

CLEAN AGILE

BACK TO BASICS



CURMUDGEONLY GRUMBLINGS.

- My personal recollections and observations over 20 years.
- Programmers and non-programmers alike
- Non-technical.



A SMALL IDEA

- Agile is a small idea about the small problem of small programming teams doing small things.
- A small idea from the 1950s that we lost in the 1970s because we thought we should be doing big things.
- Why was this lost?

THE DISCONTINUITY

- The original programmers were old.
- The knee of the exponential growth in the '70s.
- An impossibly young cohort of twenty-somethings (mostly male).

THE RE-AWAKENING

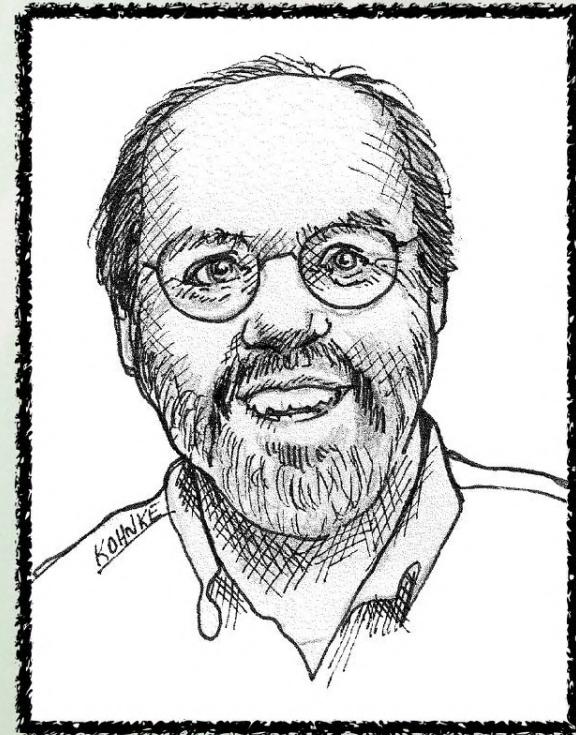
- The mid-'90s reboot of the small idea that spread like wildfire.
- We called it Agile.
- And it changed everything.

TWENTY YEARS ON...

- The idea has been muddled.
 - Lean, Kanban, SAFe, LeSS, modern, skilled, etc. etc. etc.
 - Not bad ideas.
 - Not the original agile message.
- Time for another reboot of Agile: the small idea about small things.

ACKNOWLEDGED

- Ward Cunningham
- Kent Beck



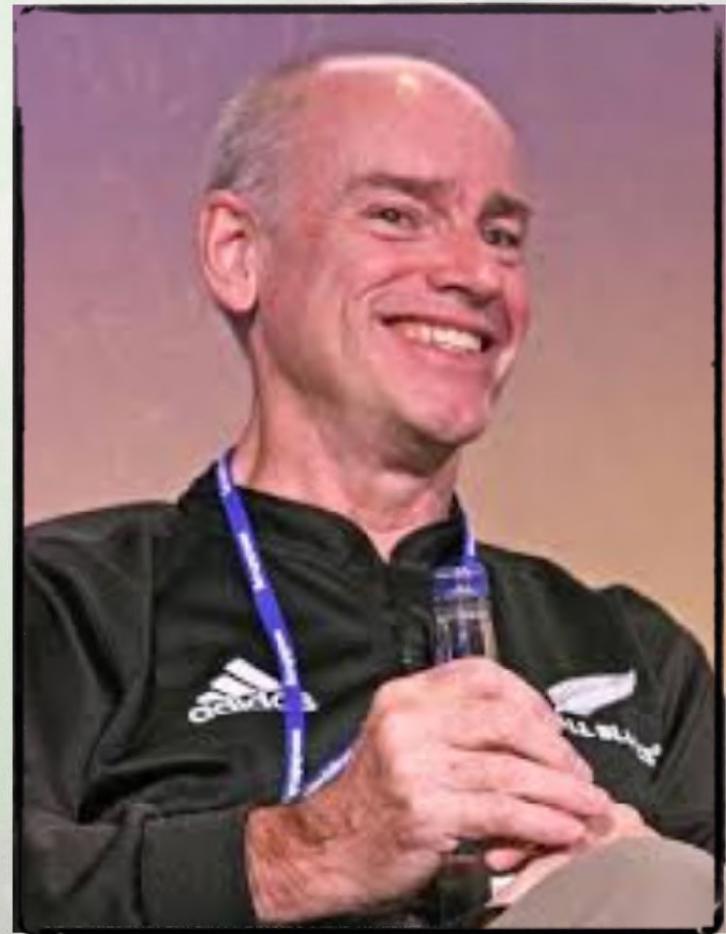
ACKNOWLEDGED

- Martin Fowler
 - A steady hand.



ACKNOWLEDGED

- Ken Schwaber
 - Indomitable energy.



MARY POPPENDIECK

- Selfless and inexhaustible.



ACKNOWLEDGED

- Ron Jeffries, the warm and gentle, conscience of the early days of Agile.



ACKNOWLEDGED AND IN MEMORIUM

- Mike Beedle
 - Fought the good fight.



ACKNOWLEDGED

- Arie van Bennekum,
- Alistair Cockburn,
- James Grenning,
- Jim Highsmith,
- Andrew Hunt,
- Jon Kern,
- Brian Marick,
- Steve Mellor,
- Jeff Sutherland,
- Dave Thomas.

ACKNOWLEDGED

- Jim Newkirk
 - My business partner at the time.
 - Tireless effort...



ACKNOWLEDGED

- Object Mentor



ACKNOWLEDGED

- Agile Alliance Founding.





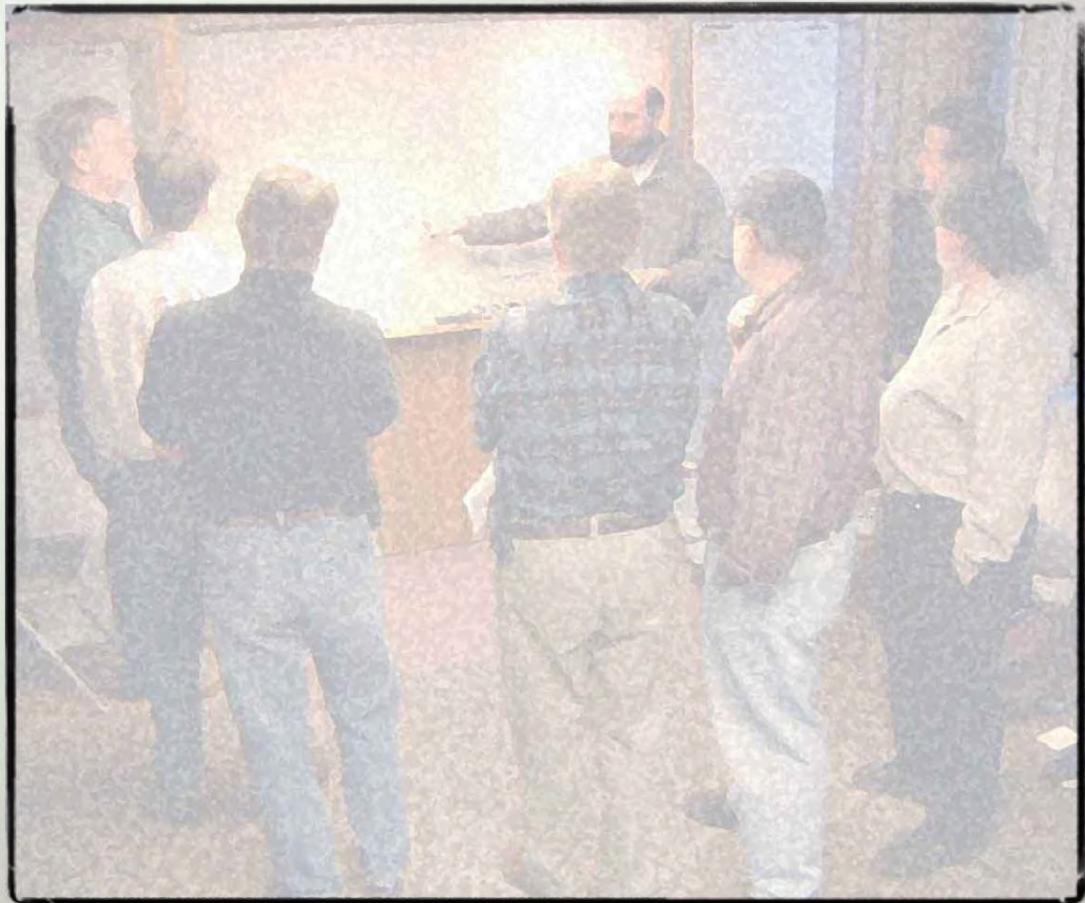
INTRODUCTION TO AGILE.



FEBRUARY, 2001

SNOWBIRD, UTAH

- 17 Experts.
- Deplorable state of Software.
- Light Weight Process Summit.
- Agile is born.
- But how did it all begin?



THE HISTORY OF AGILE



IN THE BEGINNING...

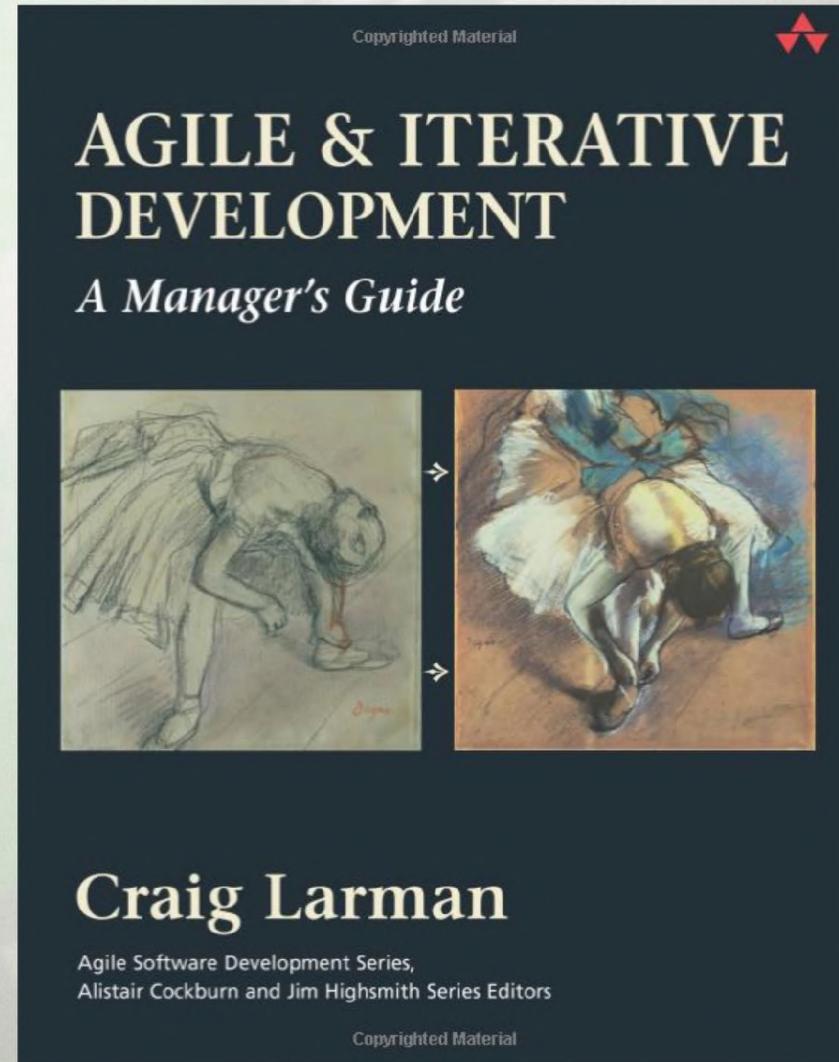
- 50,000 years ago?
 - The idea of choosing small intermediate goals and measuring the progress after each is just too intuitive, and too human, to be considered any kind of a revolution

IN MODERN INDUSTRY?

- It's hard to say.
 - the first steam engine, mill, internal combustion engine, airplane...
 - Probably produced by techniques that we would now call Agile....taking small measured steps is just too natural and human for it to have happened any other way.

IN SOFTWARE?

- Turing?
- Mercury Space Capsule
- IBM Federal Systems Division
- ...in the wild.

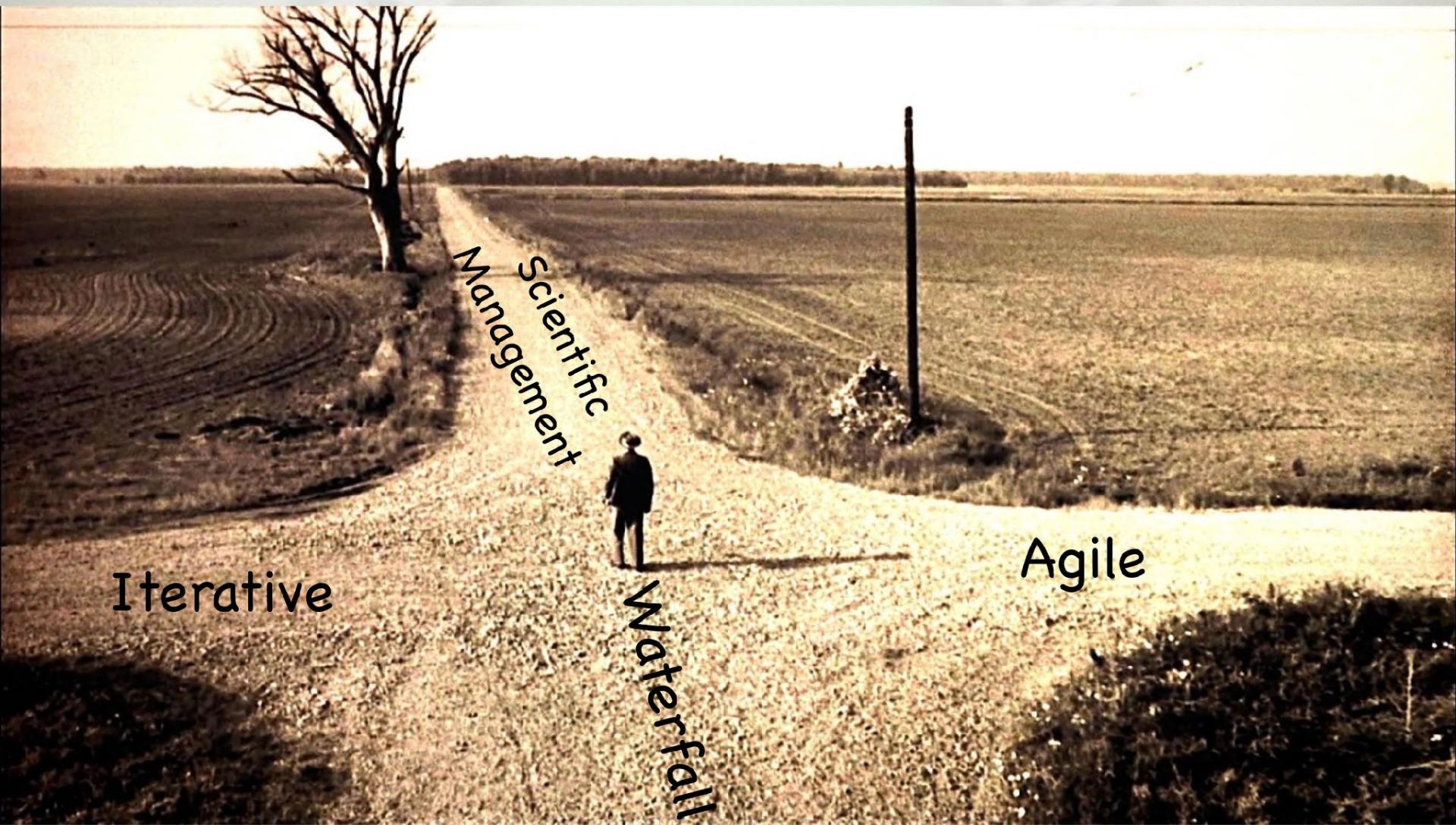


NOT THE ONLY GAME IN TOWN

- Scientific Management
 - Fredrick Winslow Taylor
- As old as the Pyramids, Stonehenge, etc.
- Because the idea of repeating a successful process is just too intuitive, and human, to be considered some kind of a revolution.



