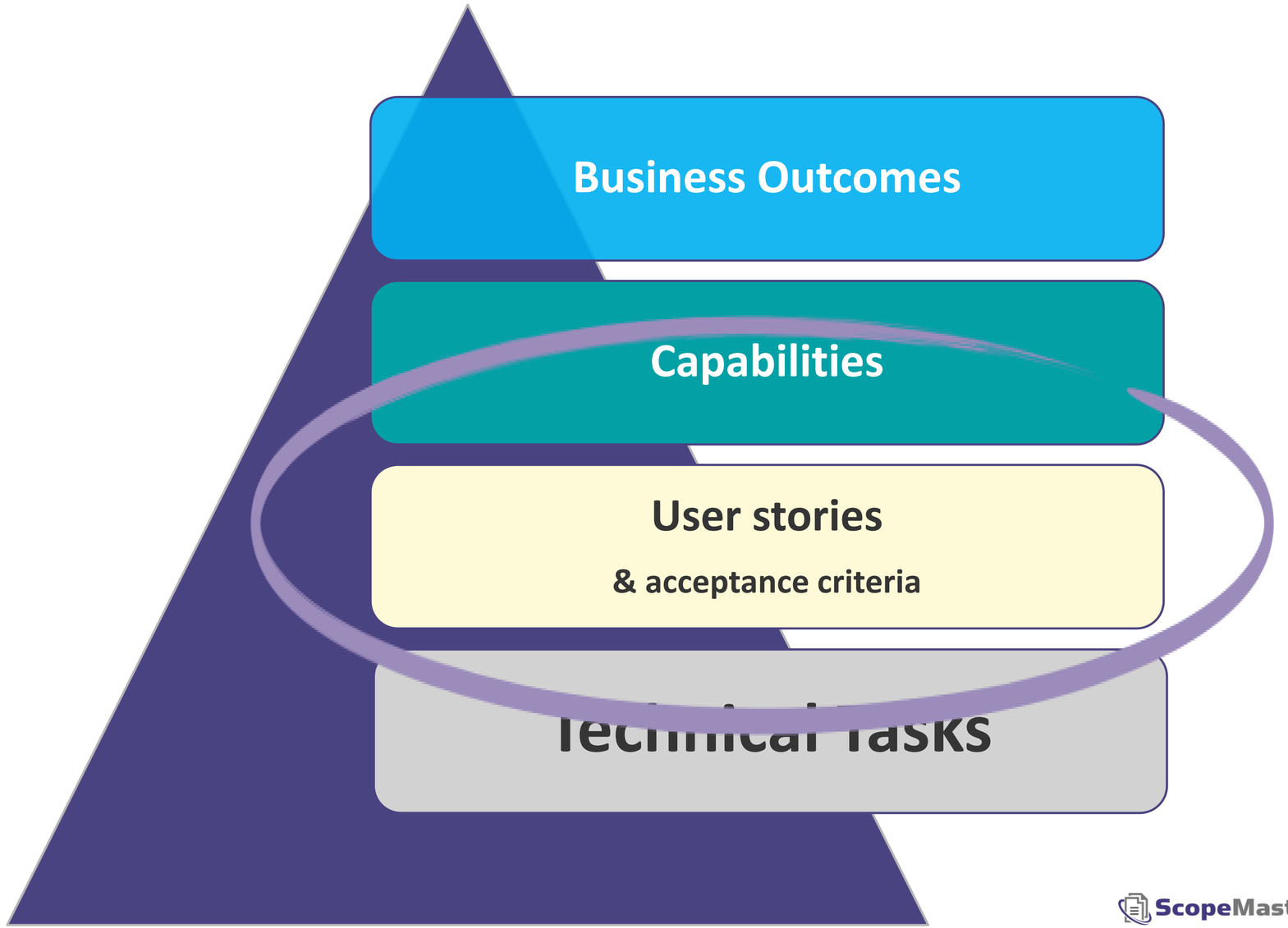
Challenges with requirements quality

- Lack of formal training
- Confusion about what good looks like.
- Excessive detail and miss the big picture.
- Unknowns, biases and change

ScopeMaster

Granularity – for functional sizing

Software Requirements Hierarchy:



