
# *ESUG 2009: [Project planning].*

Tim Mackinnon

(www.iterex.co.uk)

# *Smalltalk and Planning*

Smalltalk - synonymous with the invention and refinement of many techniques and technologies from GUI's, unit testing, refactoring, vm's, and project planning.

**Planning?** NOT glamorous, but the secret to successful projects. As developers we **HATE it**!

*This session will review, clarify and myth-bust some of the common techniques. More importantly, I will present what new ideas have surfaced around successful teams and the way they plan. Planning doesn't have to be tedious and boring, it can be rapid and successful.*

# *Tim Mackinnon - Who are you?*

- 2006 – Iterex (Iterative Excellence)
  - Tailored Consulting/Coaching for Agile projects
  - iPhone Development (ReDo, WonderWorld)
- 2003 – ThoughtWorks
  - Agile enablement coaching
  - Papers on release estimation techniques
- 1999 – Connextra
  - Formed one of the first Agile teams in the UK
  - Invented “Mock Objects” test technique
  - Pioneered Iteration Retrospectives, XtC
- 1996 – OTI
  - Developer on UniBowser/VA-Modeler (early agile practices)
  - Uni-Browser framework, early UI predecessor to Eclipse



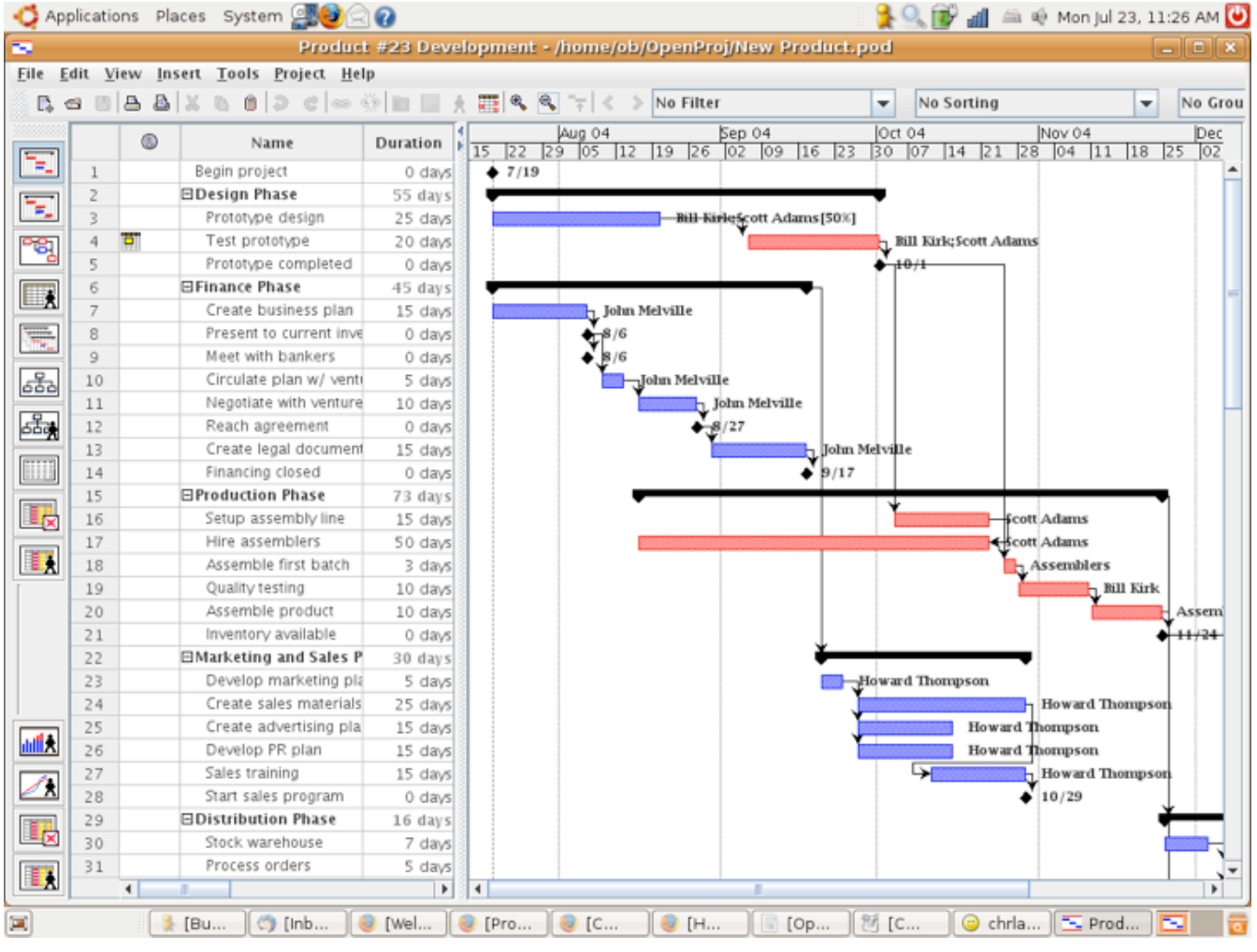
I use many languages and environments, but Smalltalk is still my favourite and most productive environment.

# Thank you Smalltalkers...




Its unusual to start with Thank you, but most of the ideas in this presentation come from Smalltalk and Smalltalkers.

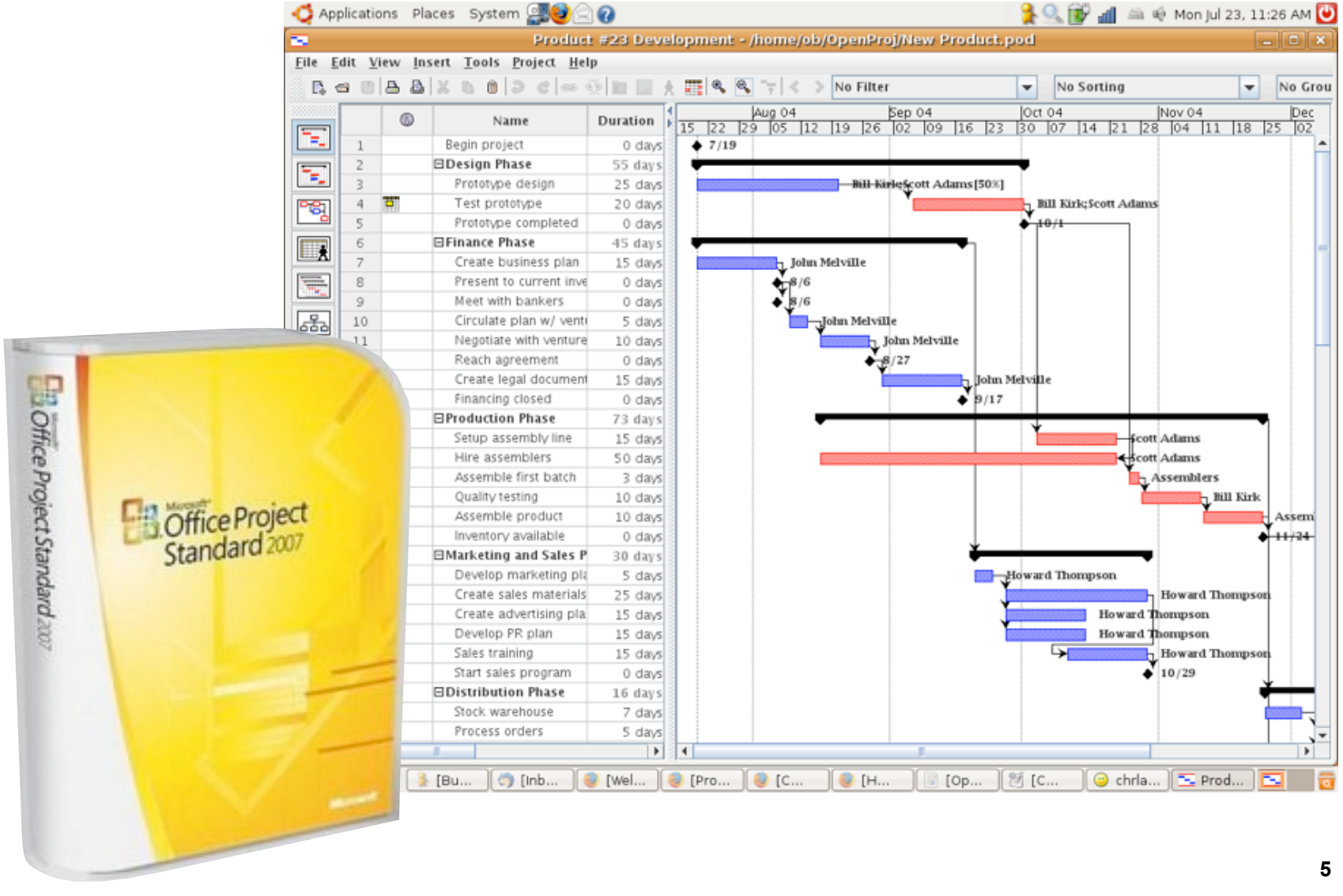
# Microsoft Project Tutorial...






There are other ways of planning instead of using MS-Project

# Microsoft Project Tutorial...





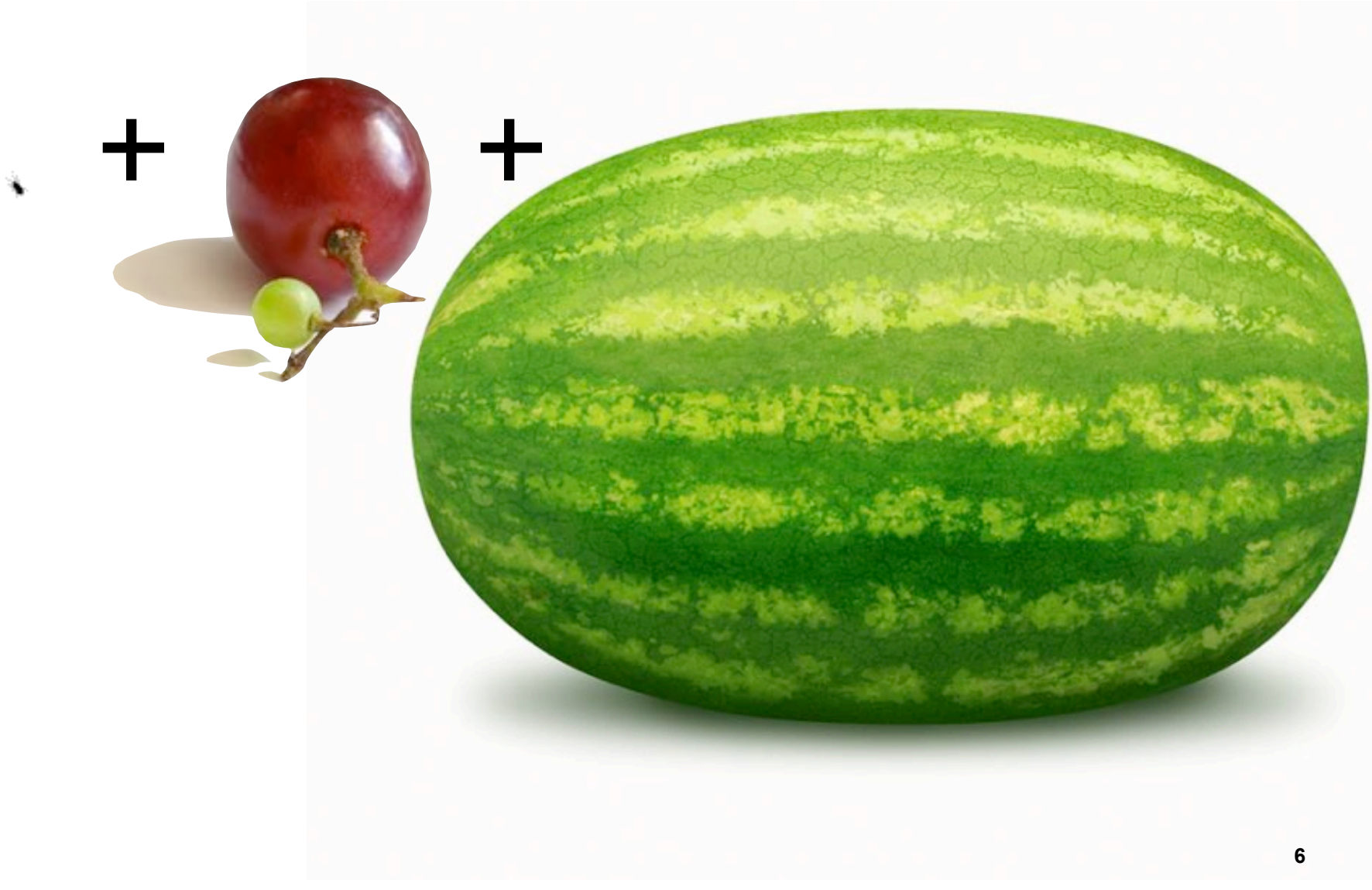

There are other ways of planning instead of using MS-Project

# Microsoft Proje




There are other ways of planning instead of using MS-Project

# *Apples+Oranges = ?*





Described the “Dashboards” planning experience – hours were added to days, were added to weeks and a final number came out of the exercise which was deemed to be the “end date”

# *Apples+Oranges = ?*




Described the “Dashboards” planning experience – hours were added to days, were added to weeks and a final number came out of the exercise which was deemed to be the “end date”

# *IEEE article by Tom DeMarco (Jul/Aug 2009)*

- I'm gradually coming to the conclusion that software engineering is an idea whose time has come and gone.

- Software development is and always will be **somewhat experimental**. The actual software construction isn't necessarily experimental, but its conception is. And this is where our focus ought to be. It's where our focus always ought to have been.

# The Agile Approach...



The wonder of agile... Scrum leprechauns and XP wizardry

# The Agile Approach...


Image attribution: Renewtek



The wonder of agile... Scrum leprechauns and XP wizardry

# The Risk of Flacid Scrum!



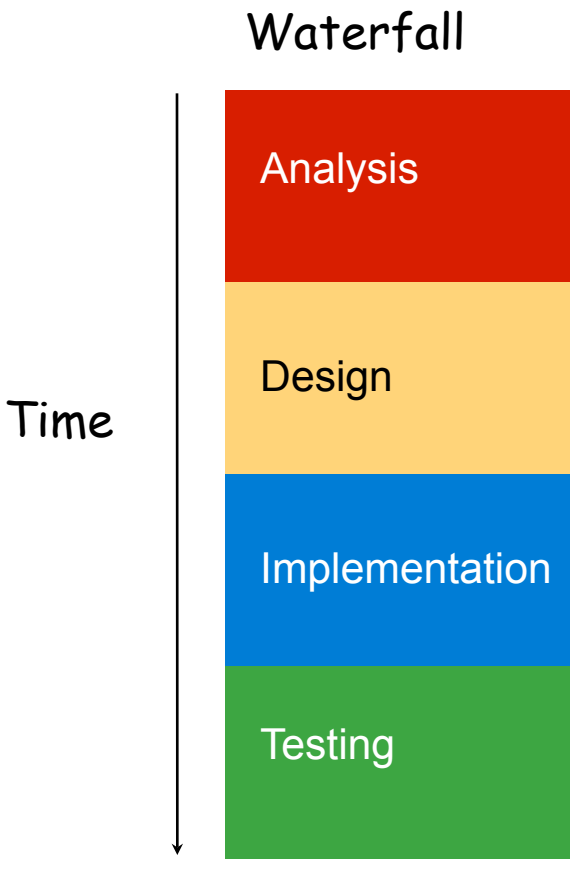
Many projects jump on the Scrum bandwagon - it seems easy, but without good engineering discipline you can watch a burndown chart just extend out to infinity

# *XP: Turning Extreme into Excellence!*

Take common sense practises to extreme levels - “turning the dials to 11!”

