Overview

Effective Requirements

User Stories

Estimating Work

Planning Poker

Effective Requirements

What is a Requirement?

A feature, behavior, or constraint to be added to a system

A prelude to a conversation

A request for someone to do work

A request for software to change

A Requirement is NOT

A solution design

A decision about implementation

Typically illustrative of the final deliverable

The source of truth

Complete

Consistent

Correct

Modifiable

Ranked

Traceable

Unambiguous

Verifiable

NASA Requirements

But, can I explain it to my mom?

Interfaces

Functional Capabilities

Performance Levels

Data Structures/Elements

Safety

Reliability

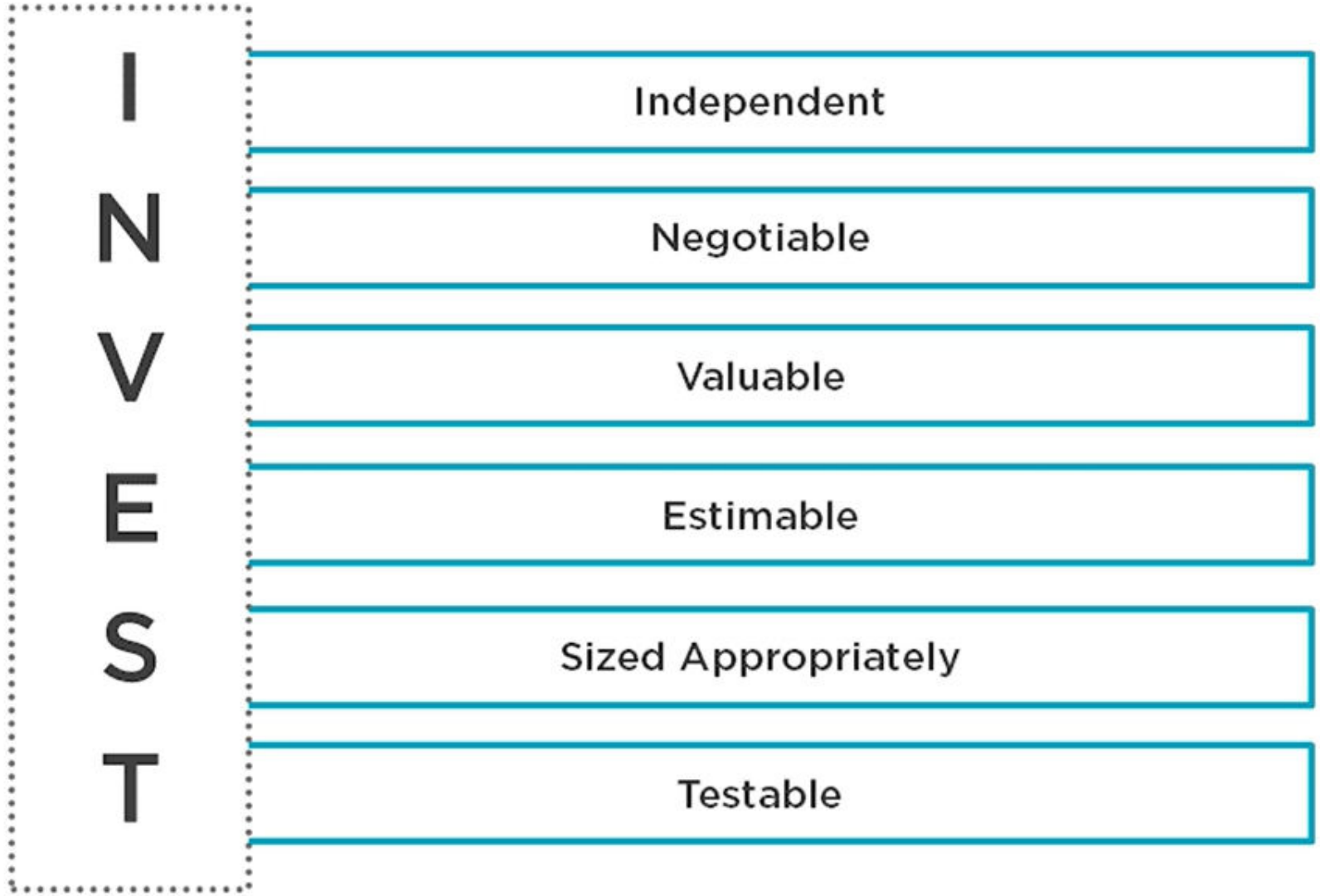
Security/Privacy

Quality

Constraints/Limitations

IEEE Requirements

> No way can I explain all this to mom


I
N
V
E
S
T
Independent
Negotiable
Valuable
Estimable
Sized Appropriately
Testable

User Stories

User Story Recipe:

As a <role> I want <feature>
so that <benefit>.

Some User Stories

As a traveller, I want to reserve a room

As a vacation planner, I want to see pictures of the hotels

A Note on Roles

Vacationer

Hotel Owner

Travel Agent

Trip Planner

Parent

Why User Stories Work Well

They are simple to write and understand

Software requirements is a communication problem

They elicit detail in conversation

Requirements analysis is effective when performed collaboratively

Full intent can rarely be modeled or represented 100%

The User Story Conversation

As a user with a reservation,
I want to cancel my reservation
So that I get a refund.

Does user get full or partial refund?

Refund to credit card or site credit?

How far ahead must reservation be cancelled?

Same for all hotels?

For all site visitors or can frequent travelers cancel later?

Confirmation provided to user? How?

Details as smaller sub-stories

AS A user with a reservation, I WANT to cancel my reservation SO THAT I get a refund

As a premium member, I want to cancel at the last minute with no penalty so that I get a full refund

