

little “a” agile

1

Agile vs. agile:

Agile – Proscriptive, rigid, not adaptive, does not allow alternatives

agile – Adaptive, experiment based, evolving

Presentation is about alternatives to “by the book” Agile.

Practices we tried, evaluated, and either continued or discarded



6 month project

2 developers

1 customer representative with high involvement

In hindsight, budget and timeline not clearly defined at start of project

Targeted 4 browsers, tested IE and FF fully.

Java, GWT, and Groovy

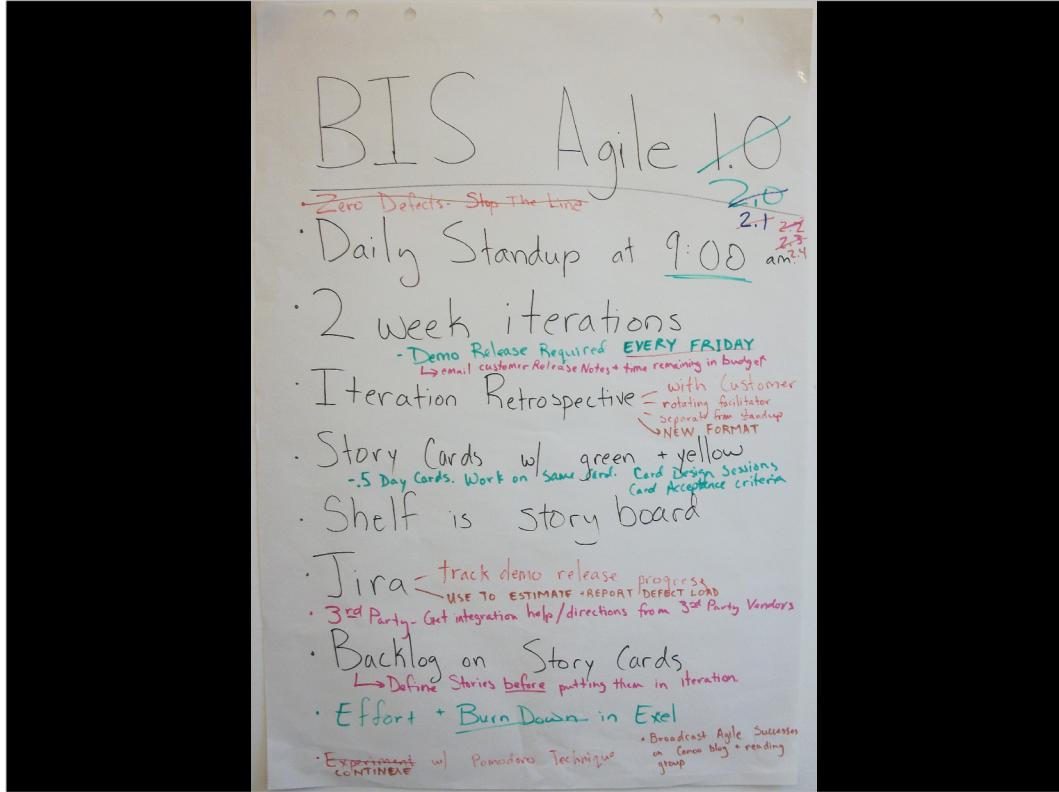
Mix of business and technical tasks:

