Adapting Software Estimation with the Trend from Project to Product

Why are we seeing a Trend?

1. “Project Failures”
2. Success stories (esp Saas startups)
3. Agile typically has a “product owner” role.

Approaches to software change delivery

Project
Once and done.

Flexible by scope. Can be a wrapper for product approach.

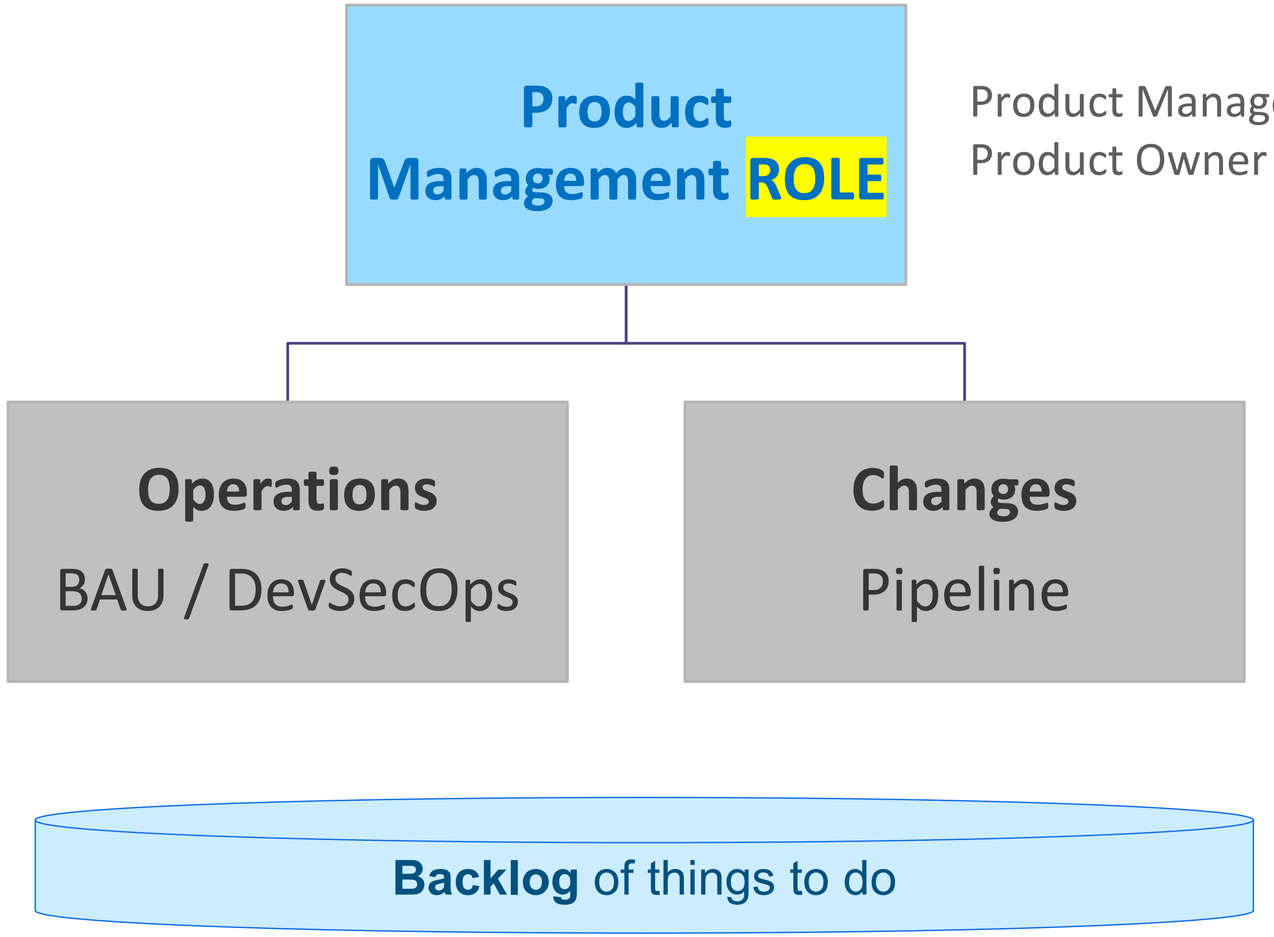
Product
Single solution/code-base, single backlog.

Ideal for new, hardware + software developments..

DevSecOps
Continuous maintenance + improvement

Finite change capacity. Backlog driven, rapid re-prioritisation.

