

# Busting Myths of Software Estimation for Better Decisions



1

## What are Myths?



**Myths are commonly held opinions about something, which is typically wrong**

2



1. ***"Estimation is wasteful."***
2. ***"Only the team can decide how to estimate."***
3. ***"We don't have enough detail to estimate."***
4. ***"Fixed price agile is not possible."***

3

**Estimate**

4

# Estimate (verb) An Estimate (noun)

## Effort? Duration? Cost? Size?

5

**"Estimate"** many meanings - causing confusion:

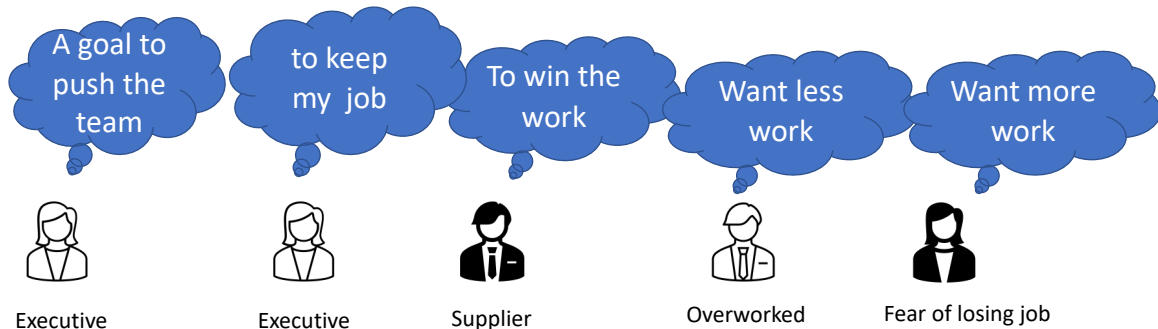
1. Project level budget with cost & schedule expectations.
2. Effort guess by the delivery team
3. Obfuscated guess by the delivery team
4. A Gating Step
  - A box ticking exercise to get more money for the project
5. Goals
  - To challenge the delivery teams
6. Commitments
  - Promises

6

## What do we mean by Estimate?



- Depends who's asking:



7

## How do we estimate?



- Whatever the executives request
- Team discussions to derive story point estimates
  - Time-consuming
  - Obfuscated and gamed
  - Non-standard
  - Not useful at a project level
- Expert guess of staff months at an epic/capability level
- High level Reference Class Estimation
- Approximate manual functional point estimate
- Automated functional point estimate
- Automated functional point estimate with insights (omissions, ambiguities)
- Automated functional point estimate with insights post refinement

8

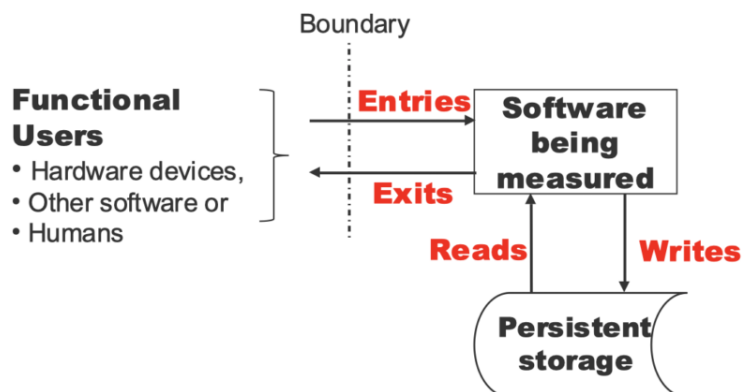
## Why use Functional Sizing



- Objective
- Transparent
- Traceable
- Defendable
- Consistent
- Benchmarkable
- Automated - Can be done with minimal distraction
- Excellent proxy for EVM
- Multiple uses (scope, productivity, resourcing and quality)
- High correlation to effort
- Applicable to all types of software
- Use COSMIC or IFPUG or Simple FP
- You need to know it anyway to build, ie not wasted effort.

9


## COSMIC Function points




**The 'Data Movement' is the unit of measure: 1 CFP**

Copyright © COSMIC 2017

10


Estimation is Wasteful




## Agile Teams

Discuss each user story, what it means how it will be delivered and tested, complexities, then guess how much effort it will take and then explain, then obfuscate the result into story points.