Ported from GWT 1.4 to 2.0

Increased low test coverage

Added new many features. Improved richness

* Show the application here



Started with 7 Working agreements, updated them as we proceeded and evolved

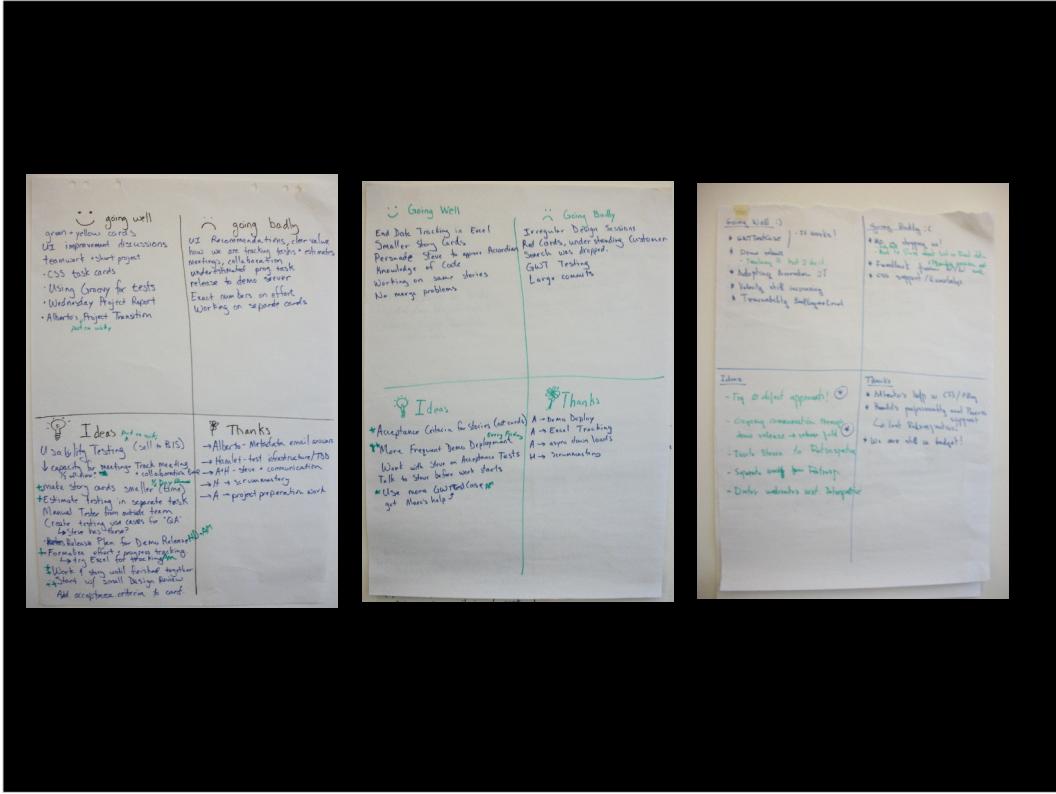
Most important:

- * Iteration Retrospectives
- * Story Card "Hamcordians"
- * Story Card based backlog