CROSSROADS IN THE '70S

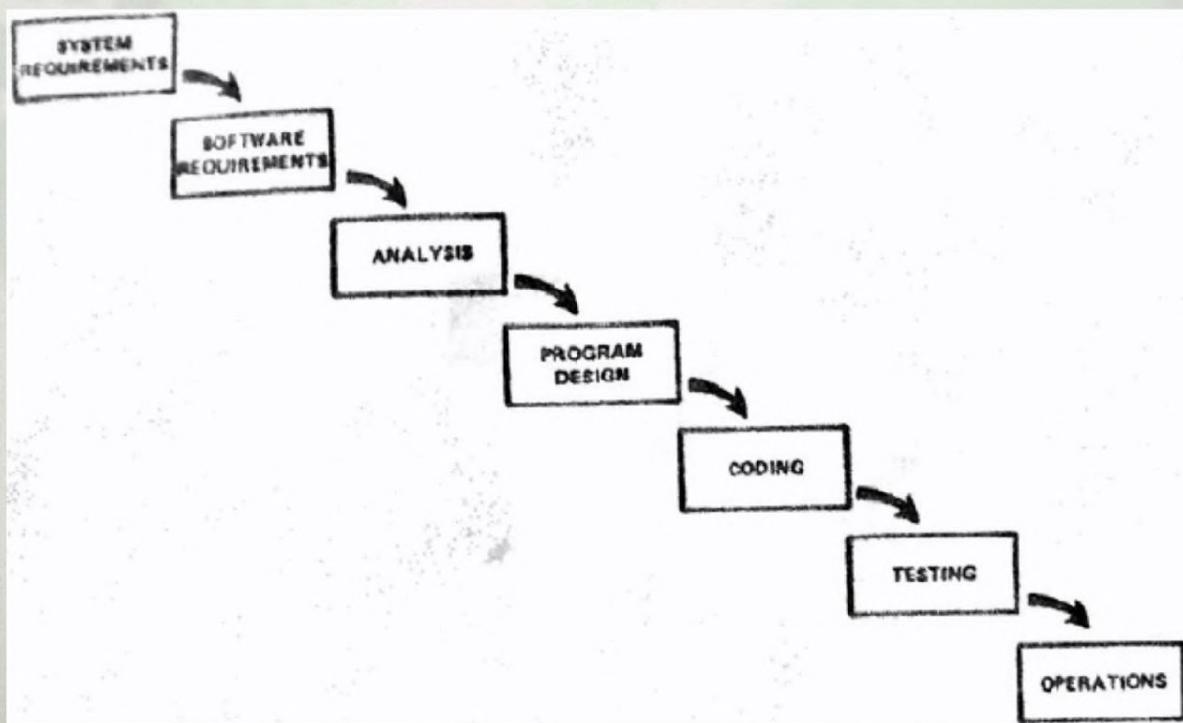


THE DILEMMA

- Pre-agile
 - Best for low cost-of-change and partially defined projects.
- Scientific Management
 - Best for high cost-of-change, well defined projects.
- Which is Software?

THE WATERSHED.

- 1970. Winston Royce.
- *Managing the Development of Large Software Systems*



I ENTER HERE...

- Age 18.
- Hired as a programmer at ASC.
- 360s. V620f. Assembler.

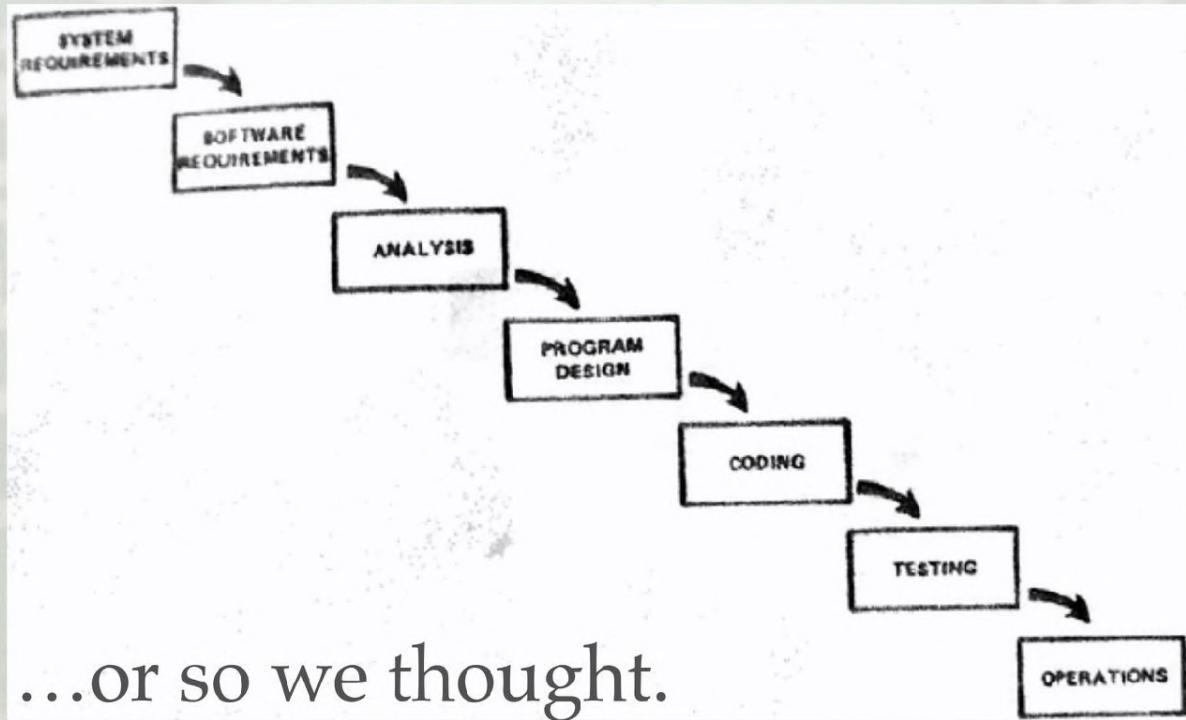


WHAT PROCESS DID WE USE?



¹ A standard card form, IBM electro 688157, is available for punching statements from this form.

WATERFALL WAS A GODSEND!

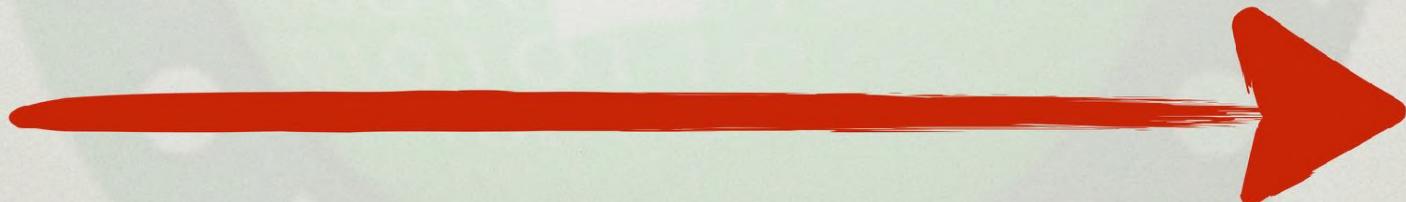


**...AND IT DOMINATED US
FOR 30 YEARS.**

1970



The Waterfall era



2000

...it dominated, but it didn't work.

THIRTY YEARS OF FAILURE

- We tried.
- and tried.
- and tried.
- How could this fail?
- Try harder...
-

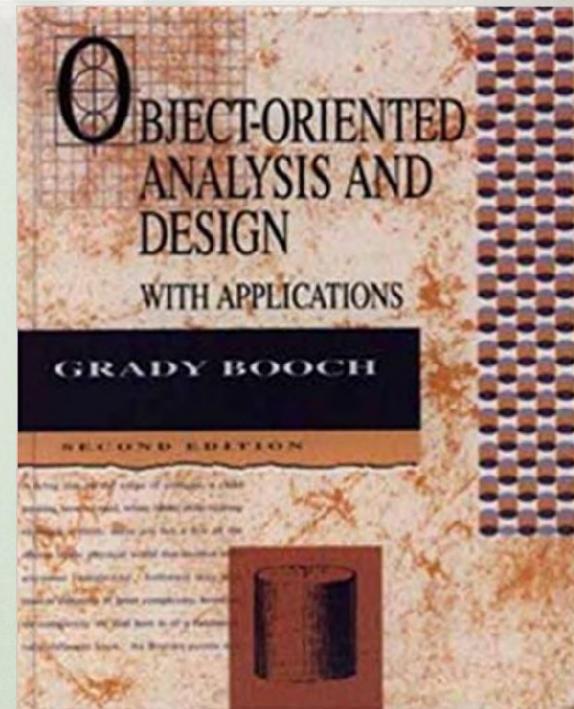


THE LEVEL OF INDOCTRINATION...

- The triumvirate of triplets...
 - OOP -> OOA & OOD
 - SP -> SA & SD
- We could not conceive of another way to work...

...AND THEN SUDDENLY WE COULD.

- Hints of Agile in the Smalltalk community in the late '80s
- Booch's 1991 book added more.
- Cockburn's Crystal methods.
- The Patterns community.
 - Cope's 1994 paper.
 - SCRUM, Schwaber, Beedle, Devos



I ENTER AGAIN.

- C++ Consultant asked about process.
- XP
- OOP '99, Kent Beck.
- Medford.
- XP Immersions.

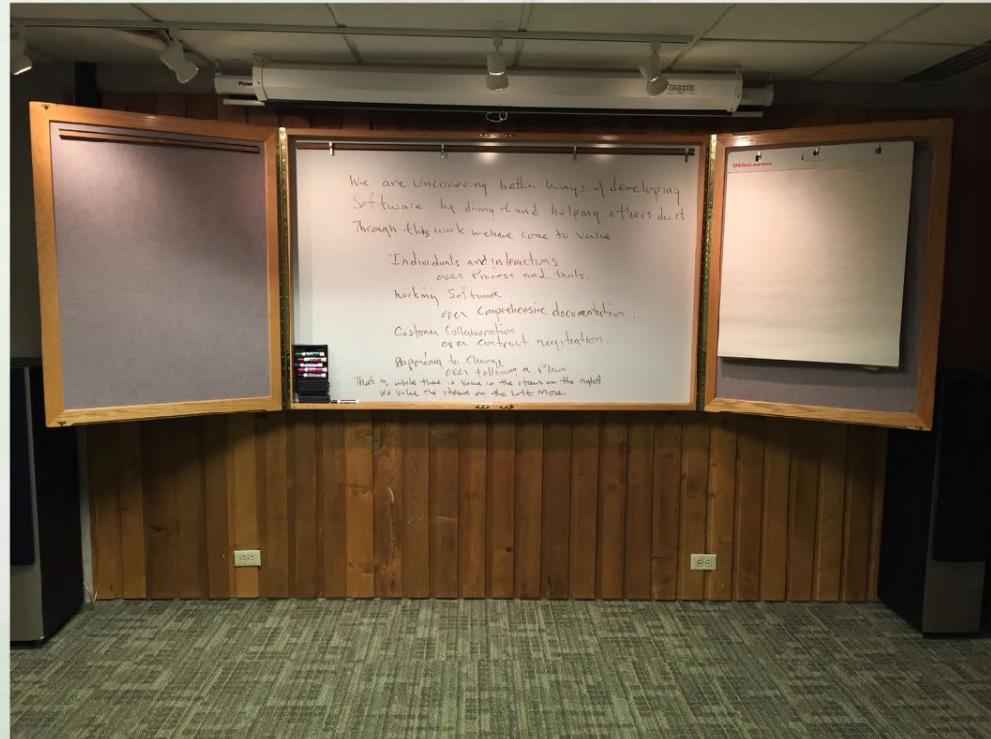


THE LEAD-UP TO SNOWBIRD.

- XP-Leadership.
- The meeting with Fowler
- Invitations.
- Cockburn
- The Lightweight Process Summit.

SNOWBIRD

- 17 Attendees!
 - Demographics.
- Diverse viewpoints
 - XP, SCRUM,
FDD, DSDM,
Crystal, Prags.
 - Mellor & Fowler.



THE MEETING.

- I kicked it off with a call for a manifesto.
- Cards
- Bumbling.
- The magic...

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

THE AFTERMATH

- Ward puts up the website and requests signatures.
- The Principles.
- Unexpected Groundswell.
- Crazy growth.
- No repeat.

AGILE OVERVIEW



How do you manage a software project?

- Badly?
- Hope and prayer?
- With Great Difficulty
- Dictate and Motivate?`

Mismanagement of a project leads to:

- Producing the wrong product.
- Producing a product of inferior quality.
- Being late.
- Working 80 hour weeks.

The inevitable trade-off.

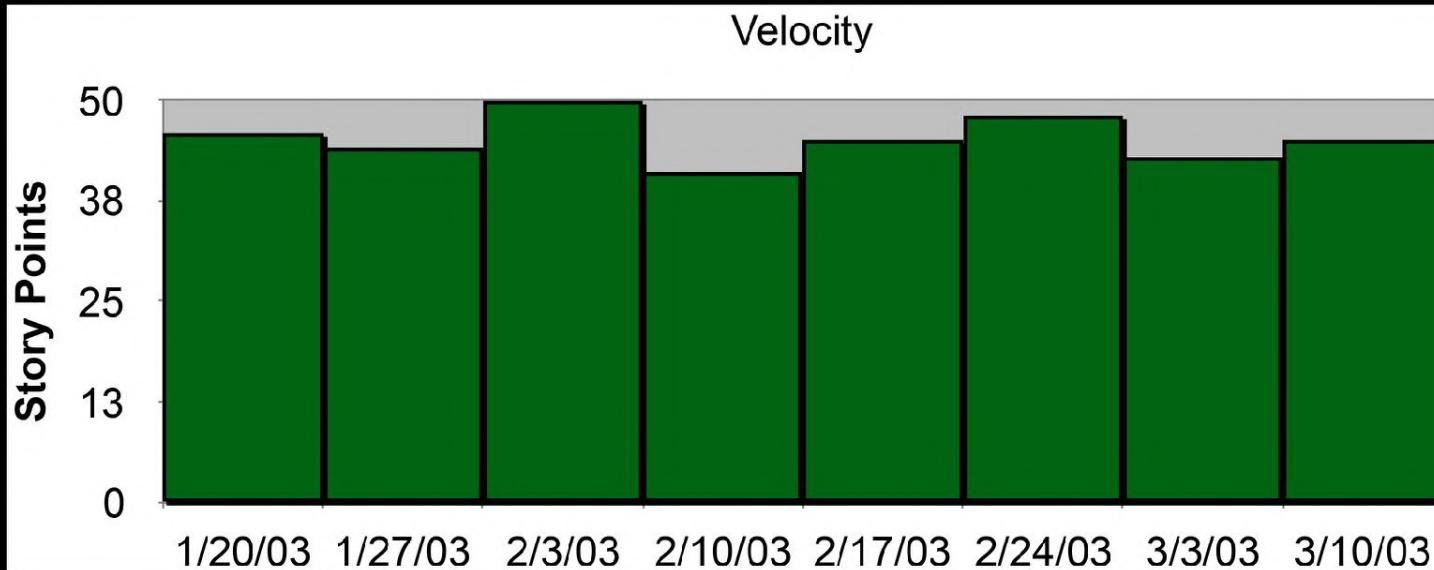
- Good (Quality)
 - Fast (Time to Market)
 - Cheap (Cost Effectiveness)
 - Done
-
- Pick any three...

Finding the optimum solution.

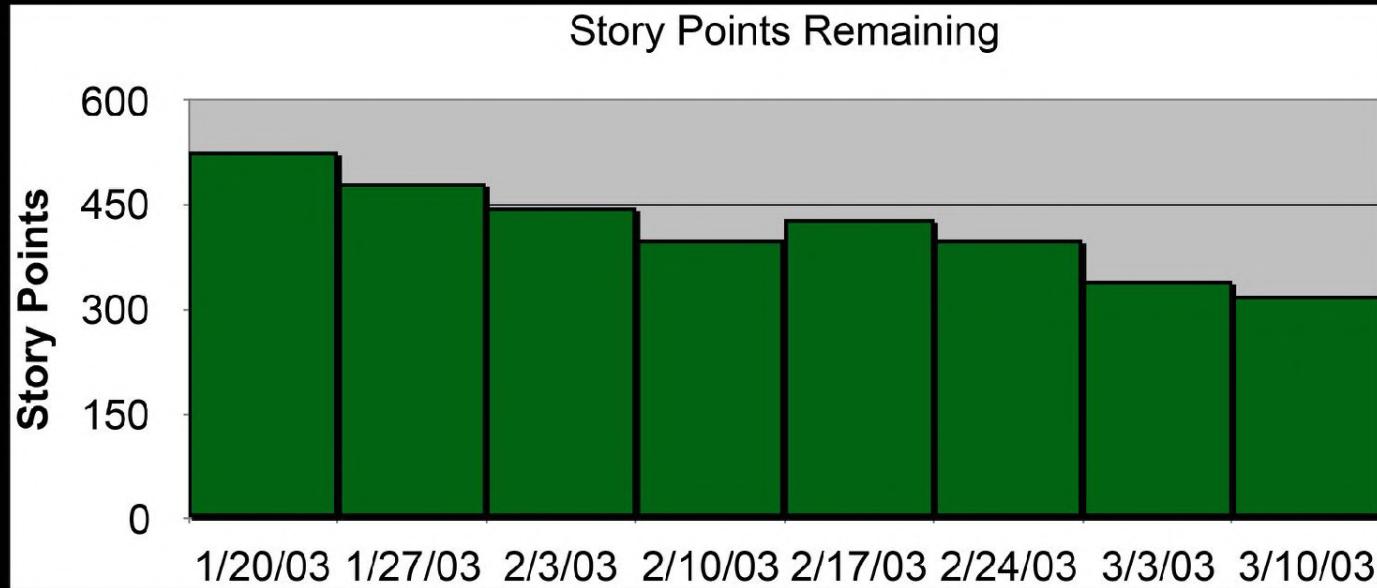
- We need to manage the project to the best possible outcome.
- An outcome that maximizes all four qualities.
- To do this, we need:

Data.