As a non-premium member, I want to cancel up to 24 hours in advance so that I get a 50% refund

As a site member, I want an email confirmation of my cancelled reservation so that I can have a record of the transaction

Signs Stories are Working

- Focus shifts from writing to talking
- Stories are understood by customer and developer
- At estimation time, they are the right size
- Participative design is occurring
- Emphasis is on the user goals, not the system's attributes

Estimating Work

Estimates Are Necessary

To plan and proceed deliberately

To get a feel for costs

To calculate potential ROI

To understand the size of something

To know if work even can be done

To weigh options

Ways to Estimate Software

Darts

Give it to
a Manager

Ask the expert

Without
“bothering” the
developers

Deadly Estimation Warning Signs

Estimates are given without looking at historical performance.

Someone other than the team is doing the estimation.

Estimates are treated as promises.

Estimates are rejected because they don't fit an already existing plan.

"I just want to know when it will be done."

"That's bigger than it should be."

"That's smaller than it should be."

The Typical Estimation Process

PM: Hey Bill, how long to _____?

PM to self: They always say that. So, 2.5 weeks. I'll make it 3.

PM out loud: Thanks Bill. I'll go write the specs now.

Dev to Self: I'm busy, that'll take 2 days I can't afford to lose. What can I say to make him go away?

Dev out loud: About a week.

Dev to self: I can stall those out for weeks.

How do we measure software work?

Lines of Code

Buckets per day

Cycles Per Month

Mega Jewels per nanosecond

Coffees per day

Rotations Per Minute

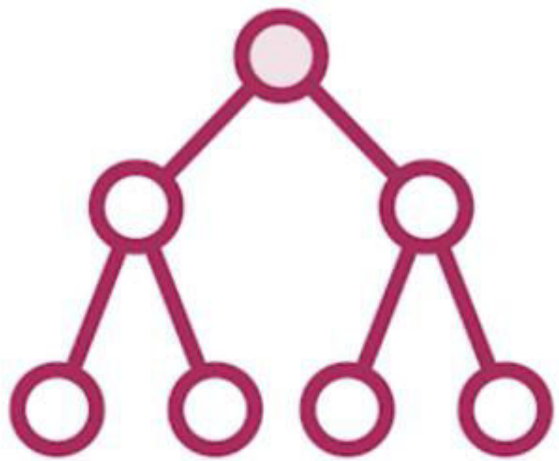
Feature Points

Kilowatts per hour

Hertz

Miles Per Hour

Story Points



- Very common way to estimate work
- Based on size and complexity, not duration
- Unitless and numerically relative
- Different for each team of estimators
- Points are additive, unlike time
- Based on historical reality
- Easy to use and understand