BIS Agile 1.0

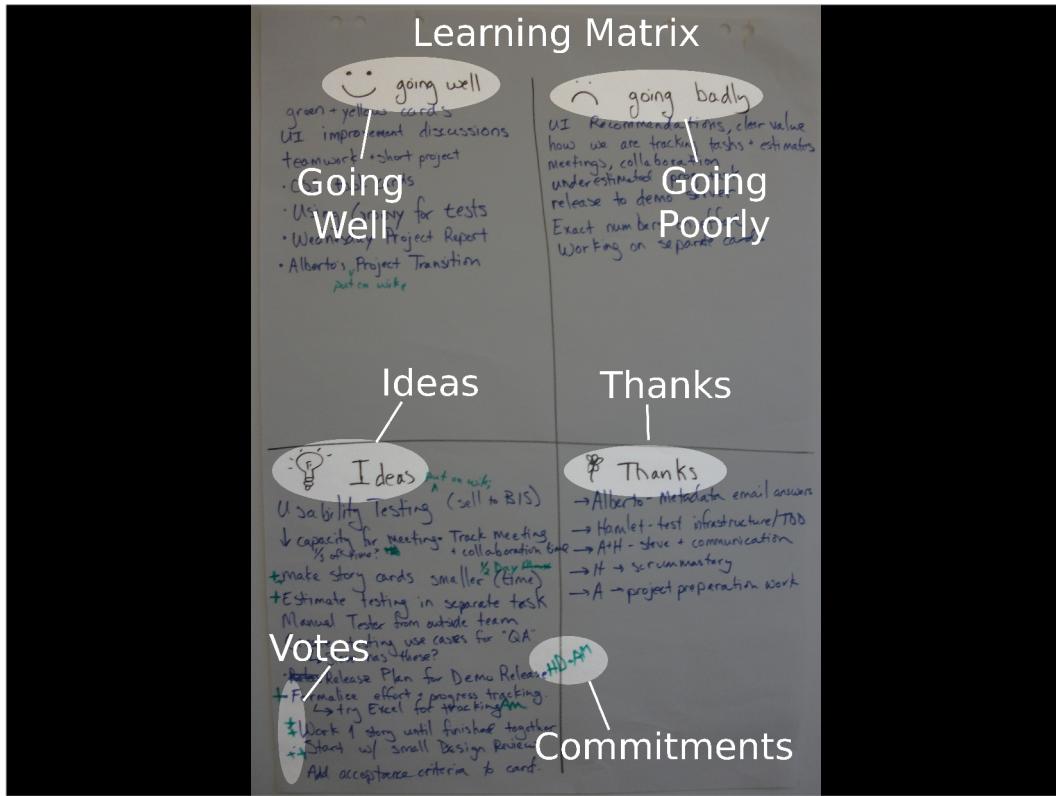
- Zero Defects - Stop The Line
- Daily Standup at 9:00 am^{2.0 2.1 2.2 2.3 2.4}
 - 2 week iterations
 - Iteration Retrospective
 - Demo Release Required **EVERY FRIDAY**
↳ Email customer Release Notes + time remaining in budget
 - with Customer
↳ rotating facilitator
↳ Separated from standup
↳ NEW FORMAT
 - Story Cards w/ green + yellow
 - 5 Day Cards. Work on same card. Card Design Sessions
Card Acceptance criteria
 - Shelf is story board
 - Jira - track demo release progress
↳ USE TO ESTIMATE + REPORT DEFECT LOAD
 - 3rd Party - Get integration help/directions from 3rd Party Vendors
 - Backlog on Story Cards
 - ↳ Define Stories before putting them in iteration
 - Effort + Burn Down in Excel
 - Experiment w/ Pomodoro Technique
 - ↳ CONTINUE
 - ↳ Broadcast Agile Successes
on Concur blog + reading group

Retrospectives were very positive.



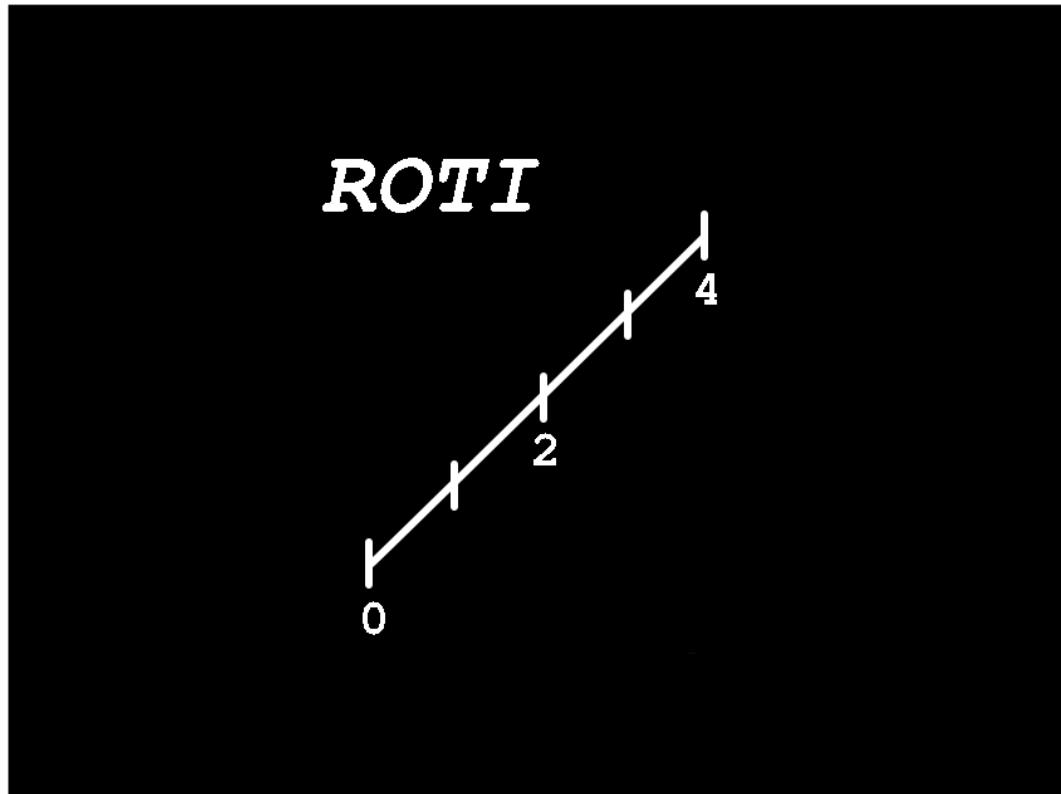
Retrospectives held every 2 weeks. Usually had 4 participants. Stood entire time.