ScopeMaster

Challenges to early estimation

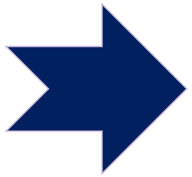
- We don’t know all the requirements
- Inappropriate granularity (too little, too much)
- We don’t want to do all the requirements work
- Unknowns (some will be knowable and some not)
- Changing requirements
- Technical Unknowns
- We tend to ask the technical folks not the expert BAs.
- Gameable effort estimates (Tshirt, story pts, story #)
- Politics and inexperienced leadership
- Other human & commercial factors

Unknowns – use good analysis techniques

Biases – use functional sizing

ScopeMaster

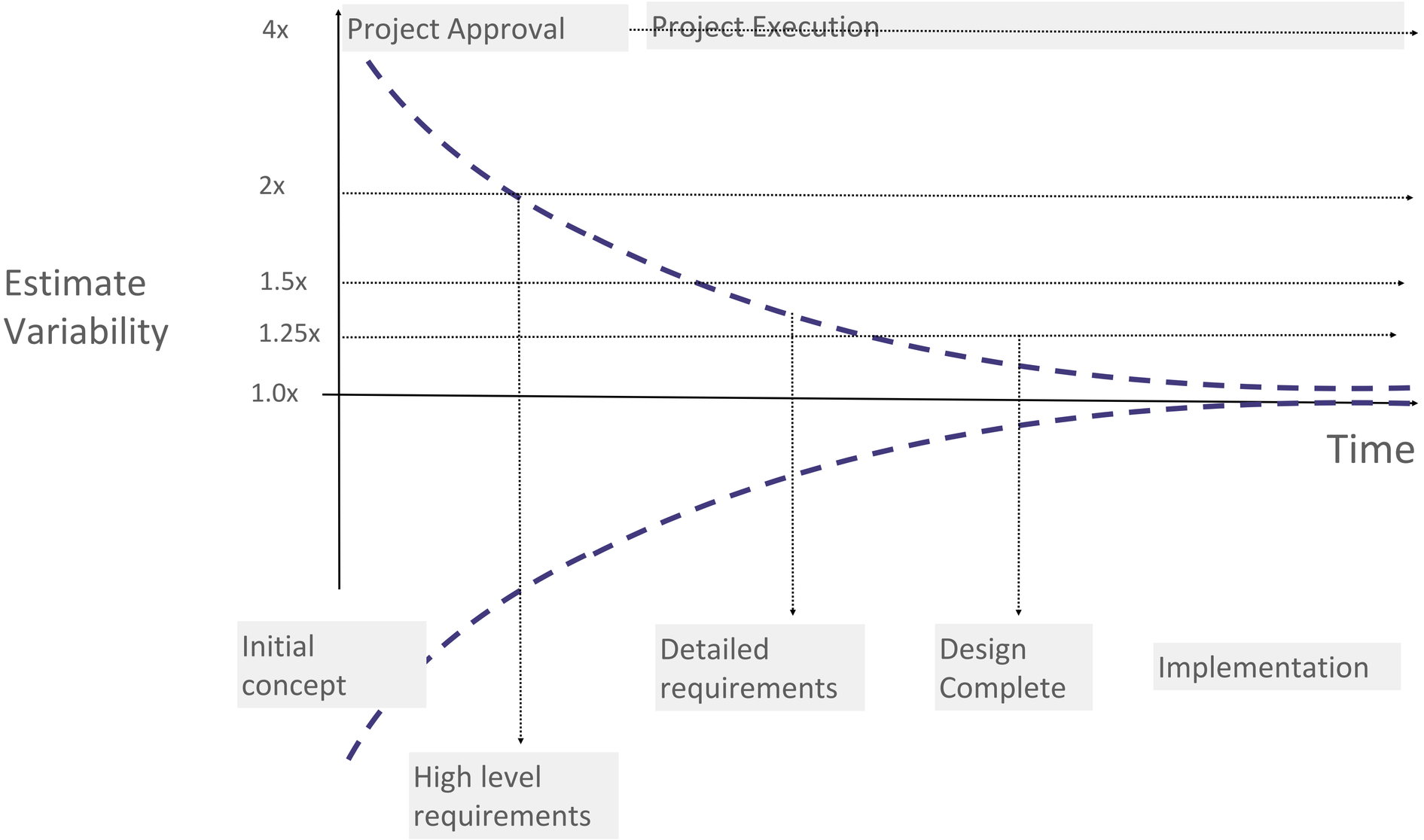
Early Estimation – Scope Sizing



Project	Typical Perception	Actual
Knowns	**Knowns**	**Knowns**
Unknowns	**Knowable Unknowns**	**Knowable Unknowns**
	Unknowable Unknowns	**Unknowable Unknowns**