11

Auto-interpretation


MGSA-74 AWS-101 Sprint 6 Java Outsourced Status In Progress Estimated CFP: 8

As a registrant who successfully completed a Course (becoming a Certified ScrumMaster or Certified Scrum Product Owner), I am emailed a link to a survey about the course and instructor, so I can submit feedback on the course.


Edit
Analysis 2
Quality A
History 12
Tests 4
Comments
Related
More

Detected Functional Steps	Data Movements
<div style="display: flex; align-items: center; margin-bottom: 10px;"> <span style="font-size: small; margin-right: 5px;">registrant</span> <div style="border: 1px solid #ccc; border-radius: 15px; padding: 2px 5px; display: inline-block;">+ complete course</div> </div>	<p style="font-size: small; margin: 0;"><u>create</u></p> <ul style="list-style-type: none"> <li><span style="background-color: #333; color: white; padding: 0 2px;">E</span> input new <b>course</b> data</li> <li><span style="background-color: #333; color: white; padding: 0 2px;">R</span> check if <b>course</b> id exists</li> <li><span style="background-color: #333; color: white; padding: 0 2px;">W</span> write new <b>course</b></li> <li><span style="background-color: #333; color: white; padding: 0 2px;">X</span> return error or confirmation</li> </ul>
<div style="display: flex; align-items: center; margin-bottom: 10px;"> <span style="font-size: small; margin-right: 5px;">instructor</span> <div style="border: 1px solid #ccc; border-radius: 15px; padding: 2px 5px; display: inline-block;">+ submit feedback</div> </div>	<p style="font-size: small; margin: 0;"><u>create</u></p> <ul style="list-style-type: none"> <li><span style="background-color: #333; color: white; padding: 0 2px;">E</span> input new <b>feedback</b> data</li> <li><span style="background-color: #333; color: white; padding: 0 2px;">R</span> check if <b>feedback</b> id exists</li> <li><span style="background-color: #333; color: white; padding: 0 2px;">W</span> write new <b>feedback</b></li> <li><span style="background-color: #333; color: white; padding: 0 2px;">X</span> return error or confirmation</li> </ul>


Functional size : 8 CFP

Detects and estimates functional intent & size

12



### Cost to produce an estimate (300 stories):




**Story Points**

3-30 minutes per story.

300 stories x 0.25 hrs x 3 staff x \$100

= \$22,500k

28 staff days (> 9 days of duration)



**Automated CFP or FP**

0.1-3 minutes per story


300 x 0.016 hours x \$100  
= \$500 effort + \$3000

= \$3500 or \$5000 refined

2 staff days




**14x Faster, Better AND 5x cheaper**

13



### AI can size in ISO standards (COSMIC & IFPUG)

**Detected**



-  **60** ILFs + EIFs
-  **58** Els
-  **49** EOs + EQs

From 63 (57%) requirements

**Function Points**

	Low	Avg	High
Data	420	600	900
EPs	370	477	691
<b>Totals</b>	<b>790</b>	<b>1077</b>	<b>1591</b>

**Omissions**




-  **133** Els
-  **22** EOs + EQs

Missing CRUDs on 60 ILFs

**Function Points**

	Low	Avg	High
Data			
EPs	598	775	1063
<b>Total</b>	<b>598</b>	<b>775</b>	<b>1063</b>

**Ambiguities est.**

-  **45** ILFs + EIFs
-  **37** Els
-  **43** EOs + EQs

47 ambiguous requirements (43%)

**Function Points**

	Low	Avg	High
Data	313	448	671
EPs	276	356	516
<b>Totals</b>	<b>589</b>	<b>804</b>	<b>1187</b>

**AI can reveal most of the knowable unknowns**

14

AI can size in ISO standards (COSMIC & IFPUG)



**AI can reveal most of the knowable unknowns**

15


User Stories



**A misinterpreted user story = waste**

And the birth of a **bug**

16



### Best Project level Estimates use Functional sizing

#### Project Functional Sizing+

- The best sizing approach known for project estimation
- Standardised, transparent and auditable
- Valid, correlates well to effort
- Requires little or no effort by the technical team members
- Addresses new / changed / deleted and migration functionality
- With AI
  - Can expose knowable unknowns
  - Can expose requirements quality
- **Does not cover NFRs**

