



UNIVERSITÀ DI PISA



ISTITUTO DI SCIENZA E TECNOLOGIE  
DELL'INFORMAZIONE "A. FAEDO"



**HIIS Laboratory**

The Human-Computer Interaction Group

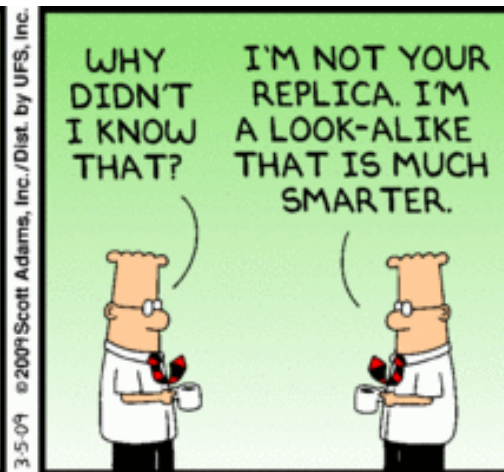
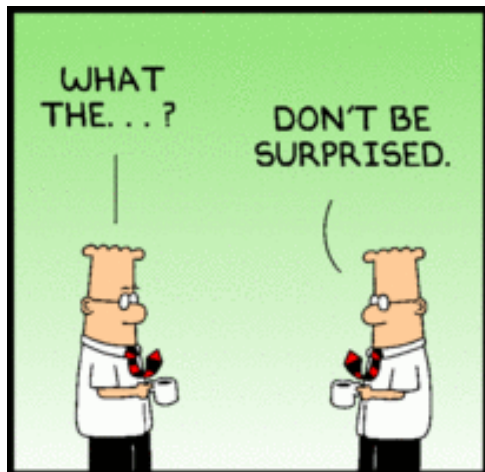
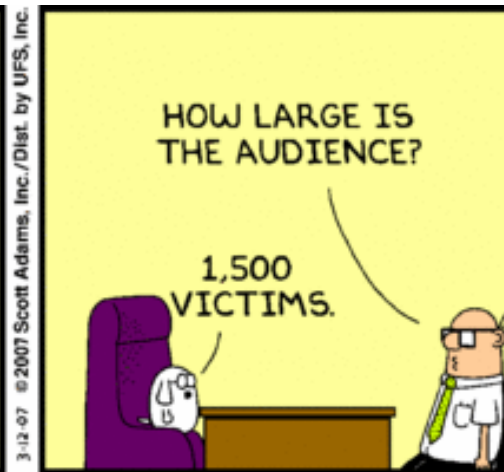
# Lean and Agile Development With Scrum (Part 2)

**Lucio Davide Spano**

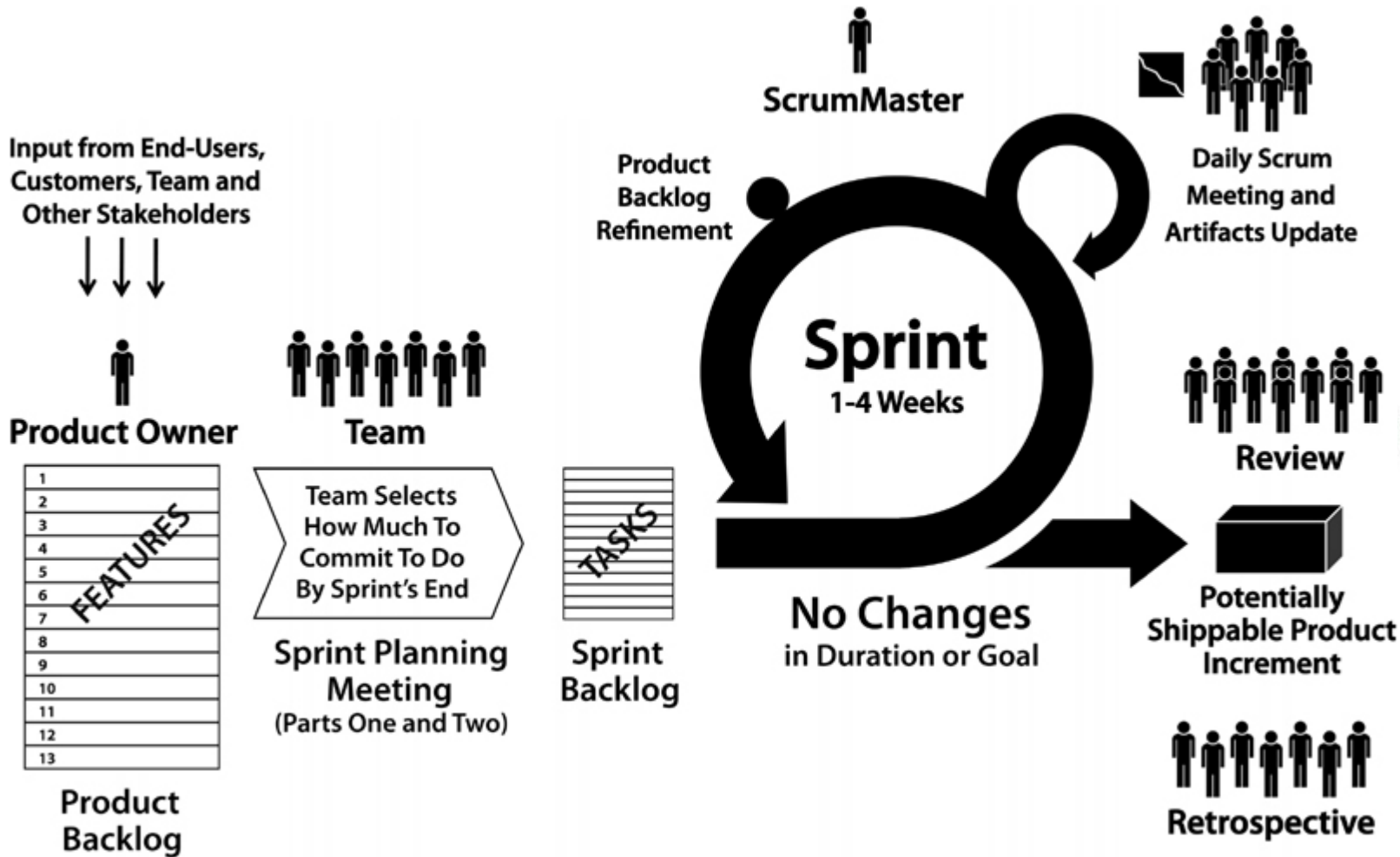
lucio.davide.spano@isti.cnr.it  
spano@di.unipi.it

7 May 2012

# Dilbert intro



# Summary



# Sprint Review

- Done at the end of the Sprint
- Not a simple demo
- Inspect and Adapt
  - **P.O.** learns what is going on the development of the product
  - The **Team** learns what is going on with the market
- The point is not only to show what has been done
- Collaboration for improving the creation process
- Scrum Master reports on the items that met the definition of done
  - No cheating!