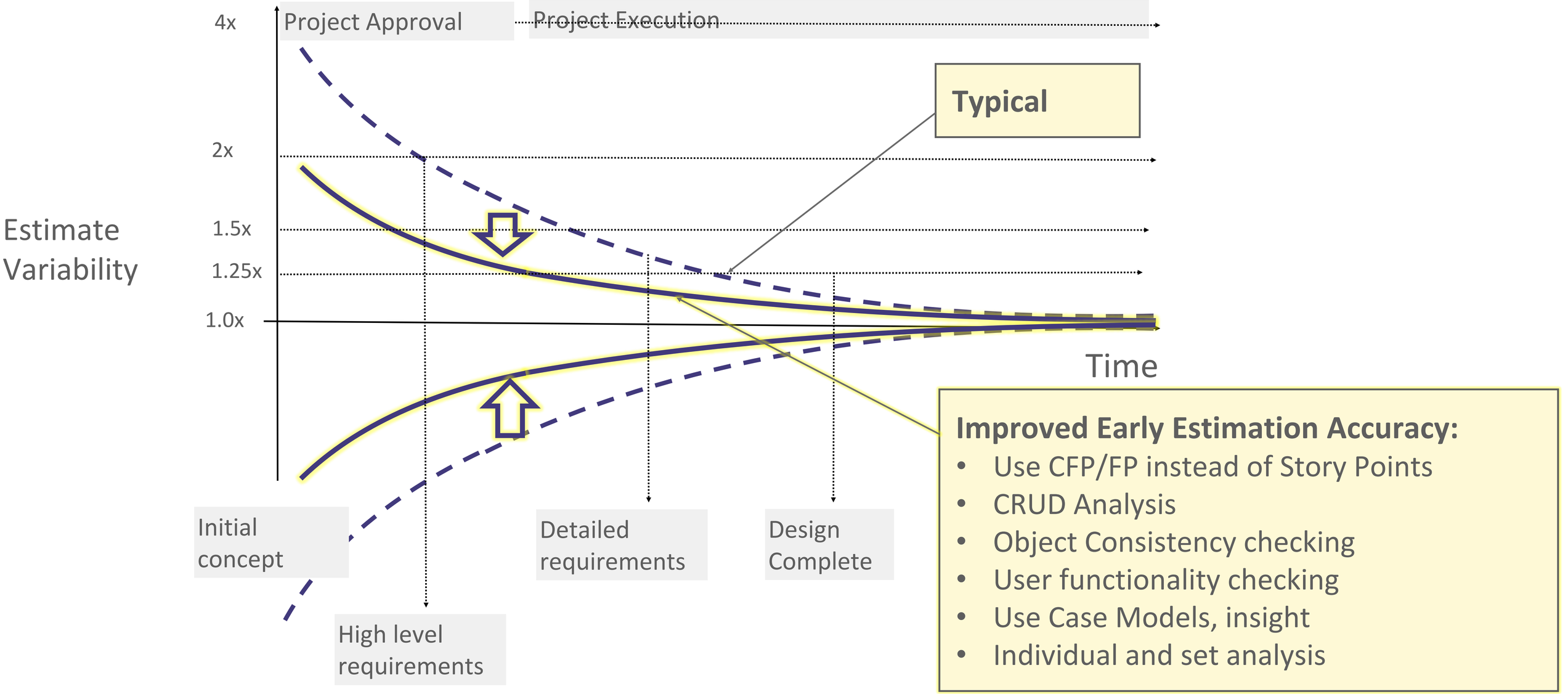
ScopeMaster

Typical Estimation Accuracy



ScopeMaster

Improved Early Estimation Accuracy



Improved Early Estimation Accuracy:

- Use CFP/FP instead of Story Points
- CRUD Analysis
- Object Consistency checking
- User functionality checking
- Use Case Models, insight
- Individual and set analysis