What is Product Management?



Product Manager similar to Product Owner

1. Product manager has high domain knowledge
2. Clear ownership of the product success.

A new product launch is always a project

PROJECT

Minimum Viable Product

Must manage:

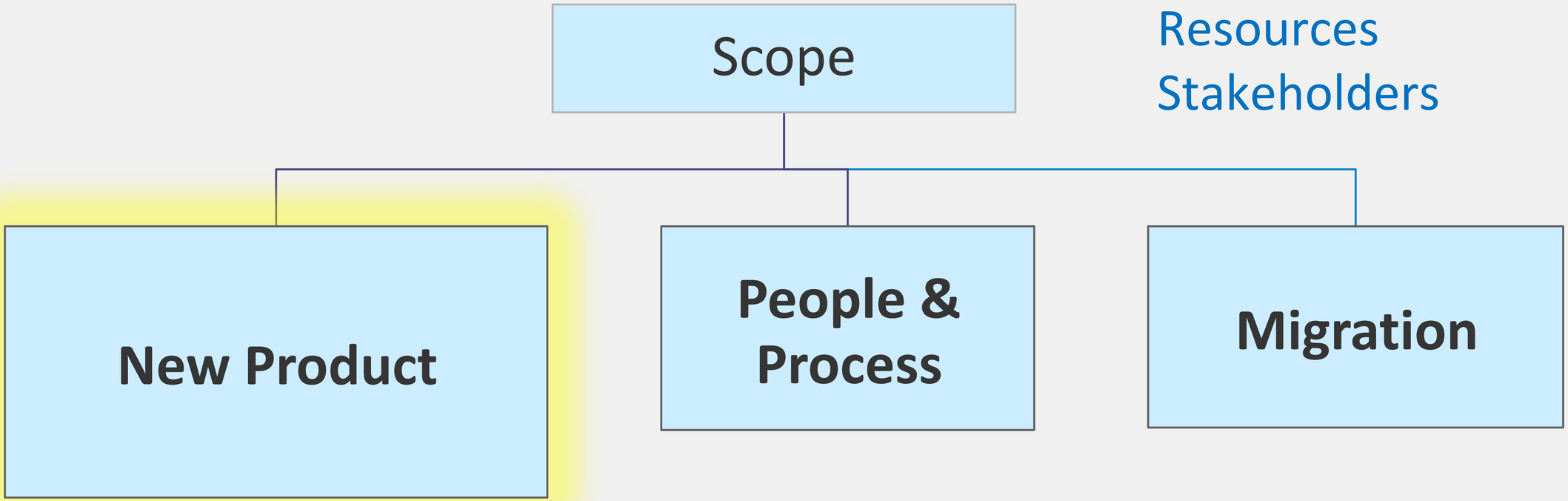
Risk

Scope

Schedule

Resources

Stakeholders



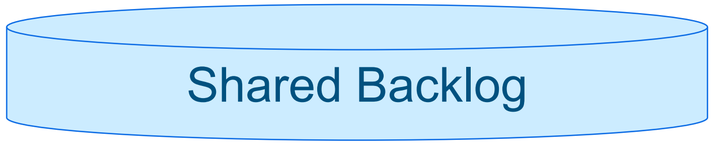
MVP = Minimum viable product

Product Approach – new product introduction

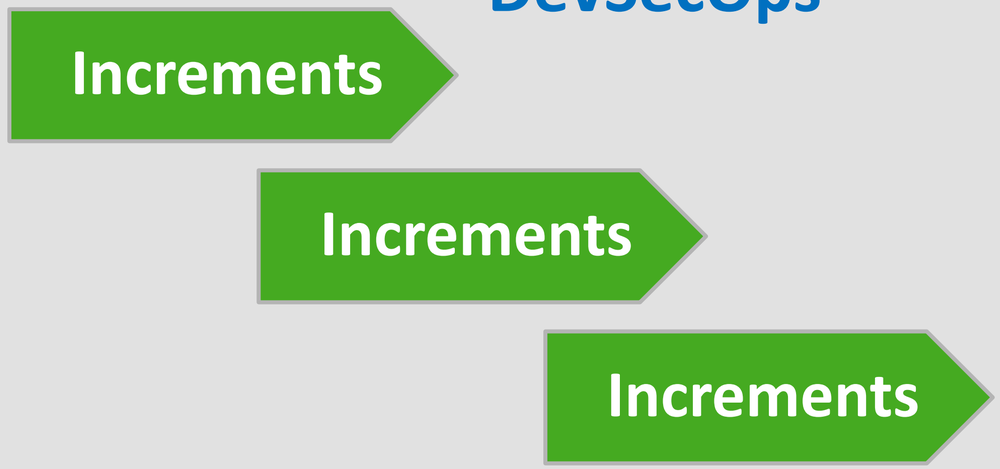
Product Management is a ROLE not a delivery methodology