# Sprint Retrospective

- Again inspect and adapt
- The subject is the *process*
- Participants
  - Team
  - Scrum Master
  - Product Owner (only if invited)
- What is working well
- What could work better
- All members should express their opinion

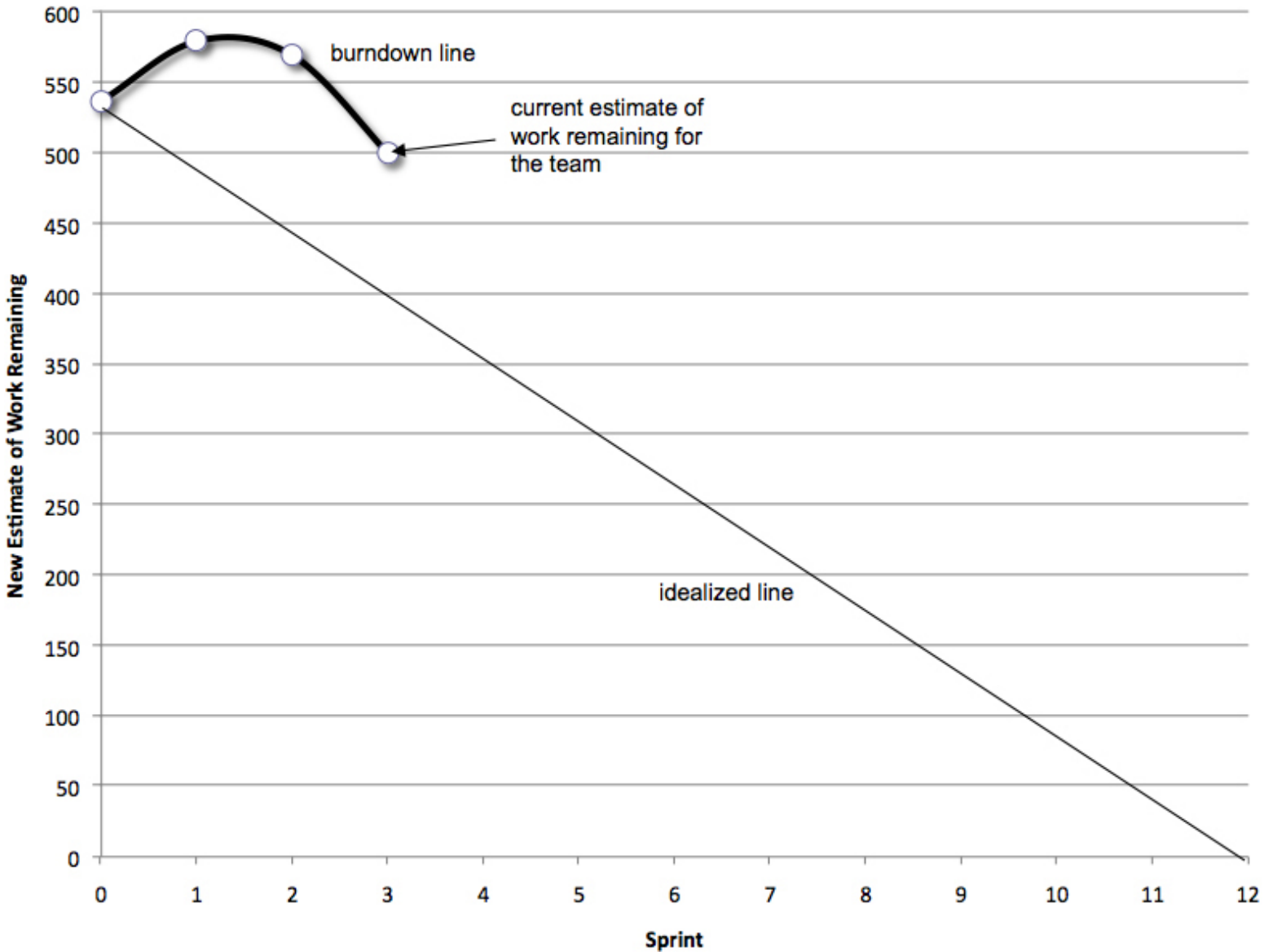


# Release Backlog



Item	Details (wiki URL)	Priority	Estimate of Value	Initial Estimate of Effort	New Estimates of Effort Remaining at end of Sprint...				
					1	2	3	4	5
As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	...	1	7	5	0	0	0		
As a buyer, I want to remove a book in a shopping cart	...	2	6	2	0	0	0		
Improve transaction processing performance (see target performance metrics on wiki)	...	3	6	13	13	0	0		
Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	...	4	6	20	20	20	0		
Upgrade all servers to Apache 2.2.3	...	5	5	13	13	13	13		
Diagnose and fix the order processing script errors ( <a href="#">bugzilla ID 14823</a> )	...	6	2	3	3	3	3		
As a shopper, I want to create and save a wish list	...	7	7	40	40	40	40		
As a shopper, I want to to add or delete items on my wish list	...	8	4	20	20	20	20		
...				...	...				
			<b>Total</b>	<b>537</b>	<b>580</b>	<b>570</b>	<b>500</b>		

# Release Burndown Chart





# Starting next Sprint



Input from End-Users,  
Customers, Team and  
Other Stakeholders



Product Owner

1
2
3
4
5
6
7
8
9
10
11
12
13

FEATURES

Product Backlog



Sprint

(Product Increment)

# Repeat Forever



Daily Scrum Meeting and  
Daily Facts Update



Review



Potentially shippable Product Increment



Retrospective





# Scrum in Large Companies



- Excel sheet from Scrum Alliance
  - At least Adobe, Google and Yahoo
  - A deeper look into the file
- Report on Scrum adoption at Adobe
  - Green, P. Measuring the Impact of Scrum on Product Development at Adobe Systems (HICSS 2011)
  - 1/3 of Adobe uses Scrum (main development method)
  - But... some of them adopted only the name!
  - Good point: release defects and customer satisfaction increases for products developed by Scrum Teams (real ones)
  - Bad point: hand-off acquisitions

# Problems with Scrum

## ■ Robert Martin (Object Mentor Inc.)

1. No technical practices
2. 30 day Sprints are too long
3. Tendency of Scrum Masters to arrogate project management powers
4. C for Certified Scrum Masters
5. Insufficient guidance for the backlog structure
6. Anti-management bias
7. Automated testing not included in practice (see point 1)
8. No real consensus on Large-scale Scrum