Using Story Points

Pile o'
User Stories

Defect A | Cost: 20

Defect B | Cost: 30

Requirement A | Cost: 100

Requirement B | Cost: 100

Requirement C | Cost: 30

Constraint A | Cost: 20

We can see right away

Which work items cost the most

Total cost of all the work

Total cost to an iteration

Story Point Values

(Include big and small outliers if you want: 0, 1/2, 100, 300)

Can you distinguish a 1-point story from a 2?

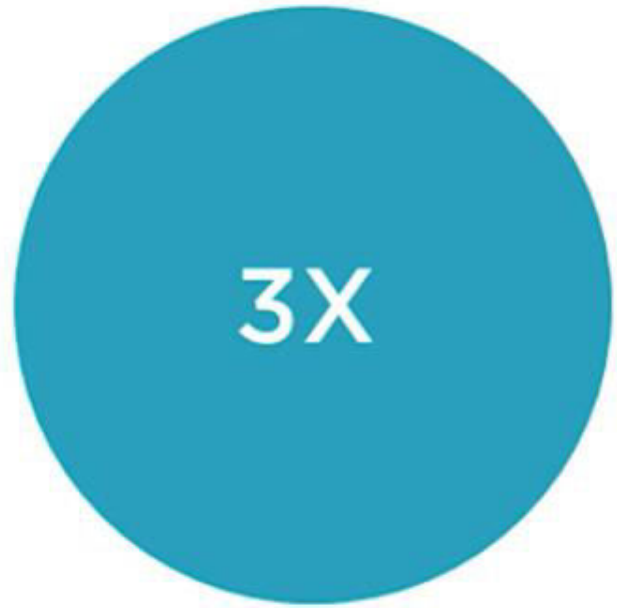
Can you distinguish a 17 from an 18?

How about a 99 from a 100?

Use units that make sense:

- XS, S, M, L, XL, XXL
- 1, 2, 3, 5, 8, 13, 20, 40
- 1, 2, 4, 8, 16, 32

Project Last Week



This Project

Estimates are Not Promises

If estimates are used against you this is a people problem, not a problem with the estimates. Address it.

Planning Poker

Estimating with Groups

Group derived estimates are demonstrably more accurate than estimates by individuals

Political Trading Marks

Iowa Electronic Markers

Intrade.com

Politicalmarket.cnn.com

Who Wants to be a Millionaire?"

Polling the audience is accurate 91% of the time

When guessing the number of jellybeans in any given jar, the average of all guesses is typically within 2–3% of the correct answer.