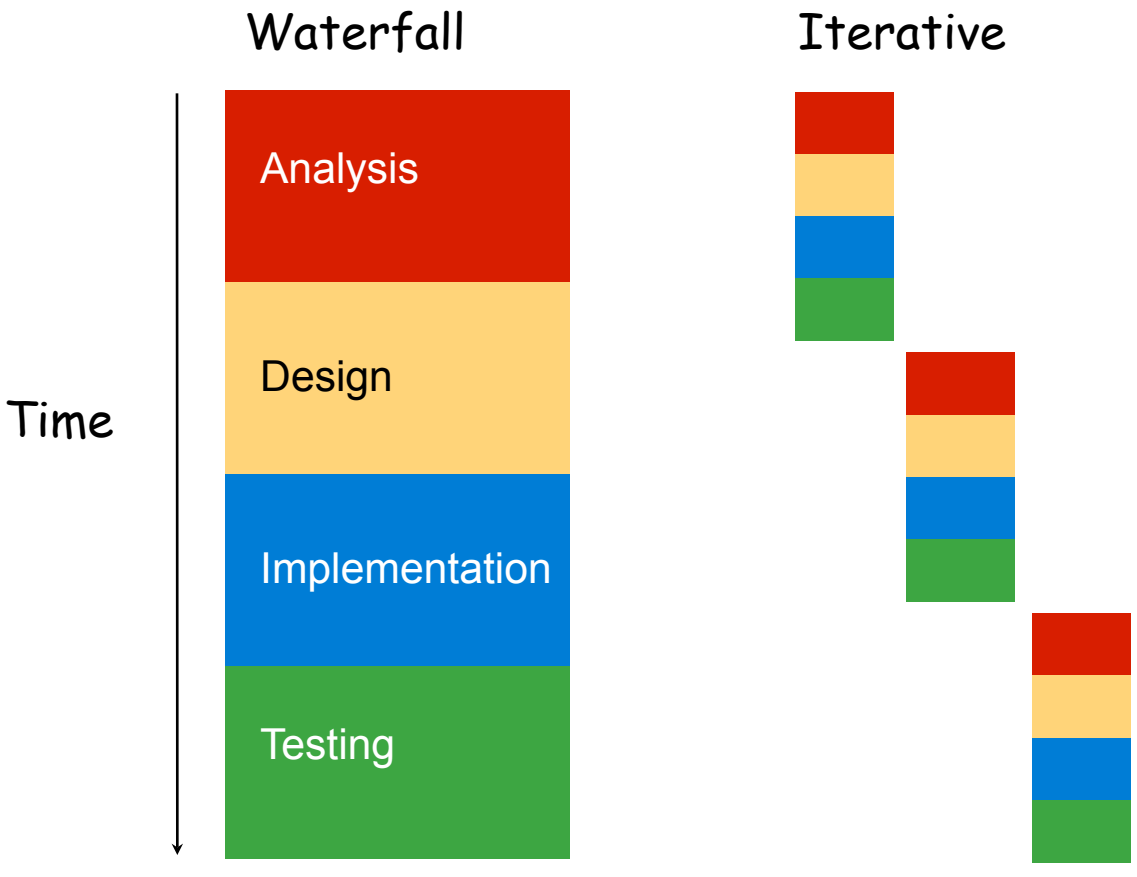
- **If code reviews are good**, *review code all the time* (Pair Programming)
- **If testing is good**, *everybody will test all the time* (Unit Testing)
- **If design is good**, *make it part of everyone’s daily business* (Refactoring)
- **If simplicity is good**, *always leave the system with the simplest design that supports it’s current functionality.* (The simplest thing that could possibly work)

# Comparison of Approaches



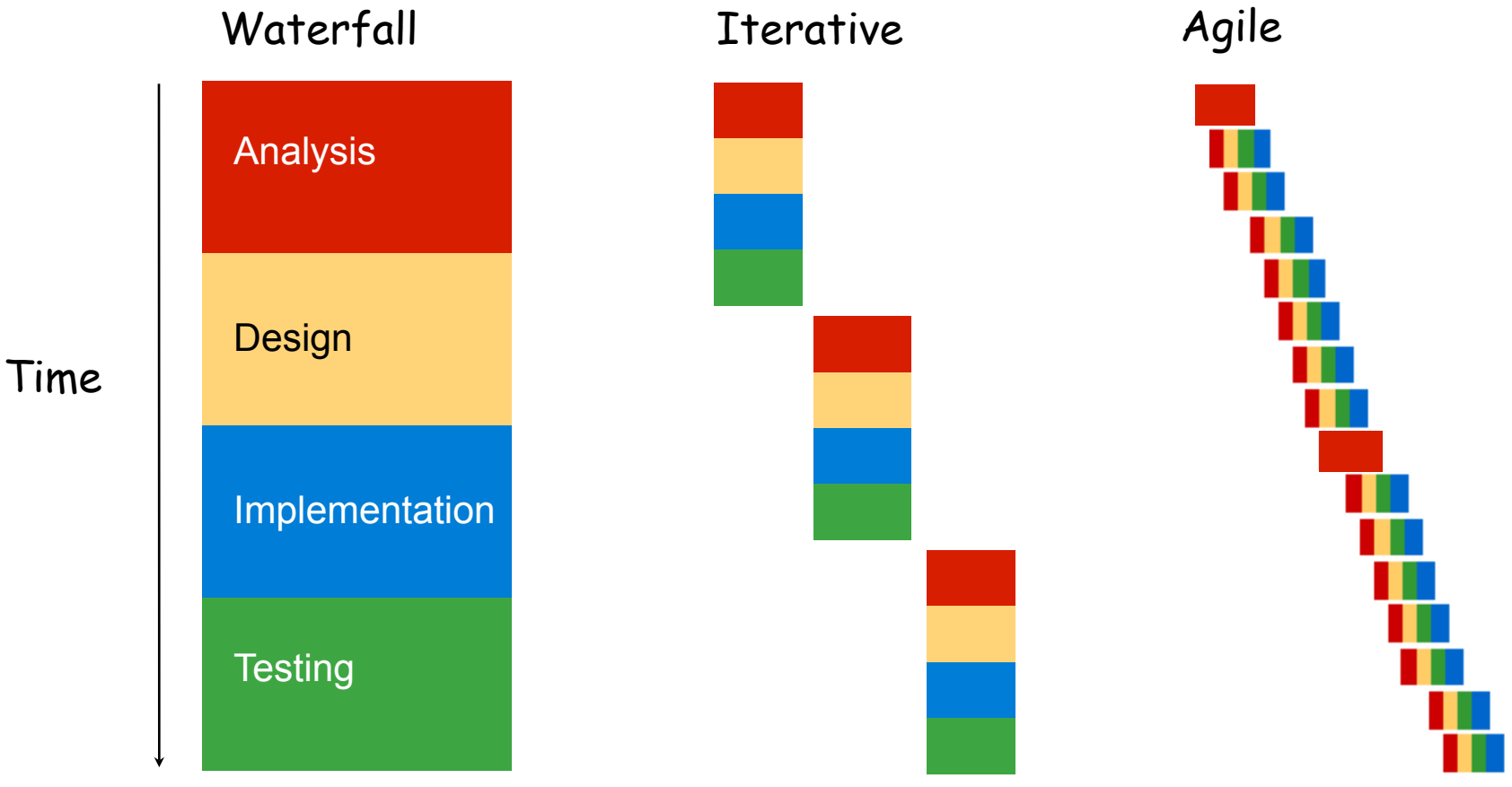

Explained differences between Waterfall, Iterative, Agile. Hilited the added Red blocks in Agile for more corporate environments.

# Comparison of Approaches




Explained differences between Waterfall, Iterative, Agile. Hilited the added Red blocks in Agile for more corporate environments.

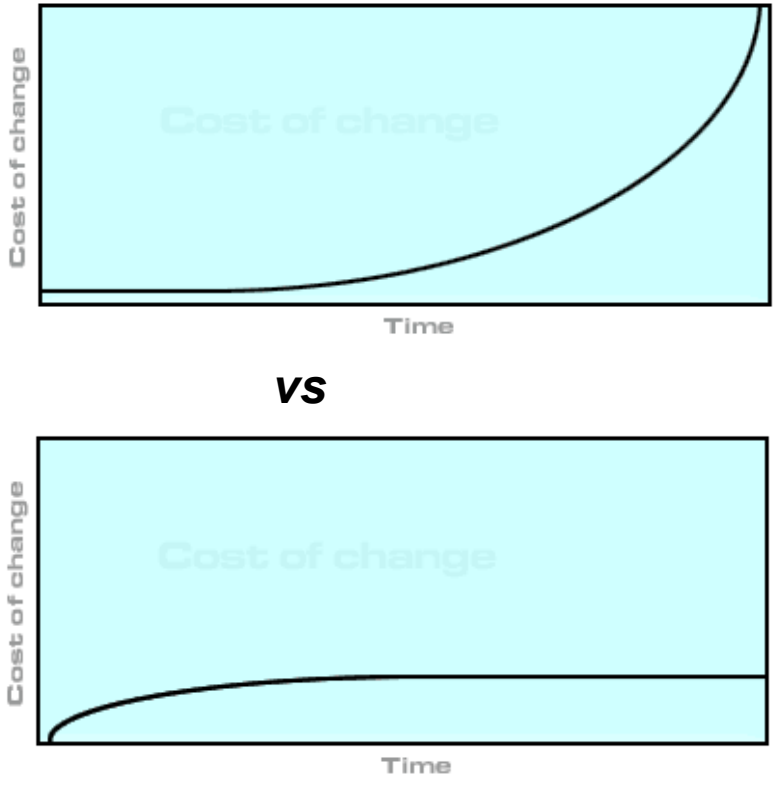
# Comparison of Approaches



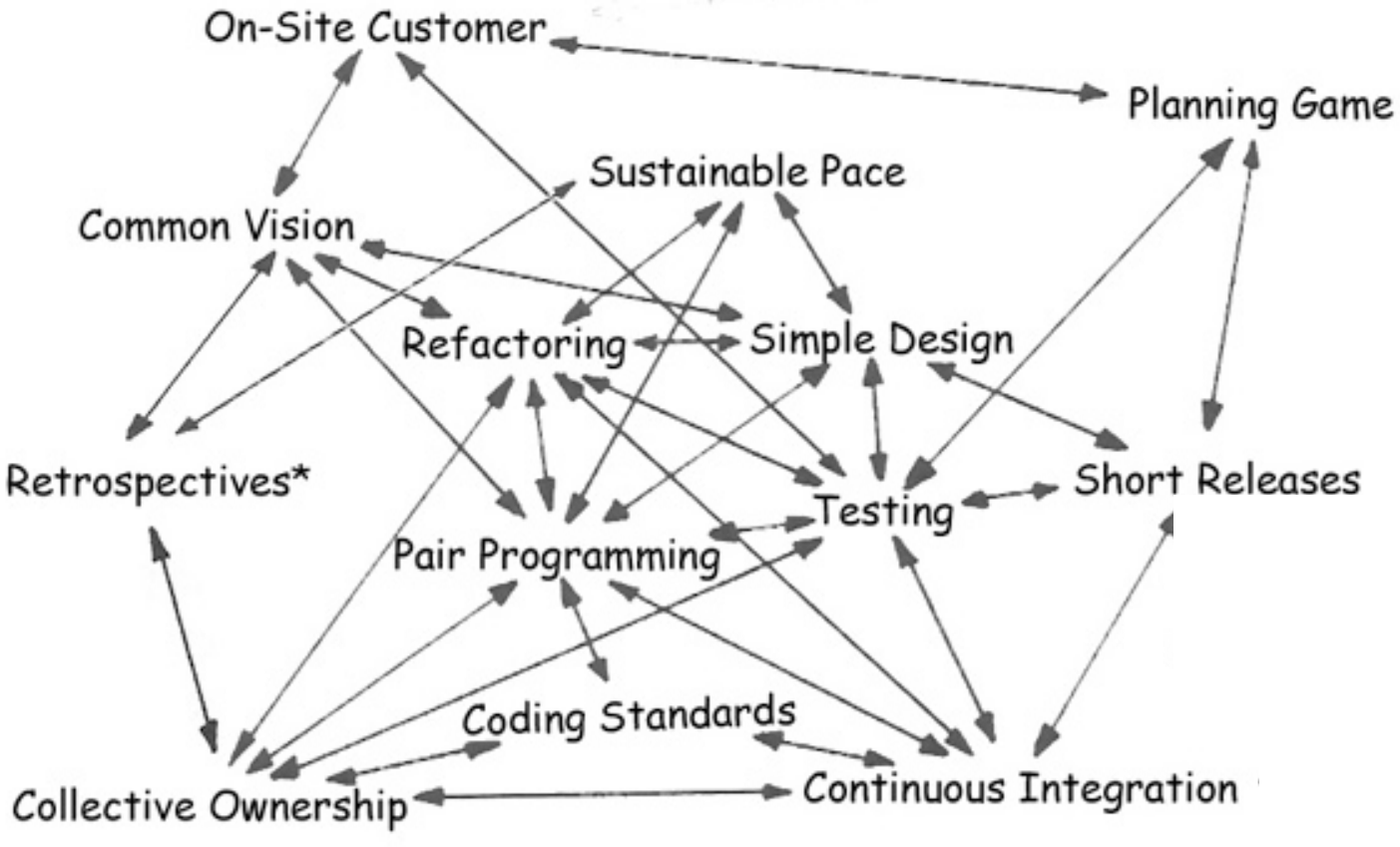

Explained differences between Waterfall, Iterative, Agile. Hilited the added Red blocks in Agile for more corporate environments.

# The Importance of maintaining Simple Design

- Do the simplest thing that can possibly work
- Always have the simplest/ smallest system
- Don't predict future requirements, they slow you down
- Remove redundancy
- Refactor for simplicity
- Remember YAGNI (you aren't going to need it)