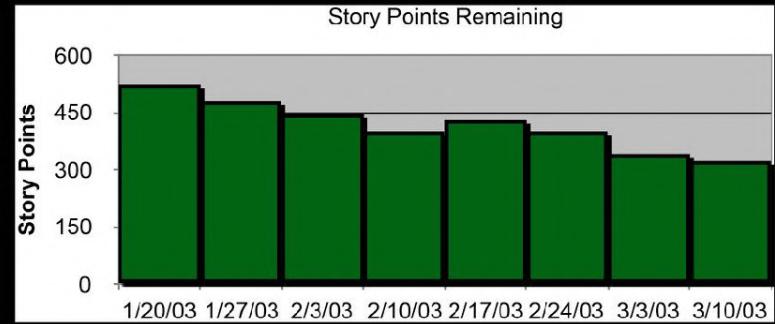
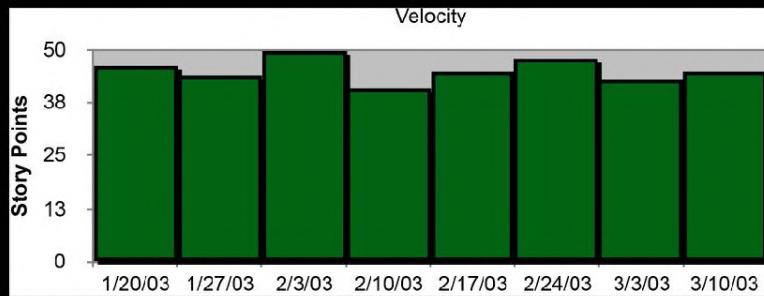
Wouldn't this be great?



...and this...



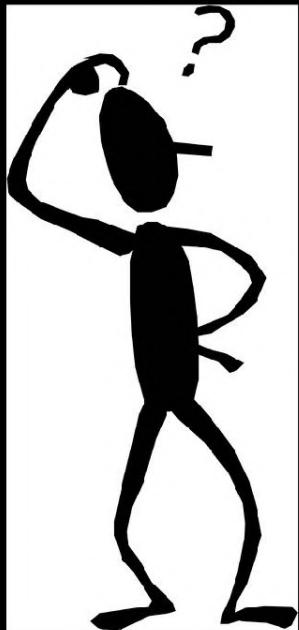
If we had these two charts on the wall...



Then managers could just look at them to see
The status of the project.

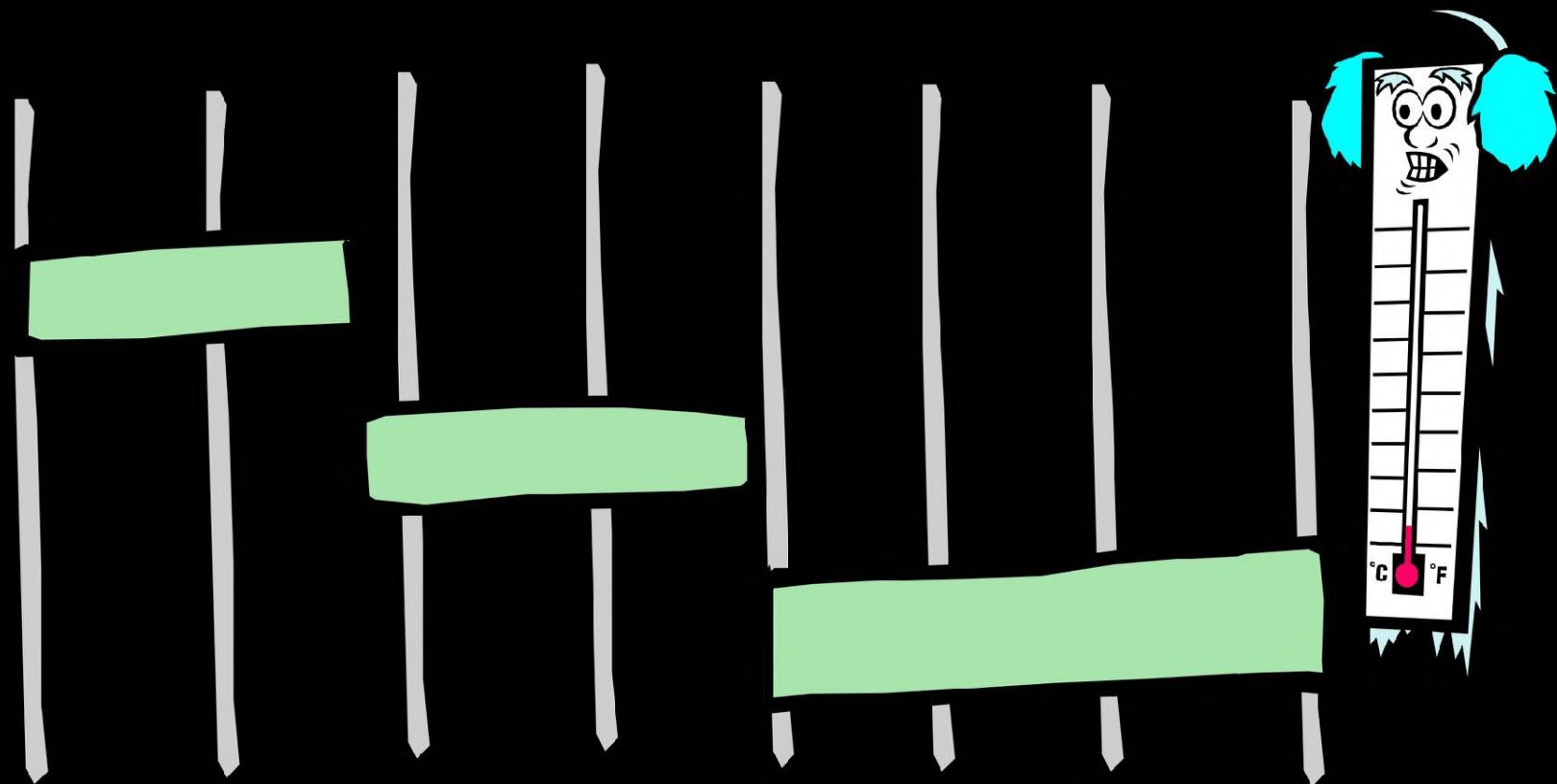
The Management Paradox

*What is the first thing
known about a project?*

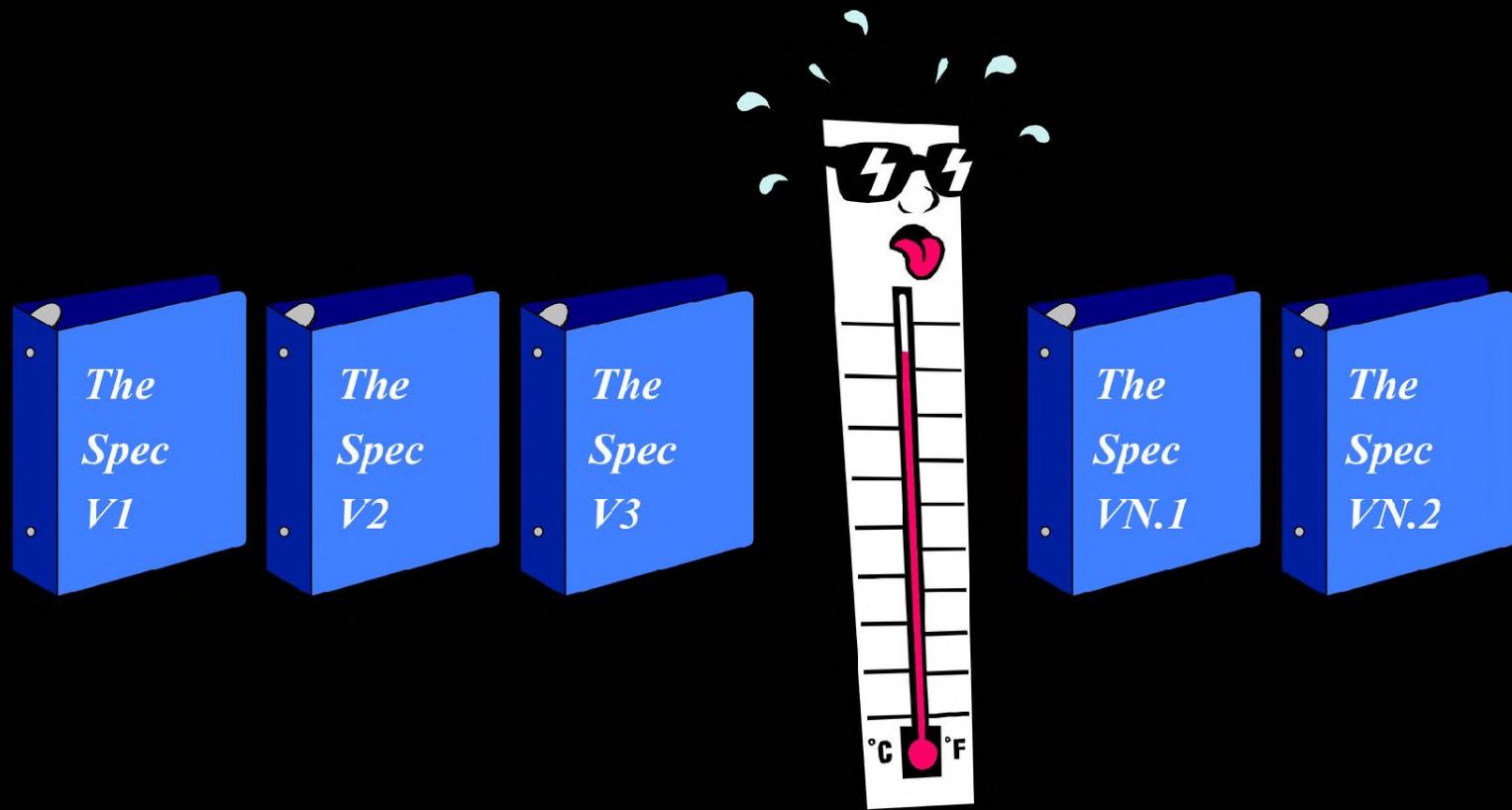




The Delivery Date is Frozen

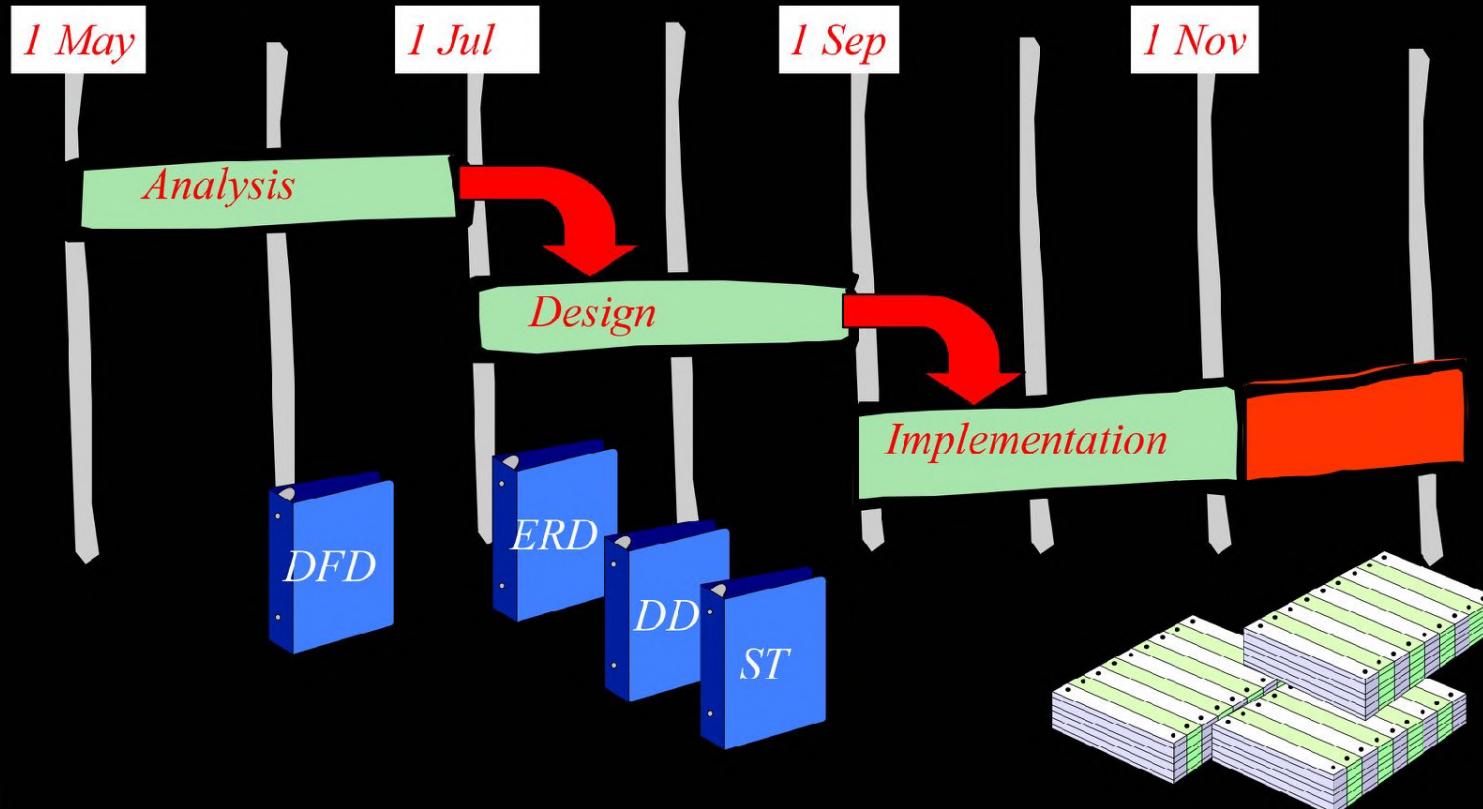


The Spec is Never Frozen

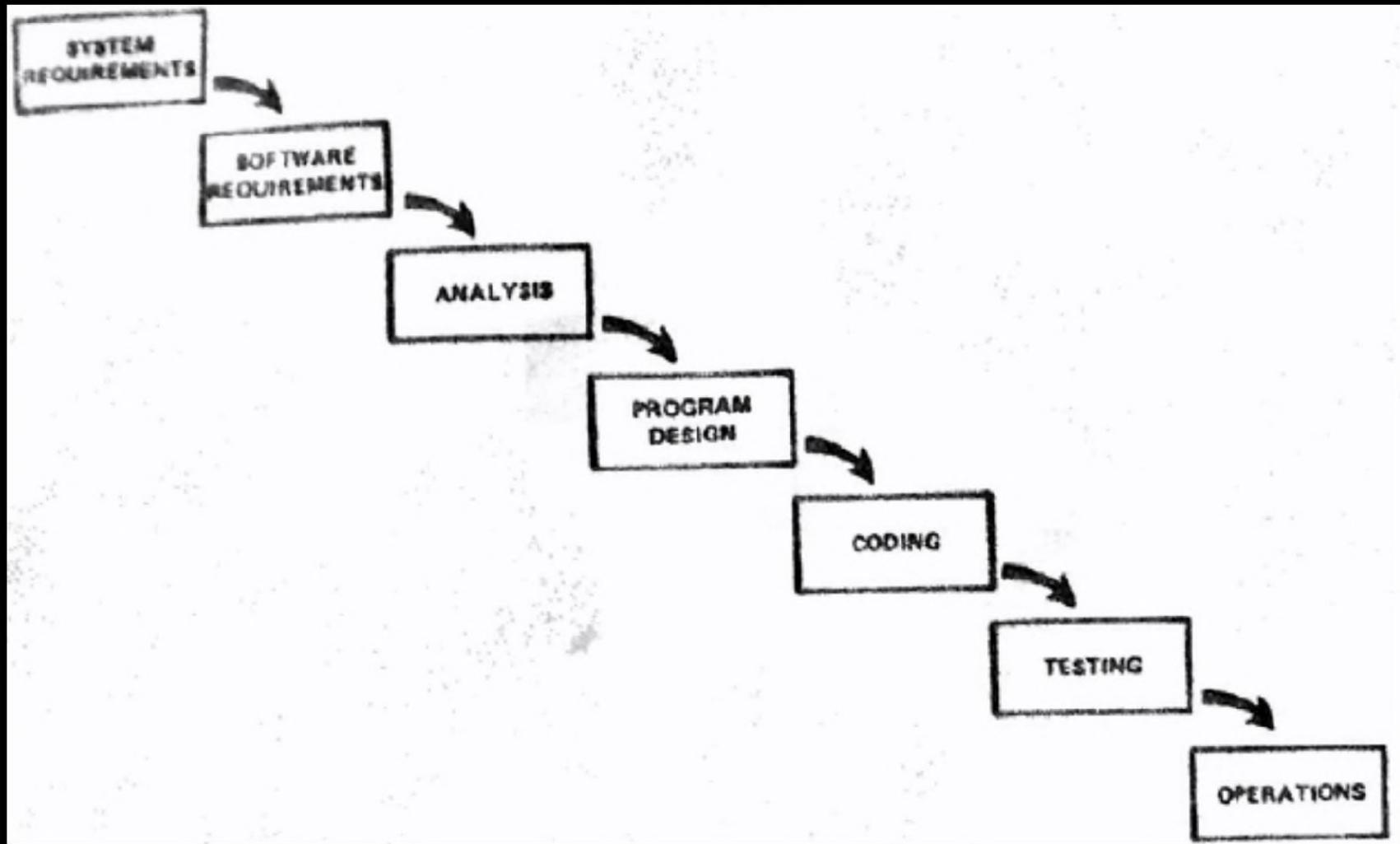


The Waterfall Model

- Managing the Development of Large Software Systems
- Dr. Winston W. Royce — 1970



Royce's actual diagram.