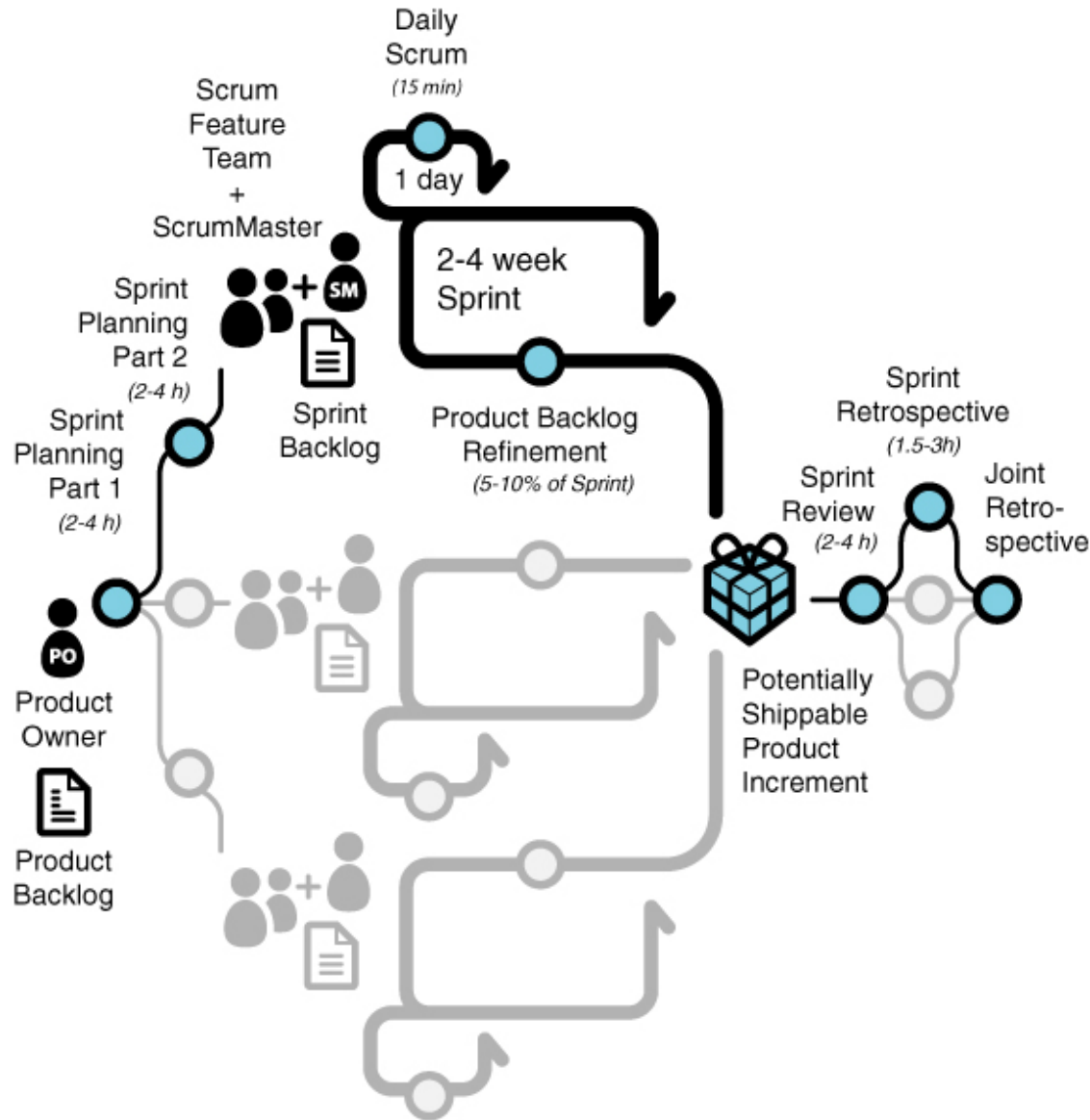


# Large Scale Scrum

- Is Scrum good for big development companies?
- How to manage different Teams?
- What if the Product Owner cannot manage the entire product alone?
- Two different frameworks proposed by Craig Larman
  - 1 P.O. and up to 10 teams
  - Area P.O., over 10 teams
- It is not an improvement of Scrum
- A set of tools



# Large Scale FW 1



# Elements (1)

## ■ Sprint Planning Part 1

- One for all teams
- All teams members if they can fit the room
- Team representative otherwise

## ■ Sprint Planning Part 2

- One for each team in parallel

## ■ Daily Scrum

- Allocate them at different times
- Members of other teams can join if interested

## ■ Product Backlog Refinement

- Again all team members (or representatives)

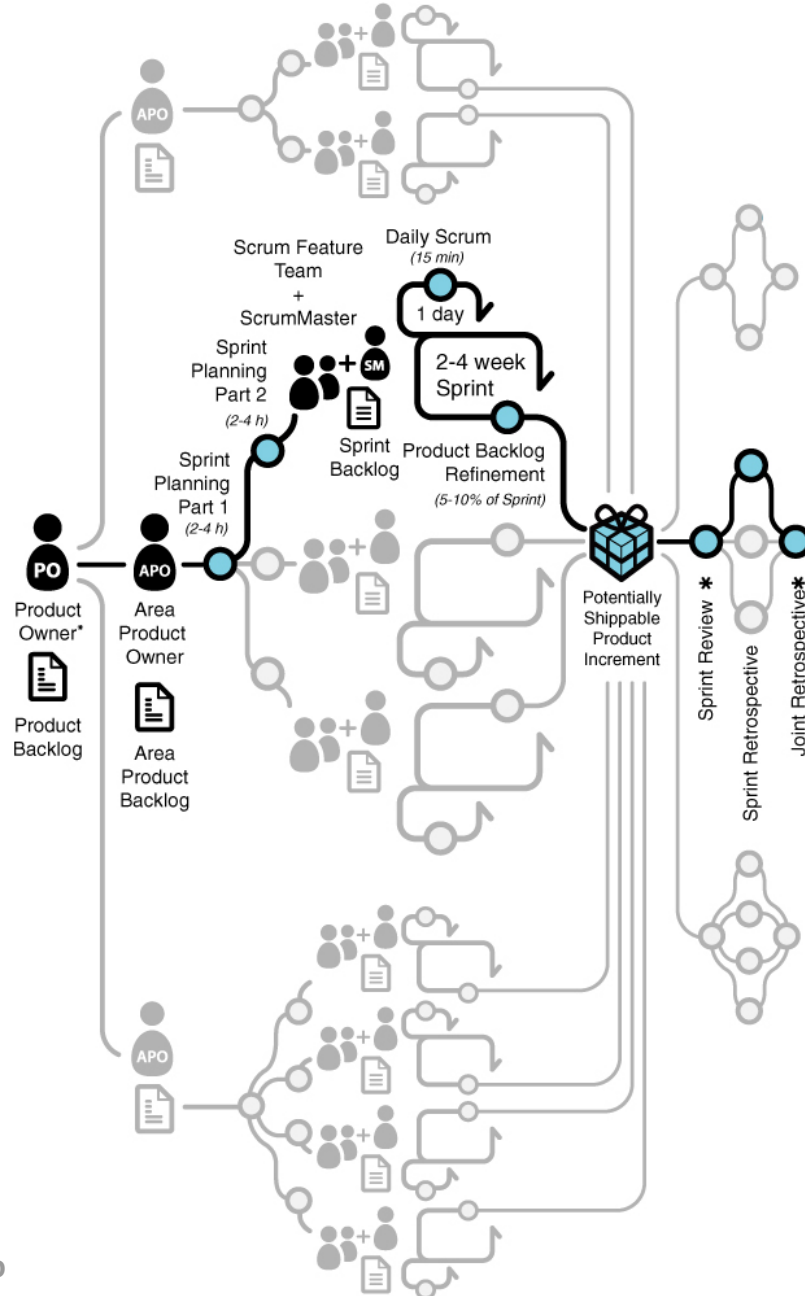


# Elements (2)

- **Sprint Review**
  - One for all teams
- **Sprint Retrospective**
  - One for each team
- **Joint Retrospective (optional)**
  - Few representatives share the experiences
- **Definition of done**
  - It applies for all teams involved
  - Maintained into a wiki page all teams refer to
  - Joint Retrospective for changes
- **Continuous integration**
  - Shifted from planning to code.
  - Unique PSPI



# Large Scale FW 2





# Elements (1)



## ■ Product Owner

- Identify major requirements areas
- Divide the Product Backlog in Area Backlogs
- Area Product Owners (Product Owner Team)

## ■ Pre-Sprint Planning

- The Product Owner discusses with the Area Owner the priorities
- Decide together where the entire product should go

## ■ Sprint planning

- Separate meeting for each requirement Area
- Area Product Owner and Teams of that area

## ■ Product Backlog Refinement

- Separate refinement for each Area
- Area Product Owner and teams

# Elements (2)

## ■ Sprint Review

- Separate Sprint Review for each Area
- Teams and Area Product Owner
- Product Owner participates when interested