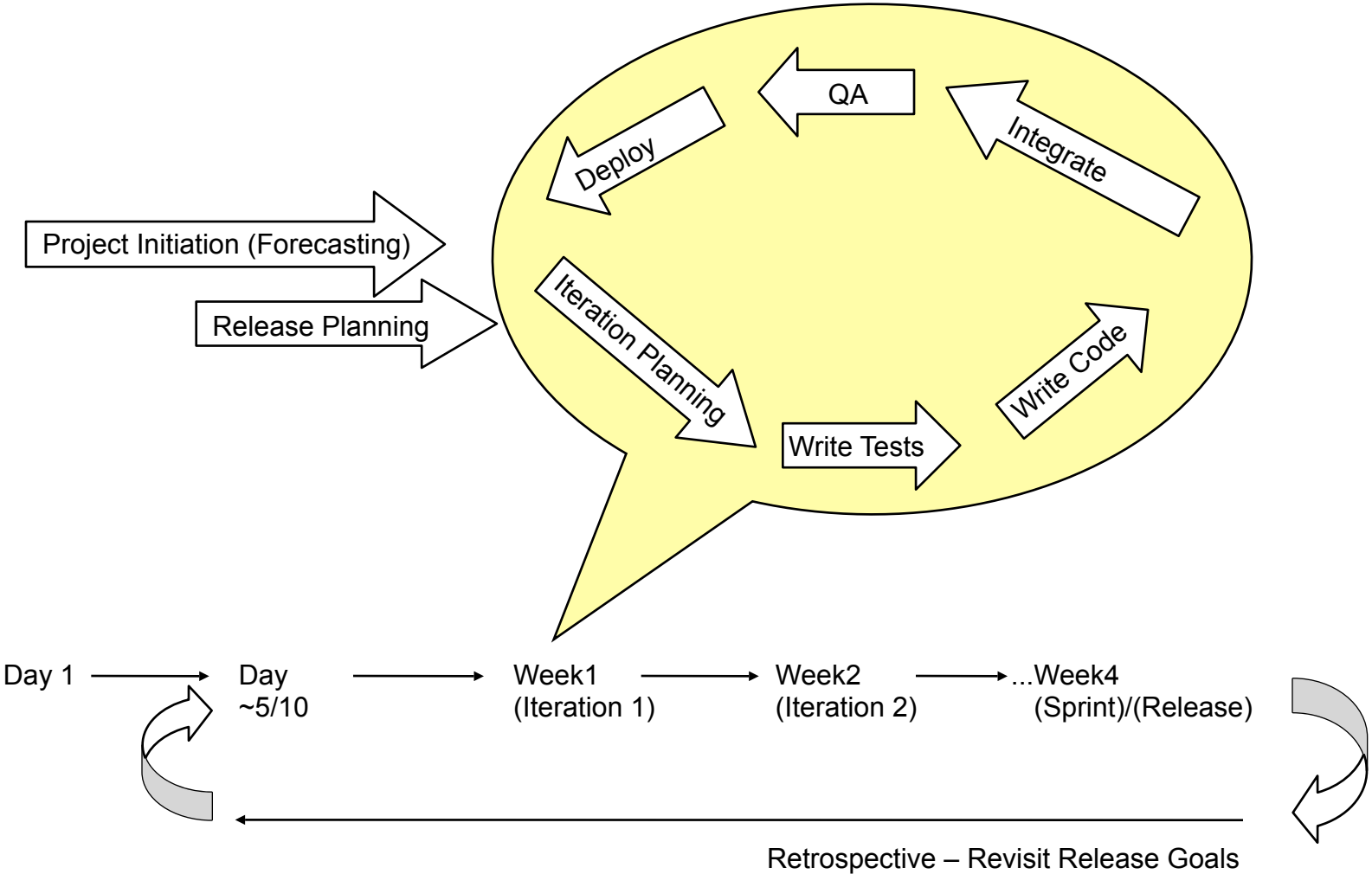


# Reinforcing Practices Flatten the Cost Curve



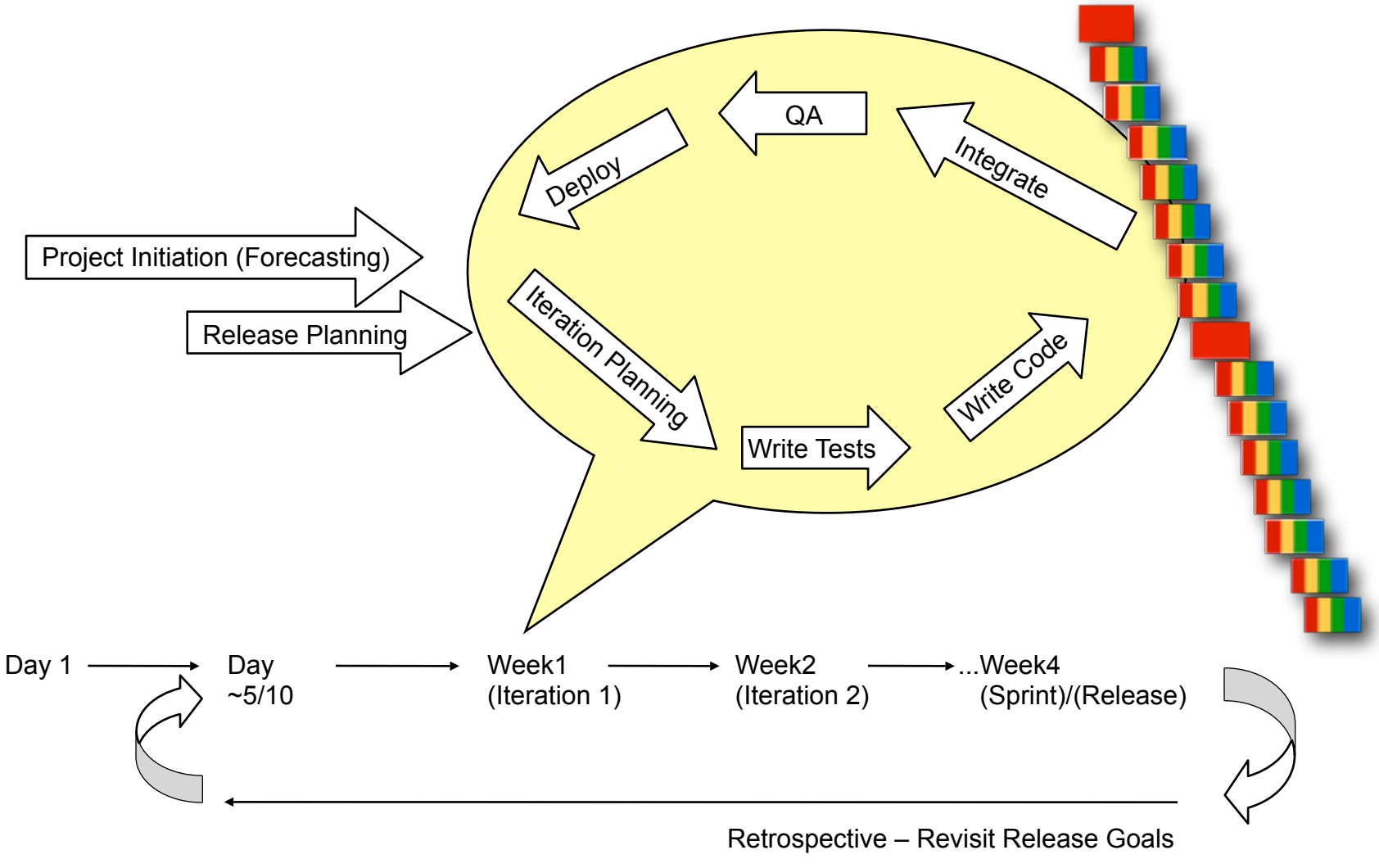

Described the tilting platform of reinforcing practices – trying to keep a balanced platform you can build on

# Planning for an Agile Lifecycle…




Clicked to show incremental iterations stairs – a reminder that these cycles are the little slices discussed earlier

# Planning for an Agile Lifecycle…




Clicked to show incremental iterations stairs – a reminder that these cycles are the little slices discussed earlier

# So how do you plan then?

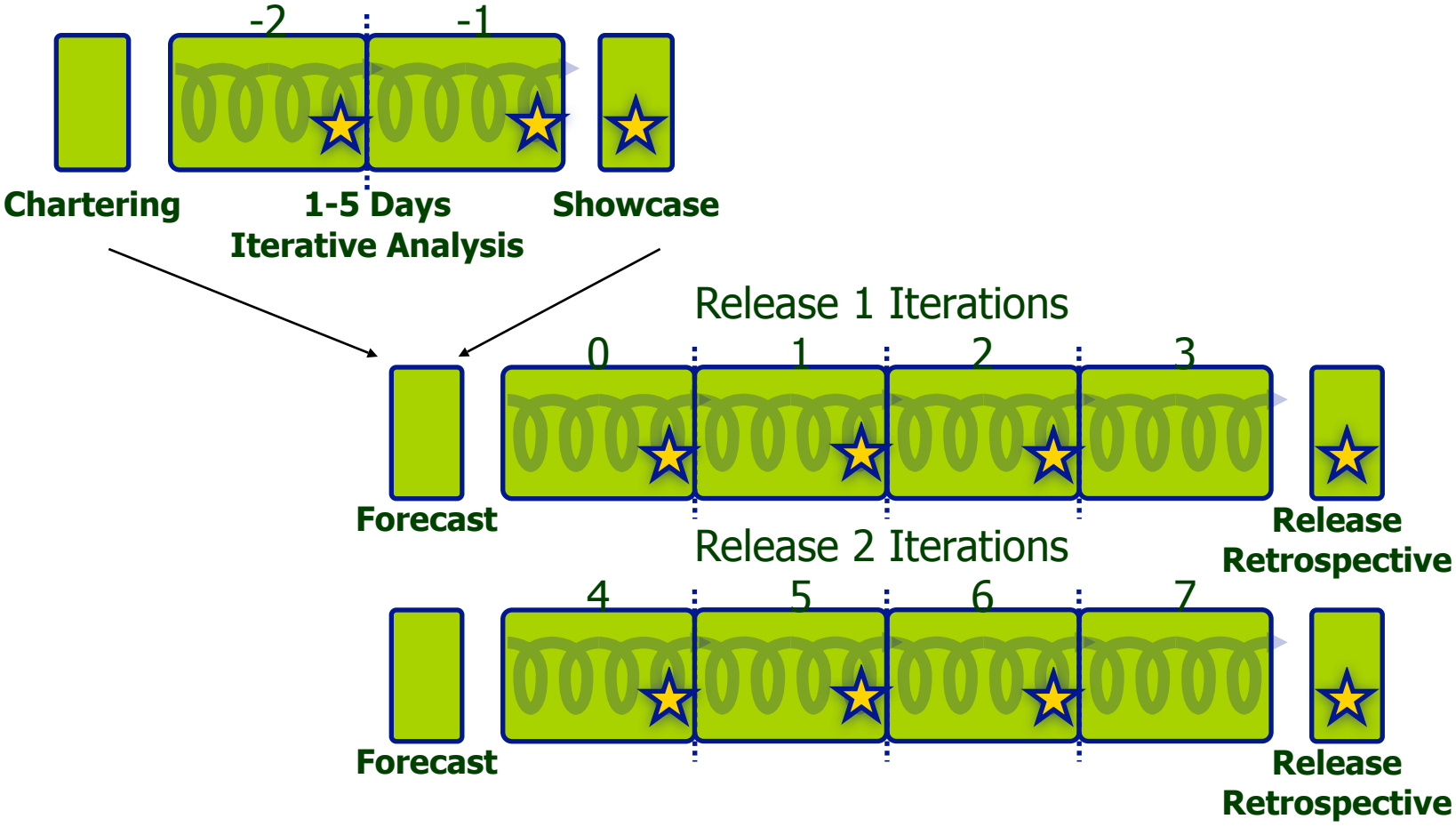
# Create a Project Backlog

image attribution: Thoughtworks



You need a backlog of requirements (best with tactile cards) that you can prioritise with the team (including customers)

# *Forecasting – predicting the journey*

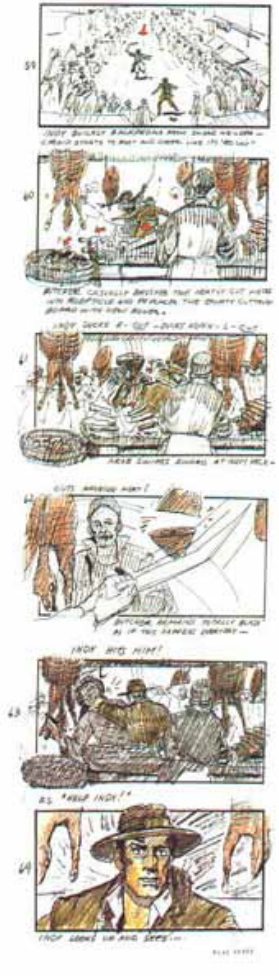


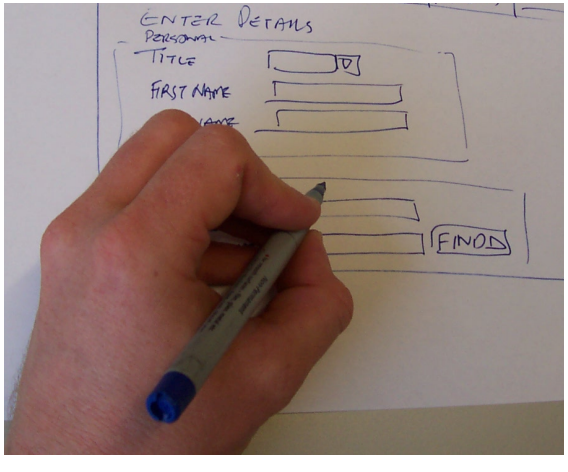

Forecasting – the weather metaphor (ie. its not perfect, but a guide)

The idea is to build up requirement “stories” to determine how the project can break up into smaller releases

# So where do artefacts come from ?

- Real users, product owners, team members
  - Examples from existing systems
  - Kick-off workshops
- Workshops with the whole team exploring options with users
  - Prototypes
  - Persona's
  - Story boards/Whiteboarding
  - High level cards (Epics)

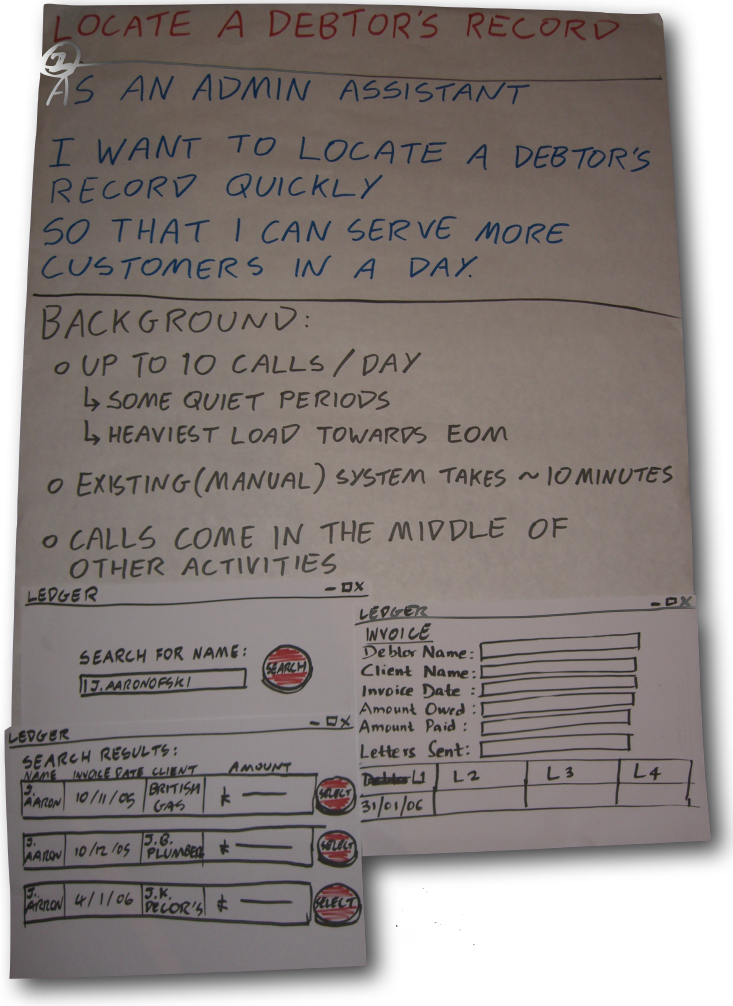





Described the Elseveir team and how they were good at writing stories (they are publishers)

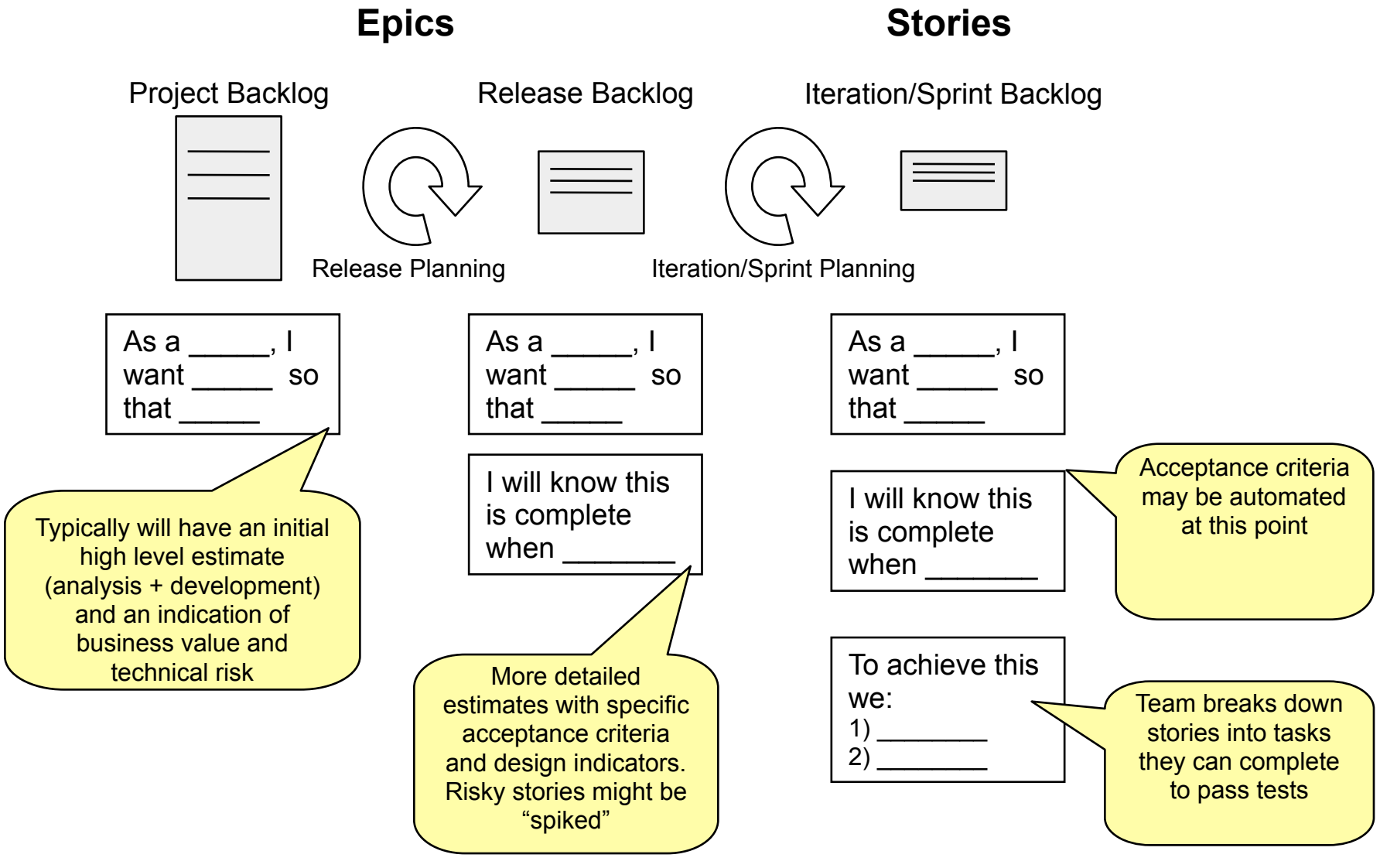
# Creating High Level Visual Artefacts

- High level stories called "Epics"
- Give them useful names
- Describe a goal of a persona and business reason for epic
- Fill in important reminders and background information
  - Observations, previous systems
  - Technical data from developer investigations
- Include visual sketches, diagrams, screenshots
- Hang them in your workspace