Here are the artifacts from the sessions.



Learning Matrix

1. Brainstorm
2. Discuss Options
3. Decide on actions
4. Retrospect on the Retrospective...



A short way to evaluate any meeting.



My favorite idea for improvement
came during a retrospective with
the customer...

Customers were invited to the retrospective.
We wish we had done this more and earlier.

GWT Wins

UIBinder

OOPHM Debugging

Event Bus

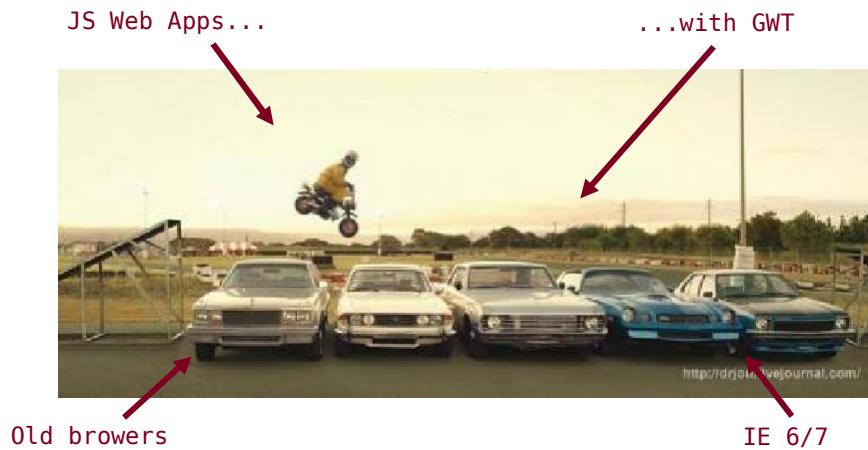
SpeedTracer

New Containers (no more <table>)

GWT 2.0 was a big improvement over 1.4

It was worthwhile to upgrade.

GWT 2.0



10

So much time was spent on cross-platform code,
especially IE.

Biggest required skill for us was CSS



“We should offer a discount to our customers for ***not*** supporting IE6.”

This recommendation came from the customer. It is a good idea.