— The Wisdom of Crowds, James Surowiecki

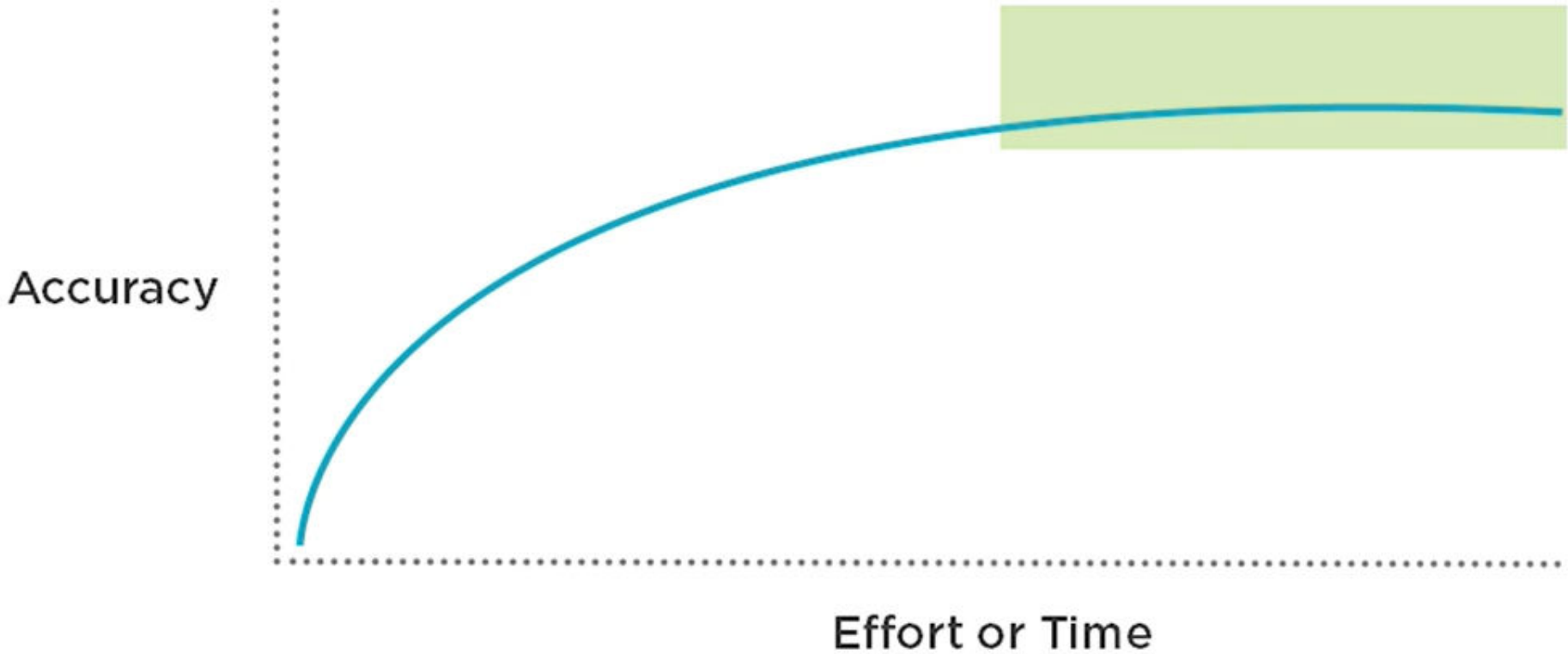
Together, we are smarter than any one of us.