Epics, and forming a useful vision of hi level stories

# *Different levels of stories*

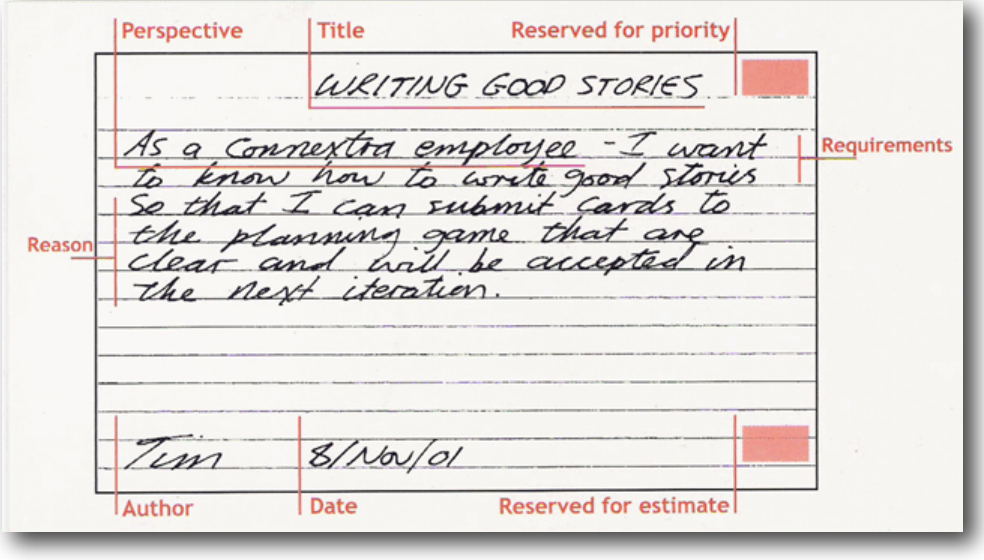



But a story Card is still a Placeholder / Token (for further JIT conversation)

Mature teams generally avoid task breakdown – or do this when pairing on a story in play

# Story Card Technology…

- Cards, a simple effective requirements capture tool
- Resist overusing technology
- Simple and tactile

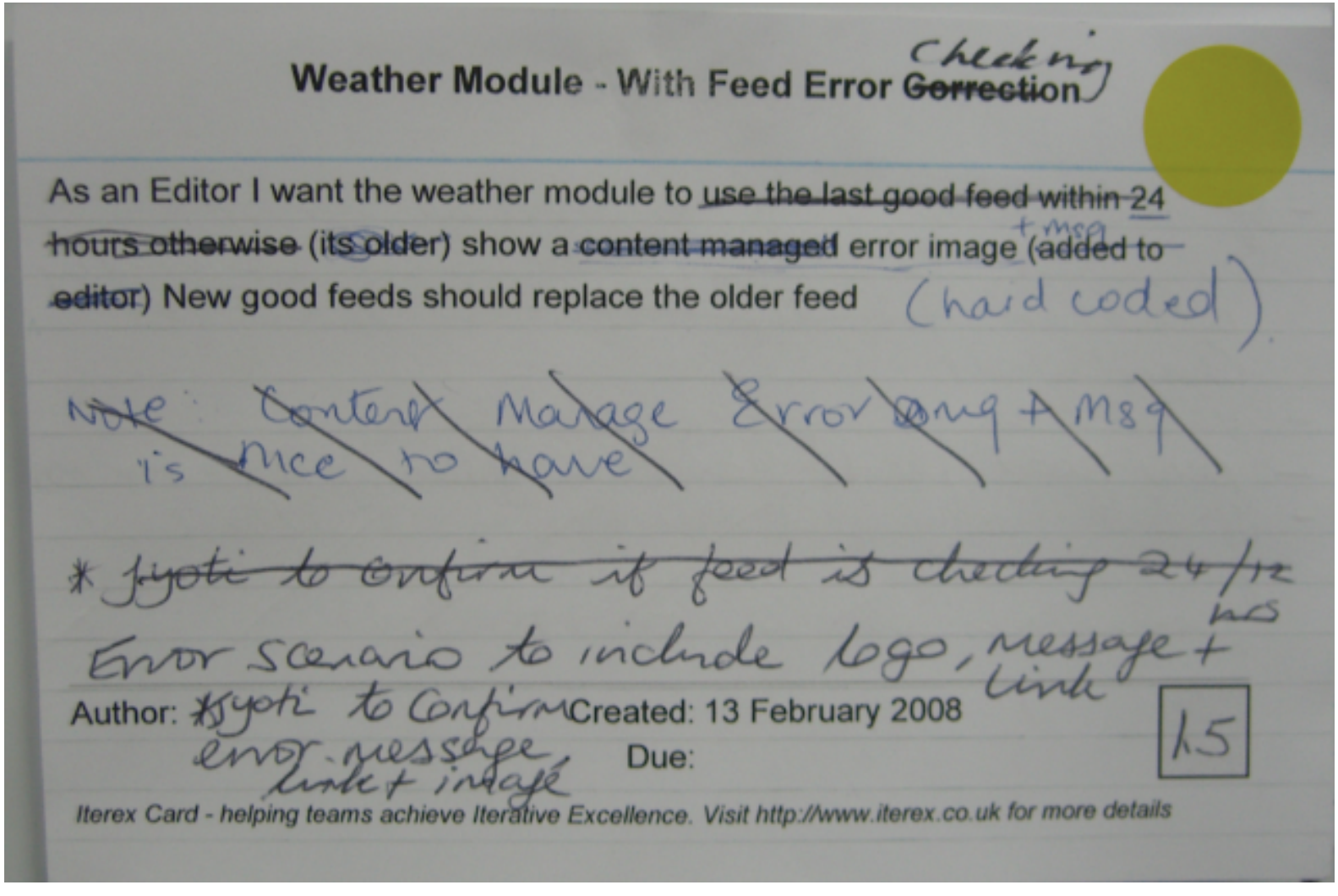



- JIT requirements

- A Story has enough information to allow a basic estimate

But what are stories? – A common format helps, this is the blueprint we designed at “Connextra” which has become widely popular, although there are other variations you can try

# Cards are flexible, even when printed






This is a card printed using “Iterex Planning Cards”, which was then subsequently adjusted when meeting with a customer (several times as work progressed)

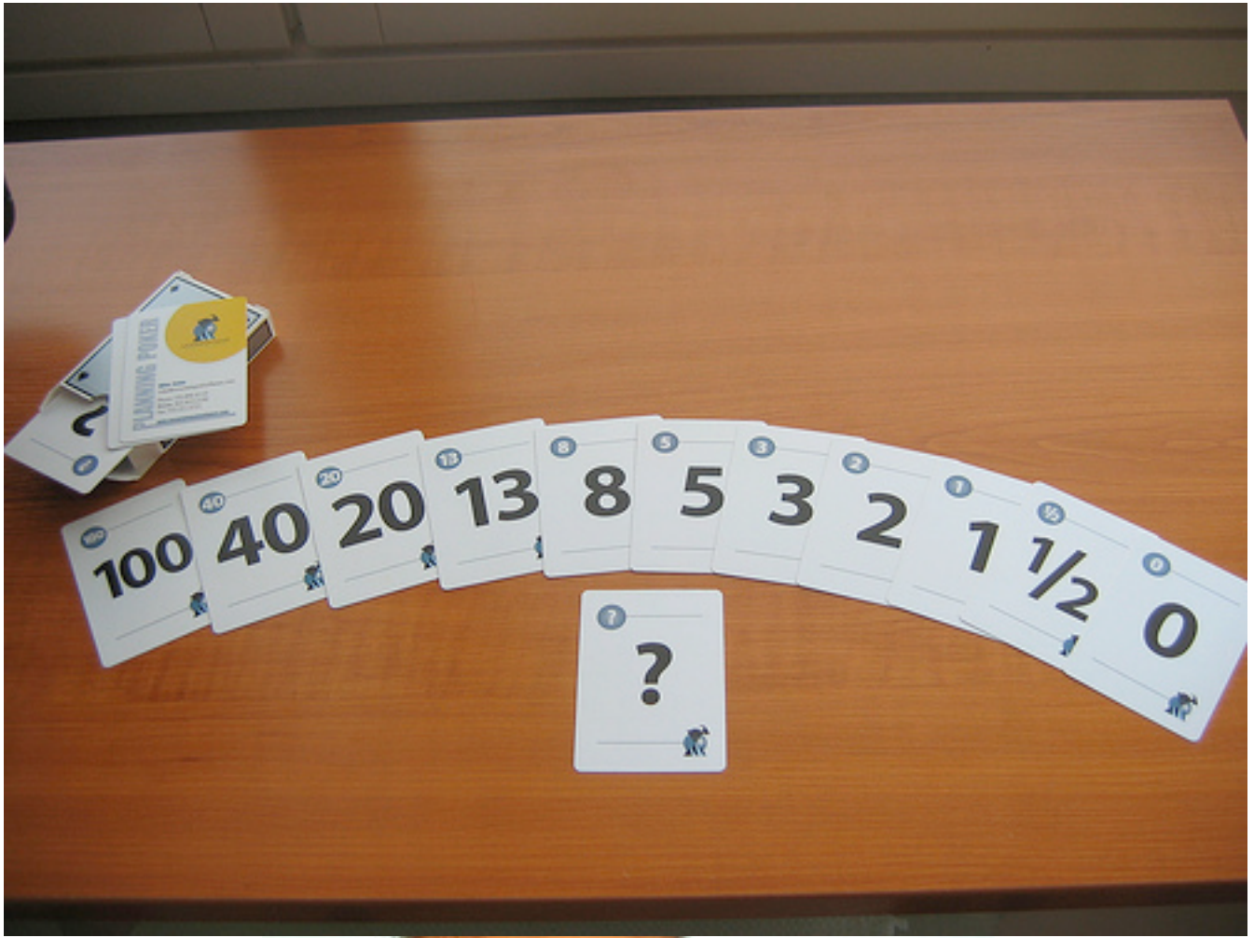
# *Metrics: Velocity*

- Velocity is an unfortunate name (widely misused in industry)
- Mistaken for speed when really its about range (MPG)



Originally used Load Factor – too confusing. Wanted an easy budget figure that the business understood.

# Simple Estimation Requires a Unit




Planning Poker Cards



Talk about selling planning cards vs. using your hand (no-one forgets their hand)

# Simple Estimation Requires a Unit

Planning Poker Cards



Talk about selling planning cards vs. using your hand (no-one forgets their hand)

# *Calculating Velocity*

| | Estimate | Velocity (total) |
|---|---|---|
| As a _____, I want _____ so that _____ | | |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# *Calculating Velocity*

| | Estimate | Velocity (total) |
|---|---|---|
| As a _____, I want _____ so that _____ | 2 | |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# *Calculating Velocity*

| | Estimate | Velocity (total) |
|---|---|---|
| As a _____, I want _____ so that _____ | 2 | ✓ |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# *Calculating Velocity*

| | Estimate | | Velocity (total) |
|---|---|---|---|
| As a _____, I want _____ so that _____ | 2 | ✓ | 2 |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# Calculating Velocity

| | Estimate | | Velocity (total) |
|---|---|---|---|
| As a _____, I want _____ so that _____ | 2 | ✓ | 2 |
| As a _____, I want _____ so that _____ | 1 | | |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# Calculating Velocity

| | Estimate | | Velocity (total) |
|---|---|---|---|
| As a _____, I want _____ so that _____ | 2 | ✓ | 2 |
| As a _____, I want _____ so that _____ | 1 | ✓ | |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# Calculating Velocity

| | Estimate | | Velocity (total) |
|---|---|---|---|
| As a _____, I want _____ so that _____ | 2 | ✓ | 3 |
| As a _____, I want _____ so that _____ | 1 | ✓ | |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# Calculating Velocity

| | Estimate | | Velocity (total) |
|---|---|---|---|
| As a _____, I want _____ so that _____ | 2 | ✓ | 3 |
| As a _____, I want _____ so that _____ | 1 | ✓ | |
| As a _____, I want _____ so that _____ | 3 | | |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# Calculating Velocity

| | Estimate | | Velocity (total) |
|---|---|---|---|
| As a _____, I want _____ so that _____ | 2 | ✓ | 3 |
| As a _____, I want _____ so that _____ | 1 | ✓ | |
| As a _____, I want _____ so that _____ | 3 | ✓ | |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# Calculating Velocity

| | Estimate | | Velocity (total) |
|---|---|---|---|
| As a _____, I want _____ so that _____ | 2 | ✓ | 6 |
| As a _____, I want _____ so that _____ | 1 | ✓ | |
| As a _____, I want _____ so that _____ | 3 | ✓ | |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# Calculating Velocity

| | Estimate | | Velocity (total) |
|---|---|---|---|
| As a _____, I want _____ so that _____ | 2 | ✓ | 6 |
| As a _____, I want _____ so that _____ | 1 | ✓ | |
| As a _____, I want _____ so that _____ | 3 | ✓ | |
| As a _____, I want _____ so that _____ | 2 | | |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# Calculating Velocity

| | Estimate | | Velocity (total) |
|---|---|---|---|
| As a _____, I want _____ so that _____ | 2 | ✓ | 6 |
| As a _____, I want _____ so that _____ | 1 | ✓ | |
| As a _____, I want _____ so that _____ | 3 | ✓ | |
| As a _____, I want _____ so that _____ | 2 | ✗ | |



Emphasise that velocity is simply the sum of the estimates on cards that were completed

# The Planning Game (but should it be fun?)



These are real customers making decisions based on the velocity budget measured by the team.

# The Velocity Budget and Yesterday's Weather

7



Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

# *The Velocity Budget and Yesterday's Weather*

## Last Velocity

**7**

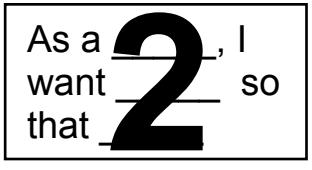


Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

# *The Velocity Budget and Yesterday's Weather*

## Last Velocity

**7**



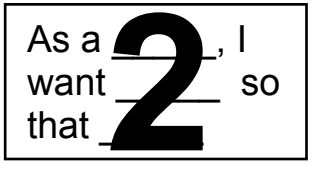



Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

# *The Velocity Budget and Yesterday's Weather*

**Last Velocity**

**7**




**Current Cost**

**2**

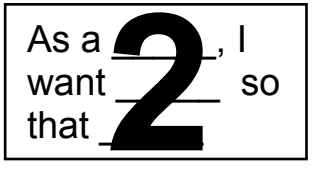


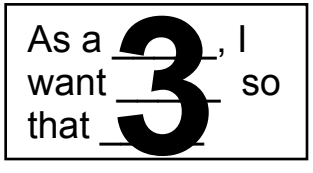
Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

# *The Velocity Budget and Yesterday's Weather*

**Last Velocity**

**7**




**Current Cost**

**2**



Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

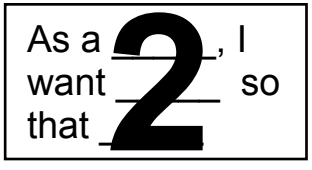
# *The Velocity Budget and Yesterday's Weather*

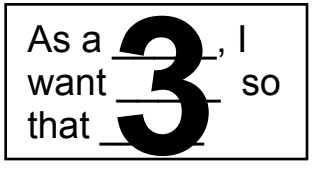
**Last Velocity**

**7**

**Current Cost**

**5**






Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

# *The Velocity Budget and Yesterday's Weather*

**Last Velocity**

**7**

**Current Cost**

**5**

As a ____, I want ____ so that ____

**2**

As a ____, I want ____ so that ____

**3**

As a ____, I want ____ so that ____

**3**



Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

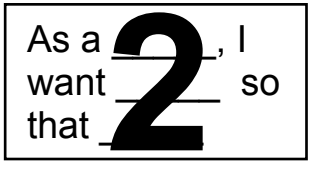
# *The Velocity Budget and Yesterday's Weather*

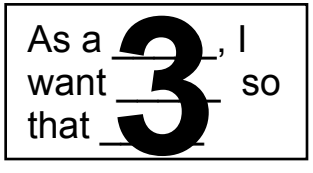
**Last Velocity**

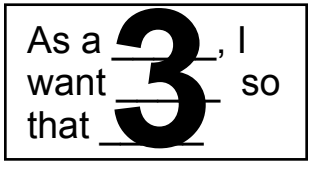
**7**

**Current Cost**

**8**





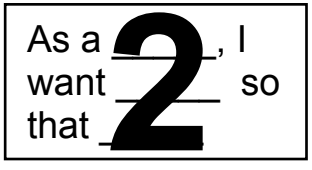

Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

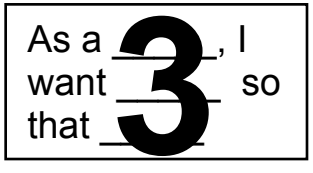
# *The Velocity Budget and Yesterday's Weather*

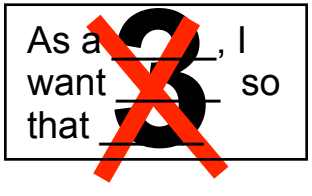
**Last Velocity**

**7**

**Current Cost**





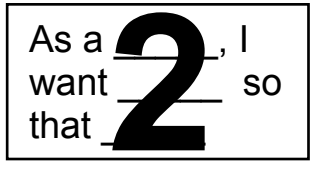

Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

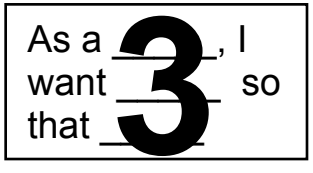
# *The Velocity Budget and Yesterday's Weather*