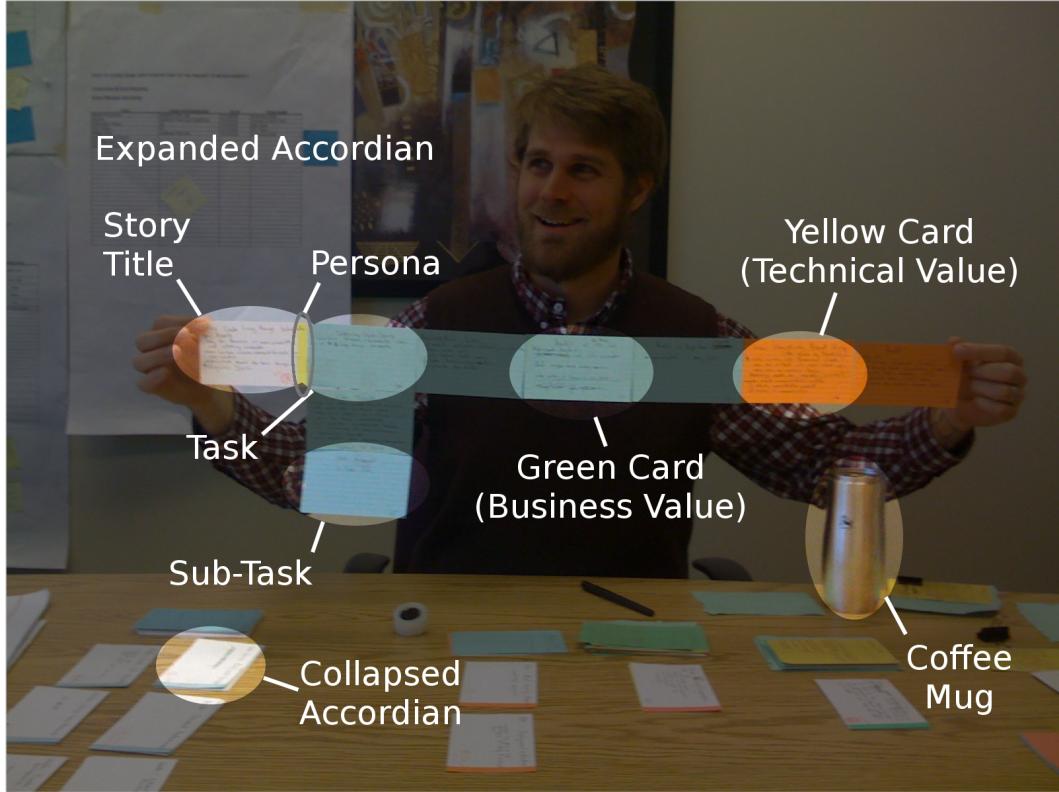
BIS Agile 1.0

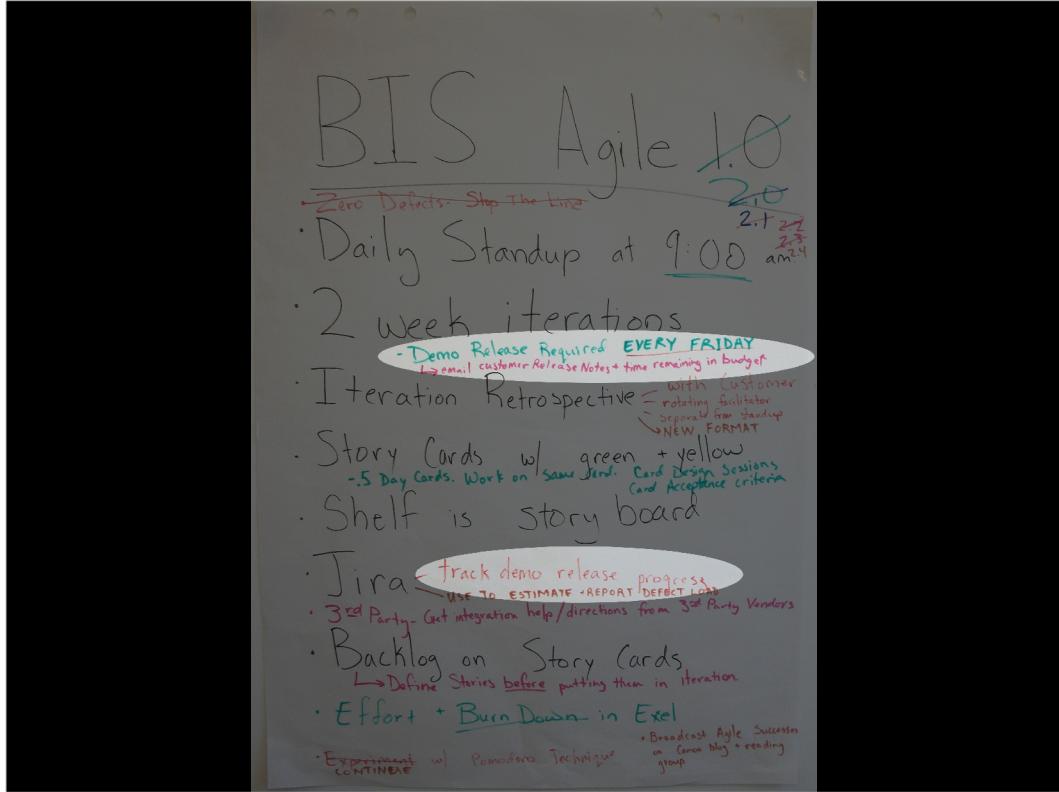
- Zero Defects - Stop The Line
- Daily Standup at 9:00 am<sup>2.0
2.1
2.2
2.3
2.4</sup>
 - 2 week iterations
 - Iteration Retrospective
 - Demo Release Required **EVERY FRIDAY**
 - ↳ Email customer Release Notes + time remaining in body of
 - ↳ rotating facilitator
 - ↳ Separated from standup
 - ↳ NEW FORMAT
 - Story Cards w/ green + yellow
 - 5 Day Cards. Work on same card. Card Design Sessions, Card Acceptance criteria
 - Shelf is story board
 - Jira - track demo release
 - ↳ USE TO ESTIMATE + REPORT DEFECT LOAD
 - 3rd Party - Get integration help/directions from 3rd Party Vendors
 - Backlog on Story Cards
 - ↳ Define Stories before putting them in iteration
 - Effort + Burn Down in Excel
 - Experiment w/ Pomodoro Technique
 - CONTINUE
 - Broadcast Agile Successes
 - ↳ on Concur blog + reading group