Royce's Observation

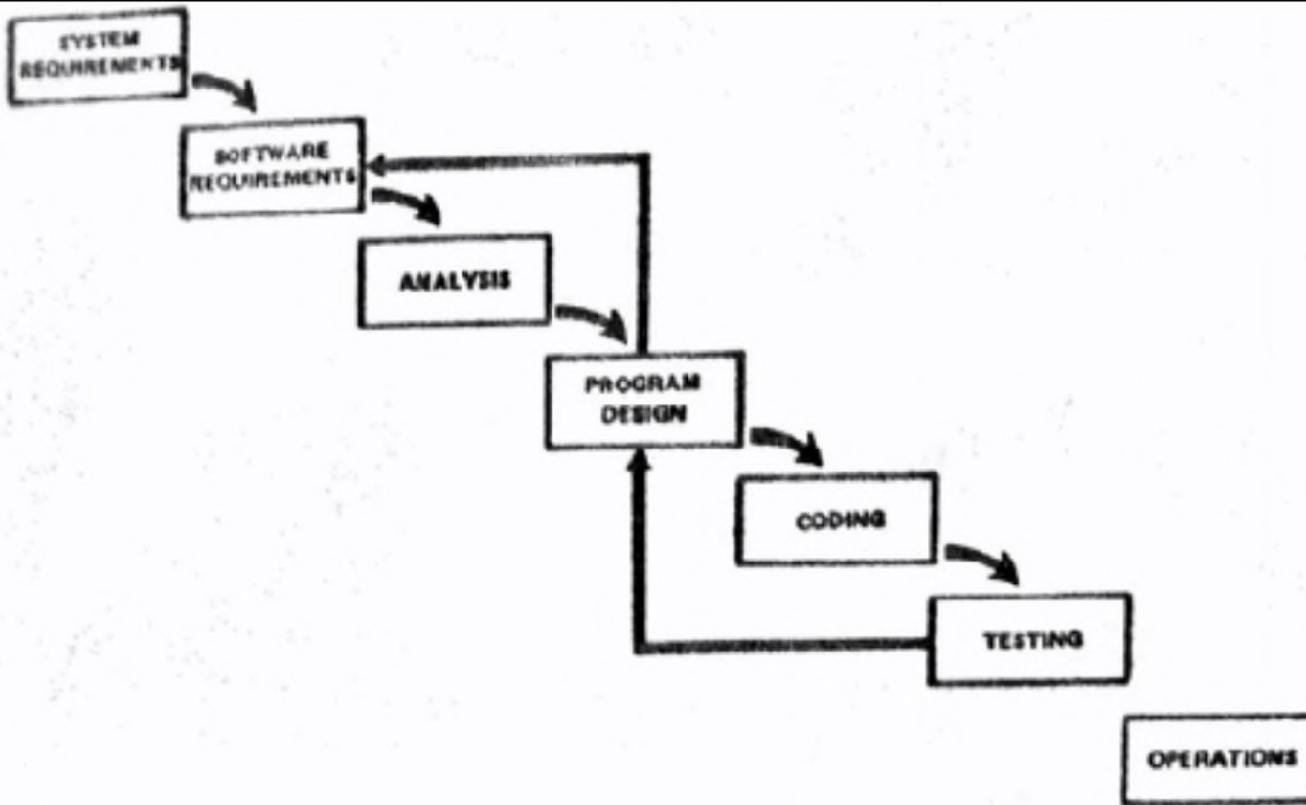
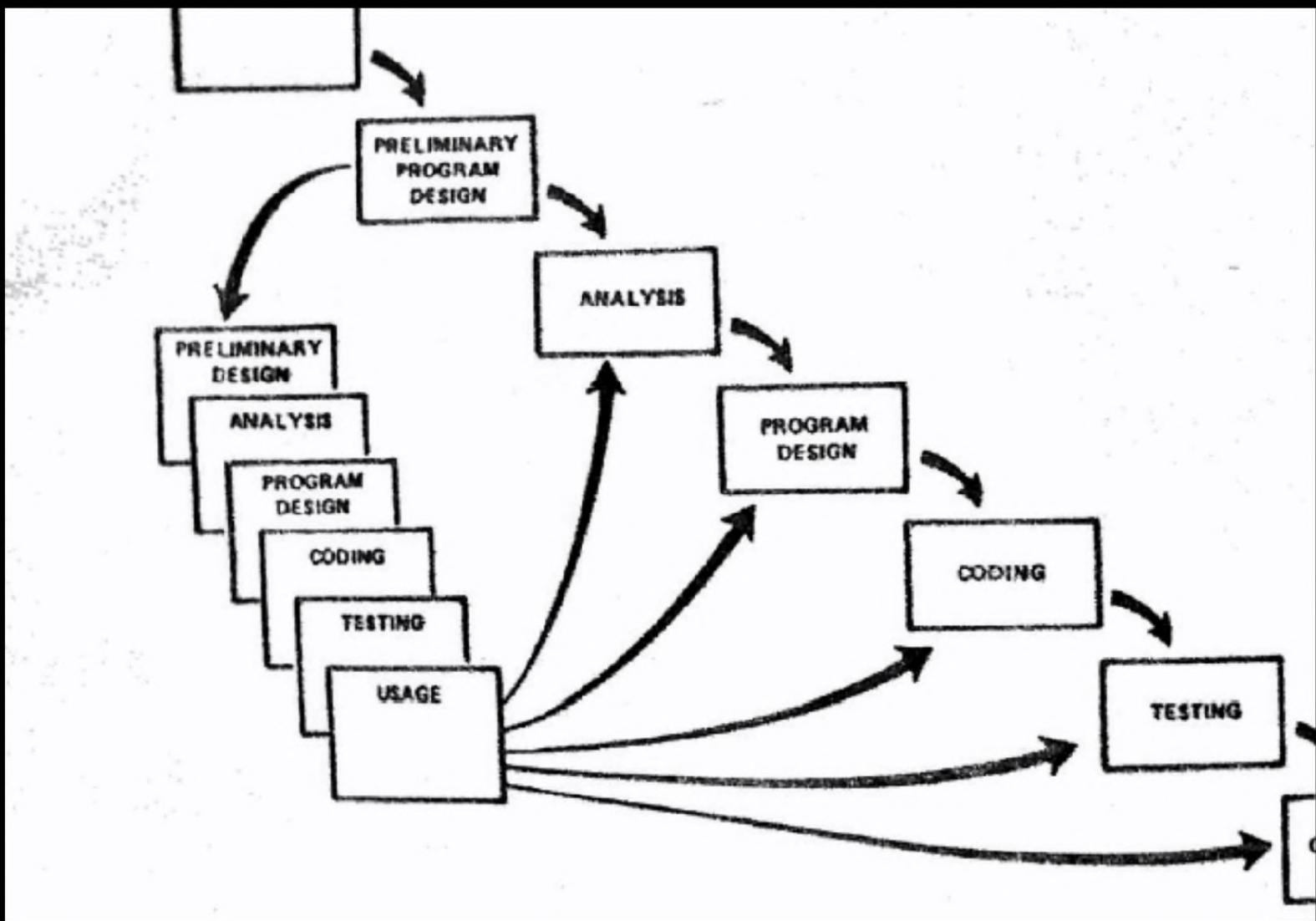


Figure 4. Unfortunately, for the process illustrated, the design iterations are never confined to the successive steps.

Royce's Conclusion

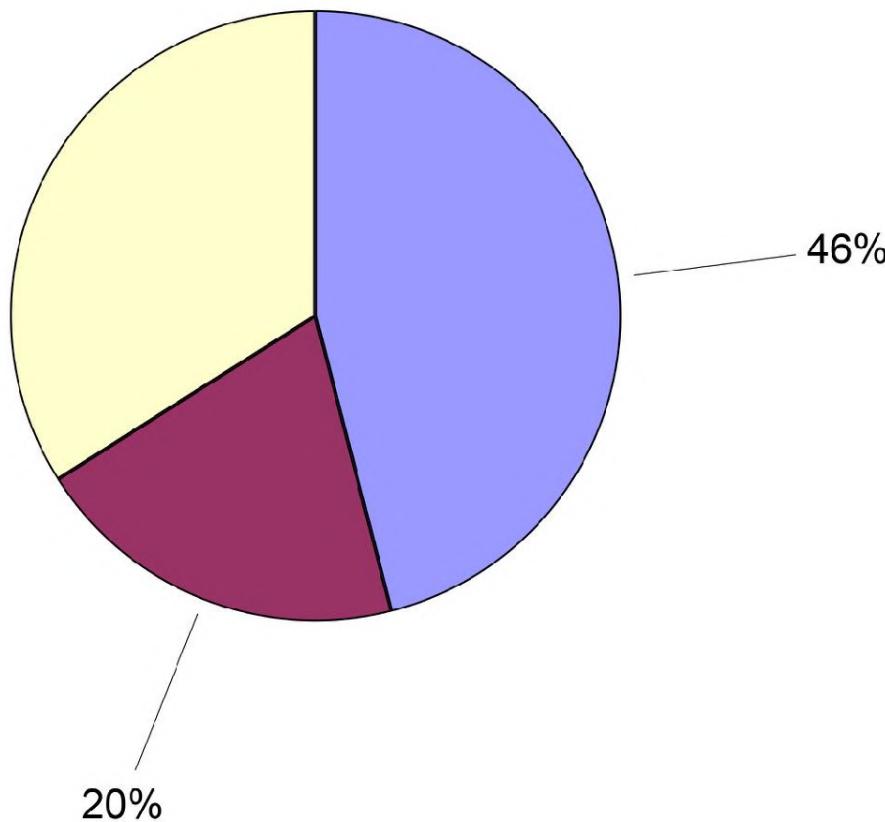


How did W/F get to be the norm?

- DoD asked a Tech Writer to create 2167
- Countries and Companies copied the good ol' DoD.

Requirements and Failure

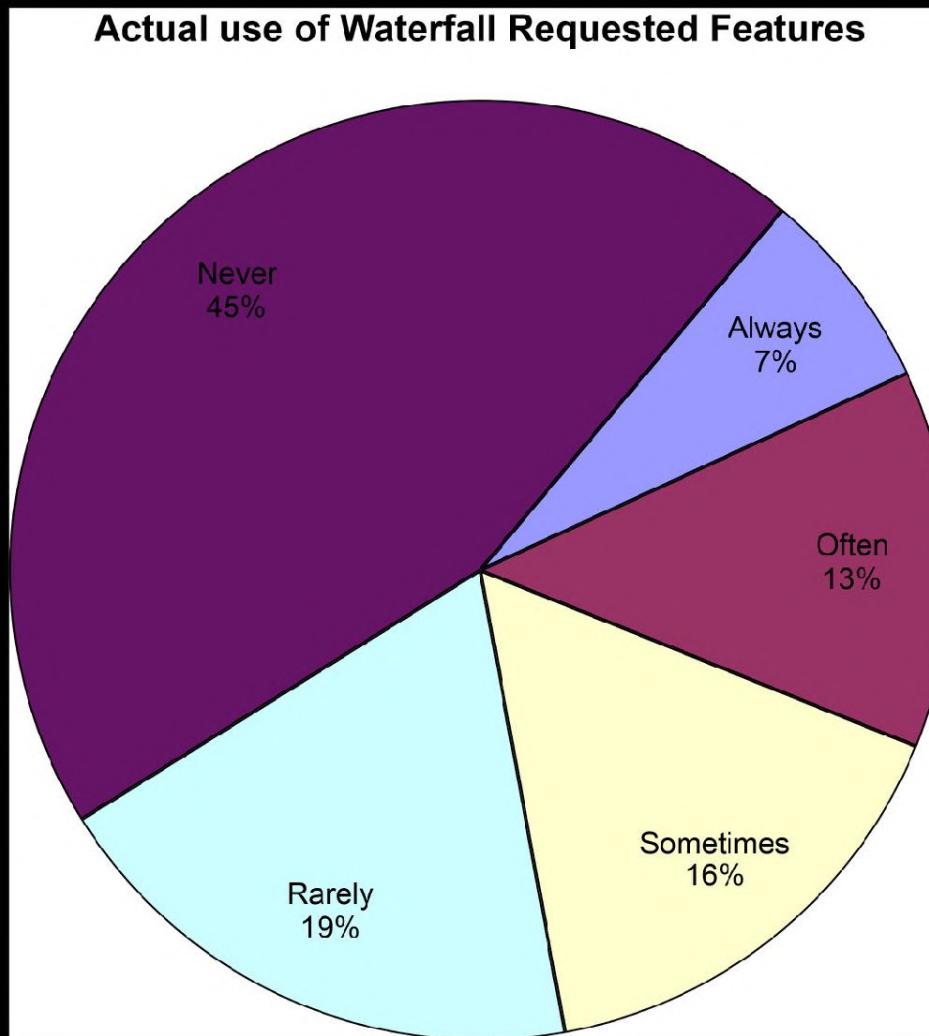
**\$37B worth of DoD
projects using 2167A**



Jarzombek Study.

Failure attributed to use of waterfall.

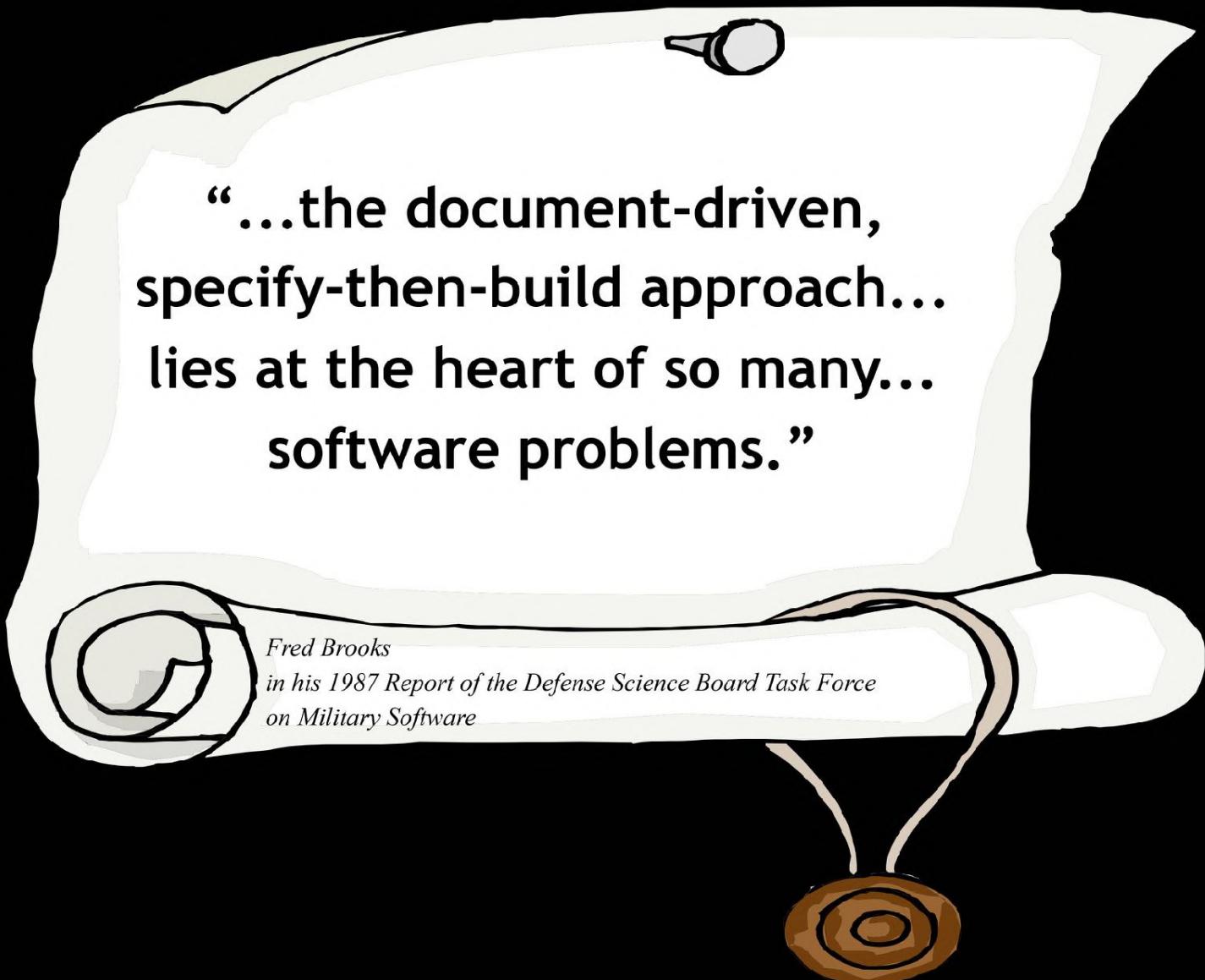
Over specification.



Long Projects Fail.