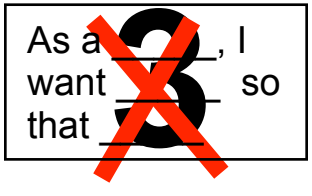
**Last Velocity**

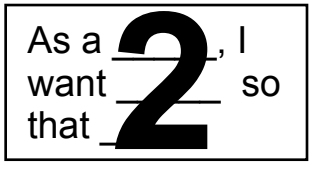
**7**

**Current Cost**



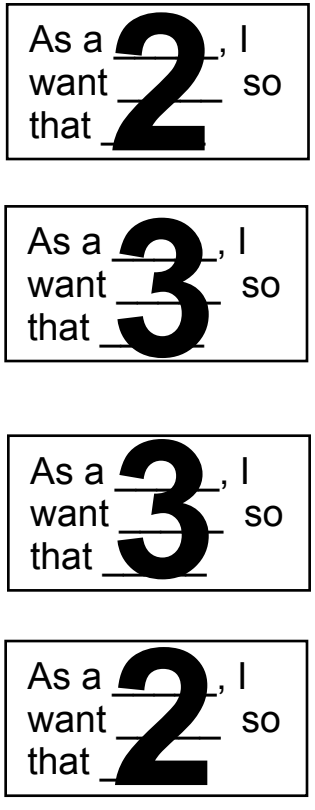



Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

# *The Velocity Budget and Yesterday's Weather*

**Last Velocity**

**7**




**Current Cost**

**7**



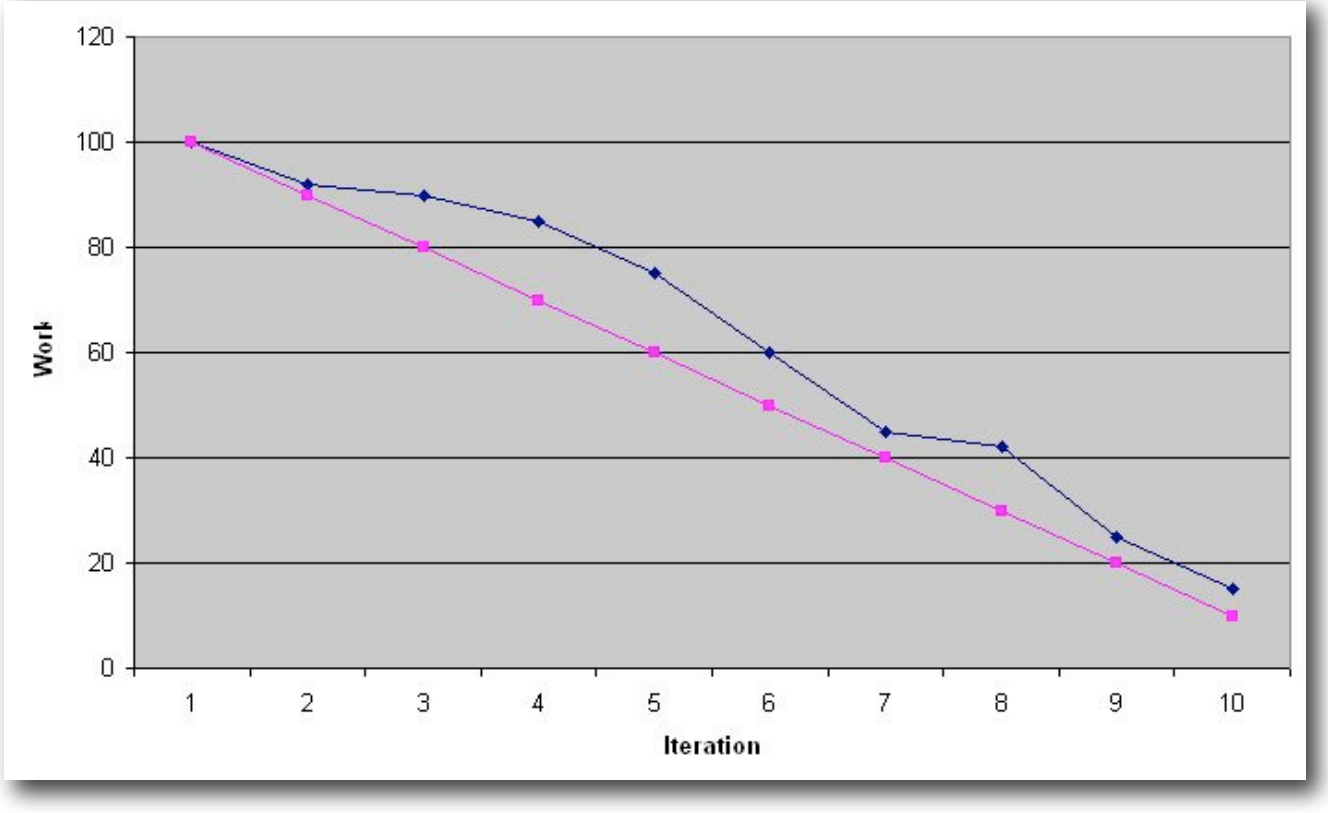
Demonstrate how you use your last velocity as a budget for accepting cards in the next iteration. When you exceed the budget you have to reject that card (either select a new one, or split it some way)

# The Planning Game (making decisions)



Re-emphasize how in these cases the customers had to work with the team to take out cards and find suitable replacements that could fit (this is a collaborative effort)

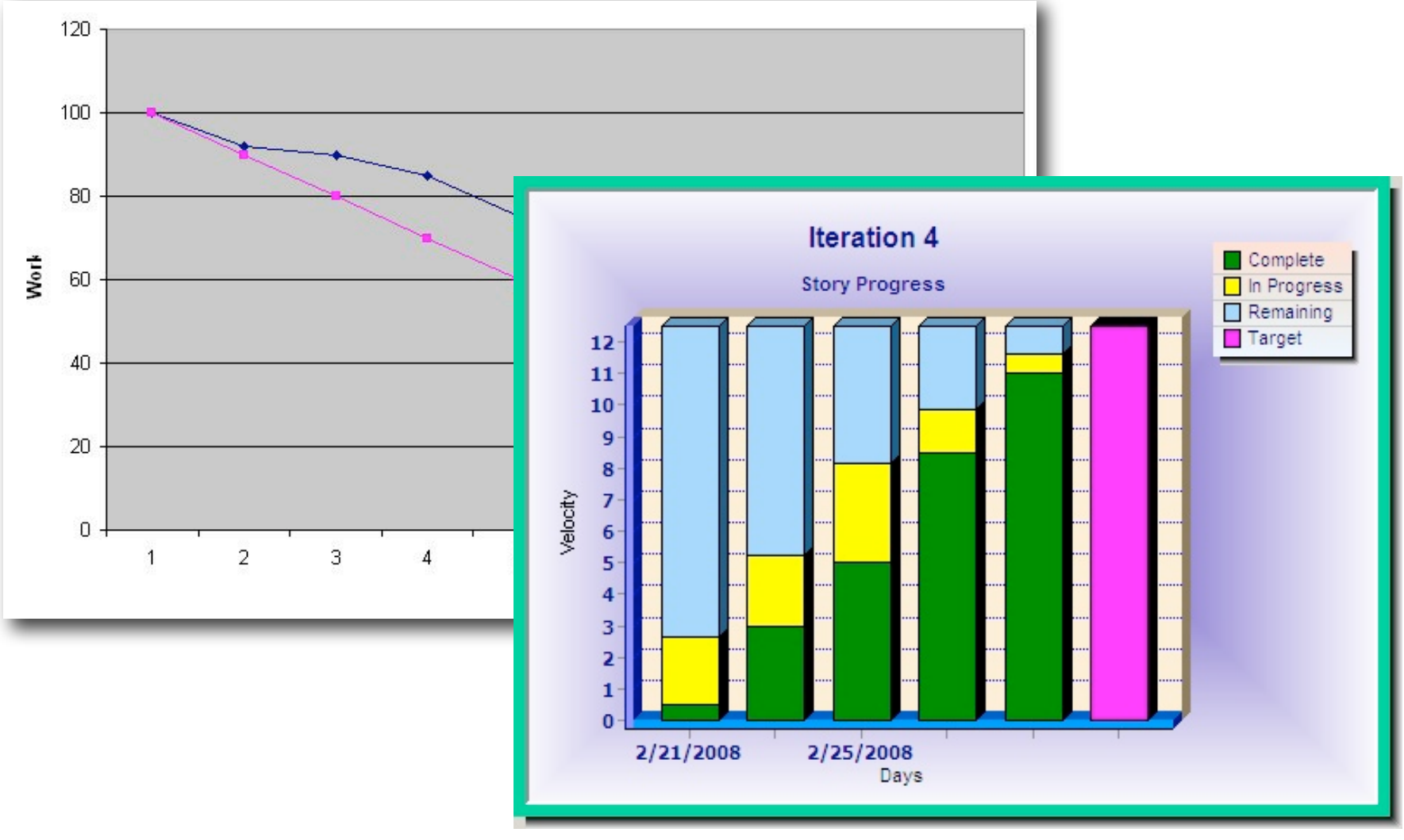
# Tracking: Burndown vs Burnup






Real burndown and burnup diagrams from the Scrum template, and Iterex planning cards

# Tracking: Burndown vs Burnup






Real burndown and burnup diagrams from the Scrum template, and Iterex planning cards