ScopeMaster

What makes for good requirements

Clear = Functional Intent
Concise = Objects / ILFs
User-oriented = User focused
Testable = ie. clear functionality
Measurable = ie. clear functionality
Consistent = Object naming
Complete = within and across stories
Unique = not duplicated
Design-free = no technical/implementation *
Valuable = needed for business value *

Functional sizing helps to expose

Heavy-lifting of this analysis can be automated

Knowable Unknowns

Low impact on sizing *

ScopeMaster

Completeness - of an individual user story

Before

As a sales agent I want to be able to edit a contact's profile

3 CFP

After:

As a sales agent, first verify that I have permissions to the profile, then verify that the contact profile is not locked, then I can edit the contact's profile,

8 CFP

There seem to be 3 objects and 3 actions, but it is still clear and sizeable.

Refined requirements tend to be bigger than unrefined

Knowable Unknowns

ScopeMaster

Needs splitting

As a sales agent I want the contact profiles synchronised between the CRM, mailing system and customer app.

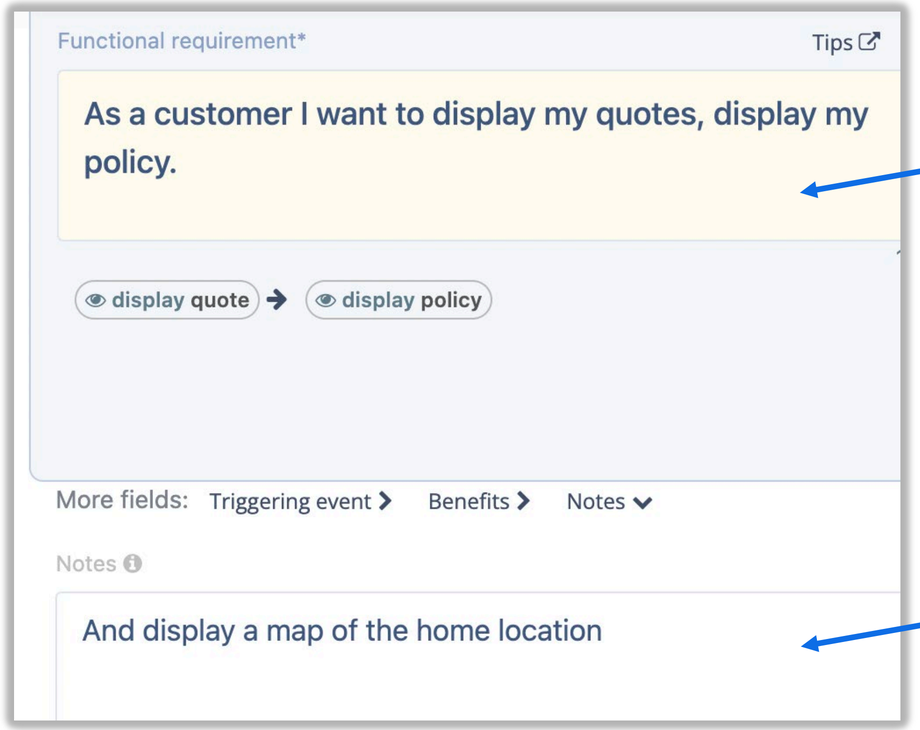
40+ CFP

These examples highlight why the practice of counting user stories is an unsafe basis for cost estimation

Knowable Unknowns

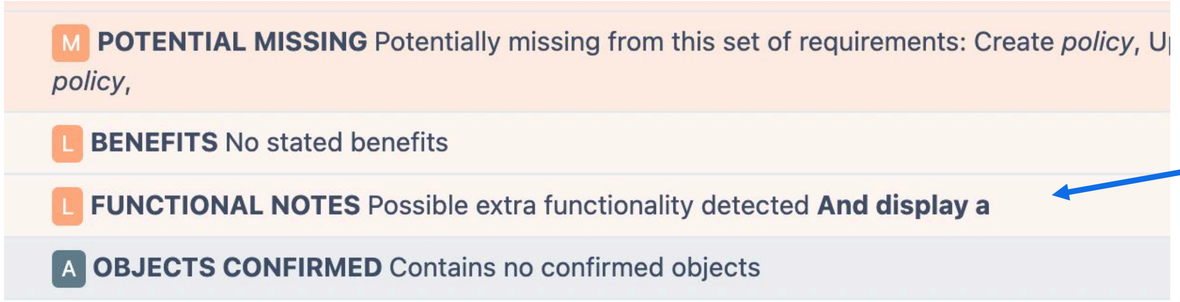
ScopeMaster

Completeness – Buried functionality



User story

Functionality buried in the Acceptance criteria



detected

Functionality gets buried in the acceptance criteria

With NLP this can be detected instantly

Knowable Unknowns

ScopeMaster

Completeness of a set – Automated CRUD analysis

757
Requirements

Exposes:
1) **Naming** inconsistencies
2) **Missing** functionality
3) **Duplicated** functionality