Japanese proverb

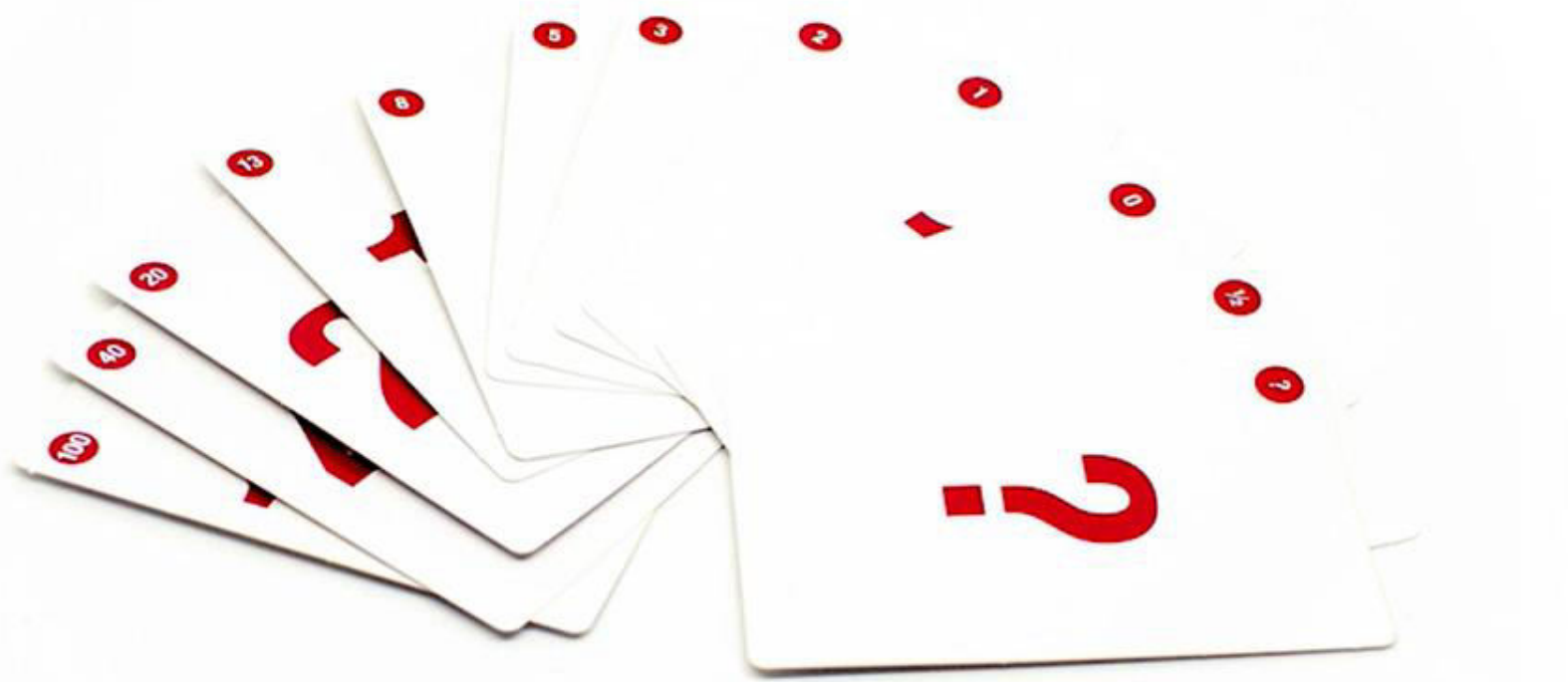
Myth

With more time, estimates get significantly more accurate

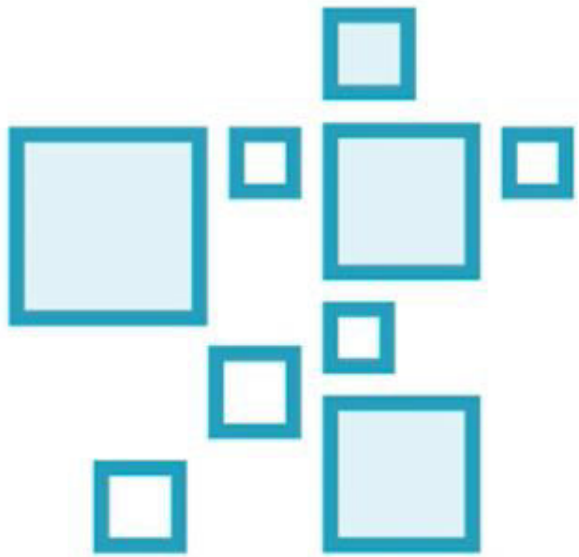
Estimation is Expensive



Planning Poker Cards



Why Planning Poker Works



- Emphasizes relative sizing
- Focuses most estimates within an order of magnitude
- Everyone is heard
- Finds hidden requirements and details
- Estimators must justify estimates
- It is iterative

Planning Poker Rules

1. Each estimator has a deck of estimation cards

2. Customer/Product Owner reads a story and it's discussed briefly

3. Each estimator selects a card that's his or her estimate

4. Cards are turned over so all can see them (synchronously)

5. Discuss differences (especially outliers)

6. Re-estimate until estimates converge

A Real Work Item

Check table widths before Check-in or Save

Before a user saves or checks in a document, test all of the tables in the document to see if they follow the XHTML rules. If they don't, throw a warning to the user.