# Really Tracking Progress




Described planning boards, and avatars with standups for showing progress

# Really Tracking Progress



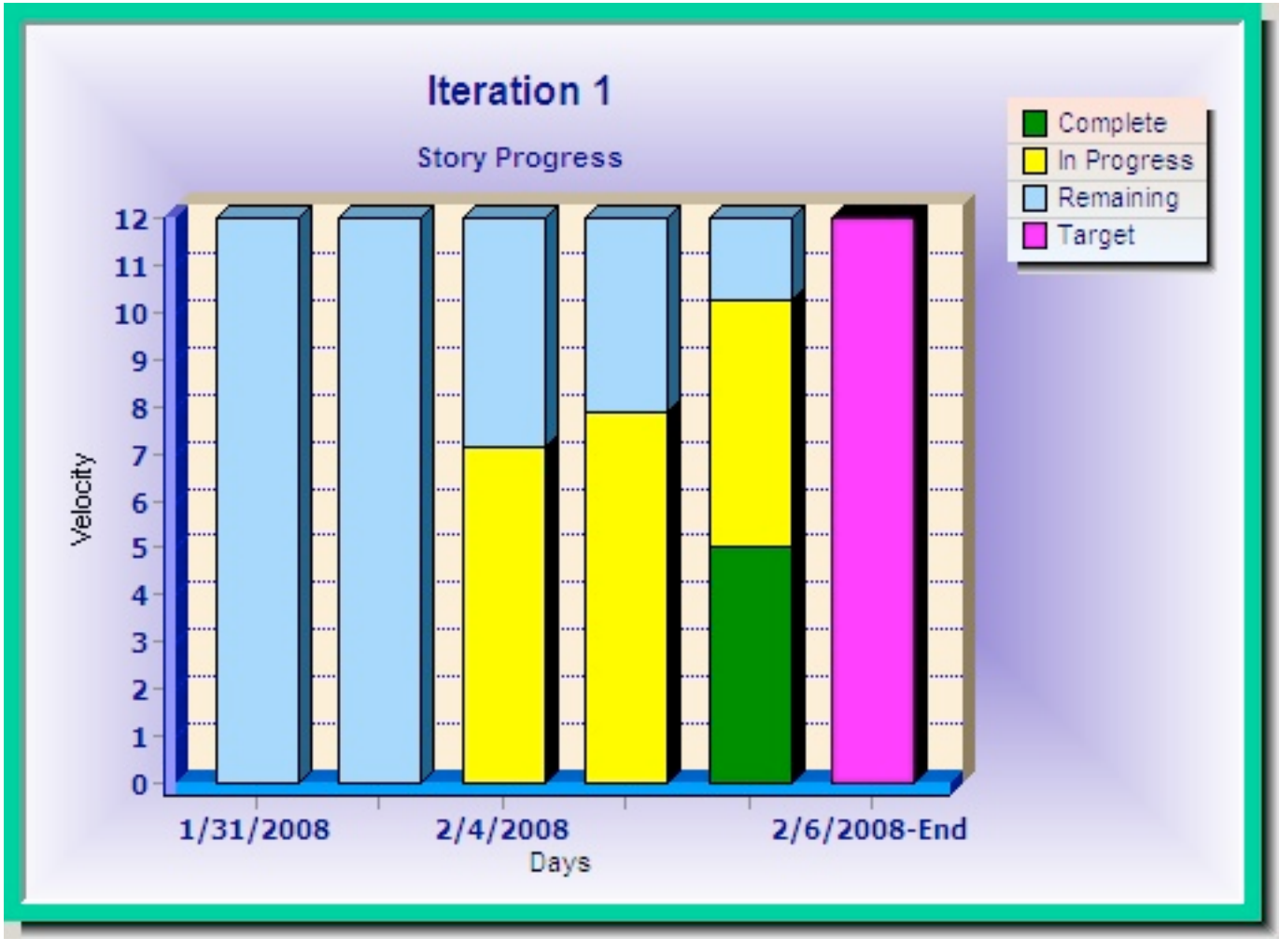

Described planning boards, and avatars with standups for showing progress

# Really Tracking Progress




Described planning boards, and avatars with standups for showing progress

# Burnup Signatures

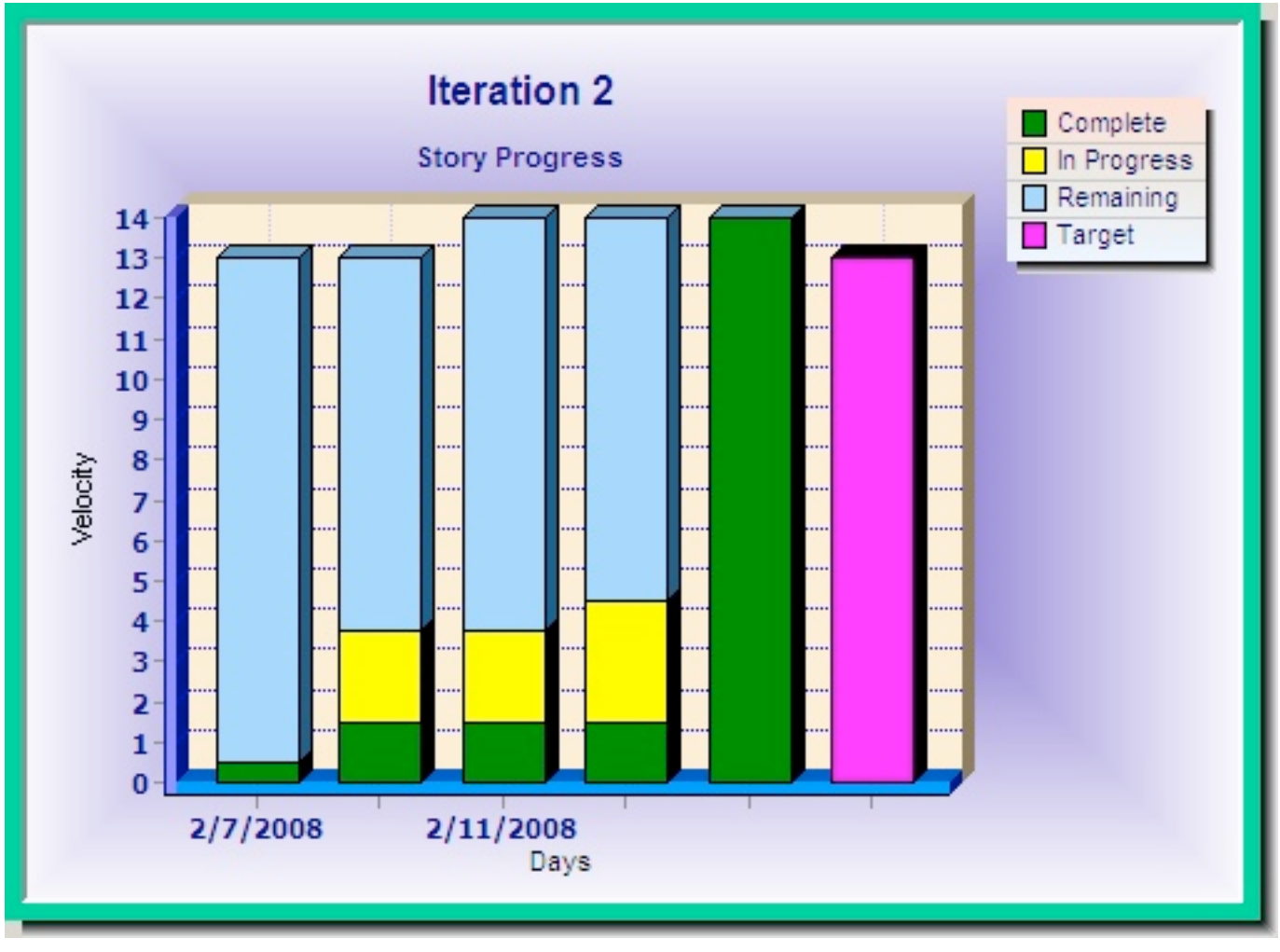



Tracking diagrams from Iterex Story Card



Ask audience how well they think the team is doing in these situations

# Burnup Signatures

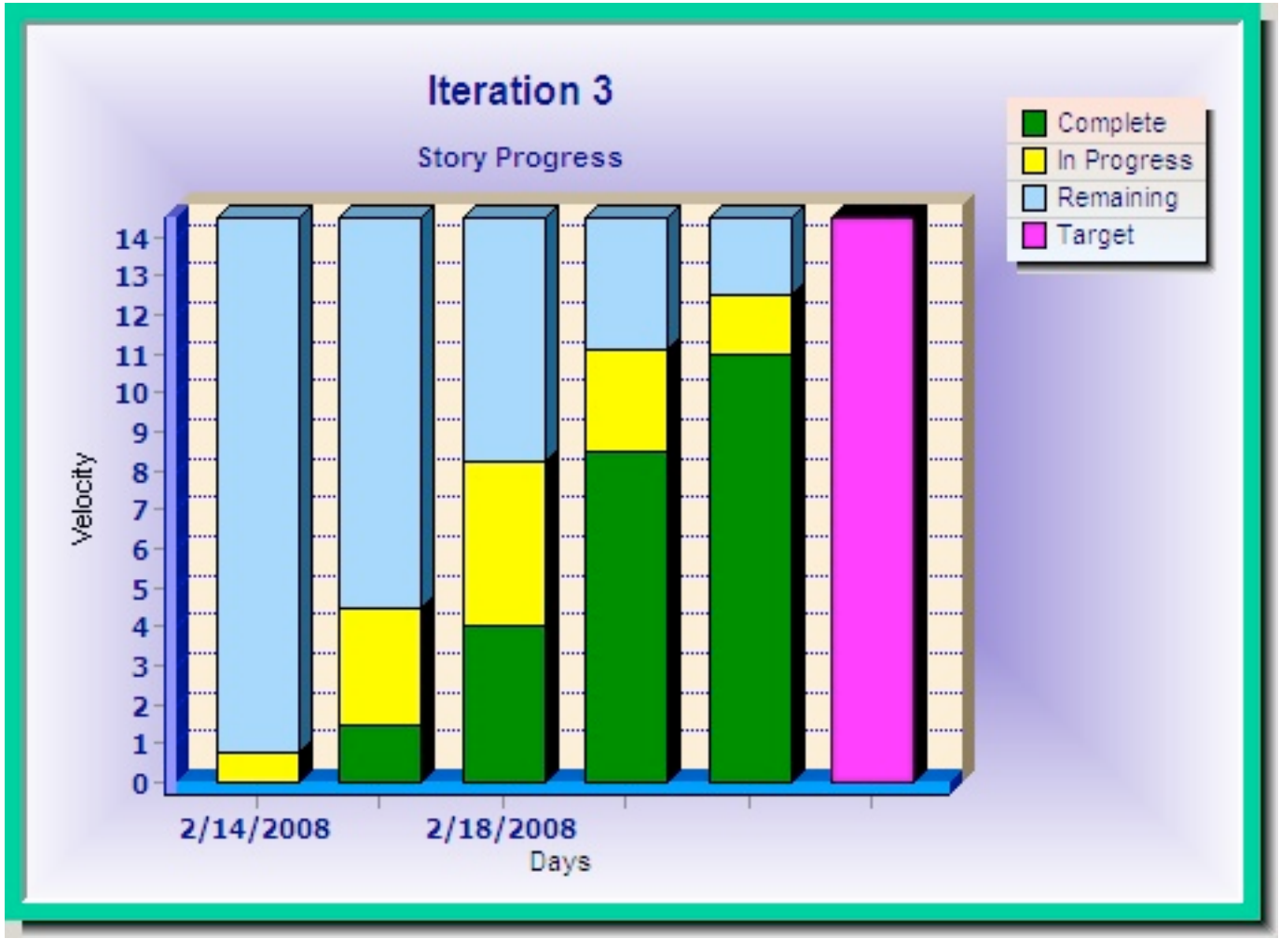



Tracking diagrams from Iterex Story Card



Ask audience how well they think the team is doing in these situations

# Burnup Signatures

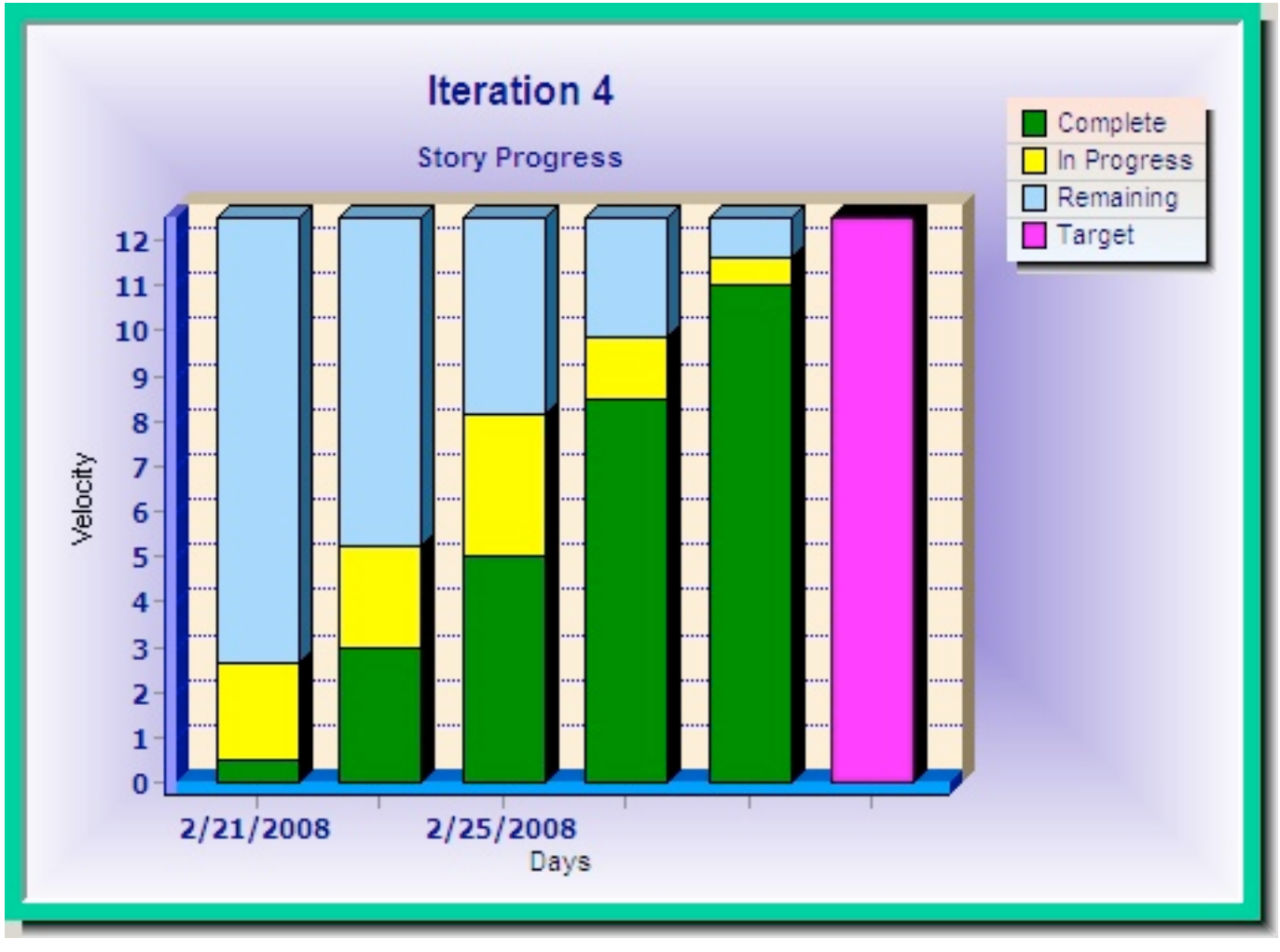



Tracking diagrams from Iterex Story Card



Ask audience how well they think the team is doing in these situations

# Burnup Signatures

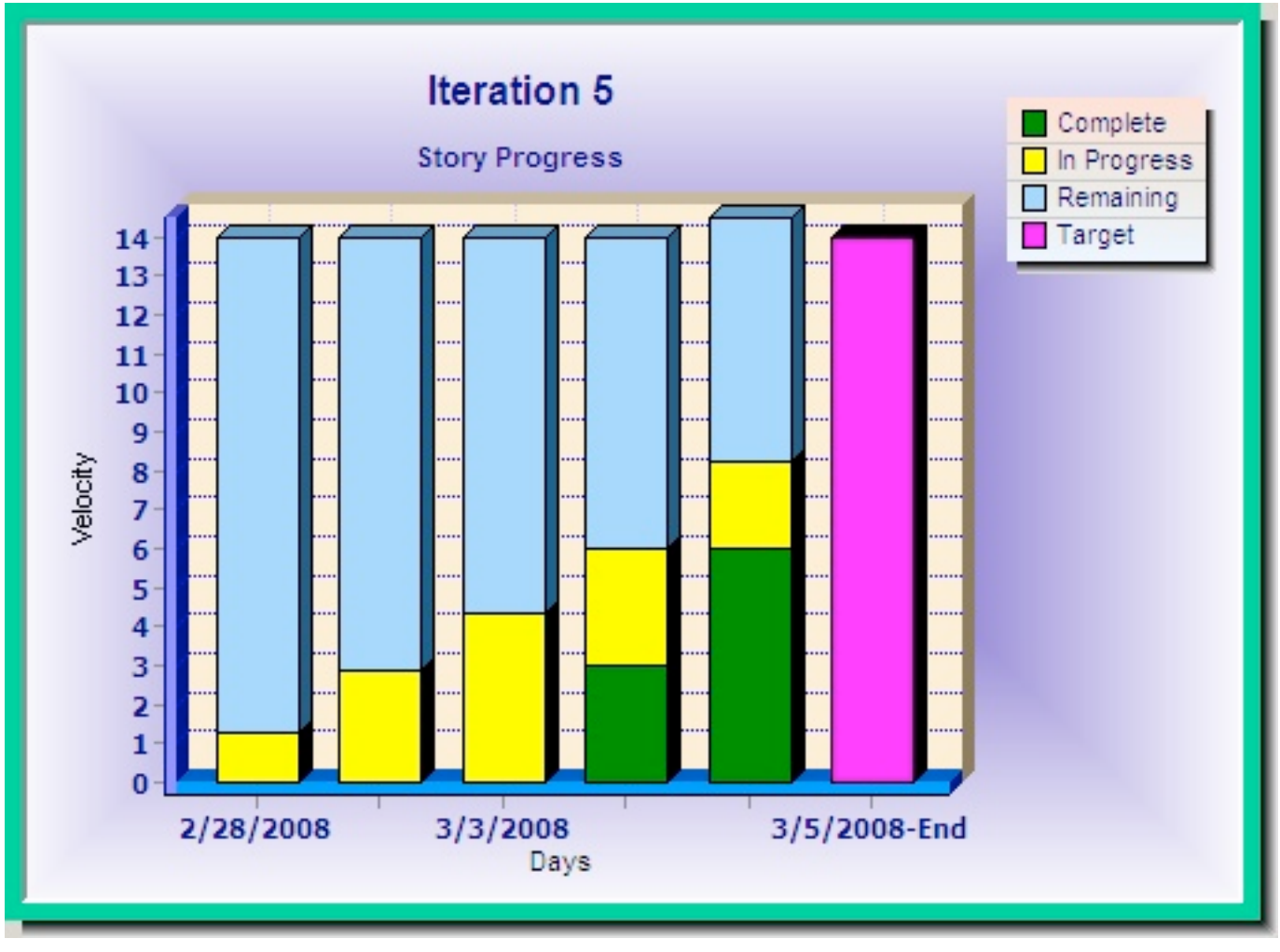



Tracking diagrams from Iterex Story Card



Ask audience how well they think the team is doing in these situations

# Burnup Signatures

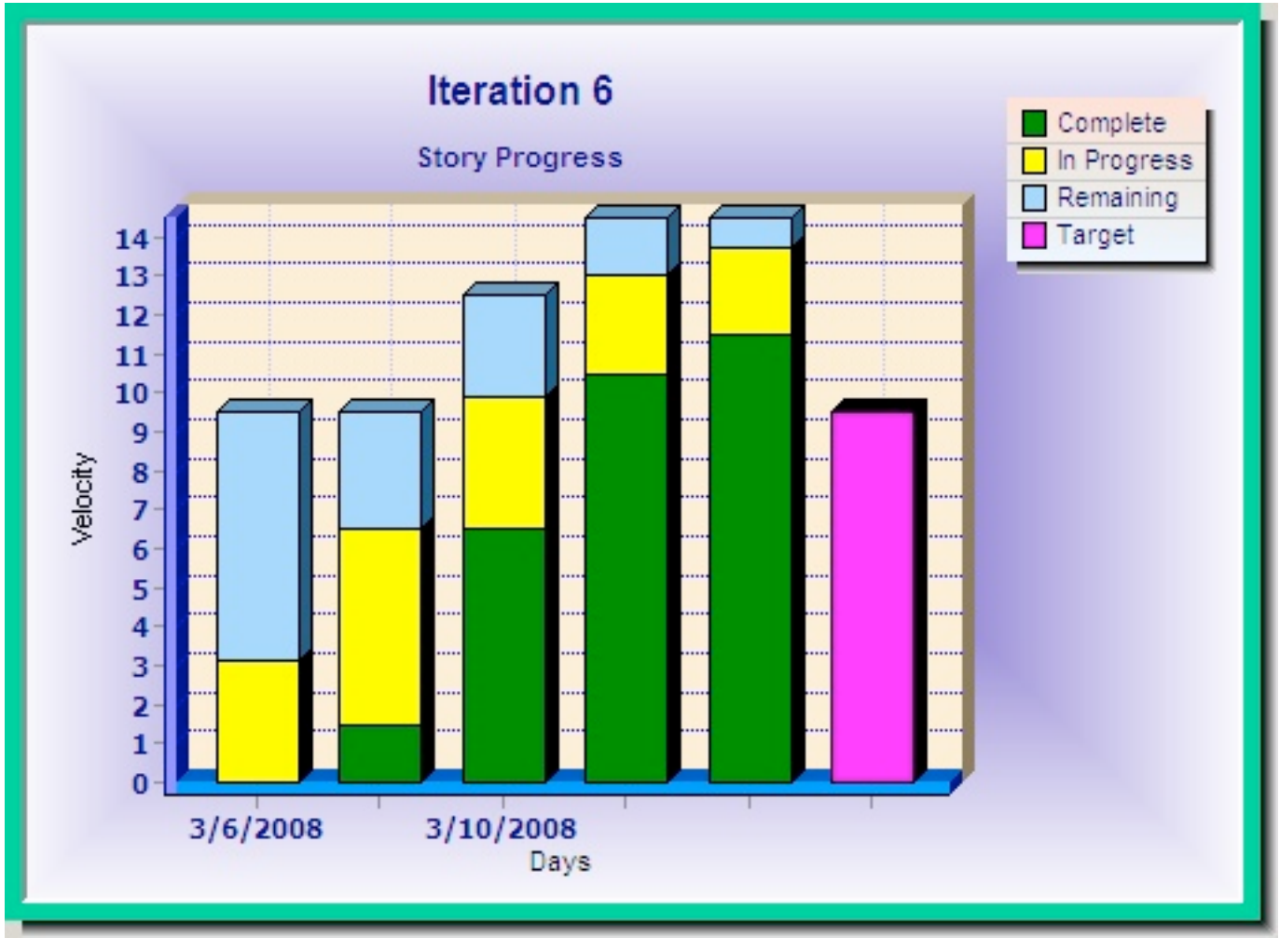



Tracking diagrams from Iterex Story Card