Work and backlog were tracked with Story Cards.



This is me holding a “Hamcordian”. It communicates a lot of information.

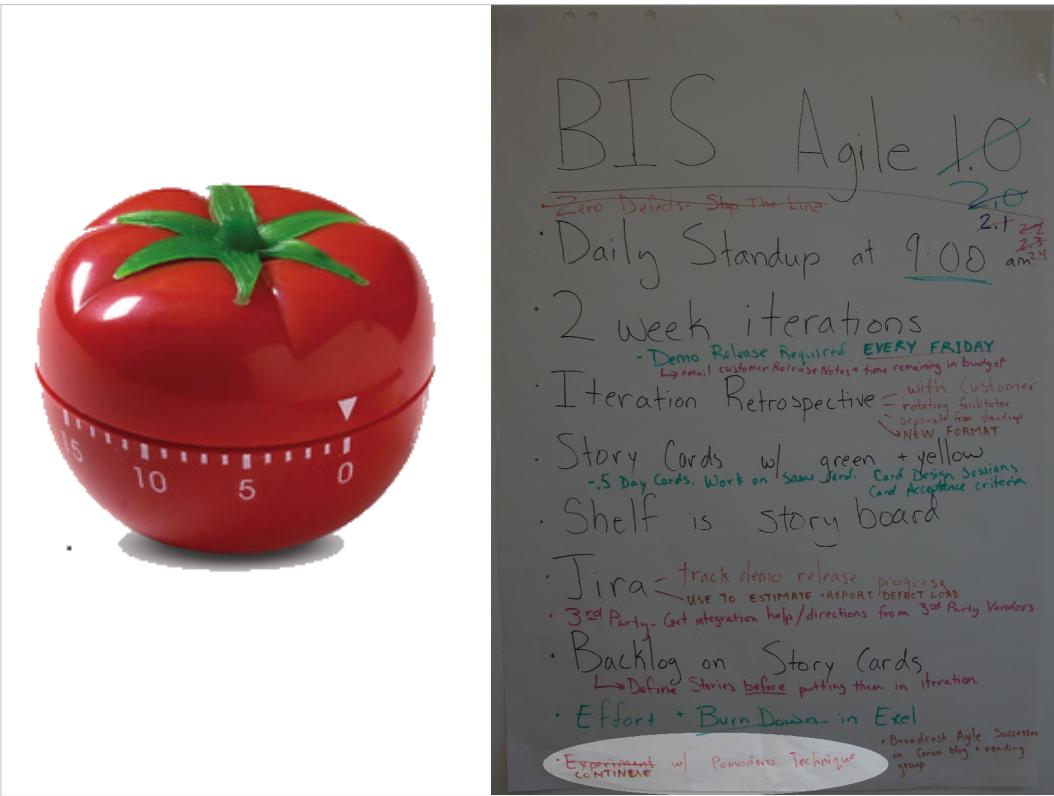




A real release to demo kept us honest about "sashimi". It was available to the customer but not a production deployment.