## ■ Joint Review

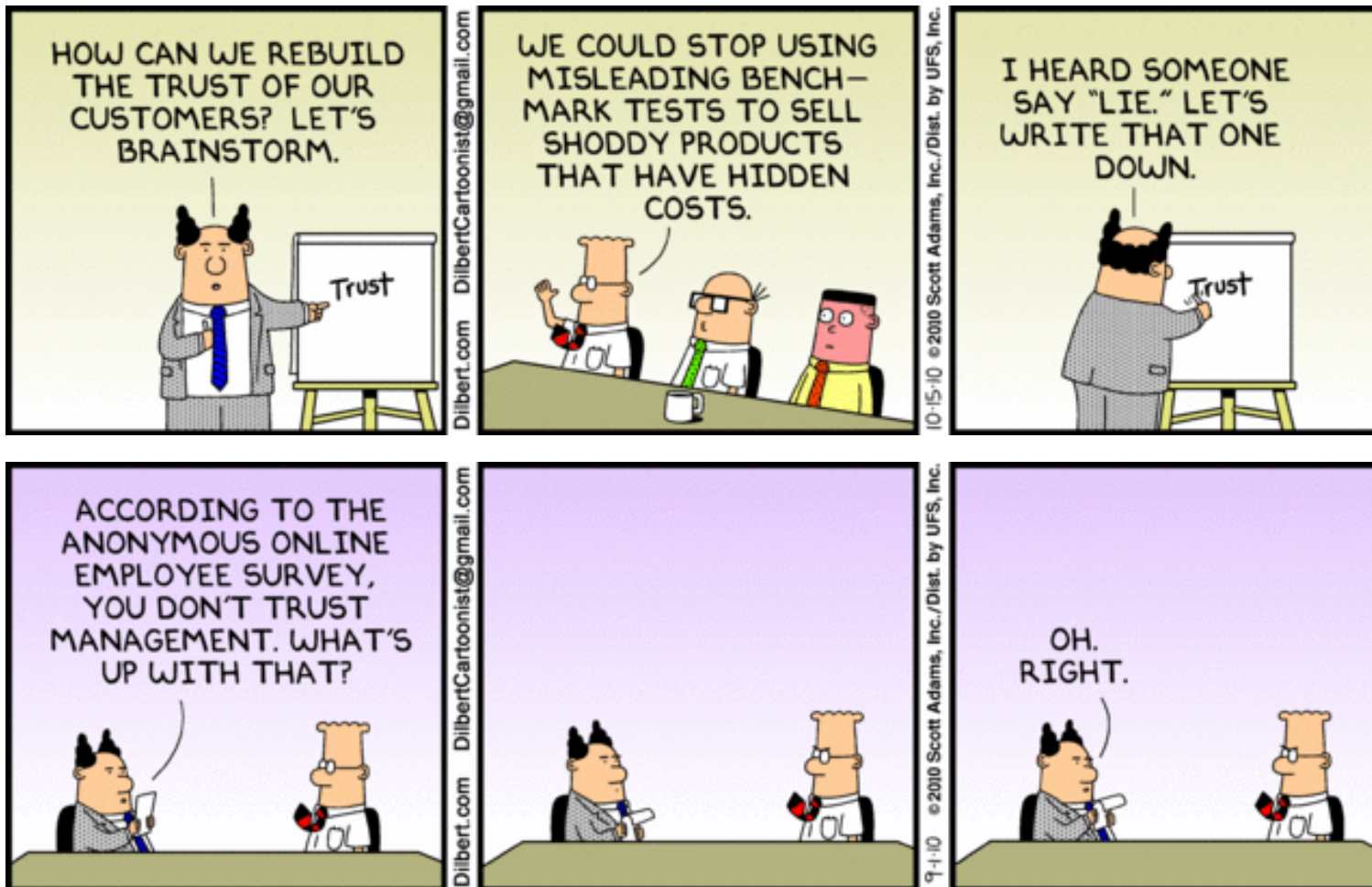
- Review on things that are crucial for the next release
- Representative from various Teams

## ■ Joint Retrospective

- Discuss system-level learning and improvement points
- It can happen at both area and overall product level



# Dilbert again...



<http://www.dilbert.com>

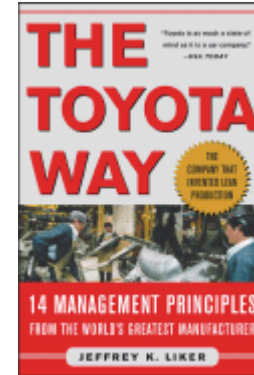
# Back to the origins



The Toyota way:  
***Watch the baton, not the runners!***

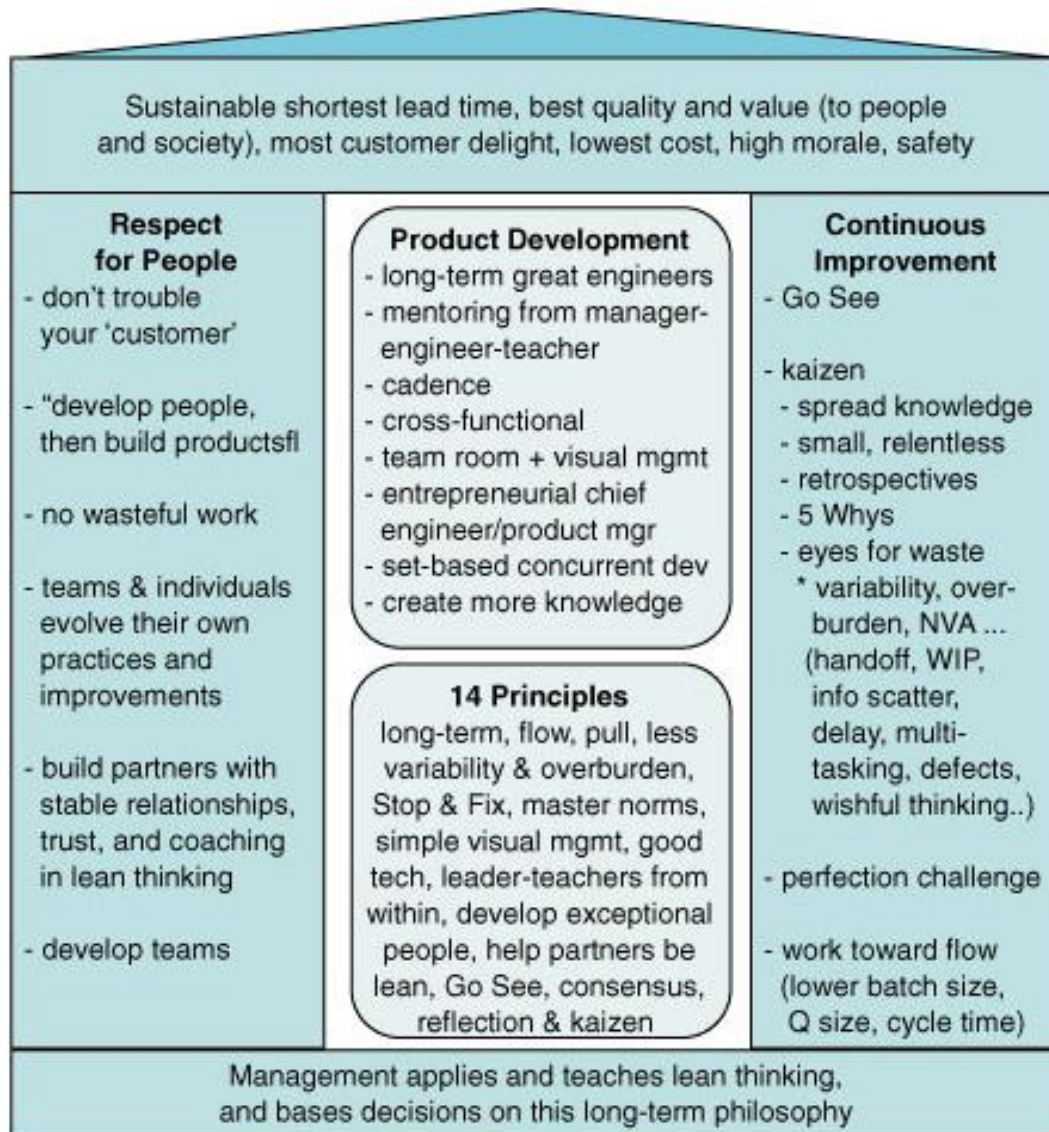
# Lean Thinking

- Description of the Toyota system (1990-2006)
- *Lean Production VS Mass Production*
- Try to compete on:
  - Ability to Adapt
  - Avoid of Inventory
  - Work in small units
- Against
  - Compete on scale economy
- Agile and Scrum have their foundation on Lean Thinking