Ask audience how well they think the team is doing in these situations

# Burnup Signatures

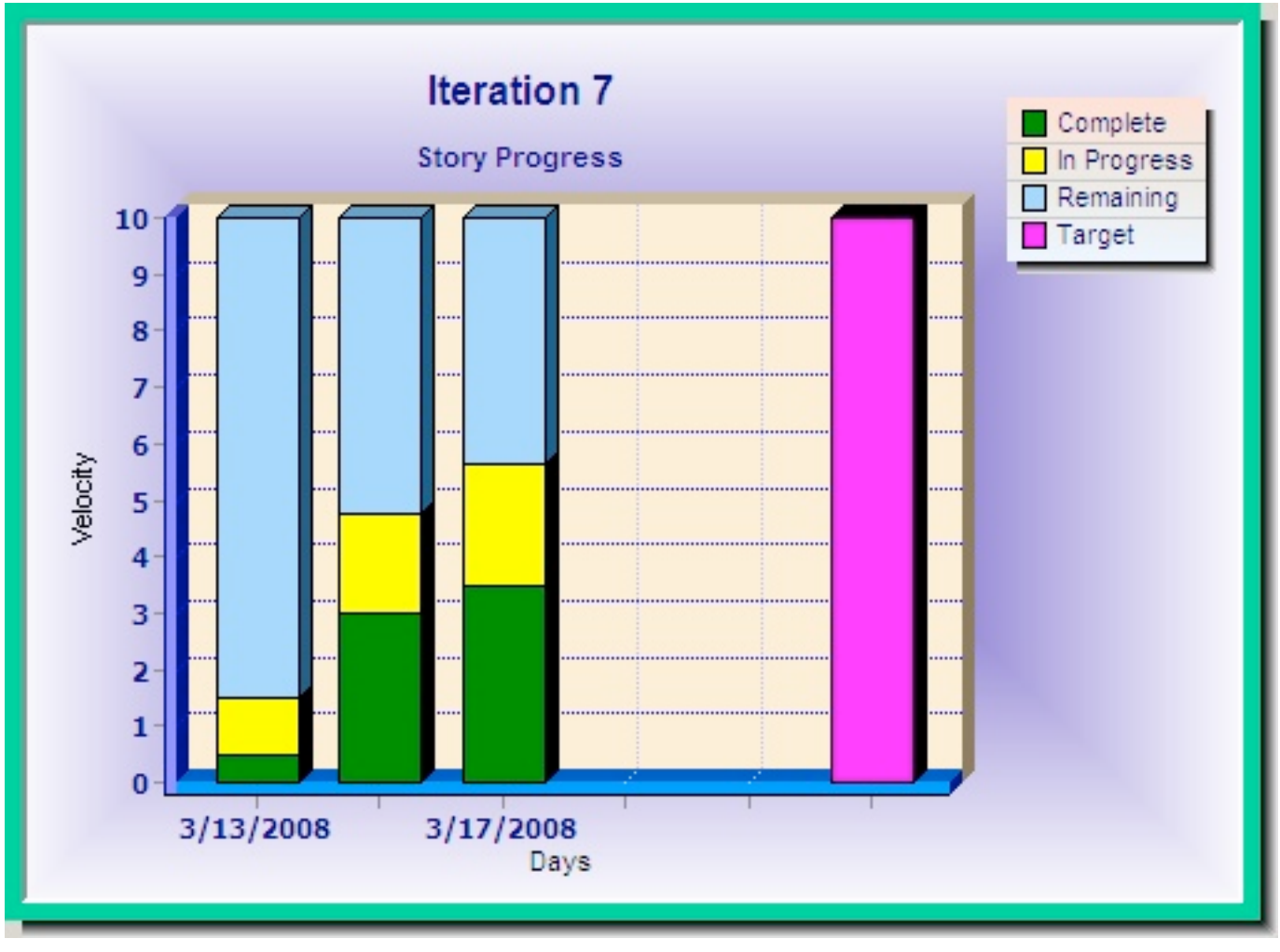



Tracking diagrams from Iterex Story Card



Ask audience how well they think the team is doing in these situations

# Burnup Signatures

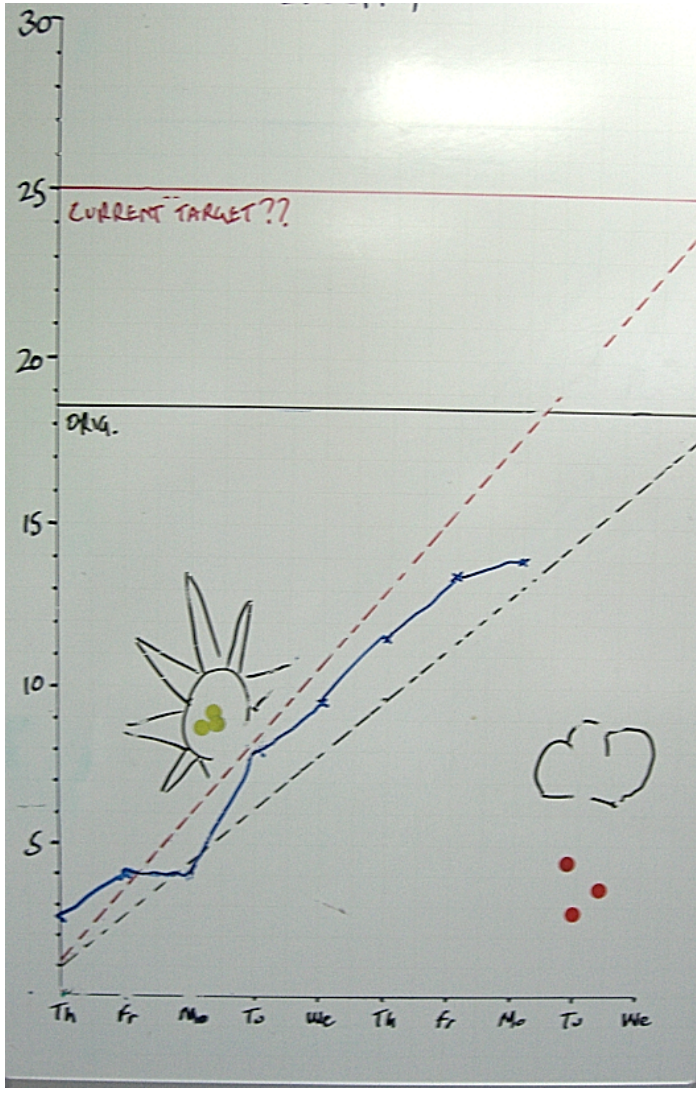



Tracking diagrams from Iterex Story Card



Ask audience how well they think the team is doing in these situations

# Low-Tech can also work better...



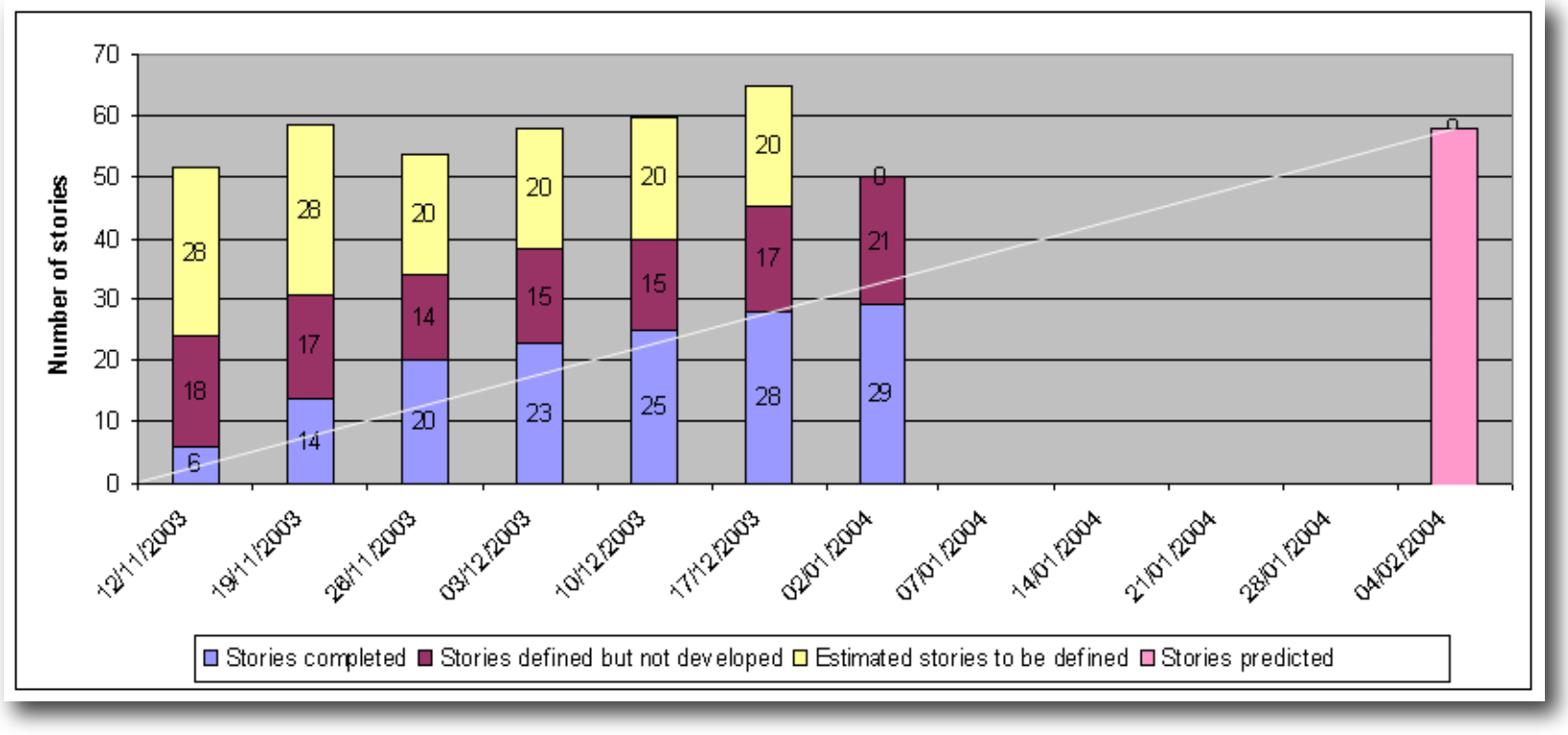



A simple whiteboard that team members update each day can be very educational

# *Leaning towards Equal Cards*

- Keep cards to between 0.5 to 2 days
- Errors tend to cancel each other out
- Allows for easier believable forecasting
- Experiment with Avg. card size

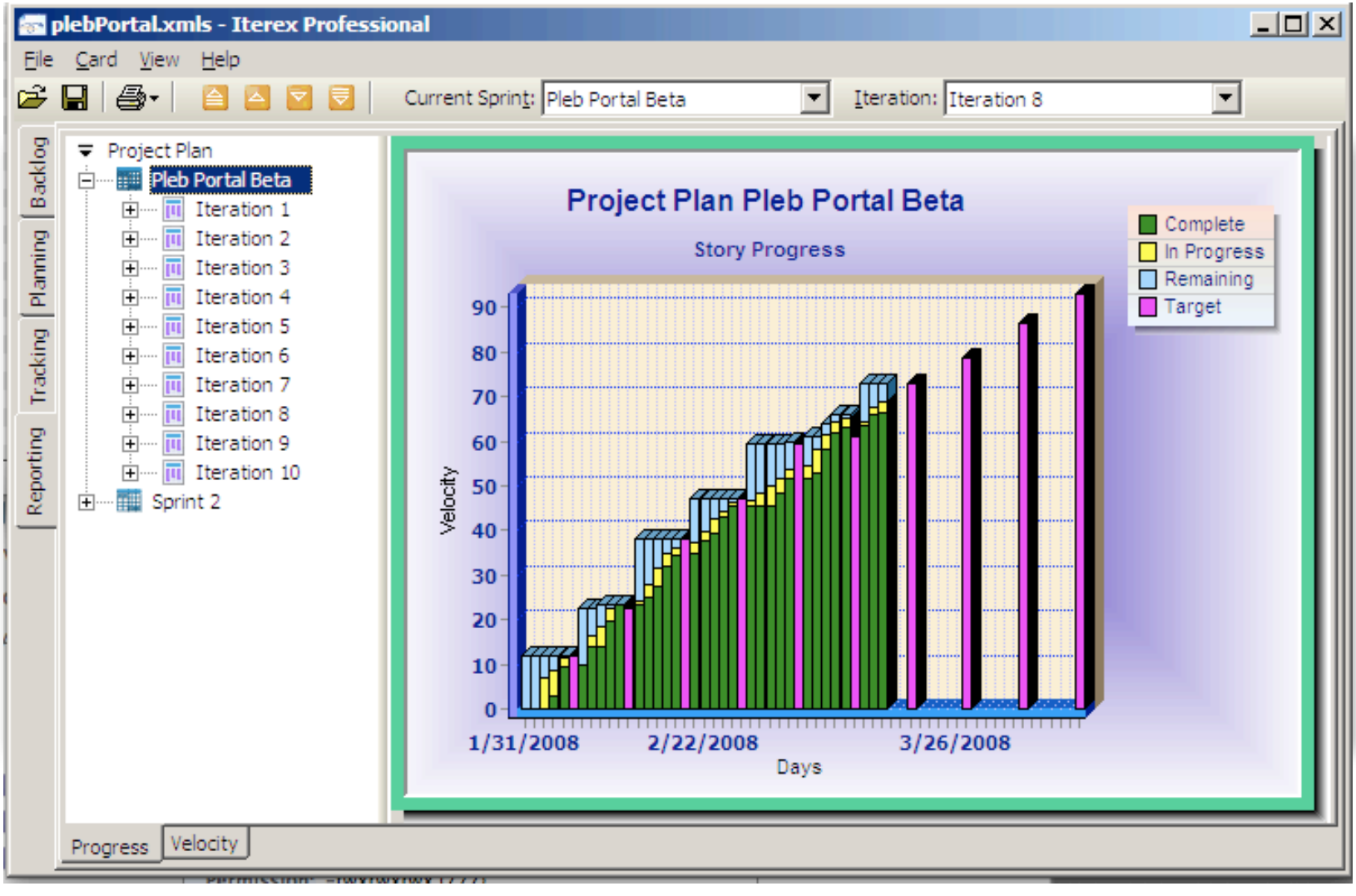
# Projecting Velocity






Talked about early experiments, and avoiding a trending line which leads to better questions of whether things are on track

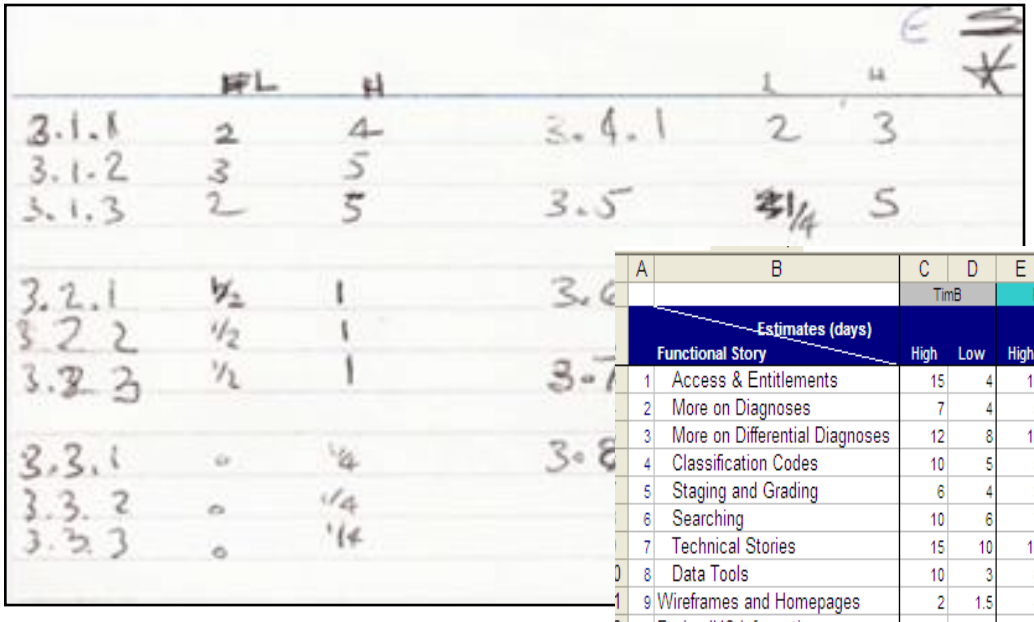
# Projecting Velocity






Talked about early experiments, and avoiding a trending line which leads to better questions of whether things are on track