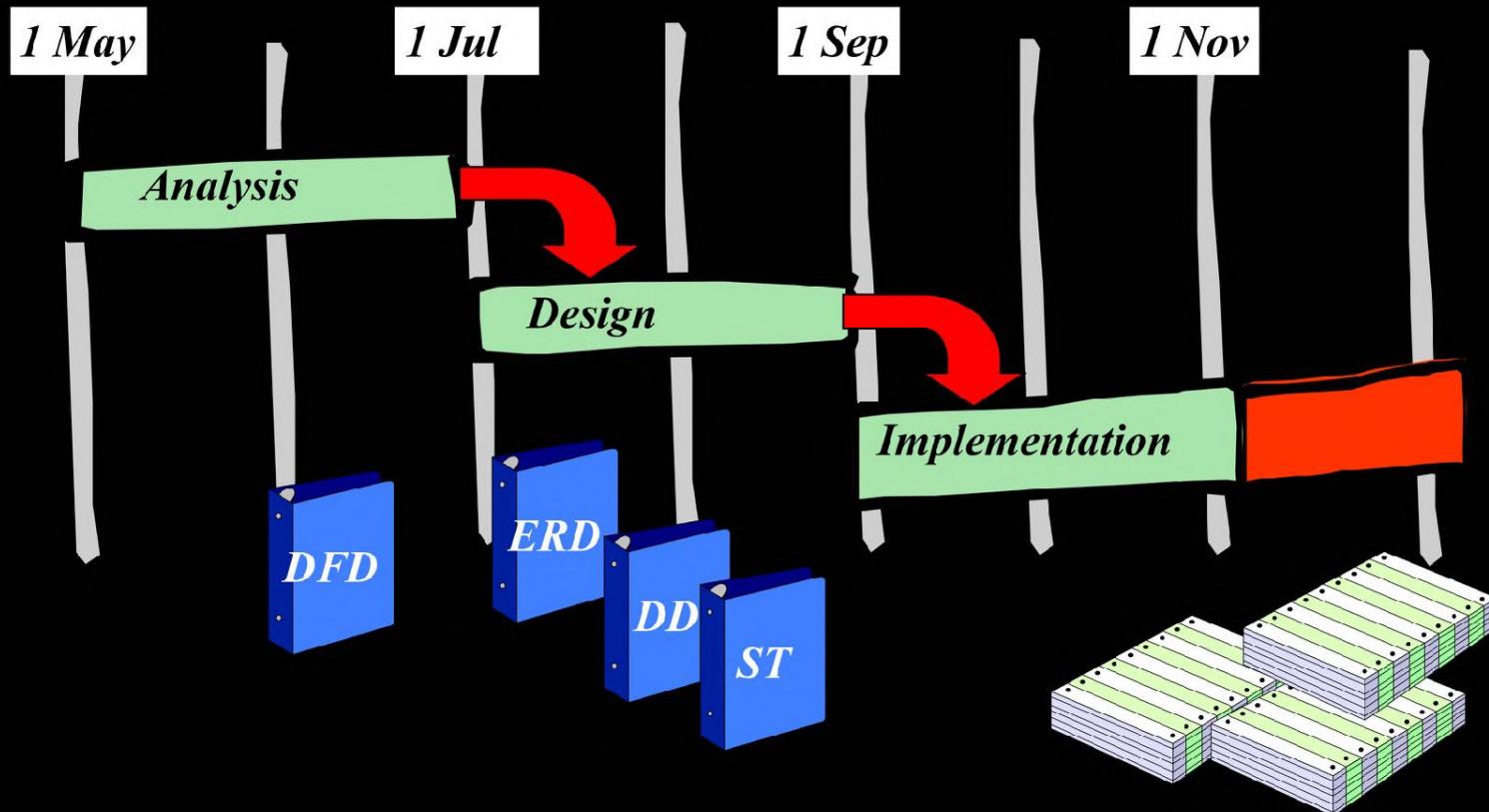
Fred Brooks' Task Force.



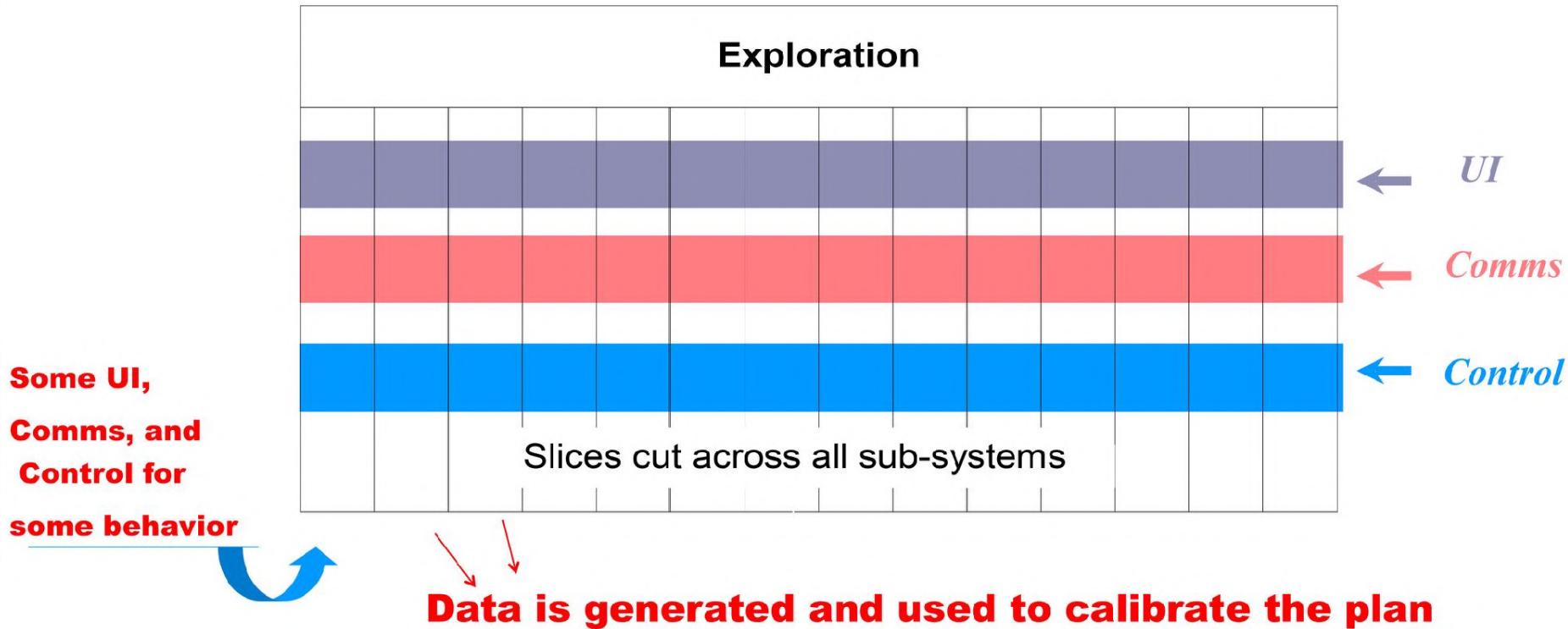
“...the document-driven,
specify-then-build approach...
lies at the heart of so many...
software problems.”

*Fred Brooks
in his 1987 Report of the Defense Science Board Task Force
on Military Software*

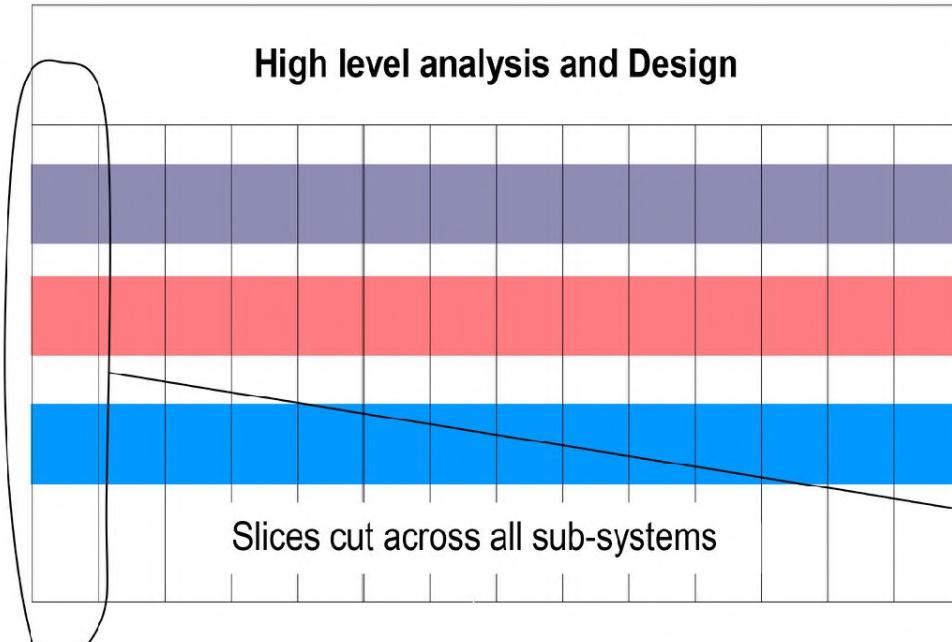
Let's go to a meeting.



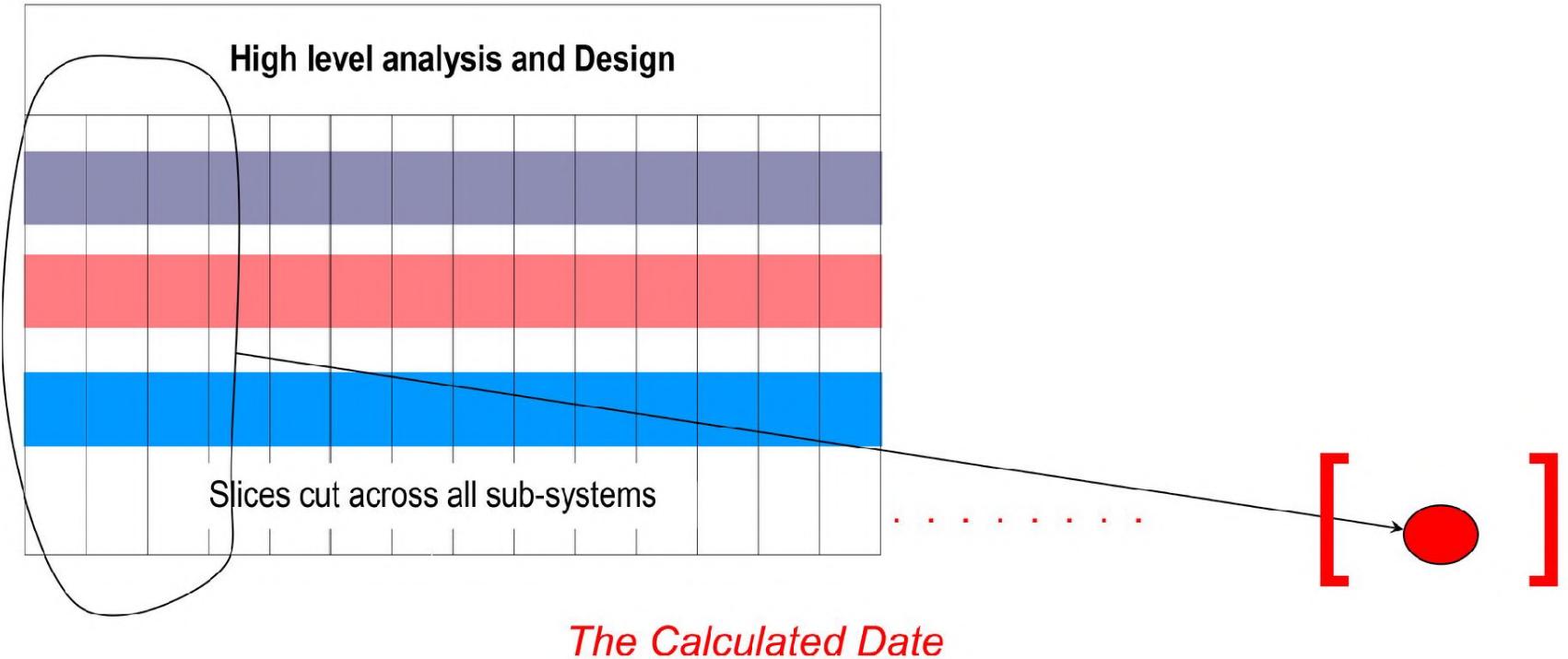
Iterative Development.



Calculate the Date.



More data shrinks the error bars.



Without data, all you can manage is:

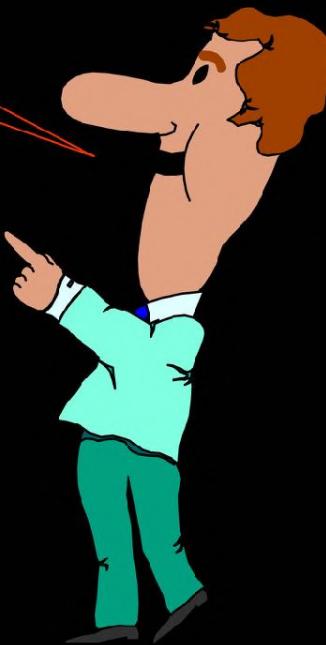


Without data managers can:

Dictate!

This project will be
done on time!

Or HEADS will
ROLL!

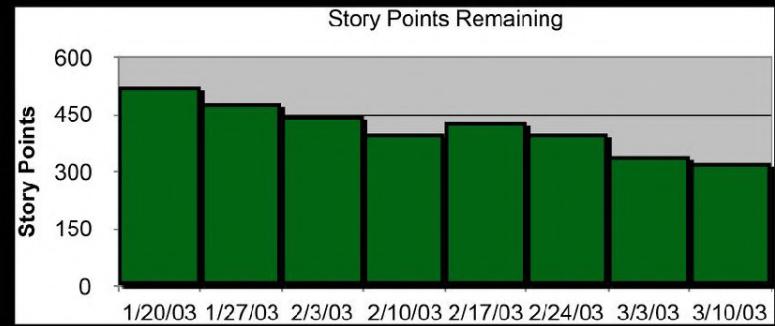
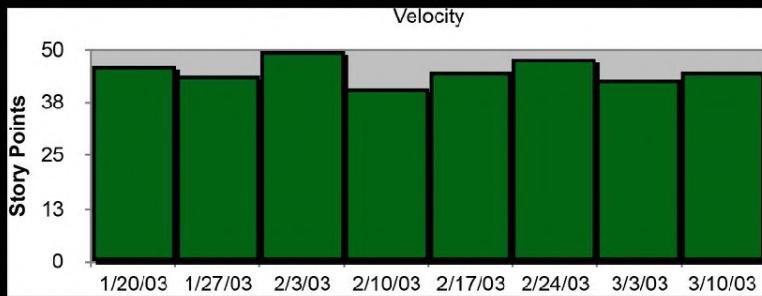


OR....

Motivate.



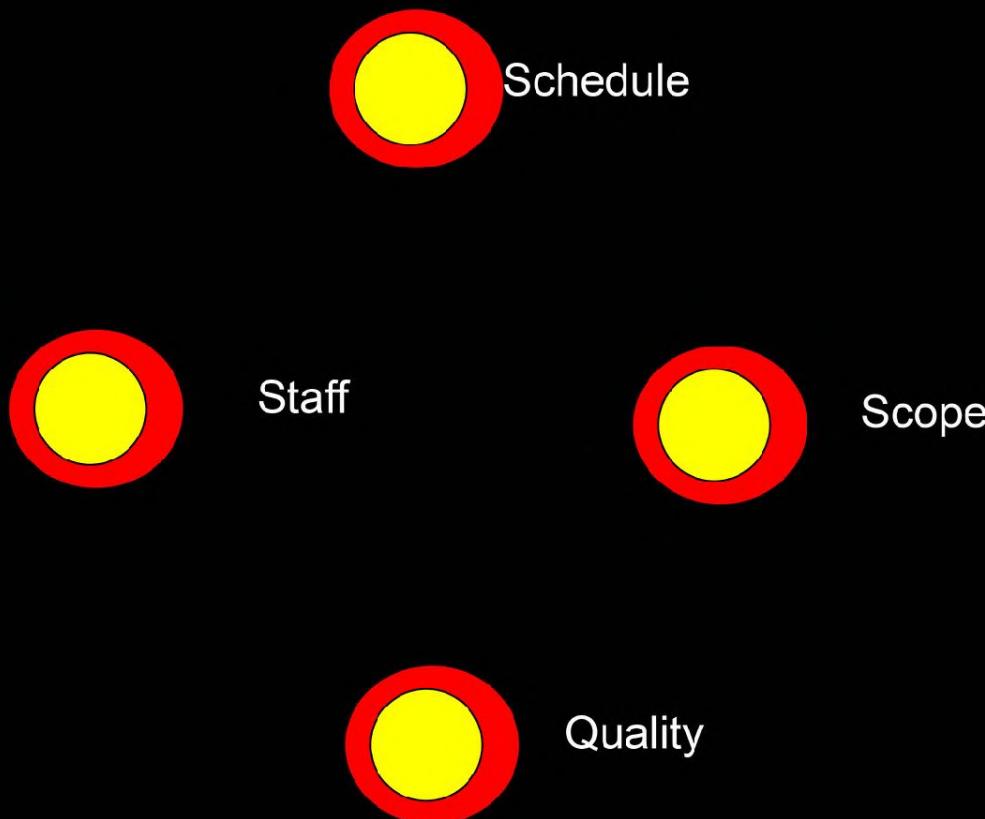
But when we have data...