Don't throw an error.

3

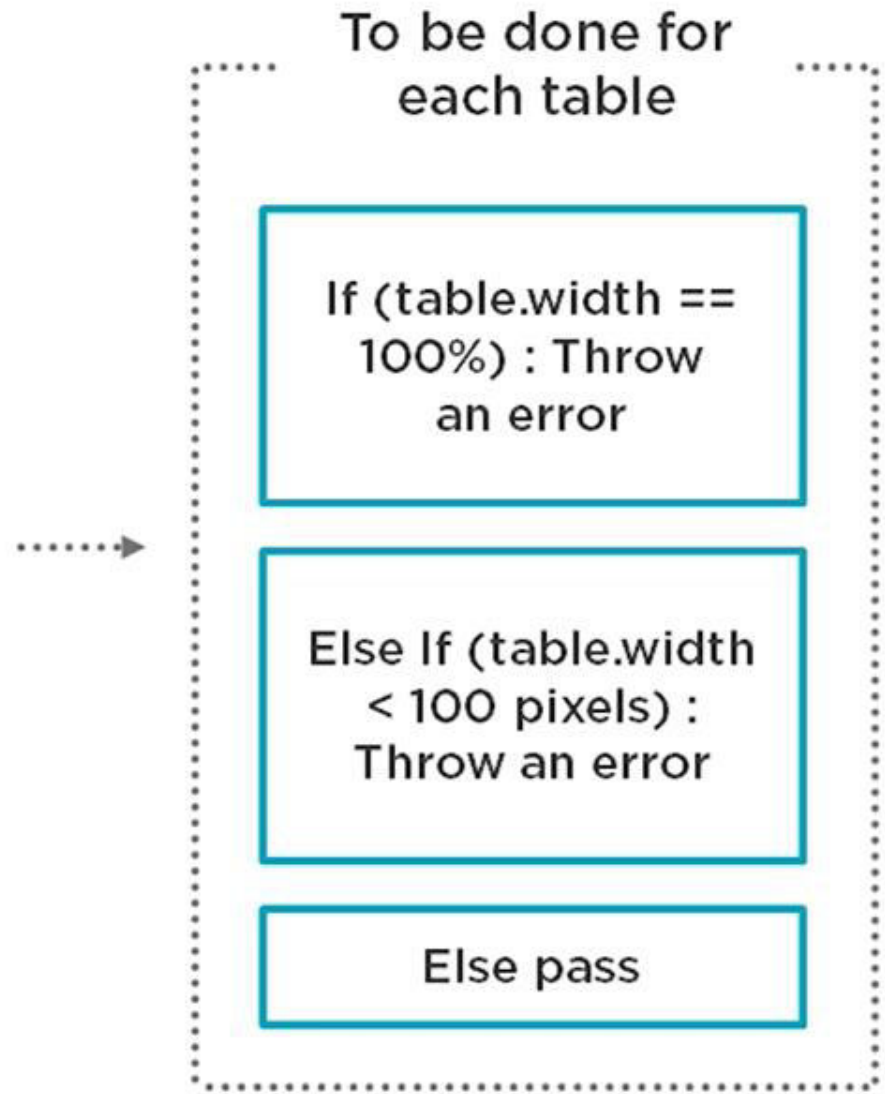
A Real Work Item

Check table widths before Check-in or Save

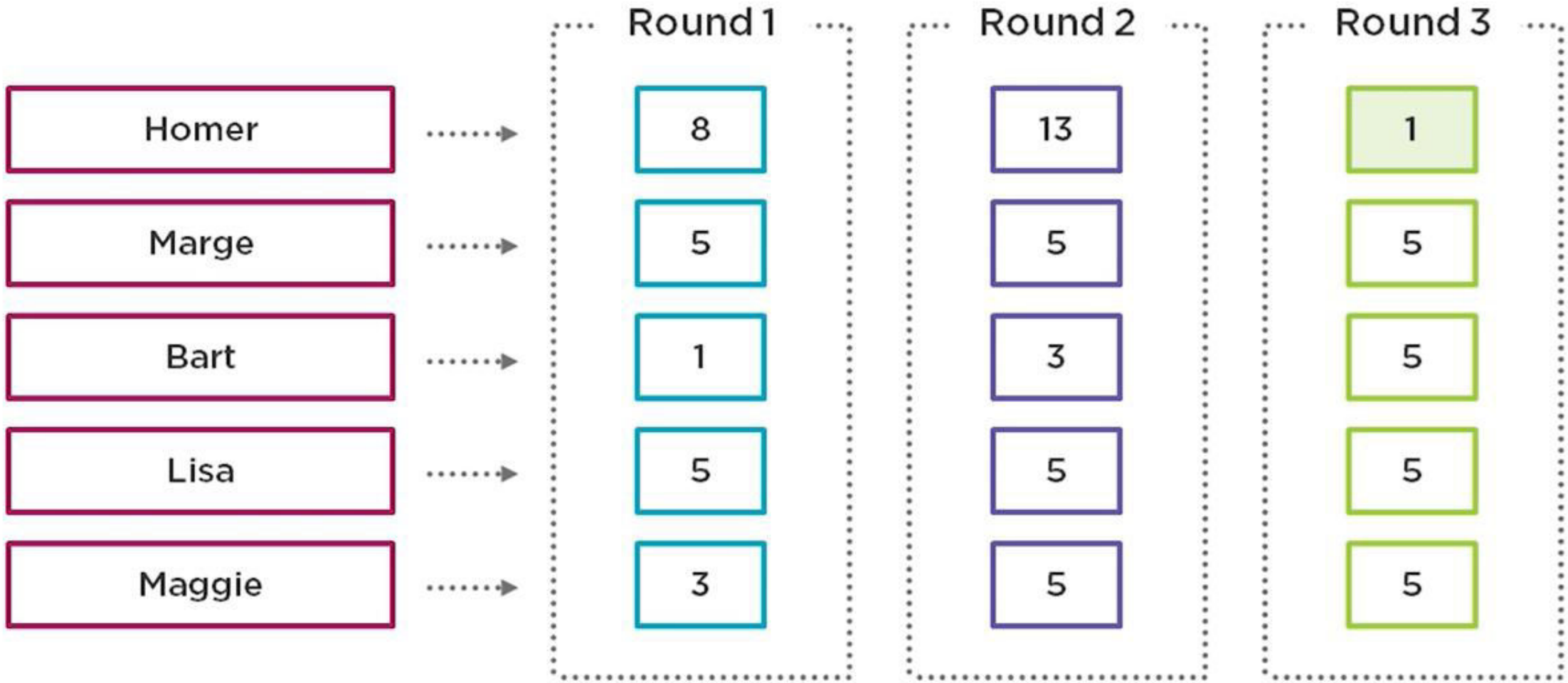
Before a user saves or checks in a document, test all of the tables in the document to see if they follow the XHTML rules. If they don't, throw a warning to the user.

Don't throw an error.

5



Planning Poker



Options for Handling Conflict

Aim for consensus, not unanimous agreement

- Wait for convergence
- Average the estimates
- Toss out high and low
- Send the item back for re-definition

Make a Planning Poker Deck

?, 1, 2, 3, 5, 8, 13, 20, 40...

Try These

Backlog Item	Estimate
Mow my lawn	5
Move your slacker friend from his mom's house	
Paint my house	
Write Pong in Silverlight	
Add a new team member	
Make 8 pounds of confetti	

References

Blink: The Power of Thinking Without Thinking, Malcolm Gladwell

The Wisdom of Crowds, James Surowiecki

Agile Estimating and Planning, Mike Cohn

User Stories Applied for Agile Software Development, Mike Cohn

PlanningPoker.com