# What about hi-level planning?

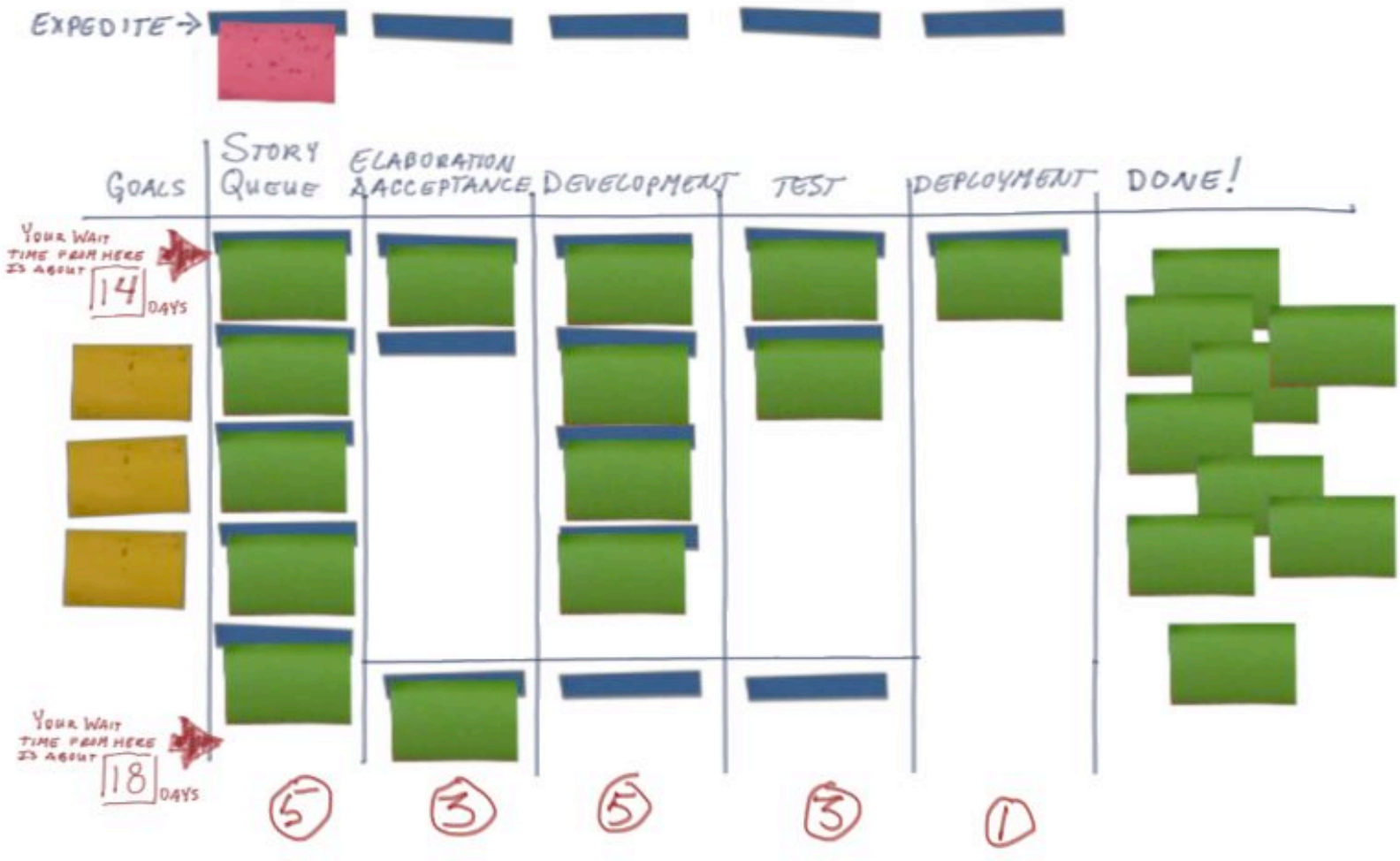



| | | TimB | | Paul | | Matt | | TimM | | JeffL | | Sam | | Dave | | Nev | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Functional Story \ Estimates (days) | High | Low | High | Low | High | Low | High | Low | High | Low | High | Low | High | Low | High | Low | Avg |
| 1 | Access & Entitlements | 15 | 4 | 10 | 5 | 10 | 6 | 12 | 8 | 6 | 3 | 14 | 7 | 10 | 6 | 15 | 10 | 9 |
| 2 | More on Diagnoses | 7 | 4 | 5 | 3 | 5 | 4 | 7 | 4 | 4 | 2 | 5 | 3 | 6 | 3 | | | 4 |
| 3 | More on Differential Diagnoses | 12 | 8 | 13 | 6 | 15 | 10 | 20 | 15 | 10 | 3 | 10 | 5 | 12 | 7 | | | 10 |
| 4 | Classification Codes | 10 | 5 | 5 | 3 | 6 | 4 | 8 | 5 | 4 | 2 | 5 | 2 | 10 | 5 | | | 5 |
| 5 | Staging and Grading | 6 | 4 | 3 | 2 | 6 | 4 | 5 | 5 | 5 | 3 | 5 | 3 | 5 | 2 | | | 4 |
| 6 | Searching | 10 | 6 | 7 | 5 | 8 | 4 | 15 | 10 | 10 | 5 | 10 | 5 | 10 | 4 | | | 8 |
| 7 | Technical Stories | 15 | 10 | 10 | 5 | 9 | 6 | 15 | 10 | | | 12 | 6 | 25 | 12 | | | 11 |
| 8 | Data Tools | 10 | 3 | 8 | 6 | 10 | 8 | 10 | 5 | | | 10 | 5 | 15 | 10 | | | 8 |
| 9 | Wireframes and Homepages | 2 | 1.5 | 4 | 2 | 2 | 2 | 5 | 3 | | | 3 | 1.5 | 2 | 1 | | | 2 |
| 10 | Revise IHC information | 0 | 0 | | | | | | | | | | | | | | | 0 |
| 11 | Protocol for handling specimens | 3 | 2 | 2 | 1 | 5 | 3 | 5 | 3 | 5 | 2 | 2.5 | 1 | 3 | 2 | | | 3 |
| 12 | Reporting information | 5 | 2 | 3 | 1 | 4 | 2 | 5 | 2 | 2 | 1 | 10 | 1 | 2 | 1 | 6 | 3 | 3 |
| 13 | Data Tools 8.2 (incl 8.1) | 15 | 10 | 15 | 8 | 16 | 12 | 15 | 10 | | | | | 25 | 15 | | | 0 |
| 14 | . | | | | | | | | | | | | | | | | | |
| 15 | . | | | | | | | | | | | | | | | | | |
| 16 | . | | | | | | | | | | | | | | | | | |
| | Total | 110 | 59.5 | 85 | 47 | 96 | 65 | 122 | 80 | 46 | 21 | 86.5 | 39.5 | 125 | 68 | 21 | 13 | 69 |

| Pairs | 2 | |
|---|---|---|
| Predictions - 2 pairs | Velocity | Months |
| Pessimistic | 1.25 | 6.9 |
| Optimistic | 2.25 | 3.8 |
| Avg | 1.75 | 4.9 |
| Other | 2 | 4.3 |

Described experiments with hi level planning and individual blink estimation

# Kanban + No Estimation




Reconstructed from photos of boards used at Yahoo



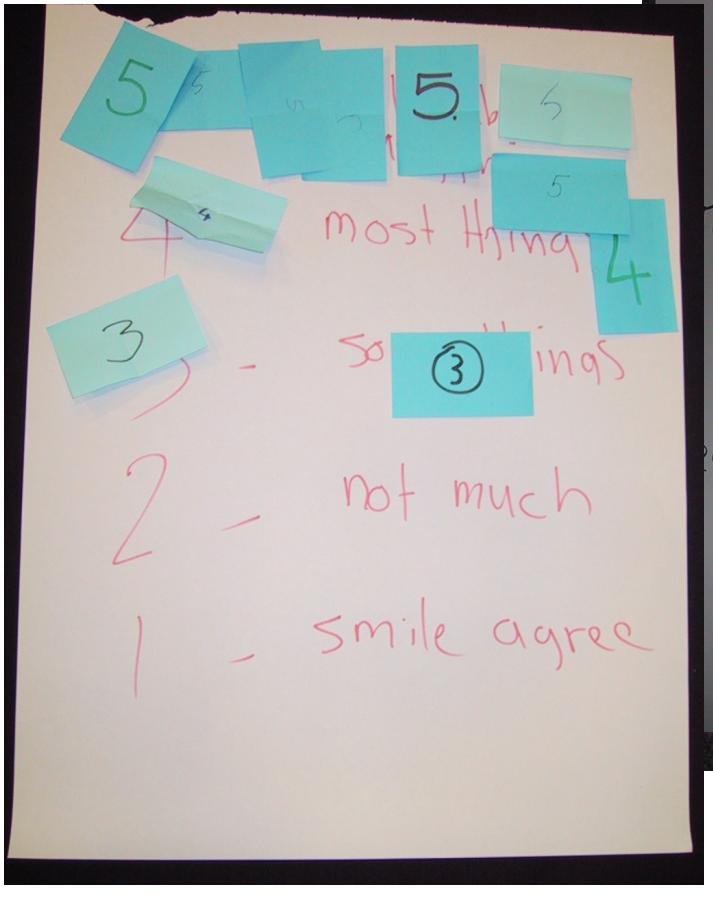
Talked about KanBan card system, queue sizes, siloing issues.
Driver to force choice of stories (without velocity)
Driver to split/change stories (without estimation)

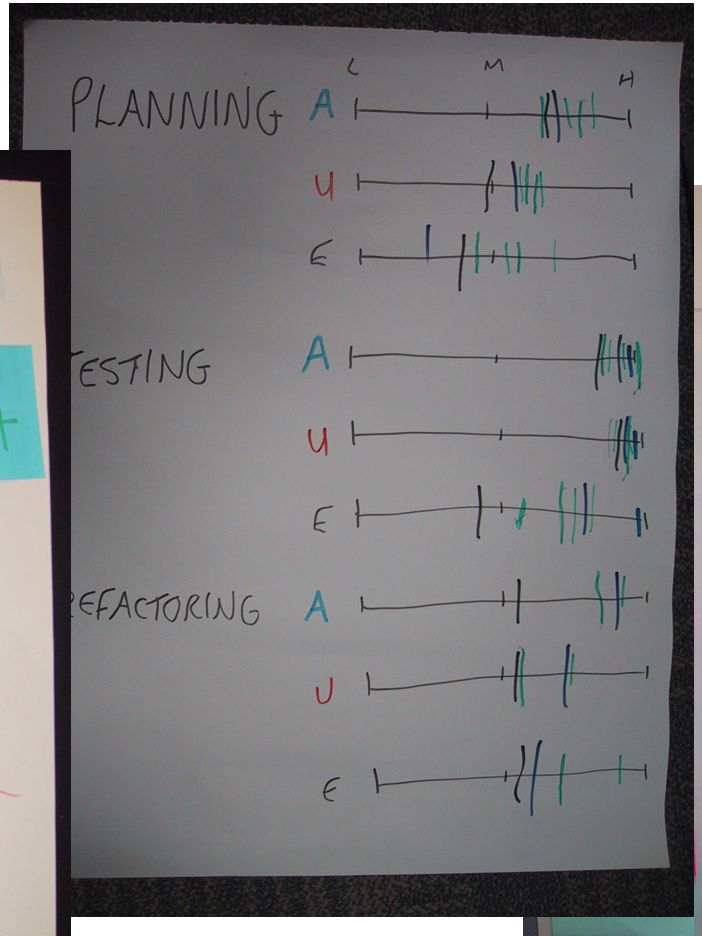
# Fishbowl estimation

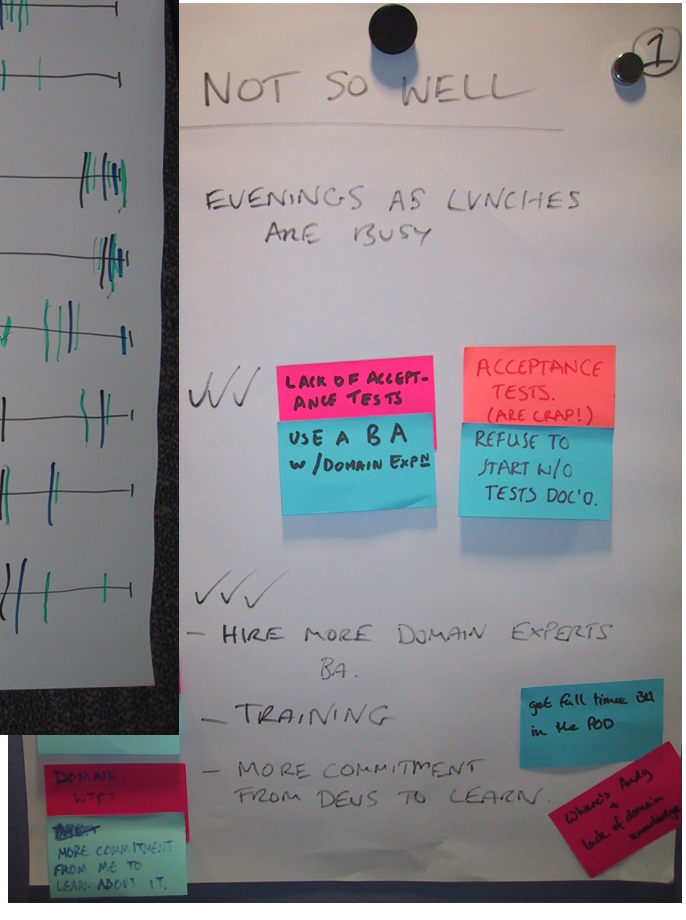


Alternative estimations techniques (courtesy Energized Work)

# Measuring other project aspects…

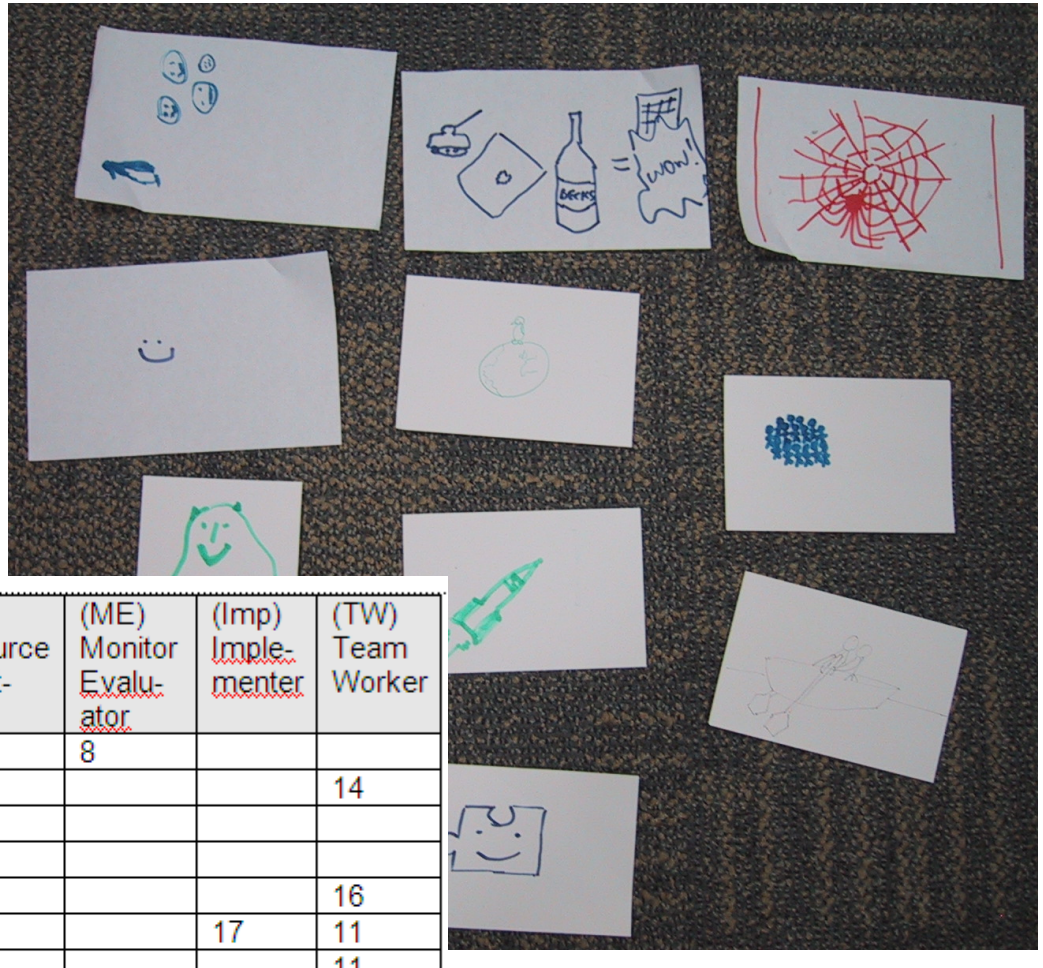



Safety check, practices sliders, retrospectives...

# *Honest feedback not just lip service…*




| Role /Name | (SH) Shaper | (CO) Chair | (CF) Completer Finisher | (PL) Plant | (RI) Resource Invest-igator | (ME) Monitor Evalu-ator | (Imp) Imple-menter | (TW) Team Worker |
|---|---|---|---|---|---|---|---|---|
| N | 17 | 8 | | 17 | 8 | 8 | | |
| A | | 20 | 10 | | | | | 14 |
| Joe | | 11 | | 22 | 17 | | | |
| S | | 11 | 12 | 10 | | | | |
| W | | | 12 | | 13 | | | 16 |
| C | 11 | 17 | | | | | 17 | 11 |
| K | | 12 | | | 14 | | | 11 |
| T | | | 9 | | 19 | | | 16 |
| **Total** | **28** | **79** | **43** | **49** | **71** | **8** | **17** | **68** |

Other interesting activities to try: Project pictures, Belbin team roles

# Don’t be afraid to make process fun...