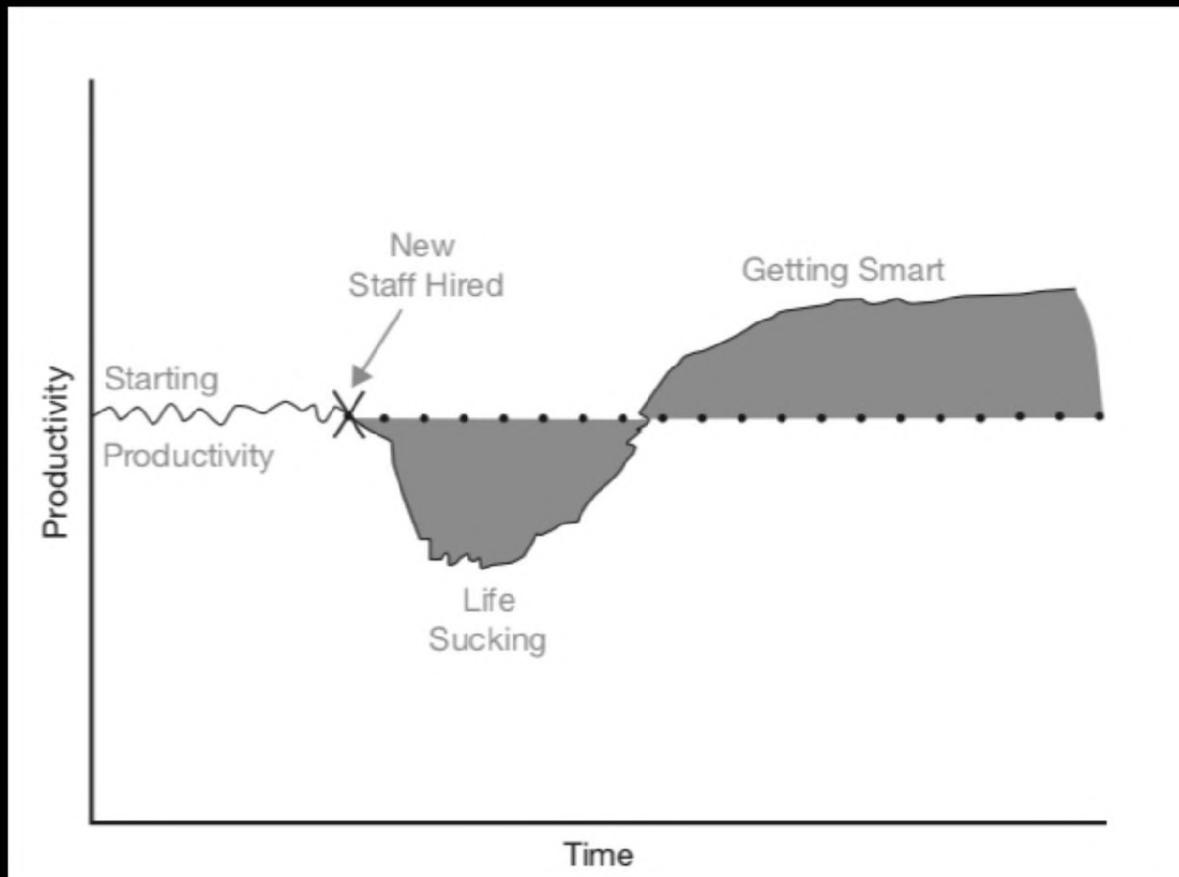
Managers can....

manage.

The control knobs of project mgt.

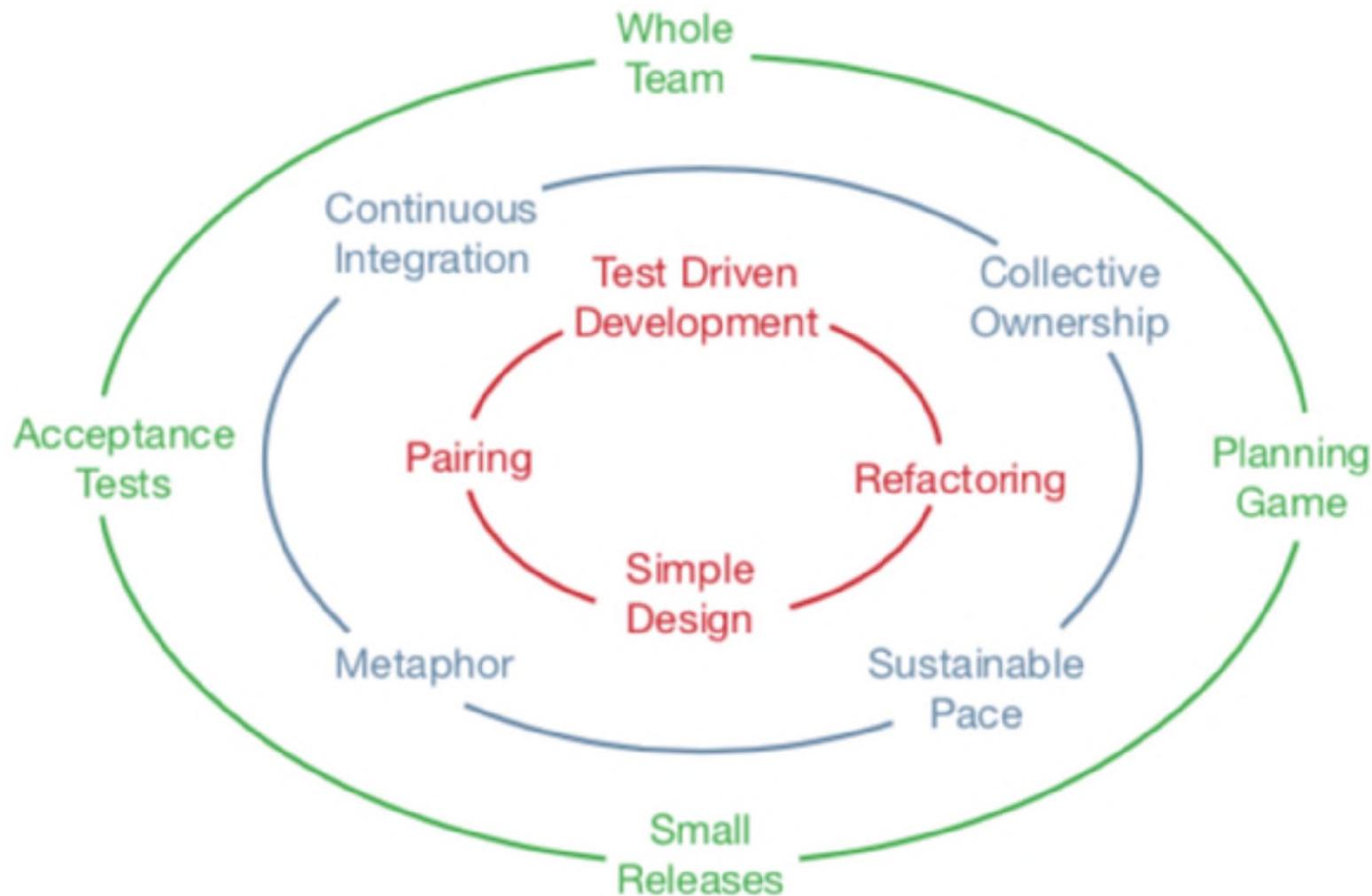


Adding Staff



THE CIRCLE OF LIFE.



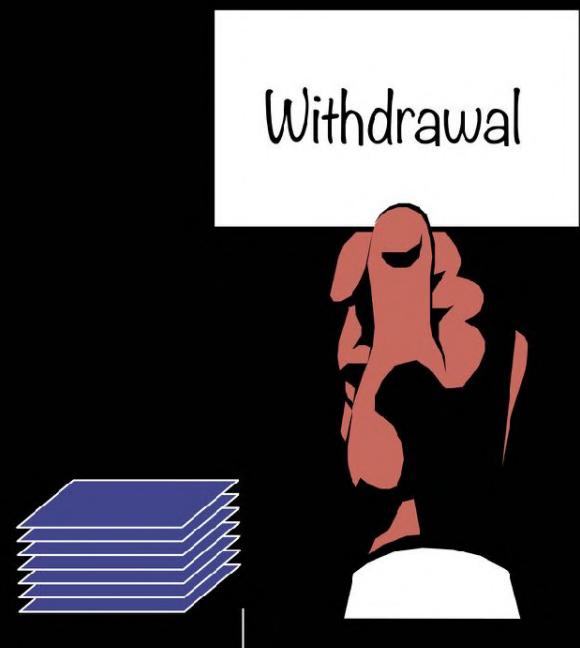


Planning Game

User Stories
Release Planning
Iteration Planning

Stories

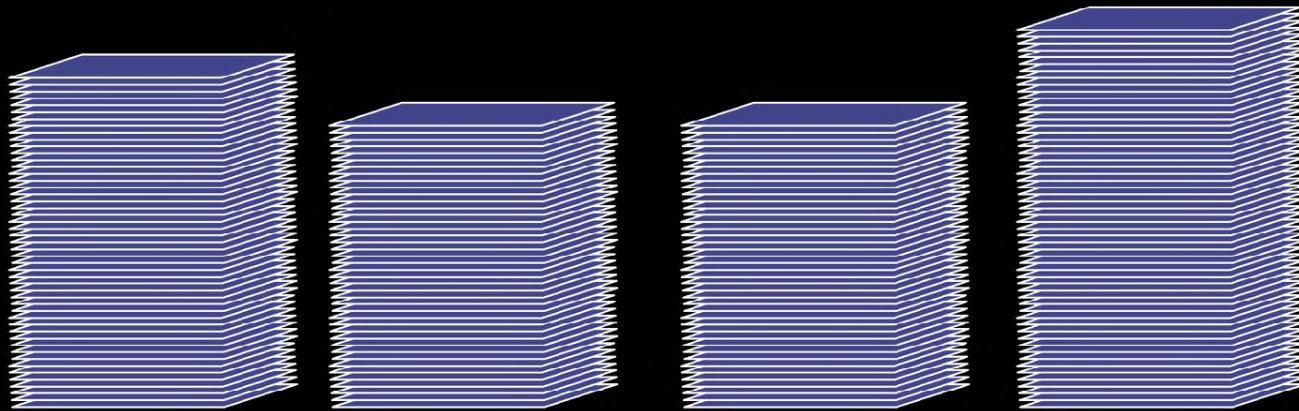
- Name of a feature, or short description
- Brief specifications of customer requirements
- Is a token for a conversation
- Owned by the customer
- INVEST: A good user story criteria
 - Independent
 - Negotiable
 - Valuable
 - Estimable
 - Small
 - Testable



Story Estimation

- Team assigns a estimate or price to each story
- Estimates are in dimensionless but proportional units
 - A 2 takes twice as much effort as a 1
- Planning Poker and Flying Fingers.
- Customer uses price to juggle priorities and acquire resources
- Unknowns can be eliminated through use of *spikes*, a research story

User Stories Define the Project Backlog





**“In preparing for battle I have
always found that plans are
useless, but planning is
indispensable. ”**

Dwight David Eisenhower

The Purpose of Planning

It is more important to work on the most valuable things first than to predict the future.



- Even the best plans must be continually refined
- Requires continuous and meaningful feedback

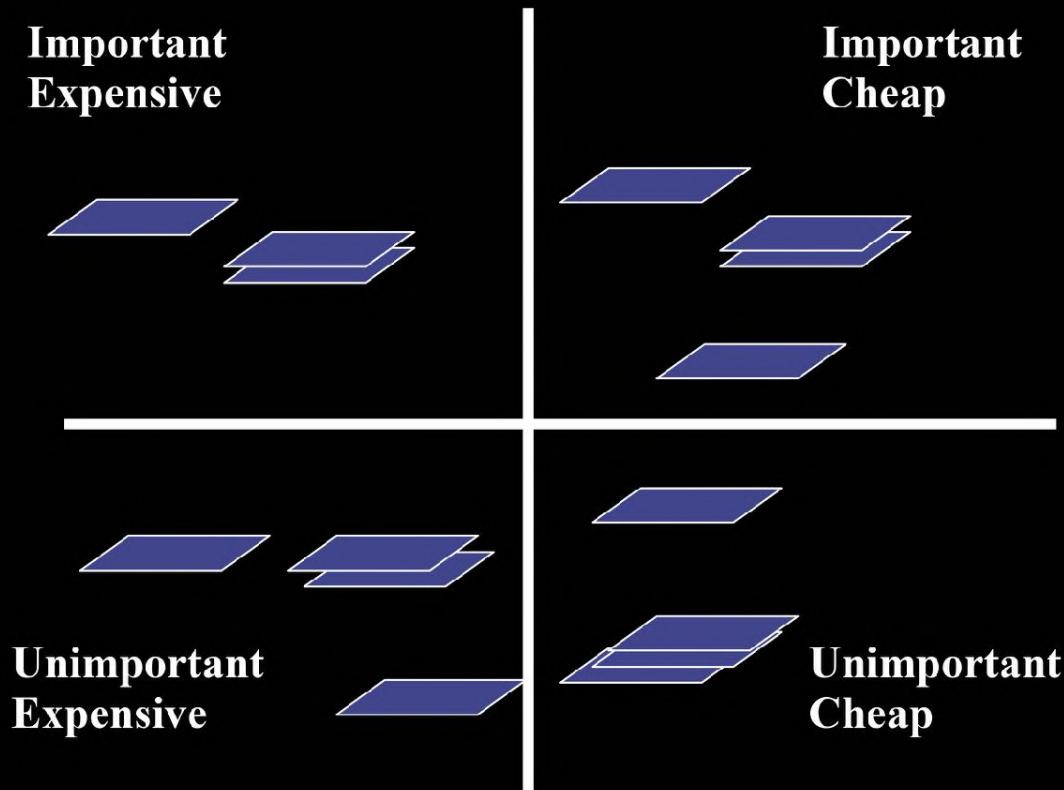
- Attendees
 - Whole team

- Input
 - Product Vision
 - Initial Story List (verbal)
 - Critical dates

- Output
 - Deck of initial stories
 - Architectural vision
 - Initial estimates
 - Velocity guesstimate

Release Planning

The 4-way Decision.



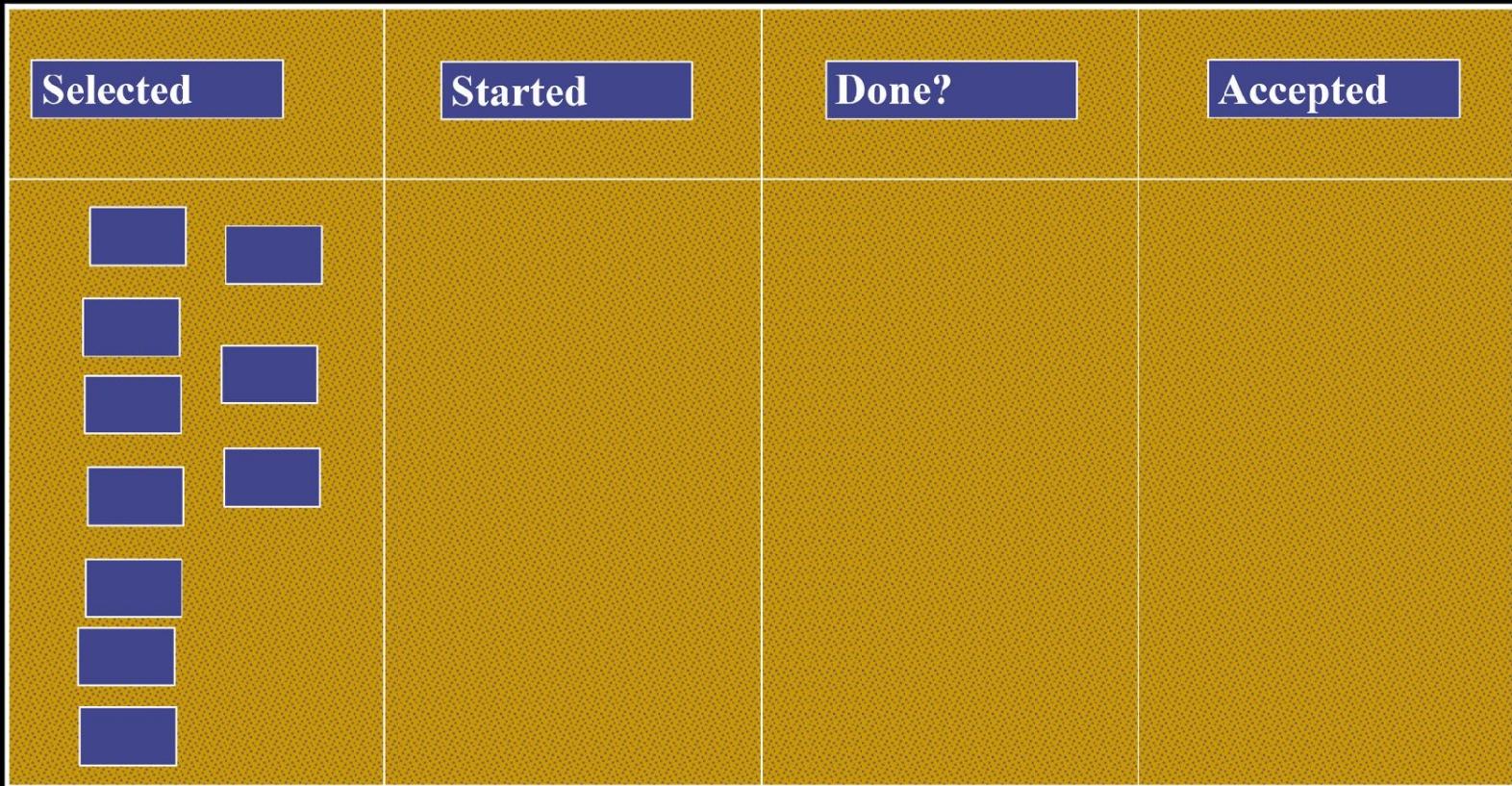
The Iteration

- ➊ Getting Stories Done.
- ➋ The midpoint.
- ➌ The Demo.