# The Lean Thinking House

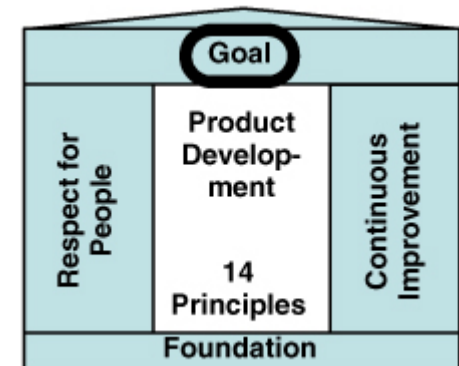


# Lean Goal



*Sustainable shortest lead time, best quality and value (to people and society), most customer delight, lowest cost, high morale, safety*

- Try to shorter the cycle times
  - With Respect of people
  - With Continuous improvement
  - Without reducing quality
  - Without working at unsustainable pace
  
- Remove bottlenecks
  - Avoid local optimization!

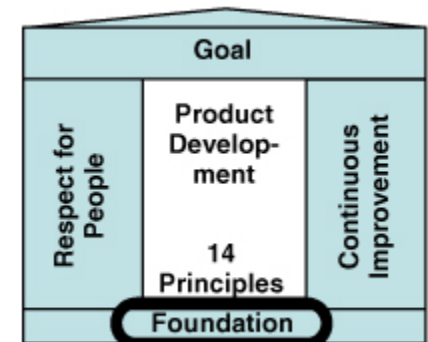




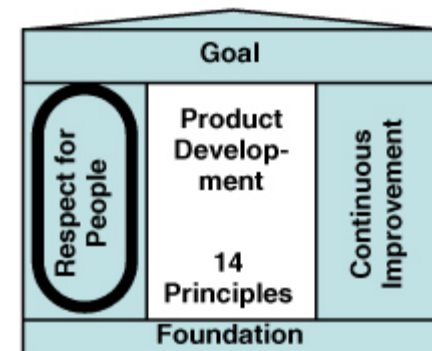
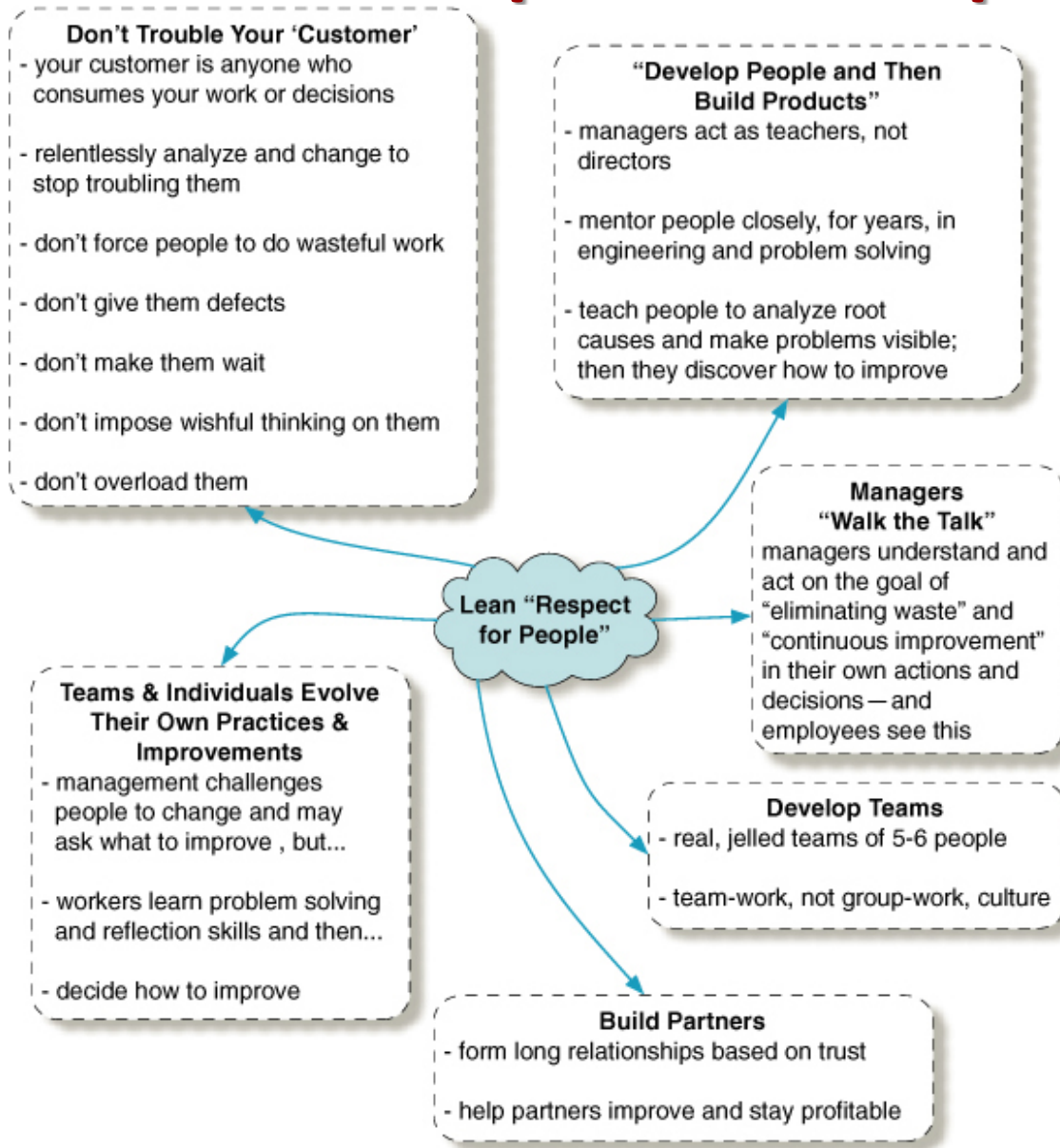
# Lean Foundation

*Management applies and teaches lean thinking, and bases decisions on this long-term philosophy*

- New people learn from the others
  - See the whole
  - Apply the lean thinking in different domains
  - Continuous improvement
  - Go and See VS relay on report numbers
- Train people to think themselves (*spend time teaching and mentoring others*)
- *Stop and fix*
- Long-term advantages