CRUD and consistency analysis

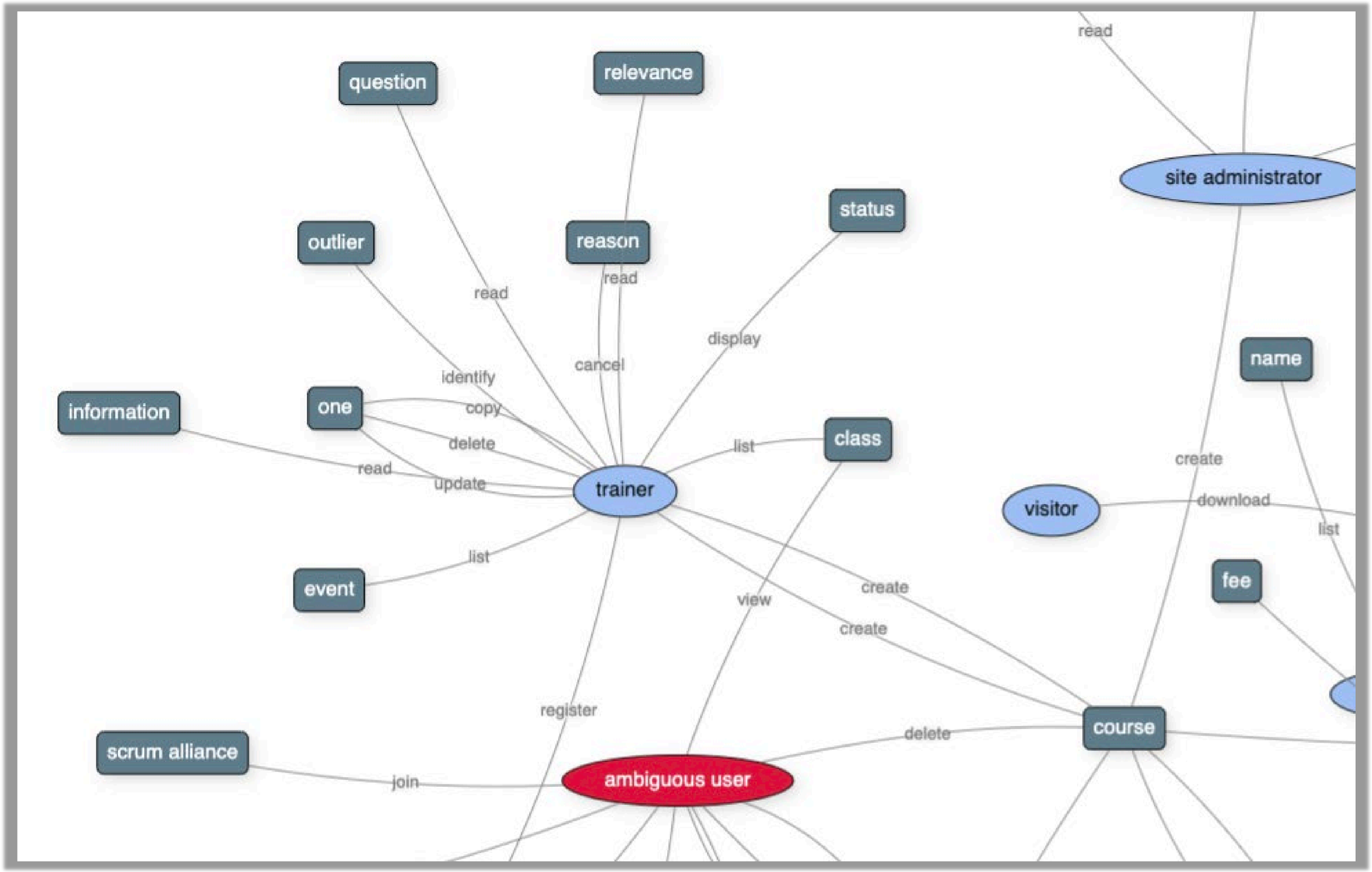
Find and fix potential inconsistencies, missing and duplicate Stories.