PROJECT

Minimum Viable Product



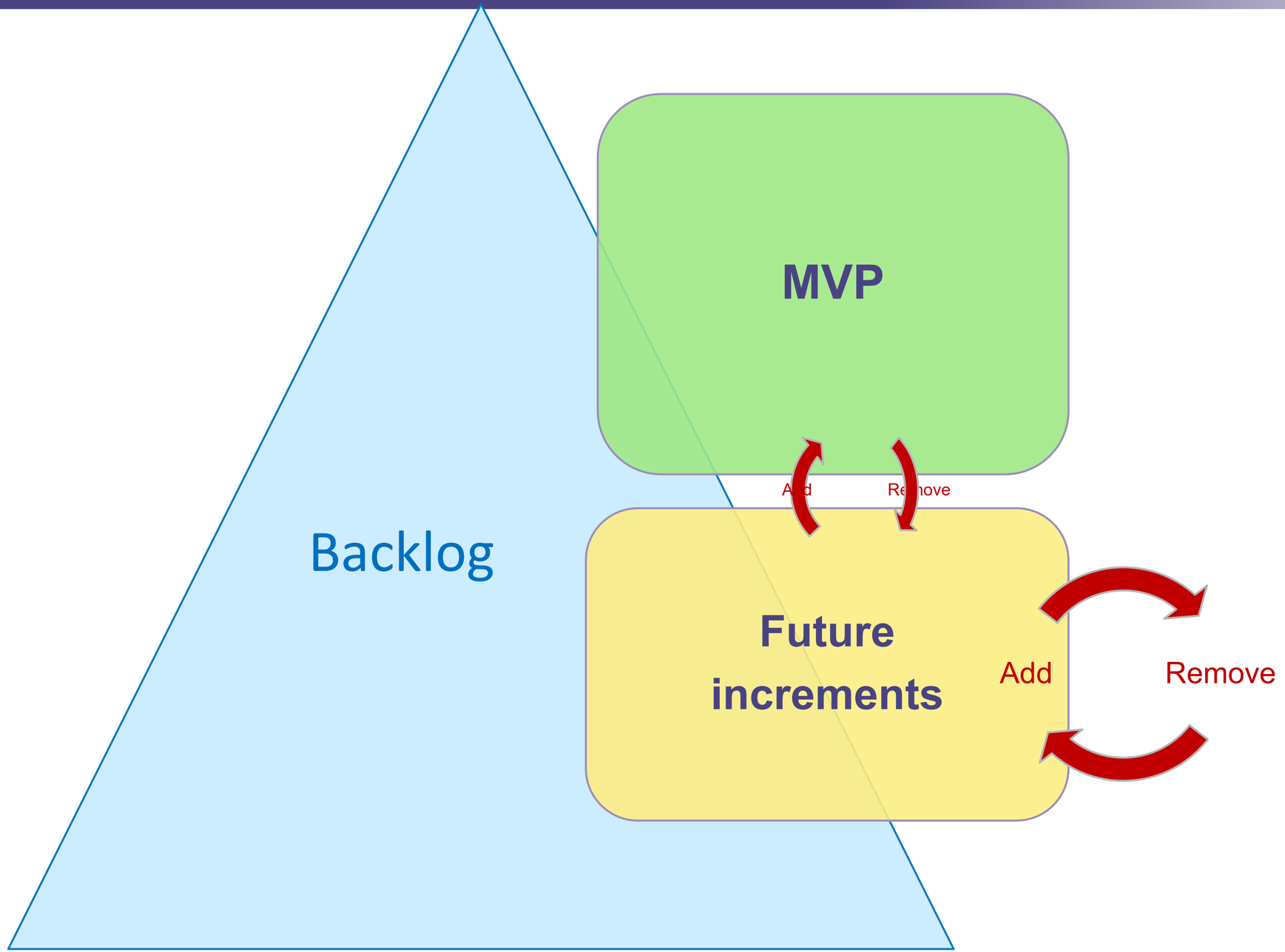
DevSecOps



Thereafter, **incremental** improvements

Suitable if the majority work is delivering a new product

Scope Changing

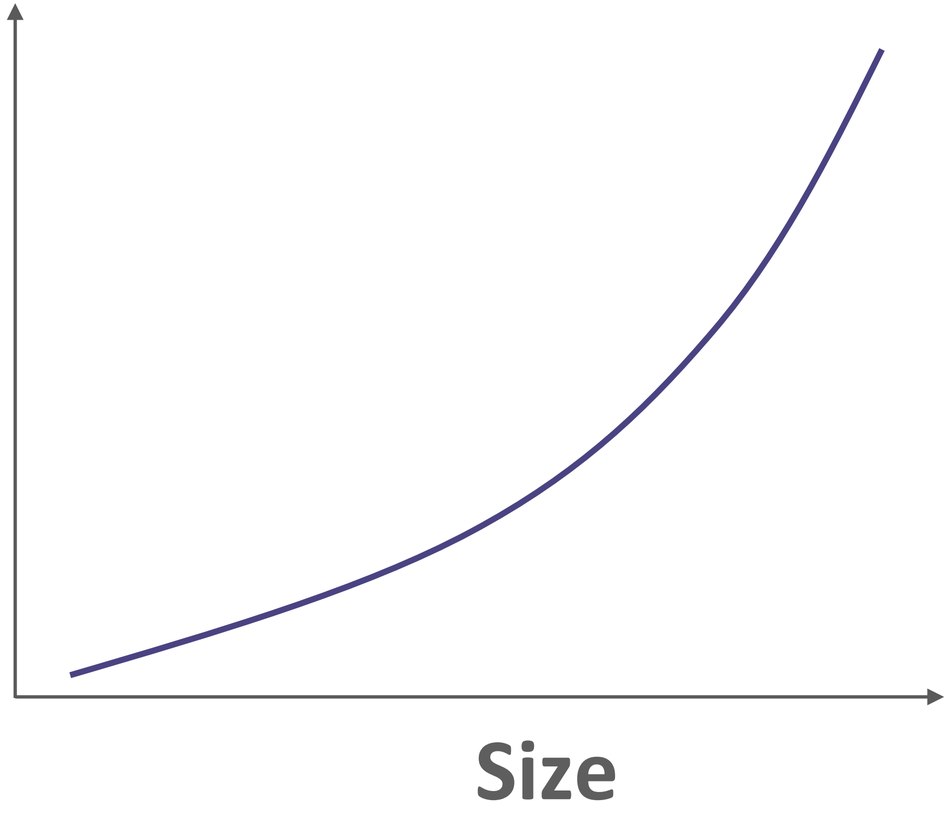


Warning!

Excessive volatility can distract, cause rework and slow progress.

Software Work does not Scale – opposite!