**COST: ~\$20 per user story**


**CHEAPER** 5x  
**FASTER** 2x-10x  
**BETTER** objective, standard, auditable

#### Team "Effort Estimation" in Story Points

- Tend to focus on effort rather than outcome
- Oriented around the opinion of the estimator's time it would take.
- Only deal with knowns.
- Subjective and inconsistent
- Usually in story points (that are meaningless at a project level)
- Time consuming to create
- Technicians tend to resent doing them
- Can reveal some activities that may not be noticed by looking just at functionalities
- Provide feedback on requirements quality
- Useful for non functionals and infrastructure work


**COST: ~\$100+ per user story**

17



### Estimation is Wasteful - or is it?

Estimated Cost: **£1m** - 1 year build  
 Estimated Benefits: **£5m** - 5 years @ £1m p.a.



**1 year delay & doubled cost.**  
 cost increased from £1m to £2m,  
 + one less year of benefits £1m  
 + subsequent benefit is 25% lower p.a. £1m  
 + opportunity cost £1m  
**Total consequential cost £4m higher**

5x

Real cost was  
£5m vs £1m

18

Common Myths about Early Software Estimation ScopeMaster

1. **Estimation is wasteful.** BUSTED! Premature task level estimation is probably wasteful but early project estimation is extremely valuable
2. **Only the team can decide how to estimate** BUSTED!
3. **We don't have enough detail to estimate.**
4. **Fixed price agile is not possible.**

**Start with why?**

19

We don't have enough detail to estimate ScopeMaster

How

What & Why

**Business Outcomes**

**Capabilities / Epics**

**User stories**

**Technical Tasks**

**User story**

- Functional and/or Non Functional
- Acceptance Criteria
- Benefit
- User(s)
- Triggering event
- Links or dependencies
- Notes
- Domain specific jargon
- Dual language

ScopeMaster

20

## Not enough Detail to Estimate?

### Start concise

**Epic**  
Maintain products

13

➔

### Elaborated / refined

**Story**  
As an administrator  
I can update products

3

**Story**  
As an administrator  
I can create a product

4

#### Suggests Elicitation Questions

Elicitation Questions to help you discover the real requirements

- What assumptions have been made with this requirement? Use the notes file
- Will all readers share a common understanding of the users and objects in the requirement?
- What problem does this requirement solve, there's no stated benefit?

#### Quality feedback

Quality B About the quality scores of

- B **BENEFITS** Contains declared benefit(s)
- C **CONCISENESS** Concise, ratio of words per CFP is 3 (less than 10 is good). Total words: 10
- C **FUNCTIONAL SIZE** Suitable size 6 CFP
- E **POTENTIAL MISSING** Potentially missing from this set of requirements: Create practice

21

## AI accelerates refinement

### Object Detection+

CRUD and consistency analysis Find and fix potential inconsistencies, missing and duplicates