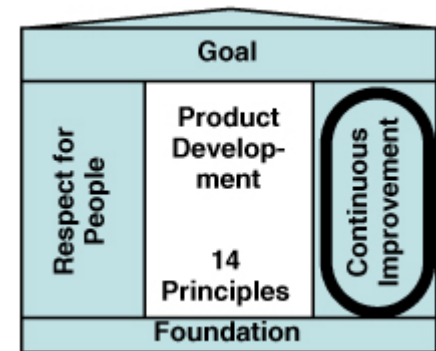
# Pillar One: Respect for People



# Pillar Two: Continuous Improvement

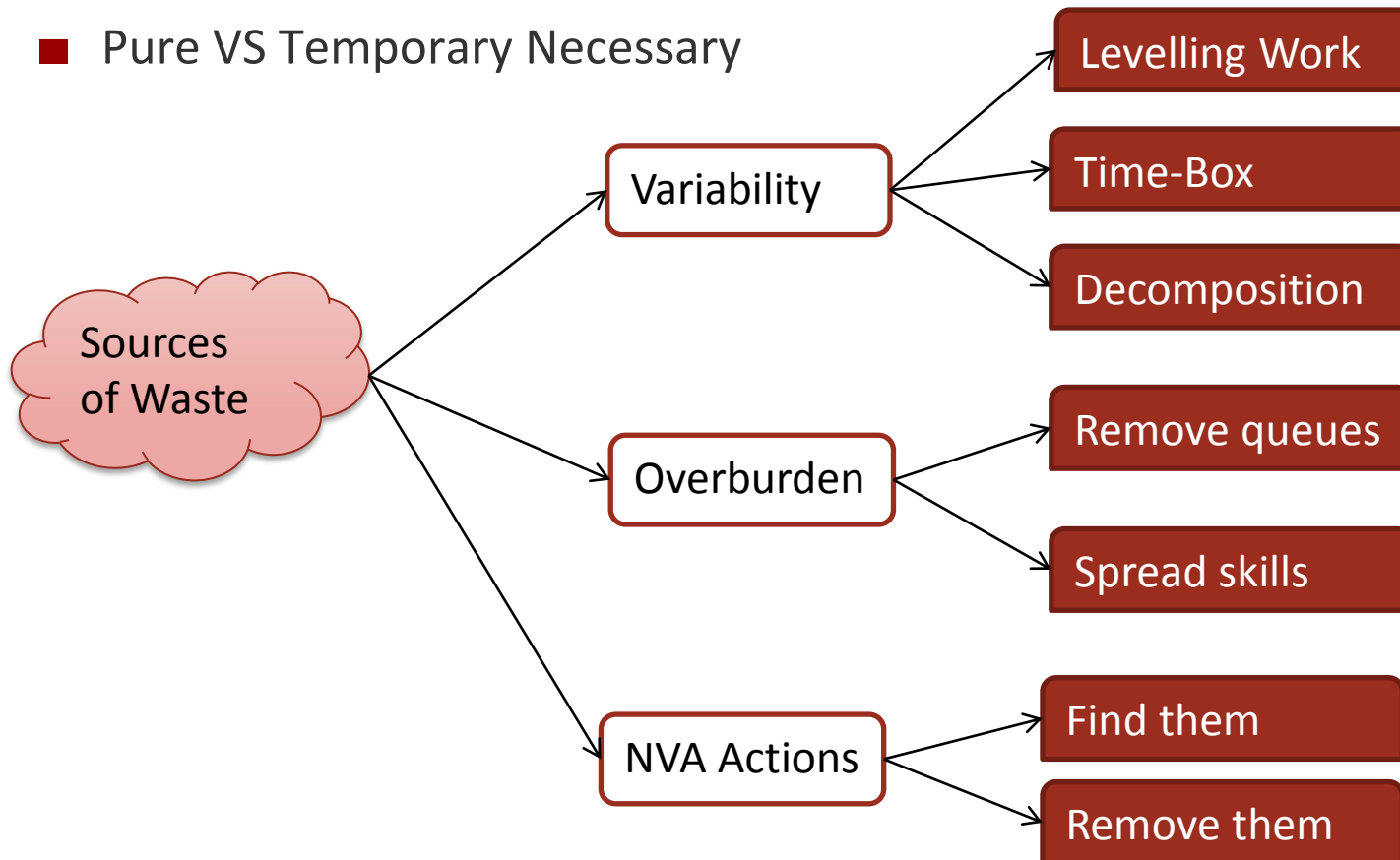


- Go see for yourself
  - Do not receive information only via reports
  - Go where the work is done and see it
  - Understand problems and opportunities
- Improve for improvement's sake, endlessly
  - Choose techniques the team agreed to try, until they are well understood
  - Experiment until you find a better way
  - Repeat forever
  - **Scrum retrospective, Daily scrum ...**
- Perfection challenge



# Value and Waste

- **Value:** The moments of action or thought creating the product that the customer is willing to pay for.
- **Waste:** All other moments or actions that do not add value but consume resource.
- Pure VS Temporary Necessary



# Removing Waste



## Non Value Adding Actions

Overproduction of features, elements ahead of the next step, duplication

Waiting, delay

Handoff, conveyance, moving

Extra processing (includes extra processes), relearning, reinvention

Partially done work, work in progress (WIP)

Task switching, motion between tasks; interrupt-based multitasking

Defects, testing and correction after creation of the product

Under-realizing people's potential and varied skill, insight, ideas, suggestions

Knowledge and information scatter or loss

Wishful thinking (for example, that plans, estimates, and specifications are 'correct')

# 14 Principles



## Principles

Base management decisions on a long-term philosophy, even at the expense of short-term financial goals.

Move toward flow; move to ever smaller batch sizes and cycle times to deliver value fast & expose weakness.

Use pull systems; decide as late as possible.

Level the work—reduce variability and overburden to remove unevenness.

Build a culture of stopping and fixing problems; teach everyone to methodically study problems.

Master norms (practices) to enable kaizen and employee empowerment

Use simple visual management to reveal problems and coordinate.

Use only well-tested technology that serves your people and process.

Grow leaders from within who thoroughly understand the work, live the philosophy, and teach it to others.

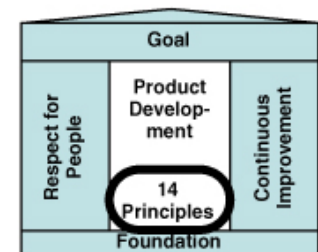
Develop exceptional people and teams who follow your company's philosophy.

Respect your extended network of partners by challenging them to grow and helping them improve.

Go see for yourself at the real place work to thoroughly understand the situation and help.

Make decisions slowly by consensus, thoroughly considering all options; implement rapidly.

Become and sustain a learning organization through relentless reflection and kaizen.