Cost
& Risk of failure



small is beautiful

What is small..?

What is the optimum **size**?

PROJECT

MVP

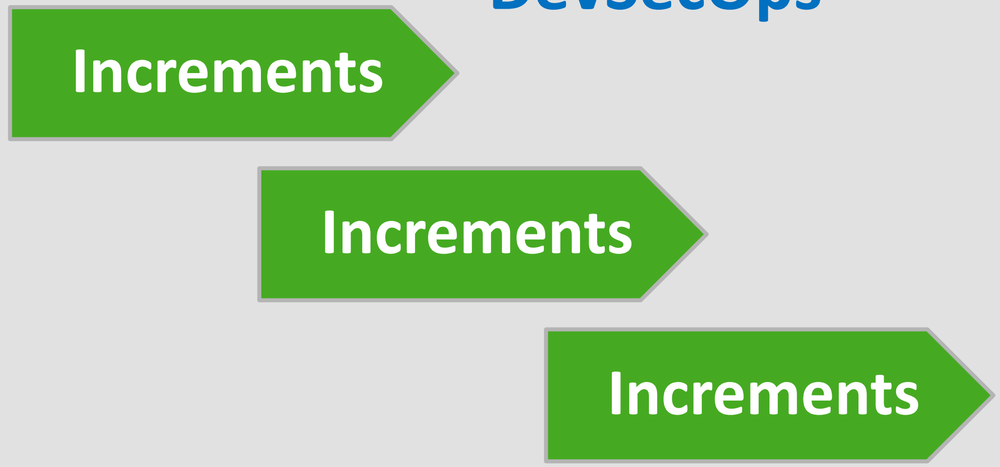
Ideally < 500 CFP because deliver in a few months max to get **fast feedback**

50% for reactive work.
50% change agenda

Limited **UNLESS** you can safely organise into multiple teams.