Objects (228)	Create 132 (96)	Read 66 (162)	Update 75 (153)	Delete 16 (212)
' button	1411359 JRACLOUD-73056	Missing +	Missing +	Missing +
ability	Duplicate 1112938 JRACLOUD-71524 1112940 JRACLOUD-71526 1121434 JRACLOUD-71588 1504899 JRACLOUD-74076	Missing +	Missing +	Missing +
access	Missing +	1168295 JRACLOUD-72050	Missing +	

Knowable Unknowns

ScopeMaster

Completeness of a set – Use Case Modelling



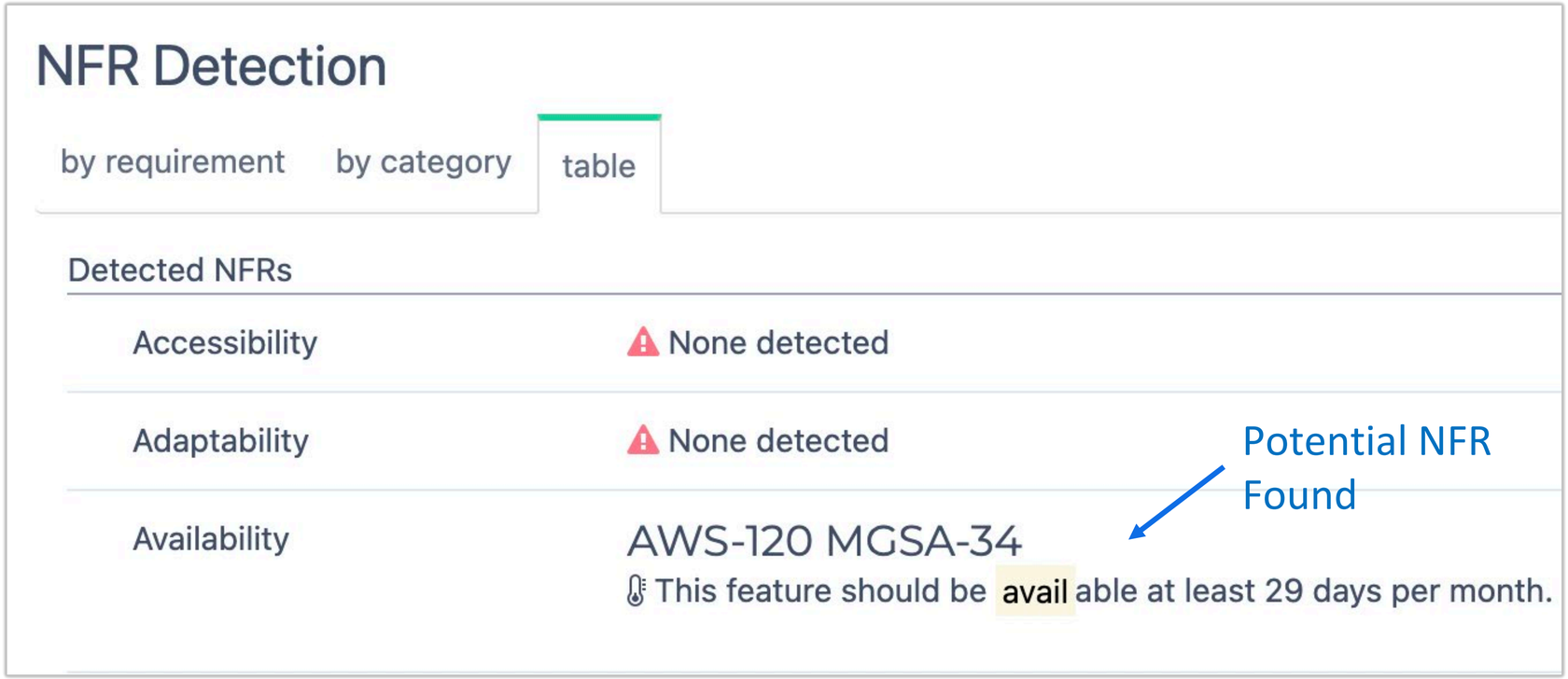
Easily Spot

Ambiguous users
Inconsistencies in personas
Inconsistencies in object names
Complexities of requirements

Knowable Unknowns

ScopeMaster

NLP can detect potential NFRs



NFRs Affect Cost

Spotting them early can minimise the cost impact on a project.