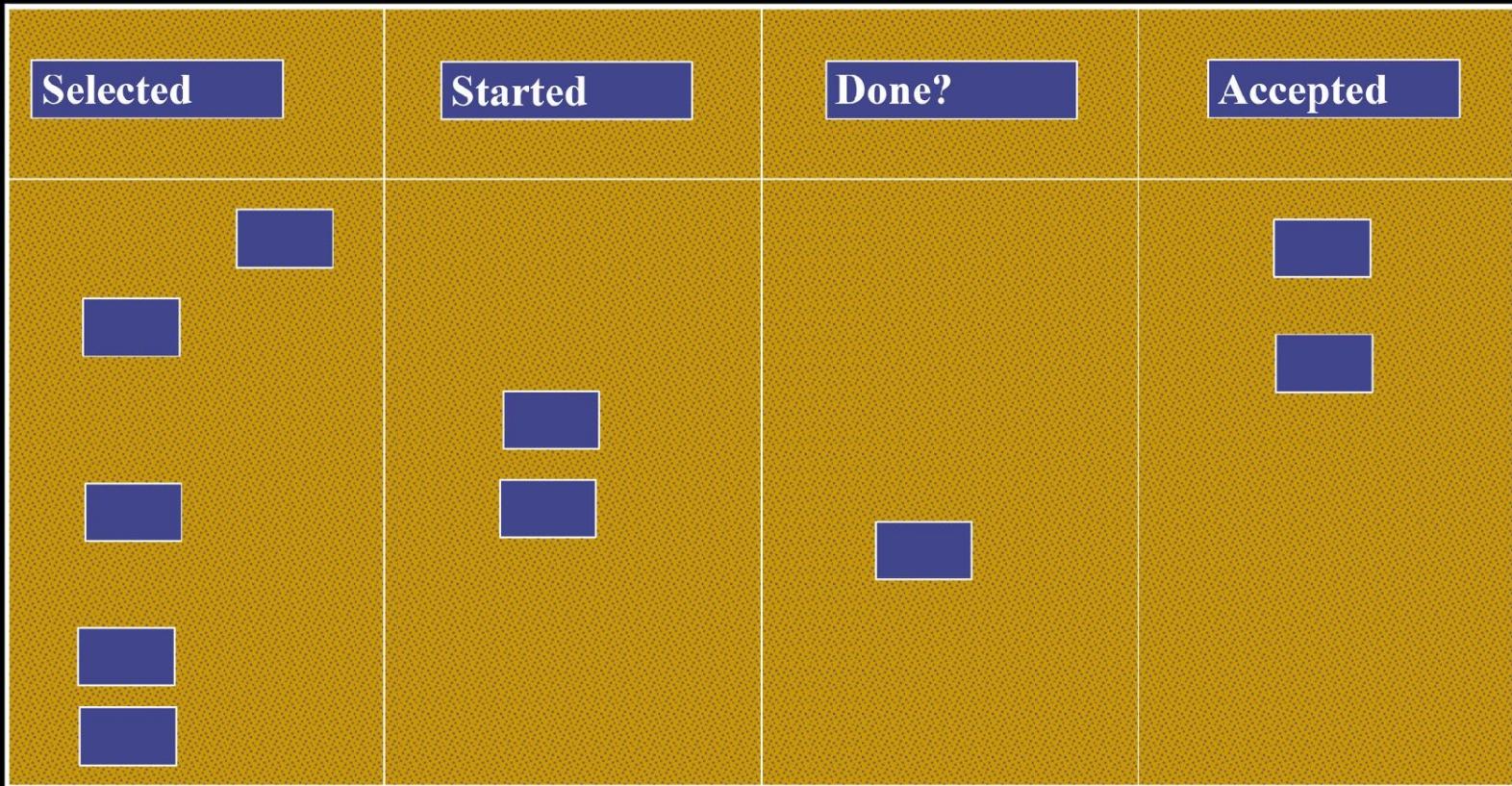
- ➍ Yesterday's weather.

- ➎ Rinse and repeat.

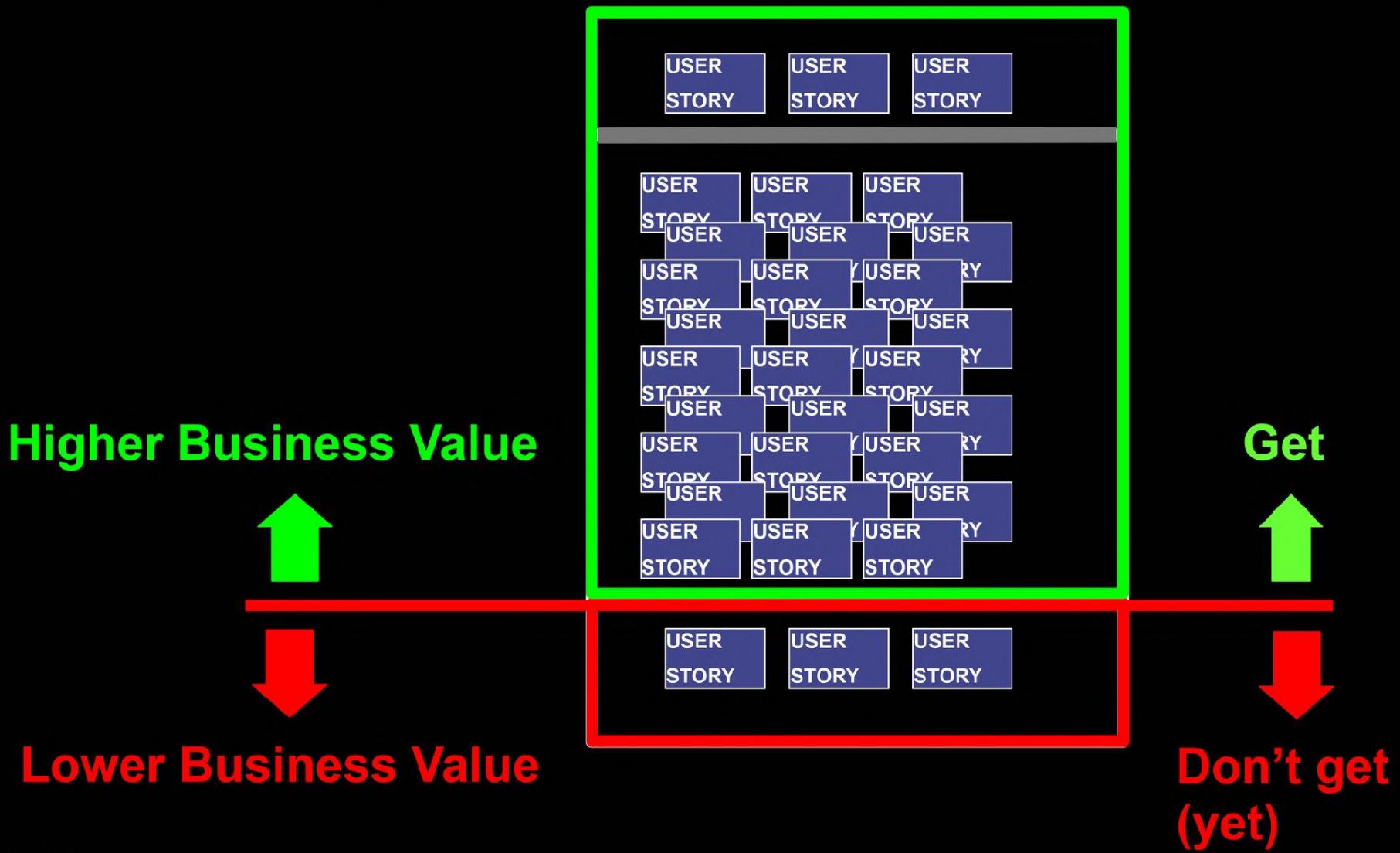
Iteration Tracking Bulletin Board



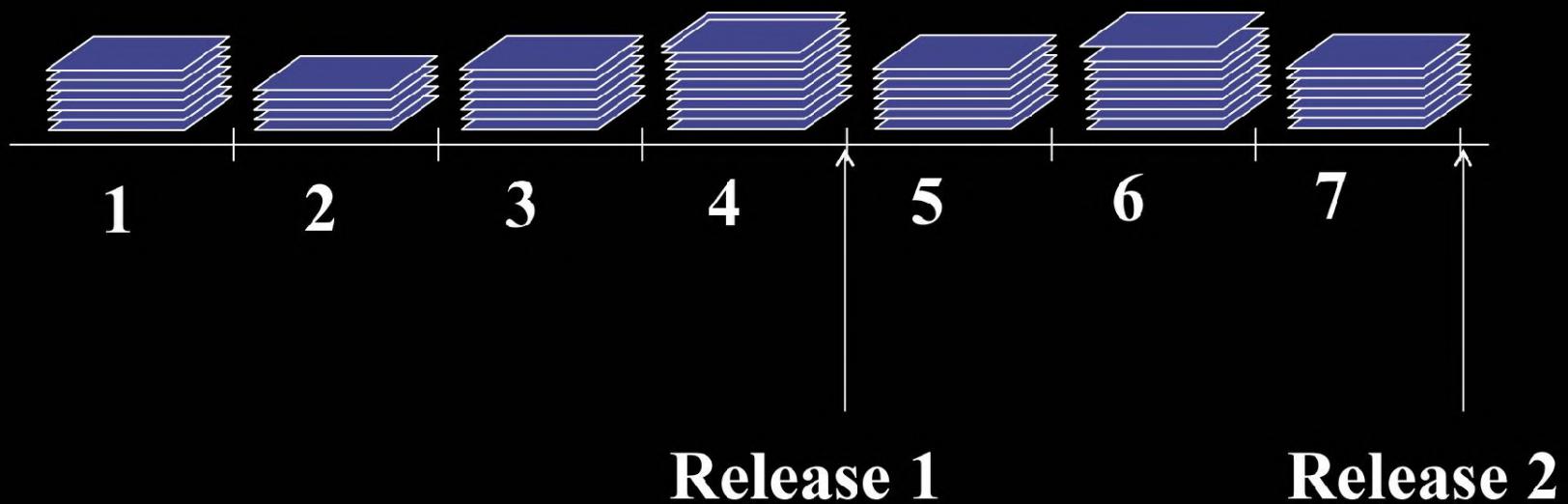
Iteration Tracking Bulletin Board



Managing Scope



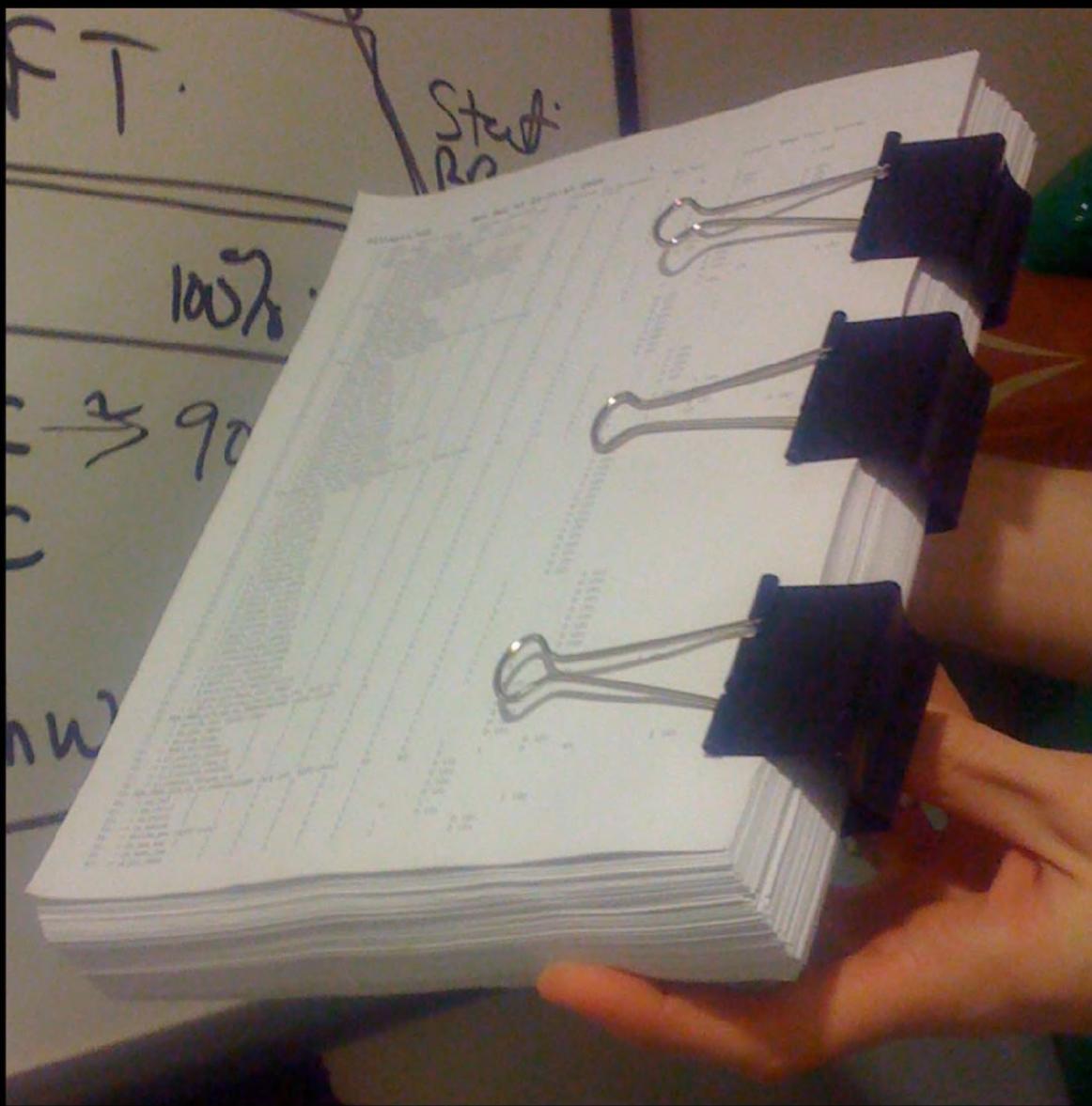
Release Plan



WRITING AUTOMATED ACCEPTANCE TESTS

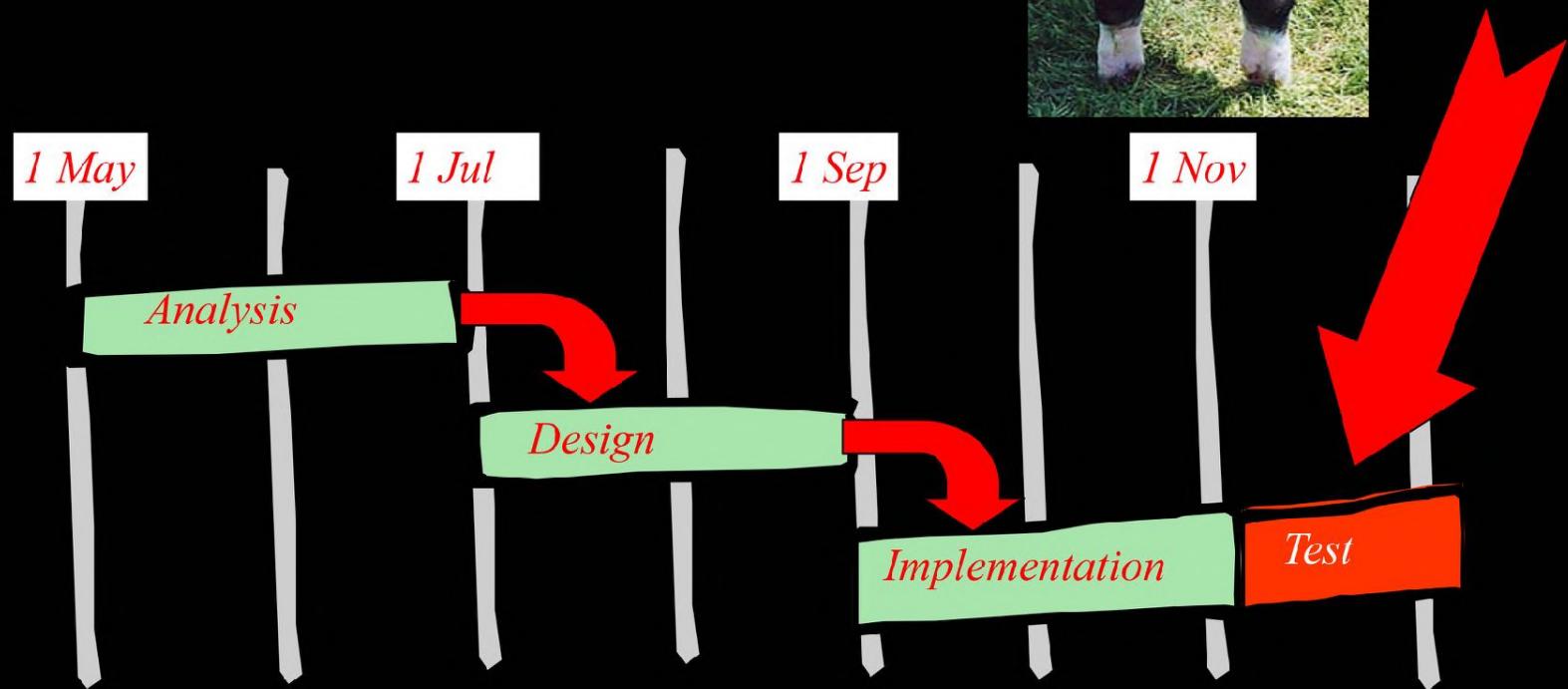
Copyright © 1998-2013 by Robert C. Martin
All Rights Reserved

The Test Plan



The Traditional schedule for Quality.