Tracking work in a tool (JIRA) made the release process easy.

Would like to automate the demo deployment.



Pomodoro Technique:

Divide the next 2 hours into 4 25-minute slices.

Plan out on paper what you hope to accomplish.

Set a timer for 25 minutes and work without interruption or distraction

Take a 4-5 minute break.

Repeat until done, then take a 15-20 minute break.

Idea is to focus without getting too tired.

Planning aspect is the most helpful.

This was an interesting experiment. Some of us are continuing with it.

Some things that did
not work great...

The screenshot shows a spreadsheet with the following data extracted:

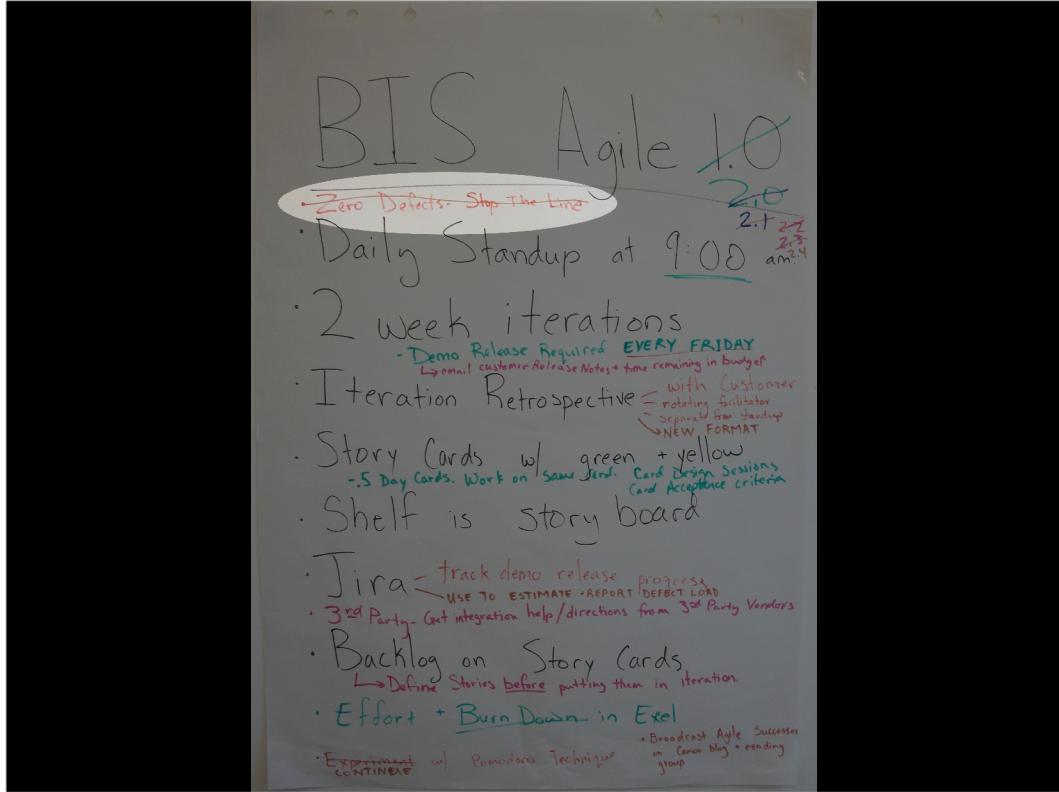
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Date 24.03.10															
2		Iter 0	Iter 1		Iter 2		Iter 3		Iter 4		Iter 5		Iter 6		Iter 7	
3		04.01	11.01	18.01	25.01	01.02	08.02	15.02	22.02	01.03	08.03	15.03	22.03	29.03	05.04	
4	Capacity (PD)	10	10	10	10	10	10	9	6	10	4	10	9	8	8	
5	Used	10	10	10	10	10	9	6	10	4	10	3			124	
6																
7																
8	Tasks in offer	Budget	Booked	Req.												
9	Gen. Improv	20		3.5												
10	GWT 2.0 Ref	19		0.5												
11	Accordion	12		0												
12	Sorting	7		0												
13	Switch Axes	14		0												
14	Results component	19		2.5												
15	Freeze, Rows	5		3												
16	Test / Perf.	9		3.5	Max											
17	Management	7		2	Mockups, Planning, Retrospectives											
18	Total	112	102	15	117	-5%										
19																
20																
21	Technical	52		9.5												
22	Functional	53		3.5												
23	Management	7														
24																
25																
26																
27																
28	Search	15	2	7												
29																
30																
31																
32																
33																