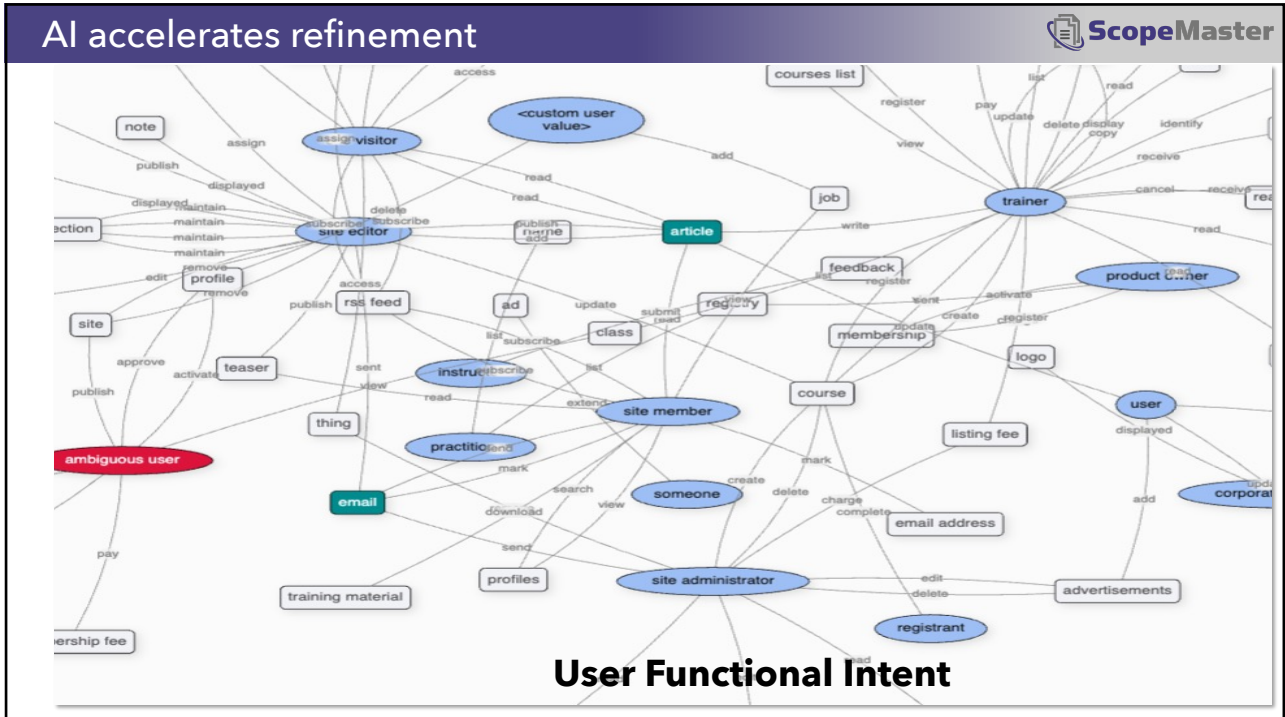
### CRUD Analysis

Category	Item	Status	ID
feedback	AWS-101 MGSA-74	Missing	MGSA-37
help	Missing	Missing	MGSA-97
information	Missing	Missing	MGSA-20
item	Missing	Missing	MGSA-20
job	AWS-107 add job	Missing	AWS-106

### Visualisation of User Interactions

## AI stimulates thinking that leads you to better and more complete requirements

22



23




Not enough Detail to Estimate? ScopeMaster

1. You can estimate better than you realise
2. Use tools and reference data
3. Use multiple techniques
4. Do the work

24

## Common Myths about Early Software Estimation



1. **Estimation is wasteful.** 
2. **Only the team can decide how to estimate.** 
3. **We don't have enough detail to estimate.** 
4. **Fixed price agile is not possible.**

Premature task level estimation is probably wasteful but early project estimation is extremely valuable

25

## Fixed price with Agile



### Why?

1. Cost (and schedule) predictability
2. Avoid project overruns/failure
3. Flexibility of Agile
4. Balance of cost & risk with supplier
5. Prevent commercial benefit of rework

26

Create a win-win arrangement ScopeMaster

**Need a commercial arrangement  
that drives positive behaviours**

27

How to achieve Fixed Price Agile ScopeMaster

```
graph BT; A[Standardised sizing (in CFP)] --> D((Fixed price Agile)); B[Fixed price per CFP contract structure] --> D; C[Automated Requirements Analysis] --> D; A --- A1[Objective, transparent]; B --- B1[Predictable]; C --- C1[Fast, fair, transparent];
```

CFP = ISO Standard COSMIC Function Points ScopeMaster

28

## Fixed price and Agile



### Contract structure

Aligns interests of both parties to delivery valuable, high quality software quickly and with minimal waste.

### Contract includes:

1. Firm fixed price per CFP
2. Ensures the customer provides quality requirements to an agreed leadtime, but not all up front.
3. Developers commit to a delivery rate and delivery quality.
4. Developers are rewarded for delivering early, penalized for late or poor quality.
5. Size and quality are assessed objectively.

**You can have both Agile and Fixed price.**

29

## Common Myths about Early Software Estimation



**1. Estimation is wasteful.**



Premature task level estimation is probably wasteful but early project estimation is extremely valuable

**2. Only the team can decide how to estimate**



**3. We don't have enough detail to estimate.**



**4. Fixed price agile is not possible.**



**Start with why?**

30