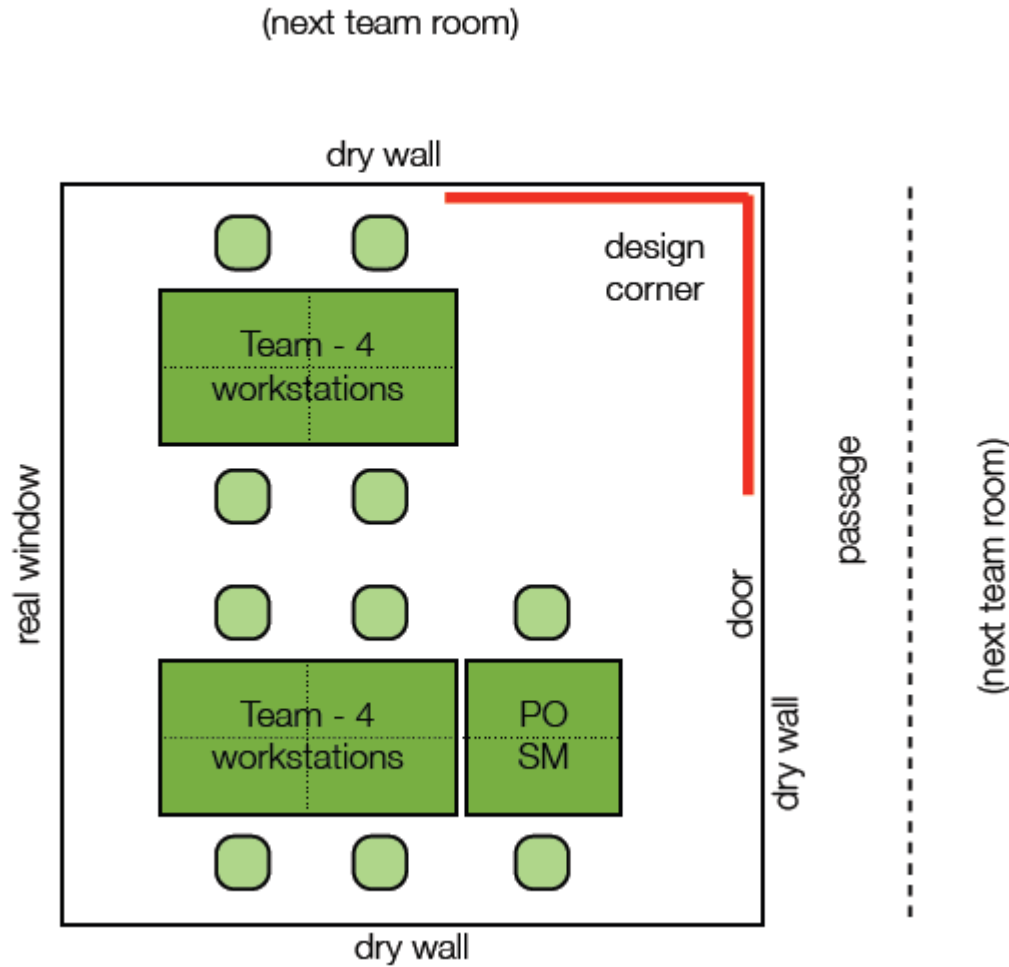


# Visual Management (1)

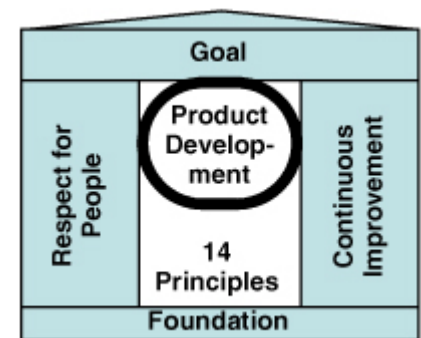
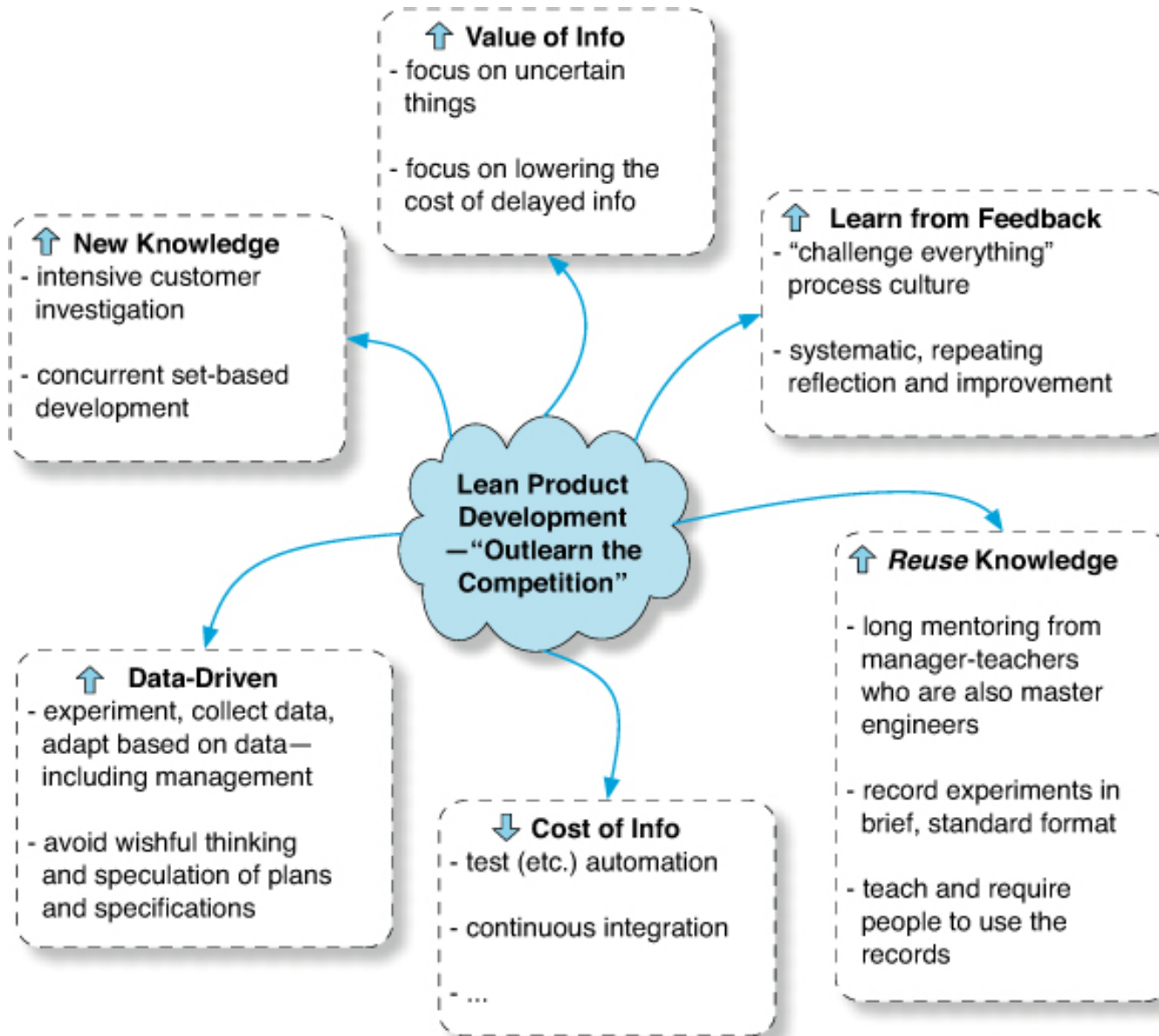