Know team productivity CFP/sprint

DevSecOps



And why do we Estimate?

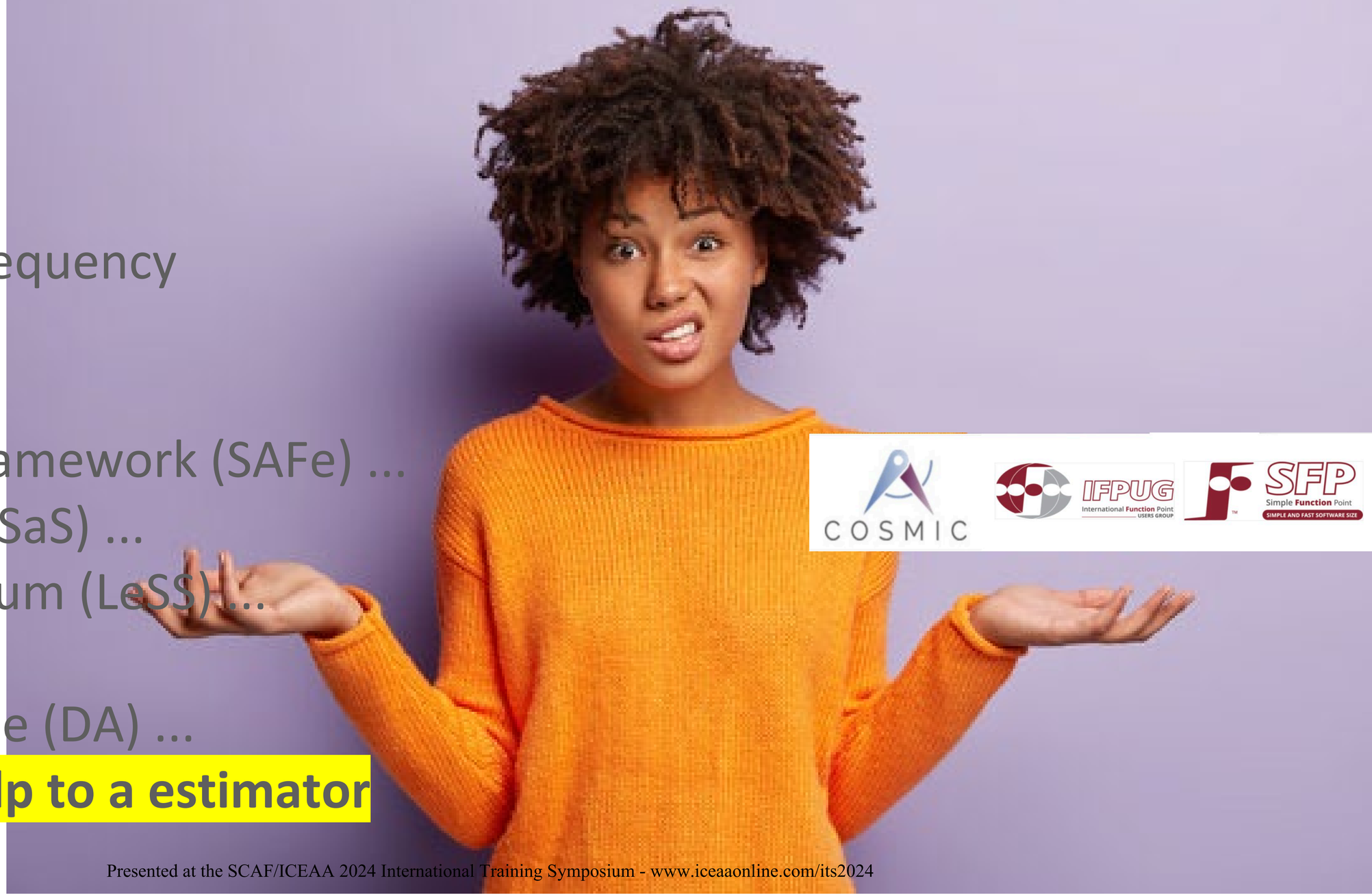
To make sound decisions

About what to do, and what not do with scarce resources.