Knowable Unknowns

ScopeMaster

Summary of Requirements Quality impact on Estimates

Requirements Quality Attribute	Context	Indicative Range: actual vs initial estimate
Functional completeness of a requirement	Requirement	**0 to +400%**
CRUD Completeness	Set	**-20% to +400%**
Missing requirements, revealed through modelling. (inc user oriented).	Set	**0 to 70%**
Ambiguous functional requirements	Requirement	**0 to +300%**
Object Naming Inconsistency	Set	**-150% to +20%**
Methodology Choice: IFPUG vs COSMIC vs SFP	Set	**-30% to +30%**
Sizing Precision (automated vs manual)	Set	**-15% to +15%**
NFRs that are actually functional	Set	**0 to 30%**

ScopeMaster

Summary of Quality impact on Estimates

Requirements Quality Attribute	Context	Indicative Range: actual vs initial estimate	Explanation	Typical observation
Functional completeness of a requirement	Requirement	**0 to +400%**	Functionality is often omitted or buried in acceptance criteria. Actual can be 4x larger than initial size estimate.	Most requirements understate functionality and end up 50% - 100% bigger than initially stated/estimated.
CRUD Completeness	Set	**-20% to +400%**	We sometimes see only one function mentioned when a full set of CRUD is required. Duplicates are far less comon.	Most sets only include only 50% of required CRUDs.
Missing requirements, revealed through modelling. (inc user oriented).	Set	**0 to 50%**	Manual or automated modelling can expose “hard-to-reach knowable uknowns”.	Typically, 10- 30% of missing functionality can be detected this way.
Ambiguous functional requirements	Requirement	**0 to +300%**	Functional ambiguities due to poor language use, often mask understatements of scope.	About 40% of all requirements are initially unsizable. Using a tool like ScopeMaster from the outset eliminates this problem very quickly.
Object Naming Inconsistency	Set	**-150% to +20%**	Inconsistent object names can lead to overestimate of size. (The only item in the table that leads to early overestimation.)	This is common and tends to overstate initial automated size detection.
Methodology Choice: IFPUG vs COSMIC vs SFP	Set	**-30% to +30%**	The gross FP count discrepancy between these methodologies is less significant than other factors.	Automated IFPUG estimates are governed by the ILF complexity assessment which is very hard with NLP. COSMIC does not suffer this variability.
Sizing Precision (automated vs manual)	Set	**-15% to +15%**	Whether using automation or sizing manually, rarely more than 15% variance for CFP.	A formal test has shown automation in CFP to be within 15% of a manual count.
NFRs that are actually functional	Set	**0 to 30%**	When an NFR is assumed to be not functional but actually is.	functional security requirements are often overlooked.

ScopeMaster

Results of using NLP across a set of requirements

Functionals Detected

Functional Steps	66 found in 58 requirements
Functional Objects	49 found in 58 requirements

Total Functional Size Estimate

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

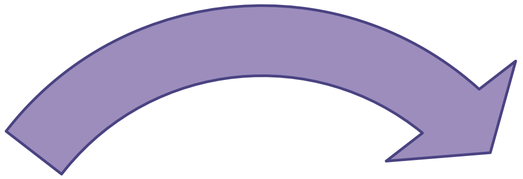
ℹ CFP = COSMIC Function Points

Knowns

Knowable Unknowns

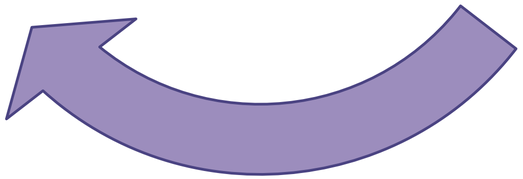
ScopeMaster

Mutual Benefit



Requirements Quality

EstimationQuality



Functional sizing and automated analysis expose knowable unknowns