- Waiting at the sphincter muscle.



When Quality comes at the end.

- It is under the most pressure.
 - It has the least flexibility.
 - It is a high stress tedious job.
 - It is error prone.
-
- Quality cannot be tested in.

The Agile View:

Quality is a *specification* role...

...not a verification role!

A feature is not specified...

Until it's acceptance test is written.

Who Writes Acceptance Tests?

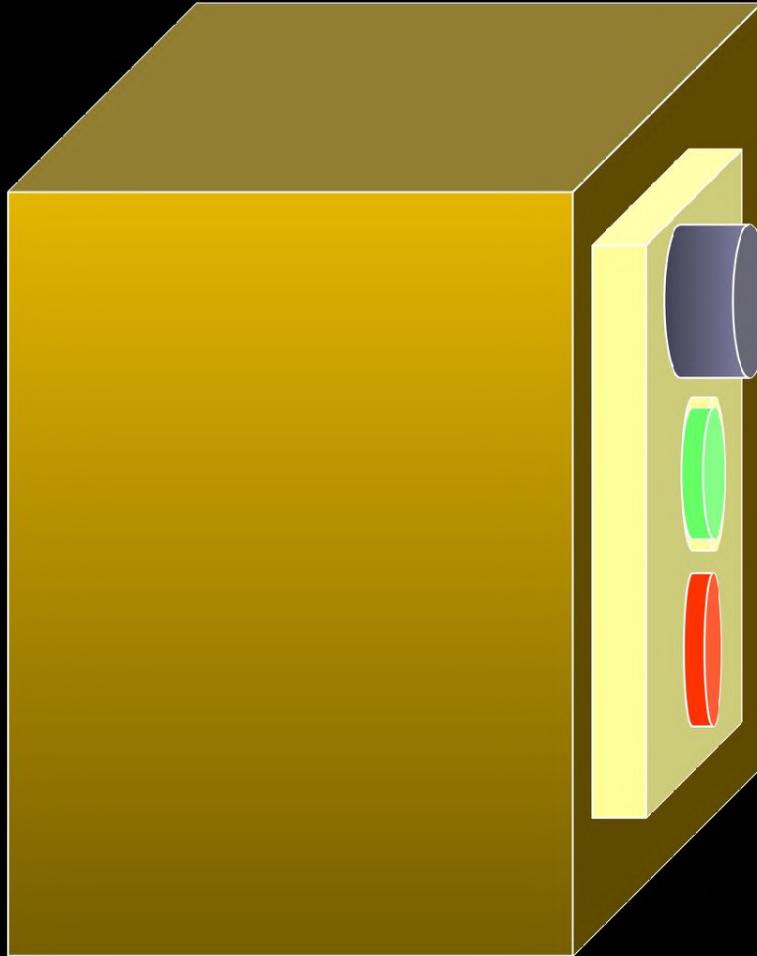
- Business Analysts
 - Happy path.
- QA Test Writers, and Testers
 - Corner and boundary cases.
 - Try to break it.

Acceptance Tests Are...

 Automated.

- written in a very high level language.
- executed frequently.
- written by the stakeholders.

Imagine you have a button to press that would tell you if the system worked



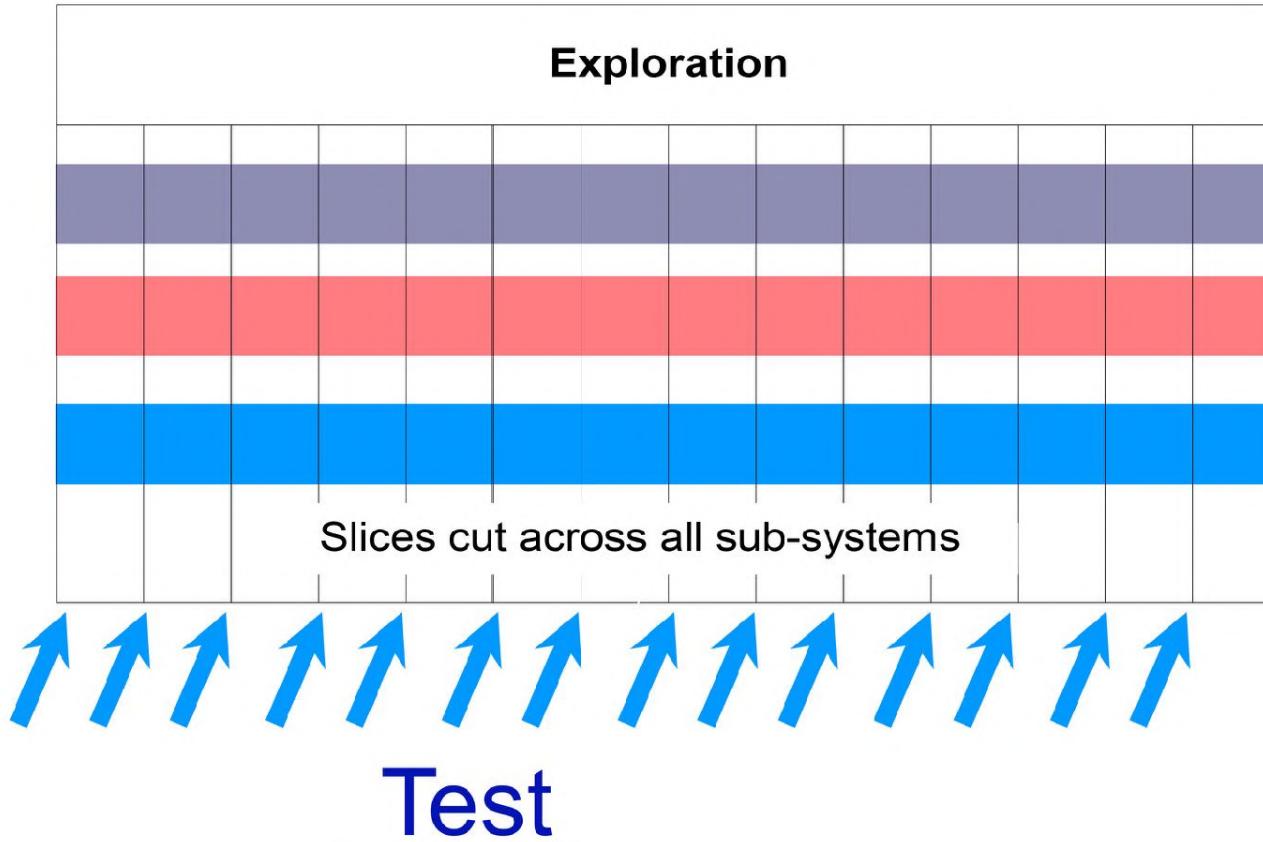
How often
would you
press it?

It becomes a conditioned response.

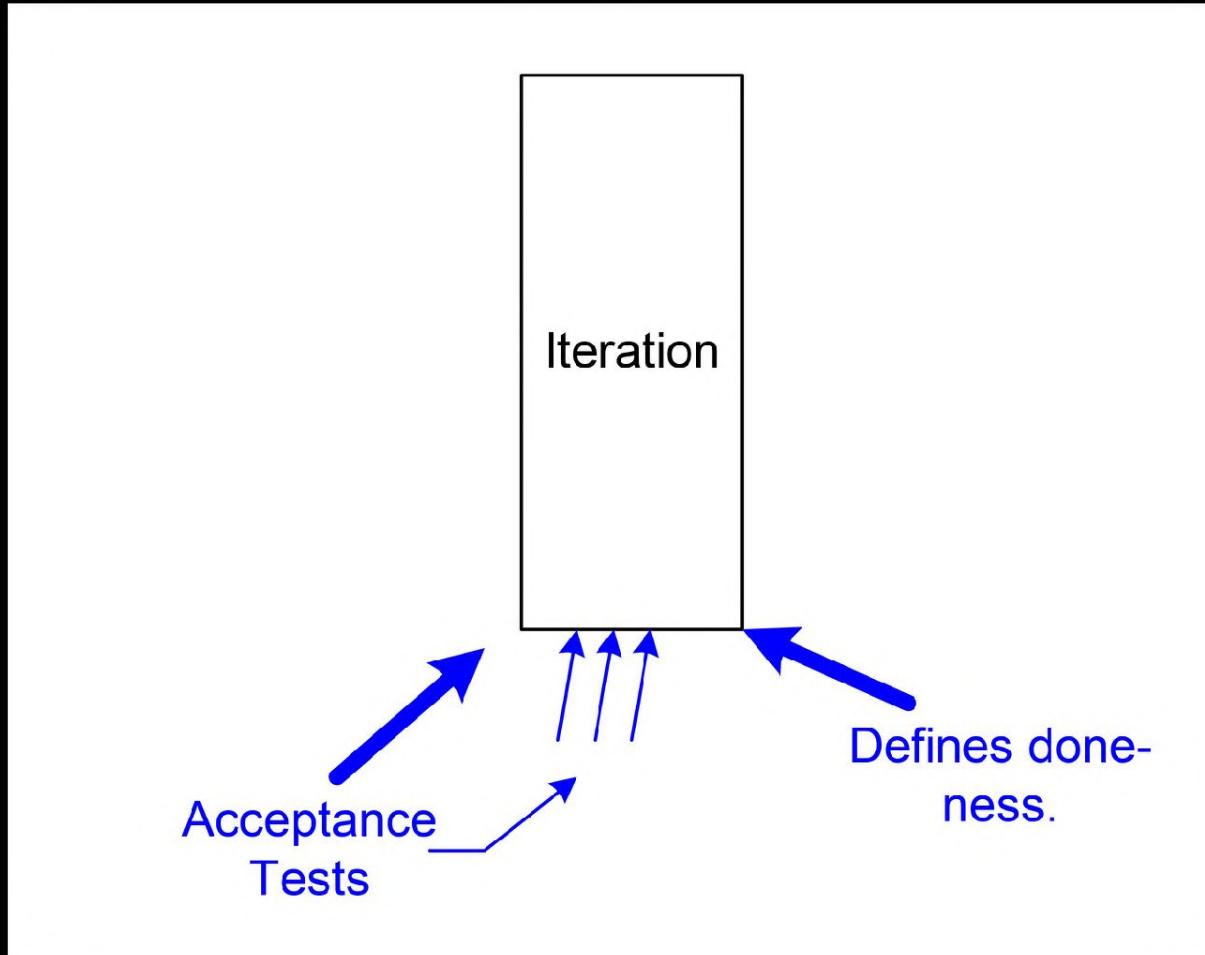


When is the best time to write
Automated Acceptance Tests?

At the start of each iteration.

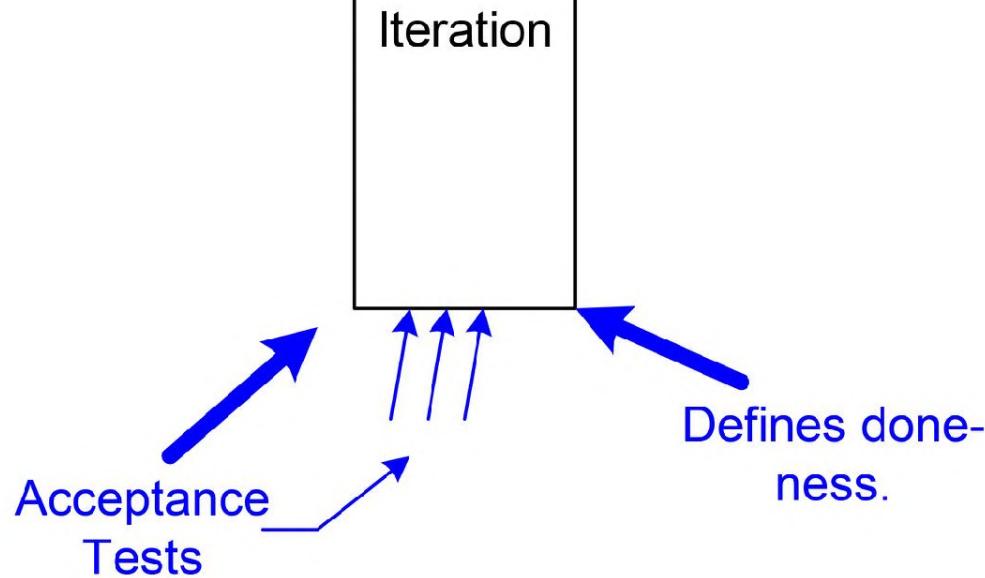


Tests specify each iteration.



Tests specify each iteration.

Tests are really specs!

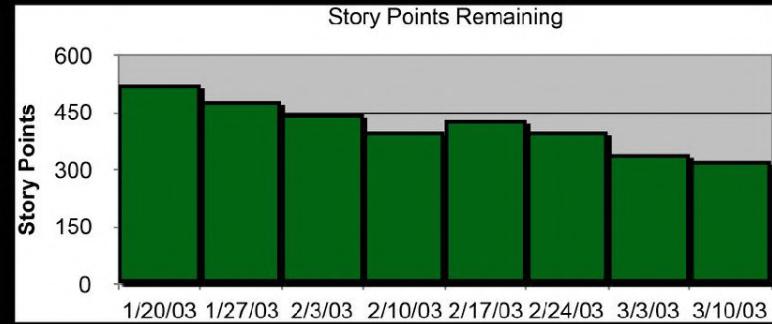
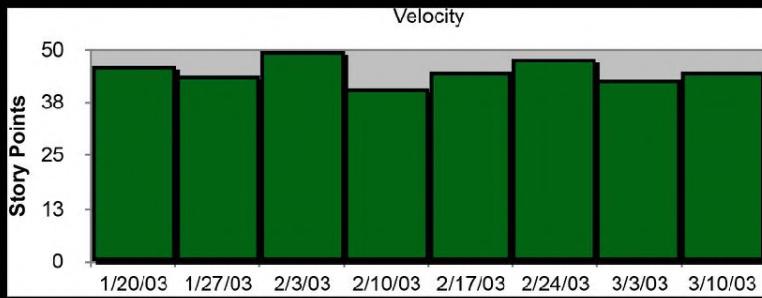


How do you know a feature is done?

A feature is not done...

Until all it's acceptance tests pass.

Knowing when a feature is done allows...



These features have passed their Automated Acceptance Tests.

What do they look like?

A Simple Example

ADD VALID SALESPEOPLE

The salesperson record consists of just three fields: *id*, *firstName*, and *lastName*. Each of these fields are non-blank strings with no other constraints. An attempt to add a salesperson with an invalid field should fail.

For example: If we try to add these nine salespeople to the system, the first three should succeed, and the last six should fail because they have blank or null fields.

Add Salespeople				
<i>id</i>	<i>firstName</i>	<i>lastName</i>	<i>valid?</i>	
U999999	Robert	Martin	true	
U840818	Mitzi	Schofield	true	
7734	Jason	Billings	true	

Add Salespeople				
<i>id</i>	<i>firstName</i>	<i>lastName</i>	<i>valid?</i>	
blank	Bill	Jacobs	false	"blank id"
1111	blank	Johnson	false	"blank first name"
2222	Joe	blank	false	"blank last name"
null	Bob	Fitzwater	false	"null id"
3333	null	Fravitz	false	"null first name"
4444	Pete	null	false	"null last name"

If we fetch the salespeople from the database, only the first three from above should be present.

Get Salespeople		
<i>id</i>	<i>firstName</i>	<i>lastName</i>
U999999	Robert	Martin
7734	Jason	Billings
U840818	Mitzi	Schofield

The result.

Add Salespeople			
id	firstName	lastName	valid?
U999999	Robert	Martin	true
U840818	Mitzi	Schofield	true
7734	Jason	Billings	true

Add Salespeople				
id	firstName	lastName	valid?	
blank	Bill	Jacobs	false expected true actual	"blank id"
1111	blank	Johnson	false expected true actual	"blank first name"
2222	Joe	blank	false expected true actual	"blank last name"
null	Bob	Fitzwater	false expected true actual	"null id"
3333	null	Fravitz	false expected true actual	"null first name"
4444	Pete	null	false expected true actual	"null last name"

If we fetch the salespeople from the database, only the first three from above should be present.

Get Salespeople		
id	firstName	lastName
U999999	Robert	Martin
7734	Jason	Billings
U840818	Mitzi	Schofield
<i>null surplus</i>	Bob	Fitzwater
<i>3333 surplus</i>	null	Fravitz
<i>2222 surplus</i>	Joe	blank
<i>1111 surplus</i>	blank	Johnson
<i>4444 surplus</i>	Pete	null
<i>blank surplus</i>	Bill	Jacobs

Test Driven Development

The Three Laws of TDD

- 1 - You are not allowed to write any production code unless it is to make a failing unit test pass.
- 2 - You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures.
- 3 - You are not allowed to write any more production code than is sufficient to pass the one failing unit test.