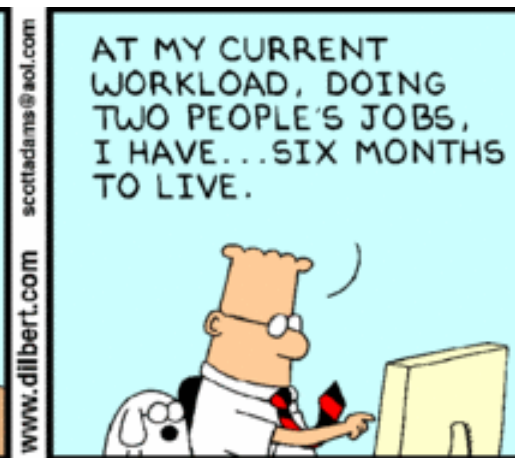
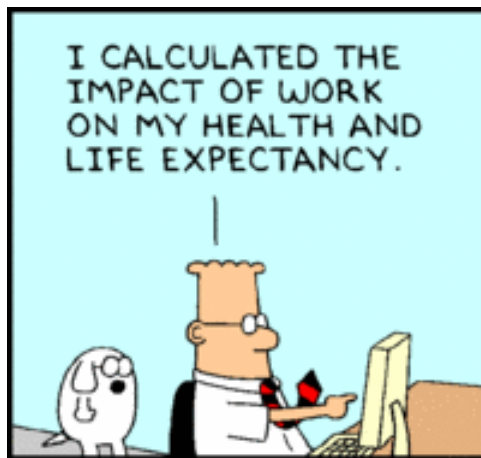
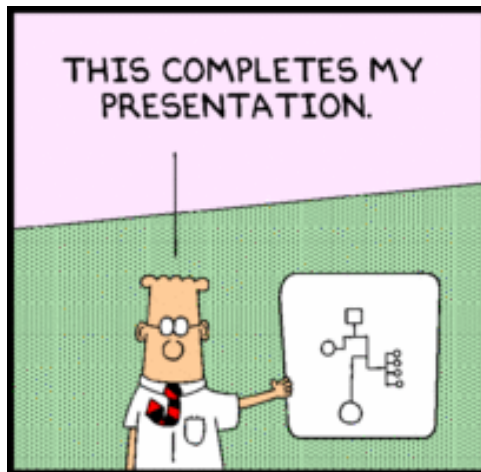
# Visual Management (2)



# Product Development



# Dilbert break



# Queuing Theory

- Queues exist not only in manufacturing, but also in product development
- Yes, *those* queues...
  - Same concept of variability of requests and service time
  - But we can't reuse *exactly* the same solutions
- Examples:
  - Products in portfolio
  - New Features for one product
  - Detailed requirement specification waiting for design
  - Design documents to be coded
  - Code waiting to be tested
  - Code waiting to be integrated

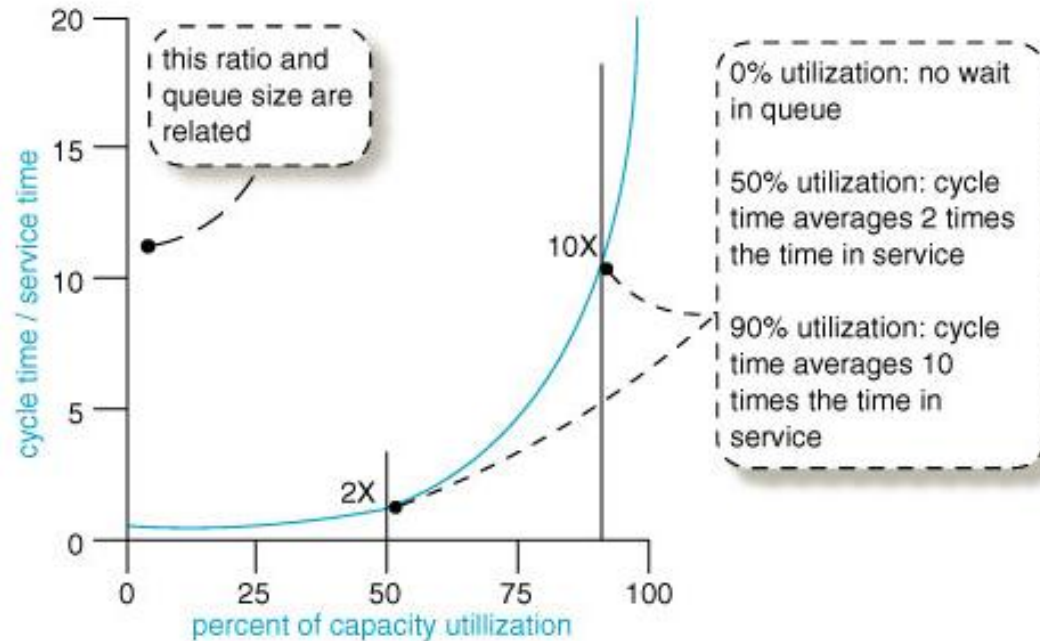


# Why queues are a problem

- **Work in Progress**
  - Increase average cycle time
  - Work with an investment of money without return (yet)
  - Hide defects
  - Affect the cost and the ability to react to change (replanning)
- **Shared Resource queues**
  - Slow the development
  - Delay feedback
  - Stretch cycle time
- They should be removed...
- Remove bottlenecks!



# The M/M/1 queue



- Request arrive at different times with different effort
- Test or programming effort take variable time
- People work faster or slower, get sick or work longer or shorter



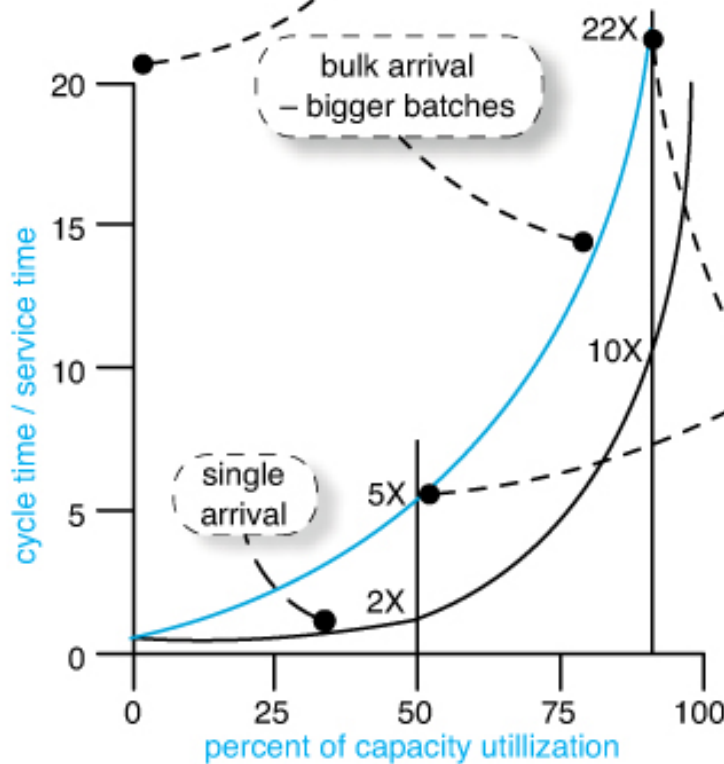
# Batches of work



ratio of cycle time (queue time + service time) divided by service time. for example:

- service time: 2 weeks      - queue time: 2 weeks  
- cycle time = 4              - **ratio = 2**

- service time: 2 weeks      - queue time: 8 weeks  
- cycle time = 10             - **ratio = 5**



**Analogy to Traditional Product Development with Big Batches**

50% utilization: cycle time averages 5 times the time in service

90% utilization: cycle time averages 22 times the time in service





# So what?

- Local optimization: keep the workload high
  - Simply does not work
- Try to reduce variability:
  - Tasks of almost the same size
  - Keep the same utilization level of people
- Remove WIP with handoff
  - Waterfall approach
- Variability at the front end of multi-step queued system has the worst impact
  - Requirement engineering
- Do not hide the queues you have
  - Visual management



# References

- The Scrum Guide  
<http://www.scrum.org/scrumguides/>
- Do Scrum Better  
<http://www.scrumsense.com/resources/do-better-scrum>
- Scrum Primer  
<http://www.scrumprimer.com/>
- *Craig Larman, Bas Vodde*  
Scaling Lean and Agile Development  
Pearson, 2008
- Kicking ScrumBut  
<http://www.scrumalliance.org/resources/1122>



# Questions?



**Thank you!**