Popular Metrics related Jargon

ScopeMaster

PBIs
Story Points
T-Shirt sizing
Deployment frequency
Flow metrics
Planning poker
Scaled Agile Framework (SAFe) ...
Scrum@Scale (SaS) ...
Large Scale Scrum (LeSS) ...
Nexus. ...
Disciplined Agile (DA) ...
= **very little help to a estimator**



COSMIC
IFPUG International Function Point USERS GROUP
SFP Simple Function Point SIMPLE AND FAST SOFTWARE SIZE

Does Estimation Differ Product vs Project? NO

ScopeMaster

Developers and Contractors:

"Functional sizing is wasteful, outdated and not agile."

We say:

"Knowing the functiona size is essential for sou decisions"


Be rational, equip yourself as best you can...

How is estimation done vs how it should be done

ScopeMaster

“No estimates”

Story Points or Story Counts

Functional Sizing (manual)

Automated Functional analysis and sizing

Think about why…?

They Say	We say
We learn as we go	Most unknowns are knowable* The rest is risk management.
Estimation is BRUF	Functional Sizing can be done at high granularity
Estimates are abused by managers	Story points & counts, yes, but not CFP
Estimation is a distraction	Functional Estimation is automated
It's wasteful to estimate prematurely	Almost effortless*
Focus on flow and flowmetrics	Does not facilitate sound decisions
Just build and focus on value	But, it might not be worth starting!
Don't know when it will be ready	We do.
We can't tell you the cost	We can.

*If using automated sizing & analysis

Automated analysis - Nature, speed and precision

ScopeMaster

Each Story

Clear Functional Intent: Reduces misunderstanding, avoidable discussions and rework.

Quality:
Ambiguity
User-oriented
Completeness
Sizeability
Testability

Sizing:
CFP, IFPUG or SFP
Consistent
< 15% of a manual sizing – (verified)

Speed:
Consistent
< 15% of a manual sizing

Insight: Instant feedback = Agile

Set of Stories

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

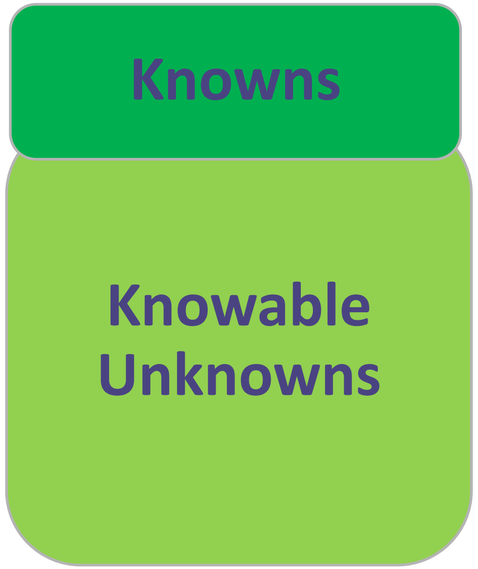
Exposing the knowable unknowns

ScopeMaster

Example:

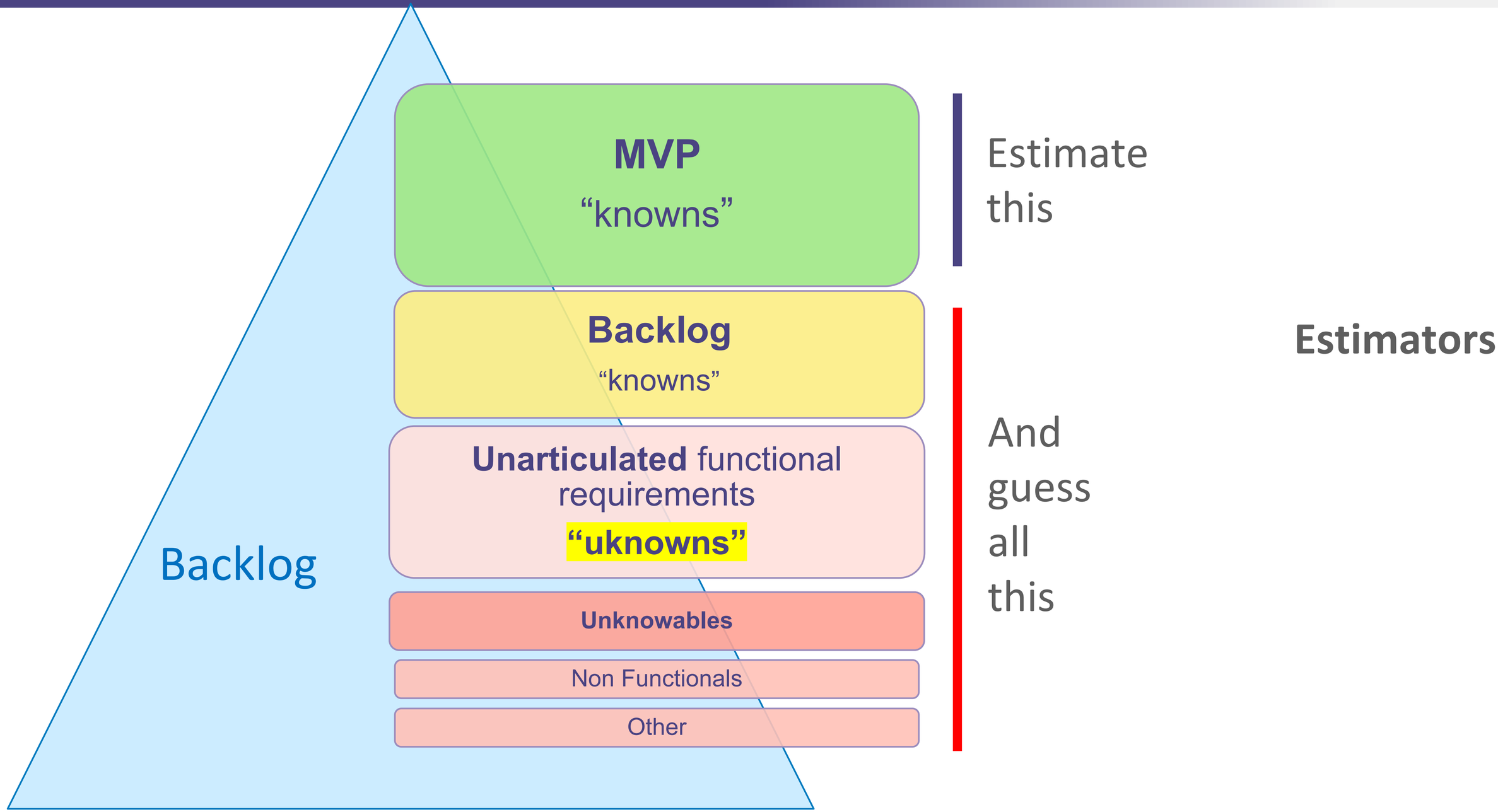
Total Functional Size Estimate			
Sized requirements	58	259 CFP	← Functionality Found
Ambiguous requirements (ie. no functionality detected)	59	263 CFP *Estimated*	⇠ *Inferred*
All functional requirements (sized + ambiguous)	117	**522** CFP *Estimated*	
Potential missing requirements (from CRUD analysis)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

CFP = COSMIC Function Points



Traceable data-driven evidence for cost estimators and project managers of the likely final cost – BEFORE investment commitment.

Scope:


Backlog
MVP
"knowns"
Backlog
"knowns"
Unarticulated functional requirements
"uknowns"
Unknowables
Non Functionals
Other
Estimate this
And guess all this
Estimators

Scope:

ScopeMaster

Backlog

MVP

"knowns"

NON- MVP

"knowns"

Unarticulated functional requirements

"Knowable uknowns"