"All projects are fixed bid. Some bids are just more visible than others."
-David Hussman

Managing a $\frac{1}{2}$ fixed bid and $\frac{1}{2}$ T&A project was difficult.

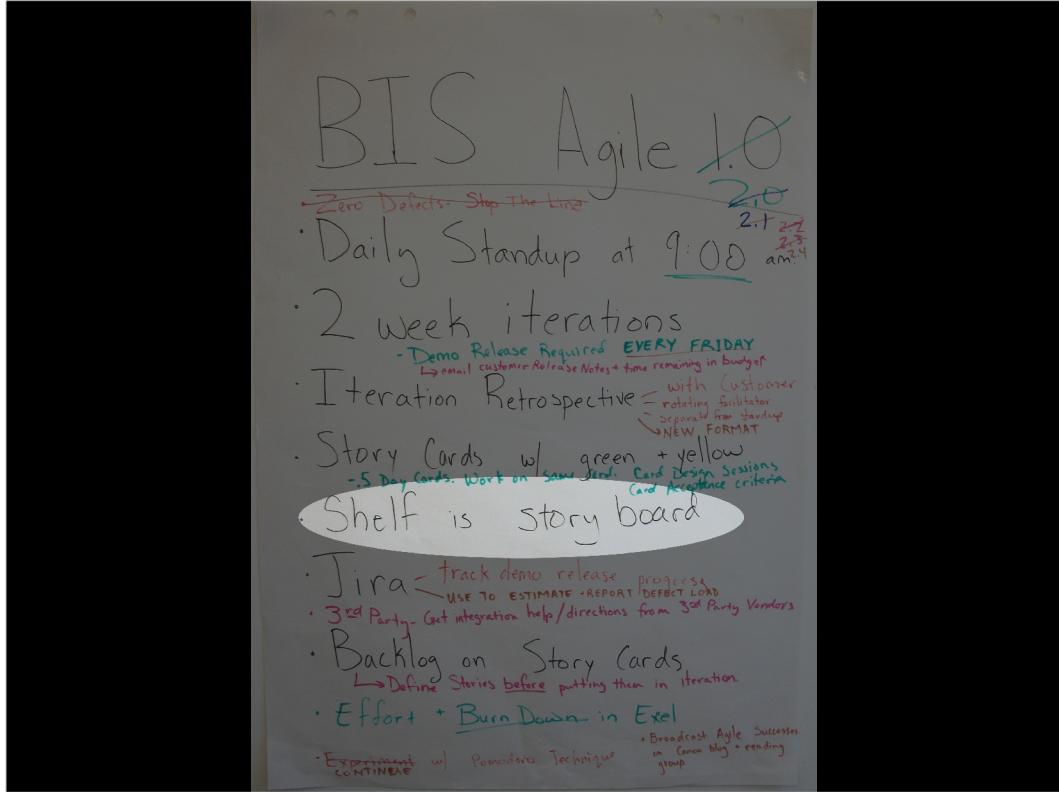
As always, there is a need to track your time in minute detail, which is a hassle.

Also, we had duplication between the story cards and our “burndown” spreadsheet. But a tool would be too heavy of a process.



We tried a zero defect stop the line process for one iteration.

It was too rigid. Some defects do not need to be fixed.



Taking more time for a proper workspace and information radiator would have been an improvement.

Creating a big, visible chart is important but easy to forget and easy to do wrong.

~~LEARN~~

- Broadcast Agile Successes
on Canoo blog + reading
group

Our last retrospective note was to do broadcast our successes on the Canoo blog and to groups like this.