Unknowables

Non Functionals

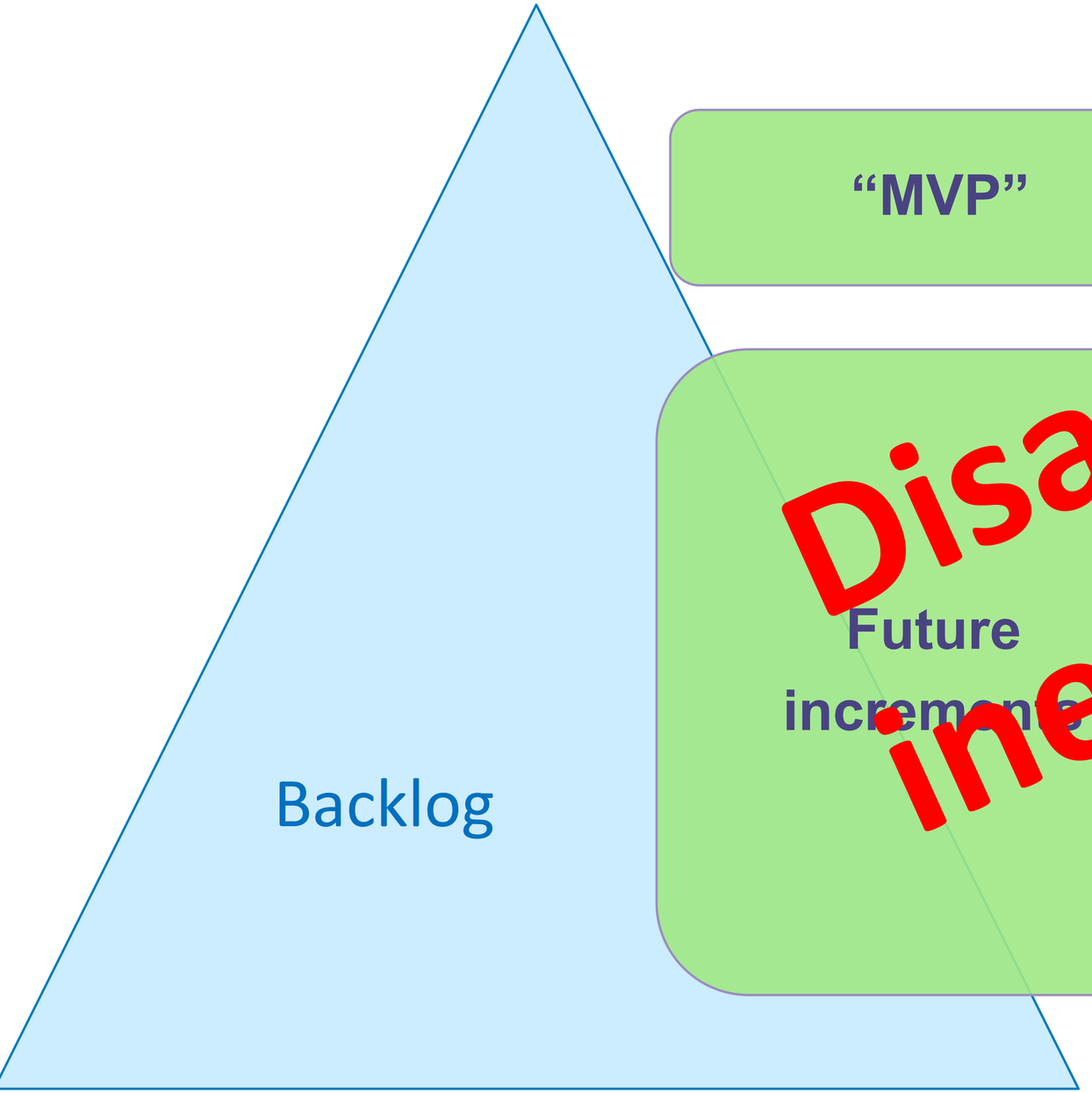
Other

Estimators:

Estimate this instantly

And guess this

Project OR Product, irrelevant Size and Quality Matter



1,800 CFP (taking 2 years)

Planning

27,000 CFP

(in the next 2 years)

15x productivity increase!

Disaster is inevitable.

Incentive biases are dominating bad behaviours.

Project OR Product Size and Quality Matter

ScopeMaster

Sample MVP Product or Project: 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

1year delay cost

4x

the initial estimate!

Why failure / under estimate?

Why do software initiatives go over budget and schedule?

1. Poor executive engagement
2. Unrealistic expectations of cost and time
3. Too ambitious (size)
4. Poor requirements
5. Poor project mgt (scope & risk)
6. Poor attention to quality (esp early)

For cost estimators

ScopeMaster

Automated Backlog Analysis

Functional
Sizing

Requirements
Insight

COSMIC · IFPUG International Function Point Users Group · SFP Simple Function Point — Simple and Fast Software Size

Instantly expose issues that affect cost and schedule

Objective, defendable, estimates,
requiring negligible tech input

Data driven decisions

Cost estimators need auditable, traceable, data-driven objective basis for cost and benefit delivery estimation, whether using a Project or Product approach

Estimating in Product World - conclusion

Objective estimation is just as necessary in a product world

Objective estimation is **NOT about story points**, story counts or T-shirts, but ISO standards functional size – CFP or FP

Product approach tends to have a **backlog of options** for consideration, some of which will never be done, but they need some level of estimation regardless.

Automated functional analysis and estimation, allows **effortless sizing of all the backlog** AND avoids distracting developers.